Placement Deliverables

WIL Placement

Jonathan Roos
S5133317

Placement Deliverables

The aim of this document is to outline the deliverables of Project Flying Fox (PFF) and provide justification of the choices made with evidence of how the project objectives were satisfied.

Project Flying Fox had one main outcome that aimed to meet multiple objectives. The catalyst for PFF was to find a solution for counting flying foxes for ecological purposes. The company that instigated the idea of a software solution was counting the bats in these images by hand. This is obviously a very time-consuming and tedious task that is prone to human error, which is the type of job that computers are great for. Therefore, the main objective for the project was to create a proof-of-concept software that was cable of accurately identifying and counting the bats. The secondary objective was to determine if the resulting software had the potential to be developed further for commercial use in the Ecological industry for identifying and categorising different types of animals. Figure 1 shown on the right is an example of the aerial thermal images that we were working with. The distinct dark circles are the bats.

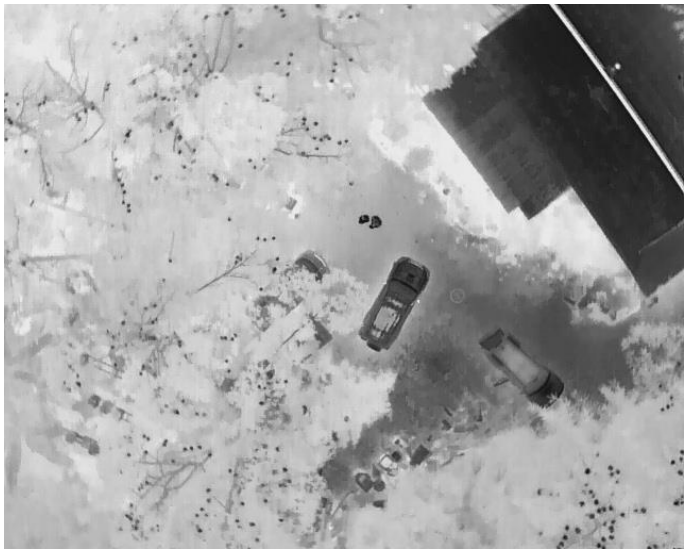

Figure 1

At the begging of the placement, the development team and I held a meeting that discussed ideas of how to approach the problem at hand. It was brought up that the use of machine deep learning could potentially be useful for this use case. However, as no one on the team had a deep understanding of machine learning, we were unsure if this idea was feasible. The first 3-4 weeks of the project was dedicated to researching the best methods to achieve the project goals. This included time used to learn about machine learning and fill in the knowledge gaps. At the end of the 12 weeks, the final project that was delivered to the industry partner was a prototype software that can detect and count the number of flying foxes in trees, using data from aerial thermal images capture by a drone. This was achieved using a Python program that uses computer vision libraries and a convolutional neural network that was trained on data collected from these thermal images. Figure 2 shown on the right is the result of figure 1 being run through the flying fox detector (218 bats found).

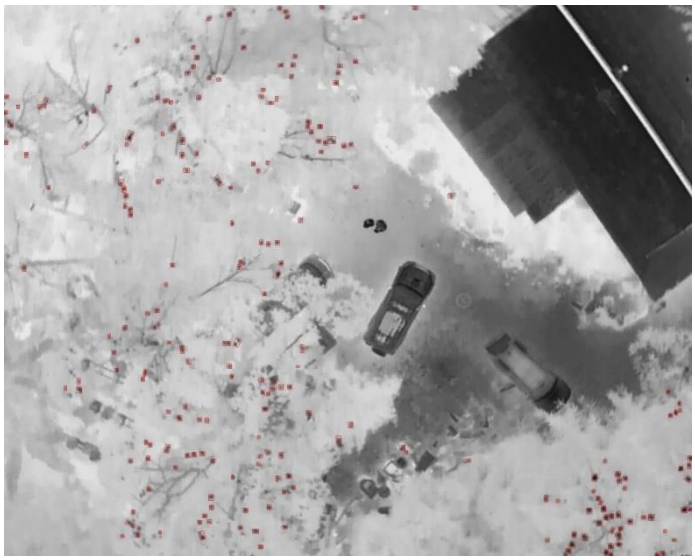

Figure 2

Key Decisions

Some of the key decisions that influenced the projects trajectory included what programming language should be used to develop the software? What frameworks within the chosen software should be implemented? What machine learning framework to use? How to deliver the software to the client or potential end users? Firstly, we chose to develop the project using Python. This decision was made because Python is the industry standard for computer vision, especially for machine learning related tasks. It has a large online community with an extensive range of widely used libraries and frameworks. Python is also the language that I have had the most experience and am most comfortable with. Machine deep learning is also almost exclusively conducted using Python. Thus, it was an easy decision to make as there are essentially no other programming languages that would make more sense to use for PFF other than Python.

Next, we had to decide what methods we would use to solve the issue at hand. The computer vision library we decide to use is called OpenCV. Similarly, to our choice of programming language, this library was also chosen because it is widely used in industry for programming tasks related to what PFF was attempting to achieve. OpenCV is the main brains of the project as it is used to process the thermal images in a way that allows the bats to be detected. In retrospect, I still am happy with the decision of using OpenCV as it had the requirements that we were looking for, and I have not come across any other library alternatives in my research so far. Shown in figure 3 is an example of the processing that OpenCV applies to the thermal images. Figure 3 is the result of a function called adaptive thresholding. This image is then what is used to detect the contours of the potential bats.

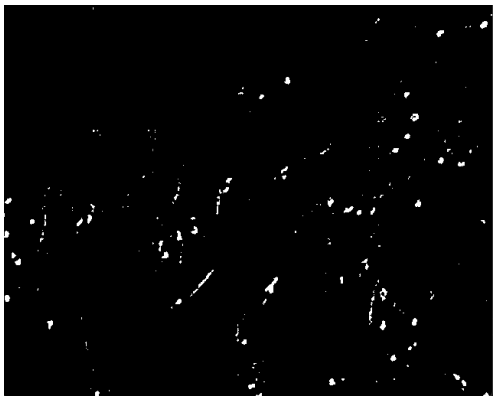

Figure 3

Using OpenCV, we were able to create a software that could detect the bats. However, it also detected many parts of the image that were not bats, things like tree branches for example. Therefore, the next key decision was to make use of a neural network. A neural network is a form of machine deep learning. Essentially, it is a series of complex mathematical equations that 'learns' to recognise relationships in data. The process of 'learning' requires data for the neural network to train on. Therefore, in theory, if we collected images of bats and things that were not bats (e.g. tree branches), we could train the neural network to identify what is and is not a bat. This was the rationale behind deciding to implement a neural network in the project as it would remove any false positives from the bat scan and increase the accuracy. Without this key step in the process, the flying fox detector highly overestimated the number of bats. Thus, it was an integral decision for the project.

Conclusion

In conclusion, the deliverable produced for the industry partner satisfies the requirements and met the objective of the project. To confirm that the project was on track during the placement, a proof-of-concept meeting was held during week 6 of the placement. This gave the industry partners an opportunity to provide any guidance or feedback on the direction the project was heading. This ensured that we didn't stray from the project scope, resulting in a deliverable that met the requirements. The resulting software was not intended to be a polished app that is ready for commercial use. The aim of the project was to produce a prototype that provides evidence that the concept was possible, and that is what was delivered.