OBSTACLE AVOIDANCE USING CONVOLUTIONAL NEURAL NETWORK FOR DRONE NAVIGATION IN OIL PALM PLANTATION

by

LEE HUI YIN

Thesis submitted in fulfilment of the requirements for the

Bachelor Degree of Engineering (Honours) (Aerospace Engineering)

June 2019

ENDORSEMENT

I, Lee Hui Yin hereby declare that I have checked and revised the whole draft of dissertation as required by my supervisor.

_____

(Signature of Student)

Date:

_____

(Signature of Supervisor)

Name:
Date:

ENDORSEMENT

I, Lee Hui Yin hereby declare that all corrections and comments made by the supervisor and examiner have been taken consideration and rectified accordingly.

_____

(Signature of Student)

Date:

_____

(Signature of Supervisor)

Name:
Date:

_____

(Signature of Examiner)

Name:
Date:

DECLARATION

This thesis is the result of my own investigation, except where otherwise stated and has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any other degree.

_____

(Signature of Student)

Date:

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project.

First and foremost, I have to thanks my project supervisor, Dr Ho Hann Woei for providing me all support and patient guidance which made me completed the project. His willingness to give his time so generously has been very much appreciated. Without his valuable and constructive suggestions in this research work, this thesis would not have been possible.

I would also like to extend my thanks to Mr Amir, the technician of the laboratory for his help in offering me the resources required in this research. Furthermore, I would also like to show gratitude to my friends for their motivation and help. Finally, I wish to thank my parents and my family members for their support and encouragement throughout my study.

OBSTACLE AVOIDANCE USING CONVOLUTIONAL NEURAL NETWORK FOR DRONE NAVIGATION IN OIL PALM PLANTATION

ABSTRACT

In Malaysia, oil palm plantation is one of the vital sectors that contribute to the country economy. In recent years, drones are widely applied in the precision agriculture due to their flexibility and capability. However, one of the challenges in a low-altitude flight mission is the ability to avoid the obstacles in order to prevent the drone crashes. Most of the previous literature demonstrated the obstacle avoidance systems with active sensors which are not applicable on small aerial vehicles due to the cost, weight and power consumption constraints. In this research, we present a novel system that enables the autonomous navigation of a small drone in the oil palm plantation using a single camera only. The system is divided into two main stages: vision-based obstacle detection, in which the obstacles in the input images are detected, and motion control, in which the avoidance decisions are taken based on the results from the first stage. As the monocular vision does not provide depth information, a machine learning model, Faster R-CNN, was trained and adapted for the tree trunk detection. Subsequently, the heights of the predicted bounding boxes were used to indicate their estimated distances from the drone. The detection model performance was validated on the testing images in term of the average precision. In the system, the drone is programmed to move forward until the detection model detects any closed frontal obstacle. Next, the avoidance motion direction is defined by commanding a yawing angle which is corresponded to the x-coordinate in the image that indicated the optimum path direction with the widest obstacle-free space. We demonstrated the performance of the system by carrying out flight tests in the real

oil palm plantation environment in two different locations, where one of them is a new place. The results showed that the proposed method was accurate and robust for the drone vision-based autonomous navigation in the oil palm plantation.

OBSTACLE AVOIDANCE USING CONVOLUTIONAL NEURAL NETWORK FOR DRONE NAVIGATION IN OIL PALM PLANTATION

ABSTRAK

Di Malaysia, pertanaman kelapa sawit merupakan salah satu sektor penting yang menyumbang kepada ekonomi negara. Kebelakangan ini, dron digunakan secara meluas dalam pertanian ketepatan. Walau bagaimanapun, antara cabaran untuk misi penerbangan di ketinggian rendah adalah keupayaan untuk mengelakkan perlanggaran daripada halangan untuk mengelakkan kemalangan drone. Kebanyakan sastera terdahulu menunjukkan sistem halangan pengelakan dengan sensor aktif yang biasanya tidak digunakan dalam kenderaan udara kecil disebabkan oleh kekangan kos, keberatan dan penggunaan kuasa. Dalam kajian ini, kami membentangkan satu sistem baru yang membolehkan navigasi autonomi sebuah dron kecil di ladang kelapa sawit dengan menggunakan kamera monokular sahaja. Sistem ini dibahagikan kepada dua peringkat utama: halangan pengesanan berasaskan penglihatan dan kawalan gerakan berdasarkan hasil dari peringkat pertama. Oleh sebab penglihatan monokular tidak memberikan maklumat kedalaman, antara satu teknik pembelajaran mesin, Faster R-CNN dilatih dan disesuaikan untuk pengesanan batang pokok. Selanjutnya, ketinggian kotak perbatasan yang diramalkan menganggarkan jarak halangan tersebut dari dron. Model pengesanan dinilai berdasarkan purata ketepatan dengan imej yang tidak termasuk dalam kumpulan latihan sebelum ini. Dalam sistem ini, drone diprogramkan untuk bergerak ke depan sehingga model pengesanan mengesan sebarang halangan frontal yang berhampiran. Seterusnya, arah pergerakan elakan ditakrifkan dengan mengarahkan sudut yaw berdasarkan koordinat-x yang menunjukkan arah laluan optimum yang mempunyai

ruang bebas daripada halangan yang paling lebar. Kami menunjukkan prestasi sistem ini dengan melakukan ujian penerbangan dalam persekitaran ladang kelapa sawit sebenar di dua lokasi yang berbeza. Antara satu lokasi ialah lokasi yang baru. Keputusan tersebut menunjukkan bahawa kaedah yang dicadangkan itu adalah calon yang tepat dan kuat untuk navigasi autonomi dron berpandukan penglihatan di sebuah ladang kelapa sawit.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UAV	:	Unmanned Aerial Vehicle
CNN	:	Convolutional Neural Network
R-CNN	:	Region-based Convolutional Neural Network
WI-FI	:	Wireless Fidelity
ROS	:	Robot Operating System
GPU	:	Graphic Processing Unit
COCO	:	Common Objects in Context
RBG	:	Red-Green-Blue
ReLU	:	Rectified Linear Units
TP	:	True Positive
FP	:	False Positive
TN	:	True Negative
FN	:	False Negative
HFOV	:	Horizontal Field of View
VFOV	:	Vertical Field of View

LIST OF SYMBOLS

O_n	:	Output of a neuron
w	:	Weight value of the neuron
b	:	Biases value of the neuron
α	:	learning rate
IoU	:	Intersection over Union
P	:	Precision
R	:	Recall
N	:	Number
AP	:	Average Precision
h_{object}	:	Height ratio of an object appears in the image
w_i	:	Width between bounding boxes and image borders [pixel]
x_{opt}	:	X-coordinate of the optimum path direction line
φ_{opt}	:	Steering angle to head the optimum path direction [deg]
$\varphi_{desired}$	:	Desired Heading angle [deg]
$\varphi_{heading}$	:	Current Heading angle [deg]
K_p	:	Proportional gain in control
(x_b, y_b, z_b)	:	Drone body reference axes
(x_i, y_i)	:	Image coordinate axes

$(x_{i_{j,min}}, y_{i_{j,min}})$: Coordinates of the upper left corner of the j^{th} bounding box

$(x_{i_{j,max}}, y_{i_{j,max}})$: Coordinates of the bottom right corner of the j^{th} bounding box

CHAPTER 1

1 INTRODUCTION

An unmanned aerial vehicle (UAV), generally known as a drone, is defined as an aircraft without a human pilot onboard for the navigation control. A drone is a remotely controlled aircraft that is either flown by a human pilot at the ground or navigated autonomously by the pre-programmed automation systems. Originally, drones are designed to be used in the operations which are conducted in the remote, dull or dangerous situations, especially for the military defence purposes. With continuous research and developments, nowadays, drones are utilized widely in the military and civilian applications, such as military real-time monitoring, resource exploration, civil surveillance, cargo transportation and agricultural planning.

In recent years, drones are started to be applied in precision agricultural (Zhang and Kovacs, 2012, Mogili and Deepak, 2018) due to the flexibility and capability of a drone compared to the labour dependent techniques. Furthermore, the ground sensing and advanced technology of satellite remote sensing (Drusch et al., 2012) that were applied in smart plantations previously, are very useful but the operation and equipment costs are too high, especially for the small to medium scale enterprises. Hence, the implementations of drones with the onboard cameras or sensors become the relatively low cost alternatives to the small scale enterprises to perform precision agricultural missions without comprising the required performances.

The drones can be used to collect images and other information from the onboard sensors. A mission can be performed efficiently and effectively by processing the data obtained from the drones. Using the data gathered and processed, drones can assist in

many tasks in the plantations, such as the plantation analysis and planning (Chebrolu et al., 2018), plantation surveillance (Herwitz et al., 2004), and the subsequent monitoring of fields to ascertain health and growth which include the crop monitoring (Lelong et al., 2008) and soil sampling analysis (Demattê et al., 2018, Huuskonen and Oksanen, 2018, Ivushkin et al., 2018).

In Malaysia, agriculture sector, especially the oil palm plantation, is one of the important sectors that contribute to the country economy. Malaysia is known as the second largest palm oil producer in the world after Indonesia (Alam et al., 2015). With the current technology developed, in Malaysia, the drones are utilised in the smart oil palm plantations by flying the drone at a high altitude to perform precision agriculture missions, such as tree counting and monitoring (Li et al., 2016, Tugi et al., 2015, Chong et al., 2017). However, to obtain a high-resolution image from a high altitude in order to observe the soil conditions for the health assessments, the drone tends to carry a heavy and costly multispectral or hyperspectral camera (Chong et al., 2017) which causes a high power consumption as well. The high power consumption of the instruments will affect the flight endurance and range of the drone.

At a low-altitude or near the ground flight, the drone can captures the images at a nearer distance to the target objects, such as the soil or crop. Hence, a commercially available camera is sufficient to capture clear images for the further data processing and analysis. In addition, compared to the applications of the drone at a high altitude, the drone in a low-altitude flight can be used to perform more missions, such as crop monitoring, and fertiliser spraying with a suitable quantity based on the condition of the soil. The applications of drone at a low altitude offer more cost-effective and efficient solutions compared to the labour in a vast field.

However, the greatest challenge to overcome in a low-altitude flight is the ability of the drone to avoid collision from the obstacles in order to prevent the drone crashes. Traditionally, the drone navigations are dependent on the pilots which might prone to have human errors in the operations. To achieve autonomous navigation, the capability of obstacle detection is a significant key in order to identify and feedback the information of the surrounding environment to the system. The system will produce a motion command based on the environment information to avoid the detected obstacles. This reduces the risk of collisions caused by the human operation handling errors.

Commonly, the drone obstacle avoidance systems are performed by using the active sensing sensors (Fasano et al., 2008), such as LIDAR (Ramasamy et al., 2016), inertial sensors, ultrasonic and infrared range finders (Gageik et al., 2015) or the passive sensors, such as RGB-D camera (Iacono and Sgorbissa, 2018), stereo camera (Barry et al., 2018) or multiple cameras, etc. The cost, weight, and power consumption of these instruments become a great constraint to the drone mission. Therefore, a single camera is preferable in an obstacle avoidance system as it is light-weight, commercially available, and low power consumption.

Generally, monocular vision-based obstacle detection methods can be divided into two categories: motion-based and knowledge-based methods. Optical flow (Lee et al., 2010, Yoo et al., 2011, Eresen et al., 2012) is a typical approach used to detect and estimate the depth based on the motion information. The obstacle may be missed due to the wrong detection of feature points, or there is no feature point available in the image. Since monocular vision does not allow accurate and robust distance geometric measurement, often machine learning-based solutions or the approaches which are combined with the optical flow methods (Ho et al., 2018) have been proposed.

Since Alexnet (Krizhevsky et al., 2012) won ImageNet competition in 2012, Convolutional Neural Network (CNN) became the gold standard for image classification. Since then, CNN approaches were improved until now they outperform humans in the ImageNet challenge (Alom et al., 2018). Lately, CNN has been applied with a great success to the detection, segmentation and recognition of objects in images (LeCun et al., 2015). For instance, CNN approaches were applied in the vehicle detection from aerial images (Qu et al., 2017), the medical image detection (Hoo-Chang et al., 2016) and the fruit detection from images by Faster Region-based CNN (Faster R-CNN) (Sa et al., 2016), etc. Thus, CNN is a good option to be trained as a detector in an obstacle avoidance system for drone autonomous navigation.

Nowadays, there are few models that demonstrated object detections with CNN approaches. Deformable Parts Models (DPM) (Felzenszwalb et al., 2010) used the sliding window approach (Vedaldi et al., 2009) which the classifier was run at evenly spaced locations over the entire image, to detect the presence of the targets in the image. To bypass the problem of selecting a huge number of regions when using sliding window algorithm, the selective search method (Uijlings et al., 2013) was proposed to extract just 2000 regions from the image which were used as region proposals in R-CNN model (Girshick et al., 2014). The other demonstrated approaches for region proposal were the Edgebox (Zitnick and Dollár, 2014) and the grouping superpixels (Rantalankila et al., 2014).

CNN acted as a feature extractor to feed the features into a Support Vector Machine (SVM) to classify the presence of the objects within the candidate region proposals. However, R-CNN model need a long computational time as it required a forward pass of the CNN for every single region proposal in every single image. Hence, Spatial Pyramid Pooling networks (SPPnets) (He et al., 2014) were proposed to speed up

the R-CNN model by sharing computation with spatial pyramid pooling. However, the fine-tuning algorithm proposed in SPPnets cannot update the convolutional layers and affect the accuracy. Fast R-CNN model (Girshick, 2015) achieved a higher detection quality and shorter test time by introducing the RoI (Region of Interest) Pooling which allowed every RoIs to share the forward pass of a CNN for an image. Despite all, the slow external region proposals in the models were affecting the test time performances.

To overcome the limitations, the Fast R-CNN and Region Proposal Network (RPN) were merged in the Faster R-CNN model (Ren et al., 2015) to share the computations and use neural networks instead of external region proposal methods to propose regions. With the advantage of short test time, the Faster R-CNN model was proposed to use for real-time object detections. Next, the evolution of You Only Look Once (YOLO) models up to YOLOv3 model (Redmon et al., 2016, Redmon and Farhadi, 2017, Redmon and Farhadi, 2018) had brought the object detection algorithms to another higher performance level by modelling detection as a regression problem. YOLO algorithms eliminated the need for the region proposal method and used the entire image during the training and testing process. Thus, it implicitly encoded contextual information about classes as well as their appearance. However, the accuracy of a single shot detector, YOLO, had a lower accuracy compared to the region-based algorithms. Furthermore, currently, the open source algorithm of the model is not publicly available in the online resources yet.

From the other side, Faster R-CNN algorithm is publicly available in the open sources which the users can either construct the network from scratch or import a pre-trained network to be trained as an obstacle detector. The outstanding performance of Faster R-CNN model was proved in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and Common Object in Context (COCO) 2015 competitions.

Faster R-CNN was the foundation of the first place winning entries in several tracks of these competitions. To the best of the knowledge, this research is the first approach to adapt Faster R-CNN in drone navigation.

In this research, by using the monocular vision, a machine learning approach is proposed to allow the drone to recognise the features of the tree trunks in the images then perform the obstacle detection and avoidance control in an oil palm trees field. The approach involves the configuration of Faster Region-based Convolutional Neural Network model (Faster R-CNN) as the detector to localise the frontal obstacle. The system is divided into two main stages: vision-based obstacle detection, in which the obstacles in the images captured by the front single camera are detected, and motion control, in which the avoidance decisions are taken based on the result from the first stage and sent back to the drone.

1.1 Problem Statement

Commonly, a small multi-rotors drone, such as a micro aerial vehicle, has a limited flight endurance which is around 15 to 30 minutes only. This is due to the limitation of the battery and the power consumption by the rotors to provide the thrust to the drone. Hence, the additional payloads, such as sensors or multi-camera, applied on the drone to detect the presence of obstacles will reduce the flight endurance and range of the drone significantly. This will be a constraint for the further development and applications of drone in precision agriculture. Hence, it is essential to develop an obstacle avoidance algorithm which eliminates the needs of sensors in object detection for the autonomous navigation purpose. A single camera onboard of the drone is an alternative replacement to the sensors.

Furthermore, although Faster R-CNN performs well in the common object detection in different competitions and challenges, it is necessary to verify its practicality in a real-life scenario and practical implementation on the drone navigation. Besides, the factors that affect the performance of a Faster R-CNN model need to be investigated to construct a scratch detector model and optimise the performance in the detection of the tree trunks.

Lastly, in a real flight, the environment factors might affect the stability of the drone. When flying in the plantation, the drone might not able to fly at a constant height from the ground due to the external issues, such as the ground might be soft after rain and unexpected gust, which can cause the drone to become unstable. This also causes the view of the camera is different from time to time. Thus, this is a challenge to overcome in a vision-based obstacle avoidance system.

1.2 Objective

The research work in this thesis is performed to achieve the following objectives:

1. To study the performance of a CNN-based detector on different flight motions.
2. To construct and optimise the performance of a scratch Faster R-CNN model with different specifications.
3. To compare the performance of a scratch model and the pre-trained Faster R-CNN models in the tree trunks detection.
4. To develop an obstacle avoidance algorithm and implement in drone autonomous navigation in oil palm plantation.

1.3 Thesis Layout

This thesis consists of 5 main chapters which are Introduction, Literature Review, Methodology, Result and Discussion, Conclusion and Recommendation. Each chapter is further divided into several sub-chapters as appropriate.

Chapter 1 introduces the applications of drones, especially in the agricultural sector, followed by the background of the evolution of the CNN approach in the detection task. The problem statement and objective of this research work are also presented in the first chapter. Chapter 2 focuses on the review of approaches done by the previous researchers on the drone autonomous navigation.

Chapter 3 is divided into 2 main sections to discuss the approaches used in obstacle detection and avoidance Motion Control, respectively. The methods used to train and implement the Faster R-CNN model in the drone image processing and the flight control are discussed.

Chapter 4 shows the results of the performance of the detector models and the results from the real flight tests of the drone in the oil palm trees fields. Finally, the findings of the work are concluded and recommendations to improve the performance in the drone autonomous navigation in the future are done in Chapter 5.

CHAPTER 2

2 LITERATURE REVIEW

Nowadays, many researchers use monocular images as the inputs to control the drone in the autonomous navigation. However, with monocular vision, the depth information is not available directly. Hence, there were few approaches proposed to overcome this limitation to utilise a single camera in the drone autonomous navigation.

Firstly, using a single camera, the movement of the drone was used to reconstruct the scenes in the Structure from Motion (SfM) approaches. From a small set of consecutive images, a regularized depth map was computed and subsequently used for the waypoint generation (Alvarez et al., 2016). A direct depth estimation approach was proposed by enabling real-time computation of dense depth maps and navigation in a cluttered outdoor environment (Daftry et al., 2016). However, in the SfM-based obstacle avoidance scheme, the drone was not able to avoid dynamic obstacles which were moved during mapping or between mapping cycles. Moreover, the mapping cycle required the computational memory to store and compare the consecutive frames of the scene in order to obtain the depth information.

Most of the monocular vision obstacle avoidance research focused on the demonstration of the accurate depth measurements as the monocular vision cannot provide depth information directly. The common methods are optical flow-based methods and SLAM-based methods. An optical flow method was proposed to obtain the tested structure in a 3D space environment based on the gradient method of Lukas-Kanade (Gosiewski et al., 2011). By comparing the sequential images, the model found

out whether the obstacle was getting closer. Hence, the movement of the drone were controlled by producing a steering command which was inversely proportional to the optical flow difference between the two sides of the image (Agrawal et al., 2017).

Moreover, a variety of bio-inspired optical flow navigation methods have also been proposed (Zufferey and Floreano, 2006). For instance, the translational optic flow that was inspired by the method that the insects employ for the collision-free navigation, was demonstrated (Serres and Ruffier, 2017). However, the optical flow-based method cannot acquire precise distance, which may limit the usages in some specific missions.

By contrast, SLAM-based methods were proposed to provide precise metric maps with a sophisticated SLAM algorithm. With the algorithm proposed, the drones navigated and avoided obstacles with more environment information obtained from the low-cost ultrasonic and infrared range finders (Gageik et al., 2015). For instance, the application of Scanning LIDARs for SLAM on the drone navigation was demonstrated successfully in the indoor environments (Bachrach et al., 2009). As the instruments proposed previously were heavy to the drone, a method based on Oriented fast and Rotated Brief SLAM (ORB-SLAM) was proposed to process the video stream of the front camera (Esrafilian and Taghirad, 2016). First, it computed the 3D locations of the drone and generated a sparse point cloud map. Then, it enriched the spare map to denser. Finally, it generated a collision-free roadmap by applying potential field method and quickly exploring the Random Tree (RRT). These SLAM-based obstacle avoidance systems performed much more complex tasks though, but usually fail at high speeds since they reconstructed the environment from frame to frame triangulations.

Some approaches detected the presence of the frontal obstacles and then adjusted the motion control to avoid them. These approaches are commonly categorised as the

sense-and-avoid mechanism. The classification algorithms were used to detect the types of environment, and then for each environment perspective cue, the desired direction for the drone to fly was extracted (Bills et al., 2011). However, this approach was limited to the simple indoor environment only, and it cannot be applied in complex and clustered outdoor condition.

Moreover, the changes in the size area of the obstacles were proposed to predict the depths (Al-Kaff et al., 2016, Al-Kaff et al., 2017). Furthermore, a feature point detector, Speeded Up Robust Features (SURF) was proposed for the fast obstacle detection (Aguilar et al., 2017). Lastly, the outline of obstacles from Multi-Scale Oriented Patches (MOPS) and the spatial coordinates of feature points from the Scale Invariant Feature Transformation (SIFT) algorithm were proposed to merge in order to show the 3D information of the obstacles (Lee et al., 2011). However, the drawback of these algorithms is they only work for the obstacles stored in the database.

There were several approaches which were applied to the similar environment as our research. The Dagger algorithm was presented to learn and predict the control from the human expert through dense forest environments (Ross et al., 2012). Furthermore, a hybrid collision avoidance scheme which consists of the Rapidly exploring Random Tree (RRT) as the global path planner and a fuzzy logic method, was proposed as the local collision avoidance mechanism. For the development of the proposed path following scheme, the extended Kalman filter was utilized for estimating the cross-track error of the flight in the hazardous environment (Liu et al., 2018).

Recently, deep learning-based solutions have proposed to improve real-time performance in a complex unknown environment. Firstly, Convolutional Neural Network (CNN) was used to learn a control strategy that mimics an expert pilot's choice

of action to navigate autonomously at the indoor (Kim and Chen, 2015) and outdoor environment (Giusti et al., 2016), respectively. For instance, in a navigation approach demonstrated at the outdoor environment (Giusti et al., 2016), the camera orientation estimation was framed as a three-class motion classifications which were Left, Front and Right. A set of forest trail images was captured with three head-mounted cameras, each pointing in one direction. Given one frame input, the model decided the next optimal move. However, this work was demonstrated to follow a specific path, but this research is working on the general navigation.

There were few researchers who presented deep learning solutions to predict the depth of the scenes (Liu et al., 2016, Jia et al., 2016, Chakravarty et al., 2017). A fully convolutional network which was fed with both images and optical flows, was designed to obtain fast and robust depth estimation (Mancini et al., 2016). In addition, a two-stage obstacle avoidance deep reinforcement learning system was proposed. It was composed of a depth predictor fully convolutional neural network followed by a double-Q network (D3QN) which consists of a convolutional network and a duelling network to predict the Q-value of angular actions and linear actions in parallel (Xie et al., 2017).

Most recent, J-MOD2 which was a novel CNN architecture, was proposed to jointly learn the task of the depth estimation, and the obstacle detection from the image feature extracted by the fine-tuned VGG19 network (Mancini et al., 2018). The approach was tested and evaluated in a virtual forest scenario on the Unreal Engine software environment. Furthermore, a saliency detection algorithm was developed by using a deep CNN to extract monocular visual cues and Radial Basis Function (RBF) neural network in an actor–critic reinforcement learning module to control the motion of the drone (Ma et al., 2018). The types of the state-of-the-art drone sensing and detection methods are summarised in Figure 2-1.

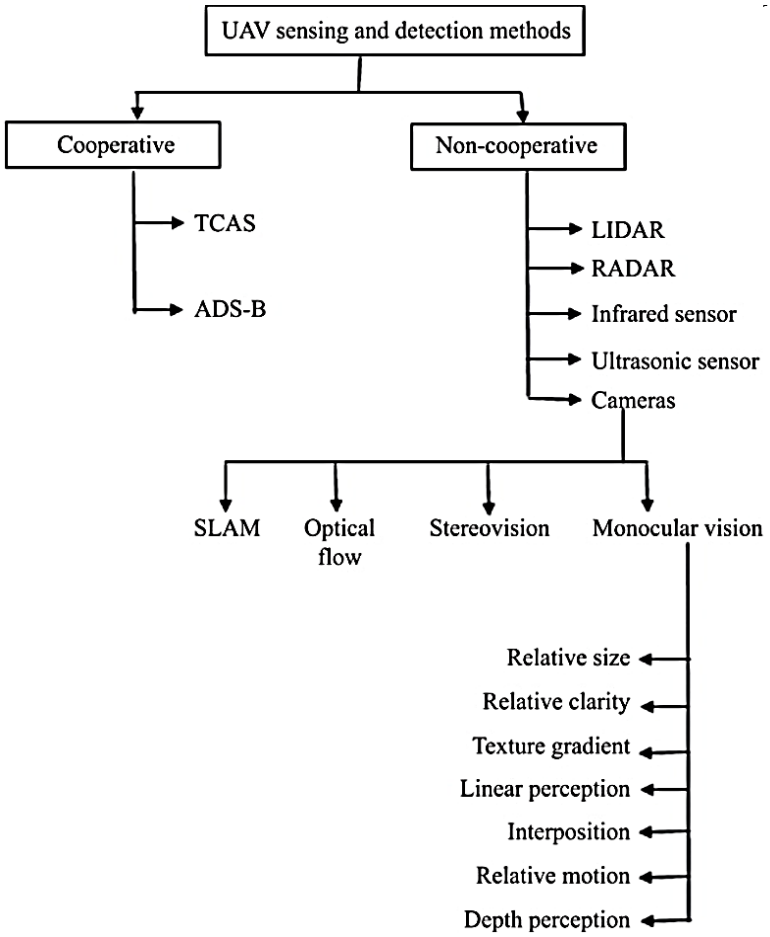

Figure 2-1: Summary of Drone Detection and Sensing Methods (Aswini et al., 2018)

This research proposes to adapt Faster R-CNN model as the tree trunk detector. This is because Faster R-CNN model has shown a short computational time and high mean Average Precision (mAP) in other robust applications. The drone is proposed to avoid the frontal obstacles when it detects the approaching obstacles by estimating the distance from obstacles. In contrast to the approaches that compared the consecutive frame of images to estimate the distance of obstacles, the heights of the bounding boxes are used to indicate the estimated distance. Hence, the computational memory for the previous image frames is not needed. This algorithm aims to have a high mAP and low computational time in order to be suitable for the drone vision-based navigation application.

CHAPTER 3

3 METHODOLOGY

In this study, a commercially available quadrotor, namely the Parrot Bebop Drone 2 is used as the airborne platform to collect the dataset and involve in the flight test to evaluate the result. The performance of a scratch model and pre-trained models of Faster R-CNN with different convolutional bases trained for the tree trunk detection, are compared. The trained model with the best performance is selected to adopt in the obstacle avoidance system.

The Robot Operating System (ROS) framework is used to communicate with the software development kit (SDK) of the Parrot Bebop Drone 2 using the developed bebop_autonomy package. The drone sends the data collected onboard, such as the images and attitude information, to the ground control station through Wi-Fi connection. After the ground control station makes the motion control decision based on the detection result, the command is sent back to the drone to avoid the critical obstacle if it is necessary.

This system was run within ROS Kinetic on Linux Ubuntu 16.04 with Intel Core i7-7500U MB CPU and 32GB RAM. Nvidia 940MX GPU was used for extensive mathematical computations in detection algorithm to free up CPU cycles for other jobs and speed up the computation time for the detection. This is because the detection algorithm deals with the complex convolutional computation and the large size data as the input is an image. At the end of the project, the real field environment flight tests are carried by using Parrot Bebop Drone 2 in several oil palm tree fields to evaluate the performance of the obstacle avoidance system.

3.1 Object Detection

3.1.1 Data Collection and Annotation

In order to train and validate a detection model, a large dataset which consists of the images taken from the target environment is required. In this research, Parrot Bebop Drone 2 is used as the airborne platform to fly and record the flight videos in an oil palm trees field in Engineering Campus of University Sains Malaysia. The Bebop Drone 2 features 14 megapixels with a fish-eye lens and can produce a 3-axes full HD 1080p video at 30 frames per second. The exact dataset collection location is shown in Figure 3-1.

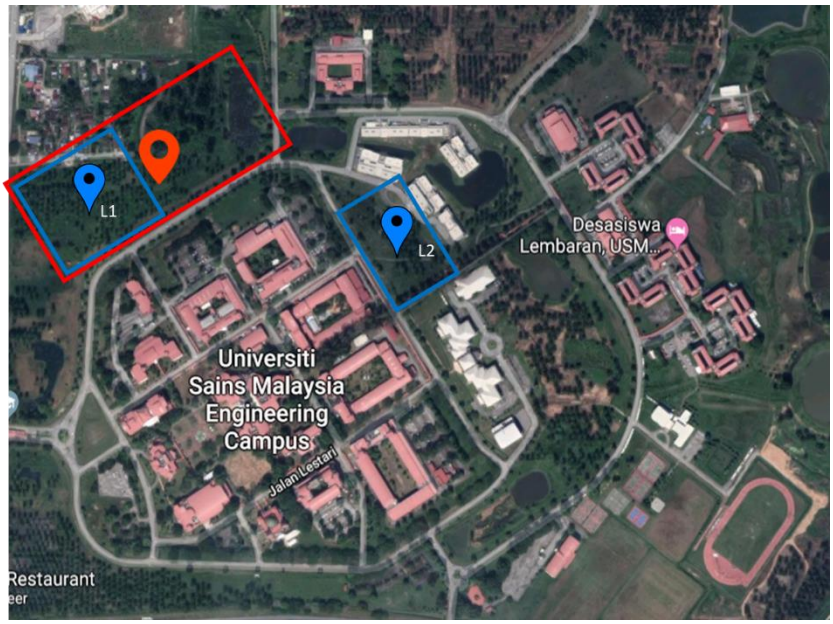

Figure 3-1: Map of USM Engineering Campus. The red rectangle with icon shows the area of the location of the dataset collected, whereas the blue rectangles with icons show the test flight locations which labelled as Location 1 (L1) and Location 2 (L2), respectively.

There are two types of dataset collected in the form of flight videos captured using the front looking camera onboard of the drone. During the flight to collect both types of dataset, the drone is flown at a constant flying height of 1.5 m from the ground in the oil palm trees field. The built-in autopilot functions with the inputs from the

onboard sensors including a 3-axes gyroscope and an ultrasonic sensor which can analyzes the flight altitude up to 8 meters, are able to control and maintain the drone in a constant height during the flight.

Figure 3-2 shows the body axis reference system fixed to the centre of gravity of the drone. The x_b-axis is positive toward the direction of the forward movement, whereas the z_b-axis is aligned with the gravitational force. The y_b-axis is perpendicular to both axes and is directed in such a way that (x_b, y_b, z_b) is a right-hand triad.

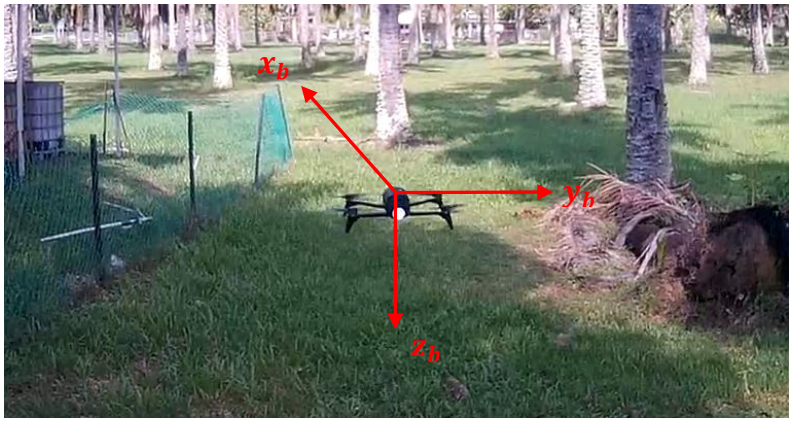

Figure 3-2: Reference Body Axis (x_b, y_b, z_b) of the drone

The first type of dataset is collected by alternating positions of the drone along the x_b-axis and y_b-axis, separately in the oil palm trees field to compare the performance of the networks in detecting the obstacles in the different motions. The second dataset consists of the flight videos of the drone random navigation in the oil palm trees field to train the detector with a real flight scenario. The data collection is conducted in three different lighting conditions in the same environment to ensure the model can detects the obstacles in any lighting condition.

The videos recorded by the drone are originally in size of 1920 x1080 pixels. The videos are resized to 426 x240 pixels before proceeding to the object annotation. This is

because the original input image size is too large, and it causes the training process of the detector becomes expensive in terms of computational time and specifications of the hardware needed. These images are resized to suit with the memory size of the available GPU in the laptop to train the detector. Furthermore, this can reduce the training process time to a reasonable period.

Before the detection model is trained, the ground truth annotations are needed for both training and validation dataset. A ground truth represents the correct position and class of the object interested in the image. The dataset is labelled manually in Video Labeler application in MATLAB. All the oil palm tree trunks are labelled with rectangular Regions of Interest (RoI) labels in every frame of the videos with the aids of the built-in automated algorithms in the application, such as Point Tracker and Temporal Automation Algorithms.

In this scenario, the only type of object of interest is the tree trunk of the oil palm tree in the image. Therefore, there are only two classes of object in an image which are a tree trunk or the background. All the bounding boxes are defined by the coordinates of the upper left point in the image together with the width and height of the boxes. Figure 3-3 shows the example images with the rectangle RoI labelled around the object of interest, “tree_trunk”. Finally, the labelled ground truth is exported to a table form and used in detector performance validation or training.

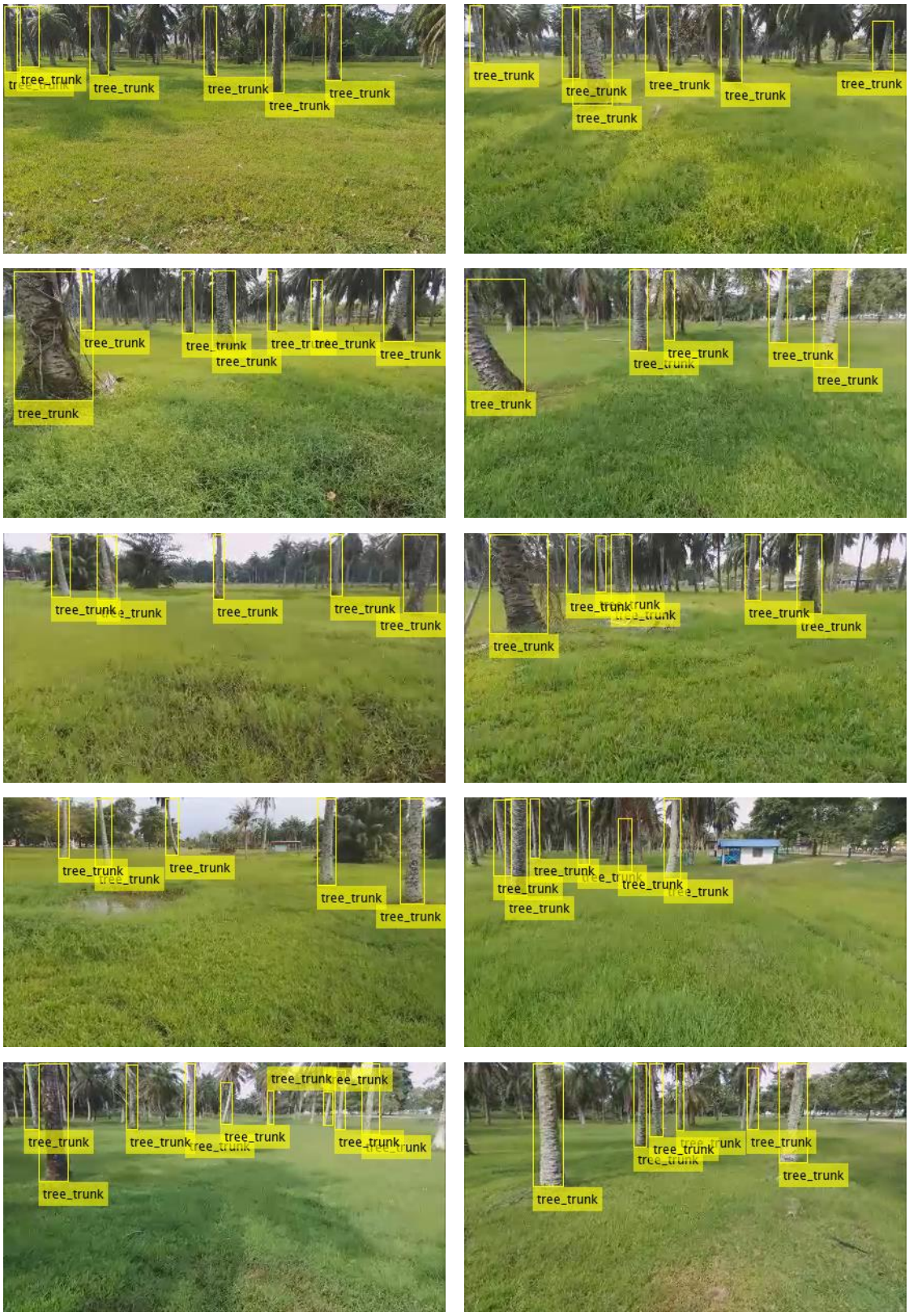

Figure 3-3: Examples of the ground truth labelled in the images. The yellow boxes are the labelled bounding boxes to represent the tree trunks in the images.

3.1.2 Network Architecture and Training

Next, the detector network is constructed and trained. There are two types of models trained to compare their performances. The first type of the model is a scratch model, in which the architecture of the Convolutional Neural Network base need to be defined by the user. This includes the number of layers, type of the activation function, size of the filter, padding and stride in each convolutional layer that need to be considered before constructing the scratch model. The weight and bias of each node in each layer is trained from zero to a value that can fit to the dataset. The training process is done with MATLAB algorithm.

The second type of the model is the pre-trained model that imported from the online resource. The architecture of the model is well defined, and it has been trained in a large dataset previously. In order to transfer the ability of the feature extraction and classification from the previous tasks to the identification of the tree trunks in oil palm trees field, the training method is different from that of a scratch model as the transfer learning technique is needed. The training process is done with Tensorflow algorithm in Python language under Linux environment.

3.1.2.1 Scratch Models

The first step is done by defining the CNN network architecture. There is a total of four different detectors created to investigate the effect of the amount of training data and the number of the convolutional layers on the performance of the detectors. This can be a guideline to construct and train a scratch model with the specifications that are the most suitable to solve this detection problem, before it is implemented in the obstacle avoidance system.

There were four detectors trained with different specifications. The first parameter manipulated was the number of training images. Firstly, Detector A was trained with more than 4,000 images, whereas the other 3 detectors were trained with 1,500 images only. The images in the training dataset of the other 3 detectors were extracted from the training dataset of Detector A. In other words, the training data of Detector B, C and D are the same and are the subset of training data of Detector A.

Secondly, the parameter manipulated is the number of repeating blocks that consists of convolutional and activation function ReLU layers. The convolutional layer is the important part of a detection model as it extracts the features and produces the convolutional feature map to feed into the last layer and identify the presence and location of the tree trunk in the images. As a deep architecture of the convolutional neural network is computationally expensive in training a new model in terms of training time and GPU memory space required, the architecture of the scratch model created needs to keep simple and shallow. Therefore, there were 3 detectors, Detector B, C and D created with the different number of repeating convolutional blocks in the range of 2 to 5 layers. Table 3-1 shows the summary of the architecture of CNN of Detector A and B.

The architecture of the CNN is started with an image input layer. The image input layer feeds the images to a network and applies data normalization. The type and size of the input layer are defined. For common classification tasks, the input size is typically the size of the training images. But in the detection tasks, the CNN need to analyse the smaller sections of the whole image which is the image of the object of interest, so the input size must be similar to the size of the smallest tree trunk in the dataset. Since RBG images are fed into the network, an input size of [32 32] with the depth of 3 is selected. Furthermore, the data transformation is applied when the data is forward propagated through the input layer by subtracting the average image in training.

Table 3-1: The CNN architecture of Detector A and B

No.	Layer Name	Description
1	Image Input	32x32x3 images with 'zero-center' normalization
2	Convolutional layer	32 3x3 convolutions with stride [1 1] and padding [1 1 1 1]
3	ReLU	ReLU activation layer
4	Convolutional layer	32 3x3 convolutions with stride [1 1] and padding [1 1 1 1]
5	ReLU	ReLU activation layer
6	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
7	Fully Connected	64 fully connected layer
8	ReLU	ReLU activation layer
9	Fully Connected	2 fully connected layer
10	Softmax	Softmax activation layer
11	Classification Output	Classification

Next, the middle layers are made up of repeated blocks of convolutional, Rectified Linear Units (ReLU), and maximum pooling layers. These layers form the core building blocks of the convolutional neural networks. The convolutional layers apply a convolution operation to the input with the use of filters or kernels and produce a feature map. In the convolutional layer, 32 filters with the size of 3 x 3 are used to scan along the images with a stride [1 1]. The number of filters is equal to the number of neurons that are connected to the same region of the input which determines the number of feature maps. Moreover, a stride of padding is added to the input feature map borders to ensure that the spatial output size is the same as the input size.

Then, the activation step applies a transformation to the output of each neuron by using activation functions. In here, ReLU is selected as the activation function to perform a threshold operation to each element of the input from the convolutional layer as shown

in Equation (3.1). This function takes the output of a neuron, O_n and maps it to the highest positive value, or maps it to zero if the output is negative.

$$f(x) = \begin{cases} O_n, O_n > 0 \\ 0, O_n \leq 0 \end{cases}. \quad (3.1)$$

Afterward, the max pooling layer is introduced in the architecture of CNN to reduce the dimensionality of the feature map. This is done by taking the maximum value from each 3 x 3 patch area of an image, and then placing it in a new matrix next to the maximum values from other patches. The rest of the information contained in the activation maps are discarded. This can help to simplify the following layers and reduce the number of parameters that the model needs to learn. However, to avoid down-sampling the data prematurely, the number of pooling layer is kept as low as 1 layer only. This is because down-sampling in the network might discard image information that is useful for the training.

The final part of the network is composed of two fully connected layers and a softmax layer. A fully connected layer combines all of the features extracted by the previous layers across the image to identify the larger patterns. The last fully connected layer combines the features to classify the images. Hence, the output size of the last fully connected layer of the network is equal to the number of the object classes in the image which are the tree trunk and the background.

At this point, the network produces outputs that can be used to predict whether the input region of interest belongs to the tree trunk or the background. Then, the softmax function calculates a probability for the object on the image being predicted which known as a confidence score. Finally, the classification layer uses the probabilities returned by the softmax activation function for each input to assign the input to the class of objects. Figure 3-4 illustrates the architecture of the CNN network of Detector A and B.

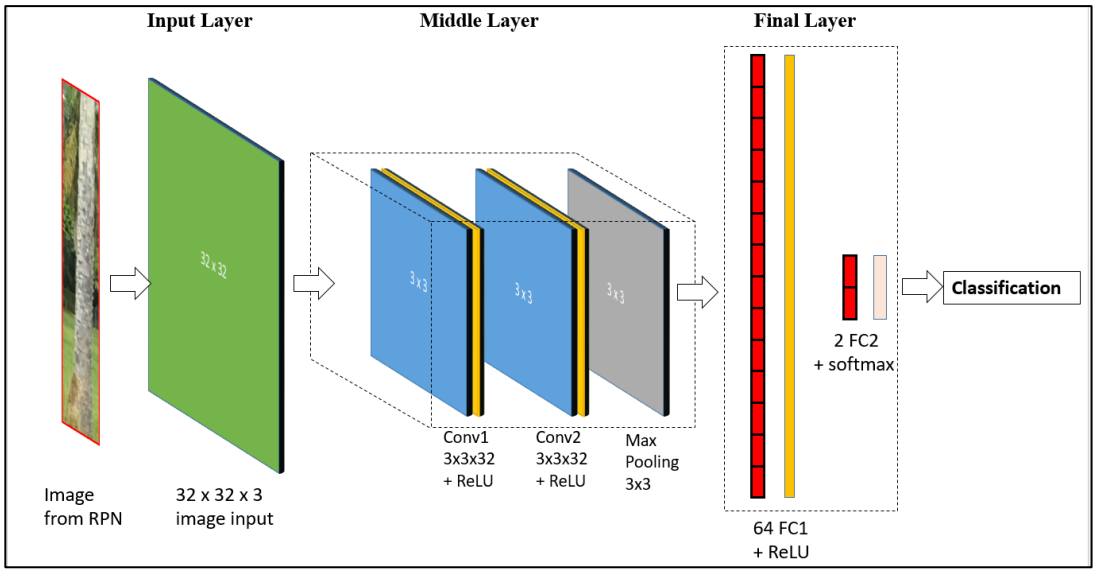

Figure 3-4: Architecture of the CNN network of Detector A and B. Conv. represents the convolutional layers, whereas FC represents a fully connected layer.

After the architecture of the network is defined, the Faster R-CNN detector is trained in four steps. The first and second step train the Region Proposal Network (RPN) and CNN network defined above in a Faster R-CNN, respectively. The third and fourth step combine both networks from the weight resulted from the first two steps, such that a single network is created for detection. In detail, the RPN training is initiated by fixing the shared convolutional layers, and only fine-tuning the unique layers of RPN in the third step. On the other hand, in the fourth step, the unique layers of detector network are fine-tuned, while the convolutional layers are fixed.

The algorithm used in the training is Stochastic Gradient Descent with Momentum (SGDM) optimizer. The stochastic gradient descent algorithm evaluates the gradient and updates the parameters using a subset of the training set. For Faster R-CNN training, the mini batch size must be equal to 1 as the training algorithm creates a training batch by sampling multiple regions within an image. At each iteration, the algorithm takes one step towards minimizing the loss function, $J(w_l)$.

The gradient descent algorithm updates the network parameters which are the weights, w and biases, b of each neuron to minimize the loss function by taking small steps in the direction of the negative gradient of the loss. Equation (3.2) and (3.3) show the gradient descent algorithm applied to update weight and bias of each neuron during the training process.

$$w_{l+1} = w_l - \alpha \nabla J(w_l). \tag{3.2}$$

$$b_{l+1} = b_l - \alpha \nabla J(b_l). \tag{3.3}$$

The learning rate, α determines the size of the steps taken to reach a minimum loss. In here, a value of 0.001 is selected as the optimum learning rate as a large value of α may miss the global minimum and caused it fail to converge to a solution, whereas a small value will take too long time before the minimum point is reached.

3.1.2.2 Pre-trained Model by Transfer Learning

Transfer learning is one of the techniques in machine learning, which reuses a model that trained for a general task by transferring the outcome in the previous training to another similar target task. The model is known as the pre-trained model which has been trained on a large benchmark dataset for a mission, such as classification, localization or segmentation, which should be similar to the target task. The model contains the weights and biases in each layer that represent the features of dataset trained initially, and can be transferred to a different type of data. These models are available in the online resources, and can be imported then trained to solve the target problem.

There are two pre-trained models selected to be trained for the tree trunk detection. Both pre-trained models were trained previously on the COCO dataset which has a total