BSc project - Computer Vision and Aerial Photography for Animal Detection

April 8, 2019

Abstract

Knowledge of animal populations is a vital necessity for wildlife conservation. Traditional surveying techniques involving manual labour are ineffective and time-consuming, therefore an innovative method applying a deep learning object detection algorithm to images or videos captured by drone flights was introduced as a solution for automated analysis on large visual datasets. The objective of the project is to design and prototype a methodology derived from this stated idea to study squirrels, a common animal in London. The object detection algorithm YOLO v3 was implemented and a small-scale overhead dataset of squirrels was collected and labelled for retraining the model. Additionally, a live stream of drone video was set up for conducting analysis on devices with a high level of computational power. However, the training through transfer learning and the real-time detection on drone videos were not achieved successfully due to programming difficulties, which can be expected to be resolved by appropriate technical supports and investments.

Contents

1 Introduction **2**

2 Theory **3**
2.1 Computer Vision 3
2.2 Convolutional Neural Network 4
2.3 Object Detection - YOLO v3 6
2.4 Flying Mechanism and Stabilisation of Drones 7

3 Methods, Results, and Discussions **9**
3.1 Implementation of YOLO v3 9
3.2 Data Collection and Training - Squirrel 10
3.3 Live Streaming and Real-Time Analysis 11
3.4 Thermal Camera 12

4 Conclusions **13**

Project Summary

The project aims to combine the deep learning-based computer vision and drone-driven aerial photography in order to create an efficient solution for animal detection. More specifically, the team planned to modify and train the published object detection model YOLO v3 to recognise squirrels, and utilise the model to analyse the aerial videos filmed by the DJI Phantom Pro 4 v2 drone for real-time investigation. In this report, the motivation and objective of the project are established in the **Introduction** section. This section discusses the importance of wildlife conservation and how drone technology and deep learning neural network can be employed for animal detection. Following that, the **Theory** chapter addresses the essential theoretical foundation of computer vision powered by the application of convolutional neural network, the functional principle of the object detection algorithm YOLO v3, as well as the flying and stabilising mechanisms of drones for acquiring visual data of good quality expeditiously. The practical methods, results obtained, and relevant discussions are condensed into the **Methods, Results, and Discussions** section to explain the work undertaken and the overall progression clearly and logically. This section covers the implementation of the YOLO v3 algorithm, the preparation of the training data, the training strategy, the set-up of the live drone video stream, the methods experimented for real-time object detection on drone videos, as well as the potential introduction of thermal imagery. Finally, the **Conclusions** chapter briefly sums up the outcomes achieved throughout the project and suggests some possible improvements for further research.

1 Introduction

In recent centuries, the development of human society and overpopulation have led to increasing exploitation of natural resources and the environment. The growth of human activities, such as deforestation and inadvertent introduction of species, inevitably causes loss of biodiversity and even mass extinctions. The destruction of ecosystems has profound impacts on the survival of the human race and other species, since it incapacitates important natural processes including regulation of the atmosphere and climate, water purification and retention, as well as soil formation and fertilisation. A healthy ecological balance also has remarkable economical, ethical, and aesthetic values to the

advantage of humans. The conservation of wildlife is therefore a pressing and indispensable mission to protect endangered species and restore the environmental equilibrium [1][2].
The conservation of species generally involves the recovery of the natural habitats and reintroduction of the animals. In order to accelerate the progress by positive human intervention, it is crucial to acquire an overarching understanding of the populations of animals, including their population sizes, locations, ranges, and distributions. Traditional surveying techniques are ineffective for observing cryptic, low-density animals or animals inhabiting remote, isolated areas. This implies that the research and further protection of keystone species governing the wellbeing of the ecosystems are challenging. To overcome this problem, the application of drone was proposed by biologists to smoothly inspect a large area from an aerial viewpoint. With the incorporation of the thermal camera, animals with distinctive heat signature can be shown clearly against the background and easily identified. Aerial photography driven by drone provides a cheap, fast solution for data collection [3][4]. However, employing the conventional method to manually process and analyse the large volume of visual data gathered by drones is inaccurate and time-consuming. Consequently, machine learning was introduced to enable automated real-time detection of animals from large-scale datasets. This means that population assessments can be completed efficiently by algorithms designed for recognising and counting certain animals [5][6]. In summary, wildlife conservation researchers have invented and implemented a powerful new method for animal detection using deep learning algorithms on drone-derived aerial imaging. As an extension, similar techniques can also be applied to prevent poachers from hunting and harming animals in nature reserves.
This project was motivated by this revolutionary method combining and applying two cutting-edge technologies, namely drone and machine learning, to perform wildlife detection for conservation purposes. The objective of this project is to implement and train a deep learning object detection algorithm for a chosen target animal, and integrate the aerial video stream from the drone with this trained model to achieve real-time surveying on the animals. More precisely, the project entails implementing the deep learning object detection algorithm, collecting data for training, training the model, and setting up the drone video stream, as well as understanding the theoretical basis of computer vision, convolutional neural network, object detection, and flying mechanism of drones.

2 Theory

2.1 Computer Vision

Vision, more precisely visual perception, is the ability to interpret the light signals received by eyes and acquire information regarding the surrounding environment. From object recognition to spatial awareness, human and many sentient beings have the vision as their primary means of perception. Although vision is often taken for granted as it is generally deemed effortless in everyday life, the neurological processes behind the visual cognition are immensely complicated and intricate that it needs to be facilitated by numerous brain components.
Naturally, in our quest to recreating machines rivaling or surpassing human performances, the implementation of vision is essential. Computer vision has thus prospered because of the versatility of its applications. Broadly speaking, computer vision is the technology for analysing and interpreting digital images or videos to extract meaningful information and make logical decisions. In the context of images classification, the information is the distinct attributes for the object in the image and the decision is predicting what the object is. In recent years, computer vision has contributed to many substantial breakthroughs, for instance, navigation of self-driving vehicles and screening of diagnostic medical scans. Although its competency is still inferior to human vision in some aspects, it has been proved to have remarkable applicability as well as future potential.

To some extent, computer vision is about recognising patterns. With sufficient visual data of a certain object exemplifying some reappearing features such as colours and shapes, machine learning algorithms can retrieve patterns to formulate a characteristic profile for that object, hence recognise it. In the past few years, with the development of artificial intelligence, great improvements have been achieved for this learning process. One of the most prominent advancement is the use of neural networks to process large datasets and solve complicated problems [7].

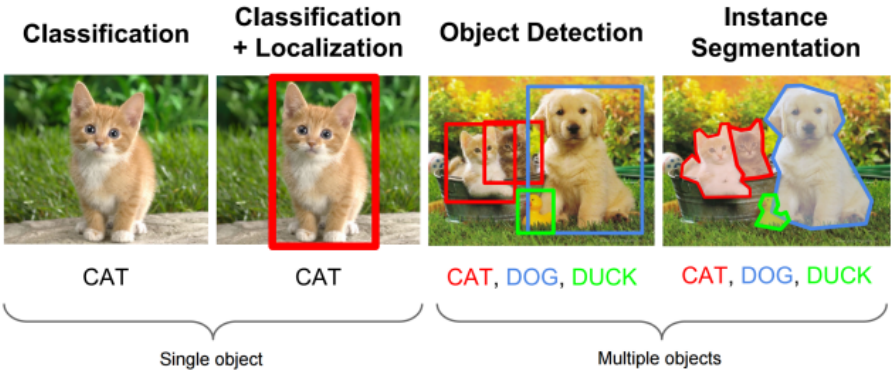

Figure 1: Common computer vision tasks include classification, object detection, and segmentation. More advanced tasks such as facial recognition and pose estimation developed in recent years by the invention of the convolutional neural network [15]

2.2 Convolutional Neural Network

Computer vision is an imitation of biological cognition, therefore it is only natural that its solution is inspired by biology and neuroscience. Neurophysiologists Hubel and Wiesel conducted comprehensive research on information processing in the visual system in the 1960s and discovered that "in the cerebral cortex, signals are analysed in sequence by cells with the specific tasks of interpreting contrasts, patterns, and movements". The studies indicate that certain groups of neurons are arranged in anatomical columns in the visual cortex and distinct neurons are responsive to corresponding visual features. This leads to the invention of neural networks, an architecture or framework structured conceptually based on these properties of visual cortex [8].

A neural network consists of multiple layers that are analogous to the columns in visual cortex. A layer is made of multiple nodes that are equivalent to neurons, and the nodes in adjacent layers are connected to each other by edges with the outputs of a layer becoming the inputs of the next one. A node is a point of computation that processes an aspect of data and each edge connected is associated with a set of weights. A weight is a normalised value assigned to an input quantifying the significance of it for a certain task. When passing through a node, the weighted sum of all inputs from edges is obtained, much similar to how neurons respond to stimuli, the node will then determines the output by its associated activation function [9] [10].

Overall, a neural network is composed of an input layer, many hidden layers, and an output layer. The objectives of the input and output layers are self-evident, whereas the hidden layers perform the functions of interpreting the data and transforming inputs to outputs. A neural network with multiple hidden layers is described as 'deep'. The hidden layer structure enables the deep neural network models to compute from low-level to high-level features incrementally, therefore eliminates the need of domain expertise and feature extraction for conventional machine learning, meaning larger sizes of data can be analysed and more complicated challenges can be solved [11].

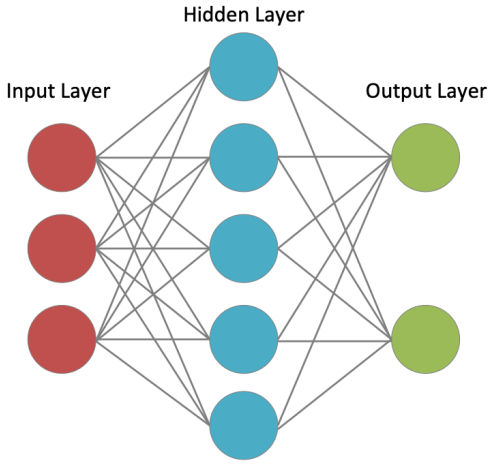

Figure 2: A single layer neural network with an input layer, a hidden layer, and an output layer. Nodes in adjacent layers are connected to each other. In deep learning networks, there are multiple hidden layers for more computational operations.

The input visual data, most commonly images and videos for computer vision, is pixel data, meaning that an image is represented and stored as a matrix of pixel values from 0 to 255. For an RGB image, there exist three colour channels, hence it can be expressed as three two dimensional matrices stacked over each other. The hidden layers involve identifying light and dark areas, categorising lines and shapes, further recognising high-end features, and deciding the final classification or localisation. The output layer can produce new images or videos containing bounding boxes enclosing objects or headers indicating the classes, it can also give other predictions and commands depending on the purpose of the model [12].
With this architecture, the neural network has the ability to learn and predict. The learning refers to the process of finding the optimal weights from inputs that give predictions closest to target outputs. The values of weights are changed during the training to find the best combination that highlights the most correlated, important features and disregards the irrelevant ones. On the other hand, the predicting process applies the weights learned from the prior training and produces an output accordingly. For an image classifier, the neural network learns from images (input) to attain weights that best fit the images to their labels (target output), the weights can subsequently be used for identifying the content of an image (output).
In more technical terms, a neural network learns from backpropagation and predicts by forward propagation. For an untrained model, all weights are randomly assigned initially. During the course of training, each labelled sample is first passed through the neural network and an incorrect output is calculated accordingly. By comparing this incorrect prediction and the expected labelled outcome, the error can be evaluated from the cost function. This error value is subsequently propagated backwards to adjust all weights via gradient descent, an optimisation algorithm for minimising the cost function. The adjusted weights obtained after this procedure straightforwardly result in reduced errors, hence higher precision. The process is repeated for all samples and the training can be considered as completed after the model has learned from a sufficiently large dataset and achieve a high standard of accuracy [13].
The discussion in this report focuses on the conceptual descriptions of the structure and functions of neural networks. The mathematical details are beside the point of this report, thus not discussed further here.

2.3 Object Detection - YOLO v3

Object detection is simply classification with localisation on multiple objects. In the early twenty-first century, many efficient machine learning models for image classification already came to existence, but the true ground-breaking advancements commenced in 2012 when convolutional neural networks were first introduced. The approaches using deep learning CNNs became the most prominent and standard solution for computer vision ever since. Nowadays, there exist various well-recognised deep learning algorithms designed for object detection, they can generally be categorised into two groups depending on whether a method is based on classification or regression. Briefly speaking, classification methods need to first identify and select some regions of interest in an input image, then proceed to classify each region. This means that the predictions have to be made repeatedly for all individual regions, hence the process is slow and expensive. Algorithms including Region CNNs (R-CNNS) and its derivatives such as Fast R-CNN and Faster R-CNN all belong to this group. Conversely, regression methods such as Single Shot MultiBox Detector (SSD) and YOLO are significantly faster. YOLO, standing for "You Only Look Once", is a state-of-the-art object detection algorithm that applies deep learning and a fully convolutional neural network. The general conceptual outline is distinctive since an image is only forwarded through the neural network once. The neural network takes into account the features from the entire image to predict each bounding box, as well as the class of the object inside each box simultaneously [14][15].

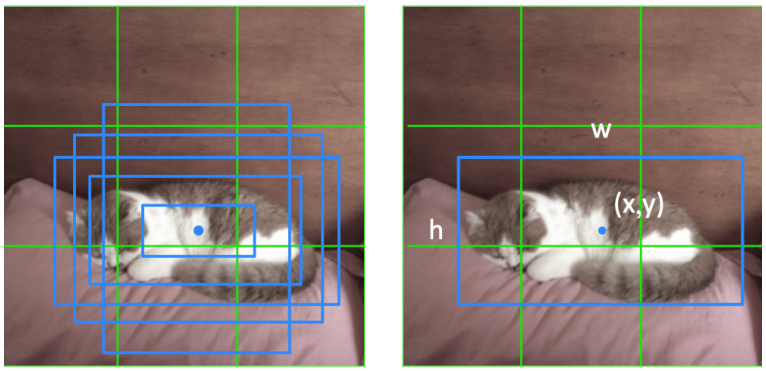

Figure 3: An example visualising the bounding box predictions in 3x3 grid cells. The left image shows 5 anchor boxes predicted, and the right image displays the bounding box (x,y,w,h) with the highest confidence level.

YOLO divides the input image into an $S \times S$ grid of cells. Each grid cell detects one object as well as predicts B bounding boxes (anchor boxes) enclosing it. In other words, the grid cell containing the centre of an object is responsible for predicting the bounding boxes surrounding that object.
A bounding box is defined by five predictions: normalised x-coordinate of the centre x, y-coordinate of the centre y, height h, width w, and confidence score quantifying how likely the box contains the object and how accurately the box encloses it. Consequently, the box with the highest confidence score determines the localisation of the object [16][17].
A grid cell also predicts C conditional class probabilities, each of which represents the likeliness of the detected object belonging to a specified class. The class probability map of the image can be plotted according to the highest value for each grid and classify the object inside the box. As a

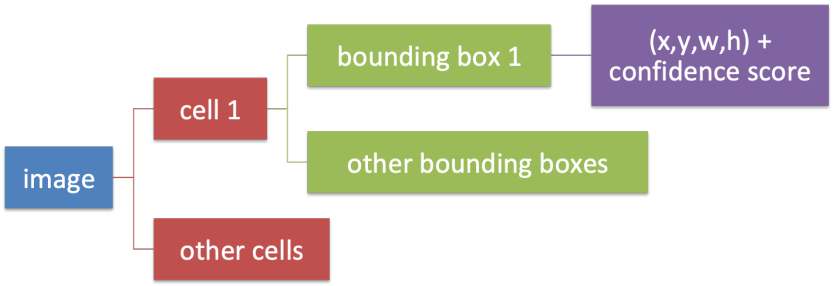

Figure 4: The simplified outline of bounding box prediction. Several bounding boxes are predicted for each grid cell and each has a confidence score quantifying the likeliness of the box containing the object and the accuracy of the enclosure.

result, there exists $S \times S \times B$ bounding boxes and $S \times S \times B \times C$ class probabilities as outputs. To reduce complexity and avoid redundancy, the non-maximum suppression technique is used. The vast majority of the bounding boxes are then disregarded due to their low confidence scores [18]. In conclusion, each object detected is assigned with an optimally predicted bounding box indicating the position. The highest class probability associated with the box then determines the class it belongs to. The YOLO algorithm thus achieves both localisation and classification efficiently.

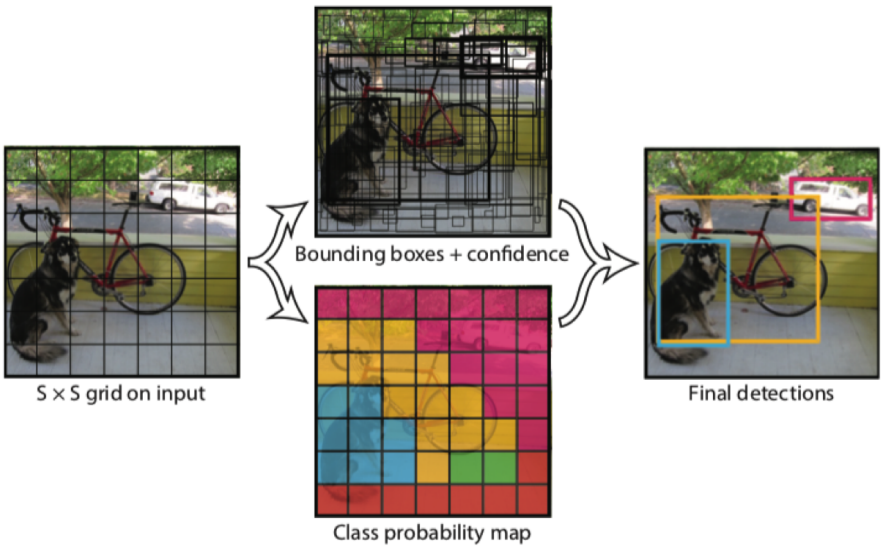

Figure 5: The YOLO v3 algorithm takes into account both the confidence scores of bounding boxes and the class probabilities for all cells to determine the final localisation and classification to accomplish object detection [16].

2.4 Flying Mechanism and Stabilisation of Drones

Drone imagery has developed rapidly in recent years with the introduction of computer vision and machine learning to allow intelligent autopilot. At the same time, it is still important to acknowledge that the advancements in the field of aerial photography are built upon the fundamental physics and engineering of drones. The physical flying mechanism for drones demonstrates great simplicity

in principle: a propeller on drone generates a thrust depending on its rotational speed and direction. Each propeller can be individually controlled and the combination of the thrust forces leads to an overall drone movement.

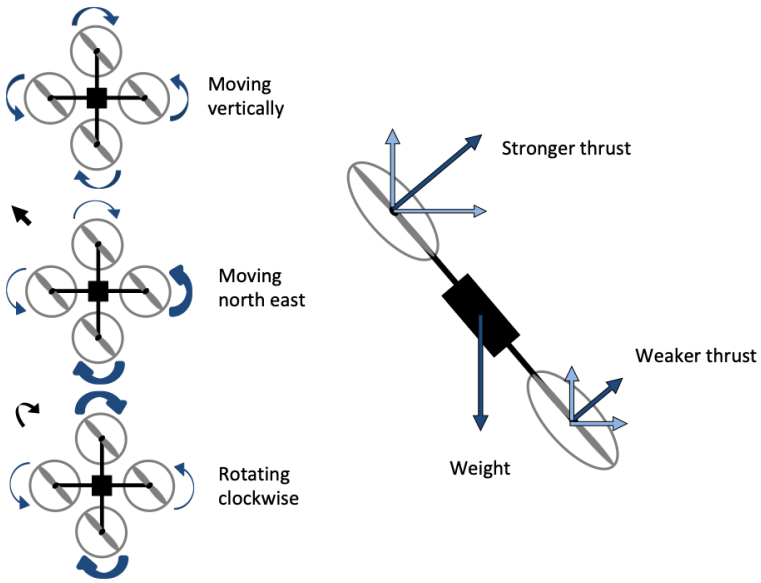

Figure 6: The illustrations of three examples of movements of a quadcopter drone and a free body diagram for a tilted drone moving horizontally.

For vertical movements of a drone such as hovering, ascending, and descending, all propellers produce an equal amount of lift and forces only exist on the vertical plane. When hovering, the drone reaches a mechanical equilibrium by having each propeller create a lift force balancing a quarter of the drone weight. In order to ascend or descend, the magnitude of the upward lifts is changed by adjusting the angular speed uniformly. To avoid unwanted revolution, two sets of diagonally connected propellers should rotate in opposite directions to eliminate the total angular momentum so that net torque is zero.

On the other hand, horizontal movements rely on the symmetric structure of drones. As the thrust is always perpendicular to the propeller disks, the horizontal force components can only be generated by tilting the drone. To initiate a forward, backward, or sideway motions, the pair of neighboring propellers opposite to the intended movement direction needs to have a greater rotational rate compared to the other pair. The difference in thrusts tilts the drone, and the horizontal forces moves the drone sideways. In these cases, a pair of adjacent propellers has one spinning clockwise and the other one anticlockwise with the same angular frequency, therefore each pair has zero angular momentum. The drone experiences no torque and does not turn.

In terms of the rotation of a drone, a net angular momentum is produced by increasing the rotational speed for a diagonally connected pair of propellers and decrease it for the other pair, in conjunction with having the two paired propellers revolve in the same direction. Take clockwise rotation as an example, the pair spinning in the clockwise direction has a greater angular speed compared to the anticlockwise pair to create a net clockwise angular momentum leading to rotation.

The discussion so far concerns only the purely vertical, horizontal, and rotational movements. However, great advancements in aeronautic engineering have allowed drones to fly in an agile manner with various combinations of and smooth transition between different movement modes. Typically, drones are navigated by the remote controller: the left stick is responsible for forward/backward, left/right movements, and the left stick for elevation and rotation.

Automatic stabilisation of drones is necessary for the acquisition of good quality data, as wind and other external factors could compromise the smoothness of the flight. By incorporating sensors to detect unplanned motions and introducing algorithms for real-time reaction, drones can readjust their positions and movements to continue a flight without turbulence. Accelerometers are often integrated into drones to sense linear movements. In contrast, rotational motions are monitored by gyroscopes. Upon receiving raw data from the sensors, the flight control board can compute suitable corrections and sent to propellers to stabilise the drone. Additionally, the aerial camera on the drone has an independent stabilisation system, gimbal, for maintaining the fixed positions regardless of the drone motion.

3 Methods, Results, and Discussions

In this report, methods, results, and discussions are condensed into one section because many ideas were experimented throughout the course of the project, and not all of them lead to tangible results that can be discussed. The connections between attempts and the progression of thoughts can be explained more logically and lucidly with this arrangement.

Declaration of Work Undertaken

The work is evenly distributed between two members of the project team, Jo-Kuang Liao and Rohan Prasad. The team generally worked two days a week together on the same aspect of the project throughout the spring term. It should still be noted that Jo-Kuang worked slightly more on exploring the solutions for real-time detection, whereas Rohan worked more on labelling and trained a classifier as an experimental test.

3.1 Implementation of YOLO v3

The object detection algorithm selected for the project is YOLO v3 since it outperforms other models in terms of the efficiency for real-time processing. It runs remarkably faster than other methods with comparable accuracy [19]. In Addition, its predictions are global in the sense that it considers the entire image at once with a single neural network. The default, recommended implementation for YOLO is with Darknet, an open source neural network framework written in C and CUDA, but other frameworks such as AlexeyAB and Darkflow also have their own merit depending on the operating system, the framework preferred, and the available sources up to date [20].
PyTorch was chosen as the alternative as it is easy to pick up and more suitable for experimental projects and prototypes. PyTorch is a relatively new open source deep learning platform developed by Facebook. Although it is currently less mainstream compared to TensorFlow, it has major advantages including its support for dynamic neural networks and parallelism.
Two versions of YOLO v3 with PyTorch 0.4 on GitHub, programmed by Ultralytics LLC and Erik Linder-Norén respectively, were downloaded to complement each other, alongside the official pretrained weights for 80 default classes. Some minor editings were made on Xcode and Spyder to enable interchangeable uses of files and codes from two repositories. The program was run on terminal locally and on Google Colaboratory on cloud.
YOLO v3 provides three modes of detection on different types of files: *detect.py* for images, *video_demo.py* for videos, and *cam_demo.py* for webcam. The tests have confirmed that the functions work for JPG and AVI files, as well as on webcam, for many categories of objects including people, dog, and bicycle.

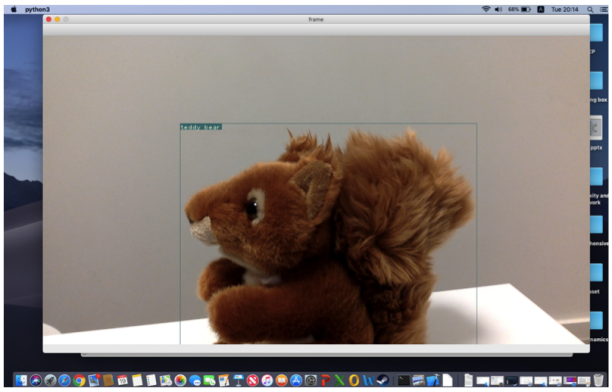

Figure 7: The default YOLO v3 algorithm classified a stuffed toy of squirrel as a teddy bear. The turquoise square box indicates the location of the object and the label in the top left corner shows the class it belongs to.

3.2 Data Collection and Training - Squirrel

Squirrels are among the most common and familiar mammals in London. They have populations scattered in most green areas in London and their frequent interactions with human have rendered them insensitive to dangers. For these reasons, the squirrel was chosen as the ideal target animal subjected to detection for this project. Squirrel is not included in the pre-trained classes for YOLO v3, therefore further training is required for the program to learn to recognise this particular animal. Collection of squirrel images was the first step towards training. Simply scraping from Google images provides an abundant amount of information but it might not be appropriate for the purpose of the project. The aerial images obtained are overhead images, and it is a relatively unconventional angle for photography before the recent popularisation of hobbyist drones. Consequently, it is preferred to have real aerial photographs or similar data for the training.
Due to the regulation on drones and the continuously windy weather, a small-scale dataset of overhead images of squirrels was attained manually in Hyde Park manually as a temporary solution. Selfie sticks were utilised for taking overhead pictures while maintaining a sufficiently long distance from squirrels to avoid startling them. In the end, approximately 30 images containing squirrels were captured as JPG files with the size of 4608×3456 pixels. The images were then labelled, boxes are drawn to specify the locations of squirrels and each box can be expressed into the Darknet format by four parameters: the normalised x centre, y centre, height, and width. Furthermore, the images were resized to 416×416 as required by the training function *train.py*.

Figure 8: Some examples of the image collected and labelled in the overhead squirrel dataset.

The function *train.py* takes input parameters including epoch and batch size. Epoch represents the

number of time the entire training dataset passed forward and backward through the neural network. It is favourable to have a large number of epochs as it allows more opportunities for the weights to be modified in the neural network and hence improves accuracy. Contrarily, batch size is the number of samples propagated through the neural network together in one go [21].
Training models from scratch requires a large amount of data, preferably labelled. Obtaining such datasets for every class of interest is challenging considering the time and manual labour required for labelling. Nevertheless, the model was retained from scratch with only 32 labelled images of squirrels as an experimental attempt. Over 30 epochs with a batch size of 16, a set of weights was attained with an accuracy of 1, indicating overfitting. Overfitting means that the model has learned the random noise in the training dataset that does not apply to general data, to the extent that the model's performance on new data is compromised. As a result, the trained mode failed to recognise the squirrels in the Youtube videos tested, as expected for a model developed based on highly inadequate data. The result suggested that a more effective training strategy is necessary.
Transfer learning is a deep learning method where a model previously developed for a task is repurposed as the starting point for other new, related tasks. The utilisation of pre-trained models that is already knowledgeable on low-level and mid-level features can facilitate expeditious learning progress for new objectives and improved performance for existing ones. Furthermore, transfer learning can cross the boundaries of isolated tasks and address more complex problems [22].
The transfer learning script for YOLO v3 uses the official weights and only retrains 3 convolutional layers out of all 106 layers. The official weight was renamed as the latest weight and the training was resumed. However, some errors occurred in the process of transfer learning and have not been solved yet. More debugging and studying of the source codes, as well as technical support from experts might be needed to overcome this problem.

3.3 Live Streaming and Real-Time Analysis

Real-time object detection using convolutional neural networks is computationally expensive and requires high-end graphics processing units (GPUs). The task is commonly achieved by installing an embedded system or redirecting the tasks to the cloud server. For the embedded system method, an AI computing device with high performance and low power demand such as NVIDIA Jetson TX1, is mounted to the drone to execute real-time tracking or detection [23]. Implementing systems with greater weight and power consumption is sometimes necessary but impractical due to the specification of the drone used. The cloud approach, on the other hand, allows low-level computation to be completed locally on the central processing unit (CPUs), and more complicated operations involving the application of CNNs can be carried out on the cloud server [24]. Nevertheless, these two methods were not adopted for this project due to the technical and hardware limitations.

Figure 9: The overview of the aerial video stream from drone to laptops or PCs, the arrows in between steps represents the Wi-Fi connections.

Alternatively, a stream for aerial videos with the drone on one end and a device with a high level of

computational power such as PC or laptop on the other end was established. The live video recorded by the drone camera on DJI Phantom Pro 4 v2 is sent through the wireless connection to the DJI GO 4 app on a mobile device, as the app is only available on Android and iOS. From this point onward, the video needs to be broadcasted on any platforms, for example Facebook or Youtube, in the real-time messaging protocol (RTMP) format via Wi-Fi. The streaming service selected for the project is Bambuser as it provides all the crucial features on an intuitive user interface. The video can then be shown on any devices connected to the internet. The stream was implemented successfully with a delay of approximately three to five seconds caused by the intermediate steps between the drone and the end device.
The real-time object detection via YOLO v3 on live video stream is another challenge. YOLO v3 currently only supports the analysis on images, videos, and webcam. In order to run it on online videos, the most feasible approach is to monitor the desktop screen directly as a whole. YOLO Live is an application on GitHub which allows detection on anything displayed on the screen. Unfortunately, YOLO Live is written in C++ whereas YOLO v3 in PyTorch is programmed in Python. The attempt at incorporating YOLO Live into the existing structure was unsuccessful due to the discrepancy between languages and the team members' lack of experiences with C++.
An attempt at modifying the source code for webcam *cam_demo.py* has been made by importing the Python MSS library to take screenshots as a form of input. However, the original webcam code employs the *VideoCapture()* class from the OpenCV library, meaning that the conversions of functions are complicated and the data formats are fundamentally different. The pixel data obtained from the screenshots has four channels and it was unclear how they correspond to the standard RGB channels. As a result, no substantial change was accomplished to assist the analysis for live videos on the internet.
The delay and the real-time detection problems can both potentially be solved by revisiting the two recommended methods discussed previously - using the embedded system or the cloud server. The technical difficulties associated with these methods, for example drone programming and connecting stream to cloud server, can be overcome by investing in hardware and advanced drone software.

3.4 Thermal Camera

DJI Phantom Pro 4 v2 is inherently equipped with a camera for standard aerial photography. Nevertheless, traditional visible light imagery encounters some limitations concerning wildlife detection, for example, the investigations on nocturnal animals at nighttime or forest animals hindered by trees, or surveying in poor weather conditions. Thermal (infrared) cameras have demonstrated great applicability in the field of wildlife research as an upgrade for visible light cameras.
Thermal cameras have the ability to detect infrared radiation emitted by objects, which is a measure of the surface temperature. The intensities of the observed infrared radiation can then be converted into electrical impulses and displayed as a heat map or numerical values. As a result, living animals with a body temperature higher than their surrounding environments can be distinguished easily from the backgrounds on the thermal images. This means that the localisation of animals can be done by simply taking a threshold value depending on the mean of the whole image to achieve segmentation. The contour obtained from the segmentation and the thermal information can subsequently be used to classify the animals by a model which has learned the animals' shapes and thermal signatures [25][26].
The team had the intention to implement thermal imaging into the existing design and has explored different channels of acquiring a thermal camera in short time through borrowing, renting, and purchasing. Inquiries have been made to the Aerial Robotics Laboratory at Imperial College London to check the accessibility of infrared cameras compatible with DJI Phantom Pro 4 v2 or any drones

already equipped with thermal cameras. However, they do not have any suitable devices available. COPTRZ, a commercial drones and drone accessories provider, gave a quote of £3,685.00 for a DJI Zenmuse XT2 infrared camera and an installing mount. The team had made the purchasing request to the project supervisor, yet the approval did not come through until near the end of the project and no real progress was made in this direction. It is expected that the effectiveness of the method can be greatly improved and the succeeding can make significant breakthroughs with the incorporation of the thermal camera.

4 Conclusions

In this project, YOLO v3 was selected as the object detection algorithm for conducting wildlife detection on squirrels. The implementation of YOLO v3 with PyTorch as the framework was successful and is capable of identifying 80 default classes. A small-scale overhead dataset for squirrel was collected and labelled manually. The team had made an attempt at applying transfer learning strategy to train the published YOLO v3 model with the gathered data, yet the result turned out to be unsuccessful due to programming difficulties. On the other hand, the real-time aerial video stream was set up to feed the data to laptops successfully. The team did not manage to integrate the stream into the existing object detector to perform real-time analysis either. Overall, the objective of the project was only partially completed and substantial improvements can potentially be made with more technical learning and support, as well as necessary investment in hardware and software.

Acknowledgements

The author would like to thank her project supervisor Dr. John Hassard for the guidance and tuition and thank her project partner for the substantial contributions made to all aspects of the project. The author would also like to show gratitude to the members of the term one team for their support and assistance.

References

[1] McDaniel, M., Sprout, E., Boudreau, D. and Turgeon, A. (2011). *Conservation*. [online] National Geographic Society. Available at: https://www.nationalgeographic.org/encyclopedia/conservation/ [Accessed 7 Apr. 2019].

[2] Marshall, M. (2015). *What is the point of saving endangered species?*. [online] BBC.com. Available at: http://www.bbc.com/earth/story/20150715-why-save-an-endangered-species [Accessed 7 Apr. 2019].

[3] Corcoran, E., Denman, S., Hanger, J., Wilson, B. and Hamilton, G. (2019). Automated detection of koalas using low-level aerial surveillance and machine learning. *Scientific Reports*, 9(1).

[4] Hodgson, J., Baylis, S., Mott, R., Herrod, A. and Clarke, R. (2016). Precision Wildlife Monitoring Using Unmanned Aerial Vehicles. *Scientific Reports* 6, 22574.

[5] Gonzalez, L. et al. (2016). Unmanned Aerial Vehicles (UAVs) and Artificial Intelligence Revolutionizing Wildlife Monitoring and Conservation. *Sensors* 16, 97.

[6] Chen, C. Liu, K. (2017). Stingray detection of aerial images with region-based convolution neural network. *2017 IEEE International Conference on Consumer Electronics - Taiwan* (ICCE-TW), Taipei, 2017, 175-176.

[7] Brownlee, J. (2019). *A Gentle Introduction to Computer Vision*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/what-is-computer-vision/ [Accessed 7 Apr. 2019].

[8] Harvard University Brain Tour. (2016). *A Nobel Partnership: Hubel & Wiesel*. [online] Available at: http://braintour.harvard.edu/archives/portfolio-items/hubel-and-wiesel [Accessed 7 Apr. 2019].

[9] ujjwalkarn (2016). *A Quick Introduction to Neural Networks*. [online] the data science blog. Available at: https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/ [Accessed 7 Apr. 2019].

[10] Ognjanovski, G. (2019). *Everything you need to know about Neural Networks and Back Propagation*. [online] Towards Data Science. Available at: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a [Accessed 7 Apr. 2019].

[11] Mahapatra, S. (2018). *Why Deep Learning over Traditional Machine Learning?*. [online] Towards Data Science. Available at: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063 [Accessed 7 Apr. 2019].

[12] Ollion, C. (2018). *What's easy/hard in AI & Computer Vision these days?*. [online] Medium. Available at: https://medium.com/CharlesOllion/whats-easy-hard-in-ai-computer-vision-these-days-e7679b9f7db7 [Accessed 7 Apr. 2019].

[13] Nielsen, M. (2018). *Neural Networks and Deep Learning*. [online] Neuralnetworksanddeeplearning.com. Available at: http://neuralnetworksanddeeplearning.com/chap2.html [Accessed 7 Apr. 2019].

[14] Agarwal, R. (2018). *Object Detection using Deep Learning Approaches: An End to End Theoretical Perspective*. [online] Towards Data Science. Available at: https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a [Accessed 7 Apr. 2019].

[15] Ouaknine, A. (2018). *Review of Deep Learning Algorithms for Object Detection*. [online] Medium. Available at: https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852 [Accessed 7 Apr. 2019].

[16] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection*. [online] arXiv.org. Available at: https://arxiv.org/abs/1506.02640 [Accessed 7 Apr. 2019].

[17] Hui, J. (2018). *Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3*. [online] Medium. Available at: https://medium.com/jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088 [Accessed 7 Apr. 2019].

[18] Maj, M. (2018). *What is object detection? Introduction to YOLO algorithm*. [online] Appsilon Data Science | End to End Data Science Solutions. Available at: https://appsilon.com/object-detection-yolo-algorithm/ [Accessed 7 Apr. 2019].

[19] Redmon, J., Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. [online] University of Washington. Available at: https://pjreddie.com/media/files/papers/YOLOv3.pdf [Accessed 7 Apr. 2019].

[20] Enrique, A. (2018). *Object detection with YOLO: implementations and how to use them*. [online] Medium. Available at: https://medium.com/@monocasero/object-detection-with-yolo-implementations-and-how-to-use-them-5da928356035 [Accessed 7 Apr. 2019].

[21] Brownlee, J. (2019). *What is the Difference Between a Batch and an Epoch in a Neural Network?*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/ [Accessed 7 Apr. 2019].

[22] Sarkar, D. (2018).*A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning*. [online] Towards Data Science. Available at: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a [Accessed 7 Apr. 2019].

[23] Han, S., Shen, W., Liu, Z. (2016). *Deep Drone: Object Detection and Tracking for Smart Drones on Embedded System*. [online] Stanford University. Available at: https://web.stanford.edu/class/cs231a/prev_projects_2016/deep-drone-object__2_.pdf [Accessed 7 Apr. 2019].

[24] Lee, J., Wang, J., Crandall, D., Sabanovic, S., and Fox, G. (2017). Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles. *2017 First IEEE International Conference on Robotic Computing (IRC)*, pp.36-43

[25] Christiansen, P., Steen, K., Jorgensen, R. and Karstoft, H. (2014). Automated Detection and Recognition of Wildlife Using Thermal Cameras. *Sensors*, 14(8), pp.13778-13793.

[26] Radovic, M., Adarkwa, O. and Wang, Q. (2017). Object Recognition in Aerial Images Using Convolutional Neural Networks. *Journal of Imaging*, 3(2), p.21.