Design and Development of an Interactive Autonomous Monitoring and Surveillance Drone

by

Yunze Li

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Mechanical Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

April 2020

© Yunze Li, 2020

THESIS EXAMINATION INFORMATION

Submitted by: **Yunze Li**

Master of Applied Science in **Mechanical Engineering**

Thesis title: Design and Development of an Interactive Autonomous Monitoring and Surveillance Drone

An oral defense of this thesis took place on April 20, 2020 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	MARTIN AGELIN-CHAAB
Research Supervisor	HAOXIANG LANG
Examining Committee Member	JING REN
Thesis Examiner	AKRAMUL AZIM

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

This thesis describes the design and development of the novel autonomous UAV that has powerful GPU unit and potential sensing capability. It can run ROS (Robot Operating System) and compatible with packages developed under ROS. With the waterproof enclosure, it can land on the surface of water to do measurement of water parameters. Meanwhile, a human machine interaction mechanism using gesture recognition is developed and implemented for the developed drone to understand human commands. A novel way of image processing using depth information is developed for distance measurement. Meanwhile, a standard calibration method is applied and a vision-based object identification is developed and implemented in the on-board computer. It mainly achieves the function of licence plate identification and recognition. It includes a deep natural network and an OCR modular to achieve the identification, segmentation and recognition. The results showed the achievement of 95% success rate.

Keywords: Drone; design and development; machine vision; machine learning

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

_____ Yunze Li

STATEMENT OF CONTRIBUTIONS

Parts of this thesis have been published or submitted for publication in the following:

Y. Li, M. Cong, and H. Lang, "Design and development of a sensor node and autonomous UAV for real-time water quality monitoring," IEEE *International Conference on Industrial* Technology, Toronto, Canada, March 22-25, 2017.

B. Wang, Y. Li, H. Lang and Y. Wang, "Hand gesture recognition and motion estimation using the Kinect sensor," *Mechatronic Systems and Control (in press)*

The writing of the manuscripts, data collection and research conducted in this work and the mentioned publication(s) was completed by the author. Co-authors reviewed and provided technical support when required.

ACKNOWLEDGEMENTS

Firstly, I would like to express my deep gratitude to my supervisor, Haoxiang Lang, who guided me throughout this project.

Secondly, I wish to acknowledge the help provided by the technical and support staff in the Faculty of Engineering and Applied Science department of Ontario Tech University.

Last but not least, I would like to show my appreciation to my parents and family for the encouragement and support which helped me in completion of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

RGB	Red, Green, Blue
UAV	Unmanned Aerial Vehicle
HCI	Human Computer Interaction
AI	Artificial Intelligence
ASL	American Sign Language
HSV	Hue-Saturation-Value
S.U.R.F	Speed Up Robust Features
S.I.F.T	Scale-Invariant Feature Transform
GRASP	General Robotics & Autonomous Systems and Processes
DNN	Deep Neural Network
CNN	Convolutional Neural Network
R-FCN	Region-Based Fully Convolutional Network
FPN	Feature Pyramid Networks
SSD	Single Shot MultiBox Detector
DSSD	Deconvolutional Single Shot Detector
DSOD	Deeply Supervised Object Detectors
OCR	Optical Character Recognition

LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
LPRS	License Plate Recognition System
ABS	Acrylonitrile Butadiene Styrene
PID	Proportional Integral Derivative
LQR	Linear Quadcopter Regulator
OpenCV	Open Source Computer Vision Library
ROI	Region Of Interest
WSN	Wireless Sensor Network

NOMENCLATURE

K_P	*Proportional Gain*
K_I	*Integral Gain*
K_D	*Derivative Gain*
$u(t)$	*Control Input*
$e(t)$	*Error*
$x_d(t)$	*Desired State*
$x(t)$	Present State
b	Distance Between the Start and Farthest Points
c	Distance Between the End and Farthest Points
a	Distance Between the Start and End Points
m_{ji}	Spatial Moments
(x^i, y^i)	Mass Center
D_0	The Largest Distance
Z	Depth
f	Focal Length

Chapter 1. Introduction

A drone is an unmanned aircraft system that is usually comprised of an unmanned aerial vehicle (UAV), and controller that is generally on the ground and used as a form of communication between the user and UAV. Drones have a very long history with the first unmanned aerial vehicles being used as early as World War I. Since these early beginnings; colossal steps have been taken to make drones more compact, affordable, versatile, and efficient. This has made the creation, possession, maintenance and operation of a drone very simple thereby requiring minimal amount of training. Drones can be quickly deployed and only require remote pilotry, thus making them an ideal tool for use in a variety of industries. From entertainment to military uses; they have become an integral tool in various applications. These reasons also make it a perfect candidate for autonomous monitoring and surveillance applications.

The application of Unmanned Aerial Vehicle (UAV) developed rapidly since 1914 when the first military unmanned aerial vehicle was invented in Britain. It gradually turned from military utilization to civil use and recently played a critical role in disaster reduction and disaster prevention. One of most related examples is the unmanned aerial vehicles (UAV) of the Belgian Army, which was deployed for the surveillance, and the detection of oil spills in the area of southern part of the North Sea in 2008.

1.1 Scope and Objectives

The scope of the thesis is to design and develop an autonomous drone that has advanced on-board sensing and computing capability. The task requires background of Mechanical, Electrical and Software Engineering.

- Mechanical Engineering: The physical design of the drone which will account for ensuring appropriate placement of electrical components (ie. motors and wires). This also included selecting appropriate materials.
- Electrical Engineering: The interfacing between various components between the drone such as the motors, power source, flight controller and receivers which are crucial for successful operation. Meanwhile, the on-board computational power and advanced sensing are also challenging.
- Software Engineering: While the drone required programming to be controlled from a ground based remote and to program the camera, this area also allows for the most amount of future growth. Computer vision and machine learning are two key technologies that are needed to be implemented in the designed drone for vision-based monitoring and surveillance.

There are several design specifications required of the drone in this thesis which are listed below:

- **Safety**

The ability to land the drone safely upon loss of connectivity or unforeseen event. The drone must be able to land itself safely if the operator activates it (or the drone activates it upon loss of connection). It will stop motion and slowly descend to the ground. An added functionality could be to return to the point of launch via GPS coordinates and land itself with a programmed emergency flight termination protocol.

- **Accuracy in Localisation**

It is important for the drone to be able to accurately determine its own location as well as the target object's location. Since the focus of the project is monitoring and surveillance, it is important that the drone can accurately convey the location of tracked objects accurately through the combination of the GPS coordinates of the drone and the vision system.

- **Capable of Transporting Onboard Equipment**

The drone must be able to carry its own weight and equipment without creating issues with the flight characteristics. This ties into the drone being able to function predictably and it being user friendly. Since the targeted application require advanced computational power and sensing capability, the designed drone needs to have enough payload without affecting its own maneuverability.

- **User Friendly**

The drone must be easily operated to allow for safer, more stable flight. This also aids in the search for the missing person as the picture will be more stable and easier to view. The drone can be flown by at minimum novice pilot and the controller uses the same or similar controls as standard commercial drones so that anyone who has flown a drone before will easily be able to use it.

- **Sufficiently Waterproof**

In the case of undesirable weather conditions and specific application (e.g., water quality monitoring), the drone must have some weatherproofing to guard the electronics from being soaked in wet weather. This could also aid in protection of the electronics in the case of a crash and floating on the surface of water.

- **High Resolution Image**

A higher resolution image will allow for a more accurate distance measurement for stereoscopic vision as well as a more visually crisp image for the operator to view when searching for the target. The on-board vision system is required to provide sufficient vision information while. The more megapixels the camera has the more accurate and longer distance can be detected/calculated. This also ties into the accuracy in location category.

The detailed objectives of the work are as follows:

- Design and development of a functioning drone with enough payload for sensing and on-board computing units.
- Validation of the functionality of the developed drone including sensing, positioning, localization and on-board computing.
- Development of the vision-based interactive method with human.
- Design and development of deep learning based on-board object identification and tracking system.
- Validation of the developed system in different application.

1.2 Expected Contributions

The main research contributions of this thesis are listed below:

1) Design and development of a functioning drone with advanced on-board computational power and sensing.
2) Implementation of vision based integrative feature in the drone using hand gesture recognition.

3) Implementation of deep learning framework and OCR for monitoring and detection of license plates.
4) Proof of broad applications of the developed system: an example of water monitoring system using the drone is demonstrated.

1.3 Outline of the Thesis

Chapter 1 introduces the research area by discussing the motivation, scope and objectives of the work.

Chapter 2 provides a detailed literature survey of relevant work which includes current design of the drone system and their applications, autonomous navigation methodologies, advenced vision sensing and computer vision and machine learning.

Chapter 3 discusses design and development of the prototype in detail by covering all aspects of the mechanical, electronic hardware architecture design and the functionality development of the drone.

Chapter 4 explains the interactive features, specifically, the hand gesture recognition and motion estimation.

Chapter 5 proposes a real-time machine learning framework that can be applied in the developed system for efficient and accuracy vehicle and its license plate detection and recognition.

Chapter 6 presents the results from physical experiments as well as the applications fo the developed system.

Chapter 7 concludes the thesis by describing the benefits and limitations of the proposed research as well as recommendations for future works.

Chapter 2. Literature Review

A drone is an unmanned aircraft system that is generally comprised of an unmanned aerial vehicle (UAV), and controller that is generally on the ground and a form of communication between the user and UAV. As drones can be quickly deployed and operated, which makes makes an ideal tool for use in a variety of industries. From entertainment to military uses; they have become an integral tool in various applications. These reasons also make it a perfect candidate for monitoring and surveillance applications where other autonomous system can't work efficiently such as mobile ground robot. The use of drone in monitoring and surveillance application usually can cover much larger are comparing the use of mobile robotic system. Meanwhile, the functionality of the drone is not affected by the change of terrain and ground situation. Therefore, there are many available design and implementation of the drone in the market for these purposes. There are also undergoing development in the research areas where researchers are looking for solutions to overcome some of the drawbacks of the drone in monitoring and surveillance application. Some of the examples are short of computational power, lack of advanced sensing and processing capability, and autonomy. Due to the versatility of drones, further developments can be added to this drone to increase the capabilities such as autonomous abilities.

2.1 Project-related Review of Current State of the Art

Drones have a vast amount of applications in industry today. Currently many different drones are used today in environmental monitoring and surveillance. Many of these drones are custom built or more commonly repurposed existing commercial drones. DJI commercial drones, for example, are used in various applications. They are retrofitted

with the equipment they need to do the job as opposed to being custom built for that application. Another cutting-edge application of the Drone is being used in disaster and rescue situations today. It is worth noting that most of these drones have an operator and are not autonomous. A recent example of rescue drones were the rescue people off the coast in Australia by dropping an inflatable payload to the individuals in distress. Another example was the rescue of a drowning woman in Spain. The main difference between the two is that the inflatable payload in the latter is attached to the drone and must be removed from the attachment to the drone.

Open source drone platforms are widely available. Since one of the major goals of the project is the modularity of the drone, it raises the question of whether the base platform should be re-designed and built rather than following the recent popular design. Many development drone platforms are available from companies such DJI, Parrot, and Freefly, etc. [1]. Creating an affordable platform that can be developed, open source, and improved by other students will allow for a more flexible platform that is tailored to modularity and cost-effective.

In terms of software, UGCS and PX4 are examples of software that is compatible with many commercially available drones. It allows the operator to program the drone to fly specific patterns within a defined search area. It is basically an advanced surveying program that allows for telemetry, restricted airspace, altitude, and flightpath control (among various other functions). The software for UAV control is widely available and relatively sophisticated (as demonstrated by the UGCS software) [1],[2]. Drone SAR pilot is another software developed to allow for positioning and localization.

2.2 Human-machine Interaction

Human computer interaction (HCI) is a hot topic in today's research fields such as artificial intelligence (AI), computer vision, image processing and robotics [3-9]. Body gestures are commonly used in identification of human interactive gaming systems such as Microsoft Xbox and Sony Play Stations. However, hand gestures [10][11] are also an effective way in human-human communications. Such systems have been developed to implement formal sign languages (e.g., American Sign Language (ASL)), which are serving pretty well in communities of people who have disability in speaking and listening.

With the development of computer vision technologies, great progresses have been made in face recognition [3], motion estimation [4] and sign language interpretation [5]. However, there are still challenges existing in hand gesture identification and understanding due to multiple degrees of freedom geometry of human hands, poor vision systems in identifying and tracking and other factors such as occlusions and understanding the motion of fingers and hands.

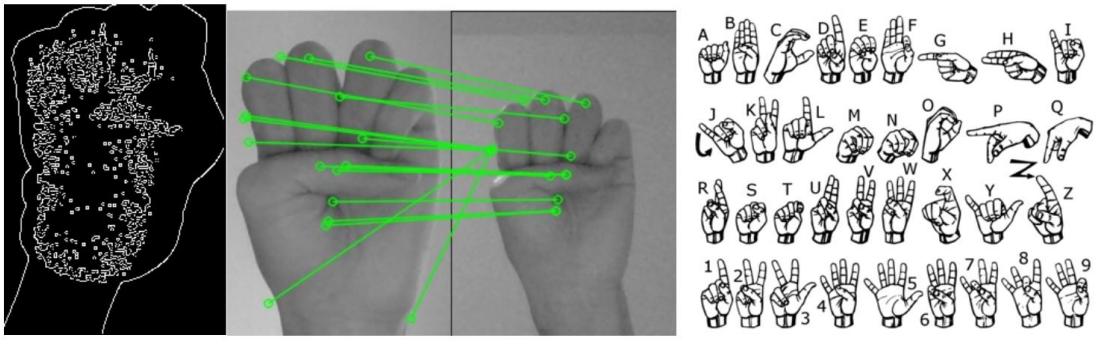
A common solution to retrieve reliable hand information and address the above challenges is to rely on wearable devices (e.g. electronic gloves [6]). Although using additional devices usually provide us with more accurate and reliable information, it will raise the cost expenses and also hamper the movements of hands. Moreover, the additional devices are usually not always available which makes it not a commonly used method.

Vision-based techniques [5-8] have become a potential solution because they are powerful non-contacting sensing technologies mimicking human vision. In most of the current applications, the HSV (Hue-Saturation-Value) color space [12] and traditional

feature-based object identification and tracking [13] are commonly applied for segmentation and identification of hands. However, some complex environments with changing background and illumination dramatically influence the performance of the system.

Recent algorithms of image processing and computer vision can fail due to environmental factors such as illumination, background and occlusions. Furthermore, vision algorithms usually result in high computational burden which can become the root cause of poor real-time performance in practice. Algorithms like S.U.R.F (Speed Up Robust Features) and S.I.F.T (Scale-Invariant Feature Transform) have been created to help the computer recognize predetermined objects in an image which work by locating feature points on an object from a reference image and then trying to find them on the actual image [14][15].

Figure 2.1 shows the feature-based hand and gesture detection developed in the GRASP (General Robotics & Autonomous Systems and Processes) Laboratory at Ontario Tech Univeristy. The project converts the images captured from a webcam to useful hand information. Then the computer vision program can decide which letter or number the user is trying to show according to the ASL (American Sign Language) which is shown in Figure 2.1(c). Figure 2.1(b) shows the detected representation of the letter “E” according to the ASL. These algorithms work well for solid immutable objects but have difficulties with objects that can drastically change their shapes like hands or liquids due to the change in feature points and their locations [16][17].



(a) (b) (c)

Figure 2.1 Feature based hand and gesture detection: (a) feature detection (a fist example); (b) matching the letter “E”; (c) letter and number representation in ASL.

Depth information, especially Kinect sensor has been unitized to improve reliability and accuracy of hand tracking. [18] reviewed recent methods and databases in vision-based hand gesture recognition, including those who are using Kinect sensor. Depth information from the Kinect sensor is utilized to improve both static and dynamic hand gesture recognition. However, there are still a lot of existing challenges in fully understand the dynamic gesture in terms of segmentation, motion estimation, hand gesture modeling and reasoning. For example, in [19], a framework for recognizing hand gestures by using the depth information is proposed. It acquires 3-D hand contour and achieve the matching by simplified hand model. The proposed method has limitation of segmentation and accurate contour model.

2.3 Deep Learning in Object Identification and OCR

Object identification is the key research area in computer vision since the understanding and perception of the environment is the fundamental problem of many

autonomous and intelligent system [20, 21]. Traditional computer vision approaches general to address the object identification problem by dividing it into several sub-questions: informative region selection [22, 23], feature extraction [24, 25], classification [26, 27] and template matching [28, 29].

Thanks to the recently developed Deep Neural Network (DNNs), research has shifted the research direction from the traditional ways to biological inspired methods, Deep Learning [30, 31] that has deeper architecture from the tradition Neural Networks. Meanwhile, there are variation of the overall network in different applications using Deep Learning such as genetic object detection [32, 33], salient object detection [34, 35], face detection [36, 37] and pedestrian detection [38, 39]. They share a similar architecture with slightly change in different layers for specific applications.

With the development of the model and applications, several models are available for difference applications that can be considered as benchmark. Some of the well-known models and architectures include: AlexNet [40], GoogleLeNet [41], VGG [42], ResNet [43]. These models are deeply studied, and analysed, new architecture and improvement of the original ones are published which create huge impact on the entire academic and industrial communities [44, 45]. With the development of the computer hardware and parallel computing technologies, network with huge number of the layers with millions of labeled training data have become possible (e.g., ReLU operation [46]).

A typical CNN (Convolutional Neural Network) architecture is the VGG16 which is shown in Figure 2.1. The advantages of the CNN against the traditional computer vision methods are 1) hierarchical feature representation, 2) achieving multi-objective optimization; 3) dealing with high-dimensional data.

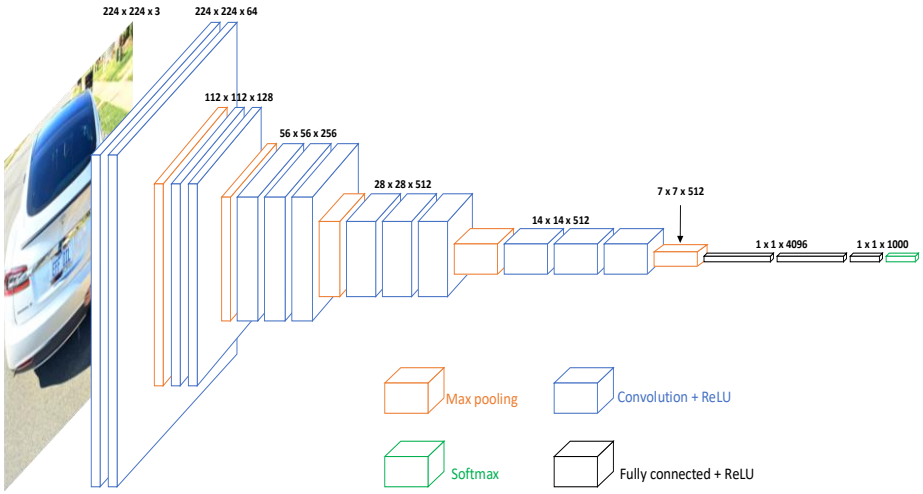


Figure 2.2 Architecture of VGG16 [47].

Generic object detection has been widely investigated by many researchers, aimed at localizing and classifying existing objects. There are two types of frameworks shown in Figure 2.3. Region proposal based method firstly generate region proposals followed by classifying each proposal into different object categories, which includes R-CNN [48], spatial pyramid pooling (SPP)-net [49], Fast R-CNN [50], Faster R-CNN [51], region-based fully convolutional network (R-FCN) [52], feature pyramid networks (FPN) [53], and Mask R-CNN [54]. On the other hand, the regression/classification-based methods, such as ultiBox [55], AttentionNet [56], G-CNN [57], YOLO [58], Single Shot MultiBox Detector (SSD) [59], YOLOv2 [60], deconvolutional single shot detector (DSSD) [61], and deeply supervised object detectors (DSOD) [62], utilizes a unified framework to achieve final results (categories and locations) directly by regarding object detection as a regression or classification problem.

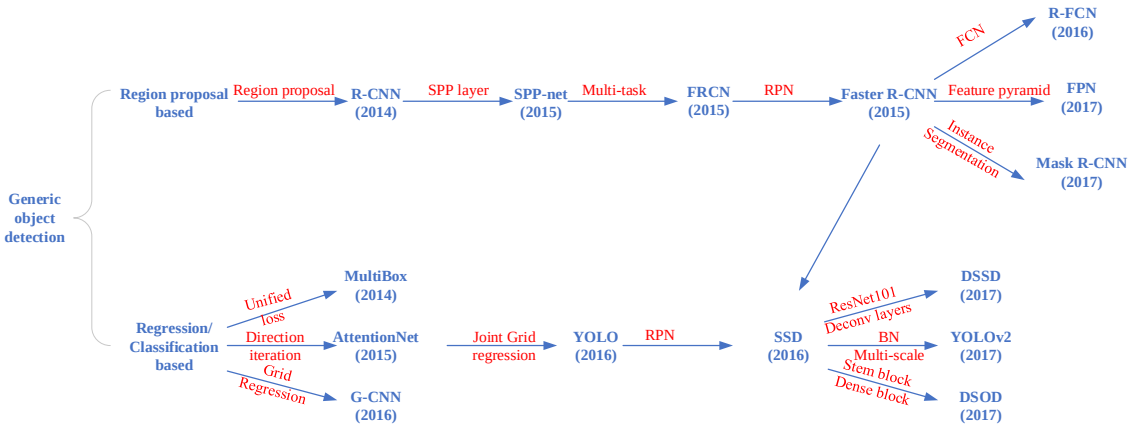


Figure 2.3 Two types of frameworks: region proposal based and regression/classification based [63].

In addition to the object identification using Deep Learning, there are other applications such as natural language processing, voice recognition and understanding, machine translation, applications in medical area. One of the areas that this thesis will be exploring is Tesseract OCR (Optical character recognition) using LSTM (Long Short-Term Memory) engine mode for recognition of pre-processed cropped LPs (license plates). Tesseract LSTM OCR is the most accurate OCR engine so far adopting an artificial recurrent neural network (RNN) architecture in the field of Deep Learning [64]. Therefore, Tesseract LSTM OCR engine will be utilized for segmentation and character recognition after licence plates being detected and cropped in my license plate recognition system (LPRS).

Chapter 3. Design and Development of the Drone

The main goal of the developed drone is to be able to complete autonomous navigation, localization which can remotely identify and locate a subject and provide position information relative to the operator of the drone. In this case, the drone will have to be able to know its position, and this is accomplished by having a GPS and altimeter on board which provide the global position of the drone. The object that is being searched for must be identified by the drone automatically by using the onboard vision and computing units. Once identified, the drone must be able to relay the location information of the target relative to the drone operator. This will be achieved by using a link transformation matrix as well as manipulation calculations to accurately determine the distance and angle of the target object relative to the drone.

The proposed system will equip with a dual camera setup utilizing stereoscopic vision so that the distance of the target from the drone can be measured. One camera will be a standard (RGB) camera. The other will preferably be a thermal or IR camera. The thermal/IR camera combined with the regular camera will allow for easier target acquisition. Once the target is found and the distance is calculated from the drone (height found with altimeter), the GPS coordinates of the drone can be utilized to inform the operator where the object/subject is relative to the operator's position.

3.1 Engineering Requirements

In order to accomplish the goals set out there are a couple of combinations of sensors that can be used. To accomplish the localization task, a stereo camera in combination with an RGB camera can be used. These would work together to determine the distance of the

person from the drone using convergence of the two camera views. The thermal camera would help with identifying the person to the drone's pilot where they are not yet visible through the RGB camera. Laser positioning in combination with an RGB camera could also be an option since laser positioning can have a long range. The final option would be to use an ultrasonic sensor to determine the distance of the target relative to the drone, of course in combination with an RGB camera. Ultrasonic sensors are useful in misty or foggy environments because dust and fog do not block the waves emitted by the sensor however, they do not have long range. Out of all the aforementioned options, the thermal camera in combination with the RGB camera is the most feasible due to average price points of equipment as well as equipment availability.

Additional Requirements

Other requirements for the drone include a GPS system so that the drone can be positioned relative to the Earth as well as a height sensor to determine elevation from the surface. To accomplish the goal of carrying a heavier payload and the ability to move quickly, the drone motors would have to be upgraded to accomplish the amount of thrust required by the added weight. At this stage, brushless motors are the best option because they are widely available on the market and are capable of producing the power output required to fly the drone. The drone should be reliable in its functionality as it is being used in urgent situations. This is accomplished mainly through programming and calibration. Ensuring that all sensors are correctly calibrated at the start of each flight will reduce error in readings and programming the drone to account for as many variabilities as possible will also help to achieve consistent results.

For each of the components mentioned, the weight should be minimized while keeping in mind the budget so that the drone will be able to fly as efficiently as possible as well as have the option for the added capability of releasing support objects via an added-on release mechanism. Weight minimization can be accomplished in the execution of this project by ensuring that materials used are the lightest available for the allowed budget while maintaining durability and long-term sustainability. The programming, while complex enough to compute the necessary distances and paths, must be user friendly enough to be controlled by a non-professional with limited practice and exposure to this particular drone.

Bonus Features

Additional requirements that should only be considered once the aforementioned primary requirements are met are long battery life, waterproofness, easily visually locatable, easy storage. Long battery life can only be achieved by procuring a larger battery which will add weight to the drone or by finding an alternative battery type that can last longer. The latter is outside of the scope of the project so in order to prolong the flight time of the drone, a larger battery would have to be used if the other components can compensate for the additional weight. Waterproofness is ideal for using the drone in rainy weather or heavy mist conditions. This can be accomplished by enclosing all water sensitive parts in a water-tight sealed section and obtaining waterproof versions of certain components (for example, waterproof camera). This is not of high importance because visibility would be a major issue in moderate to heavy rain regardless of the waterproofness of the camera so it is unlikely that the drone would be operating in very wet conditions. Ease of visual locatability can be done by making simple changes to the drone such as adding an LED or

strobe light so that it can be identified in the sky or painting it in a bright colour. Making the drone easy to store can be solved three potential ways. The drone can be modular enough to be easily disassembled and reassembled, the drone can be foldable or collapsible, or the drone can be smaller. Decreasing the size of the drone is the least feasible option of the three given the currently existing challenges with battery life and equipment requirements.

3.2 Design Specifications

Looking into the components specified in the engineering requirements can determine the various quantities and qualities to use to achieve the stated problem definition. As mentioned in the previous section, a thermal camera and an RGB camera will be used to act as stereoscopic vision for the drone. Thermal camera differs from regular cameras for they observe heat rather than light. Due to this, able to consistently be used 24/7 and does not vary with weather such as fog. It has the ability to detect the smallest contrast in temperature as low as 0.01 degrees Celsius. Generally, it can go up to 50 km depending on the range of the thermal camera in use. Since a thermal camera relies on the difference in temperatures, it will be inadequate to view objects clearly and only view the outlines of the object. In addition, having limited resolution of 640x480. The RBG camera is then used for this case, it has values ranging from zero to 255 for colour. This is due to the conversion between the camera itself and device where the visual results are outputted. The resolution for a RGB camera varies depending on the monitor being in use. Using a GPS drone tracker will help allocate the position of the drone which is connected to a network of satellites around the earth providing a global position system.

3.3 Design of the Drone

The main component of this project has been the overall design and build of the physical drone prototype. As it is made of off the shelf components, there have not been a requirement for a prototype process. The best option would be to 3D print non-standard parts, since this allows complex and unique parts to be produced. The material, acrylonitrile butadiene styrene (ABS) plastic, is also lightweight, and having a readily available 3D printer allows for ease of access. The design consists of a few standardized components; any component that could accomplish the task using standardized components have been replaced with a standard part. This results in our prototype process consisting of a combination of 3D printed parts with standardized parts such as screws, dowels and springs.

Meanwhile, the failure modes and effects analysis after design phase has helped to highlight some key components that are vital to the proper and safe function of the drone. The methods suggested for failure prevention or mitigating can be seen in the later section.

Figure 3.1 showed the overall CAD view of the designed drone. The frame of the developed drone is carbon fiber. The gimbal assembly is manufactured using Additive Manufacturing (3D printing) The gimbal assembly currently consists of one servo motor that can orient the tilt axis of the camera. The developed system is shown in Figure 3.2.

Figure 3.1 CAD model of fully assembled drone.

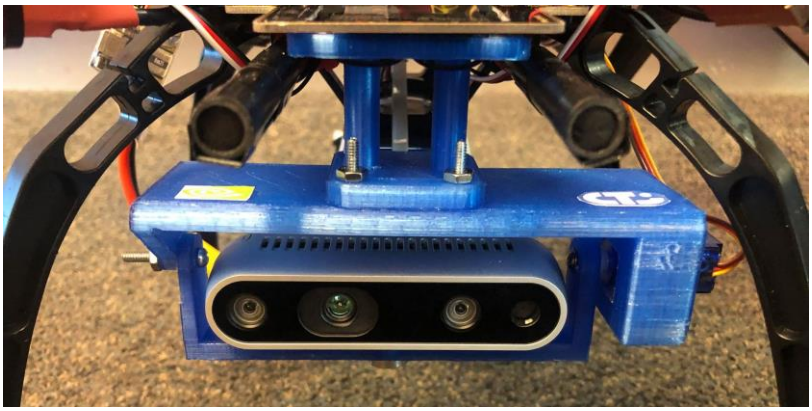


Figure 3.2 Custom gimbal.

The Intel Realsense custom gimbal designed is the exact shape of the camera and allows a tilt motion of the camera using a servo motor which can be connected to the controller. The tilt option of the gimbal itself allows for easier and more stable flight as the drone won't have to be flown in a tilted position, or even maintain a tilted position while the camera searches the area. Figure 3.2 shows how the gimbal (blue) is attached to the drone. The camera is secured with a screw that goes through the gimbal into the camera's previously existing tripod hole.

Figure 3.3 demonstrates the connection of key components of the drone. The figure also shows the wiring of the circuit. The 6-cell battery runs at 22.2 volts and runs through a DC-DC step down voltage regulator. The power that goes directly to the pixhawk is a safe, controlled voltage that allows for the operation of the flight controller (this was supplied with the flight controller upon purchase). The power coming out of the other end of the breakaway goes directly to the power bus that is integrated into the PCB of the drone frame. A DC-DC converter is connected in parallel to the circuit and is set to 12 volts to power the Jetson TX2. Four ESC's are also connected to the PCB. Power is split in parallel to four ESC's each connected to the motors. The ESC control plugs are connected the main out ports (1 through 4) on the pixhawk. The Intel Realsense camera is connected and powered from the USB 3.0 port on the Jetson TX through the Orbitty carry board. The GPS module is connected to the GPS and 12C connection on the Pixhawk. The telemetry 2 port is used to connect to the Jetson TX2 as mentioned before. The RC receiver is connected via an SBus connection to the Pixhawk to the corresponding port (this can be seen in the circuit diagram). Lastly the servo for the gimbal is connected to the Auxiliary out1 port.

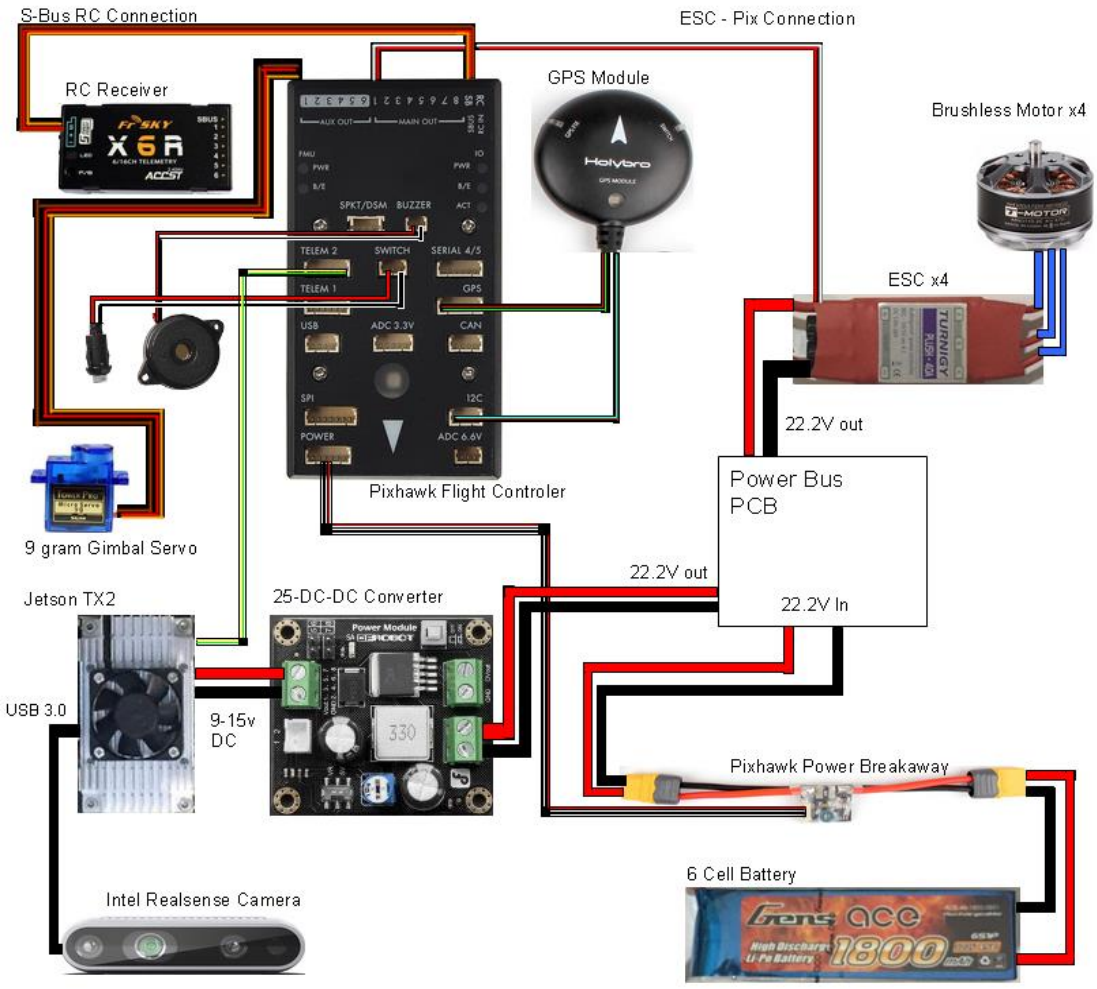
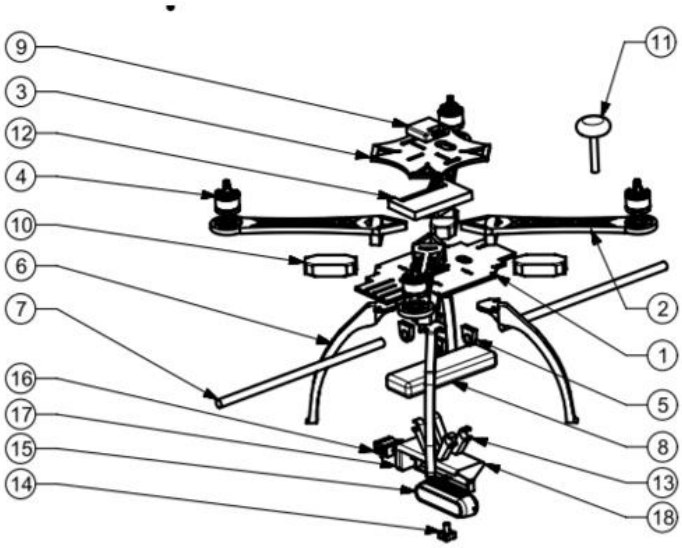


Figure 3.3 Key components and circuit diagram.

3.4 Key Components

Figure 3.4 illustrates the overall design of the drone in an exploded view. It lists the overall construction of the drone along with the detailed components listed on the right side. The key components of the drone are listed and explained in the later section.



PC NO	PART NAME	QTY
20	[M2.5X6 SOCKETHEAD SCREW]	8
19	[M2.5X8 SOCKETHEAD SCREW]	4
18	TOP PLATE	1
17	BOTTOM PLATE	1
16	SERVO MOTOR	1
15	INTEL_REALSENSE_DEPTH_CAMERA_D435_STEP	1
14	CAMERA SCREW	1
13	ROD HOLDER	2
12	JETSONTX2	1
11	GPSMODULE	1
10	40AMPESC	4
9	OPENPILOTCC3D	1
8	LIPO_3000MAH	1
7	DRONEROD	2
6	DRONE LEG	4
5	RODFASTENER	4
4	MOTORASSEMBLY	4
3	TOP_FRAME	1
2	ARM	4
1	BOTTOM_FRAME	1

Figure 3.4 Exploded view of full drone assembly.

Drone Frame

The drone frame is used to keep all components in place and secured. Different types of frame materials and configurations can be used. The proposed drone frame in this project is to aim to a small form factor shown in Figure 3.4.

Remote Control

Based on the Figure 3.5, to navigate, the up, left, down, right controls are used to move the cursor on the small screen should own. The menu key can be used to pick a selection or for editing a command. The selection option allows different functions to operate. The exit key is used to exit any option currently chosen on the screen. This will allow the cursor to go back to the top of the screen. Center switch is used to turn on the remote. The right stick allows to move drone forward or backwards when motioning the right stick up or down. Moving the right stick left or right allows the drone to move either left or right. The left stick allows to drone to be raised or lowered when moving the left stick up or down.

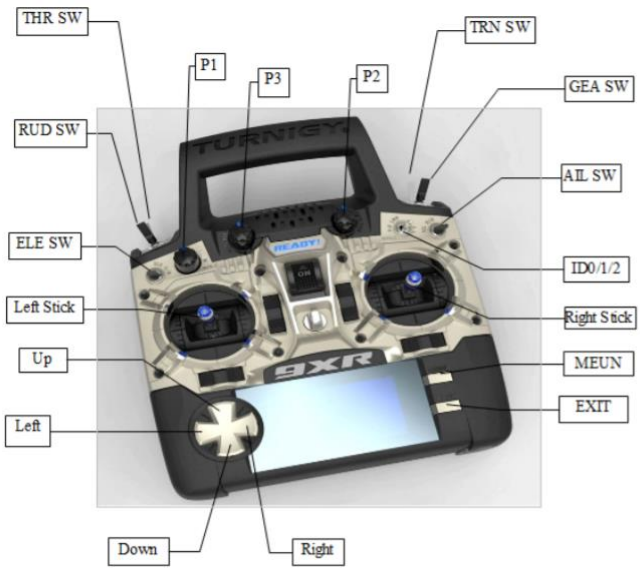


Figure 3.5 Remote controller for drone.

Intel RealSense Camera

The intel RealSense D435 equipped on the drone is a depth camera, powered by a USB and consists of a pair of depth sensors, Red Green Blue (RGB) sensor, and an infrared projector. The depth camera has advanced stereo depth algorithm for accurate depth perception and long range. The Intel RealSense Camera is held and controlled by an in-house designed gimbal that contains four components, a rod holder, servo motor, a bottom plate, and a camera screw. The servo motor is used to help the depth camera alternate its view. The rod holder, bottom plate, and the camera screw keep the RealSense Camera stabilized.

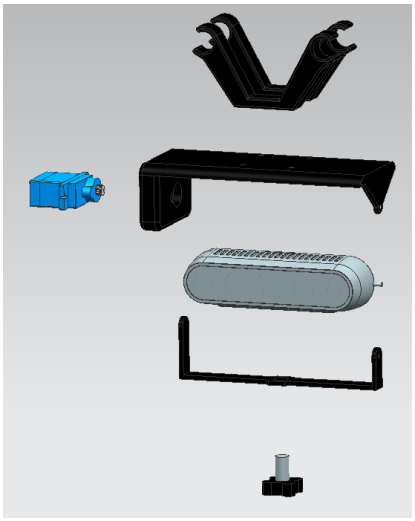
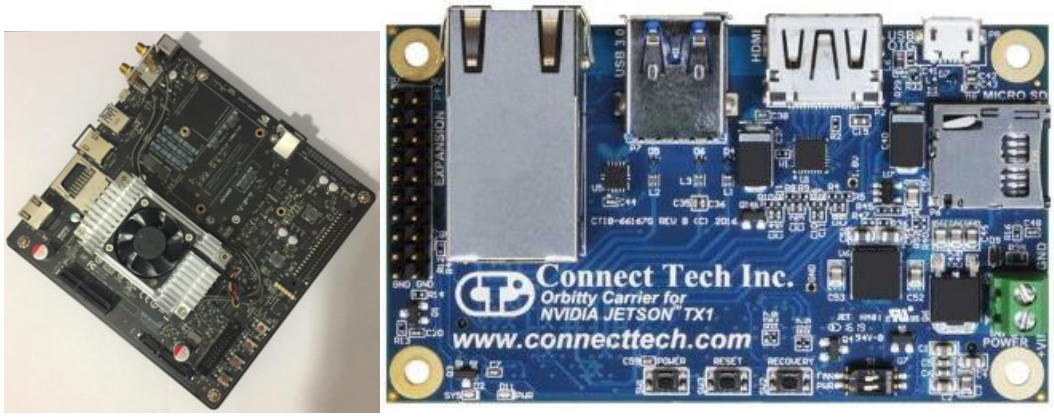


Figure 3.6 Exploded view of single-axis gimbal.

Jetson Orbitty Carrier Board

The NVIDIA Jetson TX2 with its development board is a powerful single board computer that is able to run full Linux system and deep learning frameworks because of its powerful GPU unite. However, the size of the carry board makes it impossible to directly install on drone. Figure 3.7(a) showed the dimension of the Jetson TX2 with its original carry board. In order to address the issue and implement the powerful SBC to the developed drone for advanced vision and deep learning frameworks for object identification and tracking, a replace board from Connect Tech Inc. is adopted in this project. The dimension of the new carry board is shown in Figure 3.7(b) which is more suitable in UAV platforms and applications.



(a) (b)

Figure 3.7 a). Original Jetson TX2 development board; b). Obitty carry board.

Transmitter/Receiver Module

The transmitter module is used to allow for the drone to communicate to the flight controller for manual control/override. As the stock Turnigy X9 is not compatible with the FRsky receivers, this is needed to properly communicate with the receiver. The FRsky receiver is solely used for communications between the transmitter/operator and the drone. The receiver will not connect to any of the ESC units connected to the motors as its only function is to allow for communications and control from the transmitter used by the operator. The FRsky module is used as it is compatible with the pixhawk and uses the SBUS style of communication.

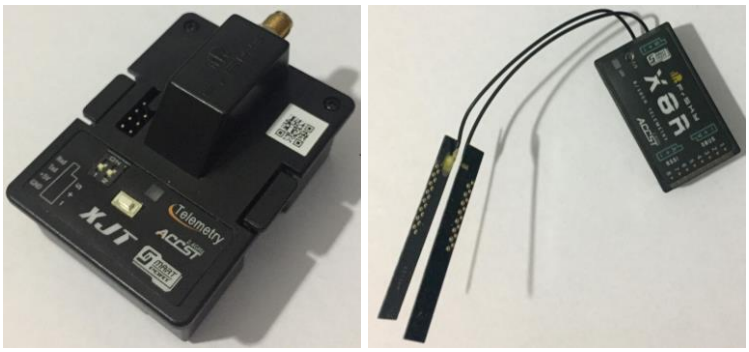


Figure 3.8 FRsky transmitter module and receiver.

GPS Module

Figure 3.9 shows the GUP module used in conjunction with the Pixhawk Flight controller.

Figure 3.9 GPS module.

Pixhawk Flight Controller

Figure 3.10 depicts the pixhawk flight controller. This is the main communication between the onboard computer (jetson), the off board computer running ground control software, and the manual control transmitter. The pixhawk is responsible for monitoring all the information about the drone in flight and relay that to the onboard and offboard computer. This device tracks pitch in multiple axis, as well as position through the GPS module (drone direction via compass and location via GPS). This flight controller uses the onboard sensors to keep the drone stable during flight and to allow mavros on the onboard computer to keep the drone operating within safe parameters. The flight controller can also be used to aid in manual operation by auto leveling the drone as well as have an auto land feature and a fly home feature (among various other capabilities). The pixhawk is the "secondary brain" of the drone and handles all of the flight characteristics.

Figure 3.10 Pixhawk flight controller.

Motors and ESC Modules

The motors and ESC units used for the project can be found in Figure 3.8 below. The ESC units used are the Turnigy plush 40A 6-cell units. They are chosen so that there would be a sufficient amount of buffer for the amperage requirements of the motors. They are not expected to draw 40A but it leaves an acceptable buffer range to be on the safe side. The motors used in the project are Quantum 475KV motors. They are selected in conjunction with the propellers as graphs showed the amount of thrust with respect to the throttle setting.

Figure 3.11 ESC (Left) and motors (Right).

Battery

Figure 3.12 depicts the battery used for the project. The 6-cell battery is chosen for the project because it has a decent capacity for the purposes of this project as well as being able to supply sufficient power to the entire system. It has a capacity of 1800 mAh and supplies a voltage of 22.2 volts.

Figure 3.12 LiPo battery.

Propellers

Figure 3.13 depicts the propellers selected for the project. They are made of carbon fibre and have a span of 11 inches with a pitch of 3.7. This was selected in conjunction with the motors as the span of the propellers as well as the pitch determines the amount of thrust produced by the motors at a certain speed. At 50% throttle, these particular propellers paired with the quantum motors produced more than enough thrust to lift the drone and its payload. On the drone there are 2 clockwise propellers and 2 counter clockwise propellers. This is done because of how the motors are configured in a quadcopter. It requires a clockwise and counter clockwise motor on each side of the drone, this allows it to behave with the stable flight characteristics quadcopters are known for (as well as pitch and yaw controls).

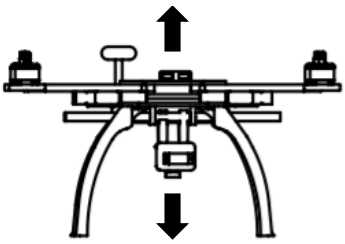
Figure 3.13 Carbon propellers.

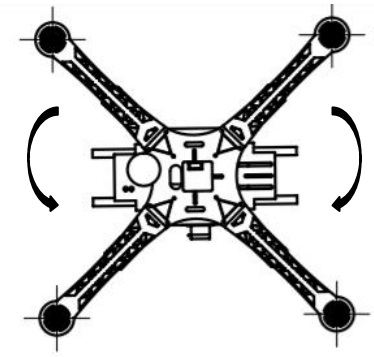
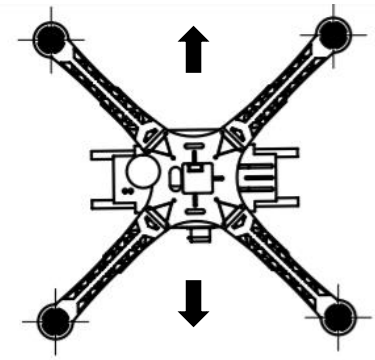
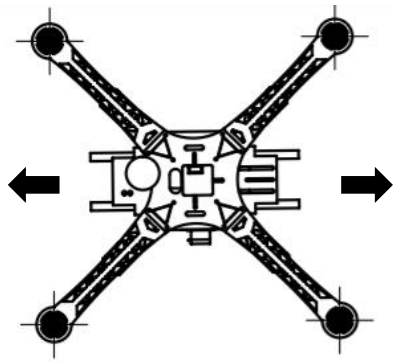
3.5 Functionality Design and Development

Manual Control

Flight control is established through the Turnigy T9 controller. The left and right joysticks control the drone flight as described in the table below.

Table 3.1 Left and right joystick control.

Altitude Control	
	Slightly move the left joystick up to raise the drone (altitude). Slightly move the left joystick down to lower the drone (altitude). When the joystick is in neutral position, the drone will maintain current throttle setting (may not produce a hover).
Yaw Control	

	Slightly move the left joystick to the right for clockwise rotation of the drone. Slightly move the left joystick to the left for counter-clockwise rotation of the drone. When the joystick is in neutral position, the drone will maintain current yaw angle.
Lateral Control	
	Move the right joystick to the right to fly the drone to the right. Move the right joystick to the left to fly the drone to the left. When the joystick is in neutral position, the drone will maintain current pitch unless auto-hover is enabled.
Forward/Reverse Control	
	Move the right joystick up to fly the drone forward. Move the right joystick down to fly the drone backward. When the joystick is in neutral position, the drone will maintain current pitch unless auto-hover is enabled.

To adjust the performance of the drone, various control parameters were adjusted to improve the flight characteristics of the drone. When set to stabilized mode, the onboard drone controller attempts to level the drone when no input is given to the right control stick. To do this, the maximum roll and pitch rate were adjusted as well as the P variables for these parameters in the proportional, integral, derivative (PID) controller. The PID controller is used to adjust the position of the drone such that it achieves the desired position. The controller determines what the desired position is, then compares it to the current position of the drone. The difference between the current position of the drone and the desired position is the error. This is then used in the PID to correct the drone to the desired position. The larger the error is, the faster the correction of the drone orientation.

Control

Due to quadcopter's under-actuated properties, keeping in stability or staying at a desired attitude becomes very challenging. Control of a quadcopter is fundamentally difficult, making it an attractive topic. Over several decades, researchers have investigated different kinds of control systems to achieve quadcopter hovering. L. M. Argentim et al. utilized Linear Quadcopter Regulator (LQR) controller to maintain an indoor micro quadcopter balance [65]. In [66] the parameters of PID controller were optimized and the simulations were demonstrated in order to compare the performance between optimized PID and Back-step controller. A robust cascade PID algorithm was proposed based on linear model of quadcopter in [67], which improved the stability by lowering the effect of external disturbance. In order to study the mathematic modelling of the quadcopters, Newton-Euler Equations were introduced for total forces and torques acting on the system.

Based on the model, many control algorithms can be developed to achieve better performance.

In order to maintain a balance and achieve hovering for a quadcopter, a classic PID controller is commonly utilized. Advantages of the classic PID controller are:

- simple structure
- easy implementation
- well developed

The general form of a classic PID controller is shown below:

$$e(t) = x_d(t) - x(t)$$

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

where K_P, K_I and K_D are the proportional gain, integral gain and derivative gain, respectively, u(t) is the control input, e(t) is the difference between the desired state $x_d(t)$ and the present state $x(t)$.

The pitch rate and roll rate values were changed from 6.5 to 6, and then to 5.5. The P values for both the roll and pitch rates were changed to 0.45. These two changes made a very significant improvement to the response, and overall flight characteristics of the drone. The drone was quick to correct to a level position with very minimal amounts of overshoot. This resulted in much more stable and predictable flight characteristics.

Camera and Vision

The camera testing results are seen and indicated in the list below by checkmarks for passed categories and blank boxes for failed categories.

Utilizing a ROS package called find_object_2d, as viewed in Figure 3.14, the camera is capable of detecting the object selected by the user. Using focal length equations, the results of the test was fairly accurate when comparing the calculated value to the measured value with an error of about 1 cm. This is mainly in the case of when the camera and object are decently parallel to each other. If the camera or object is angled to a large degree, the error increases proportionally.

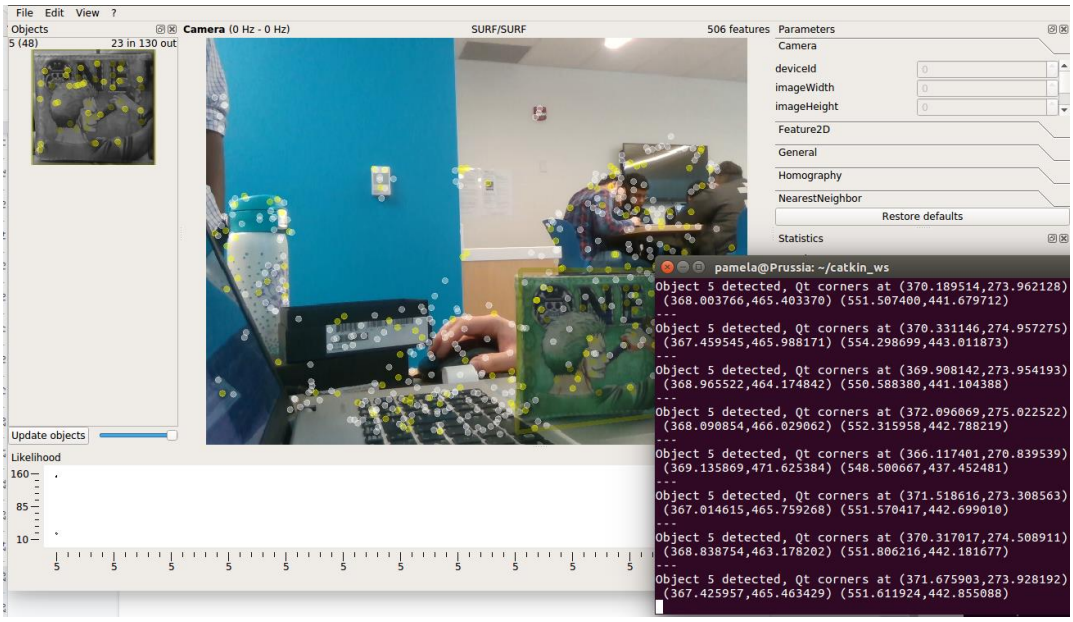


Figure 3.14 On-board camera testing.

Localization and Visualization

The pixhawk flight controller is selected for the project to be the onboard all in one flight controller. This module works in conjunction with the QGroundcontrol software for aspects related to flight controls. All of the system elements such as the ESC's, servos, onboard computer and GPS were connected to the pixhawk. The QGroundcontrol software has many tools available for the operator that allows for the adjustment of the PID controllers for stability as well as flight planning. It is capable of much more customization,

the GPS monitoring, flight path planning, and PID stabilization are considered in the project.

All flight data is recorded onboard the drone and can be accessed when grounded. This gives an astounding amount of statistics and is accessible through the QGroundcontrol software and can be downloaded to a computer. Figure 3.15 below shows the flight path planning capabilities of the QGroundcontrol software. The software has many customizations available to control the flightpath of the drone. Two of the main settings are the point to point navigation and the survey navigation types. The point to point navigation can be customized changing the altitude settings, orientation of the drone, speed of the drone point to point, and specific maneuvers (such as takeoff and landing) at each individual point. The survey navigation can be altered to create more paths, changing the surveying angle/path, creating more passes and altitude settings. One of the major safety features integrated into QGroundcontrol is the flight fencing and no fly zones. No fly zones can be designated and the drone will automatically avoid those locations. This creates a layer of safety when the drone is being operated in autonomous mode. Another additional layer of safety is added when a flight fence is introduced (denoted by the orange outer line in Figure 3.15). The drone will detect with GPS when it is flying outside of the fence and will direct the drone back to the non-restricted zone. The drone will also react when connection is lost. If the drone is experiencing undesirable conditions or loses contact with the operator the drone will return to the takeoff point at a safe altitude and will land the drone.

Figure 3.15 QGroundcontrol mapping.

Prior to all flights, the drone should be calibrated using the software. This is relatively simple to do as all that needs to be done is plug the drone into the ground control station laptop. The prompts for calibrating the accelerometers, gyroscope, and level horizon are simple and easy to follow. When complete, power cycle the drone to complete calibration.

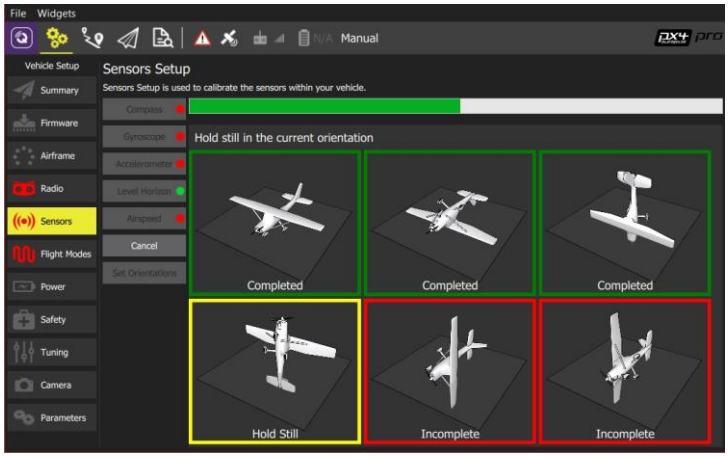
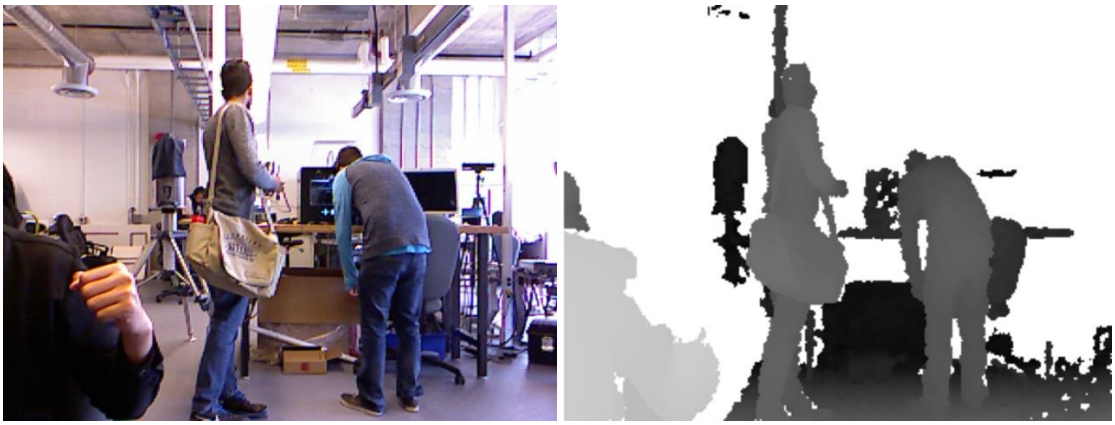


Figure 3.16 Drone calibration.

Chapter 4. Vision-based Interactive Features

In this section, a stereo camera is proposed to be utilized to acquire depth information for segmentation of the hand from its background and motion estimation. The proposed usage of depth data in segmentation will decrease the distortion and noise from the environment. However, there are still some challenging issues. For example, the depth information will lose information/features of objects inside the camera scene, especially in the hand case because the hand is a small part of the whole image. In order to have a better understanding of the hand gesture and future motion estimation, the original RGB color image and depth image (shown in Figure 2) will be considered at the same time.

There are two objectives in the proposed project: 1) robust and real-time detection for static hand gestures and 2) motion estimation of the fingertips (velocities) by optical flow, which analyzes the difference between adjacent frames. The overview of the system will be presented. The algorithm of hand recognition is described in the following section. A detailed discussion of the calibration and calculation of motion estimation of fingertips will be shown in Section 3. Section 4 will show the evaluation results of the proposed method. The conclusion is shown in Section 5.



(a) (b)

Figure 4.1 (a) RGB color image; (b) depth image.

4.1 Framework and Overview of the System

The framework of the proposed system architecture is demonstrated in Figure 4.2. It shows the overall procedure of hand detection and motion estimation of the fingers, which includes segmentation using depth images, filtering, contour identification, feature detection and tracking, calibration of the camera and motion estimation (specifically estimation of velocities). The system implementation is developed under the framework of OpenCV (Open Source Computer Vision Library) and Kinect development framework for using the Kinect sensor for image and depth information acquisition.

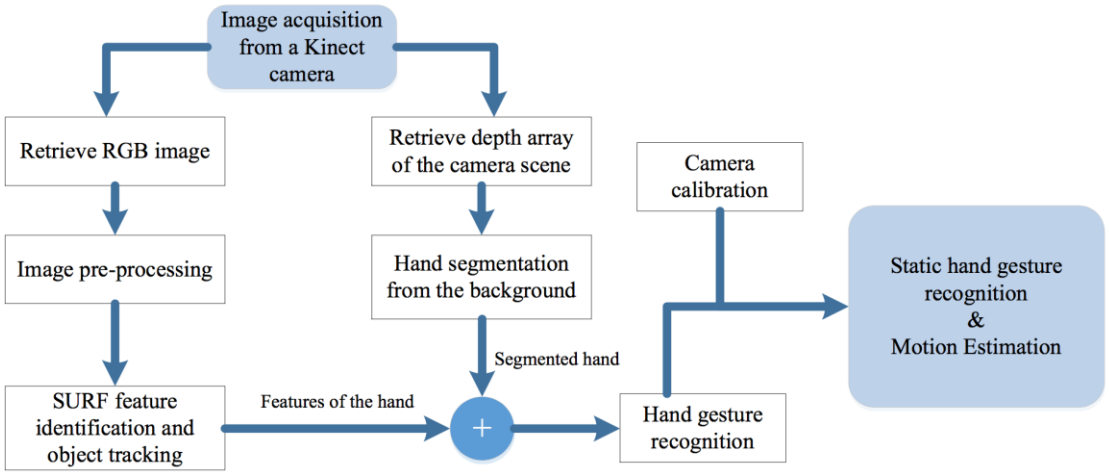
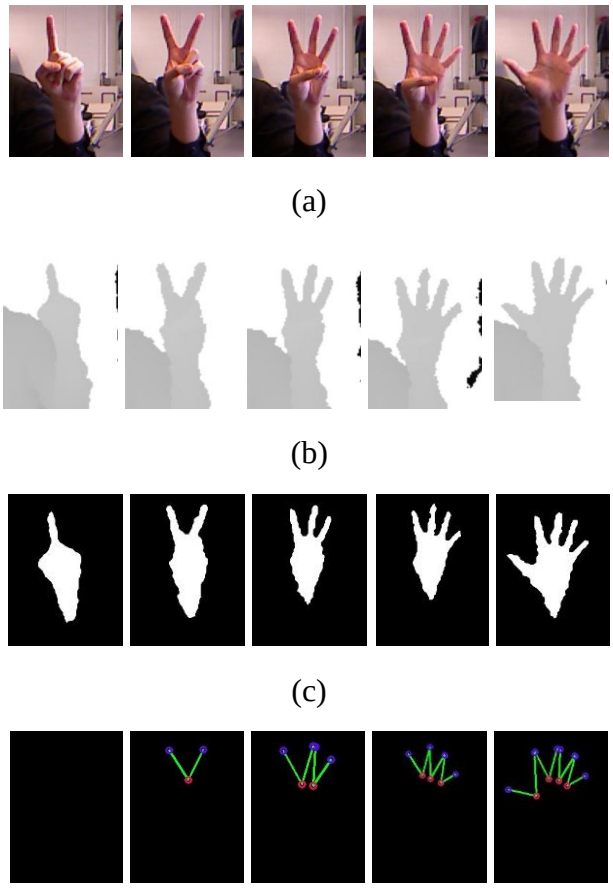


Figure 4.2 Framework of hand gesture understanding and motion estimation.

Kinect camera has an infrared projector and receiver for the purpose of finding depth data. Since the resolution of the depth image is low and the original image is affected by noise from the environment, pre-processing is needed to reduce the undesirable interference and non-linear noise. Therefore, a group of filters are applied to eliminate the noise for better computation in the next stage. Meanwhile, a threshold is set to segment the hand from the background depending on the depth image post-processed by the previous step. Next, the binary image of hand contour is analyzed and compared by the contour of the hand and its external polygon, which can be utilized to decide the number of raised fingers. Reasonable number of fingers that are detected will be utilized for hand gesture recognition. Third, the 3D representation of fingertip localizations can be determined by

using the calibrated camera information. Finally, the motion estimation of fingertips can be calculated by comparing the difference from the frames, which are shown in Figure 4.3.



(a)

(b)

(c)

(d)

Figure 4.3 Demonstration of static hand gesture and identification results: (a) original color images; (b) depth image of the original color images. (c) segmented hand images (d) corresponding poses of the fingers.

Figure 4.3(a) shows the original RGB images from the Kinect camera; Figure 4.3(b) demonstrates depth data of the camera scenes. These two types of images, which are acquired from the Kinect sensors, are fed into the unit of pre- processing and threshold

operations. The post-processed images are shown in Figure 4.3(c), which illustrate the shape of the detected hand. Finally, the contour of the hand is drawn, and the representation of the current finger gestures are represented in Figure 4.3(d).

4.2 Pre-processing

Pre-processing is essential for almost every vision system in order to provide a better and relatively noise-free image. As mentioned above, the resolution of the depth image provide by the Kinect sensor is only 320×240, which is too low for identifying hand in the image where the hand is a small portion. The linear interpolation and Gaussian filter are utilized to increase the resolution of the image to 640×480 and avoid the sudden value change between adjacent pixels, respectively.

For segmentation, a group of threshold values are set to eliminate the background depending on the results from the SURF hand tracking algorithm. A region of interest (ROI) will be preserved; and the background will be eliminated from the original image. Once the hand has been identified and segmented from the background, the ROI is converted to a binary image as shown in Figure 4.3(c). The area of the hand in the figure has approximate of 50×50 pixels. Some distortion can be found in the image due to the low resolution of Kinect sensor.

4.3 Hand Gesture Recognition

Having acquired the binary image of the hand, the Suzuki's method for border detection [16] is utilized to identify the contour of the detected hand. Another threshold is set at this stage to eliminate other small contours caused by noise. Therefore, the largest contour will be preserved; and a clean hand is segmented from the image for further

analysis. Next, the convex hull is detected using Sklansky's algorithm from the 2D point array [17]; and convexity defect is calculated from the difference of contours and its external polygon as shown in Figure 4.4.

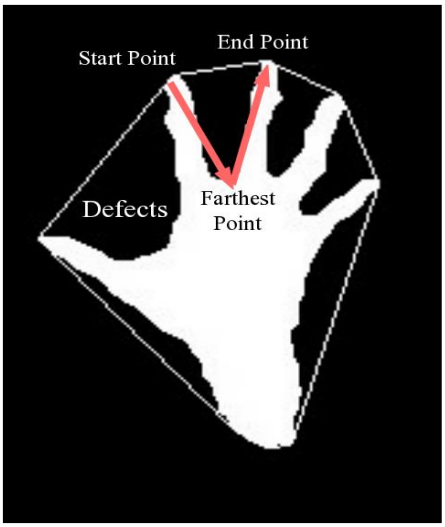


Figure 4.4 External polygon and defects.

All the intervals between fingers are found in Figure 4.4. However, there are still some undesirable intervals (e.g. the interval between the thumb and finesse, little finger and finesse and probably part of the arm.). In this circumstance, the angle of the defect is calculated by an equation below:

$$\theta = \cos^{-1}[(b^2 + c^2 - a^2)/(2 \times b \times c)\,] \tag{4.1}$$

where b is the distance between the start and farthest points; c is the distance between the end and farthest points; and a is the distance between the start and end points. Then, a threshold is set to further identify fingers based on the factor that the angles between two

fingers cannot be greater than 90°(θ > 90°). Therefore, the fingers can be precisely determined according to the processes described above. The detected result is shown in Figure 4.3(d) where the red points represent the farthest points of defects, while the start and end points are labeled blue. However, there are no angles if there is only one finger or no fingers in the image. In order to distinguish it from a fist, an addition procedure is taken which will be explained in the section below.

4.4 Finger Detection

In order to solve the problem of one finger/no finger detection, an algorithm is introduced to recognize the palm and fingers. The basic procedure for fingertip detection is summarized in Algorithm 1.

Algorithm 1: Finger detection

1. Let r be the radius of palm. Initially, set r =0.
2. Calculate the spatial moment and the mass center of the contour.
3. When raised finger is more than one, set r according to the farthest distance between mass center and defects edge.
4. If there is no eligible defect, use the same r in prior frames.
5. Hollow the palm part based on the mass center and radius.
6. Draw the contour of each finger.

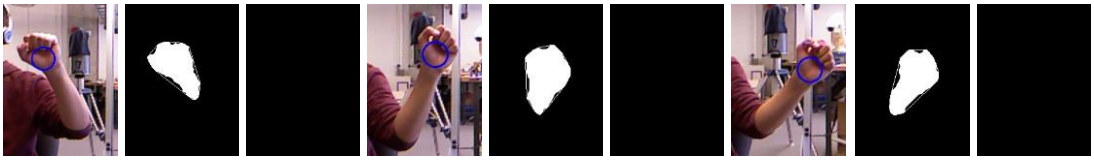
Spatial moments of the hand contour are used to calculate the mass center which is employed to locate the center of the palm. As proposed by Green [18], the spatial moments (m_{ji}) are calculated by this equation:

$$m_{ji} = \sum_{x,y}\left(array(x,y) \cdot x^{i} \cdot y^{j}\right) \tag{4.2}$$

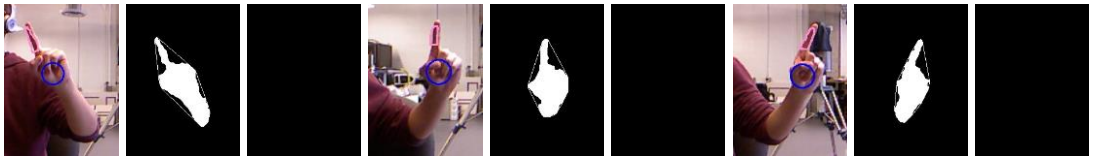
And the mass center (x', y') is computed as:

$$x' = \frac{m_{10}}{m_{00}}, y' = x' = \frac{m_{01}}{m_{00}} \tag{4.3}$$

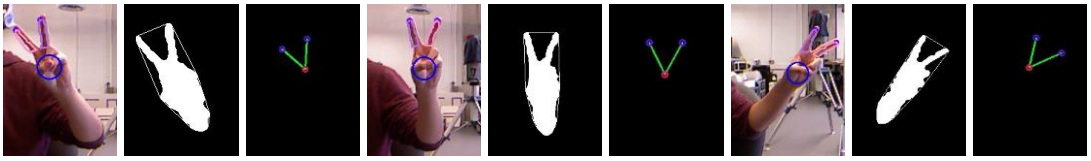
In the previous section, the farthest points are located and marked in red in Figure 4.3(d). Based on that data, the distances between each farthest point and mass center (x', y') are calculated subsequently, and the largest distance (D_0) is recorded. Therefore, we can utilize the lengths to locate the palm and fingertips and draw an approximation of the palm.



(a)



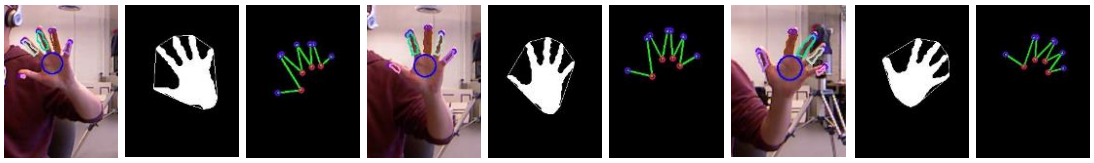
(b)



(c)

(d)

(e)



(f)

Figure 4.5 The test images and results. (a) no finger. (b) 1 finger. (c) 2 fingers. (d) 3 fingers. (e) 4 fingers. (f) 5 fingers.

The circumstance has been discussed when there is no or just one raised finger. For distinguishing one raised finger and a fist, the biggest radius is gained from the former

image when there are two or more fingers. If there is remaining hand part in the image after eliminating the palm part, it can be interpreted as one finger. On the contrary, if nothing is left, it is a fist, as shown in Figure 4.4 and Figure 4.5.

4.5 Data Acquisition of Each Fingertip

In order to acquire 3D data, a 2D coordinate system is defined on the image plane. After localizing the fingertips from the above steps, the depth data is also obtained from the original depth image array. A series of image data including the RGB images and depth data from subsequent frames are retrieved and stored for further processing.

4.6 Calibration of Camera and Model Construction for Speed Estimation

Kinect has a fixed-focus camera, and its focal length can be obtained by the camera calibration toolbox. The model of camera can be approximately regarded as central perspective (also called pinhole perspective) projection model as shown in Figure 4.6 [21].

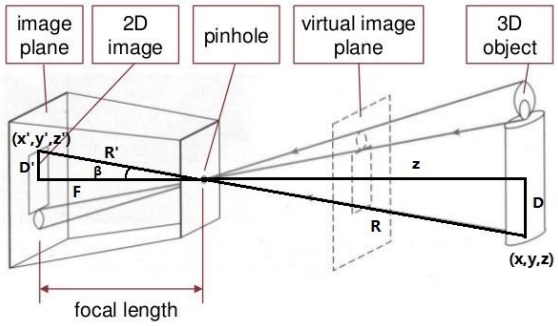


Figure 4.6 The pinhole image model.

The basic principle is based on similar triangles and the image on the image plane has similar proportion with the object. The position of each given points on the image plane can be found below: firstly, a rectangular coordinate system is set and the origin of the

coordinate frame is at the pinhole. The depth (Z) is shown on Figure 4.6. After the camera calibration, the focal length (f) and other intrinsic parameters of the camera are all known. The angle of a certain pixel can be calculated based on the angle range. Therefore, R' and D' in Figure 4.6 can be computed as:

$$R' = \frac{f}{\cos(\beta)}; D' = f \cdot \tan(\beta) \tag{4.4}$$

The corresponding point (x, y) in the workspace can be found in the image plane as (x', y'). The depth is retrieved using the depth information and is marked as R. In this case, $z = R \times \cos(\beta)$, and (x, y) is obtained using the similarity relation between the coordinates in the workspace and image plane as:

$$\frac{R'}{R} = \frac{x'}{x} = \frac{y'}{y} \tag{4.5}$$

Therefore, the position of a point (x, y, z) in the 3D workspace can be determined. Since the position of each fingertip on the image plane has already been acquired from the previous section, and the motion can also be acquired from adjacent frames, the position of a fingertip can be calculated by using the equation below:

$$Distance = \sqrt{x^2 + y^2 + z^2} \tag{4.6}$$

Therefore, the motion estimation can be acquired by taking into account of the position changes of the fingertips and the time intervals of the frames.

4.7 Evaluation

The performance of the proposed hand gesture recognition is tested on 6 gestures from 0 to 5 and each of them contains 3 orientations. Figure 4.7 shows the original data

and results. All these setting conditions of the fingers can be successfully identified, which also shows from the robustness of the system and from the experimental results in the videos. Figure 4.7 illustrates the situation that two fingers are gradually moving close to each other. From the result, the proposed system is able to track the fingers and estimate the motion of them (e.g., the interval between two fingers is extremely small, however, it is still working and is recognized as two fingers.).

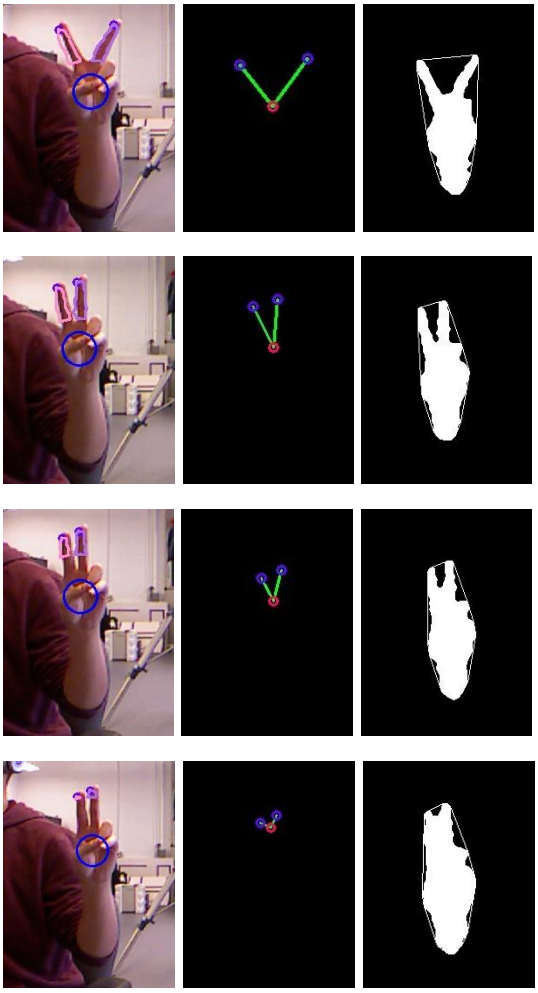


Figure 4.7 Finger interval test and results.

For the motion estimation, the estimated velocities are shown on the fingertip of the colorful image. Figure 4.8 demonstrates the results including hand tracking and motion estimation when a hand is moving in the front of the Kinect camera. In the first picture, the arm is resting on a chair, and there is no speed (Speed: 0.00 cm/s). The rest of two pictures show the estimated speed on the original images. Furthermore, the average cost of the processing time is 0.1331 second/frame, which can achieve real-time processing and extend to other applications in robotics.

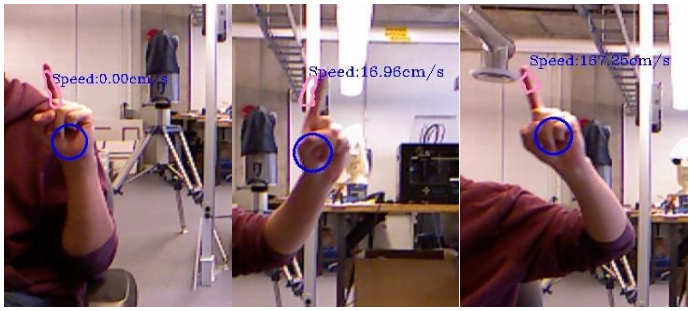
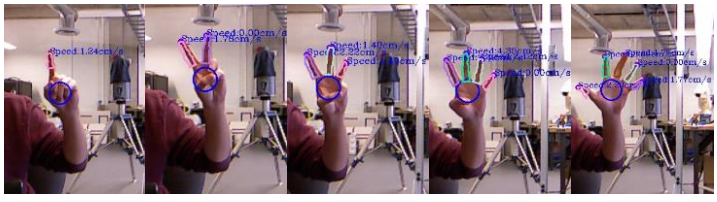
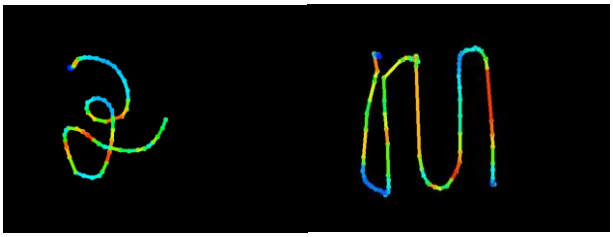


Figure 4.8 Experiment of motion estimation.

The proposed method can be employed in robotic visual servoing where object identification and tracking are two key points for a successful system (Figure 4.9(a)). Meanwhile, in the human-robot interaction, it can be served as a robust static and dynamic hand gesture recognition and further extended to recognize more gestures like ASL (American Sign Language).



(a)



(b)

Figure 4.9 (a) Real-time motion estimation. (b) understanding the motion of the fingers.

Chapter 5. Deep Learning for Object Identification and Tracking

This chapter presents the development of deep learning application which is running on the developed drone on-board system to address vision-based object identification and tracking functions. The need for such function is crucial which will significantly broaden the applications of autonomous UAVs. An good example and application is to utilize autonomous drone to carry out the surveillance of the parking lots.

5.1 System Overview

Figure 5.1 shows the overall flowchart of how proposed deep-learning based license plate recognition system (LPRS) works. The system contains three modular (license plate identification, character segmentation and character recognition). Although LPRS has been widely investigated by many researchers, there are still existing challenges needed to be overcome such as poor file resolution, bad lightening condition, different language characters, etc. In order to classify and localize the LP (license plate), an object detection model called MobilenetV2-SSD (single shot multi-box detector) is utilized in this project.

After cropping the detected LP, tesseract OCR will used to operate segmentation and recognition process.

The following sections will demonstrate SSD, Mobile-netV2 and Tesseract OCR in detail.

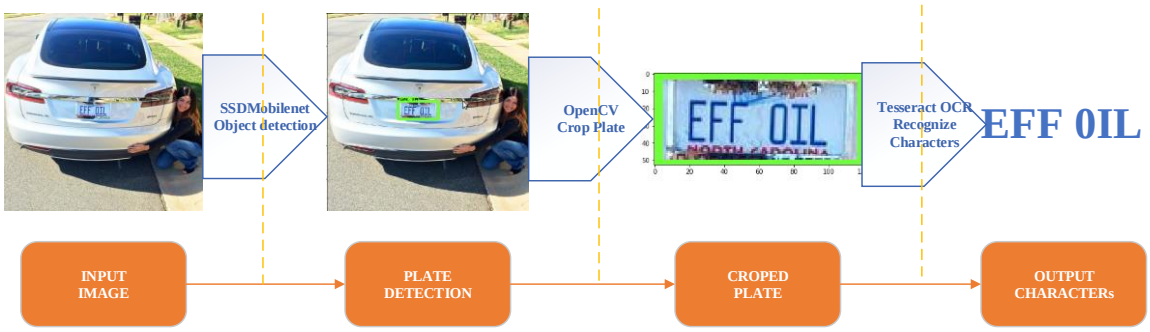


Figure 5.1 License plate recognition system pipeline.

5.2 Architecture of SSD

SSD is a method for localizing and classifying multiple objects in images using a single forward pass neural network. Figure 5.2 illustrates the architecture of a typical SSD. As it is showed in Figure 5.2, VGG-16, which performs great on high quality image classification, is popular structure in transfer learning. Therefore, the current work in this project adopts the original SSD framework as the base network. By replacing the original VGG fully connected layers, a set of auxiliary convolutional layers (from conv6 onwards) were added to extract features at multiple scales and progressively decrease the size of the input to each subsequent layer as shown in the Figure 5.2.

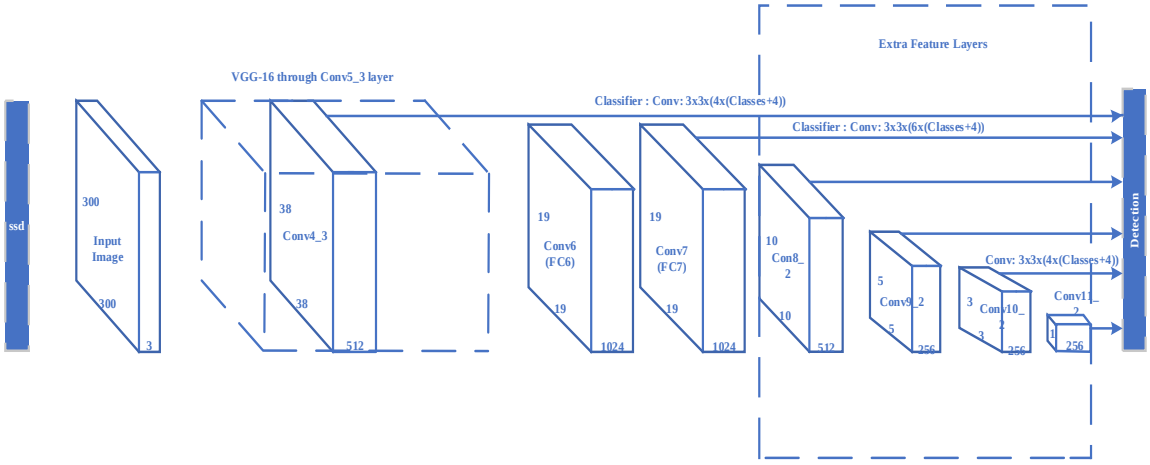


Figure 5.2 Architecture of SSD [59].

5.3 MobileNetV2 for Image Classification

MobileNetV2 [68] is the second version of MobileNet models family, which are small size, low-latency, low-power for the implantation for resource constraints use cases in many application areas. In this project, SSD-MobilenetV2 model will be realized and implanted in TensorFlow as the LP detection model which will be introduced in detail in next section.

5.4 SSD-MobilenetV2 model on TensorFlow setup

5.4.1 Annotating images

First, a customized dataset of labeled license plate images is prepared by using a tool named labelImg, which provides a user-friendly GUI and saves label files (.xml) in the popular Pascal VOC format. Figure 5.3 shows how a vehicle image with license plate is manually labeled in the labelImage system.

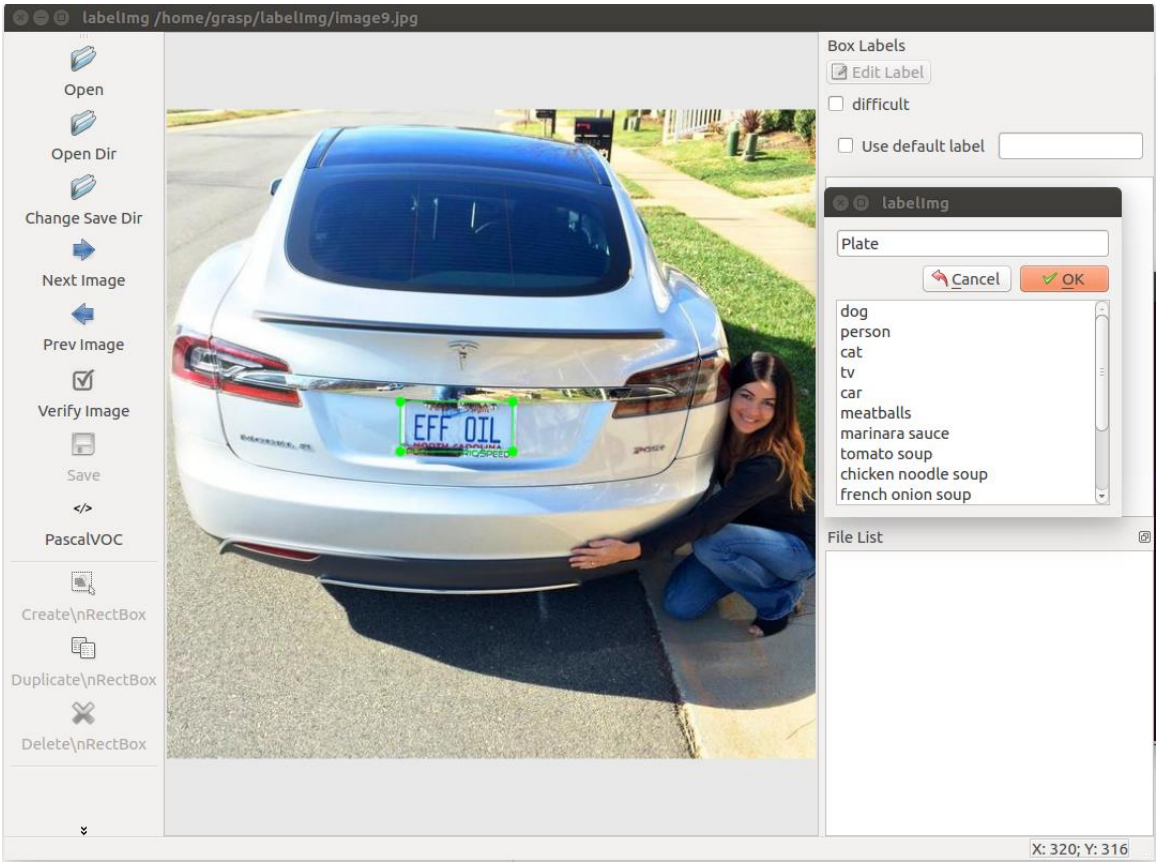


Figure 5.3 LabelImg license plate.

Having labeled all license plate images, a set of *.xml files, one for each image, will be generated and stored. For each of the file in the dataset, a unified *.csv file is required by converting from individual *.xml files. All the .csv file finally need to be converted to .record file which is required and accepted by the TensorFlow for future training and implementation.

5.4.2 Creating Label Map

TensorFlow requires a label map, which utilizes an integer value to represent each of the labeled image in the original picture. Therefore, after mapping, each labeled image has

a corresponding integer number as its representation in training and implementation in TensorFlow. As it is shown in Figure 5.4, the labeled image named “plate” has an integer ID which is 1 in this case during the training and implantation.

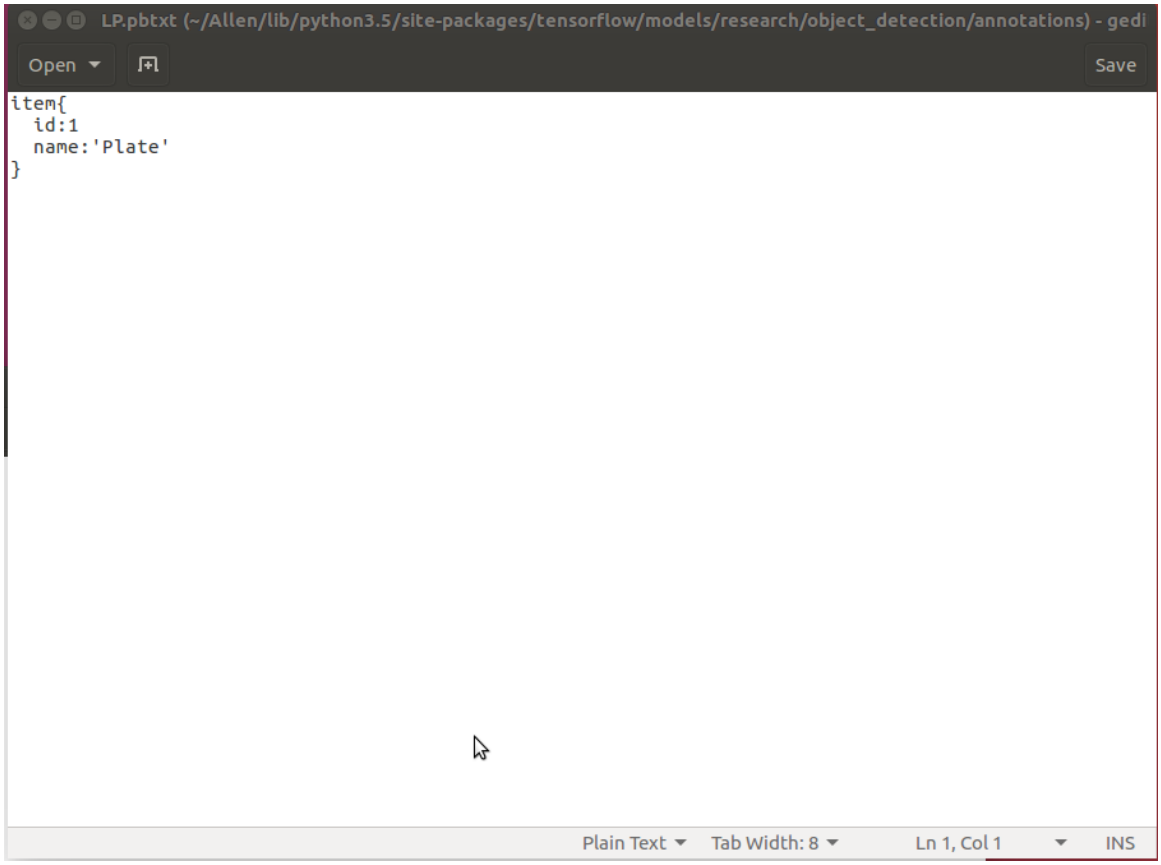


Figure 5.4 Label map.

5.4.3 Training model

The straining stage starts with a pretrained SSDMobileNetV2 model with random weights, which provides general structure of the network. The labeled dataset prepared in the previous section is utilized in training the network. The model states is stored in the TensorFlow during the training as the checkpoints (.ckpt files), which is a TensorFlow data structure. Figure 5.5 shows the model states during the training process after initiating of

the training. As it is shown in the windows, it indicates the number of completed training steps and the loss at the current network training stage. In this application, the model was trained using 9000 steps and the final loss is around 1.

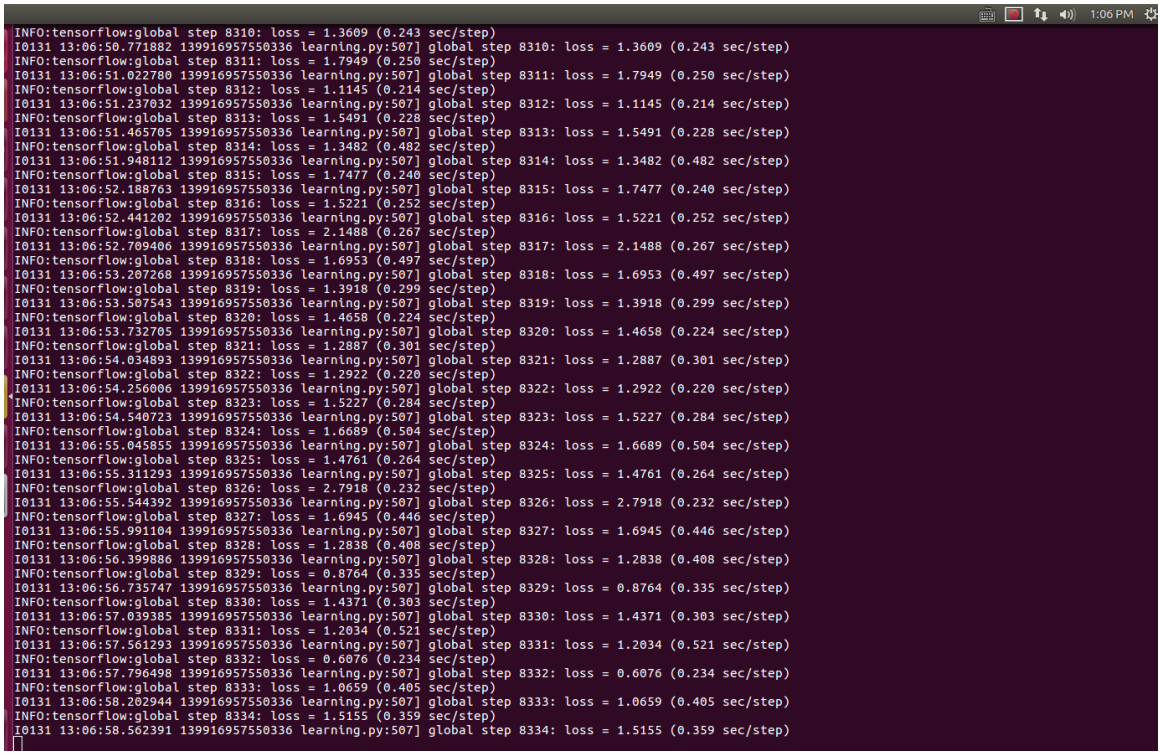


Figure 5.5 Training process in TensorFlow.

5.4.4 TensorBoard

Tensorboard is a visualization tool that provide graphical information regarding the current status of the network. It is a user-friendly feature tool which can provide continuous data for monitoring and visualizing the training procedure and its parameters such as a number of different training/detection performance metrics, Figure 5.6 shows the loss of the model and other performance measure provided by the Tensorboard.

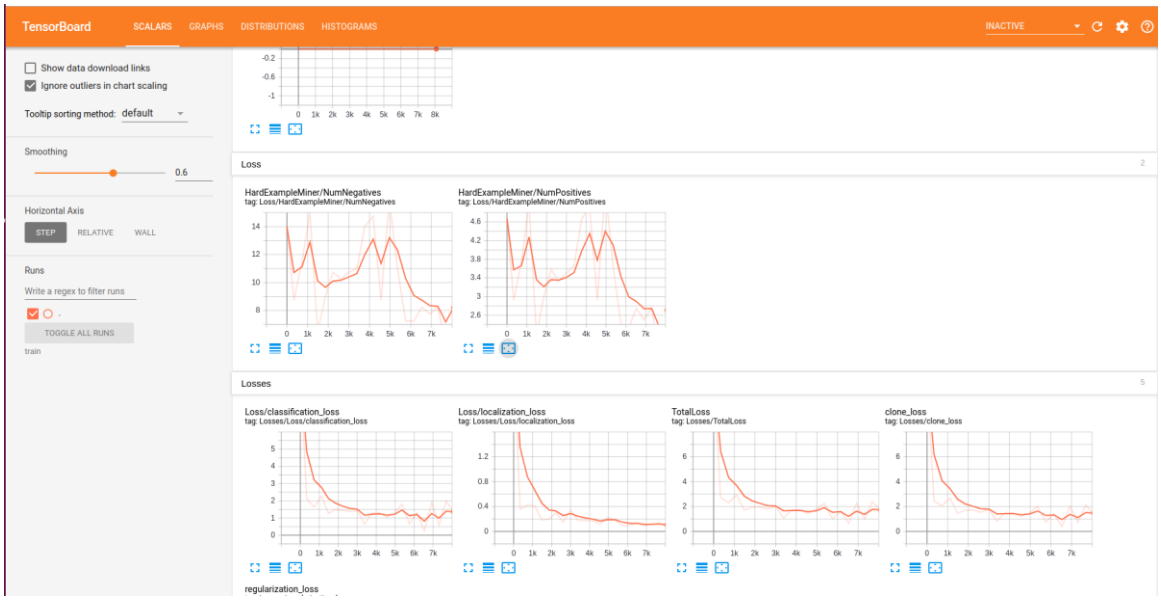


Figure 5.6 Information on Tensorboard.

5.4.5 Model Export

Once the training process is completed and satisfactory, the train network will be extracted and utilized in the application of the project: license plate detection.

5.5 License plate detection

After model training completion and exportation, the model is ready to implement the functions of localization and detection of license plate from an image. Figure 5.7 demonstrates the license plate detection result. The license plate is identified and localized by a green bounding box and classified ‘plate’ above the bounding box.

Figure 5.7 license plate detected by SSDMobileNetV2 model.

5.6 License plate character recognition

In this project, a technology named OCR (Optical Character Recognition) will be utilized to read and identify characters and numbers on the registration plates of vehicles.

5.6.1 Image preprocessing

Several preprocesses are needed before feeding the labeled images (image inside the green bounding box) to tesseract OCR engine. First, coordinate of the bounding box is

utilized to crop the license plate from the original image as the ROI (Region of Interest) as shown in Figure 5.8.

Figure 5.8 Cropped images

Second, the colored images then are converted to grayscale to reduce computation complexity as illustrated in Figure 5.9.

Figure 5.9 Grayscale license plate.

5.6.2 Tesseract OCR character recognition

Pre-processed license plate images are sent to LSTM Tesseract OCR Engine and the results are shown in figure 5.10. As it is shown in the experimentation, the succeed rate of the OCR in this application is 95%.

```
ELECHIK
```

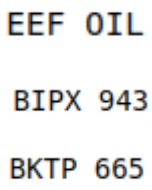


Figure 5.10 Output character from OCR.

Chapter 6. Applications and Experimentation

6.1 Vehicle and License Plate Detection

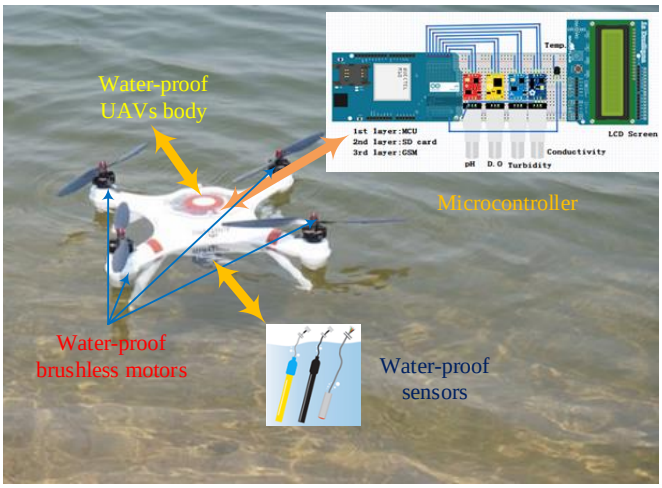
One of the tested applications of the developed system is vehicle license plate detection and recognition. It is crucial to have the developed system in this application since the application computer vision using deep learning requires intensive GPU power which, in this project, is sufficient by using the Nvidia Jetson TX2 single board computer. Meanwhile, drone has better maneuverability comparing the mobile robot and large coverage areas. The license plate detection and recognition model in the project was developed by using the methods that described in Chapter 5. The success rate of the overall system in terms of the recognition rate is 95%.

6.2 A Sensor Node with Mobility in Water Monitoring Network

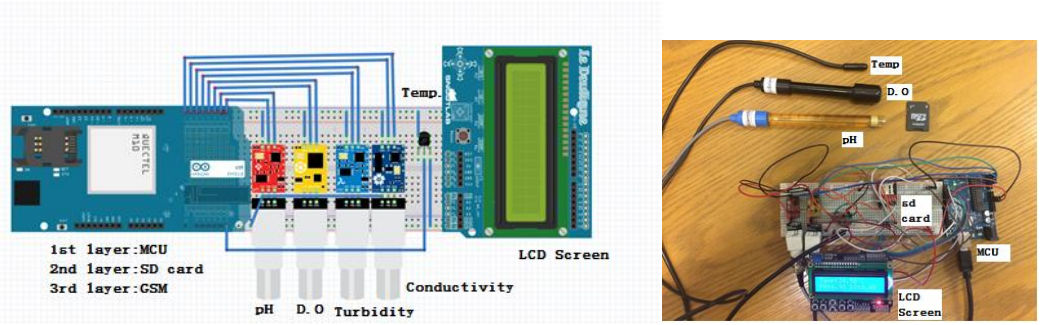
Another application of the developed drone is the implementation of a wireless sensor node with mobility in a water quality sensor network. The proposed wireless sensor network (WSNs) is effective for real-time spatiotemporal data collection. However, there are challenges in applying wireless sensor networks in water quality monitoring application because of cost, fouling and maintenance of water sensors, wireless communication and data transmission of large-scale monitoring and data. In some cases, researchers need more accurate and reliable water data in different location of the water body (e.g., river, lake) for analysis. In this case, fixed sensor network nodes may not be able to serve it because of the cost for number of sensor nodes required to cover the monitored area. Therefore, a powerful sensor node with mobility will be a great complement for the existing sensor network.

The project aimed on integrating water sensors with a water proof UAVs, acting as a moving sensor node and gateway in a water quality monitoring sensor network. The proposed approach could greatly extend the range of monitored area and amount of data. It also improves the robustness and reliability of the existing sensor network in terms of functioning as a moving sensing node and providing a gateway of exchanging and collection of sensor network data.

The developed drone with a water proof housing is utilized in this project (shown in Figure 6.1). It has ability to land and surf on water without damaging its motors and electronic components. It can also dive into the water at approximately one meter in depth for a short period of time. Advanced navigation controlled has been developed to guide the UAVs to any pointed location autonomously. Water proof sensors were carried on the bottom by the UAVs for measuring water parameter (e.g., temperature, Ph, depth, flowrate). Once it is measuring, they can wirelessly communicate and send measured data to the onboard microcontroller inside the water proof shell through Bluetooth.



(a)



(b)

Figure 6.1 (a) Application of the developed drone in water quality monitoring application; (b) Development of the water quality sensor node.

Preliminary field test results demonstrate the feasibility of the approach for measuring water parameters and data communication with existing sensor node in wireless sensor network (shown in Figure 6.2). The proposed UVA with water sensing node can be extended to measure parameter in deep water after completing the optimization of the UAVs for underwater maneuverability.

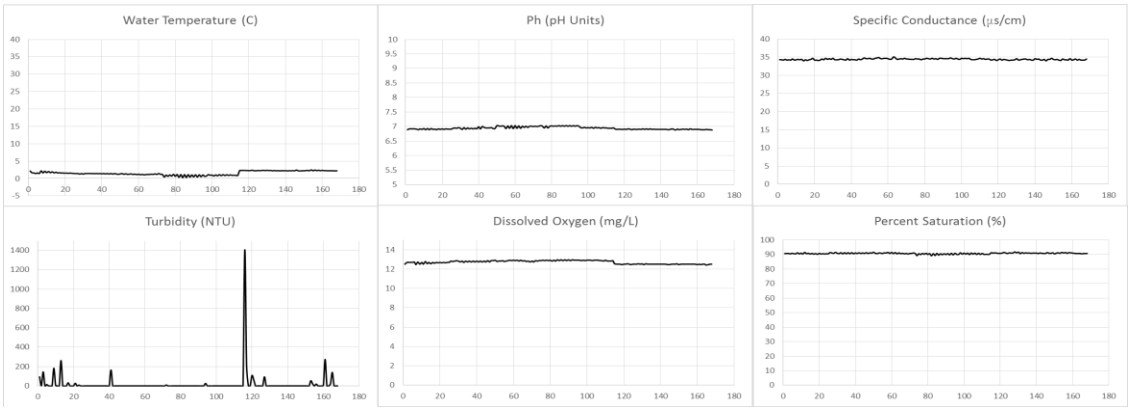


Figure 6.2 Real-time data of water parameters in Humber river at Humber village bridge.

Chapter 7. Conclusion and Future Work

7.1 Conclusion

In this thesis, a customized autonomous drone was designed and developed with its detailed design specification, part section, and achievement of the functions. The drone will be able to fly autonomously in outdoor environment. The system contains a powerful computation unit, specifically powerful GPU unit, stereo camera system and advanced sensing unit (e,g, water quality sensors). The software was developed under ROS and open-source packages including navigation, mapping functions. The localization can be accomplished by using the mounted stereo camera in combination with the GPS coordinates of the drone.

Hand gesture recognition plays a significant role in HCI technique because it can mimic the communication among humans and is an intuitive way to interact with robots. In this thesis, a robust real-time hand gesture recognition and speed estimation was proposed using the stereo sensor. A novel way of image processing using depth information was developed for distance measurement. Also, a reasonable and simple calibration method was applied to enhance the robustness and accuracy of speed estimation.

Finally, a vision-based object identification was developed and implemented in the on-board computer. It mainly achieves the function of licence plate identification and recognition. It includes a deep natural network and an OCR modular to achieve the identification, segmentation and recognition. The results showed the achievement of 95% success rate.

7.2 Future Work

Recommendations for future work related to the design and development of drone, human-machine interaction and object identification using deep learning are as follows:

- Fully integrated with ROS and utilize the ROS functions to build a modular and easy to upgrade drone.
- Estimation result is robust to background variance in Chapter 4. The proposed approach has the potential to be extended to other robotic applications such as vision servoing for manipulation, environmental monitoring and surveillance.
- If possible, more outdoor experimentations will be better to approve the effectiveness of the design. Due to the regulation and limitation of flying drone in the outdoor environment, it may be achievable with the coordination with certain department.

References

[1] N. Siriphun, S. Kashihara, D. Fall and A. Khurat, "Distinguishing Drone Types Based on Acoustic Wave by IoT Device," 2018 *22nd International Computer Science and Engineering Conference (ICSEC)*, Chiang Mai, Thailand, 2018, pp. 1-4.

[2] M.C. Harvey, J.V. Rowland, K.M. Luketina, "Drone with thermal infrared camera provides high resolution georeferenced imagery of the Waikite geothermal area," New Zealand, *Journal of Volcanology and Geothermal Research*, Volume 325, 2016, Pages 61-69, ISSN 0377-0273,

[3] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol.35, no.4, pp.399-458, Dec. 2003.

[4] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Trans. Intell. Transport. Syst*., vol.1, pp.98-107, June 2000.

[5] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110-1120, Aug. 2013.

[6] G. Dewaele, F. Devernay, and R. Horaud, "Hand motion from 3D point trajectories and a smooth surface model," in *Proc. Eur. Conf. Computer Vision*, Prague, Czech Republic, 2004, pp. 495-507.

[7] J. P. Wachs, M. Kolsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications, "*Commun. ACM*, vol. 54, pp. 60-71, 2011.

[8] A. Alostaz, “Optimized automated tracking of a moving object with a robotic eye system,” *Control and Intelligent Systems*, Vol. 44, No. 1, pp. 18-26, 2016.

[9] L.H. Kim, “Design and implementation of artificial intelligent motorized wheelchair system using speech recognition and joystick,” *Control and Intelligent Systems*, Vol. 33, No. 2, pp. 102-109, 2005.

[10] J. L. Raheja, R. Shyam, U. Kumar, P. B. Prasad, “Real-Time Robotic Hand Control using Hand Gestures,” *Second International Conference on Machine Learning and Computing*, February 9-11, 2010.

[11] A. Chaudhary, Robust Hand Gesture Recognition for Robotic Hand Control, Springer, 2017.

[12] D. J. Bora, A. K. Gupta, and F. A. Khan, “Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation,” *International Journal of Emerging Technology and Advanced Engineering*, Vol. 5, No. 2, pp. 192-203, 2015.

[13] C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” *International Journal of Computer Vision*, vol.37, no.2, pp.151–172, 2000.

[14] D. G. Lowe, “Distinctive image features from scale-invariant key points,” International Journal of Computer Vision, vol. 60, No. 2, pp. 91-110, 2004.

[15] R.Szeliski, *Computer vision: algorithms and applications*, Springer, NewYork, 2011.

[16] D. K. Ghosh and S. Ari,“On an algorithm for Vision-based hand gesture recognition,” *Signal, Image and Video Processing*, vol.10, no. 4, pp.655–662, May 2016.

[17] C. Sun, T. Zhang and C. Xu, “Latent support vector machine modeling for sign language recognition with Kinect,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 2, pp. 1–20, May 2015.

[18] P. K. Pisharady and M. Saerbeck, “Recent methods and databases in vision-based hand gesture recognition: a review,” *Computer Vision and Image Understanding*, No. 141, pp. 152-165, 2015.

[19] Y. Yao and Y. Fu, “Contour Model-based hand-gesture recognition using the Kinect sensor,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 24, No. 11, November 2014.

[20] C. Chen et al., “DeepDriving: Learning affordance for direct perception in autonomous driving,” in Proc. ICCV, 2015, pp. 2722–2730.

[21] A. Dundar et al., “Embedded streaming deep neural networks accelerator with applications,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1572–1583, Jul. 2017.

[22] Hongming Zhang, Wen Gao, Xilin Chen and Debin Zhao, "Learning informative features for spatial histogram-based object detection," *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, 2005., Montreal, Que., 2005, pp. 1806-1811 vol. 3.

[23] G. Leifman, E. Shtrom and A. Tal, "Surface Regions of Interest for Viewpoint Selection," *in IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 38, no. 12, pp. 2544-2556, 1 Dec. 2016.

[24] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.,* vol. 60, no. 2, pp. 91–110, 2004.

[25] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *in Proc. CVPR,* 2005, pp. 886–893.

[26] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.,* vol. 55, no. 1, pp. 119–139, 1997.

[27] P. F. Felzenszwalb et al., “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[28] R. M. Dufour, E. L. Miller and N. P. Galatsanos, "Template matching based object recognition with unknown geometric parameters," *in IEEE Transactions on Image Processing,* vol. 11, no. 12, pp. 1385-1396, Dec. 2002.

[29] Hill A., Taylor C.J., Cootes T. (1992) “Object recognition by flexible template matching using genetic algorithms”. *In: Sandini G. (eds) Computer Vision — ECCV'92. ECCV 1992. Lecture Notes in Computer Science,* vol 588. Springer, Berlin, Heidelberg

[30] H. Shin et al., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *in IEEE Transactions on Medical Imaging,* vol. 35, no. 5, pp. 1285-1298, May 2016.

[31] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection with Deep Learning: A Review," *in IEEE Transactions on Neural Networks and Learning Systems,* vol. 30, no. 11, pp. 3212-3232, Nov. 2019.

[32] Cai Z., Fan Q., Feris R.S., Vasconcelos N. (2016) “A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection.” *In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016.* Lecture Notes in Computer Science, vol 9908. Springer, Cham

[33] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," *2015 IEEE International Conference on Image Processing (ICIP),* Quebec City, QC, 2015, pp. 3735-3739.

[34] J. Long et al., “Fully convolutional networks for semantic segmentation,” *in Proc. CVPR,* 2015, pp. 3431–3440.

[35] S. Xie and Z. Tu, “Holistically-nested edge detection,” *in Proc. ICCV,* 2015, pp. 1395–1403.

[36] C. Peng et al., “Graphical representation for heterogeneous face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 39, no. 2, pp. 301–312, Feb. 2017.

[37] X. Gao et al., “Face sketch–photo synthesis and retrieval using sparse representation,” *IEEE Trans. Circuits Syst. Video Technol.,* vol. 22, no. 8, pp. 1213–1226, Aug. 2012.

[38] S. Zhang et al., “Filtered channel features for pedestrian detection,” *in Proc. CVPR,* 2015, pp. 1751–1760.

[39] S. Tang et al., “Detection and tracking of occluded people,” *Int. J. Comput. Vis.,* vol. 110, no. 1, pp. 58–69, 2014.

[40] Alex Krizhevsky et al, “ImageNet Classification with Deep Convolutional Neural Networks” *in Advances in Neural Information Processing Systems 25,* NIPS 2012

[41] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). “Going deeper with convolutions”. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[42] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[43] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016.

[44] Khan, Samir, and Takehisa Yairi. "A review on the application of deep learning in system health management." *Mechanical Systems and Signal Processing 107* (2018): 241-265.

[45] Pathak, Ajeet Ram, Manjusha Pandey, and Siddharth Rautaray. "Application of deep learning for object detection." *Procedia computer science 132 (2018)*: 1706-1717.

[46] Yarotsky, Dmitry. "Optimal approximation of continuous functions by very deep ReLU networks." *arXiv preprint arXiv*:1802.03620 (2018).

[47] Sengupta, Abhronil, et al. "Going deeper in spiking neural networks: Vgg and residual architectures." *Frontiers in neuroscience 13* (2019).

[48] R. Girshick et al., “Rich feature hierarchies for accurate object detection and semantic segmentation,” *in Proc. CVPR,* 2014, pp. 580–587.

[49] K. He et al., “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[50] R. Girshick, “Fast R-CNN,” *in Proc. ICCV*, 2015, pp. 1440–1448.

[51] S. Ren et al., “Faster R-CNN: Towards real-time object detection with region proposal networks,” *in Proc. NIPS*, 2015, pp. 91–99.

[52] J. Dai et al., “R-FCN: Object detection via region-based fully convolutional networks,” *in Proc. NIPS*, 2016, pp. 379–387.

[53] T.-Y. Lin et al., “Feature pyramid networks for object detection,” *in Proc. CVPR*, 2017, pp. 936–944.

[54] K. He et al., “Mask R-CNN,” *in Proc. ICCV*, 2017, pp. 2980–2988.

[55] D. Erhan et al., “Scalable object detection using deep neural networks,” *in Proc. CVPR*, 2014, pp. 2155–2162.

[56] D. Yoo et al., “AttentionNet: Aggregating weak directions for accurate object detection,” *in Proc. CVPR*, 2015, pp. 2659–2667.

[57] M. Najibi et al., “G-CNN: An iterative grid based object detector,” *in Proc. CVPR*, 2016, pp. 2369–2377.

[58] J. Redmon et al., “You only look once: Unified, real-time object detection,” *in Proc. CVPR*, 2016, pp. 779–788.

[59] W. Liu et al., “SSD: Single shot multibox detector,” *in Proc. ECCV*, 2016, pp. 21–37.

[60] J. Redmon and A. Farhadi. (2016). “YOLO9000: Better, faster, stronger.” [Online]. Available: https://arxiv.org/abs/1612.08242

[61] C.-Y. Fu et al. (2017). “DSSD: Deconvolutional single shot detector.” [Online]. Available: https://arxiv.org/abs/1701.06659

[62] Z. Shen et al., “DSOD: Learning deeply supervised object detectors from scratch,” *in Proc. ICCV*, 2017, p. 7.

[63] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems* 30.11 (2019): 3212-3232.

[64] Singh, Jaskirat, and Bharat Bhushan. "Real Time Indian License Plate Detection using Deep Neural Networks and Optical Character Recognition using LSTM Tesseract." 2019 *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019.

[65] Argentim, Lucas M., et al. "PID, LQR and LQR-PID on a quadcopter platform." *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, 2013.

[66] Bolandi, H., Rezaei, M., Mohsenipour, R., Nemati, H., & Smailzadeh, S. M. (2013). “Attitude control of a quadrotor with optimized PID controller.”

[67] Kada, Belkacem, and Y. Ghazzawi. "Robust PID controller design for an UAV flight control system." *Proceedings of the World congress on Engineering and Computer Science*. Vol. 2. No. 1-6. 2011.

[68] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.