LOW-COST AND SCALABLE VISUAL DRONE DETECTION SYSTEM BASED ON DISTRIBUTED CONVOLUTIONAL NEURAL NETWORK

by

Hyun Hwang

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

December 2018

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Eric T. Matson, Chair

Department of Computer and Information Technology

Prof. Anthony H. Smith

Department of Computer and Information Technology

Dr. Byung-Cheol Min

Department of Computer and Information Technology

Approved by:

Dr. Eric T. Matson

Head of the Graduate Program

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

t_c A timestamp at when the associated frame was captured from the camera module

t_{cf} Same as t_c, but of the first frame

t_{cl} Same as t_l, but of the last frame

t_s A timestamp at when the associated frame is about to be processed by a worker

t_{sf} Same as t_s, but of the first frame

t_e A timestamp at when the associated frame has been finished processing by a worker

t_{el} Same as t_e, but of the last frame

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
FAA	Federal Aviation Administration
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
OER	Overall Error Rate
SBC	Single-Board Computer
SSD	Single-Shot multibox Detector
STR	System Throughput Rate
TN	True Negative
TP	True Positive
UAV	Unmanned Aerial Vehicle

ABSTRACT

Author: Hwang, Hyun. M.S.
Institution: Purdue University
Degree Received: December 2018
Title: Low-Cost and Scalable Visual Drone Detection System Based on Distributed Convolutional Neural Network
Major Professor: Eric T. Matson

Recently, with the advancement in drone technology, more and more hobby drones are being manufactured and sold across the world. However, these drones can be repurposed for the use in illicit activities such as hostile-load delivery. At the moment there are not many systems readily available for detecting and intercepting those hostile drones. Although there is a prototype of a working drone interceptor system built by the researchers of Purdue University, the system was not ready for the general public due to its nature of proof-of-concept and the high price range of the military-grade RADAR used in the prototype. It is essential to substitute such high-cost elements with low-cost ones, to make such drone interception system affordable enough for large-scale deployment.

This study aims to provide an alternative, affordable way to substitute an expensive, high-precision RADAR system with Convolutional Neural Network based drone detection system, which can be built using multiple low-cost single board computers. The experiment will try to find the feasibility of the proposed system and will evaluate the accuracy of the drone detection in a controlled environment.

CHAPTER 1. INTRODUCTION

The primary goal of this research study is to design and implement a distributed Convolutional Neural Network (CNN) based image processing system that can detect small Unmanned Aerial Vehicles (UAV), often colloquially referred to as 'drones.' The system aims to be scalable, low-cost, equipment-neutral, and easy to build from scratch using new or existing resources; this will help any individual or organization with limited resources to adopt the system without the worry of financial burdening.

1.1 Background

Drone—defined as 'aircraft without onboard pilot'—is not a new concept; both U.S. military and hobbyists were researching and building remote-controlled aerial vehicles for nearly a half a century now [1–3]. However, the limitation of technologies at those times has made building drones a challenging task. This discouragement in building a drone has changed in a relatively recent timeframe when the technology related to these drones has advanced rapidly, enabling smaller flying objects and easier mass manufacturing of drones, therefore greatly expanding the market targetted for individuals [1, 4]. According to the Federal Aviation Administration (FAA), around thirty thousand drones will be flying in U.S. airspace at any given time by 2020 [1].

Accompanied with this market expansion, many new usages for drones appeared, including, but not limited to, use in journalism [1, 5], delivery [1, 6], agriculture [1], and exploring places that are either hard for or not meant for humans to access [1, 7]. All these new applications are indeed for aiding humans on traditionally dangerous, painful, or costly tasks, thus serving humanity as a useful tool. However, every tool has two sides of good and evil, and drones are not exception. These illicit applications span from voyeurism and privacy intrusion [1, 8], smuggling at the border or into prisons [9, 10], and to the least favored, use in hostile load delivery for domestic terrorism [11].

Public interest in these illicit uses of drones peaked when there was an incident at the White House involving a drone [12]. Fortunately, the drone in the incident was not

carrying any dangerous material and did not cause any serious threat. However, the symbolic meaning of the incidents—breach of the military-grade airspace security of one of the most densely guarded place in the world by a small toy aircraft that costs mere few hundred dollars—was enough to bring a global argument regarding the possible misuse of drones. This breach is one of the examples that show the problematic nature of the difficulty in detecting such small flying objects.

Because the drone industry is yet very new compared to the traditional air industry, there are not enough resolutions against drones, regulation-wise. Traditionally, almost every human-made flying objects were a) large, b) having pilots onboard, c) legally registered to authorities, and d) having transponders attached for identification purpose. However, drones a) are small, b) have no pilot onboard, and c) have no transponders attached for being lightweight. Although the FAA has made the registration of all drones mandatory starting 2015 [13], the only thing the regulation enforces is for all drone operators to register drones to the FAA and to mark their drones with the registration number. Of course, these markings are better than having nothing, but still, are typically hard to read with the naked eyes when the drone is in the air making the usage only to track the registrant after the drone lands or crashes into protected space. Furthermore, if the drone were carrying explosives and successfully commit self-bombing, it would be hard to track since there might be not many remains. This lack of reality in drone regulation leads to the need for proactive detection and tracking system, for which it would help on counteracting the threat in the early stage.

There is a previous study conducted by the researchers of Purdue University with the title of “Air-to-Air Counter Unmanned Aerial System (CUAS)” [14] to detect, track, and intercept a suspected-hostile drone en route to a protected object. This ‘Purdue CUAS’ study proved that drone detection, tracking, and intercepting in the air is not an impossible task; in fact, Purdue CUAS is the first ever system to perform such interception of drones successfully. However, in the accessible and affordable viewpoint, Purdue CUAS is not an ideal option to choose, as it has used a made-for-military—meaning military-grade accuracy—RADAR. It is understandable as the system needed to detect such small flying objects with high accuracy. However, military-grade RADAR is neither

affordable in cost nor freely available to the general public, hence Purdue CUAS as a counter-UAV system would be neither accessible nor affordable option; this is even stated in the research paper itself. For the same reason, Purdue CUAS is neither scalable nor easy-to-build system.

When it comes to the detection part, there are a few highly accurate solutions commercially available already. However, these solutions are for sale of dedicated hardware with paid customer support, which is not what this study aims to follow: a scalable, low-cost drone detection system with easy-to-acquire day-to-day equipment.

The proposed system will satisfy the goal mentioned above and complement the downside of Purdue CUAS, via CNN-based visual drone detection method. The system will use a live stream of a camera video feed as its source, a multiple of cheap Single-Board Computers (SBC) as the processing cluster, and finally, deliver the drone detection result to any countermeasure control system. Therefore, this system is expected to satisfy low-cost (designed to use cheap components), scalable (adding more SBC will raise the system's throughput), and easy-to-build (requires not many hardware components) requirements of the study, making it 'accessible' and 'affordable.'

1.2 Problem Statement

Traditional means of detecting an arbitrary flying object using RADAR has turned out to be a difficult task. Although it is not impossible considering commercially available solutions, the approach of using high-accuracy detector equipment raises the overall cost and difficulty of acquiring such equipment, hence making these systems neither accessible nor affordable.

1.3 Research Question

This study will build a prototype system that consists of cheap and easy to acquire components only. This study will prove that the system is a feasible alternative by answering the following question:

- Can the proposed system detect a drone visually while achieving accuracy and time constraint?

1.4 Significance

Although there are many deep-learning-based object detection study done, most of them were using equipment with high computational performance. Deep learning, mainly, relies on massive floating point arithmetic operations and for that purpose Graphics Processing Units (GPU) are typically used for both training and processing. This study, however, aims to use a cluster of equipment with low computational power, which has rarely used in conjunction with deep-learning. The significance of doing so are a) units with low computational performance are incredibly cheap compared to high-performance GPU, b) using many units at once requires parallelized algorithm which is easy to scale up, c) this can be applied to the recently trending Internet of Things applications, and d) by using units with low computational performance this study sets a baseline so the end user can easily upgrade the system with better equipment without any issue.

1.5 Assumption

- The drone that appears in front of the prototype system is assumed to be controlled by an operator with mal-intent of committing a crime.
- There is a drone interceptor system, which will track and catch the hostile drone en route to the protected object.
- There is a control system for the interceptor system, which will receive the detection result from the prototype system and act accordingly.

1.6 Limitation

- Due to the problem of acquiring a military-grade RADAR to evaluate the result of the proposed system against, the evaluation took place with a scoring system that takes the overall accuracy of the detection and the ratio of time spent on processing a single test case over the actual time spent on recording the single case into consideration.

1.7 Delimitation

- To minimize the confusing factor for the prototype system and focus on detecting drones, the experiment will occur in the controlled indoor environment.
- For the same reason as above, the neural network inside the prototype system is trained to detect a single type of drone only.
- The drone type used in this study is DJI Phantom 2.

1.8 Summary

This chapter described the problem of current drone detection, a research question based on it, the significance of the new method, assumptions, limitations, and delimitations. This study focuses on seeing the feasibility of distributed, CNN-based visual drone detection system.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

This chapter discusses previous research efforts of other researchers in RADAR-based drone detection, object recognition based on the convolutional neural network.

2.1 RADAR-Based Drone Detection

Since its debut in World War II [15], RADAR has completely changed the aspects of 'seeing' in a combat. Nowadays, many things we use or experience involve RADAR, including but not limiting to, reading cloud patterns to generate weather forecasts [16], detecting surrounding obstacles for self-driving/driver-assist technologies [17], and so on. Today, almost everything that involves aircraft utilizes RADAR. Thus, adopting RADAR in order to detect an airborne drone was an entirely rational choice. However, there are pitfalls in regards to the unique characteristics of drones made traditional RADAR a bit less effective tool for detection, but not entirely obsolete.

2.1.1 Purdue CUAS Revisit

It is worth mentioning the Purdue CUAS [14] again, as the study has provided the initial motive for this study. The primary objective of the Purdue CUAS study was to detect a drone approaching the protected object and to intercept such drone before it reaches the object. Rationally, but for much higher accuracy, Goppert et al. have used a "high-precision," "military RADAR" to detect and track an inbound target; as mentioned before, this is the most traditional and rational way of detecting flying objects in the sky yet still broadly used. The location of the experiment did not contain any obstacle that might confuse the RADAR, as the place was previously a runway for aircraft. The study concludes by successfully catching the hostile drone. However, as shown in Figure 2.1, the location of the target from the RADAR seems very dispersed, even though "the target

UAS flew in a straight line.” This result shows that even with military-grade RADAR it is still hard to reliably detect and track a small UAV in a continuous time domain.

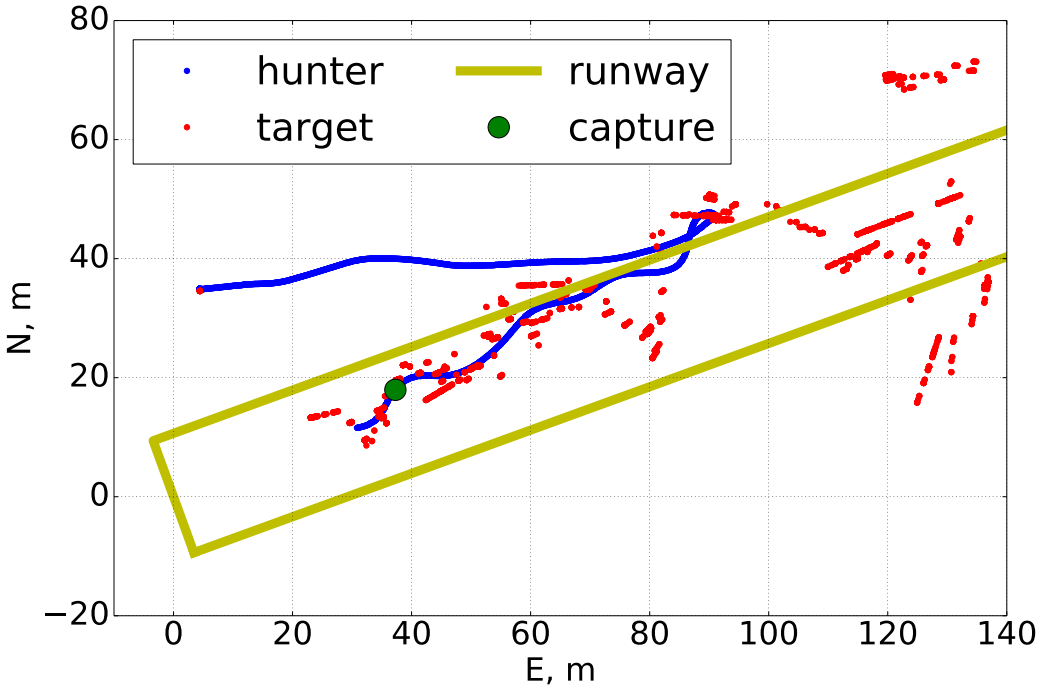

Figure 2.1. The trajectory of both hunter and the target drones, detected by RADAR [14]

2.1.2 Identifying with RADAR

One thing Purdue CUAS did not cover is the identification of the target; it is understandable if this was the delimitation of the study, but still, the system assumes the detected object is a target drone. There might be a case where a natural flying object—most notably, a bird—crosses the border of ‘No Fly Zone’ and triggers the entire intercepting mechanism. Of course, utilizing some advanced RADAR imaging techniques such as Inverse Synthetic-Aperture RADAR (ISAR) might solve the problem, but then, this will again put a strikethrough on one of the goals of this study—accessibility—since

it requires specialized RADAR hardware (e.g., phased array system) and its paired driving software. Therefore, the need for identifying flying objects still stands.

The identification of flying civilian aircraft is done with transponders. Thus, identification is not part of the RADAR itself. However, by exploiting the physical characteristics of the electromagnetic wave, getting identifiable information from a RADAR-detected object has become possible. Wit et al. [18] focused on detecting and classifying small UAVs using both continuous wave (CW) and frequency-modulated continuous wave (FMCW) RADARs by analyzing the Micro-Doppler Signature (MDS) of the target. Because typical drones reveal their rotors and blades, aside from the movement of the drone itself, there is MDS caused by the periodic rotation of blades. The authors put a focus on this characteristic of typical drones. By experimenting with a helicopter, a quadcopter, and an octocopter, the authors were able to find the type of the drone (heli-, quad-, or octocopter), the rotor diameter of the drone, and the rotation speed. However, this identification relies on the fact that the drone was stationary; when the drone maneuvering involves a complicated movement, the MDS is too minute to extract meaningful information.

Another study involving Doppler signature is by Jahangir et al. [19]. The authors, whose association is a commercial entity, developed a new RADAR system (Holographic RADAR) that can detect a small UAV up to 5 kilometers. The concept of this RADAR system is that it "stares" the entire surveillance area by using broad beam antenna as the transmitter and two-dimensional array as the receiver. Then, with all the data received from each receiver element, it classifies all detected targets by analyzing the Doppler pattern, then filters out unrelated objects such as birds. The authors have proven with real-world testing that this system can detect, classify, and track individual drones with incredible accuracy. However, the paper mentioned that this system does need substantial computational power, thus involving GPUs with specialized computing hosts for processing the input data in real-time. Impressive work indeed, but still does not fit into the low-cost category.

Thus, detecting and identifying a drone reliably—without mistakenly confusing it with birds or others—is not an impossible task, but requires either constraints or

specialized equipment, which is not what this study aims. Therefore, RADAR-based drone detection is not the way this study chose to go along.

2.2 Visual Object Recognition

Visually detecting a drone is also not a new topic. Back in Purdue CUAS, relying only on ground RADAR turned out to be inaccurate, although not entirely obsolete. To compensate such inaccuracy, Wagoner et al. [20] have proposed a system that "uses only a monocular camera on board a UAV to autonomously detect, track, and follow an unauthorized UAV for surveillance purposes." This system has three operation modes: manual, autonomous, and destroy. In the manual mode, the pilot of the "hunter UAV" flies it to near the "target UAV." Then, when switched to the autonomous mode, the hunter will detect and track the movement of the target, and follow the target by adjusting the position of itself to keep the target in the center of the perceived image from the camera. With this system, the inaccuracy of the ground RADAR can be compensated. Still, however, this does not solve the initial drone detection problem of the RADAR: it cannot distinguish some natural flying objects from drones.

2.2.1 Pitfalls of Simple Computer Vision

The aforementioned onboard monocule camera system for tracking target drone of Wagoner is based on a prior study on feature extraction of drones. Wagoner et al. [21] have conducted a preliminary experiment on figuring out the approximate distance limit of feature extraction algorithm on sUAS. Although ran in simulated, controlled environment, the feature detection algorithm has shown the performance dropping drastically when the drone is merely 2 to 3 meters away from the camera. This performance drop implies that in the real-world environment, this algorithm will not be feasible unless the drone is close to the tracking system, as in the case of [20]. This issue has also been mentioned in [20], where "noises" such as other objects within the horizon caused the tracking algorithm to lock on a wrong target (Figure 2.2). The author wrote that to solve this issue temporarily

the hunter drone had to rose until there is no horizon in its field of view anymore. This is the apparent limit of simple computer vision algorithm based drone detection: the algorithm returns merely some identifiable point of interest, not what the object actually is based on those identifiers. If the system was able to categorize objects visible to the camera, it could have pinpointed the target drone and continue chasing regardless of noise.

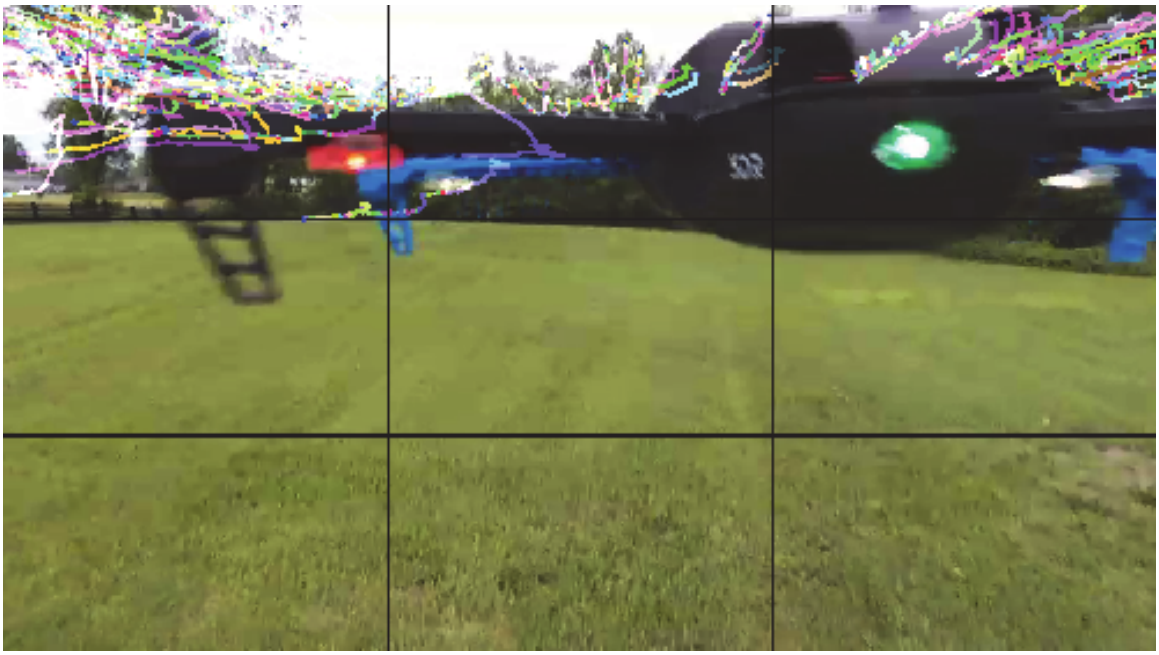

Figure 2.2. An example of noises confusing the feature extraction algorithm when the drone is near ground. [20]

Due to the above reason, there were not many aerial object detection relying on computer vision, until a revolutionary image processing and object recognition strategy came out.

2.2.2 Convolutional Neural Network

There are multiple studies done in the field of object detection using the convolutional neural network. Although the concept of CNN existed for a very long time [22–24], the actual use of CNN on categorizing arbitrary objects began recently.

One of the earliest studies on detecting and recognizing an object out of an image with a complex background is the one done by LeCun et al. [25]. In the study, the authors took large numbers of photos of small objects, with different pose and lightings, and combined with noisy backgrounds. Then these images are used to train the network, which as a result, showed at best 10.6% error rate on detecting the original object in the image.

Szarvas et al. did a study [26] to detect pedestrians with LiDAR and CNN in real-time; when LiDAR detects an obstacle, CNN is used to confirm that the obstacle is a pedestrian or not. What is essential in the study is that to lessen the burden on the computationally demanding classifier the authors have picked only the part of the image from the camera, dubbed as "region of interest" (ROI). This is where the LiDAR is involved in for picking the ROI for a possible pedestrian presence. As a result, by limiting the region for the classifier to scan, utilizing LiDAR did decrease the false positive of the system.

Karpathy et al. [27] has tried to apply CNN to videos, which then was put to the test on classifying the content of the video into one of 487 classes of sports activity. This study is significant because unlike the still images, videos have one more factor of time than images and these durations vary between each video. Therefore the authors have sliced each video into a collection of fixed clips, which then the network can easily take as an input. Because videos have time, the authors tried multiple approaches on fusing the data across the temporal domain; this enabled the motion-aware classification which increased the true positive rate later. This strategy can also be applied to this very research.

Amato et al. [28] has conducted an experiment which intends to build a lightweight CNN-based parking lot vacancy detector in a distributed manner. A node consists of a Raspberry Pi SBC and a camera, and the pre-trained CNN classifier runs on this SBC itself; thus lightweight and distributed. The authors have tested two distinct CNN architecture and compared the performance. Even though the classification job ran on SBC, the authors stated that the load average was 0.25, meaning that for that time duration the system was 75% idle and 25% busy. For that reason, this system can be said lightweight and can be a role model for implementing what this very study is trying to

build. However, it should be noted that this system took 15 seconds to scan the entire lot and detect parked cars.

2.2.3 Detecting Drone with CNN

As suggested before, CNN is neither an entirely new topic nor has not been used in drone detection, although drone detection with CNN is undoubtedly a rare topic; most of the previous studies are focused on using CNN with drones—analyzing images taken from drone-attached camera—not using CNN on drones—analyzing images of drones.

Although not applying directly on the image of drones, Kim et al. [29] have tried detecting drones by feeding the MDS spectrum and Cadence-Velocity Diagram (CVD), concatenated into a single Merged-Doppler Image (MDI), into CNN. In this study, nearly 60,000 images of two sample drones seen by an FMCW RADAR in an anechoic chamber and outdoor environment were used to train the network. The reason why the researchers used an MDI instead of MDS or CVD individually is that upon their preliminary testing both images have shown higher accuracy in some cases and lower cases in some other cases, respectively; hence, merging both into one MDI has made the network to yield the highest accuracy in any given cases. The result is spectacular, as the MDI analysis has yielded 94.1% accuracy in anechoic chamber and 100% accuracy when tested outdoor. The only downside of this study is that the researchers have focused on drones hovering in one place, not moving. As discussed earlier, this is due to the characteristics of MDS which requires additional shift-interpolation in case of moving targets, where the MDS signature of rotating blades can be easily buried under the shift caused by the movement of the drone itself.

The study by Aker and Kalkan [30] is by far the most interesting study on CNN-based drone detection, with the raw images of drones. For the experiment, the authors generated an artificial dataset by combining real images of drones and birds randomly on real videos of coastal areas. After training the network with both the artificial dataset and the real dataset from ImageNet [31], the network was put to the test, which showed around 90% accuracy. Therefore, this study has proven that artificially generated

dataset is good enough to be used on training and that CNN is an excellent approach to detect drones. However, this study has applied the neural network on still images and did not cover applying the network to videos; our study had to tackle this problem down by ourselves.

2.3 Summary

This chapter provided the review of existing research studies with the subject of traditional RADAR-based drone detection, general applications of both vision-based object recognition and convolutional neural network.

CHAPTER 3. METHODOLOGY

The goal of the proposed system is to detect a drone visually using CNN on a cluster of SBCs. Individual SBC has low computational performance, hence the delay that occurs while processing each frame of the video stream will cause the system to lag behind indefinitely. This study aims to resolve such issue by parallelizing the process: distributing each frame of the video stream across multiple SBCs. Thus, the study is to evaluate two aspects of the system: accuracy and throughput.

3.1 Proposed System Overview

The prototype system is built with many SBCs connected with each other via a network switch. A single SBC becomes a 'node.' One of these nodes has the role of 'master' which coordinates and distributes tasks to other nodes. The rest of the nodes get the role of 'worker' which repeats the following cycle: receive the task assigned by the master, perform the task, and report the result back to the master. The master node also holds the responsibility of capturing the visual circumstance around the system; hence the only available camera module is attached to the master. Figure 3.1 depicts the system.

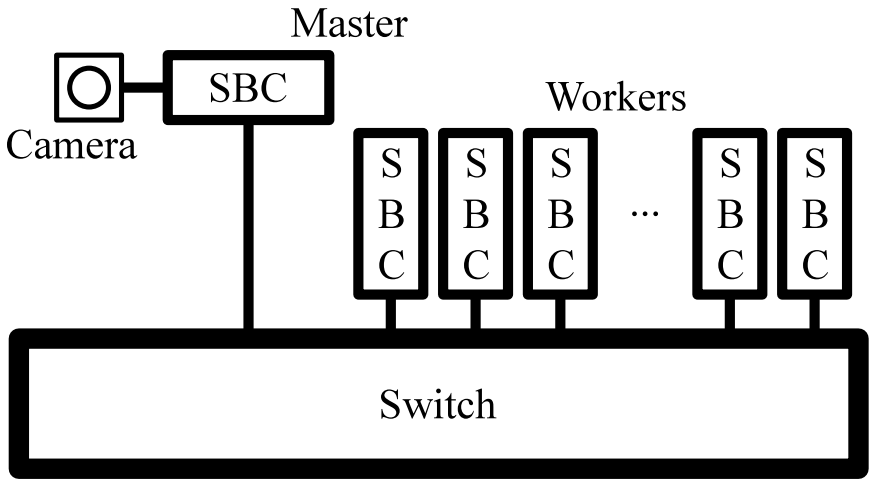

Figure 3.1. System overview

3.1.1 Hardware

The prototype system used in the experiment is built with eight SBCs, one switch, and one camera module. Although this experiment has only used seven worker nodes, this is enough setup to observe scalability, thanks to the initial design that considered distributed running environment. All of the equipment used in the experiment is generally accepted to fall under the category of 'cheap' and 'easy to acquire,' still satisfying the goal of this study. Especially, in general, a large userbase of certain equipment means a vast knowledge base exists for easy troubleshooting and guidance. It is a loss indeed not to use such 'knowledge base,' so in case there are multiple options available for certain equipment the popularity in the market is used to decide which to use. The specifications of all equipment are as follows:

- SBC: Raspberry Pi 3 Model B v1.2 [32] is chosen for its versatility, ease of acquiring one, and cheap; it costs mere $35. For the operating system on this SBC, Raspbian Stretch is chosen as it is Raspberry Pi Foundation's official OS and comes with the Foundation's full support.

- Camera module: the standard Raspberry Pi Camera Module v2 [33] is chosen for its decent performance of 1080p30, meaning it can capture full HD video at 30 frames-per-second. The camera has 54° wide view angle and sees at most n meters wide at n meters away from the camera. Figure 3.2 depicts this.

- Networking: To connect all nodes with minimum bottleneck, D-Link DGS-1024A 24-Port Unmanaged Gigabit Switch [34] is chosen.

- Drone: DJI Phantom 2 [35] is chosen, for its dominant market share [14].

3.1.2 Software

Initially, this study aimed to utilize a framework named 'Darknet' [36] and the model of 'YOLOv3' [37] for maximum performance; Darknet is a neural network

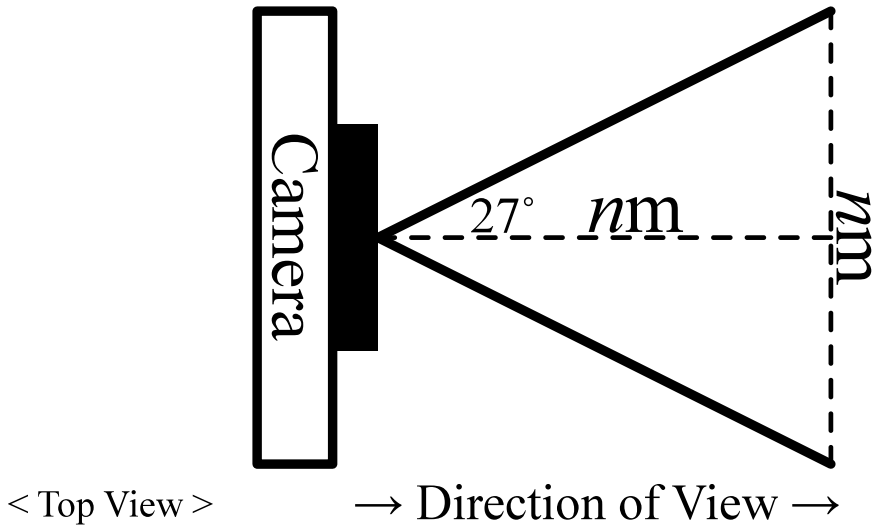

Figure 3.2. Camera overview

implementation in the C language; thus it was expected to provide high performance even on an SBC. However, this turned out not a good choice, as a preliminary test that ran on a standard laptop showed Darknet's poor performance against TensorFlow [38]. Even with decent processing power, it was underachieving, so Darknet is undoubtedly not a good fit for running on an SBC.

TensorFlow is undoubtedly the most popular framework for machine learning; therefore the same notion of 'knowledge base' mentioned above applies to this, too. TensorFlow has been extensively researched and utilized in many studies and projects, it also provides multiple models suitable for various purposes out of the box [39]. In this study, considering the special situation of running a neural network on devices with low computational performance without GPU, a model optimized for these kinds of devices is chosen: SSD [40] + MobileNetV2 [41–43].

Utilizing TensorFlow with SSD + MobileNetV2, the software stack on each of master and worker node is written from scratch. The master node is in charge of capturing the visual circumstance of the system; thus it runs a process 'FrameFetcher' that fetches a frame from the camera and enqueues it into an internal buffer. Then, whenever there is an idle worker, a process 'FrameDistributor' dequeues an unprocessed frame from the queue and hands over to the worker. The worker runs the frame through the neural network

which emits an annotated frame. The worker reports this frame back to the master, which then the master saves it into another queue for later analysis. These steps are repeated until the user turns the device off. Figure 3.3 depicts the internals of the software stack of the master and the worker, and the flow of information between them.

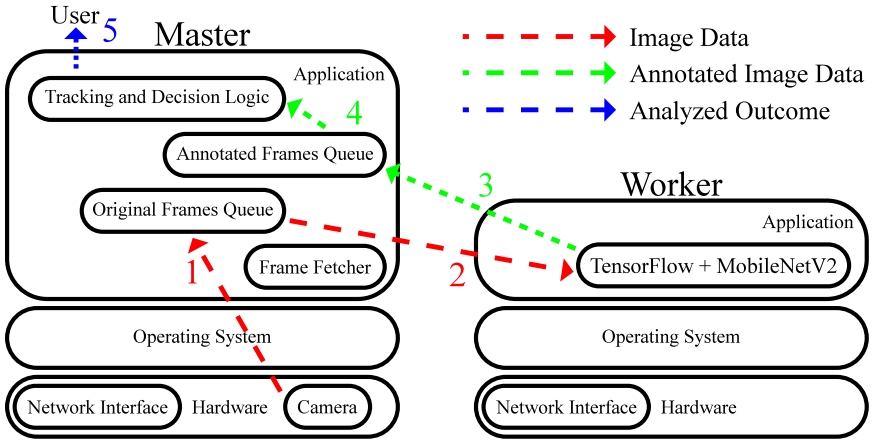

Figure 3.3. Overview of master and worker software stack, with information flow between them

3.2 Training CNN

The TensorFlow Model Zoo provides pre-trained inference graphs, but none of them are trained to detect a drone, especially the Phantom 2. Thus, training a new detector from the ground was a must, which then requires a training dataset with ground truth labeled. Generally, training requires a massive dataset, but following what Aker et al. has done in [30], with additional images taken directly for this study, a set of 700 images with ground truth labels has been made.

Training a neural network requires much great computational performance than actually using the network, so training itself was performed on a dedicated computation instance on Amazon AWS. The training has been done for 32 hours on 700 images explained above. Then the inference graph is exported frozen, distributed across the workers; thus all the workers run the same neural network graph.

3.3 Data Collection

There are two criteria this study is evaluating: the scalability in forms of system throughput and the accuracy.

A fully functional system was set to operate in an indoor environment. Then, a drone has flown in front of the camera multiple times, with different distance between the drone and the camera and with a different number of workers participating in the processing. Figure 3.4 depicts the experiment setting.

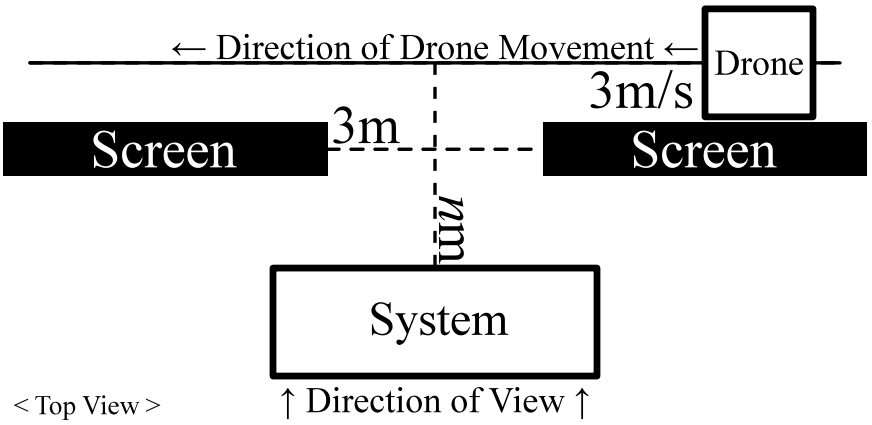

Figure 3.4. Experiment setup overview.

For measuring the system throughput, the master marked the timestamp in nanosecond resolution on each frame upon capturing each frame from the camera. Workers also marked the timestamp before and after processing the frame.

At the end of each trial, the following data were calculated for later analysis:

- System Throughput Rate (STR): as mentioned above, each frame holds three timestamps: when the master captured the frame (t_c), when the worker received the frame (t_s), and when the worker finished processing the frame (t_e). It is expected that the duration of the processing a single frame ($t_e - t_s$) is uniform while the purpose of the experiment is to measure the overall throughput of the system when the number of workers changes. Therefore, the Formula 3.1 is used to measure the throughput STR of the trial.

- Accuracy of detection: each processed frame falls into one of the four categories: each processed frame falls into one of the four categories described in Table 3.1. The accuracy was calculated with Formula 3.2, proposed by [44] as Overall Error Rate (OER).

- The number of workers: with a different number of workers, the throughput varied. This value is used to categorize results.

- The distance between the camera and the drone: The distance affected the accuracy. This value is used to classify results.

Table 3.1. Category table of the processed frame

	System Reports Drone	**System Reports NO Drone**
There IS Drone	True Positive	False Negative
There is NO Drone	False Positive	True Negative

$$\text{STR} = \frac{\text{Process Duration} = t_{el} - t_{sf}}{\text{Trial Duration} = t_{cl} - t_{cf}} \tag{3.1}$$

$$\text{OER} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3.2}$$

3.3.1 Evaluation

Among the collected data, both the STR and the OER share the same trait: the lower, the better. Therefore, as a way of evaluating the set of the distance and the worker count, the OER will be multiplied by the STR, yielding the Final Evaluation Score (FES). For example, if the STR was 7 and the OER was 15%, then the FES will be 1.05 (Equation 3.3), and if the STR was 15 and the OER was 40%, then the FES will be 6 (Equation 3.4).

$$\frac{15}{100} \times 7 = \frac{105}{100} = 1.05 \tag{3.3}$$

$$\frac{40}{100} \times 15 = \frac{600}{100} = 6 \tag{3.4}$$

When designing the experiment, an ideal threshold of the processing time STR 5 and maximum 20% OER is expected in the experiment. As this study aims to prove that the scalability is an essential factor to consider building this distributed neural network system, the configuration that shows the final score less than 1 will be selected as the most favored result.

3.4 Experiment

For each distance between the camera and the drone, and for each number of workers, a trial of flying the drone in front of the camera was conducted ten times. Then for each trial, the final score is calculated. Table 3.2 lists all of the possible distance-count pairs. As mentioned in Chapter 1, the experiment was conducted inside a controlled indoor environment to exclude any possible source of confusion on the system.

Table 3.2. List of all experiment cases

Case #:	1	2	3	4	5	6	7	8	9	10	11
Distance (m)	2	2	2	2	4	4	4	4	6	6	6
Worker Count	1	3	5	7	1	3	5	7	1	3	5

Case #:	12	13	14	15	16	17	18	19	20
Distance (m)	6	8	8	8	8	10	10	10	10
Worker Count	7	1	3	5	7	1	3	5	7

There was a slight problem on when the distance was 2 meters: because when on 2 meters away from the camera, the camera only sees 2 meters wide horizontally, making the drone exposure shorter than the rest of the cases. However, since the experiment does not use absolute values directly but calculate the relative ratio out of those values, this has caused no problem.

3.5 Summary

This chapter discussed the design of the experiment on evaluating the proposed system.

CHAPTER 4. ANALYSIS OF RESULT

This chapter lists all the data captured from entire experiment and discuss the analysis.

4.1 Observing Result

Each worker yields annotated frame image data as its output. Each trial yielded 150 frames in average and there were 200 trials, thus the experiment in whole has generated near 30,000 annotated frame images. All the images were analyzed manually for the existence and location of drone and detection bounding boxes, then the counts were organized into tables. Due to the extensive amount, all the data are attached at the end of this chapter as Table 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22 and 4.23. Entries in STR, OER, and FES were underlined if they satisfied the predefined threshold. Additional column legends are as follows:

- TD: the duration of that single trial in seconds, calculated $t_{cl} - t_{cf}$, where t_{cf} is always 0.
- PD: the duration of processing that single trial in seconds, calculated $t_{el} - t_{sf}$.

Although the analysis was primarily checking the correctness of detection bounding boxes' location, detection itself reports its level of confidence score with a bounding box. For example, Figure 4.1 shows a detection result with very high score, while Figure 4.2 shows very low score. In this study, both results were counted as True Positive as the CNN has put correct bounding box on drones. For comparison, Figure 4.3 shows an example of False Positive. However, seeing that most of the images has captured a drone as a ghost-like image, a question of 'if, given without prior context, would a human being recognize this image as a photo of a drone?' has risen. This question will be dealt later again.

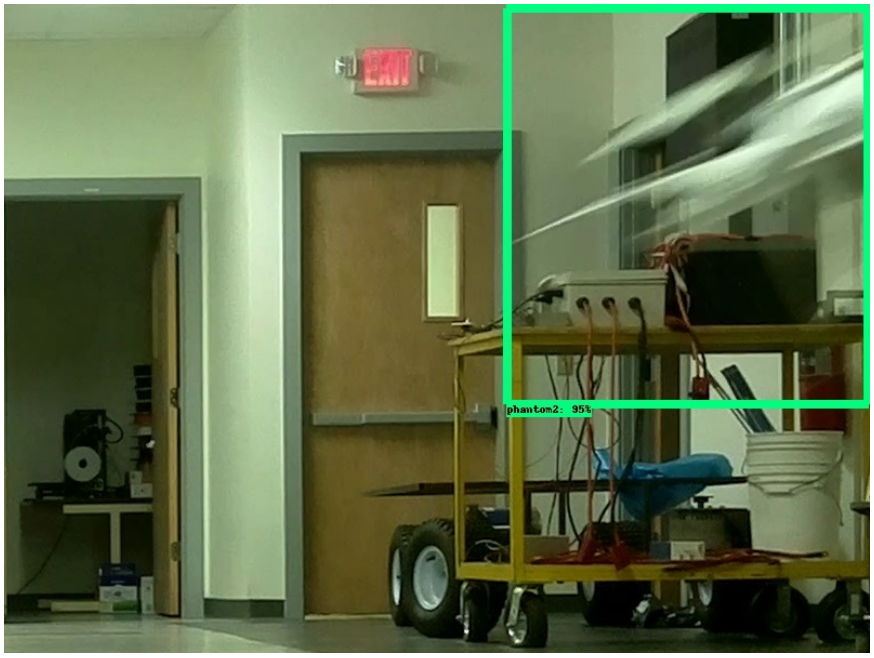

Figure 4.1. A drone was flying 4m away from the camera at the time of the capture. Analysis by the neural network reported that it is 95% sure that the object in the bounding box is a drone.

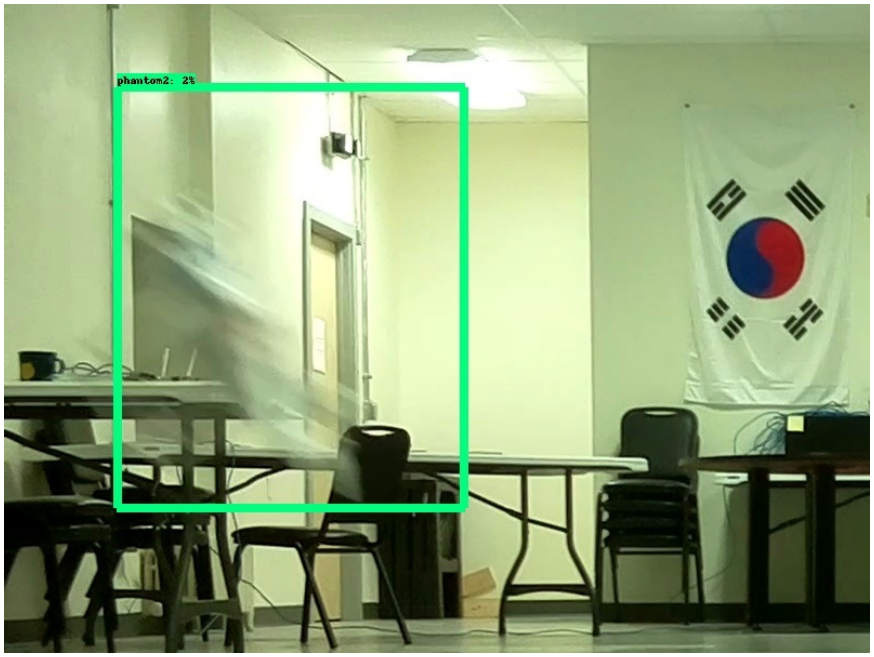

Figure 4.2. A drone was flying 6m away from the camera. Analysis reported that it is only 2% sure that the object is a drone.

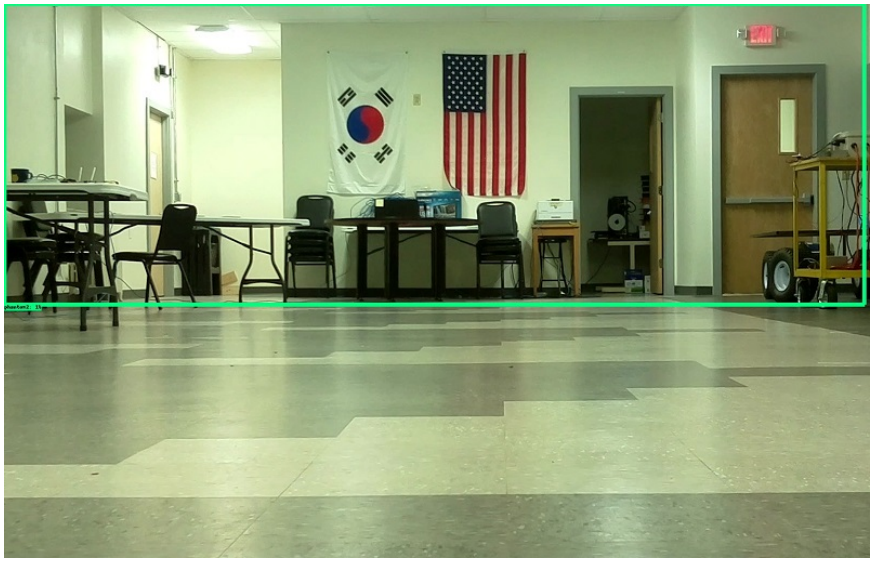

Figure 4.3. A drone was flying 4m away from the camera at the time of the capture. Analysis by the neural network reported that it is 1% sure that there is a drone inside the bounding box, which in reality, there was nothing.

4.2 Analysis

An example analysis of a single trial can be as follows: (trial #3 of Case #7, Table 4.10)

- The trial began with the start command, and the system declared timestamp 0.
- The trial was conducted for 29.407 seconds. Capturing stopped immediately following the stop command, but the processing continued, as intended.
- The neural network received and started processing the first frame 0.186 seconds after its capture.
- The neural network finished processing the last frame 64.272 seconds after the start command.
- From above two timestamps, the entire processing duration is calculated: 64.086 seconds.

- There were 168 frames.
  - 49 frames captured the drone and the neural network detected the drone correctly (True Positive)
  - 86 frames captured nothing and the neural network reported there is nothing (True Negative)
  - 1 frame captured nothing but the neural network detected the drone is there (False Positive)
  - 32 frames captured the drone but the neural network reported there is nothing (False Negative)
- The throughput is calculated $64.086/29.407 = 2.179$, meaning the neural network is twice slower in processing than the system capturing.
- Out of 168 total frames, the neural network reported wrongly in $1 + 32 = 33$ frames; the error rate is calculated $33/168 = 0.196$.
- FES is calculated $2.179 \cdot 0.196 = 0.428$. This satisfies the predefined threshold of 1.

All 200 trials followed the same step as above example analysis. The following are about evaluations conducted based on the outcome of the analysis.

4.2.1 System Scalability

For measuring whether the STR decreases as more workers are involved, for each case the arithmetic mean of STR of all 10 trials were calculated. This mean is shown in Table 4.1, from which a graph with distance as the category has been plotted in Figure 4.4. Because the neural network yields results in fixed amount of time (due to its internal design), the result of scalability was expected not to be affected by the distance between the drone and the camera and was observed not; there were no significant differences between distances under same number of workers.

With only 1 worker, the processing required ten times of the duration of the capturing, as frames are queued up in real time but the processing is being done at much

slower rate. Thus, when added 2 more workers (3 workers), the processing is distributed among workers, requiring 3 times of the duration of the capturing. When added 2 more workers again (5 workers), the processing required twice the duration of capturing. Finally, with 7 workers, processing required near-equal of the duration of the capturing, which is a more-than-expected result.

Another aspect of this result to look into is the inclination of the slopes between worker counts in the graph. A steep negative slope is observed between 1 worker and 3 workers, meaning there were drastic performance boost by just adding 2 more workers. Following that, steadier negative slope connects 3 workers and 5 workers, meaning there still is performance boost by adding 2 more workers but not as drastic as the previous. Same alignment continues with more steadier slope between 5 workers and 7 workers. It is clear that adding more worker nodes decrease STR, thus proving that this system is designed to be scalable and is confirmed scalable. As STR goes near 1, it can be said that the system is processing frames in near real time. However, as the slope inclination changes show, the performance boost per worker node addition is decreasing, meaning there will be a threshold where worker node cost will exceed the profit gained from the performance boost. It is considered inevitable with the current multiprocessing design selected for this study, as there are resources being wasted on performing transactions with shared data structured.

Table 4.1. Mean STR of each case. Values with underline indicates they satisfy predefined threshold.

Distance	**1 Worker**	**3 Workers**	**5 Workers**	**7 Workers**
2m	10.217	3.508	2.138	1.528
4m	10.175	3.592	2.203	1.595
6m	10.332	3.553	2.188	1.571
8m	10.283	3.456	2.125	1.504
10m	10.194	3.672	2.278	1.715

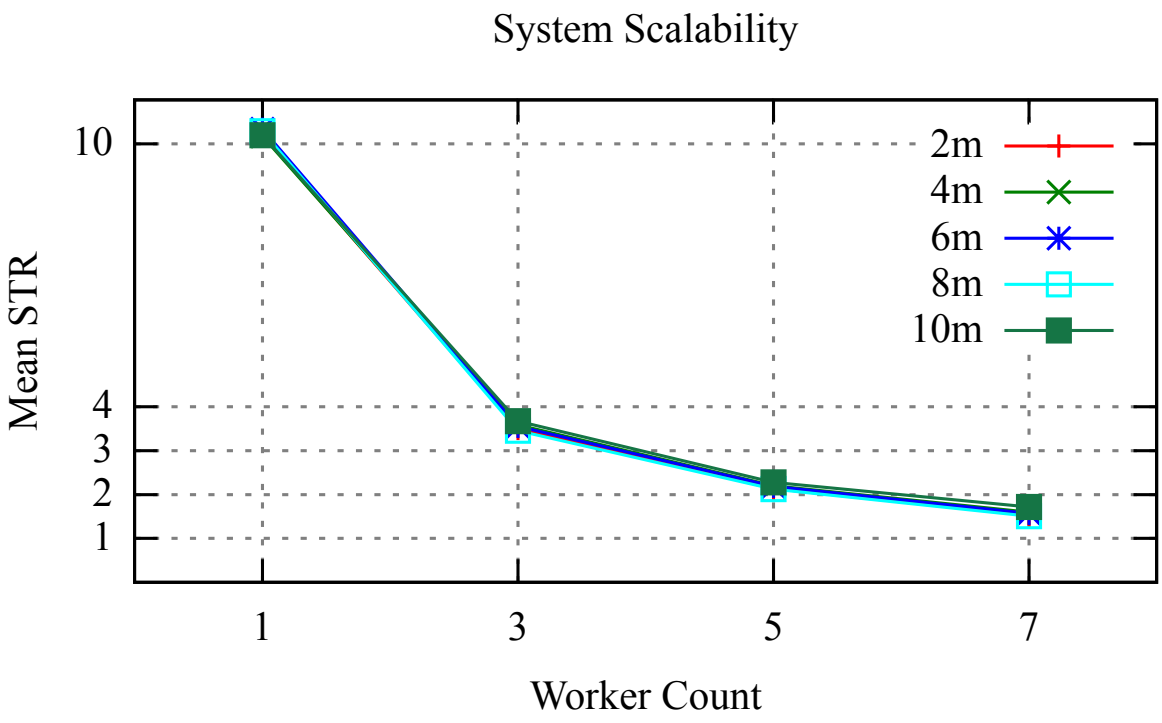

Figure 4.4. Scalability graph showing mean STR on different worker counts. With more worker, the performance boost can be easily seen, confirming that the system is scalable.

4.2.2 System Accuracy

Similar to STR, for each case the arithmetic mean of OER of all 10 trials were calculated. Table 4.2 and Figure 4.5 represent those means. Because all the worker nodes used in the experiment were sharing very same inference graph, the accuracy was expected not to be affected by the number of workers, but was observed so; the accuracy was affected by the number of workers, and the reason behind was not determined.

According to the data, while the system works as designed, there is an accuracy issue with the system, as only with 2m distance between the camera and the drone the system was able to achieve less than 20% OER. The rest of the cases, as the distance increases, the OER increased as well. However, this is not related to the core concept of the system but to the amount of the dataset that was used to train the neural network and the quality of image from the camera. Although this study has used twice the image of what Aker et al. have used in [30], due to the unique nature of ‘video’ rather than ‘still

image,' the system required more training data, as in 'video' frames a moving object typically has blurry edges, which then might have confused the neural network. Additionally, as the distance between the drone and the camera increased, due to the low quality of images generated by low-cost camera module, the 'outline' of the drone that is used by the neural network might have become less sharp, thus confusing its ability to catch edges of the drone and detecting the drone. This issue can be resolved by training the neural network further with more dataset, but at this moment this is very time consuming task as there is no ready-made dataset of drones—either holding still or moving—available, so making it is the only way as of now. Also, by replacing the camera with the one of more quality, the outline issue might be resolved, too; but doing so will again violate the 'affordable' aspect of this study. Thus, this system can be said 'fairly accurate, but only within close range; with plenty of room for improvement.'

Table 4.2. Mean OER of each case. Values underlined are ones within the predefined threshold.

Distance	**1 Worker**	**3 Workers**	**5 Workers**	**7 Workers**
2m	0.214	0.160	0.130	0.140
4m	0.616	0.506	0.413	0.497
6m	0.985	0.838	0.757	0.785
8m	1.000	0.988	0.985	0.989
10m	1.000	0.875	0.809	0.836

4.2.3 System Feasibility

As defined in previous chapter, both STR and OER are better if they are lower. From this, as an easy-to-grasp scale, FES is defined to be calculated from multiplying STR and OER. With both STR and OER obtained from the study, FES was retrieved per the equation, then again, means of each individual FES were taken. Table 4.3 and Figure 4.6 depicts these. Observing the data, the following cases showed the tendency of eligible system: Case #2 (2m + 3 workers), #3 (2m + 5 workers), #4 (2m + 7 workers), #7 (4m + 5 workers), #8 (4m + 7 workers).

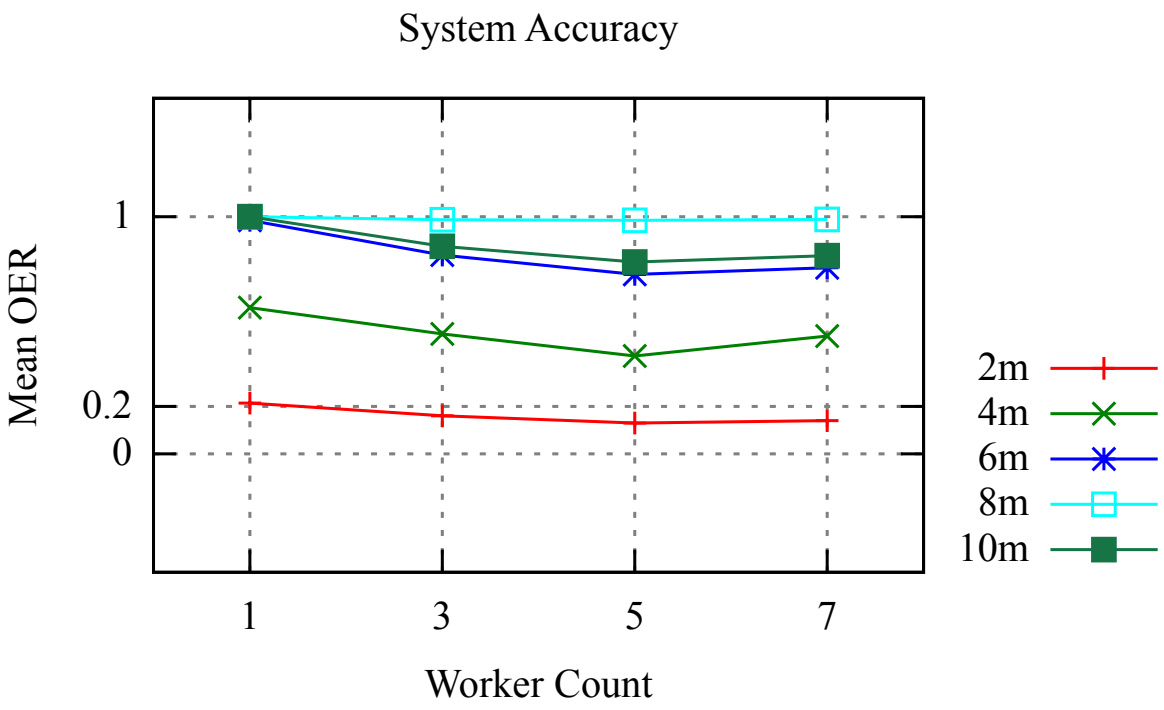

Figure 4.5. Accuracy graph showing mean OER on different worker counts. With increasing distances, the error rate has risen considerably.

Due to the nature of multiplication, STR and OER were given the relationship of mutual complementation; even if one value did not satisfy its own threshold, the other value has chance to complement that. Thus, there were cases that either STR or OER were over the threshold but the other value compensated to go under the FES threshold. Case #7 (4m + 5 workers) and #8 (4m + 7 workers) are those; although the OER was out of acceptable range, STR was low enough to drag the FES down. Training the neural network further to lower OER and adding more workers to lower STR will significantly lower FES further, making the system more reliable and more responsive.

Table 4.3. Mean FES of each case. Values underlined are ones under predefined threshold.

Distance	1 Worker	3 Workers	5 Workers	7 Workers
2m	2.191	0.563	0.280	0.216
4m	6.277	1.817	0.909	0.796
6m	10.174	2.974	1.650	1.228
8m	10.283	3.416	2.093	1.488
10m	10.194	3.212	1.837	1.433

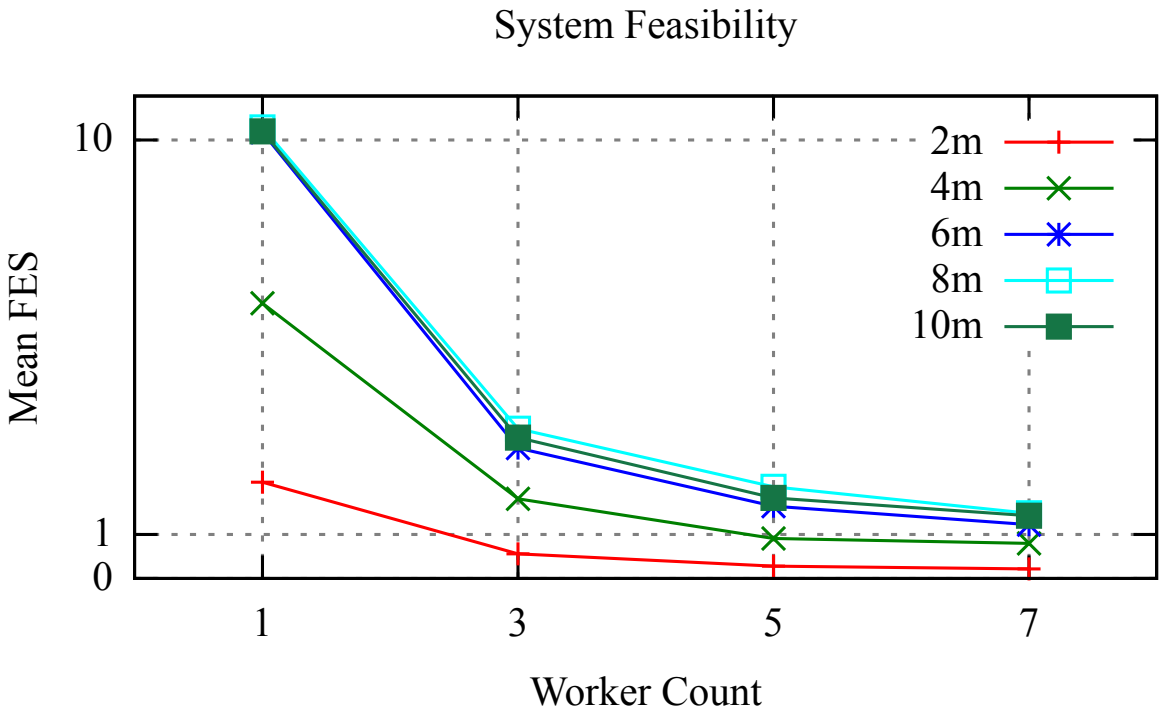

Figure 4.6. Feasibility graph showing mean FES on different worker counts

4.3 Additional Consideration

There were few issues appeared during the analysis.

4.3.1 Blurry Image of Drones

Due to the limited image quality produced by the camera module and the low performance of the master SBC, almost every image of drones appeared ‘blurry’ and

‘ghost-like.’ To tackle this problem, the initial training data set was also produced with exactly same camera module. Therefore, the inference graph used in this study knows that those blurry images are drones. However, supposing that a human being is given this frame images without any previous knowledge, it is uncertain that the human being would recognize this as a drone. Similarly, any white object moving fast can also generate similar-looking image. This issue can be easily resolved by replacing the camera module with a better working camera module, but it would raise the cost of the system, which is the violation of affordability. Right now, the only feasible resolution for this issue is to train the CNN further to see if the neural network can recognize it. This leads to the next issue, which is also linked to the maturity of the neural network.

4.3.2 Low Detection Score

Although in feasibility analysis we have concluded that 5 cases turned out to be feasible, two cases of 4m distance are unique as they have failed on OER threshold but passed FES threshold eventually. Combined with low detection scores observed from outputs, this is highly suspected to be caused by the immaturity of the CNN. Usually, a CNN requires massive data set to function as expected; however, the CNN used in this study has used mere 700 GT images, per [30]. High OER on farther distances suggests that the CNN has neither trained enough with size differences yet nor with various backgrounds, which leads back to the size of the training set. This problem of high OER is expected to be resolved with much larger training GT image set, and by knowing this possible resolution this study can be improved in the future.

Table 4.4. Data from Case #1: 2m distance & 1 worker

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	48.440	48.440	0.233	502.228	501.995	14	253	0	24	10.363	0.082	0.855
2	35.199	35.199	0.183	357.831	357.648	10	169	0	28	10.161	0.135	1.374
3	27.907	27.907	0.197	287.831	287.633	2	124	0	41	10.307	0.246	2.530
4	32.994	32.994	0.171	313.061	312.890	7	135	0	40	9.483	0.220	2.084
5	22.377	22.377	0.260	237.909	237.649	3	101	0	33	10.620	0.241	2.558
6	29.967	29.967	0.224	308.703	308.479	2	146	0	31	10.294	0.173	1.783
7	30.020	30.020	0.203	305.874	305.670	6	137	0	34	10.182	0.192	1.956
8	25.019	25.019	0.197	265.002	264.805	6	95	0	52	10.584	0.340	3.597
9	25.914	25.914	0.230	267.917	267.687	11	96	0	47	10.330	0.305	3.153
10	32.548	32.548	0.225	320.676	320.451	9	138	0	38	9.846	0.205	2.022

Table 4.5. Data from Case #2: 2m distance & 3 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	52.417	52.417	0.221	172.272	172.051	23	252	1	15	3.282	0.055	0.180
2	35.328	35.328	0.210	122.801	122.591	17	167	1	22	3.470	0.111	0.386
3	30.173	30.173	0.216	100.551	100.335	13	123	0	31	3.325	0.186	0.617
4	30.532	30.532	0.205	110.111	109.906	17	136	0	29	3.600	0.159	0.574
5	23.612	23.612	0.221	83.858	83.637	11	99	0	27	3.542	0.197	0.698
6	30.314	30.314	0.277	107.908	107.631	5	157	0	17	3.550	0.095	0.337
7	30.206	30.206	0.216	106.373	106.157	12	137	1	27	3.515	0.158	0.556
8	25.687	25.687	0.188	92.584	92.396	15	95	0	43	3.597	0.281	1.011
9	26.154	26.154	0.227	93.307	93.079	25	96	0	33	3.559	0.214	0.763
10	30.659	30.659	0.188	111.827	111.638	20	139	0	26	3.641	0.141	0.512

Table 4.6. Data from Case #3: 2m distance & 5 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	51.414	51.414	0.270	107.110	106.840	27	252	0	12	2.078	0.041	0.086
2	35.773	35.773	0.218	77.190	76.972	24	168	1	14	2.152	0.072	0.156
3	30.137	30.137	0.248	63.311	63.063	17	125	0	25	2.093	0.150	0.313
4	31.559	31.559	0.211	69.026	68.815	21	138	0	23	2.180	0.126	0.276
5	23.748	23.748	0.231	53.592	53.361	13	101	0	23	2.247	0.168	0.377
6	32.220	32.220	0.227	67.070	66.842	6	153	0	20	2.075	0.112	0.232
7	30.535	30.535	0.268	67.088	66.820	17	138	1	21	2.188	0.124	0.272
8	26.630	26.630	0.266	58.316	58.050	20	96	0	37	2.180	0.242	0.527
9	26.521	26.521	0.224	58.969	58.745	35	97	1	21	2.215	0.143	0.316
10	35.146	35.146	0.238	69.442	69.203	24	138	0	23	1.969	0.124	0.245

Table 4.7. Data from Case #4: 2m distance & 7 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	56.804	56.804	0.207	77.760	77.553	26	251	1	13	1.365	0.048	0.066
2	37.381	37.381	0.278	56.918	56.640	20	168	0	19	1.515	0.092	0.139
3	29.680	29.680	0.251	47.339	47.088	18	124	0	25	1.587	0.150	0.238
4	34.687	34.687	0.217	50.621	50.404	20	138	0	24	1.453	0.132	0.192
5	23.845	23.845	0.261	39.954	39.692	13	101	0	23	1.665	0.168	0.279
6	33.380	33.380	0.181	49.582	49.401	6	154	0	19	1.480	0.106	0.157
7	33.028	33.028	0.235	49.748	49.513	16	139	0	22	1.499	0.124	0.186
8	27.918	27.918	0.205	43.492	43.287	18	94	0	41	1.551	0.268	0.415
9	27.494	27.494	0.198	44.197	43.999	32	96	1	25	1.600	0.169	0.270
10	32.739	32.739	0.221	51.573	51.352	21	137	0	26	1.569	0.141	0.222

Table 4.8. Data from Case #5: 4m distance & 1 worker

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	64.128	64.128	0.194	645.551	645.356	34	0	0	341	10.064	0.909	9.151
2	29.550	29.550	0.192	302.562	302.369	5	46	0	124	10.233	0.709	7.250
3	28.384	28.384	0.176	293.516	293.339	20	87	0	61	10.335	0.363	3.752
4	21.281	21.281	0.218	224.916	224.697	7	28	0	94	10.558	0.729	7.694
5	28.914	28.914	0.258	298.346	298.088	9	64	0	99	10.309	0.576	5.934
6	22.303	22.303	0.173	236.803	236.630	24	26	0	85	10.610	0.630	6.680
7	22.143	22.143	0.267	228.156	227.889	7	32	0	92	10.292	0.702	7.228
8	28.149	28.149	0.174	293.708	293.534	35	42	0	91	10.428	0.542	5.648
9	30.705	30.705	0.184	311.167	310.983	17	73	0	90	10.128	0.500	5.064
10	31.142	31.142	0.210	273.990	273.780	6	74	0	79	8.791	0.497	4.368

Table 4.9. Data from Case #6: 4m distance & 3 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	65.420	65.420	0.197	223.246	223.049	118	6	5	246	3.410	0.669	2.282
2	29.721	29.721	0.246	106.116	105.870	29	48	7	91	3.562	0.560	1.995
3	28.665	28.665	0.206	102.548	102.342	38	74	1	55	3.570	0.333	1.190
4	21.788	21.788	0.251	80.599	80.348	31	13	3	82	3.688	0.659	2.430
5	29.201	29.201	0.209	104.489	104.280	35	59	2	76	3.571	0.453	1.619
6	22.941	22.941	0.207	84.888	84.681	46	22	3	64	3.691	0.496	1.832
7	22.308	22.308	0.282	81.300	81.018	25	13	3	90	3.632	0.710	2.578
8	28.121	28.121	0.235	104.386	104.151	63	44	1	60	3.704	0.363	1.345
9	30.385	30.385	0.208	109.335	109.127	38	68	0	74	3.592	0.411	1.477
10	27.496	27.496	0.265	96.529	96.264	29	65	5	59	3.501	0.405	1.418

Table 4.10. Data from Case #7: 4m distance & 5 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	67.259	67.259	0.198	138.172	137.974	156	6	8	205	2.051	0.568	1.165
2	30.219	30.219	0.246	66.830	66.584	43	48	8	76	2.203	0.480	1.058
3	29.407	29.407	0.186	64.272	64.086	49	86	1	32	2.179	0.196	0.428
4	22.477	22.477	0.202	51.212	51.010	44	9	4	72	2.269	0.589	1.337
5	29.152	29.152	0.201	66.417	66.215	46	59	8	59	2.271	0.390	0.885
6	23.405	23.405	0.229	54.139	53.910	65	20	4	46	2.303	0.370	0.853
7	23.671	23.671	0.261	52.523	52.262	36	29	4	62	2.208	0.504	1.112
8	30.044	30.044	0.188	65.965	65.777	72	44	4	48	2.189	0.310	0.678
9	31.129	31.129	0.178	68.527	68.349	49	71	1	59	2.196	0.333	0.732
10	28.005	28.005	0.269	60.661	60.392	38	58	7	55	2.156	0.392	0.846

Table 4.11. Data from Case #8: 4m distance & 7 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	73.326	73.326	0.203	100.645	100.443	136	6	8	225	1.370	0.621	0.851
2	29.988	29.988	0.232	50.072	49.840	36	48	9	82	1.662	0.520	0.864
3	34.958	34.958	0.202	48.192	47.990	47	76	1	44	1.373	0.268	0.368
4	22.807	22.807	0.207	38.565	38.357	36	0	3	90	1.682	0.721	1.212
5	29.130	29.130	0.268	49.784	49.516	41	56	6	69	1.700	0.436	0.741
6	24.189	24.189	0.268	40.133	39.865	57	0	4	74	1.648	0.578	0.952
7	23.640	23.640	0.215	38.934	38.719	32	15	4	80	1.638	0.641	1.050
8	29.898	29.898	0.220	48.797	48.576	68	42	4	54	1.625	0.345	0.561
9	31.791	31.791	0.235	50.872	50.637	41	74	1	64	1.593	0.361	0.575
10	27.294	27.294	0.215	45.541	45.326	32	51	6	69	1.661	0.475	0.788

Table 4.12. Data from Case #9: 6m distance & 1 worker

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	23.870	23.870	0.191	252.806	252.615	10	0	0	135	10.583	0.931	9.853
2	27.091	27.091	0.222	276.831	276.609	0	0	0	160	10.211	1.000	10.211
3	63.035	63.035	0.197	628.302	628.105	0	0	0	366	9.964	1.000	9.964
4	25.617	25.617	0.190	253.380	253.190	1	0	0	145	9.884	0.993	9.816
5	22.550	22.550	0.222	236.303	236.082	3	0	0	133	10.469	0.978	10.238
6	26.862	26.862	0.198	277.425	277.227	0	0	0	160	10.321	1.000	10.321
7	34.861	34.861	0.184	361.170	360.986	2	0	0	209	10.355	0.991	10.257
8	19.710	19.710	0.206	209.417	209.211	1	0	0	119	10.614	0.992	10.526
9	22.668	22.668	0.258	235.470	235.211	2	0	0	134	10.376	0.985	10.224
10	24.529	24.529	0.226	258.792	258.565	3	0	0	146	10.541	0.980	10.329

Table 4.13. Data from Case #10: 6m distance & 3 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	24.631	24.631	0.217	89.376	89.158	44	0	0	101	3.620	0.697	2.521
2	26.835	26.835	0.232	98.047	97.815	16	0	0	144	3.645	0.900	3.281
3	61.781	61.781	0.260	213.925	213.665	22	0	0	344	3.458	0.940	3.251
4	26.032	26.032	0.195	88.599	88.404	9	0	0	137	3.396	0.938	3.187
5	22.934	22.934	0.205	83.729	83.524	27	0	0	109	3.642	0.801	2.919
6	27.265	27.265	0.221	97.655	97.435	18	0	0	142	3.574	0.888	3.172
7	36.501	36.501	0.213	126.173	125.960	35	0	0	176	3.451	0.834	2.878
8	20.919	20.919	0.215	74.504	74.289	27	0	0	93	3.551	0.775	2.752
9	22.929	22.929	0.272	83.751	83.479	35	0	0	101	3.641	0.743	2.704
10	25.381	25.381	0.221	90.505	90.283	20	0	0	129	3.557	0.866	3.080

Table 4.14. Data from Case #11: 6m distance & 5 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	24.874	24.874	0.208	56.364	56.156	63	0	0	82	2.258	0.566	1.277
2	27.647	27.647	0.229	60.914	60.685	26	0	0	134	2.195	0.838	1.838
3	64.792	64.792	0.206	131.086	130.879	31	0	0	335	2.020	0.915	1.849
4	25.332	25.332	0.211	56.603	56.392	18	0	0	128	2.226	0.877	1.952
5	25.257	25.257	0.198	53.680	53.483	41	0	0	95	2.118	0.699	1.479
6	28.276	28.276	0.208	60.725	60.518	24	0	0	136	2.140	0.850	1.819
7	37.291	37.291	0.245	79.553	79.308	52	0	0	159	2.127	0.754	1.603
8	20.730	20.730	0.207	47.782	47.575	42	0	0	78	2.295	0.650	1.492
9	23.638	23.638	0.265	54.143	53.878	52	0	0	84	2.279	0.618	1.408
10	25.575	25.575	0.268	57.002	56.734	29	0	0	120	2.218	0.805	1.787

Table 4.15. Data from Case #12: 6m distance & 7 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	25.822	25.822	0.285	42.774	42.489	57	0	0	88	1.645	0.607	0.999
2	29.282	29.282	0.224	45.311	45.087	22	0	0	138	1.540	0.863	1.328
3	71.042	71.042	0.324	95.948	95.624	28	0	0	338	1.346	0.923	1.243
4	26.683	26.683	0.275	41.843	41.568	15	0	0	131	1.558	0.897	1.398
5	23.923	23.923	0.268	39.899	39.632	37	0	0	99	1.657	0.728	1.206
6	28.857	28.857	0.198	45.272	45.074	20	0	0	140	1.562	0.875	1.367
7	36.976	36.976	0.226	58.301	58.074	43	0	0	168	1.571	0.796	1.251
8	22.497	22.497	0.211	36.148	35.938	37	0	0	83	1.597	0.692	1.105
9	24.140	24.140	0.227	40.293	40.066	46	0	0	90	1.660	0.662	1.098
10	26.721	26.721	0.237	42.443	42.206	28	0	0	121	1.580	0.812	1.283

Table 4.16. Data from Case #13: 8m distance & 1 worker

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	30.039	30.039	0.181	307.937	307.756	0	0	0	178	10.245	1.000	10.245
2	25.041	25.041	0.218	256.863	256.644	0	0	0	148	10.249	1.000	10.249
3	26.720	26.720	0.201	280.699	280.497	0	0	0	162	10.497	1.000	10.497
4	27.424	27.424	0.232	277.025	276.793	0	0	0	161	10.093	1.000	10.093
5	27.770	27.770	0.254	286.789	286.536	0	0	0	166	10.318	1.000	10.318
6	33.686	33.686	0.192	342.113	341.921	0	0	0	198	10.150	1.000	10.150
7	43.436	43.436	0.183	447.563	447.380	0	0	0	260	10.300	1.000	10.300
8	25.086	25.086	0.167	261.342	261.175	0	0	0	152	10.411	1.000	10.411
9	24.222	24.222	0.202	254.614	254.412	0	0	0	147	10.504	1.000	10.504
10	29.498	29.498	0.187	297.032	296.845	0	0	0	172	10.063	1.000	10.063

Table 4.17. Data from Case #14: 8m distance & 3 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	31.284	31.284	0.205	107.379	107.174	3	0	0	175	3.426	0.983	3.368
2	24.964	24.964	0.203	90.223	90.020	1	0	0	147	3.606	0.993	3.582
3	28.881	28.881	0.240	97.393	97.152	4	0	0	158	3.364	0.975	3.281
4	28.859	28.859	0.264	96.948	96.684	0	0	0	161	3.350	1.000	3.350
5	29.835	29.835	0.205	100.449	100.245	0	0	0	166	3.360	1.000	3.360
6	35.269	35.269	0.218	117.250	117.033	6	0	0	192	3.318	0.970	3.218
7	43.432	43.432	0.218	153.346	153.129	1	0	0	259	3.526	0.996	3.512
8	26.667	26.667	0.201	91.312	91.111	2	0	0	150	3.417	0.987	3.372
9	24.654	24.654	0.280	89.132	88.852	2	0	0	145	3.604	0.986	3.555
10	28.969	28.969	0.250	104.146	103.896	1	0	0	171	3.586	0.994	3.566

Table 4.18. Data from Case #15: 8m distance & 5 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	31.830	31.830	0.223	66.644	66.421	5	0	0	173	2.087	0.972	2.028
2	26.085	26.085	0.218	56.461	56.243	1	0	0	147	2.156	0.993	2.142
3	30.144	30.144	0.228	61.420	61.192	4	0	0	158	2.030	0.975	1.980
4	28.022	28.022	0.182	61.158	60.976	0	0	0	161	2.176	1.000	2.176
5	28.222	28.222	0.193	63.118	62.925	0	0	0	166	2.230	1.000	2.230
6	34.942	34.942	0.262	73.775	73.513	10	0	0	188	2.104	0.949	1.998
7	49.159	49.159	0.203	93.759	93.556	1	0	0	259	1.903	0.996	1.896
8	25.694	25.694	0.198	57.769	57.572	3	0	0	149	2.241	0.980	2.196
9	25.698	25.698	0.197	56.335	56.138	2	0	0	145	2.185	0.986	2.155
10	30.187	30.187	0.202	64.749	64.547	1	0	0	171	2.138	0.994	2.126

Table 4.19. Data from Case #16: 8m distance & 7 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	32.717	32.717	0.224	49.210	48.986	4	0	0	174	1.497	0.978	1.464
2	27.916	27.916	0.242	42.680	42.438	0	0	0	148	1.520	1.000	1.520
3	30.857	30.857	0.227	45.572	45.345	2	0	0	160	1.470	0.988	1.451
4	28.202	28.202	0.261	44.717	44.456	0	0	0	161	1.576	1.000	1.576
5	31.390	31.390	0.264	46.643	46.380	0	0	0	166	1.478	1.000	1.478
6	36.020	36.020	0.214	54.400	54.186	7	0	0	191	1.504	0.965	1.451
7	49.333	49.333	0.234	69.013	68.779	1	0	0	259	1.394	0.996	1.389
8	27.659	27.659	0.222	43.166	42.943	2	0	0	150	1.553	0.987	1.532
9	26.309	26.309	0.205	41.927	41.722	2	0	0	145	1.586	0.986	1.564
10	32.362	32.362	0.196	47.653	47.457	1	0	0	171	1.466	0.994	1.458

Table 4.20. Data from Case #17: 10m distance & 1 worker

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	26.386	26.386	0.197	274.603	274.406	0	0	0	158	10.400	1.000	10.400
2	32.651	32.651	0.239	278.427	278.189	0	0	0	161	8.520	1.000	8.520
3	25.537	25.537	0.204	262.441	262.236	0	0	0	151	10.269	1.000	10.269
4	24.644	24.644	0.189	256.200	256.011	0	0	0	148	10.388	1.000	10.388
5	25.347	25.347	0.191	248.895	248.704	0	0	0	143	9.812	1.000	9.812
6	5.889	5.889	0.206	66.260	66.054	0	0	0	36	11.217	1.000	11.217
7	23.067	23.067	0.209	240.070	239.862	0	0	0	138	10.398	1.000	10.398
8	25.027	25.027	0.259	257.658	257.400	0	0	0	148	10.285	1.000	10.285
9	22.498	22.498	0.169	229.959	229.790	0	0	0	132	10.214	1.000	10.214
10	25.475	25.475	0.227	266.017	265.790	0	0	0	153	10.433	1.000	10.433

Table 4.21. Data from Case #18: 10m distance & 3 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	26.613	26.613	0.227	97.002	96.776	30	0	0	128	3.636	0.810	2.946
2	28.065	28.065	0.221	99.220	98.999	41	0	0	120	3.527	0.745	2.629
3	25.753	25.753	0.218	92.765	92.547	38	0	0	113	3.594	0.748	2.689
4	24.606	24.606	0.207	91.404	91.197	8	0	0	140	3.706	0.946	3.506
5	24.214	24.214	0.241	87.804	87.563	10	0	0	133	3.616	0.930	3.363
6	5.810	5.810	0.217	25.844	25.627	6	0	0	30	4.411	0.833	3.676
7	23.721	23.721	0.211	84.681	84.470	9	0	0	129	3.561	0.935	3.329
8	25.397	25.397	0.231	90.779	90.548	10	0	0	138	3.565	0.932	3.324
9	22.715	22.715	0.219	81.433	81.214	8	0	0	124	3.575	0.939	3.359
10	26.277	26.277	0.223	92.903	92.680	10	0	0	143	3.527	0.935	3.297

Table 4.22. Data from Case #19: 10m distance & 5 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	27.555	27.555	0.186	60.833	60.647	48	0	0	110	2.201	0.696	1.532
2	29.023	29.023	0.233	62.755	62.522	61	0	0	100	2.154	0.621	1.338
3	26.545	26.545	0.209	59.518	59.309	59	0	0	92	2.234	0.609	1.361
4	25.913	25.913	0.241	57.082	56.841	13	0	0	135	2.194	0.912	2.001
5	25.597	25.597	0.240	54.914	54.675	15	0	0	128	2.136	0.895	1.912
6	5.887	5.887	0.277	18.827	18.550	9	0	0	27	3.151	0.750	2.363
7	24.554	24.554	0.211	53.343	53.132	16	0	0	122	2.164	0.884	1.913
8	25.545	25.545	0.188	56.697	56.509	13	0	0	135	2.212	0.912	2.018
9	23.445	23.445	0.196	51.542	51.346	12	0	0	120	2.190	0.909	1.991
10	27.165	27.165	0.240	58.580	58.340	15	0	0	138	2.148	0.902	1.937

Table 4.23. Data from Case #20: 10m distance & 7 workers

Trial #	t_{cl}	TD	t_{sf}	t_{el}	PD	TP	TN	FP	FN	STR	OER	FES
1	29.847	29.847	0.216	45.460	45.244	40	0	0	118	1.516	0.747	1.132
2	28.363	28.363	0.195	46.145	45.950	53	0	0	108	1.620	0.671	1.087
3	28.506	28.506	0.240	43.802	43.563	50	0	0	101	1.528	0.669	1.022
4	25.779	25.779	0.240	42.327	42.087	11	0	0	137	1.633	0.926	1.511
5	25.683	25.683	0.282	41.221	40.938	13	0	0	130	1.594	0.909	1.449
6	5.872	5.872	0.228	15.425	15.197	8	0	0	28	2.588	0.778	2.013
7	23.971	23.971	0.237	40.185	39.948	14	0	0	124	1.667	0.899	1.497
8	25.803	25.803	0.249	42.951	42.702	12	0	0	136	1.655	0.919	1.521
9	23.009	23.009	0.240	38.457	38.216	10	0	0	122	1.661	0.924	1.535
10	25.628	25.628	0.210	43.590	43.380	12	0	0	141	1.693	0.922	1.560

CHAPTER 5. CONCLUSION

5.1 Summary

This study introduced an alternative way of detecting drones visually rather than RADAR, with affordability (cost effective), accessibility (easy to build from scratch), and scalability (future growth in mind) as the main objectives. RADAR is fairly expensive equipment, while the camera + SBC combination used in this study is relatively cheap and easily acquirable.

The prototype system has proven that it is more than possible to detect a drone visually, using cheap equipment and CNN. Although the system has shown a issue of accuracy, but there exists a way to improve and resolve the issue. Other than the accuracy, the design of the system is proven scalable, with mere less than 10 nodes already showing near real time performance. Therefore, we can conclude that within certain boundary (which the solution already exists), the proposed system is accurate and scalable, fulfilling the original objectives.

The design shown in this study is not bound to cheap equipments, however. If being cost effective is not a big concern, then many of the equipments used in this study can be replaced with ones of better quality and performance. A better camera will yield better image of drones with sharper edges, helping the neural network to easily recognize the outline of the drone. A better computing unit, such as GPU, will lower the fixed amount of time the neural network spends on each frame, improving the responsiveness of the system. Based on this study, there are many possible applications available.

The price of SBC is continuously decreasing while the performance is increasing, so in near future these ‘economic’ equipment can surely take part in production environment. Moreover, recently the TensorFlow project began its official support on Raspbian [45], opening wide for more practical applications. This study is believed to put a footprint in before those applications.

5.2 Future Work

During the experiment, 700 images of both artificial and real photos of a drone has been created. Currently, there is no known publicly available image dataset solely focusing on drones. Deep learning is well known for its high demand on image dataset for training, thus having a drone image dataset available will greatly aid researches in similar fields of studies. The shortcoming of this study also derives from lack of image dataset. More dataset for training means the CNN can extract more data the inference graph—roughly said, outline matrix—can be made with. With more dataset, and enough training, the neural network is expected to detect in much higher accuracy, regardless of the environment the system might run; the system could be running indoor, running on the roof of a house, running at the rooftop of a sky scrapper, et cetera. Once properly trained, the system can applied anywhere without considerable change to its configuration. All this can be done if there are more dataset, easily accessible without any restriction. Therefore, making a publicly available, online dataset of drones relying on crowd contribution will spark not only great advancement in drone research but also help other general deep learning researchers. Once this kind of large dataset is available, studies like this can be performed again for better results.

REFERENCES

[1] T. A. Rule, "Airspace in an Age of Drones," *Boston University Law Review*, vol. 95, no. 1, pp. 155–208, 2015.

[2] T. P. Ehrhard, *Air Force UAVs : the secret history*. Arlington, VA: Mitchell Institute Press, 2010.

[3] K. Dalamagkidis, "Aviation History and Unmanned Flight," in *Handbook of Unmanned Aerial Vehicles*, Valavanis Kimon P. and G. J. Vachtsevanos, Eds. Dordrecht: Springer Netherlands, 2015, pp. 57–81. [Online]. Available: https://doi.org/10.1007/978-90-481-9707-1_93

[4] D. M. Atwater, "The Commercial Global Drone Market." *Graziadio Business Review*, vol. 18, no. 2, pp. 1–7, 4 2015. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=118656985&site=ehost-live

[5] P. Chamberlain, *Drones and Journalism.*, P. (Firm), Ed. Place of publication not identified : Taylor and Francis Ltd : Routledge, 2017.

[6] B. Dane, "Drones: Designed for Product Delivery," *Design Management Review*, vol. 26, no. 1, pp. 40–48, 7 2015. [Online]. Available: https://doi.org/10.1111/drev.10313

[7] N. TEUCHERT, "Drone Inspections: HRSG Maintenance from a Bird's Eye View." *Power Engineering*, vol. 122, no. 3, pp. 10–14, 3 2018. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=128570671&site=ehost-live

[8] P. N. Borden, “The Peering Predator: Drone Technology Leaves Children Unprotected from Registered Sex Offenders.” *Campbell Law Review*, vol. 39, no. 1, pp. 167–185, 2017. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=ofm&AN=121319566&site=ehost-live

[9] A. Press, “2 Plead Guilty to Using Drones to Smuggle Heroin Into U.S.” *Time.com*, p. N.PAG, 8 2015. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=crh&AN=108888174&site=ehost-live

[10] J. Vanian, “Prisoners Are Using Drones To Smuggle In Drugs and Porn.” *Fortune.com*, p. 60, 6 2017. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=123635071&site=ehost-live

[11] B. A. Card, “Terror from Above,” *Air & Space Power Journal*, vol. 32, no. 1, pp. 80–95, 2018. [Online]. Available: https://search.proquest.com/docview/2050581361?accountid=13360https://search.proquest.com/docview/2050581361?accountid=13360

[12] M. D. Shear, “White House Drone Crash Described as a U.S. Workers Drunken Lark,” *New York Times*, vol. 164, no. 56760, p. A15, 2015.

[13] Federal Aviation Administration, “REGISTRATION AND MARKING REQUIREMENTS FOR SMALL UNMANNED AIRCRAFT,” 2015.

[14] J. M. Goppert, A. R. Wagoner, D. K. Schrader, S. Ghose, Y. Kim, S. Park, M. Gomez, E. T. Matson, and M. J. Hopmeier, “Realization of an autonomous, air-to-air Counter Unmanned Aerial System (CUAS),” in *Proceedings - 2017 1st IEEE International Conference on Robotic Computing, IRC 2017*, 2017.

[15] S. S. Swords, *Technical History of the Beginnings of Radar*, 1st ed. London, United Kingdom: The Institution of Engineering and Technology, 1986.

[16] J. A. Scheer and W. L. Melvin, *Principles of Modern Radar : Volume 3 Radar Applications*. Raleigh, UNITED STATES: Institution of Engineering & Technology, 2013. [Online]. Available: http://ebookcentral.proquest.com/lib/purdue/detail.action?docID=1632047

[17] S. M. Shammas, "The Future of Driving: A Look Into the Technology Behind Autonomous Vehicles," Ph.D. dissertation, Oakland University, 2017. [Online]. Available: http://hdl.handle.net/10323/4529

[18] J. J. M. d. Wit, R. I. A. Harmanny, and G. Prémel-Cabic, "Micro-Doppler analysis of small UAVs," in *2012 9th European Radar Conference*, 2012, pp. 210–213.

[19] M. Jahangir, C. J. Baker, and G. A. Oswald, "Doppler characteristics of micro-drones with L-Band multibeam staring radar," in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 1052–1057.

[20] A. R. Wagoner, "A Monocular Vision-Based Target Surveillance and Interception System Demonstrated in a Counter Unmanned Aerial System (CUAS) Application," 2017.

[21] A. R. Wagoner, D. K. Schrader, and E. T. Matson, "Towards a vision-based targeting system for counter unmanned aerial systems (CUAS)," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2017, pp. 237–242.

[22] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, 1995.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[24] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," in *Shape, Contour and Grouping in Computer Vision*, D. A. Forsyth, J. L. Mundy, V. di Gesú, and R. Cipolla, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19

[25] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004, pp. 97–104.

[26] M. Szarvas, U. Sakai, and J. Ogata, "Real-time Pedestrian Detection Using LIDAR and Convolutional Neural Networks," in *2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 213–218.

[27] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[28] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Systems with Applications*, vol. 72, pp. 327–334, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095741741630598X

[29] B. K. Kim, H. Kang, and S. Park, "Drone Classification Using Convolutional Neural Networks With Merged Doppler Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 1, pp. 38–42, 2017.

[30] C. Aker and S. Kalkan, "Using Deep Networks for Drone Detection," *CoRR*, vol. abs/1706.05726, 2017. [Online]. Available: http://arxiv.org/abs/1706.05726

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[32] "Raspberry Pi 3 Model B." [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[33] "Raspberry Pi Camera Module v2." [Online]. Available: https://www.raspberrypi.org/products/camera-module-v2/

[34] "D-Link DGS-1024A 24-Port Unmanaged Gigabit Switch." [Online]. Available: http://us.dlink.com/products/connect/24-port-unmanaged-gigabit-switch/

[35] "DJI Phantom 2." [Online]. Available: https://www.dji.com/phantom-2

[36] J. Redmon, "Darknet: Open Source Neural Networks in C," 2013. [Online]. Available: http://pjreddie.com/darknet/

[37] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[38] Google, "TensorFlow," 2015. [Online]. Available: https://www.tensorflow.org/

[39] "TensorFlow Model Zoo." [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *CoRR*, vol. abs/1512.02325, 2015. [Online]. Available: http://arxiv.org/abs/1512.02325

[41] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[42] M. Sandler and A. Howard, "MobileNetV2: The Next Generation of On-Device Computer Vision Networks." [Online]. Available: https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018.

[44] P. R. L. de Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "PKLot A robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415001086

[45] P. Warden, "TensorFlow 1.9 Officially Supports the Raspberry Pi." [Online]. Available: https://medium.com/tensorflow/tensorflow-1-9-officially-supports-the-raspberry-pi-b91669b0aa0