SEMANTIC DEPTH ESTIMATION WITH MONOCULAR CAMERA FOR AUTONOMOUS NAVIGATION OF SMALL UNMANNED AIRCRAFT

Tay Yong Kiat

Submitted in fulfilment of the
requirements for the Degree of
Master of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

June 2023

Acknowledgements

I would like to express my deepest gratitude and appreciation to the following individuals and institutions who have contributed to the completion of this thesis:

First and foremost, I am thankful to my supervisors, Dr. Henrik Hesse, Dr Sutthiphong Srigrarom and Dr Dave Anderson for their unwavering support, guidance, and expertise throughout the entire research process. Their valuable insights and constructive feedback have been instrumental in shaping the direction of this study.

My heartfelt appreciation to Dr Henrik Hesse, who unconditionally took on the role as my primary supervisor in the middle of my studies and persistently encouraged and guided me to complete despite all the challenges I had faced through the Covid-19 period, at work and at home. Without him, I may not have the determination to complete my course of studies.

Furthermore, I wish to acknowledge this part-time studies sponsorship provided by the University of Glasgow and Republic Polytechnic. Their generosity has significantly alleviated the financial burdens associated with this research.

I am deeply grateful to my colleagues and bosses at Republic Polytechnic for their unwavering support, encouragement, and patience throughout this challenging journey. Their belief in my abilities had been a constant source of motivation and strength.

Finally, a big thank you to my wife, Serena for her patience and sacrifices she had made over the past few years to take care of our 3 children, Celeste, Zenith, and Caleb when I had to reprioritize my time away from them to complete key research milestones. Their belief in my abilities and their unconditional love have sustained me throughout this journey. I am forever grateful for their presence in my life.

In conclusion, this thesis is a product of the collective efforts, support, and contributions of many individuals and institutions. While I have endeavored to mention everyone, I apologize if any names have been inadvertently omitted. Your support has been instrumental in this achievement, and I am sincerely grateful to every one of you.

Abstract

Demand for small Unmanned Aircraft (UA) applications in Global Navigation Satellite System (GNSS) denied environment has increased over the years in areas such as internal building infrastructure inspection, indoor security surveillance and stock cycle counting. One of the key challenges in the current development of autonomous UA is the localization and pose estimation in the absence of GNSS signals. Various methods using onboard sensors such as Light Detection and Ranging (LiDAR) have been adopted but with the compromise of take-off weight and computing complexity. Off-board sensors such as motion trackers or Radio Frequency (RF) based beacons have also been adopted but are costly and limited to a small area of operations within the sensor's range. With the advancement of computer vision and deep neural networks, and the fact that the majority of consumer and commercial UA comes equipped with high resolution cameras, it is now even more possible to exploit camera images for navigational tasks. To enhance the accuracy of traditional computer vision methods, machine learning can be adopted to model complex image variations for more accurate predictions. In this thesis, a novel approach based on Semantic Depth Prediction (SDP) was proposed for small UA to perform path planning in GNSS denied environments using its onboard monocular camera. The objective of SDP is to perform 3D scene reconstruction using deep convolution neural network using 2D images captured through a single forward-looking onboard camera thus eliminating the use of expensive and complex sensors. SDP was modeled based on open-source image data set (like NYU2 and SunRGB-D) and real image data sets taken from the actual environments to improve of detection accuracy and was tested in an actual indoor warehouse to validate the performance of the proposed SDP concept. Our experiments have shown that combining lightweight mobile Convolutional neural network (CNN) models allows feature tracking navigation tasks to be undertaken by an off the shelve Tello without the need for additional sensors. However, features of interest need to be kept within the center of each frame of image to eliminate the possibility of losing feature of interest over time. Missing objects in SDP output can be linked to partially occluded objects captured in the input image as existing networks are not able to handle missing information and thus cannot detect objects under occlusion.

Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

The demand for autonomous Unmanned Aircraft (UA) applications in Global Navigation Satellite System (GNSS) denied indoor environments have increased over the recent years from performing logistical tasks in warehouses to monitoring of crop's health in large urban greenhouses [1]. Studies have also shown that there are growing demands for logistic companies to adopt autonomous UA for inventory management applications in large indoor warehouses [2] where GNSS is not available as shown in Figure 1.1. One of the key challenges in achieving full autonomous capabilities for obstacle rich indoor environments is the ability to perform precise and reliable pose estimation, avoidance collision and path planning in the absence of GNSS. Weight and size of the indoor UA system is another important safety consideration when operating in confined spaces that are populated with high human traffic or expensive stock. Safe and practical indoor UA applications cannot be achieved without overcoming these challenges. It is therefore crucial to develop an efficient and versatile navigation solution that is suitable to meet the demands of real-world application.

1.1 Research Motivation

There have been numerous developments in the navigational backbone for autonomous indoor UA solutions over the last few decades with no perfect solution to suit all types of indoor applications and environments. The success of each solution depends on several factors ranging from environmental conditions to types of infrastructure available in respective commercial applications. For example, developments in localization methods using off-board beacons such as Ultra-Wide Band (UWB) technology [3–6] have been explored for trajectory planning. This method utilizes multiple low powered, high bandwidth UWB sensors that form a mesh network to triangulate the position of the vehicle. Although this radio frequency-based technology can produce a high position accuracy and allows the UA to operate beyond line of sight from the ground control station, it is however a costly solution due to the need for extensive numbers of UWB sensors to cover large areas and operations are limited within the boundaries of the pre-installed UWB sensors.

It can also interfere with other existing systems that operate within the ultra-wide band spectrum.

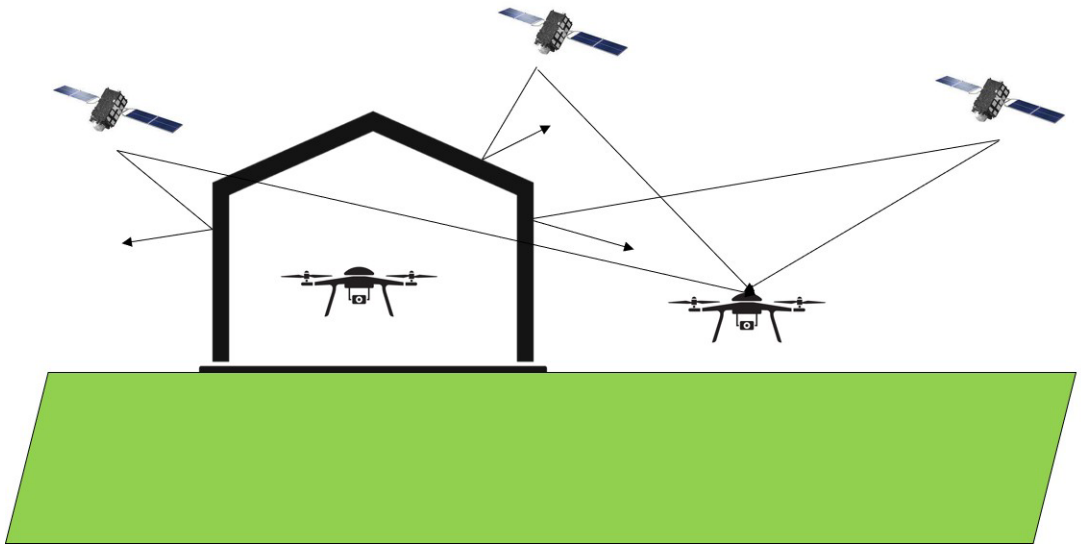

Figure 1.1: Challenges of indoor UA navigation with the absence of GNSS due to line-of-sight issues with GNSS satellite signals

Simultaneous Localization and Mapping (SLAM) [7] is another method of enabling an autonomous system to continuously map and simultaneously find its location in real time with respect to its surroundings. Figure 1.2 shows a typical SLAM architecture where it consists of a sensor which inputs sensor data to a front end for sensor dependent processing. Maps are continuously updated on the move while planning its trajectory to its new location without prior knowledge of the environment. This method is also able to react to sudden obstacles appearing in the intended path and generate new path. SLAM has been widely adopted in the consumer world for applications such as ground cleaning vacuum robots to self-driving cars where self-exploratory capabilities are necessary. There are 2 categories of sensors used for SLAM.

The use of Light Detection and Ranging (LiDAR) sensors for SLAM is a technique commonly used by many for autonomous navigation particularly when in tight confined spaces with limited lighting conditions and where feature detection is not necessary. LiDAR sensor measures distance of its laser light projection by calculating the difference in the laser return timings and wavelengths to create 3D point cloud of an object or an environment. LiDAR sensors have been predominantly used to generate real time 3D point cloud data for feature extraction of the environment and refining the state estimation with onboard inertial data [8, 9].

The relative position parameters derived from environmental features and the differences of measurements from the LiDAR at adjacent times were used to estimate and correct the errors from current navigational sensors.

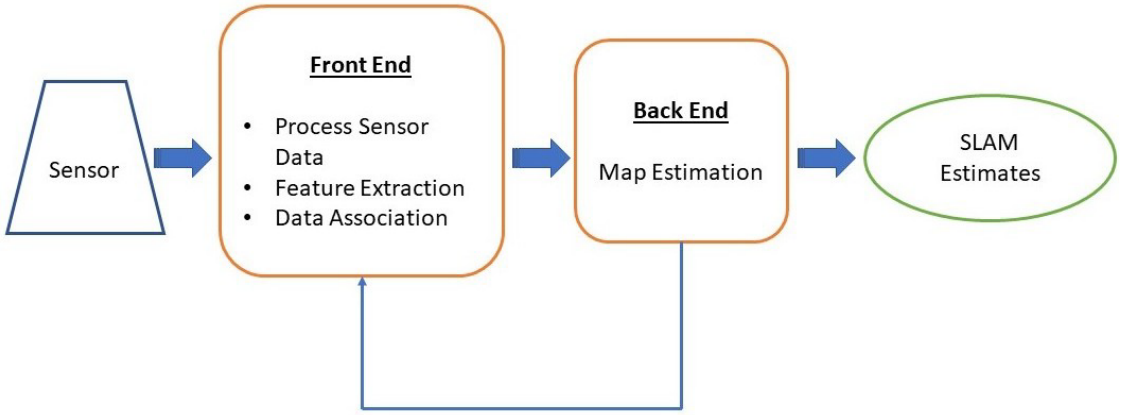

Figure 1.2: Typical SLAM Architecture

This method proves to be highly accurate but at a disadvantage due to high cost and high computational requirements from the large amount of 3D point cloud data points from LiDAR sensors. LiDAR sensors are relatively heavy and bulky which may be suitable for medium size UA that can accommodate a heavier take-off weight and computing complexity as shown in Figure 1.3. However, it is not practical for small UA weighing less than 100 grams to carry the LiDAR sensors due to its payload carrying limitation.

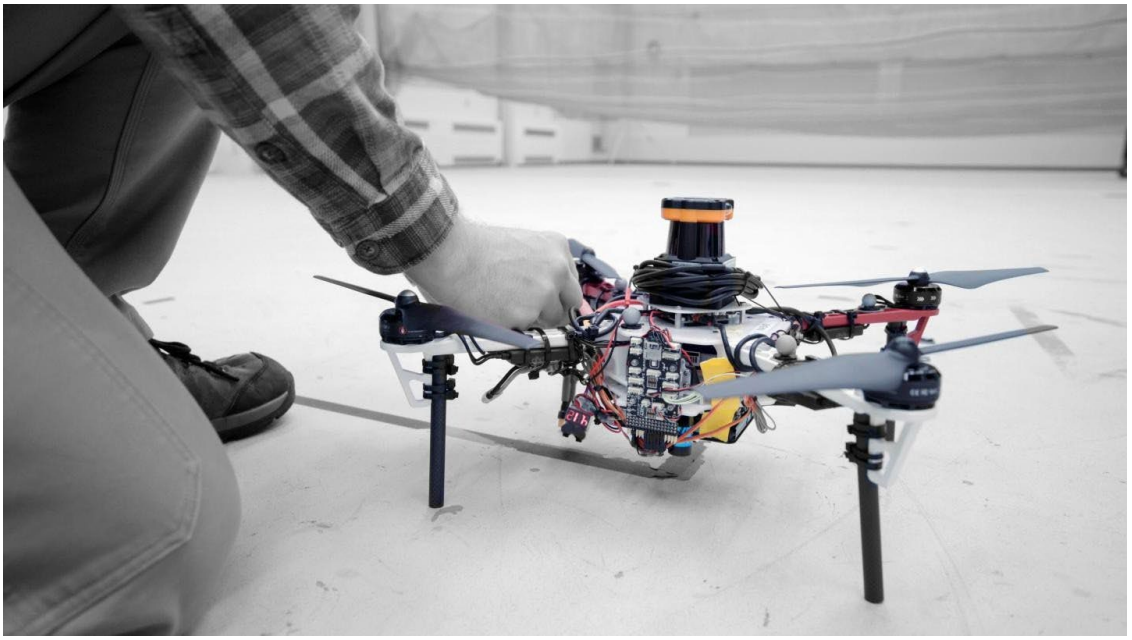

Figure 1.3: Example of a LiDAR SLAM Autonomous Multirotor developed by MIT. Source:https://news.mit.edu/2018/mit-csail-programming-drones-fly-face-uncertainty-0212

With the advancement in camera and graphics processing units (GPU) technology, computer vision approach has been a popular alternative for mobile robotics and even autonomous vehicles to achieve precise localization and pose estimation by detecting objects or obstacles through feature extraction and background noise omission. Recent surveys [10–14] indicated a growing popularity with this approach which is also known as Visual Simultaneous Localization and Mapping (VSLAM) for autonomous navigation of indoor UA in the absence of GNSS. Advancement in computer vision technologies provided several advantages and benefits leading to low cost and lightweight navigation systems which are important for small UA like the one shown in Figure 1.4. Another benefit for VSLAM approach is the ability to capture rich details of an environment with image data that is not only useful for navigational purposes but can also be used in conjunction for non-navigation applications such as surveillance, architectural, photogrammetry or infrastructure inspection purposes.

Figure 1.4: Example of a Micro UA with a monocular camera Tello Micro Drone
Source: https://www.ryzerobotics.com/tello

The challenge faced by this approach is primarily the ability to perform real time images. Processing onboard small UA without the need for high power demand to support computing resources. Depth perception is another area that can hinder the generation of 3D scene information to provide sufficient depth information of the environment. VSLAM utilizes camera-based sensors such as monocular cameras, stereo vision cameras, RGB-D cameras as shown in Figure 1.5 to produce feature rich visual maps enabling exploration of unknown environments. In addition to using forward-looking stereoscopic cameras, pose estimations can be further optimized by fusing additional sensors such as optical flow sensor and rangefinders like the Quanser QDrone 2 as shown in Figure 1.6 and Modal AI Qualcomm Flight RB5 as shown in Figure 1.7. This method however requires additional sensors and onboard processors apart from the primary stereoscopic cameras to perform autonomous navigation thus adding on weight to the small UA. Benefits of using visual base sensors address the issues over LIDAR base systems, allowing more cost-effective navigation approaches and reducing computational complexity at the same time. Despite progression in vision-based approaches, there are still present issues of fusing 2D images with other sensor information for optimal scene understanding. As such, RGB information obtained from the visual sensors need to be fused with data from existing onboard inertial sensors to provide aircraft pose estimation via an extended Kalman filter framework as explored by several VSLAM research works.

Figure 1.5: AVATAR HD PRO Monocular Camera, Stereolabs ZED 2i Stereo Vision Camera and Intel® RealSense^{TM} Depth Camera D455.

Figure 1.6: Quanser QDrone 2 equipped with Intel® RealSense^{TM} Depth Camera and Downward Optical Flow Camera Source: https://www.quanser.com/products/qdrone-2/

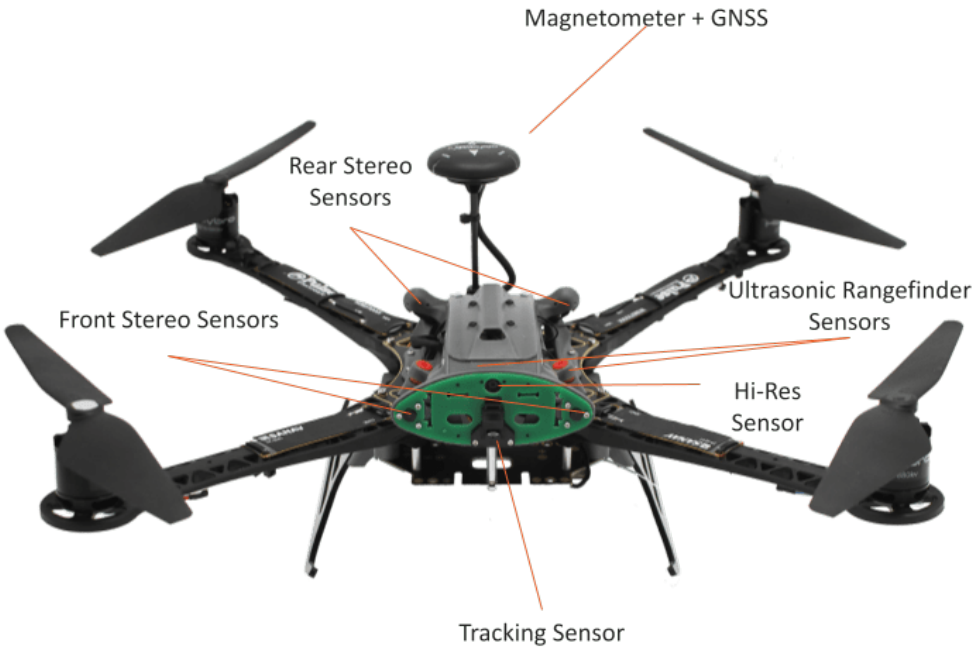

Figure 1.7: Modal AI Qualcomm Flight RB5 equipped with Stereo vision Sensors, Tracking Sensor, and Ultrasonic Rangefinder Sensors. Source: https://www.modalai.com/pages/qualcomm-flight-rb5-5g-platform

Various techniques have been developed to further enhance the accuracy of image detection and recognition for vision-based approaches. The process of image semantic segmentation allows each pixel of an image to be labeled with corresponding classifications to allow an image to be meaningful and less complex for analysis. This has drawn interest into the world of computer vision and the challenge is to segment unknown images into different parts and objects.

Guo et al [15] reviewed semantic segmentation pertaining to deep convolutional neural networks and have identified that it not only improves computational efficiency, but it can also improve accuracy by eliminating background noise. Since image segmentation is considered as mid-level representation, there are potential contributions to a wide field of visual understanding from image classification to image synthesis; from object recognition to object modeling; from high performance indexing to relevance feedback and interactive search. Garcia et al [16] surveyed on various deep learning approaches to perform semantic segmentation and had concluded that it improves computational efficiency, accuracy and can be more general than the complexities offered by conventional computer vision solutions. The key challenges however are that it requires an extensive amount of image data sets to train the algorithm and the issues of incorrect segmentation.

Semantic Segmentation is also used in autonomous driving [17] together with other sensors to achieve robust and accurate scene understanding since autonomous vehicles are fitted with multiple sensors such as cameras, LiDAR, Radars. Multiple sensing modalities can be fused to exploit their complementary properties. The accuracy of perception needs to be very accurate and deep learning with computer vision helps improve the performance of scene understanding combining multi-sensory data as shown in Figure 1.9.

SegNet [18] was developed to address the efficiency in both memory and computational time which is crucial for lightweight and dynamic micro drone applications. Quantitative assessment has proven that SegNet was able to provide better performance as compared to other widely adopted FCN and other well-known architectures. They have tested using 2D RGB images from road scenes and SunRGB-D [19] datasets and achieved good results since SegNet only stores the max-pooling indices of featured maps and uses them in the decoder network.

The concept of using vision-based navigation systems requires highly accurate and reliable real time 2D object recognition to work. Several works were accomplished by applying Convolutional Neural Network (CNN) machine learning techniques but requires complex designs which is unsuitable for adoption onto unmanned systems. Ding et al [20] designed a pipeline using both public and private image datasets to pre-trained CNN model to improve real time indoor object detection and recognition.

To further exploit the use of object detection and recognition for scene understanding, semantic segmentation can be applied to a known image for further classification at the pixel level. This will allow multi-class segmentation using pre-trained and new images for a more accurate and efficient 3D indoor scene recreation that can be applied on a small UA for path planning task. The accuracy and speed of semantic segmentation can also be greatly improved by applying CNN to enable the small UA to perform path planning without prior knowledge about new environments or positions of obstacles.

Figure1.8: Example of Semantic Segmentation of a Motorcycle Image by Guo et al [16]

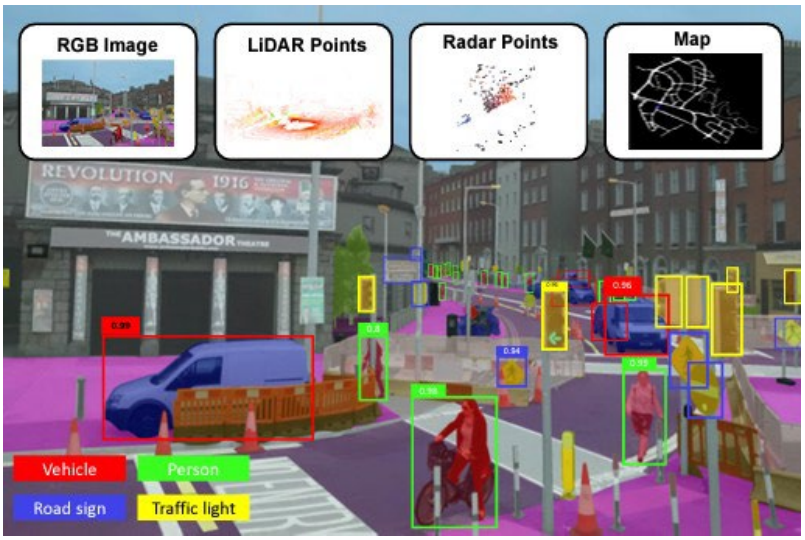

Figure 1.9: A complex urban road scene with image classification labels obtained from RGB image [16, 18]

1.2 Research Objective

As discussed earlier, it is important to reduce the weight and footprint of small UA due to its limited payload carrying ability. Such small UA is in demand by various industry sectors for indoor applications where GNSS is not readily available. Therefore, the objectives of this thesis are as follows:

1. To study how UA regulations had made an impact on the navigation constraints of small UA.
2. To evaluate current convolution neural network models for image detection and recognition, semantic segmentation, and depth inference capabilities suitable for 3D scene reconstruction.
3. Develop a multimodal framework for effective 3D scene understanding of static and dynamic obstacles by coupling image detection, semantic segmentation, and depth perception models.
4. Validate autonomous flight using SDP in GNSS denied indoor environment.

1.3 Contributions

This thesis contributes to the development of Semantic Depth Prediction method that enables efficient avoidance collision and path finding functions for small UA in GNSS denied environments. The main goal is to infer 3D scene information from 2D images using semantic depth prediction model for optimal path finding in an obstacle rich environment. Secondly, the long-term goal is to be able to apply this as an embedded vision system for small UA to achieve autonomous sense and avoid capabilities in an indoor environment. The contributions to this development are summarized below:

1. Proposing a fused multi-modal CNN method comprising of image detection and recognition; semantic segmentation and depth inference to perform indoor navigation using a single monocular camera.
2. Develop autonomous navigation method using Semantic Depth Prediction (SDP) without additional sensors apart from the integrated camera on a small UA.
3. Training of deep learning system by fusing synthetic and actual image datasets (NYU2 and SunRGB-D) for subjective identification of objects and obstacles.

1.3.1 List of Publications

The contributions of this thesis have led to the following peer-reviewed conference publications:

- Mark Tay Yong Kiat and Sutthiphong Srigrarom Laser Guided Semantic Depth Prediction System (An indoor micro-UAV navigation platform, 15th International Conference on Intelligent Unmanned Systems (ICIUS) 2019, Beijing on 28th August 2019 (http://icius2019.org/)
- Y. K. Tay and H. Hesse, "Impact of Unmanned Aircraft Regulations on Autonomous Navigation Approaches for Indoor Multi-Rotor Applications Survey," 2021 7th International Conference on Control, Automation and Robotics(ICCAR),2021,pp.201205,DOI:10.1109/ICCAR52225.2021.9463498 Awarded for “Best Oral Presentation”.

1.4 Thesis Outline

The focus of this work is to develop the use of lightweight CNN to perform computer vision-based navigation in a GNSS denied indoor environment using a small UA in a GNSS denied environment. The goal is to use the onboard monocular camera to achieve an image-based navigation using an off the shelve UA without the need for additional navigational sensors.

Chapter 2 discusses the motivation on how UA regulations had driven the way autonomous UA systems are designed. Classifications by weight with respective regulatory requirements have been put in place to ensure safe UA operations. With the increasing demand for small UA in real world applications, the challenge is to establish lightweight autonomous navigation methods with the least amount of hardware to reduce its weight with the use of computer vision and CNN satisfy both regulators and real world needs through computer vision with deep learning approaches.

Chapter 3 discusses on SDP’s approach and how various of lightweight CNN models designed for mobile applications were evaluated to determine the combination of CNN models used for object detection and recognition and joint semantic depth segmentation model that would fulfill SDP’s objective.

Chapter 4 provides experimental results of the software in the loop and hardware in the loop validation in the proposed warehouse environment and Chapter 5 concludes the findings in this thesis and future work.

Chapter 2

2 Background and Related Work

This chapter will cover several topics related to the motivation of the development of SDP for small UA. As the demand for autonomous small UA indoor applications increases, it is not possible to adopt the matured methods of navigation that are used on larger UA for outdoor applications. The development of UA regulations across the world has also played a part in determining the relative size of the UA by various weight categories that may affect the decision of navigational solutions used for the desired applications.

In addition, with technology advancements in the world of computer vision and deep learning, we will look at how deep learning methods can improve the implementation of computer vision and how it is able to minimize the need for additional navigation sensors to enable small UA with autonomous capabilities.

2.1 Impact of UA Regulations Affecting Autonomous Navigation Approaches

The demand for UA technologies in real world commercial applications [21–23] has been constantly increasing over the recent years due to the technology advancement and economic benefits. This technology has evolved from the first UA in 1783 in the form of a wind dependent air balloon to current times where palm size flying robots with smart features that can be purchased conveniently over the counter. Over the past decade, UA has been extensively used in the outdoor environment for a vast spectrum of commercial applications [22, 23] such as construction, agricultural, surveillance, entertainment, and transportation industries. One of the successes to the rapid evolution of outdoor UA application is the availability of GNSS [24] that provides the basis for autonomous navigation. GNSS technology was approved for civilian use in the 1980s and is now widely used for navigation and positioning applications. In modern day applications, UA navigation systems cannot solely rely on GNSS technology alone.

Outdoor applications in obstacle rich urban environments face intermittent GNSS outages caused by signal masking and multi path issues. GNSS is usually paired alongside inertial sensors to provide a dead reckoning system [25] where inertial sensors would provide position information in the event of momentary GNSS outage.

With the rise of indoor UA applications in the recent years., the adoption of UA technology in the supply chain sector offers competitive economic benefits for supply chain integration, shortening of cycle times to support improved customer service levels and improving supply chain responsiveness. One of the key challenges to achieve autonomous navigation capabilities in GNSS denied obstacle rich environments is the ability to perform precise and reliable pose estimation with respect to the known obstacles for avoidance collision and path planning functions. Weight and size of such air vehicles is another important consideration for safety when operating in confined spaces that are populated with high human traffic or expensive stock. Safe and practical indoor UA applications cannot be achieved without overcoming such challenges.

The increased demand for commercial UA operations has resulted in a need for national aviation authorities to maintain safety and competency standards in the interest of public safety. UA regulatory frameworks by weight classification will change the way UA are classified especially for commercial UA operations since earlier UA developers had not considered this non-existence requirement in the past as part of their design considerations.

This thesis will discuss how UA regulations have impacted the existing autonomous indoor UA navigation solutions and emerging trends for modern day UA systems considering UA regulatory requirements.

2.1.1 Common Methods of UA Classification

Classifications of UA were generally divided between military or civil applications and further broken down into the type of applications unique to specific operations. For example, within each group (military and civil), it can be further differentiated by its take-off weight; flight mechanics e.g. airplane, helicopter, multi-rotor, powered-lift; operating range and endurance; or by specific commercial applications. Due to the sharp increase in commercial UA applications, it is now important to consider how national aviation authorities across the world are classifying commercial UA.

2.1.1.1 UA Classifications by Regulations

Since UA operations involves a mixture of stakeholders that could either be aviation trained or some who are not, International Civil Aviation Organization (ICAO) had developed a set of guidance to help respective countries devise a UA regulatory framework according to their own needs without the compromising safety and economical needs. Under ICAO's definition, "*UA is defined as an aircraft intended to be flown without a pilot on board and can be remotely controlled from another place or pre-programmed to carry out a task without intervention"* [26]. However, UA regulations for commercial applications still vary across different countries depending on whether technology or safety was regarded as the higher priority. UA Regulatory framework has been constantly updated to cope with safety requirements, new commercial applications and technology advancements that are unique to the respective country's UA climate.

Most UA regulatory framework concentrates on 4 sub areas of compliance. They are opera- tor's Competency; registration of UA; type of operations and insurance. Examples of national UA framework includes Federal Aviation Authorities (FAA) in the United States implementing Part 107 Unmanned Aircraft guidelines, Civil Aviation Authorities (CAA) in the United Kingdom implementing Dronesafe initiative, European Union Aviation Safety Agency (EASA) in Europe implementing the European drone regulations and last but not least Civil Aviation Authorities of Singapore (CAAS) in Singapore implementing its Air Navigation Act 101 - Unmanned Aircraft Operations. The introduction of new regulatory requirements will eventually change the type and mass of UA systems that commercial applications will adopt due to regulatory compliance. Since the entry to market for any commercial type of UA is dependent on the authorities' regulatory approvals to operate, it is very therefore important to start bench marking against these regulations to accurately determine the potential use cases for new technological developments.

CAAS governs the use of all UA activities with Singapore's Air Navigation Order (ANO) 101 - Unmanned Aircraft Operations. UA regulations in Singapore are generally classified firstly by weight and subsequently by type of UA. UA purpose is categorized by recreational purpose, educational purpose, or non-recreational and non-educational purpose. Regardless of its purpose, it is mandatory to register any UA that has a total take off mass above 250g, regardless if it is operated within an indoor or outdoor environment. For commercial purposes, the UA operator is required to hold a valid Unmanned Aircraft Pilot License (UAPL) regardless of total take off mass.

UAPL are classified into 2 categories; Class A UAPL is required for below 25kg UA and Class B UAPL is required for above 25kg UA. Each class of UAPL is further divided into 4 UA types; Aeroplane, Airship, Rotorcraft and Powered-Lift. Figure 2.1provides an overview of the necessary CAAS regulatory requirements for the respective total UA mass.

Similarly, Federal Aviation Authorities (FAA) Part 107 requires all UA weighing between 250g to 25kg flying for work or business to be registered. From a regulatory standpoint, public safety was the key priority, and a study was conducted in 2016 by FAA Regulatory Task Force (RTF) to assess the risk levels associated with the mass-based categorization. Although it was evaluated that lightweight UA weighing less than 250g poses no lethal threat to inflict serious injuries [27], this assessment was deemed conservative due to overly simplified assumptions on impact risk evaluation. Based on accounting for the actual kinetic energy transfer of a falling UA, 250g is the conservative weight limit (i.e., safer one). The 2.2kg limit would be more realistic or representative weight limit [28]. Despite recommendations made to adjust the upper weight limit for a "low risk" UA to 2.2kg, most national aviation authorities took the conservative approach with the 250g weight threshold. It is evident from Table 2.1 that most national aviation authorities classify UA below 250g as harmless and do not impose regulatory requirements on them.

<table>
<tr><th>Purpose
UA Mass</th><th>Recreation</th><th>Education</th><th>Commercial or
(non-recreation, non-education)</th></tr>
<tr><td>UA ≤ 250g</td><td colspan="2">• Class 2 Activity Permit*</td><td>• Operator Permit
• Class 1 Activity Permit
• UA Pilot Licence</td></tr>
<tr><td>250g < UA ≤ 1.5kg</td><td colspan="2">• Class 2 Activity Permit*
• UA Registration</td><td rowspan="4"></td></tr>
<tr><td>1.5kg < UA ≤ 7kg</td><td colspan="2">• Class 2 Activity Permit*
• UA Registration
• UA Basic Training Certificate or UA Pilot Licence</td></tr>
<tr><td>7kg < UA ≤ 25kg</td><td>• Class 2 Activity Permit*
• UA Registration
• UA Pilot Licence</td><td rowspan="2">• Operator Permit
• Class 1 Activity Permit
• UA Pilot Licence
• UA Registration</td></tr>
<tr><td>UA > 25kg</td><td></td></tr>
</table>

* Only if you are operating in no-fly zones, or above 200 feet AMSL

Figure 2.1: Summary Table for CAAS UA Regulations

National aviation authorities do not specifically classify autonomous UA operations. It can be assumed that the intended use for any autonomous UA systems regardless of indoor or outdoor applications is mainly for commercial applications. As such, most commercial UA operations will require relevant permits and licenses from their aviation authorities despite operating fully autonomous system that does not require a pilot in the loop. Table 2.1 is a summary table of UA regulatory requirements by some countries.

2.1.1.2 Emerging trend in UA development below 250g

The increase in adoption of the more conservative 250g weight threshold across many national aviation authorities have started to influence UA manufacturers to review their existing product line of UA systems for the consumer market since most were designed based on applications without regulatory classifications by weight. As a leading UA manufacturer, DJI's commitment for safety led them to launch the Mavic Mini series since October 2019 that was purposefully designed with a total take-off weight of 249g to avoid the need for UA registration. Other lightweight UA systems were developed prior to the implementation of the weight dependent regulations such as the Ryze Tello that weighs approximately 80g or Parrot Mambo that weighs approximately 73g, such UA are considered basic toy UA with minimal advance features and low-resolution cameras.

UA manufacturers typically drive new technological adoption and bring forth new technology innovations into the commercial UA market. This would therefore influence researchers in the areas of UA technologies to consider regulatory weight classifications during the development of UA related technologies.

Table 2.1: UA Regulatory Requirements by Countries

Countries	UA Registration	Requirements
Australia	All weights	- To operate UA < 2kg for commercial reasons, CASA must be notified.
Canada	For UA between 250g up to 25kg	- UA pilot licence to fly UA that weigh 250 grams (g) up to and including 25 kilograms (kg)
China	For UA >250g	- All drones flown for commercial use requires a commercial UA license.
France	For UA ≥800g	- Commercial UA operators must pass a theoretical exam and undergo practical training/assessment.
Germany	For UA ≥250g	- UA > 5 kg must obtain permit to fly at night. - License required for UA > 2kg
Japan	Not required	- UA weighing 200g or more must seek permission to operate.
Singapore	For UA ≥250g	- Permits required for commercial UA operations.
South Korea	All weights	- License required for all commercial operations with ≥ UA 12kg.
United Kingdom	For UA ≥250g	- Commercial UA operations Operator ID[1] and/or Flyer ID[2] required for UA > 250g and to obtain Permission to Fly Commercially (PFCO) - Insurance is required for all commercial UA operations
United Arab Emirates	All weights	- Permits required for commercial UA operations.
United States	For UA between 250g to 25kg with exceptions for recreational flyers. N paper registration for 25kg above	- License required for all commercial operations. - Airspace authorization for UA operations outside of class G airspace

[1]Operator ID - Must be labeled on your drone or model aircraft.
[2]Flyer ID - Shows operator has passed the basic flying test.
*Accurate at the time of publishing

2.1.2 Autonomous Navigation for Multi-rotor Systems

UA navigation is the process where the system determines its position based on a reference and plans an optimal path to navigate to its desired location. Autonomy of the navigation is aided with sensors providing relevant sensor data for localization reference. A basic Multi-Rotor (MR) system architecture is shown in Figure 2.2 where navigational algorithms in the guidance, navigation and control module determine its state, position estimates and its optimal flight path with respect to its operating environment. The algorithm output commands are subsequently fed to the propulsion system to perform to execute the desired maneuvers.

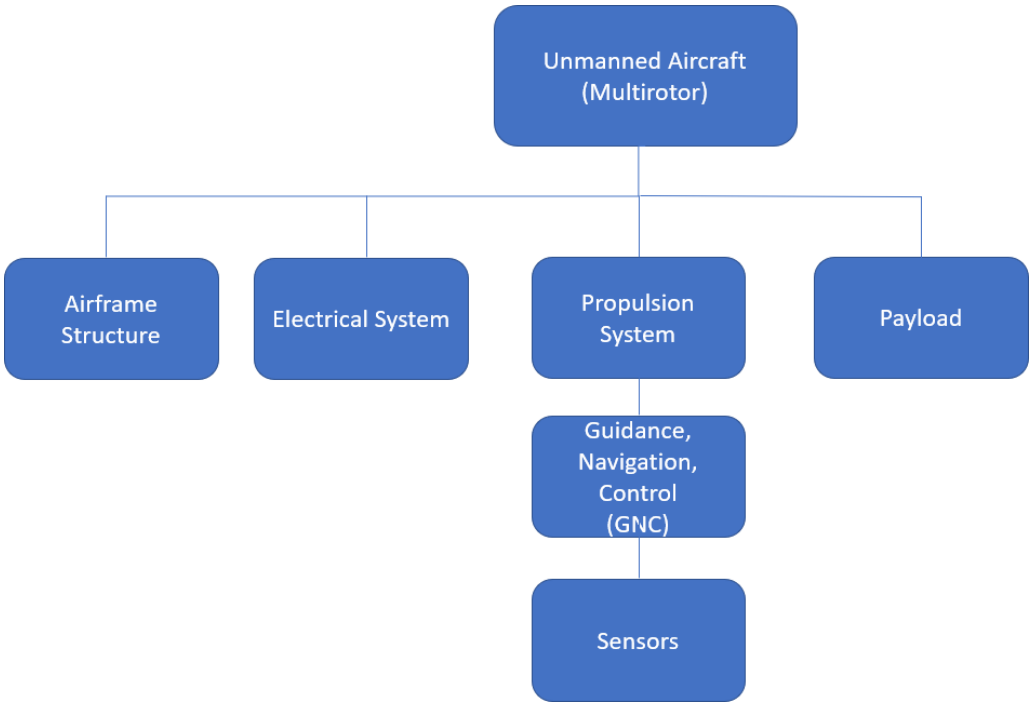

Figure 2.2: Basic Multirotor System Architecture

Autonomous navigation can firstly be classified by outdoor or indoor applications and subsequently by global or localized navigational by sensor types as shown in Figure 2.3. There are also various Indoor localization methods that can be further differentiated between off-board and on-board methods.

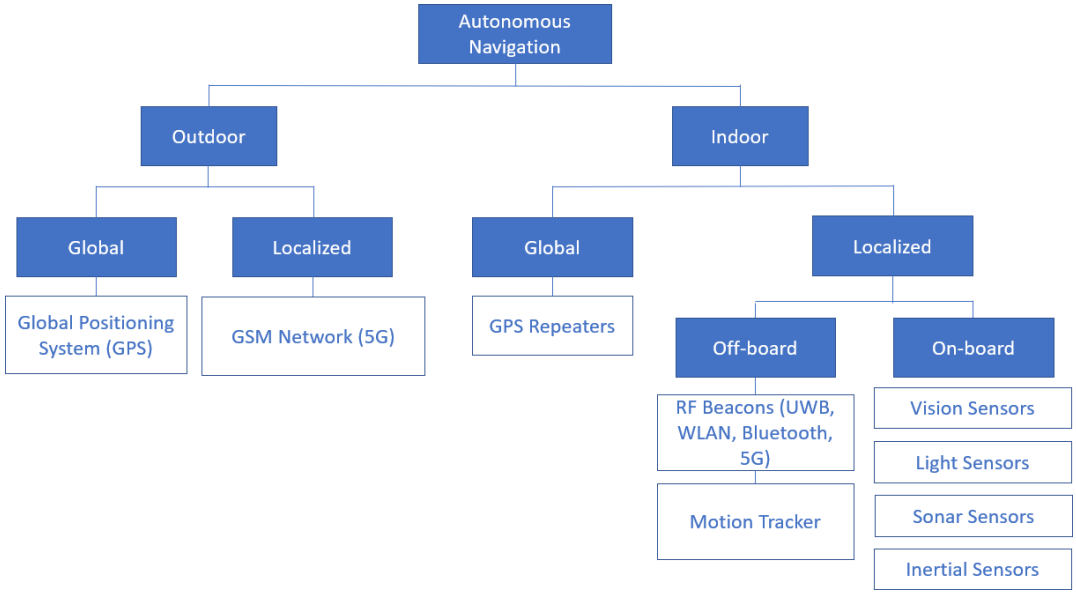

Figure 2.3: Classification of Autonomous Navigation by Sensor Type

2.1.2.1 Outdoor Navigation Methods

Most outdoor applications can be assumed to be flying over remote or rural areas where the probability of UA striking a person is not more than 0.01% [29] since human population is less dense and the risk of UA falling and causing harm to people below. In addition, the nature of outdoor applications requires a larger UA to have the capacity to carry heavy payloads such as pesticides, parcels, or even commercial grade cameras to perform its commercial task. Other considerations such as weather, endurance and range may not be an incentive to operate small UA. UA designed for outdoor autonomous flights relies on the matured GNSS method for global navigation [30] and some UA systems are also fitted with other sensors for avoidance collision capabilities [31]. GNSS has been around for decades and is a popular navigational system used in manned aviation. The increase in demand for commercial applications [32]in the areas of search and rescue, remote sensing, civil infrastructure, agriculture, supply chain and even drone taxi is pushing UA industry into a new era. Most of these UA applications perform autonomous flights using GNSS guided waypoints as the point of navigation like manned aviation. Advantages of GNSS include 24/7 availability, good location accuracy worldwide and uses standard latitude/longitude reference. The disadvantage is that GNSS signals will be attenuated by roofs and walls therefore is not suitable for indoor navigation applications without the use of GNSS repeaters.

2.1.2.2 Indoor Navigation Methods

Indoor applications are however more delicate due to confined spaces and obstacles. The potential demand for indoor applications from the supply chain industry's perspective mainly evolves around inventory management inside warehouses where UA can be used to perform stock taking and other associated processes. Although GNSS reception is poor or non-existent for indoor environments, it is possible to generate GNSS signals using Pseudolites (Pseudo-Satellites) [33] installed at corners of room to create a pseudo satellite constellation. This allows GNSS signals from each satellite to be received and subsequently relayed through indoor transmitters. No modifications were required on the GNSS receiver end and horizontal position accuracy proved to be as accurate. This solution cannot detect obstacles and other infrastructure thus would require additional sensors for collision avoidance.

Other indoor localized navigation methods can be achieved with off-board techniques such as RF beacon [34] or motion trackers [35] to track the position of the UA, this method is unable to perform obstacle avoidance on its own as well. Another disadvantage for this method is that it requires RF receivers or visual markers to be installed on the UA and that it must operate within line of sight of its transmitters or trackers. Such technique is also limited to the local area network of the installed RF transmitters and trackers thus can be costly solution if the area of operation is extensive.

Figure 2.4 summarizes some of the advantages and disadvantages across the various navigational methods. Advantages of vision sensors outweigh the other methods of indoor localized methods. There is no impact to weight since most UA are already equipped with onboard cameras therefore allowing the possibility to use the video feed for vision-based navigation tasks.

2.1.2.2.1 Computer Vision Methods for Indoor Navigation

With the advancement in camera and graphics processing units (GPU) technology, computer vision approach has been a popular alternative for mobile robotics and even autonomous vehicles to achieve precise localization and pose estimation by detecting objects or obstacles through feature extraction and background noise omission. Recent survey [11,12,13,14] indicated a growing popularity with such approach which is also known as Visual Simultaneous Localization and Mapping (VSLAM) for autonomous navigation of indoor drones in the absence of GNSS. Advancement in computer vision technologies provided several advantages and benefits leading to low cost and lightweight navigation system.

Other benefits for vision-based approach are the ability to capture rich details of an environment with image data that is not only useful for navigational purposes, but the same image data can also be used in parallel for non-navigation applications such as surveillance, architectural, photogrammetry or infrastructure inspection purposes.

Environment	Coverage	System	Advantages	Disadvantages	Impact to weight
Outdoor	Global	Global Positioning System (GPS)	- Readily available globally - Relatively low cost - Works in all weather	- Requires Line of Sight - Multipath issues in urban environment - Not able to detect obstacles	Minimal since most UA are euipped with GPS
	Localized	GSM Network (5G)	- High speed and capacity - Low latency and power consumption	- High cost to operate (Still in development) - Coverage is dependent on cell tower locations - High power consumption - Not able to detect obstacles	Minimal as 5G chipsets are integrated with advance companion boards
Indoor	Global	GPS Repeaters	- Provides real GPS signals to indoor environment	- Requires Line of Sight - Multipath issues in enclosed environment - Require to set up pseudolites or GPS repeaters - Limited to operate within pseudolite range - Not able to detect obstacles	Minimal since most UA are equipped with GPS and pseudolites are not mounted onto UA
	Localized (Off-board)	RF Beacons	- Good accuracy	- Requires Line of Sight - Accuracy is dependent of RF access points - Requires RF transcievers on UA - Multipath issues in enclosed environment - Not able to detect obstacles - Limited to operate within RF transmitter range	Minimal. Only RF reciever required on UA
		Motion Trackers	- Good accuracy - No multipath issues compared to RF beacons	- Requires Line of Sight - Not able to detect obstacles - Limited to operate within RF transmitter range	Minimal. Only visual markers required on UA
	Localized (On-board)	Vision Sensors	- No external infrastructure required - Most UA are equipped with onboard cameras - Image data provides rich information about its environment and can be useful beyond its purpose of navigation - Versatile by applying various image processing methods to localization and mapping techniques - Possible to adopt Machine Learning to improve navigation - Able to provide depth information through image processing	- Can be affected by poor lighting conditions - May require higher processing capabilities depending on image size	Nil since most UA are equipped with onboard cameras
		LIDAR Sensors	- No external infrastucture required - High accuracy - No affected by lighting conditions - Able to provide depth information	- Requires LIDAR sensor to be integrated on UA - LIDAR sensors are costly - Weight and size is not suitable for small UA - 3D point cloud does not provide rich information as compared to vision sensors - 3D point cloud requires extensive amount of computing resources - High power consumption	Yes (E.g RPLIDAR weighs 200g, Velodyne Puck Lite weighs 590g)
		Sonar Sensors	- No external infrastucture required - Low cost - Not affected by lighting conditions	- Not able to produce map of surrounding - Slower sensing rate since its base on speed of sound - low accuracy	Minimal as a typical sonar weights only 8.5g
		Inertial Sensors	- Low cost - Low power consumption	- Errors accumulates over time - Requires other sensors to improve accuracy - Subjected to magnetic disturbance	Minimal since most UA are euipped with GPS

Figure 2.4: Impact of Autonomous Navigation Methods in Different Environments

2.1.3 Conclusion

The adoption of weight dependent regulations across national aviation authorities has influenced leading UA manufacturers to consider weight requirement during the development of UA related technologies. A comparison across various navigation sensors suggested that vision sensors have several advantages over other navigational sensors without compromising significantly on the UA weight; especially for indoor applications since most off-the-shelf UA are equipped with onboard cameras. Vision sensor data is also useful in two folds; (1) to perform onboard localized navigation that is crucial for autonomous navigation in obstacle rich indoor environments and (2) the same vision data can be used for other non-navigational tasks that is equally important in commercial real-world applications. These requirements should be considered when developing a safe and lightweight indoor autonomous UA that can also allow commercial UA operators to avoid weight dependent UA regulations if necessary.

2.2 Computer Vision Methods for Autonomous Navigation

In the previous section, we learned that the key challenge to equip small UA with indoor autonomous navigation capabilities is limited by the payload carrying capabilities. For any UA to autonomously navigate in a GNSS denied indoor environment, it must understand its environment before it is able to determine the best obstacle free path to follow. Since our argument is that most UA are fitted with cameras, implementing visual based autonomous navigation approaches for small UA will help to address this issue.

Computer vision (CV) generally makes use of image capturing devices to visually attain a high-level understanding of the real world. It replicates our human vision system by acquiring information through digital images, analyzing, and abstracting relevant information for decision-making tasks. Traditional CV does not contain predictive elements thus does not have the ability to make decisions and its accuracy is dependent on the programming of the models by CV engineers. The challenge with traditional CV escalates when the complexity of image analysis increases. Fine tuning of more complex CV applications will have to be programmed manually with some compromises to achieve the most desirable outcome thus making traditional CV programming dependent.

Deep Learning (DL) helps to address traditional CV issues. DL is a branch of Machine Learning (ML) which is a subset of Artificial Intelligence (AI), and it mimics human behavior to analyze and perform a given task. The key difference between DL and ML is that the former relies on artificial neural network which mimics the human brain, therefore allowing the ability to work on larger data sets with more complex correlations as compared to ML. Although CV methods have been optimized for performance and efficiency, DL methods can improve accuracy and versatility.

2.2.1 Traditional Computer Vision Methods

Traditional CV relies on engineered algorithms to extract features such as colors, edges or objects recovered from an input image to produce a desired output. Structure from motion (SFM) technique [36, 37] has been used to perform 3D scene reconstruction in real time localization system using 2D images from monocular camera setups to address localization issues in feature rich indoor environments or areas where GNSS signals are weak or non-existent.

The implementation of SFM is typically based on a 3-step approach where the initial step is to generate video recordings of its selected trajectory path using an on-board monocular camera either manually or during a real time mission. SFM algorithm is then used to generate a 3D map of the known environment that will be used to compute its position in real time with respect to the trajectory path. Unsupervised object detection and classification method is subsequently used to perform scene interpretation from the 3D map. This method proves to be accurate only with minor changes along its planned trajectory but is not adaptable to the ever-changing environment as it is limited to the initial 3D mapping information and does not have the ability to update its initial 3D map with new environment data.

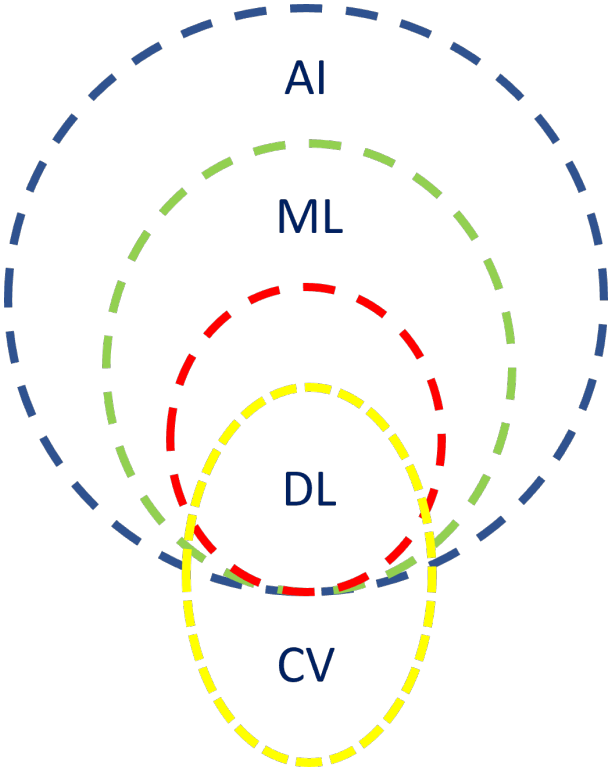

Figure 2.5: Relationship between AI, ML, DL and CV.

It is desirable to enhance visual SLAM supported methods with object detection and recognition capabilities to provide a more robust and scalable scene reconstruction. SLAM-aware object recognition system [38] incorporates multi-view object proposals and efficient feature encoding techniques onto semi-dense monocular SLAM to strengthen its object recognition capabilities. This method demonstrated that it was possible to detect and recognize objects in an unknown environment on a frame-by-frame basis across various viewpoints from a single monocular camera. Such a method was also scalable to a larger number of object categories depending on image datasets used to generate the training model. This experiment however focuses largely on a passive 2D object detection and recognition objective without depth perception of the objects that is crucial for navigation purpose.

Metric scale estimator was introduced to a monocular SLAM system [39] to establish a drift free altitude measurement with a scale estimator that transfers up to scaled positions from Parallel Tracking and Mapping (PTAM) [40] pose estimators to metric form using a single monocular built in camera on a commercially off-the-shelve AR Drone platform. Although this low-cost method proves that it is possible to rely on integrated monocular cameras for vision data to generate 3D scene representation through a PTAM system, insufficient translation caused by rapid yaw rates resulted in inaccurate estimates of the camera position and 3D map estimation and can be improved with combining IMU system with the visual data [41] to provide a more accurate pose estimation of the drone.

Stereoscopic vision [42] is a method of estimating the object's location including the third dimensional aspect using two or more images simultaneously taken of object or surroundings. This method has been gaining popularity in recent years with the advancements made with respect to the algorithms allowing accomplishments such as the "live" implementation of stereo vision in computer vision. Advantages that come with the application of stereo vision systems include:

1. It is a cheap method for the reconstruction of surroundings in 3D.
2. As it is a passive sensor, it will not be prone to interfering with other sensor devices present.
3. It can be easily incorporated with other vision objectives such as feature tracking and recognition.

3D reconstructions can be simply achieved using stereo vision cameras in two steps:

1. Extract features from both images, match the correspondent features between the set of images and generate the disparity map using these points.
2. After which, the focal length of both cameras along with the camera's geometry such as positioning and orientation, the 3D coordinates of matched features in the image pair can be calculated.

To accomplish the task stated above, the stereo cameras should be in a standard position where the optical axis is parallel and image planes being coplanar. When the cameras are positioned in this fashion, the rows in the image's frame buffer will be in line with the epi-polar lines hence matching points between the pair of images can be detected over these rows. With the physical positioning of the cameras established, the virtual parameters of the cameras must also be made known, and this is acquired through the calibration of the cameras, thus giving the parameters relating the model of the cameras within the program to the physical devices. This is essential as the coordinates and positions of the cameras must be identified to compute depth and distance information. Once the camera's parameters have been acknowledged, it is now able to match corresponding features of the two images taken by the stereo cameras and more importantly, generate the disparity between the said features. The program can be based on several stereo algorithms with different approaches, some regarding the area correlation, others basing it on feature and object detection. For the application of 3D reconstruction, implementing an area correlation algorithm with stereo cameras would be the more adequate system allowing for denser disparity maps. This is complemented by a more accurate matching of the features in the right frame for every corresponding feature in the left frame, hence creating more precise 3D reconstruction.

2.2.2 Deep Learning (DL) Computer Vision Methods- Convolutional Neural Networks (CNN)

Incorporating artificial intelligence with traditional computer vision reduces the need for human intervention, allowing an independent means of analyzing input data. Amongst the various DL methods, CNN is preferred for computer vision as it requires less pre-processing and is effectively more efficient when it comes to image processing. CNN plays a revolutionary role in applying artificial intelligence with computer vision as it is generally more accurate compared with other neural networks such as Artificial Neural Network (ANN) or Recurrent Neural Network (RNN). The introduction of CNN for image processing has improved performance and accelerated the adoption of DL in vision base approaches. Many CNN approaches have been proposed with the intention to improve detection accuracy and performance for real time applications in mobile or embedded systems. CNN extracts and analyzes data from an input image at pixel level and computes an output base on learnable weights and biases. The use of relevant image data sets for a specific task improves accuracy for image classification, object detection and semantic segmentation tasks as well. A typical CNN architecture consists of 3 layers termed as convolutional layers, pooling layers, and fully connected layers.

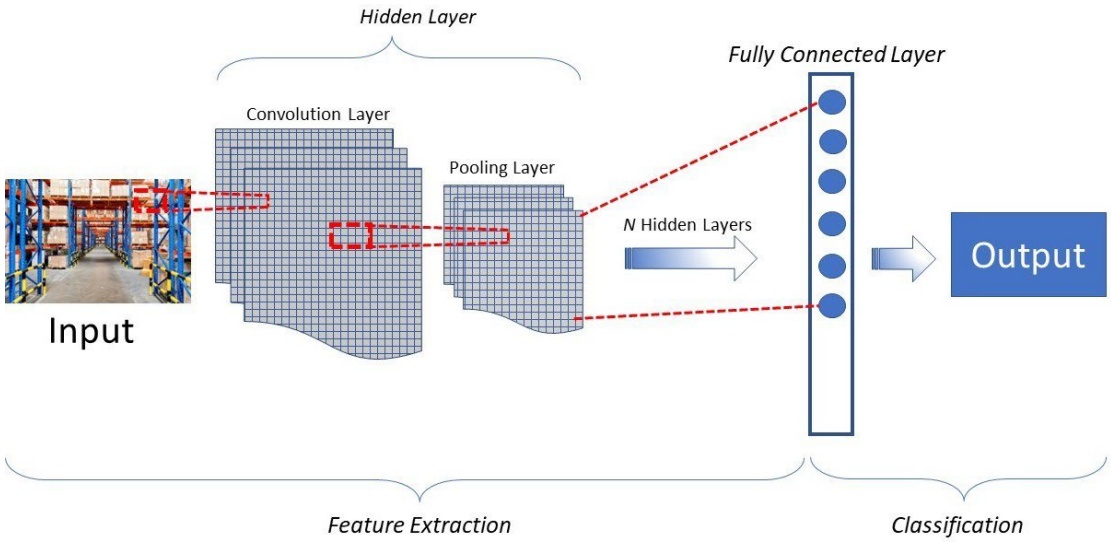

Figure 2.6: A Typical Convolutional Neural Network Architecture

2.2.2.1 Convolutional Layer

This is the key layer in a CNN where filters or kernels are applied to extract features from an input image. Each kernel of a defined N x N size moves across the input image of M x M size along its height and width to derive a dot product that associates the kernel with the input image to generate a feature map. The amount of movement each kernel moves across the input image is termed as stride and this will also determine the size of the convolutional layer output. Padding is usually added to the outer border of the input image to create more space for the kernel to cover for a more accurate analysis. The convolutional layer output is then passed on to the next layer for down sampling.

An example of the convolutional layer is shown in Figure 2.7 where a 2 x 2 kernel, stride = 2, padding = 1 will generate a 3 x 3 output.

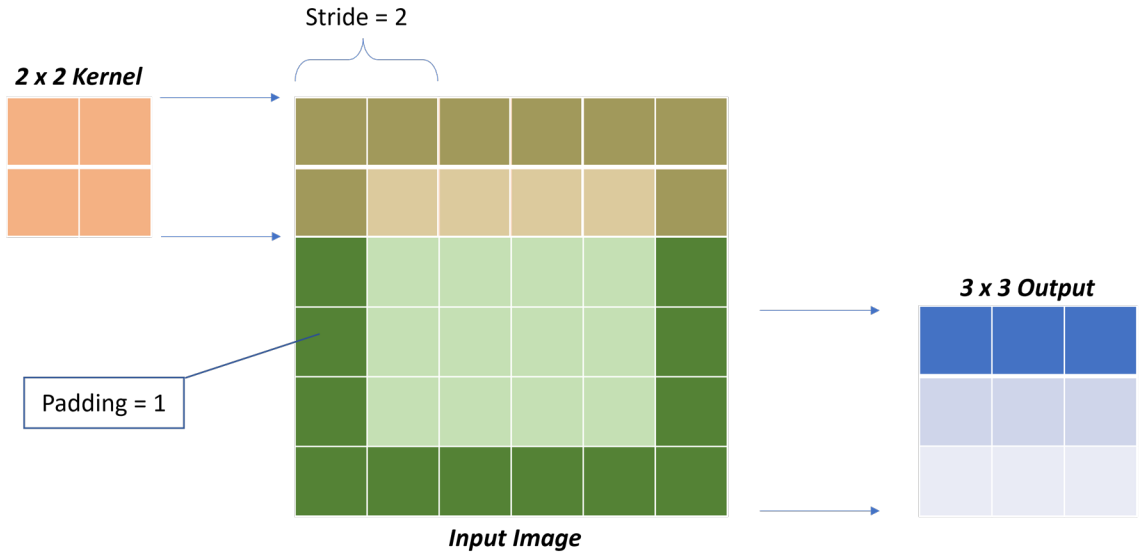

Figure 2.7: Example of how a kernel is applied to an input image.

The output size of the convolutional layer can be calculated using the following mathematical formula:

$$(((W - K + 2P)/S) + 1)$$

Where: W = Input size, K = Filter size, S = Stride, P = Padding

2.2.2.2 Activation Function

The activation function in Figure 2.8, also known as transfer function, helps to decide if the output of each convolutional layer is to be activated as an input to the next hidden layer and is typically positioned at the end of each convolutional layer. It determines the firing of a neuron base on a specific input by generating a corresponding output.

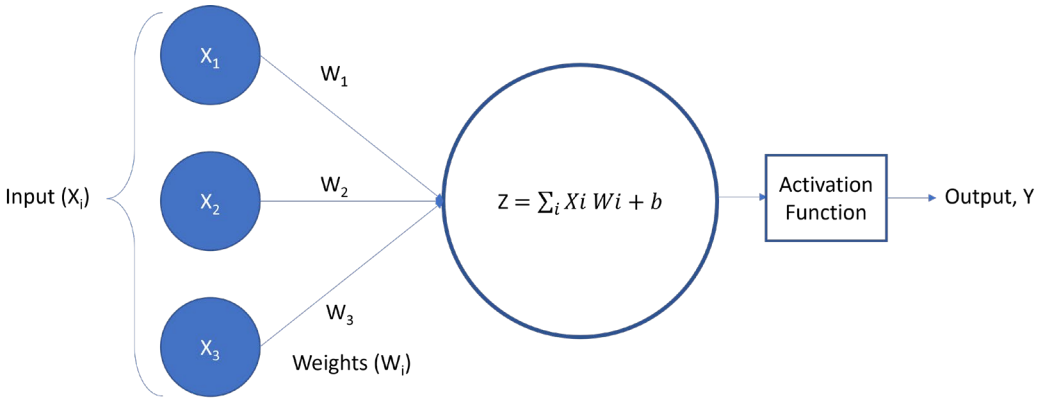

Figure 2.8: Activation Function for Neural Network

Although there are both linear and non-linear activation functions, the latter is preferred to allow the neural network to handle more complex patterns or relationships for CNN models. Some of the common non-linear activation functions used in CNN are as shown in Figure 2.9.

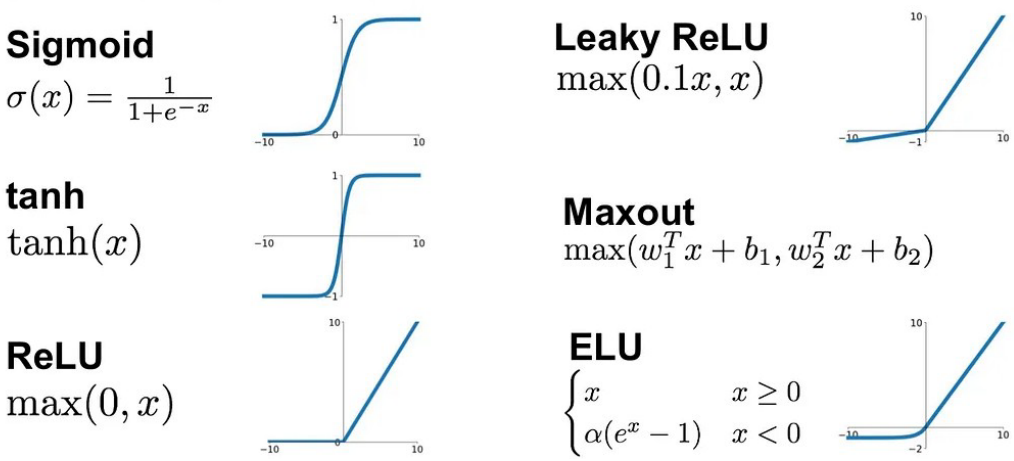

Figure 2.9: Non-Linear Activation Functions Source: https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092

2.2.2.3 Pooling Layer

The convolutional layer output passes through the pooling layer for the purpose of down sampling to reduce the memory and computing requirements without losing critical information. This is achieved by summarizing and combining the features from the convolutional layer output into a single neuron and subsequently passing it down to the next hidden layer. Pooling as shown in Figure 2.10 can be achieved either by taking an averaging or maximum pixel value covered by the kernel.

Max pooling basically takes the maximum pixel value obtained from each region of the input image that is covered by a kernel. Average pooling takes the average pixel value and min pooling takes the minimum value of the same region. The most used pooling methods are average and maximum pooling. There is no preferred pooling method as each method offers different results depending on the expected results of a given input image. For example, max pooling will determine the brightest pixel and is useful in high contrast applications. Average pooling on the other hand is used to smoothen out the images rather than determining distinct features.

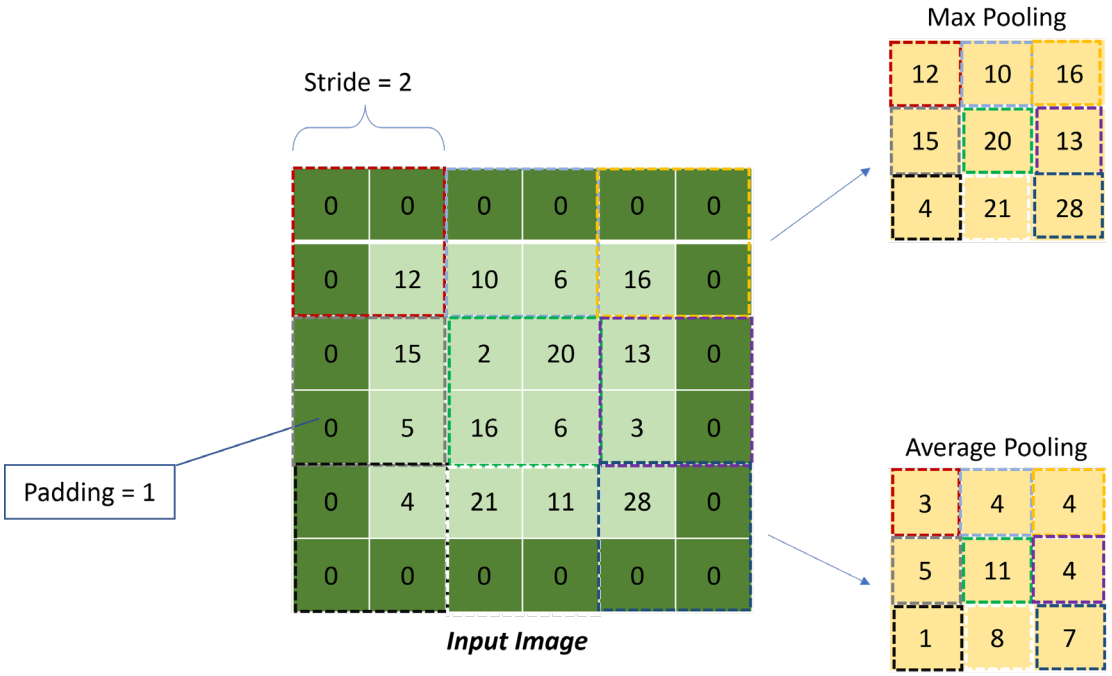

Figure 2.10: Max Pooling vs Average Pooling

2.2.2.4 Fully Connected Layer

The Fully Connected (FC) layer is also known as the classification layer and is located at the end of the CNN architecture. Each input to the FC layer is connected to the output from the last hidden layer and derives the final probabilities for each classification label. Using the example in Figure 2.11, the hidden layers will breakdown and run the kernels through the input image to analyze and filter the features based on its weights and biases in its activation function.

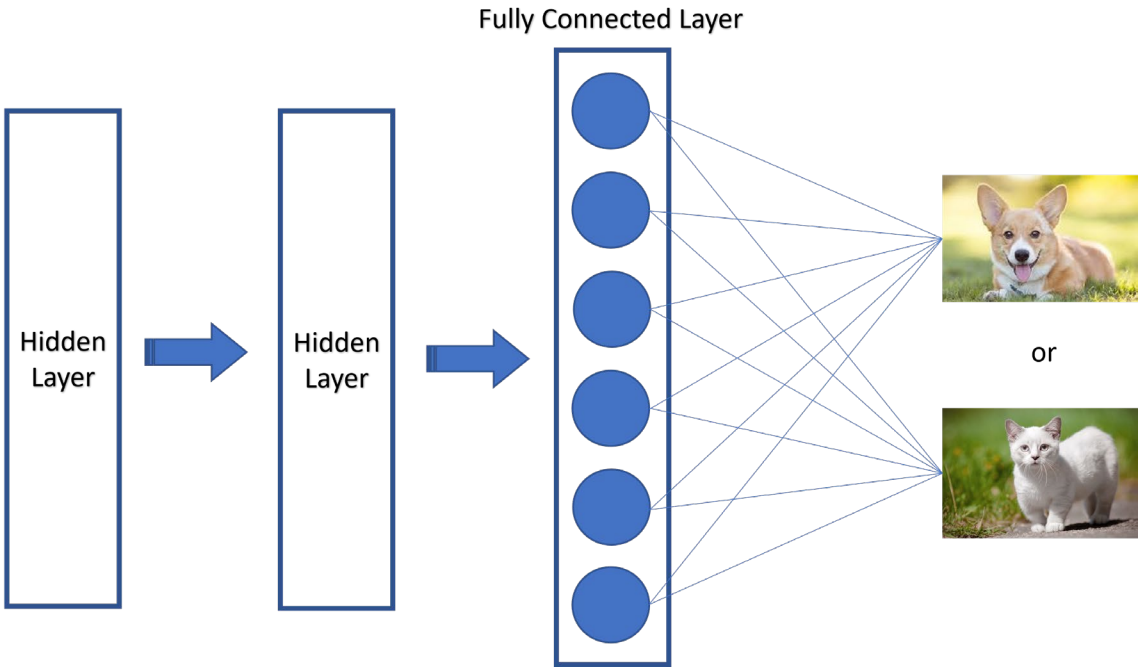

Figure 2.11: Fully Connected Layer in a CNN Architecture

2.2.3 Machine Learning Data Sets with RGB and Depth Information

In machine learning (ML), a good data set will determine the accuracy and reliability of the trained model. Although larger data sets are beneficial as it can help to improve the accuracy but may take a longer time to train due to the large amount of data, thus using existing labeled data can reduce time required to train the model. Resolution of images also plays a key role in the ML model. Ideally the higher the resolution, the better the performance. However, in some situations, high resolution images may be intentionally reduced due to processing limitations or when there is a need to normalize the resolution for a group of high- and low-resolution images.

Since the performance of any CNN model depends on good visual information and resolution for training, any degradation of image quality would have a detrimental effect [43]. Since the training process is to teach it to analyze a predetermined set of image data, training data sets will also need to be relevant to the application and must be adequately labeled for the relevant features. NYU2 [44] was developed to interpret objects, major surfaces and support relations for indoor scenes using RGBD images. The objective was to derive a 3D reconstruction through the understanding of 3D cues by recovering support relationships from the RGBD image features. NYU2 data set is made up of 1449 RGBD images obtained from Kinect sensor for a wide range of commercial and residential buildings in three different US cities, comprising 464 different indoor scenes across 26 scene classes. It consists of 35,064 distinct objects, covering 894 different classes. The labeled data set consists of pairs of RGB and depth frames that have been synchronized and annotated with dense labels for every image.

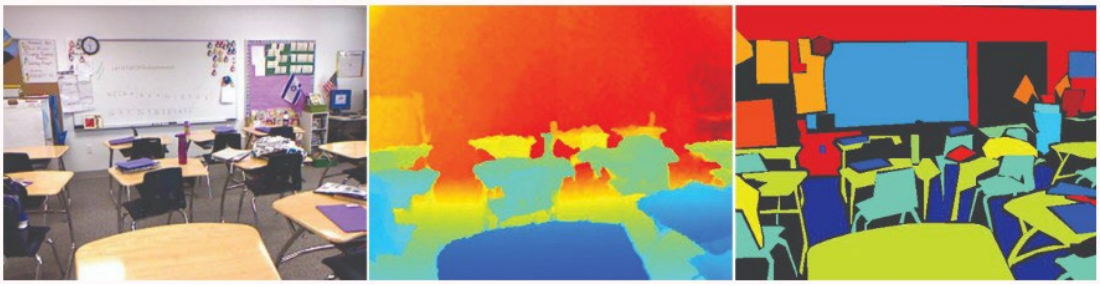

Figure 2.12: Example of NYU2 Labelled Data set - Input (Left), Depth (Centre) and Class labels (Right)

SunRGB-D [45] is another RGB-D data set developed for the purpose of major scene understanding tasks. It contains 10,000 RGBD images using 4 different RGB-D sensors; Intel RealSense, Asus Xtion, Kinect V1 and V2 and annotated with 146,617 2D polygons and 64,595 3D bounding boxes. This data set was densely annotated with accurate object orientation, room layout and scene category for each image, allowing more accurate 3D recreation of indoor scenes for data hungry algorithms for scene recognition tasks. Several combinations of six tasks; scene categorizing, semantic segmentation, object detection, object orientation, room layout estimation was used to estimate a final 3D scene which includes objects and room layout.

2.3 Conclusion

Based on the previous section, global UA regulations are driving UA form factors to a smaller footprint. With weight and space limitations, UA designed for GNSS denied indoor environment should be small and lightweight therefore the need to rely on vision-based approaches. This allows the use of simple sensors such as monocular cameras which are already an integral part of most small UA. Image data from such cameras can be used to visualize the localized environment. The use of CNN can further enhance its scene understanding ability without human supervision, to make decisions based on prior training.

Chapter 3

3 Semantic Depth Prediction (SDP) Approach

3.1 Introduction

This section introduces the concept and underlying architecture Semantic Depth Prediction (SDP) method. Semantic Depth Prediction (SDP) is a holistic approach that uses deep learning computer vision methods for scene understanding and depth inference using 2D images captured from a monocular camera. The motivation is to enable a small UA to perform pose estimation and path planning by understanding an unknown scene using pre-trained model without additional sensors to lighten its weight. There are several contributing factors that can affect its efficiency and accuracy of object detection and depth prediction that will subsequently affect the autonomous navigation performance. To achieve this, we are proposing a real time 3D indoor scene recreation by fusing 2 lightweight CNN layers to achieve semantic depth prediction data for optimal path finding in an obstacle rich environment. This is then applied onto the small UA to achieve autonomous sense and avoid capabilities within an indoor environment.

3.2 Semantic Depth Prediction Model

A lightweight semantic depth model was developed by using a series of lightweight CNN architectures so that it can be deployed on small UA intended for indoor applications. Each image from a monocular camera is processed through SDP to determine semantic segmentation labels and depth inference. Since various metrics of each layer determine the relative performance of CNN models used for SDP, it is important to select a lightweight and efficient CNN model for each layer as using models that are highly accurate may compromise on the performance metrics and using the faster networks may compromise on the accuracy metrics.

Figure 3.1 presents the end-to-end approach for SDP, which is a combination of efficient deep convolution neural network models unified through 2 computer vision tasks namely 1) Object detection and recognition and 2) Joint Semantic Depth to produce a joint semantic depth inference for an input image. The possibility of sensor fusion based on attitude control and visual sensing to corroborate with deep neural networks was explored. SDP was modeled based on open-source data sets (like NYU2 and SegNet) and data sets taken from simulation and manual flights. The inferred SDP output for each input RGB image was subsequently translated into safe probability coefficients represented in a matrix to estimate the depth and position of obstacles.

The key success to this approach depends on the magnitude and quality of image data sets to optimize the speed and accuracy of object classification and depth perception. This can be achieved by training a deep learning model using relevant images to develop generic pre-trained models for indoor scenes or unique pre-trained models for specific environments such as a typical indoor logistic warehouse. Although processing can be performed onboard the UA, this would typically require onboard GPUs for real time processing. To keep the UA lightweight, we propose to process the images off-board via a ground terminal with subsequent control commands Pitch θ, Roll φ, Yaw ψ and altitude Z commands computed and sent back to the UA.

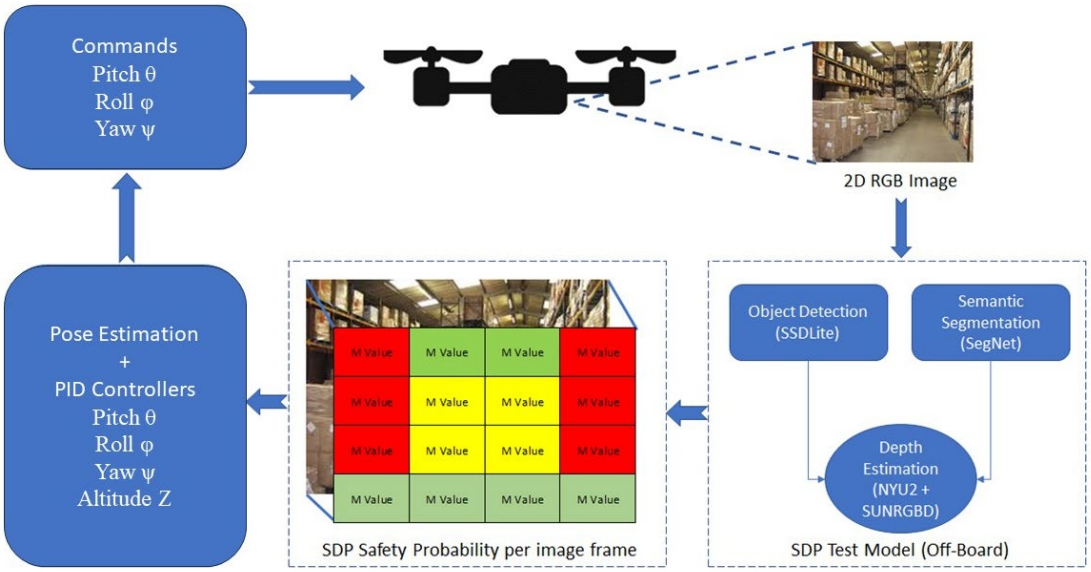

Figure 3.1: Framework overview of end-to-end Semantic Depth Prediction approach to infer a semantic depth map from a 2-dimensional input image and subsequently derive safety probability coefficients for UA control inputs.

Figure 3.2 shows how an input image is processed by SDP to generate 2 layers, 1) Object detection and Recognition Layer and 2) Combined Semantic Segmentation and Semantic depth Layer. The initial high level object detection and recognition task serves two purposes, firstly to classify and localize potential objects in each input image, for example finding bar codes to detect stocks in the warehouse as a non-navigation task. The second purpose is to detect existing features in the operational environment to initiate autonomous flight command cues. Each object of interest detected at image level detection task is subsequently processed using semantic segmentation for pixel level classification to differentiate objects that are in the foreground or background. NYU2 and SunRGB-D data sets containing RGB, and depth information were used for depth inference where depth information is used to generate the final 3D scene reconstruction. Supervised learning is applied using open-source image data sets from ImageNet, NYU2, SunRGB-D for baseline training and real image data relevant for specific operational environment is used to supplement the baseline pre-trained model to improve the accuracy of each computer vision model.

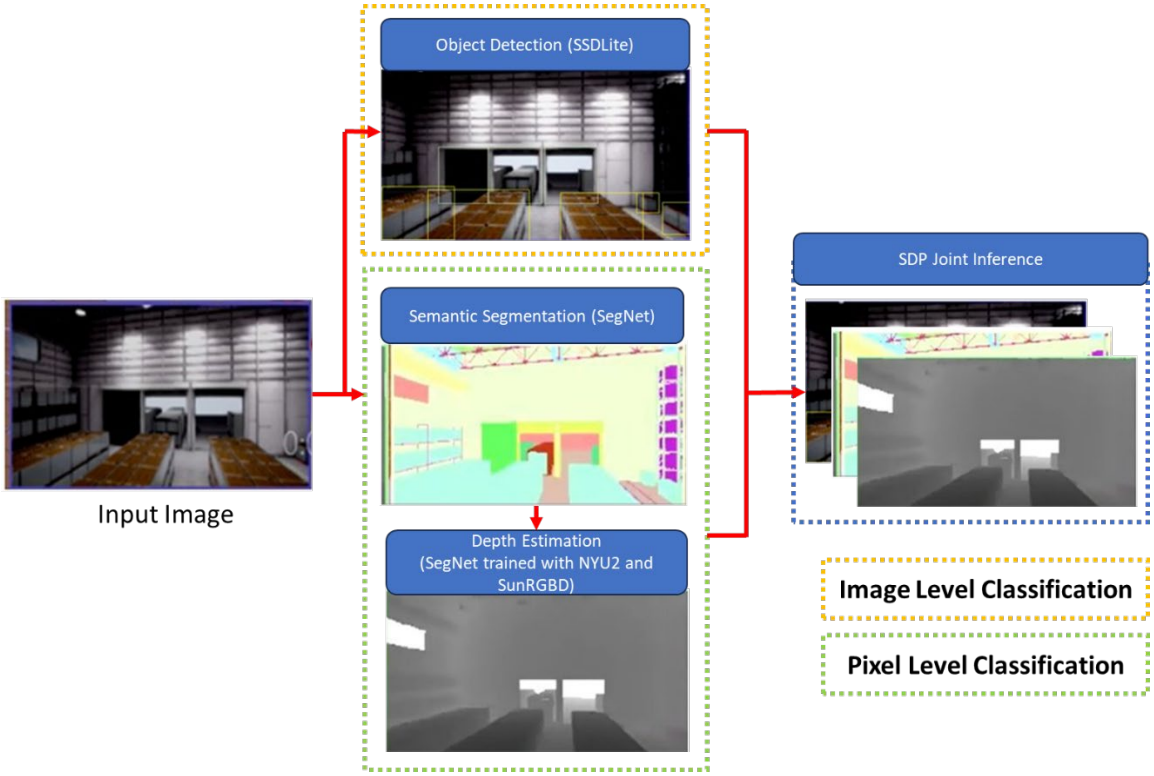

Figure 3.2: SDP joint inference using SSDLite and SegNet.

3.3 Evaluation of CNN Models for Semantic Depth Prediction Model

Earlier discussions indicated that CNN is the preferred DL architecture for image classification tasks since it is a feed forward neural network that can manage large extent of parameters found in images without compromising on its performance and efficiency. In this section, we will discuss the efficient CNN models that were selected for SDP.

SDP makes use of object detection and recognition layer, semantic segmentation, and semantic depth layer. Like any other deep learning system, training is important to derive a Pre-Trained Model (PTM) that SDP can use for the purpose of image and pixel classification. For this project, datasets from NYU2 [44], SegNet [18] and SunRGB-D [45] were used as the basis for SDP baseline training data.

3.3.1 Object detection and Recognition Layer

Object detection and recognition identifies and locates objects of interest within an image. Examples of object detection and recognition applications include counting of people in Figure 3.3 or identification of bar codes labels in Figure 3.4. The purpose of object detection and recognition is to locate and classify any existing object within an image, by labeling them through a bounding box and identifying the confidence level of the object. Deciding factors used to determine the appropriate models for this layer is based on the accuracy as well as performance of each model for this lightweight approach.

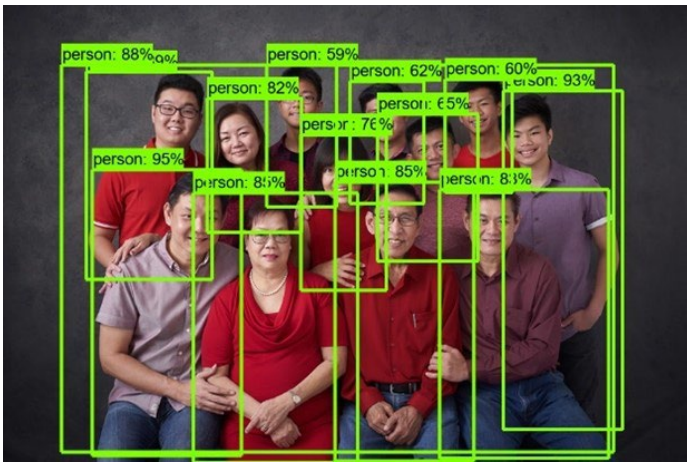

Figure 3.3: Object Detection and Recognition for people counting.

Figure 3.4: Object Detection and Recognition for Barcode identification.

The initial high-level task of real time image classification and localization for the objects of interest from each input image is performed using Single Shot Multibox Detector (SSD) as it is one of the best object detectors available, both in terms of accuracy and speed [46]. This is due to its architecture on a single-stage approach in convolution to detect multiple objects. This algorithm breaks down the image into a series of bounding boxes on each layer of the feature map and predicts the presence of an object in each bounding box. As shown in figure 3.5, each layer within the feature pyramid has a prediction parameter, and they can detect an object independently.

A typical SSD architecture makes use of the Visual Geometry Group of 16 layers (VGG-16) pre-trained with ImageNet [47] data set comprising 15 million labeled images across 22,000 categories. VGG-16 extracts low-level features from the feature map to boost the performance of the SSD capability in high quality image classification. In SSDLite, MobileNetV2 replaces VGG-16 as the base network for feature extraction using depth-wise convolutions and SSD performs the bounding box prediction function.

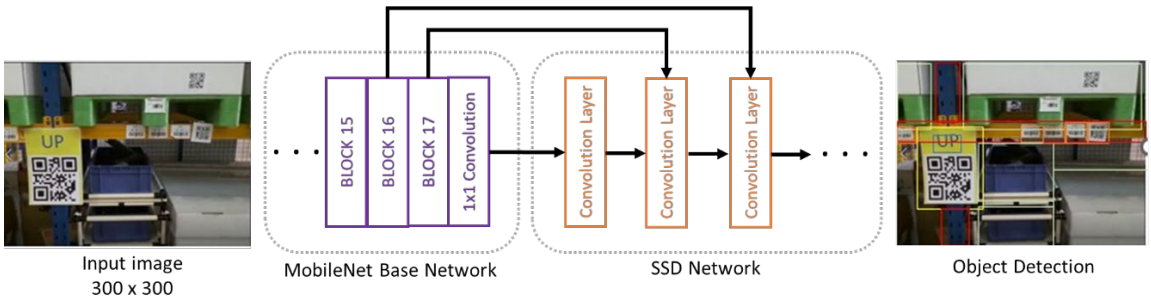

Figure 3.5: SSDLite with MobileNet for SDP Net Image Detection Layer

The evaluation was performed across 5 models to identify the most optimal option in terms of object detection model and companion computer. The main factors to determine the suitable model were based on detectability, accuracy, and performance. In our application for SDP, the object detection model should be able to detect and identify an object with high confidence to be considered reliable. The 5 models used in this evaluation are as follows:

1. SSD_Mobilenet_V1
2. SSD_Mobilenet_V2
3. SSDlite_Mobilenet_V2
4. Faster_R-CNN_inception_V2
5. RFCN_Resnet101

As compared to its more accurate Region Proposal Network (RPN) predecessors that uses 2 step function, SSD improves speed of detection by removing the need to generate regions of interest (ROI). This was achieved by using a single-stage approach in convolution to detect multiple objects. "Single stage" in SSD refers to the fact that it performs object detection in a single pass or stage, compared to a two-stage approach used by R-CNN models. This algorithm breaks down the image into a series of bounding boxes on each layer of the feature map to predict the presence of an object in each bounding box.

Detectability is defined as the model's ability to identify an object or multiple objects classifications within an image frame. This is important in real-time navigation applications as the failure to detect objects from the camera's field of view for every image frame will potentially result in collision between the drone and the undetected object. Detectability is also affected by the number of entities on an image that is presented to the object detector. For example, an object detection test between SSD_Mobilenet_V1 and SSD_MobileNet_V2 was performed on an image as shown in Figure 3.6.

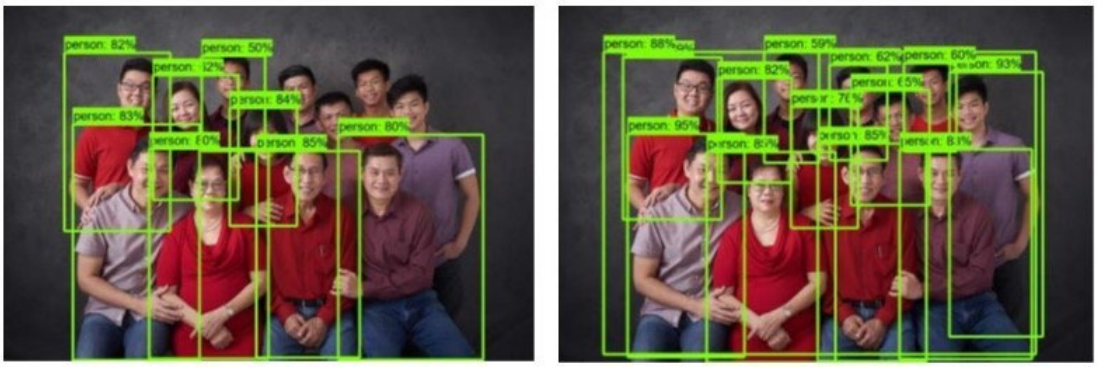

Figure 3.6: Detectability between MobileNet V1 (left) and V2 (right).

An additional test was performed for the 5 models against 50 images containing 148 object classifications as shown in Figure 3.7. ResNet 101 outperformed the other models with 1 object undetected and a 99% detection rate as compared to SSD MobileNet V1 with the lowest detection rate of 85% and 25 objects undetected.

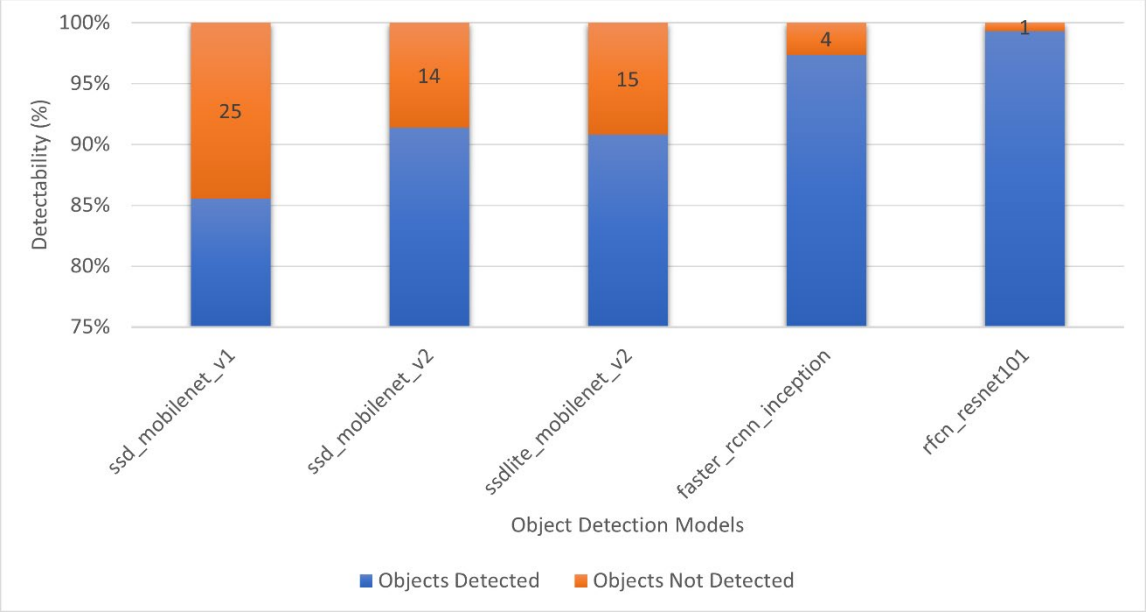

Figure 3.7: Detectability Results (%) of 5 Object Detection Model for 50 images with 148 object labels.

Another comparison between SSD MobileNetV2 and ResNet 101 shown in Figure 3.8 proves that ResNet 101 can detect small objects within an image. For the person wearing a wristwatch, ResNet 101 was able to detect the 2 classes of objects i.e. the watch at 73% and the person at 99% as compared to SSD MobileNetV2. Since ResNet 101 is a region based fully convolution network and performs object classifications at the multiple ROI within an image, it can accurately detect and classify a person in the image by associating feature maps and its votes to determine the location and class the person.

Although ResNet 101 model provided more reliable detection result with multiple ROI, with detectability as an important function for object recognition applications to identify the presence and location of each object in an image, it is not important in our application.

Figure 3.8: Detectability of SSD_Mobilenet_V2 (left) vs RFCN_Resnet101(right)

The confusion matrix method was used to assess the prediction accuracy for the 5 object detection models that we have shortlisted. This matrix is classified into 4 conditions as specified below:

Table 3.1: Confusion Matrix

N = 100	Predicted – No	Predicted – Yes
Actual – No	True Negative (TN) – 50	False Positive (FP) - 5
Actual – Yes	False Negative (FN) – 10	True Positive (TP) -35

Where:
TN = Model predicts correctly no object is present
TP = Model predicts correctly that object is present
FN = Model predicts wrongly that no object is present
FP = Model predicts wrongly that object is present

Where accuracy is:

$$Accuracy = \frac{TP + TN}{N}$$

The results in table 3.2 show that accuracy corresponds with the detectability amongst the 5 models with ResNet 101 outperforming the MobileNet variants. However, the accuracy of the SSD MobileNet variants is still relatively high with SSD MobileNet V2 at 85% accuracy.

Table 3.2: Accuracy Results of 5 Object Detection Model

SSD MobileNet V1	SSD MobileNet V2	SSDlite	Faster RCNN	RFCN Resnet101
Mean: 81.6	Mean: 85.19	Mean: 83.11	Mean: 91.45	Mean: 95
Median: 85	Median: 90	Median: 87	Median: 98	Median: 99
Std Dev: 12	Std Dev: 12.67	Std Dev: 12.69	Std Dev: 13	Std Dev: 8

Frames per second (FPS) is used as a performance indicator for the object detection models to determine how fast each model is. The higher the FPS indicates a high-performance detection model as compared with a model with low FPS. In relation to applications for autonomous navigation, a detection model with 10 FPS would indicate that it is able to process 10 continuous image frames from the video stream every second. This is an important feature for an appropriate model as a higher FPS model would result in a low latency object detection for real-time navigation models. FPS can also be affected by resolution as higher resolution images require longer time to process as compared to lower resolution images.

In this evaluation, the FPS for each model was compared against the 3 resolutions to determine the performance of each model. From the results shown in figure 3.9, it is noticeable that the stronger detection models require longer processing time for each image to pass through its complex neural network architecture, resulting in lower FPS as compared to SSD neural networks. The higher resolution images also resulted in a lower FPS when compared within each detection model due to the higher pixel count that requires longer processing time as well.

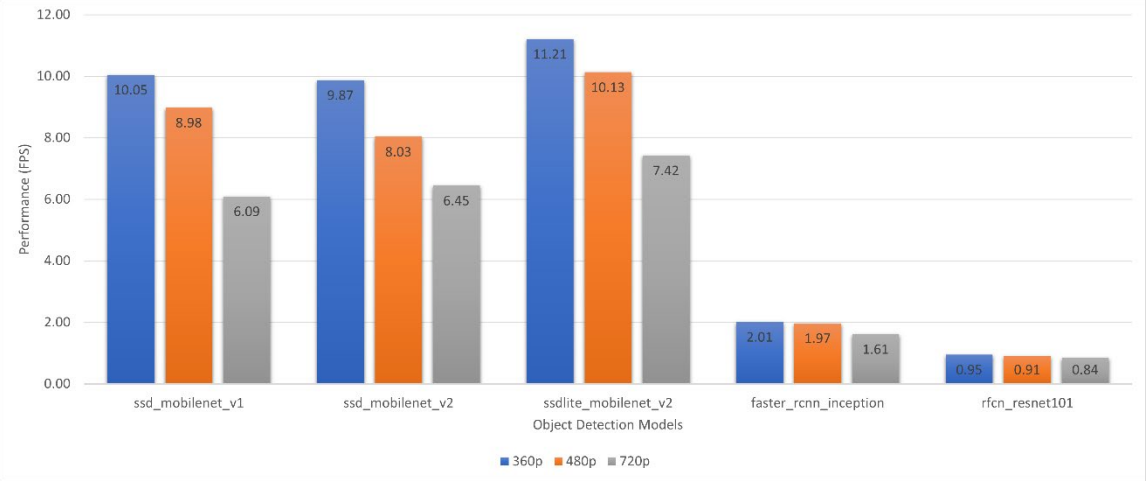

Figure 3.9: Performance Results in Frames Per Second (FPS) for 5 Object detection Models against image resolution

It is important that we choose performance over accuracy for SDP Net application since this task is a global scene understanding prior to the subsequent semantic segmentation process.

State-of-the-art base network models such as Faster RCNN [48] and ResNet 101 [49] have been proven to accurately detect fine details in an image at the compromise of performance as it could only process low Frames Per Second (FPS) as compared to MobileNet models. MobileNet models provide an acceptable accuracy rate above 80% with between 6-7 FPS for 720P images when it was tested against MS COCO data set as shown in Table 3.3. It is important that we choose performance over accuracy for SDP Net application since the UA is only required to determine the higher-level object classification for subsequent semantic segmentation process.

Table 3.3: Comparison of base network models using MS COCO data set.

	SSD MobileNetV1	SSD MobileNetV2	SSDLite	Faster RCNN	RFCN ResNet101
Detectability (%)	83	90	89	97	99
Accuracy (%)	81.6	85.19	83.11	91.45	95
Performance @ 720P (FPS)	6.09	6.45	7.42	1.61	0.84
Performance @ 480p (FPS)	8.98	8.03	10.13	1.97	0.91
Performance @ 360P (FPS)	10.05	9.87	11.21	2.01	0.95

3.3.2 Joint Semantic Depth Layer

Joint Semantic Depth Inference, as an advanced technique for deducing depth from a single 2D RGB image, integrates semantic segmentation and depth estimation seamlessly within a unified network. This method, taking a 2D RGB image as input, produces two crucial outputs: a semantic segmentation map ($F_{Semantic}$) and a depth map (F_{Depth}). The semantic segmentation map identifies objects in the image, while the depth map estimates the depth of each recognized object. The neural network undergoes training using the NYU2 and SunRGB-D datasets, combining supervised learning to recognize objects and unsupervised learning for autonomous depth discernment. The training objective is a combination of semantic ($L_{Semantic}$) and depth (L_{Depth}) losses: Objective = $\lambda_{Semantic} * L_{Semantic} + \lambda_{Depth} * L_{Depth}$ where $\lambda_{Semantic}$ and λ_{Depth} are weighting factors. The network's ability to infer depth accurately from a single RGB image is pivotal for applications like autonomous navigation. Employing a semi-supervised learning strategy, the NYU2 and SunRGB-D datasets facilitate depth computation tasks, aligning depth information with a pre-trained model for an effective inferred depth model. This comprehensive methodology defines spatial characteristics and inferred depth, significantly enhancing applications reliant on precise depth perception.

From the data shown in table 3.4, SegNet was chosen to be used in SDP as it was evaluated to be an efficient semantic segmentation model for the 2nd layer of SDP. SegNet architecture shown in Figure 3.10 is made up of 13 convolutional layers in the encoder and decoder with the last decoder output being fed through a multi class soft-max classifier to produce the classification for each pixel. For our purpose of autonomous navigation, we have assumed that it is not critical to further breakdown each classification into sub classifications e.g. sub classify humans into male or female, carton boxes into various colors or materials. The motivation to use the more efficient SegNet is the lower memory requirements as opposed to other larger models such as UNet [52] or DeconvNet [53] since the latter models use entire feature maps instead of maximum pooling indices to up sample the layers in the decoding process, making the models larger resulting in more memory required.

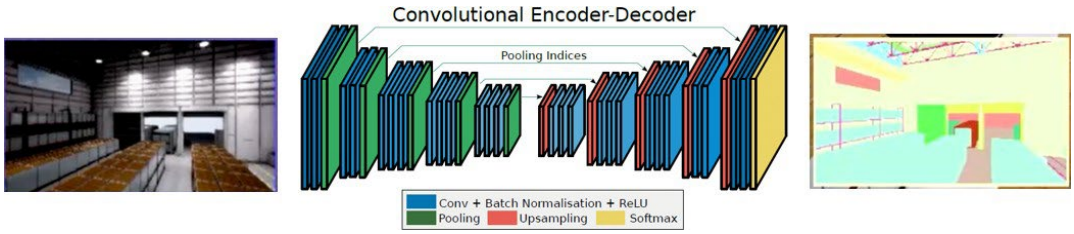

Figure 3.10: Semantic Segmentation using SegNet Model for pixel level classification.

Song et al [45] evaluated SegNet using 2D RGB images from road scenes and SunRGB-D data sets and achieved good results. since SegNet only stores the max pooling indices of featured maps and uses them in the decoder network. SegNet was used to model appearances and to understand spatial relationships in 2D but does not provide depth information which is a requirement for sense and avoid functions if applied on the small UA. Pham et al [54] proposed a method to obtain 3D progressive dense semantic segmentation using RGB-D sensor that demonstrated the capability to integrate semantic segmentation into real time indoor scanning through a 2D neural network with a novel region-aware CRF model. The quantitative assessment against other semantic segmentation models is shown in table 3.4.

Table 3.4: A comparison of computational time and hardware resources against SegNet and other architectures with SegNet being the most memory efficient during inference model [19].

Network	Forward Pass (ms)	Backward Pass (ms)	GPU Training Memory (MB)	GPU Inference Memory (MB)	Model Size (MB)
SegNet	422.50	488.71	6803	1052	117
DeepLab-LargeFOV	110.06	160.73	5618	1993	83
FCN	317.09	484.11	9735	1806	539
DeconvNet	474.65	602.15	9731	1872	877

Supervised machine learning was also adopted where synthetic 2D image data sets, semantic segmentation datasets and RGBD data sets were used as initial data sets fed into respective computer vision models to build the baseline training model. Real image data as shown in figure 4.6 that is relevant to a specific operation environment can be used to supplement the baseline PTM to improve the accuracy of each computer vision model.

Semantic segmentation is then applied to a known image for further classification at the pixel level that allows multi-class segmentation using data sets from SegNet and new images for a more accurate and efficient 3D indoor scene recreation that can be applied on a UA for path planning task. The accuracy and speed of semantic segmentation can also be greatly improved by applying CNN to enable the UA to perform path planning without prior knowledge about new environments or positions of obstacles.

The possibility of sensor fusion based on attitude control and visual sensing to corroborate with deep neural networks is explored and SDP is modelled based on open-source datasets (like NYU2 and SunRGB-D) and data sets taken from simulation and manual flights.

3.4 Training and Testing Model

Training is important to derive a Pre-Trained Model (PTM) that SDP can use for the purpose of image and pixel classification. For this project, SegNet was pre-trained with NYU2 and SunRGB-D dataset was used as the basis for SDP baseline training data to establish the following PTMs as shown in Figure 3.11. Transferred learning was also performed on the PTM using real image data set from a warehouse environment for the purpose of the experiment described in chapter 4.

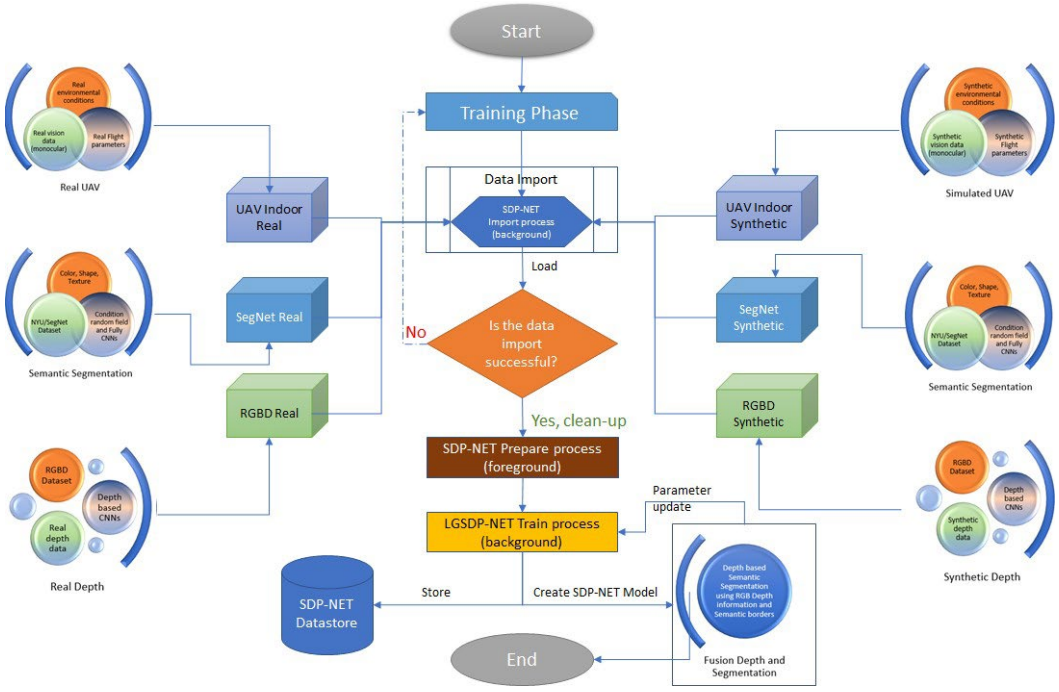

Figure 3.11: Semantic Depth Prediction (SDP) - Training Framework

After a PTM is established, the model file is transferred to the on-board computer. The training system outputs the probabilities of safety measures (M value) based on the semantic depth prediction model. Every frame that comes out of SDP is a monochromatic understanding of how the system perceives the depth information, therefore providing Far/Mid/Near (FMN) values for every frame. The outer loop (navigation loop) takes care of how to maneuver the UA.

M-Value matrix is essentially the way SDP classifies depth values of each pixel before determining it's Far/Mid/Near (FMN) values for each image frame. The abstraction of depth values from a depth map using SegNet with the integration of datasets like NYU2 and SunRGB-D, involves a strategic approach. SegNet, with its ability at learning hierarchical features and spatial relationships, processes depth maps to predict and classify depth values for each pixel. The model undergoes training using paired depth maps and corresponding ground truth depth values from datasets like NYU2 and SunRGB-D, allowing it to associate features with accurate depth information. Mathematically, during training, SegNet minimizes a loss function L over the dataset $\{D = \{(Ii, Yi)\}$, where Ii represents depth maps and Yi represents ground truth depth values. Post-training, the SegNet output S undergoes thresholding techniques based on desired depth intervals. The segmented depth map is then divided into zones for efficient computation. Within each zone, the Far/Mid/Near (FMN) values are approximated and stored in an M-Value matrix. Mathematically, for each zone, the M-Value matrix retains (maximum, minimum, and range) information of the depth values. FMN values are essentially a virtual understanding from which the navigation loop decides how to move and is performed for SDP output. Instead of relying on depth values for each pixel, the frame is split into matrix comprising of zones to reduce the amount of processing required. For every zone, the FMN value is aggregated, and we have a matrix with max and min of FMN values for every zone.

Further analysis is carried out based on the threshold information. This adjusts the M-Values. It is required to adjust the information based on the environment's light intensity (darkness in the room should be offset to adjust for the depth). Figure 3.12 below shows an example of the SDP output for a 2D image from the UA in a simulated warehouse environment. The left image represents the depth image, and the right image represents the semantic segmentation image. M-values from the depth image as shown in figure 3.13 are then used to derive safe or unsafe zones in the M values matrix.

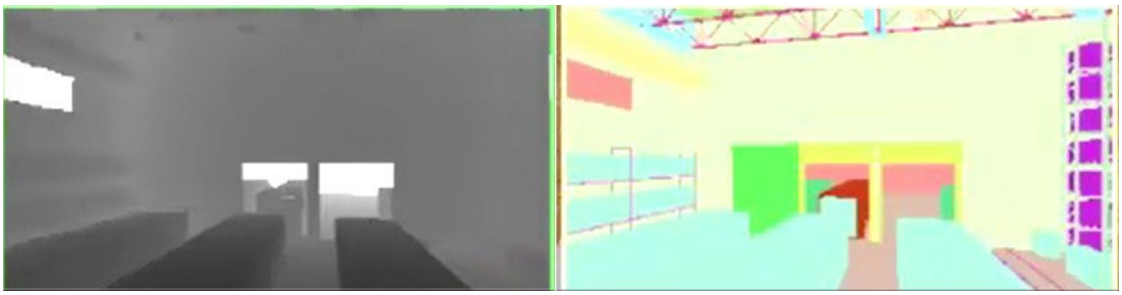

Figure 3.12: SDP Depth on the left and Semantic Segmentation Images on the right.

The output of the testing model is represented as the matrix of safety probability (M values). Here, the colors are chosen to depict the safe/unsafe zones. Higher M values indicate that the obstacles are at the far approximated distance thus indicates that the zone within the frame is safe to fly towards. M values are computed for every image frame thus the UA will continuously search navigate towards the safe zones with every new image frame.

> 90	> 90	> 50 < 80	> 90	> 90	> 90	> 90
> 90	> 90	> 50 < 80	> 90	> 90	> 90	> 90
> 90	> 90	> 50 < 80	> 90	> 90	> 60 < 90	> 60 < 90
> 90	> 90	> 50 < 80	> 90	> 90	> 50 < 80	> 50 < 80
> 90	> 90	> 50 < 80	> 90	> 90	> 50 < 80	> 50 < 80
> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60
> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60
> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60	> 40 < 60

Figure 3.13: A representation of M-Values for each image frame.

3.5 Comparison of SDP Model vs Existing Methods

Figure 3.14 denotes the comparative results of input test images, ground truth, independent SDP segmentation results as well as SDP results applied with SparseFusion (SF) and DenseFusion (DF) by SF, following by the number of combined layers used in the network (e.g., DF1 and SF5) using 5 sample test images from NYU2 dataset. Segmentation results of SDP takes into comparison with the networks trained with RGB, depth, HHA which stands for Height, Histogram, and Angle, and their combinations. The segmentation result shows that stacked RGB-D output from SDP was able identify the segmentation class labels within the foreground to the background walls and floor when compared with the ground truth data. Stacked RGB-D results also did not indicate any missing class labels when compared with the independent Depth only and RGB results. We show that SDP obtained significant improvements by extracting more informative features from depth. The inference stage is followed by a dense fully connected CRF refinement to produce the final prediction. Applying the similar loss function and post-processing, SDP is likely to produce on-par or better results.

In the second experiment, we compare the SDP to network trained with different representation of depth, to further evaluate the effectiveness of depth estimation and different variations. Stacking depth and HHA into color gives slight improvements over network trained with only color, depth or HHA. In contrast, with the depth estimation of SDP, we improve by over a significant margin, with respect to the IOU scores. We remark that depth estimation is useful as a replacement for HHA. Instead of reprocessing a single channel depth image to obtain hand crafted three-channel HHA representation, SDP learns high dimensional features from depth end-to-end, which is more informative as shown by experiments. Since the original VGG 16-layer network has 5 levels of pooling, we increased the number of layers to validate the point of saturation.

The experiments showed that segmentation accuracy gets improved from SF1 to SF5, however the increase appears saturated up to the 4th pooling, i.e., SF4. A possible reason behind the accuracy saturation was that depth had already provided very distinguished features at low-level to compensate texture less regions in RGB, and we consistently fuse features extracted from depth into the RGB-branch. The same trend can be observed with DF layers. In the third experiment, we further compare SDP-SF5, SDP-DF1 to the network trained with RGB-D input. For class accuracy, all three network architectures give very comparable results. However, for IOU scores, SF5 outperforms in 30 out of 37 classes in comparison to the other two networks. Since the class wise IOU is a better measurement over global and mean accuracy, SDP obtains significant improvements over the network trained with stacked RGB-D, showing that depth estimation is a better approach to extract informative features from depth and to combine them with color features. Figure 3.15 shows another validation results between SDP and state-of-the-art SegNet using 4 sample test images from SunRGB-D data set.

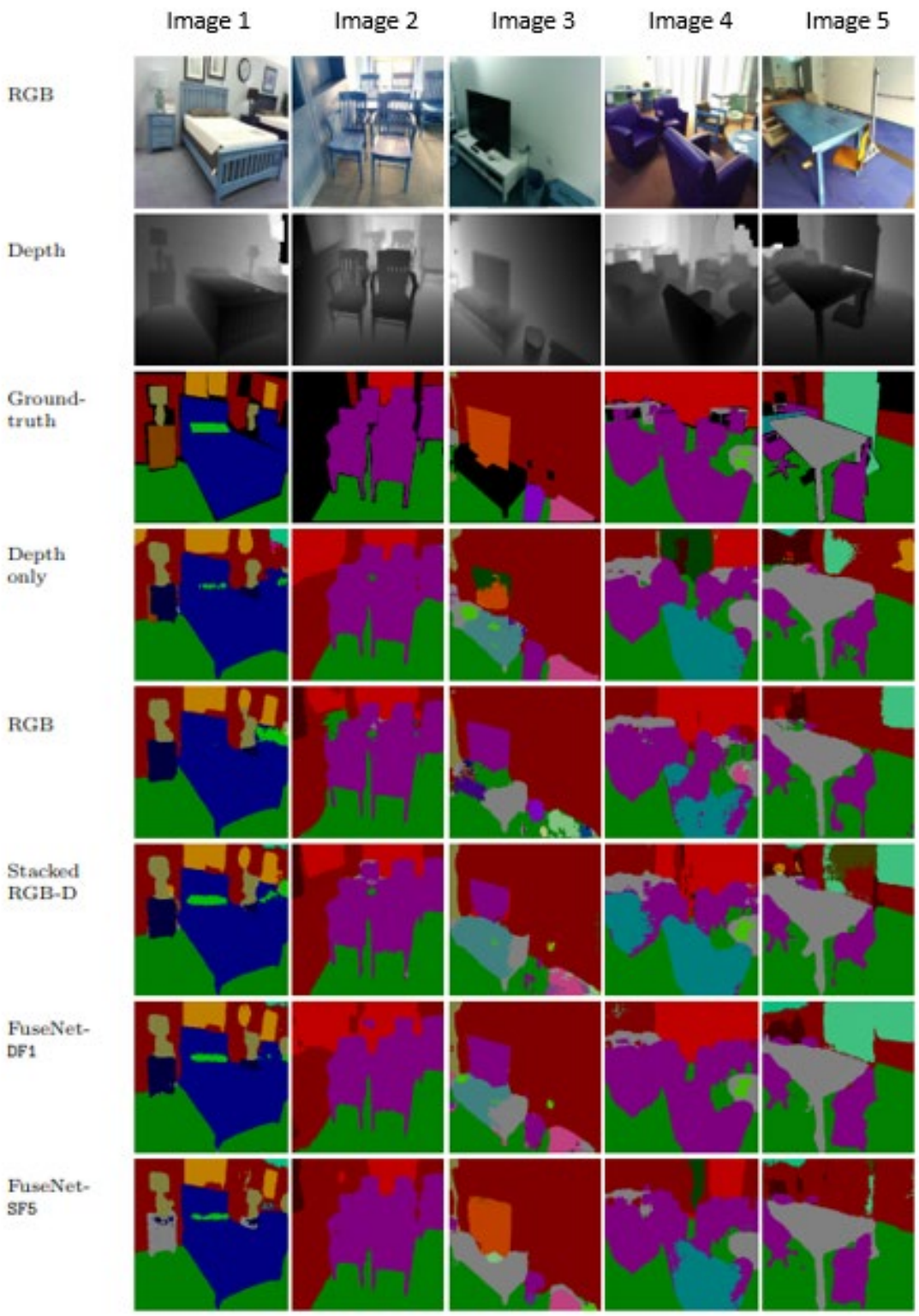

Figure 3.14: SDP against other state of the art methods

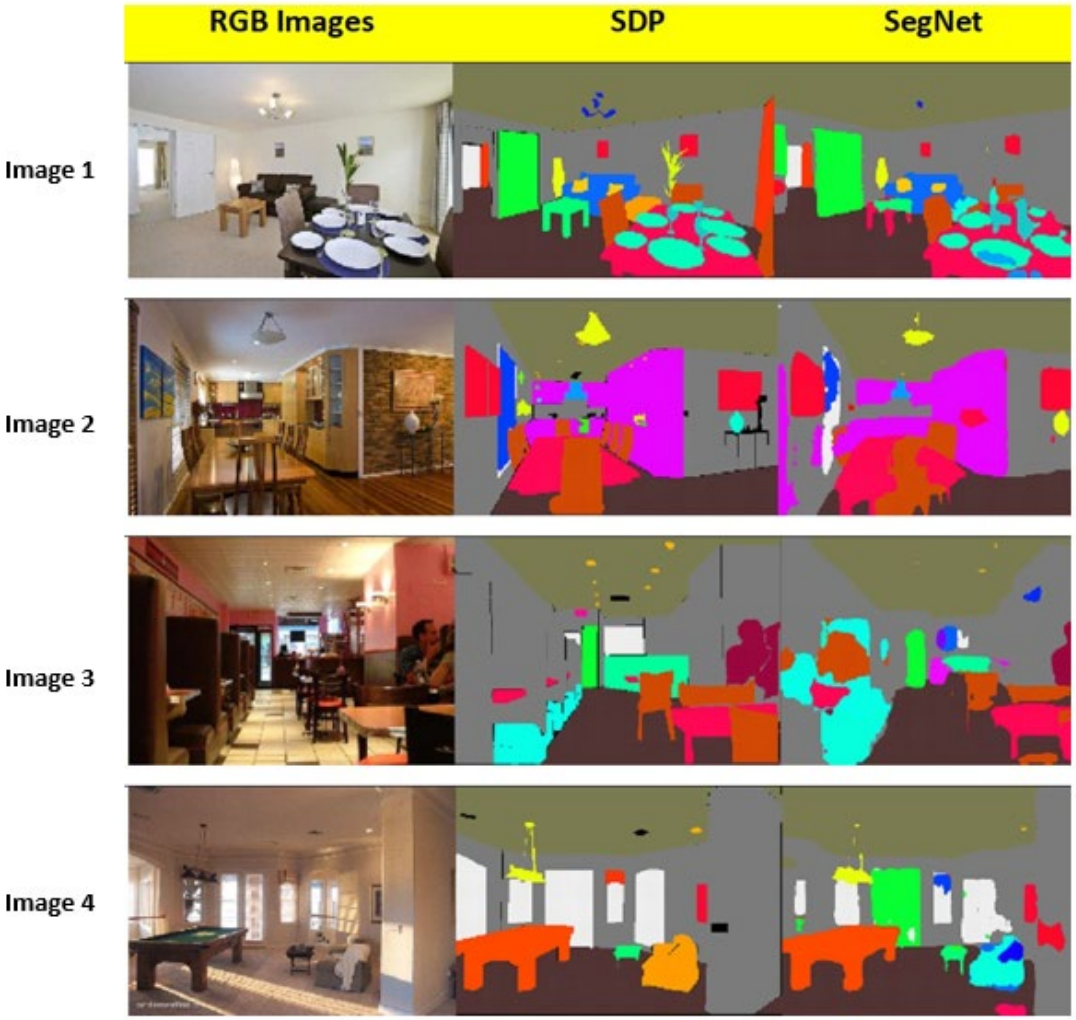

Figure 3.15: SDP against SegNet for indoor scenes (SunRGB-D data set)

3.6 Conclusion

Many CNN models have been developed for specific applications with no common model suited for all. In this chapter, an evaluation was performed to determine the possible use of available lightweight CNN models for SDP network.

For object detection and recognition layer, 5 models were evaluated using MS COCO dataset to evaluate its accuracy, detectability, and performance to identify the model for this layer. Even though RPN models such as Faster_R-CNN_inception_V2 and RFCN_Resnet101 proved to have higher accuracy and detectability scores compared to SSD models, its best performance was only at 2.01FPS as compared to SSDLite at 11.21 FPS for 360p images. The scores amongst the SSD variants were quite similar with SSDLite having the best performance amongst the 5 models across all tested FPS.

For segmentation and depth estimation layer, 4 models were evaluated using NYU2 and SunRGB-D datasets for forward and backward pass, GPU training memory, GPU inference memory and model size. Since SDP requires real-time or low-latency inferences for autonomous navigation, it is crucial to use a lightweight model with a low GPU inference memory score for the efficient use of GPU. Among the 4 models evaluated, DeepLab-LargeFOV and SegNet achieved a relatively small model size compared to FCN and DeconvNet. However, SegNet achieved the smallest GPU inference memory among the other models and is ideal for this layer.

SDP network was subsequently validated for its accuracy using the combination of SSDLite and SegNet model using both NYU2 and SunRGB-D data sets showed comparative results to other state-of-the-art networks. Stacked RGB-D results were similarly close to ground truth data and 4th pooling layer resulted in the optimal accuracy for both SF and DF. It was also observed that there was no significant loss of class labels in the segmentation output for SDP when compared between ground truth and independent SegNet results.

Overall results achieved the primary objective to combine 2 different CNN models into SDP network while retaining each model's independent accuracy and performance that would be suited for autonomous navigation as well as application-based tasks in indoor GNSS denied environment.

Chapter 4

4 Semantic Depth Prediction in Warehouse Environment

4.1 Introduction

In chapter 3, we proposed a combination of SSDLite and SegNet to create a lightweight SDP network for the purpose of inferring 3D information from 2D RGB images suited for autonomous navigation as well as application-based tasks in indoor GNSS denied environment. SDP was compared with other networks to have similar performance using NYU2 and SunRGB-D data sets. However, other challenges in an indoor environment could affect SDP's performance. Indoor environments may lack distinctive visual features that could not be identified by SDP's PTM. This can make it difficult for the UA to identify its location and navigate effectively. Actual indoor environments also contain various obstacles and structures that can occlude the camera's view making it difficult for SDP to maintain a clear understanding of the surroundings. Lighting conditions vary significantly in areas where light is obscured by objects or structures which can affect image quality and visibility. Avoiding collisions with obstacles is also crucial for safe indoor navigation. SDP must be able to detect potential obstacles and react promptly to prevent collision.

In this chapter, SDP will be validated by flying a UA system in an indoor environment. Firstly, software in the loop testing will be performed using a 6 DOF multirotor model in a simulated warehouse environment and subsequently hardware in the loop testing in an actual warehouse environment using an off- the-shelve small UA equipped with a forward mounted monocular camera. In both environments, lighting conditions were kept constant to eliminate any lighting variables to the threshold values.

4.2 Software in the Loop Testing in Airsim Environment

4.2.1 AirSim Environment

AirSim [55] is an open-source simulator for autonomous vehicles developed by Microsoft. It provides a realistic virtual environment for testing and developing autonomous systems, specifically focusing on drones and cars. AirSim is designed to be used with Unreal Engine, a popular game development engine created by Epic Games.

AirSim utilizes the capabilities of Unreal Engine to create highly detailed and visually rich virtual worlds. It simulates various environmental factors, such as weather conditions, lighting, and physics, to provide a realistic experience for testing autonomous systems. The integration with Unreal Engine allows developers to leverage its powerful rendering capabilities, physics simulation, and large-scale environments.

It can create the following conditions:

1. UA Models: Supports quadcopters, hexacopters, and custom-designed models. The models have realistic physics simulation and flight dynamics, allowing for accurate representation of flight behavior. PX4 flight controller in quadcopter was selected for the purpose of this test.

2. Flight Controller: PX4 [56] is an open-source flight control software stack designed primarily for autonomous unmanned aerial vehicles (UAVs). It is widely used in the development of commercial and academic UA projects. PX4 provides a standard to deliver drone hardware support and software stack, allowing an ecosystem to build and maintain hardware and software for scalability.

3. Environment Customization: Allows customization to simulate indoor warehouse environment to replicate real-world conditions. This enables testing drones in diverse conditions and complex environments. The warehouse environment was custom-built to recreate racking and boxes like a physical warehouse environment.

4. Computer Vision/AI: includes provisions for computer vision and AI models with UA. Sensor data can be simulated from the onboard cameras or other sensors as inputs to SDP.

5. Camera View: Custom camera view was set up in FPV perspective for RGB, Depth and Semantic Segmentation.

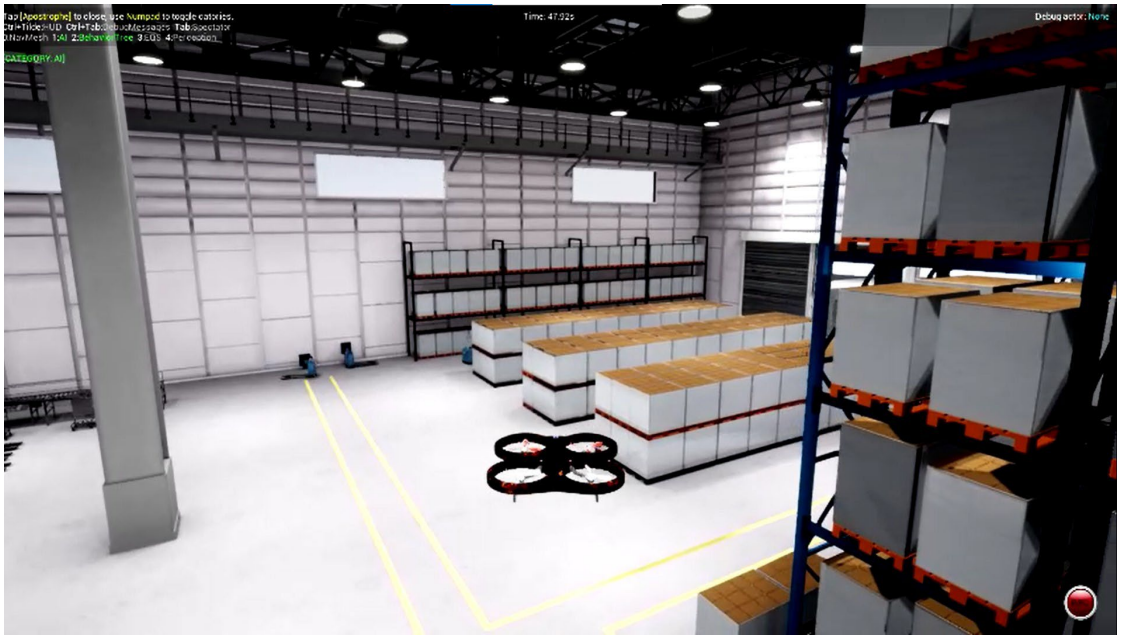

Figure 4.1: Warehouse Environment built using AirSim for SDP software in the loop testing.

4.2.2 Test Objective and Setup

The objective of the software in the loop testing was to validate the working principles of SDP. The test objective in this test case was for a quadcopter to perform a free flight within the simulated warehouse environment using SDP as the source of navigation. The quadcopter altitude was set to fly at a constant altitude of 20m above ground level and constant flight speed was set to 1m/s. +/- 1° yaw at 1hz was set to allow a wider field of view scan. 3 camera views were also set up to monitor camera image (RGB), depth estimation and semantic segmentation feed as shown in Figure 4.2. The input image from the quadcopter front camera is processed through the PTM validated in section 3 outputs quadcopter controls including steering angle to AirSim.

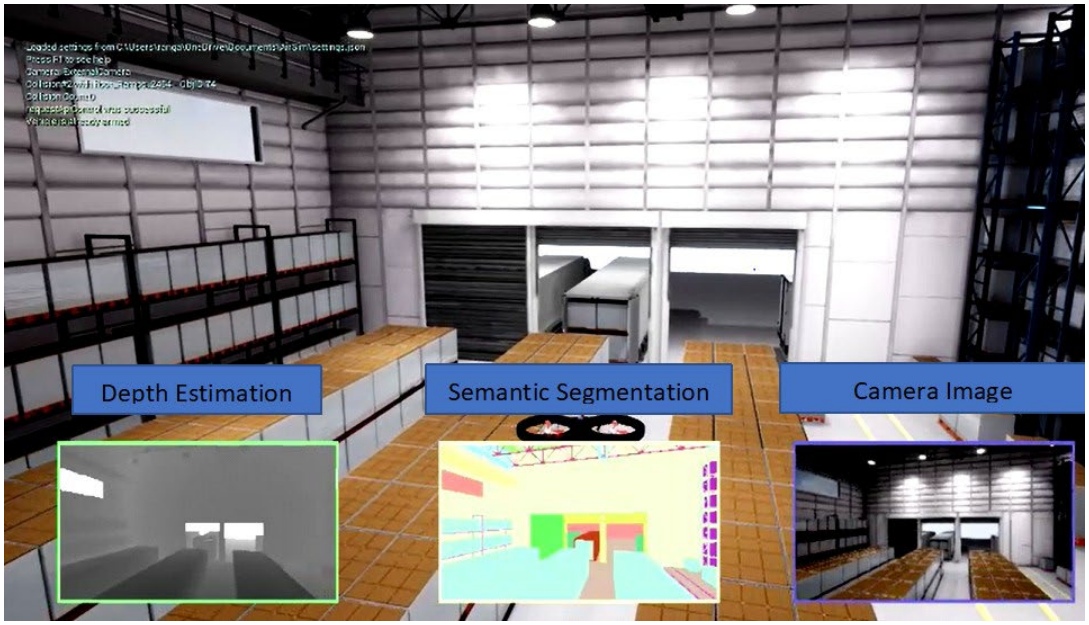

Figure 4.2: 3 camera views implemented for simulated warehouse environment.

4.2.3 Discussions

In the first flight scenario as shown in Figure 4.3, the quadcopter started its flight from the center of the warehouse and continued its flight path towards the entrance of the warehouse and made a left yaw to avoid the wall.

Figure 4.3: Simulated flight from center of warehouse to wall at 1m/s. Quadcopter made a left yaw to avoid wall and continued flight towards left wall.

In the 2nd flight scenario as shown in Figure 4.4, the quadcopter was observed to track along the wall (right of quadcopter) initially before making a left yaw to avoid the racks and continued its flight path towards the end of the warehouse.

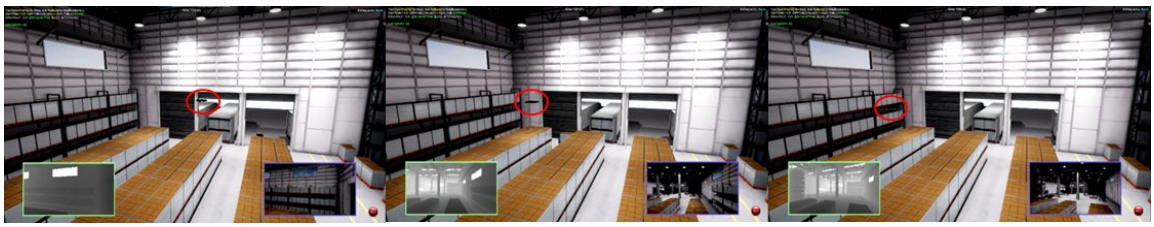

Figure 4.4: Quad copter continued its flight path along the wall before making a left yaw to avoid the adjacent rack.

In the 3rd flight scenario as shown in Figure 4.5, the quadcopter entered a narrow passage between the rack (left) and wall (right). It continued maintaining its flight path towards the end of the passageway before commencing a 90° left yaw to face the rack. It performed another 90° left yaw upon detecting the rack to face the entrance of the passage and continued its flight path towards the passage exit.

Figure 4.5: Quad copter tracked through a narrow corridor and made a 180° turn away from the end of corridor.

From the different flight scenario results in AirSim simulation, SDP was able to perform a free flight around the warehouse with zero collision rate. The +/- 1° yaw rate that was programmed to increase it field of view was effective but was noticeable with the quadcopter yawing when it is on a straight and unobstructed flight path.

4.3 Hardware in the Loop Testing in Physical Warehouse

4.3.1 Physical Warehouse Environment

An actual warehouse shown in Figure 4.6 was used for the hardware in the loop testing environment. Racks and boxes were like those that were built in the AirSim environment to minimize any variables that might be introduced that were not trained in the PTM.

Figure 4.6: Physical Warehouse Environment with racking system similar to AirSim warehouse environment.

4.3.2 Test Objective

The objective of this test in the real warehouse environment was to determine if SDP was able to recognize and perform a flight path following a specific class label and maintain a lateral separation from the racks. In contrast to the software in the loop test using AirSim, the UA will not perform a free flight in this test. Real images of the racks in yellow and blue as well as the green pallets were used to update the PTM of the SDP. Specific class labels associated in this test were specifically the boxes, racks, pallets, and the synthetic labels that were used as steering cues shown in Figure 4.7.

Images from the UA will be processed via the laptop using SDP with subsequent control commands Pitch θ, Roll φ, Yaw ψ and altitude Z commands computed and sent back to the UA.

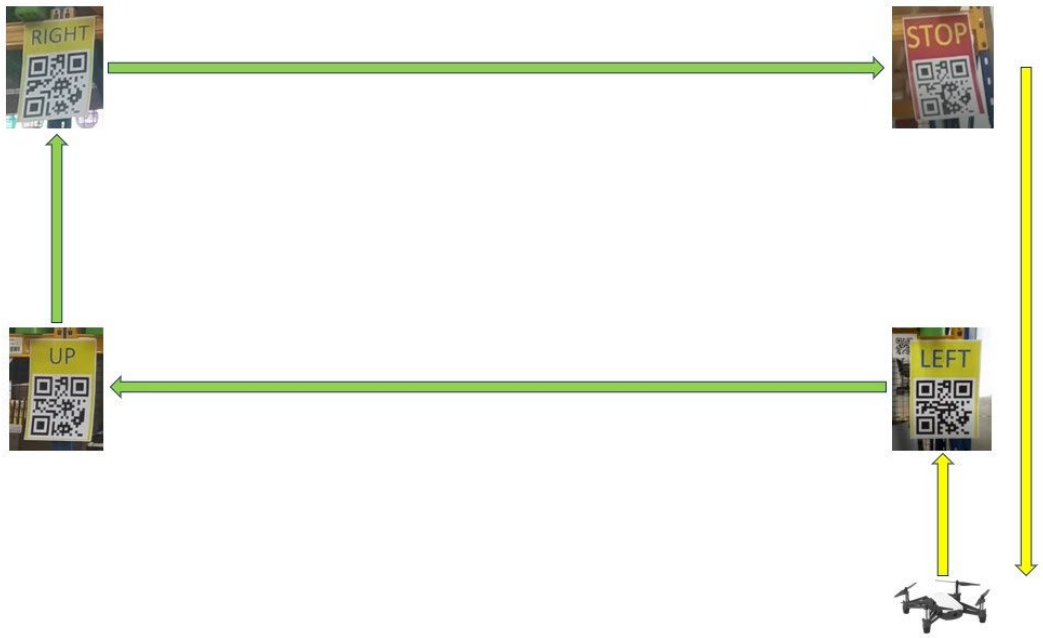

Figure 4.7: Flight path installed with synthetic steering cues using colors and letters.

4.3.3 Test Setup

The Rhyze Tello in Figure 4.8 is an off-the-shelve quadcopter used as the aerial platform for this test. It weighs approximately 80 grams and is equipped with a 720p HD video camera capable of recording at 30fps, powered by an Intel 14-core processor and 2.4 GHz 802.11n Wi-Fi module. It does not have an onboard GNSS but has 2 infra-red sensors and a downward facing camera for position hold functions. The onboard inertial Measurement Unit (IMU) includes a 3-axis gyroscope, 3-axis accelerometer and 3 axis-magnetometer provides the Tello with stability augmentation functions. All native stability augmentation functions were used for this test and only post SDP computed Pitch θ, Roll φ, Yaw ψ and altitude Z commands was injected back to the Tello. This means that without any control inputs injected back to the Tello, it will hold its last known position with minimal drift.

Figure 4.8: Ryze Tello quadcopter with forward-looking 720P monocular camera weighing 80 grams.

Since the motivation of this test was to prove that path planning can be achieved using SDP, all SDP processing was thus performed off-board the UA on a laptop to reduce the amount of computing power required on onboard the Tello. Video feed was streamed down via the 2.4GHz wireless transmission protocol to a laptop at 30fps where the images were then processed using SDP. Pitch θ, Roll φ, Yaw ψ and altitude Z commands were subsequently sent back to the Tello 2.4GHz wireless transmission protocol as shown in Figure 4.9.

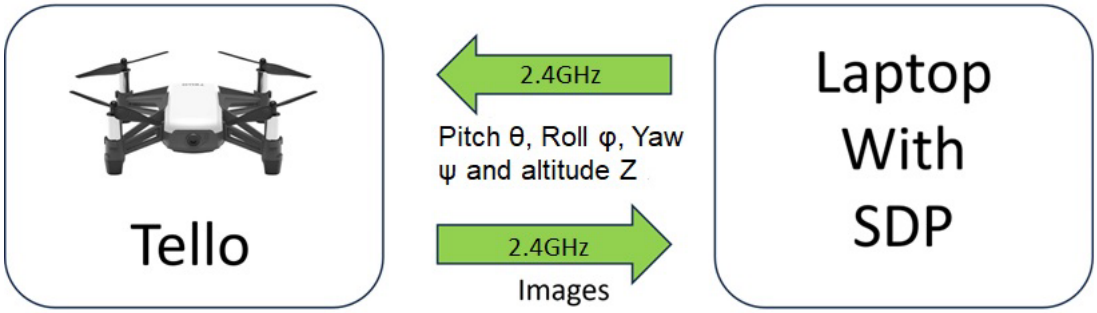

Figure 4.9: Data transmission setup

4.3.4 Flight Control Algorithm

The Rhyze Tello monocular camera as shown in Figure 4.8 processes input images at a resolution of 720P but for quicker uploading to the laptop, each image is resized to 480P. The designated flight trajectory as shown in Figure 4.7, involves an automated take-off with its camera facing the rack with a controlled velocity of 0.5m/s until the first synthetic cue is identified. Through Wi-Fi transmission as shown in Figure 4.9, every image from Tello's camera is sent to a laptop for SDP processing. Synthetic features in the form of color and letters have been positioned at the flight path corners to provide cues via object detection and recognition for commanding the Tello to roll left, climb, roll right, and stop to land through a python script when each feature of interest is detected by SDP. SDP object recognition and detection model analyze each image to determine if the captured features align with cues trained for detection. Tello executes flight control actions for alterations in Pitch θ, Roll φ, Yaw ψ, and altitude Z when

relevant features have been identified. In the absence of detected features, Tello maintains its flight path by tracking segmented labels and maintaining depth separation until the next feature of interest is detected. If no objects are detected in an image, the values for Pitch θ, Roll φ, Yaw ψ, and altitude Z remain unchanged.

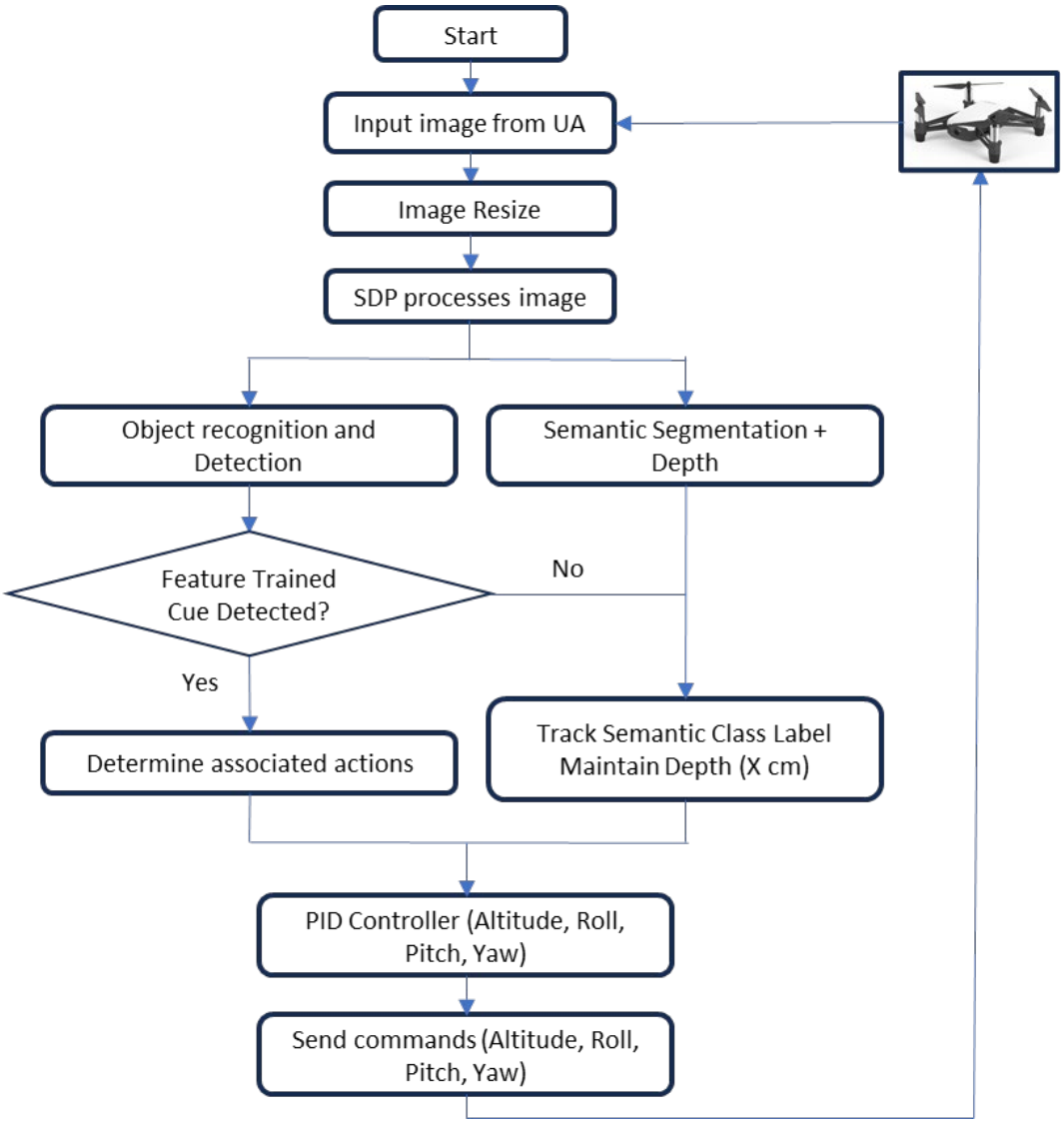

Figure 4.10: SDP Flight Control Logic with Rhyze Tello

4.3.5 Discussions

Below are the qualitative data abstracted from the flight test videos. The flight was fully autonomous and the Tello was able to maintain distance between itself and the trained rack images. Upon an autonomous takeoff at 0.5m/s climb rate, the Tello ascended to the yellow placard as shown in figure 4.10 which commanded a stop to hover before commanding a left roll to track along the lateral beam of the rack while maintaining its distance using SDP.

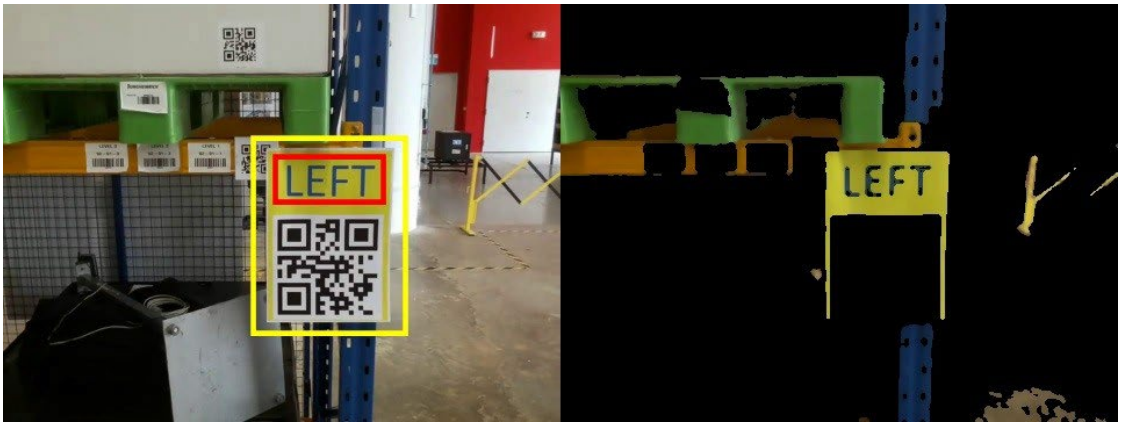

Figure 4.11: Synthetic cue to command to hover and execute left roll as shown in image on the left. The right image shows the SDP processed image.

Figure 4.11shows a sequence of images with Tello tracking the rack beam, maintaining a constant distance between the rack beam and Tello without any forward proximity sensors. It is observed that the white boxes were not processed as SDP output as it was not shown as completed boxes within the input RGB image. Other objects appear as black masks in the SDP output as they were not trained in the PTM.

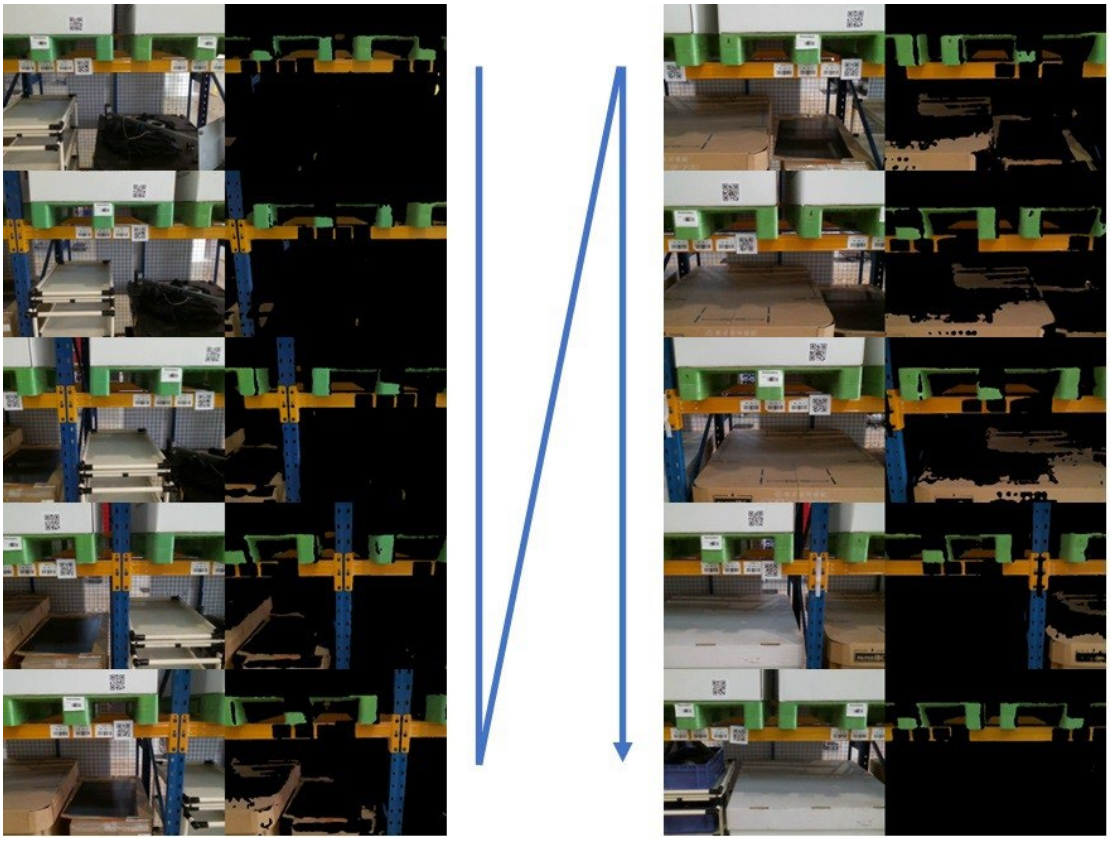

Figure 4.12: RGB input image on the left and SDP output on the right from the Tello tracking the rack beam from left to right.

As commands were set to execute upon the detection of the colored placard, it was observed that the Tello stopped hovering upon the detection of the yellow placard from the object detection and recognition model. The Tello's position was not corrected to the middle of the image frame but was positioned with a left bias as shown in Figure 4.12. In the ascend, the Tello tracked along the vertical blue rack beam maintaining constant distance as shown in Figure 4.13 but with the initial left bias as shown in Figure 4.12.

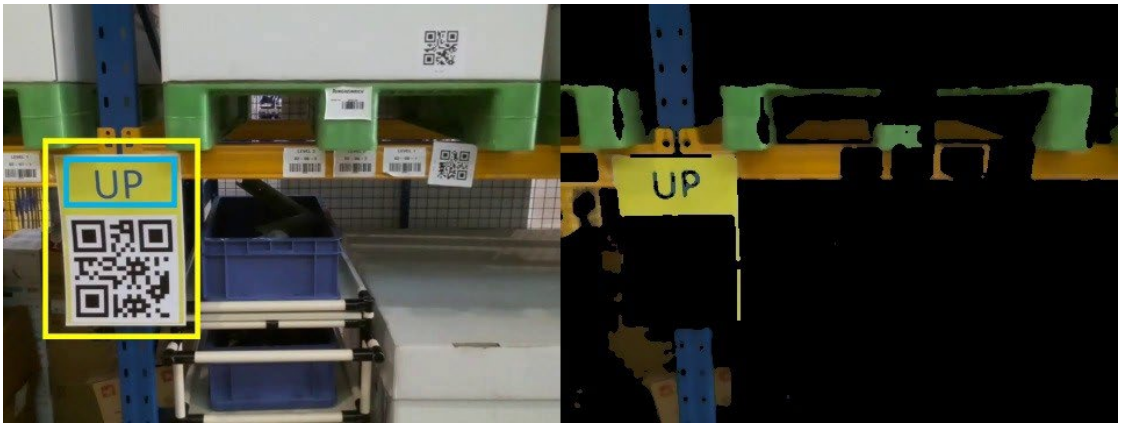

Figure 4.13: Synthetic cue to command hover and ascend as shown in image on the left. The right image shows the SDP processed image.

Figure 4.14: RGB input image on the left and SDP output on the right from the Tello tracking the rack beam in the ascend.

In the ascend as it approaches the right command placard, it was again observed that the Tello stopped to hover upon the detection of the yellow placard from the object detection and recognition model. The Tello's position was not corrected to the middle of the image frame but was positioned with a top bias this time as shown in Figure 4.14.

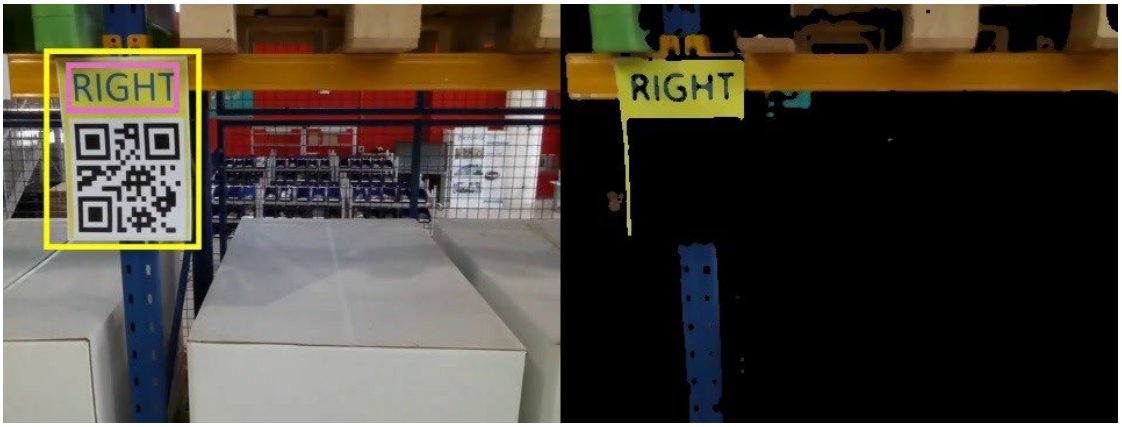

Figure 4.15: Synthetic cue to command hover and execute right roll as shown in image on the left. The right image shows the SDP processed image.

With the top bias after detecting the right command cue, it continued to track right along the rack beams with the beams biased to the top. With the accumulation of position errors in this last segment, the rack beam was at the top edge threshold of the input image frame as shown in Figure 4.15.

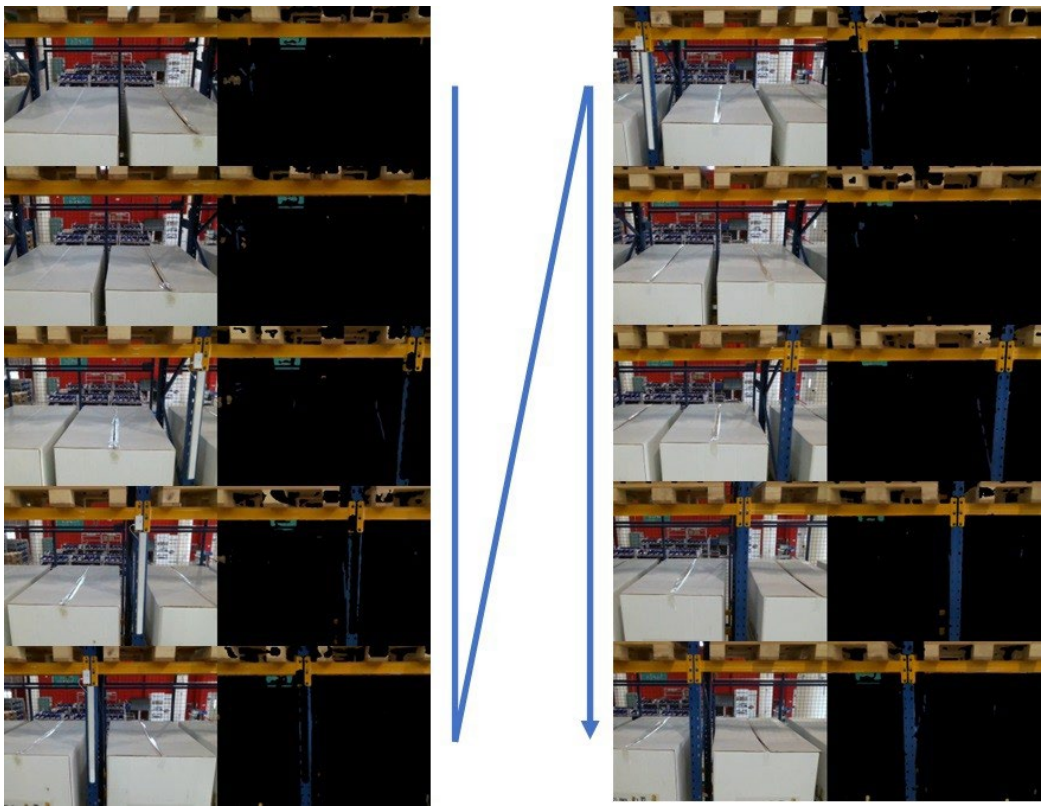

Figure 4.16: RGB input image on the left and SDP output on the right from the Tello tracking the rack beam from right to left after the ascend.

As it approaches the Stop to land command placard, it was again observed that the Tello stopped to hover upon the detection of the red placard from the object detection and recognition model and executed the stop to hover with the placard positioned on the extreme right of the input image as shown in Figure 4.16.

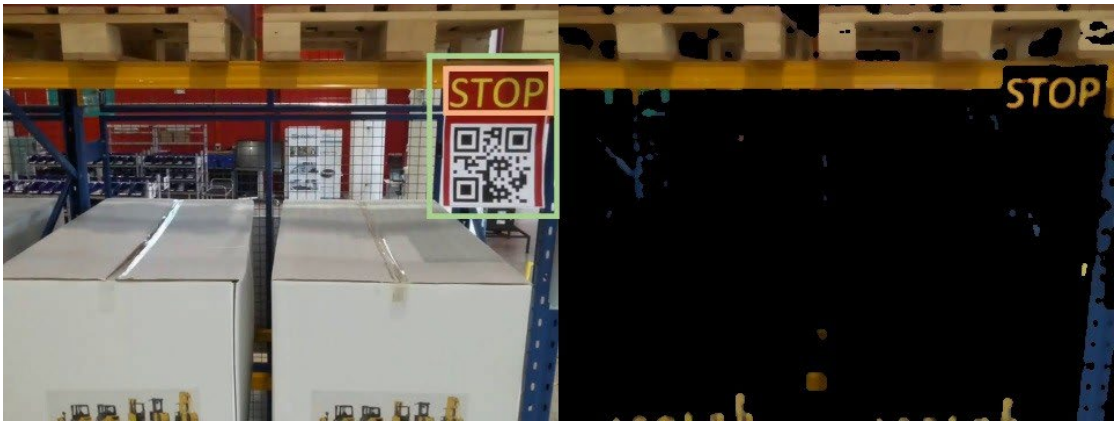

Figure 4.17: Synthetic cue to command to hover and execute landing as shown in image on the left. The right image shows the SDP processed image.

4.4 Conclusion

SDP was tested in both software and hardware in the loop to validate the possibility of combining of lightweight CNN models to recreate 3D scene using a 2D image from a monocular camera without additional sensor hardware for the purpose of autonomous navigation in a GNSS denied environment. A PTM using NYU2 and SunRGB-D open-source datasets and real data set from the indoor warehouse testing environments were sufficient for SDP to perform avoidance function in the AirSim testing as well as feature tracking autonomous flight with depth control in the actual warehouse environment. The quadcopter in AirSim was able to detect and avoid features based on the SDP and was able to continue its flight path by locating areas within the image of high threshold values in the free flight mode.

Testing with the Tello in the physical warehouse had its challenges as features of interest need to be kept within the center of each frame of image to eliminate the possibility of losing features of interest over time. Once key features such as the rack beams or synthetic cue fall out of the input image frame during flight, SDP will not be able to perform depth inference or path planning functions. Missing objects in SDP output was also likely due to partially occluded objects captured in the input image as existing networks were not able to handle missing information and thus could not detect objects under occlusion [57].

Chapter 5

5 Conclusion and Future Work

5.1 Conclusion

This thesis mainly focused on using a combination of proven lightweight CNN models for 3D scene reconstruction using 2D images for small UA. Weight dependent UA regulations set by national aviation authorities have influenced leading UA manufacturers to consider weight requirements during the development of UA related technologies. Since majority of modern UA systems are equipped with monocular cameras as an integral part of the system for the purpose of aerial photography and videography, the same camera can also be used for autonomous navigation without significantly increasing UA weight by adding other complex sensors. 2D images from the monocular camera is capable of 3D reconstruction for autonomous navigation in obstacle rich GNSS denied indoor environments and the same image data can concurrently be useful for other non-navigational tasks that is equally important in commercial real-world applications.

The motivation to scale down indoor autonomous UA has motivated the use of lightweight mobile CNN models to perform autonomous navigation without the need for additional navigation sensors that would increase the weight and size of the UA. In this thesis, several of the objectives were accomplished:

1. A study showed how global UA regulations had led to smaller UA designs driving the use of vision-based navigation to achieve the sizable footprint for safe indoor applications.
2. Combining lightweight mobile CNN models to achieve vision-based navigation using image detection and recognition, semantic segmentation, and depth inference methods for 3D scene reconstruction.
3. Validation of SDP model for autonomous navigation in AirSim and physical warehouse with possible improvements to address the issues of centroid tracking of objects as well as missing objects due to partial occlusion from input images.

As the demand for small autonomous UA operations in indoor environments increases, there is a need to explore other indoor scenes apart from warehouse environment such as indoor security surveillance or crops monitoring in an indoor greenhouse where the need for small UA is crucial to reduce the risk of collateral damage and other safety concerns.

In chapter 4, testing was conducted in a warehouse environment with consistent lighting using Tello's built in HD (720P) camera. Lighting conditions may vary if used for security surveillance of an office building where lighting may be inadequate after office hours or when intense sunlight passes through the greenhouse during midday. Varying lighting conditions can affect the SDP approach since camera-based approaches are sensitive to light.

5.2 Future Work

The following factors can be further evaluated to determine how varying lighting conditions can affect vision-based object detection and semantic segmentation performance.

1. Pixel size determines the resolution of each image. For low light applications, it is suggested that pixel size increases to allow more photons to be collected within each pixel. However, increasing pixel size would decrease the number of pixels for the same sensor size and can cause the degradation of resolution that can affect the performance of image segmentation in SDP approach.

2. Noise can be a problem as it creates a grainy effect in low light conditions and can cause distortion to an image resulting in the lack of details. When the light intensity is close to the noise level of the camera sensor, some of the pixels will appear as noise randomly.

In our experiment, Wi-Fi communication was used between the Tello and laptop for the purpose of demonstrating SDP within line-of-sight range in a localized indoor warehouse environment. Wi-fi technology is designed for Local Area Network (LAN) and is limited by its range thus requires a network of Wi-Fi nodes to expand its coverage. In contrary, 5G mobile technology on the other hand is a cellular network technology designed for Wide Area Network (WAN) mobile communications and can provide high-speed, low-latency wireless connectivity for mobile devices, IoT devices, and other applications outside of traditional Wi-Fi networks. Since commercial UA applications typically operate in Beyond Visual Line of Sight (BVLOS) conditions, implementing SDP with 5G mobile network as the communication medium will allow higher bandwidth data transfer at BVLOS range which is crucial for higher quality images to be transferred to the ground server to compute control inputs back to the UA at low latency.

Key advantages of 5G mobile network includes:

1. Ultra-high reliability and low-latency connectivity for navigation or AI on the edge without heavy onboard computing.
2. Effective for Beyond Visual Line of Sight (BVLOS) Operations without the need for the UA to establish Line of Sight communication with the ground control station.
3. Scalable network system for swarm operations to provision for localized SDP information to be shared across multiple agents.

The fusion of Convolutional Neural Networks (CNNs) and Computer Vision (CV) in autonomous navigation for small drones signifies a significant leap forward, offering enhanced object recognition, adaptability to dynamic environments, and improved situational awareness. In GNSS-denied indoor settings, this innovation empowers small Unmanned Aircraft (UA) for safer and more reliable autonomous flight without increasing its size and weight with additional onboard sensor and computing hardware.

The continuous evolution of CNNs and CV technologies not only augments navigation precision but also promises increased accessibility, reducing barriers to entry across industries. These advancements hold the potential to broaden the applications of small drones in areas such as surveillance, inspection, and search and rescue, ushering in an era of unprecedented capabilities for autonomous systems.

Bibliography

[1] Yohanes Khosiawan and Izabela Nielsen. A system of UAV application in indoor environment. Production & Manufacturing Research, 4(1):2–22, 2016. https://dx.doi.org/10.1080/ 21693277.2016.1195304.

[2] L Wawrla, O Maghazei, and T Netland. Applications of drones in warehouse operations. Whitepaper. ETH Zurich, 2019.

[3] J Tiemann, A Ramsey, and C Wietfeld. Enhanced UAV Indoor Navigation through SLAM- Augmented UWB Localization. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops), pages 1–6, 2018.

[4] Alessandro Benini, Adriano Mancini, and Sauro Longhi. An IMU/UWB/Vision-based Ex- tended Kalman Filter for Mini-UAV Localization in Indoor Environment using 802.15.4a Wireless Sensor Network. Journal of Intelligent & Robotic Systems, 70(1-4):461–476, 2013. https://dx.doi.org/10.1007/s10846-012-9742-1.

[5] Nicola Macoir, Jan Bauwens, Bart Jooris, Ben Van Herbruggen, Jen Rossey, Jeroen Hoebeke, and Eli De Poorter. UWB Localization with Battery-Powered Wireless Backbone for Drone Based Inventory Management. Sensors, 19(3):467–467, 2019. https://dx.doi.org/10.3390/ s19030467.

[6] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai Al-Ammar, and Hend Al-Khalifa. Ultra-Wideband Indoor Positioning Technologies: Analysis and Recent Advances. Sensors, 16(5):707–707, 2016. https://dx.doi. org/10.3390/s16050707.

[7] A R Khairuddin, M S Talib, and H Haron. Review of simultaneous localization and mapping (SLAM). In 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pages 85–90, 2015.

[8] Rongbing Li, Zhang Liu Jianye, Hang Ling, and Yijun. LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments. DGON Inertial Sensors and Systems, 2014.

[9] Kumar, G.A.; Patil, A.K.; Patil, R.; Park, S.S.; Chai, Y.H. A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification. Sensors 2017, 17, 1268. https://doi.org/10.3390/s17061268

[10] Y Lu, Z Xue, G S Xia, and L Zhang. A survey on vision-based UAV navigation. Geo-Spatial Inf. Sci, 21(1):21–32, 2018.

[11] T Taketomi, H Uchiyama, and S Ikeda. Visual SLAM algorithms: a survey from. IPSJ Trans. Comput. Vis. Appl, 9(1):16–16, 2010.

[12] Jorge Artieda, José M. Sebastian, Pascual Campoy, Juan F. Correa, Iván F. Mondragón, Carol Martínez, and Miguel Olivares. Visual 3-D SLAM from UAVs. Journal of Intelligent and Robotic Systems, 55(4-5):299–321, 2009. https://dx.doi.org/10.1007/s10846-008-9304-8.

[13] G Balamurugan, J Valarmathi, and V P S Naidu. Survey on UAV navigation in GPS denied environments. International Conference on Signal Processing, pages 198–204, 2017.

[14] Christoforos Kanellakis and George Nikolakopoulos. Survey on Computer Vision for UAVs: Current Developments and Trends. Journal of Intelligent & Robotic Systems, 87(1):141–168, 2017. https://dx.doi.org/10.1007/s10846-017-0483-z.

[15] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S. Lew. A review of semantic segmentation using deep neural networks. International Journal of Multimedia Information Retrieval, 7(2):87–93, 2018. https://dx.doi.org/10.1007/s13735-017-0141-z.

[16] Javier Abellan-Abenza, Alberto Garcia-Garcia, Sergiu Oprea, David Ivorra Piqueres, and Jose Garcia-Rodriguez. Classifying Behaviours in Videos with Recurrent Neural Networks, 2017. https://dx.doi.org/10.4018/ijcvip.2017100101.

[17] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. IEEE Transactions on Intelligent Transportation Systems, 2020.

[18] A Badrinarayanan, R Kendall, and Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2015.

[19] L Song, S Herranz, and Jiang. Depth CNNs for RGB-D scene recognition: learning from scratch better than transferring from RGB-CNNs, 2018.

[20] Ding. Indoor object recognition using pre-trained convolutional neural network. In 2017 23rd International Conference on Automation and Computing (ICAC), pages 1–6, 2017.

[21] Gilles & Albeaino, Gheisari, and Bryan Franz. A Systematic Review of Unmanned Aerial Vehicle Application Areas and Technologies in the AEC Domain. Electronic Journal of Information Technology in Construction, 24:381–405, 2019.

[22] Piotr & Kardasz and Jacek Doskocz. Drones and the Possibilities of Their Using. Journal of Civil & Environmental Engineering, 2016.

[23] Pamela Cohn, Alastair Green, Meredith Langstaff, and Melanie Roller. Commercial drones are here: The future of unmanned aerial systems, 12 2017. https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems.

[24] A Mulla, J Baviskar, A Baviskar, and A Bhovad. GPS assisted Standard Positioning Service for navigation and tracking: Review & implementation. 2015 International Conference on Pervasive Computing (ICPC), pages 1–6, 2015.

[25] Pavel & Davidson, Jani & Hautamäki, Collin, and Jarmo Takala. Improved Vehicle Positioning in Urban Environment through Integration of GPS and Low-Cost Inertial Sensors, 2009.

[26] ICAO Cir 328, Unmanned Aircraft Systems (UAS). Order Number: CIR328 ISBN978-92-9231-751-5,328. https://www.icao.int/meetings/uas/documents/circular\%20328_en.pdf.

[27] Anders la Cour-Harbo. Mass threshold for 'harmless' drones. International Journal of Micro Air Vehicles, 9(2):77–92, 2017.

[28] Walter & amp; Stockwell and Brendan Schulman. Defining a lowest risk UAS category, 2017.

[29] Lawrence & Barr, Richard & Newman, Ancel, & Ersin, Christine & Belcastro, John & Foster, Joni & Evans, and David Klyde. Preliminary Risk Assessment for Small Unmanned Aircraft Systems, 2017.

[30] Aurello & Patrik, Utama, & Gaudi, Alexander & Gunawan, Chowanda, & Andry, Suroso, & Jarot, Shofiyanti, and Widodo Budiharto. GNSS-based navigation systems of autonomous drone for delivering items. Journal of Big Data, 2019.

[31] J N Yasin, S A S Mohamed, M. H Haghbayan, J Heikkonen, H Tenhunen, and J Plosila. Unmanned Aerial Vehicles (UAVs): Collision Avoidance Systems and Approaches. IEEE Access, 8:105139–105155, 2020.

[32] H Shakhatreh. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. IEEE Access, 7:48572–48634, 2019.

[33] Rui Xu, Wu Chen, Ying Xu, and Shengyue Ji. A New Indoor Positioning System Architecture Using GPS Signals. Sensors, 15:10074–10087, 05 2015.

[34] J Tiemann, F Schweikowski, and C Wietfeld. Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments. 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–7, 2015.

[35] A Mashood, A Dirir, M Hussein, H Noura, and F Awwad. Quadrotor object tracking using real-time motion sensing. 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), pages 1–4, 2016.

[36] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular Vision for Mobile Robot Localization and Autonomous Navigation. International Journal of Computer Vision, 74(3):237–260, 2007. https://dx.doi.org/10.1007/s11263-006-0023-y.

[37] Yo-Ping Huang, Lucky Sithole, and Tsu-Tian Lee. Structure From Motion Technique for Scene Detection Using Autonomous Drone Navigation. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(12):2559–2570, 2019. https://dx.doi.org/10.1109/tsmc. 2017.2745419.

[38] Sudeep Pillai and John J. Leonard. Monocular SLAM Supported Object Recognition. ArXiv, abs/1506.01732, 2015.

[39] Inkyu Sa, Hu He, Van Huynh, and Peter Corke. Monocular vision based autonomous navigation for a cost-effective MAV in GPS-denied environments. In 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 1355–1360, 2013.

[40] Manfred Klopschitz, Gerhard Schall, Dieter Schmalstieg, and Gerhard Reitmayr. Visual tracking for Augmented Reality. pages 1–4, 10 2010.

[41] Wei Zheng, Fan Zhou, and Zengfu Wang. Robust and accurate monocular visual navigation combining IMU for a quadrotor. IEEE/CAA Journal of Automatica Sinica, 2(1):33–44, 2015.

[42] Andrew O' Riordan, Thomas Newe, Daniel Toal, and Gerard Dooly. Stereo Vision Sensing: Review of existing systems. 12 2018.

[43] Kannojia and Gaurav Jaiswal. Effects of Varying Resolution on Performance of CNN based Image Classification an Experimental Study. International Journal of Computer Sciences and Engineering, 6:451–456, 2018.

[44] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images ECCV 2012.

[45] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: An RGB-D scene understanding benchmark suite. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 567–576, 2015.

[46] W Liu. SSD: Single Shot MultiBox Detector. In Computer Vision - ECCV 2016. ECCV 2016, volume 9905. Springer, 2016.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: a Large- Scale Hierarchical Image Database. In IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 06 2009.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 06 2015.

[49] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in Resnet: Generalizing Residual Architectures. 03 2016.

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

[51] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12):2481–2495, 2017.

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In LNCS, volume 9351, pages 234–241, 10 2015.

[53] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. ArXiv, 05 2015.

[54] Quang-Hieu & Pham, Hua, Thanh & Binh-Son & Nguyen, and Sai-Kit Ye- ung. Real-Time Progressive 3D Semantic Segmentation for Indoor Scenes, 2019.

[55] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles, 2017.

[56] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open-source robotics framework for deeply embedded platforms," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 6235-6240, doi: 10.1109/ICRA.2015.7140074.

[57] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan Yuille. Robust Object Detection under Occlusion with Context-Aware Compositional Nets, 2020.