# Bee Dance Detection for Optimal Hive Placement and Foraging Efficiency

Eitan Gotian and Andrew Mao
M&T Class of 2026

ESE 3600
May 4th, 2023

## Abstract

This project aims to develop an automated system for detecting bee dances, specifically the waggle dance, to optimize hive placement and improve foraging efficiency. For testing purposes in this project, we have deployed a TensorFlow Lite model on a Raspberry Pi 4 with an OV5647 IR camera to capture and process images of bees performing various dances within the hive. We previously experimented with the Raspberry Pi Zero, but issues with procuring compatible wiring resulted in the switch to the larger Raspberry Pi 4. The system is designed to be low-power, resource-constrained, and suitable for remote areas or large fields with minimal internet connectivity. Our solution has the potential to revolutionize beekeeping practices, promote sustainable agriculture, and enhance pollination and crop yields.


# 1. Introduction

1.1. Bees and Pollination

Bees play a crucial role in sustainable agriculture and biodiversity due to their vital role in pollination. As pollinators, bees transfer pollen from the male reproductive organs of a flower to the female reproductive organs, facilitating fertilization and the production of fruits and seeds. This process is essential for the reproduction of various plants and the maintenance of ecosystems.

1.2. Bee Communication and Dance Behavior

Understanding bee behavior, specifically their unique communication method through dances like the waggle dance, can provide valuable insights into their foraging patterns and help improve beekeeping practices. The waggle dance is a unique form of communication used by bees to convey information about the distance and direction of food sources. By accurately detecting and interpreting bee dances, beekeepers can optimize hive placement and foraging efficiency.

1.3. Traditional Beekeeping Methods

Traditional methods of observing bee behavior and detecting bee dances require manual observation, which is time-consuming and requires expertise. In this project, we develop an automated system to detect bee dances to optimize hive placement and foraging efficiency.

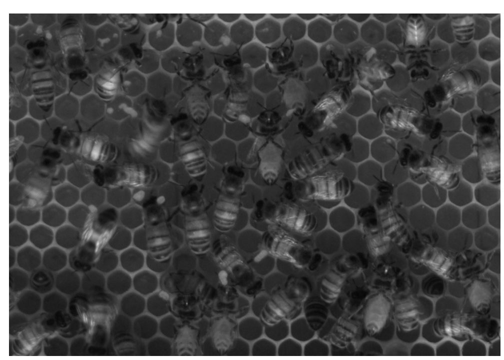
# 2. Motivation

2.1. Sustainable Agriculture
The accurate detection of bee dances can revolutionize beekeeping practices and promote sustainable agriculture by ensuring that bees are strategically placed near high-quality food sources. This can enhance pollination and crop yields while optimizing hive placement for maximum foraging efficiency.

2.2. Technology Advancements
Advancements in image processing and deep learning can be utilized to improve the accuracy of bee dance interpretation, ultimately leading to better hive placement and foraging efficiency. We explored several cutting-edge or popular technologies like u-nets, YOLOv8, and YOLOv5, which exposed us to very different approaches to computer vision. On the image labeling side, we explored the power of RoboFlow for labeling and preprocessing images for machine learning. It enabled us to pre-input at different classes we had and to label faster through its drag-to-fit interface for making bounding boxes.

2.3. Leveraging Other Resources
To produce the physical components for our project, we utilized Penn Engineering's Rapid Prototyping Lab and Additive Manufacturing Lab. This hands-on experience allowed us to acquire valuable skills in laser cutting, 3D printing, and material selection. Additionally, we met with experienced beekeepers who provided us with essential feedback on our hive design and camera placement. During the consultation, we discovered that bees tend to coat foreign objects placed in the hive with propolis. Therefore, we decided to create an interchangeable acrylic "window" that would isolate the tech components while still allowing IR light to pass through. Additionally, we added easy access to the tech components through removable panels while ensuring that the bees could not reach them.

# 3. Dataset Details



3.1. Data Collection
Our dataset consists of frames extracted from video footage captured using a PS2 camera with IR light attachments, which are optimal for detecting bee waggle dances. The dataset includes various dance types performed by bees within the hive.

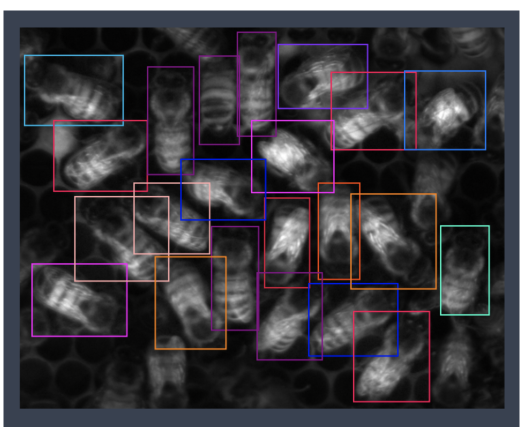
As our project progressed through various conceptual stages, we encountered several different datasets, each presenting its own set of challenges and limitations. Our goal was to find a suitable dataset for detecting and analyzing bee waggle dances in order to gain insights into their behavior within the hive.

Initially, we came across a dataset mentioned in a research paper about bee behavior. This dataset appeared to be an ideal match for our project, and we were eager to employ it. However, upon closer examination, we discovered that the dataset was obstructed and not suitable for use in its current state. This was a major setback, as we had to search for alternative datasets to meet our requirements.

We then found a dataset linked on TensorFlow's website, which appeared to be another promising option. Unfortunately, the link to this dataset was unreachable, leaving us with yet another dead-end. In our continued search for appropriate data, we also considered using pre-annotated datasets to save time and effort. However, we found that these datasets came with limited or no documentation on the output labels, making it difficult for us to decipher what the labels corresponded to and how they would be useful for our project.

Faced with these challenges, we decided to create our own dataset by hand-labeling images. To ensure our dataset was robust and could adapt to various movements and views, we gathered numerous images of bees from a wide range of sources. This approach, although labor-intensive, allowed us to have complete control over the data and its annotations. We could tailor the dataset to our specific needs and ensure that it was suitable for detecting and analyzing bee waggle dances. This proved to be helpful in achieving our project goals. In later parts of this paper, we delve into the process of hand-labeling images and discuss how this decision ultimately contributed to the success of our project.

3.2. Pre-processing

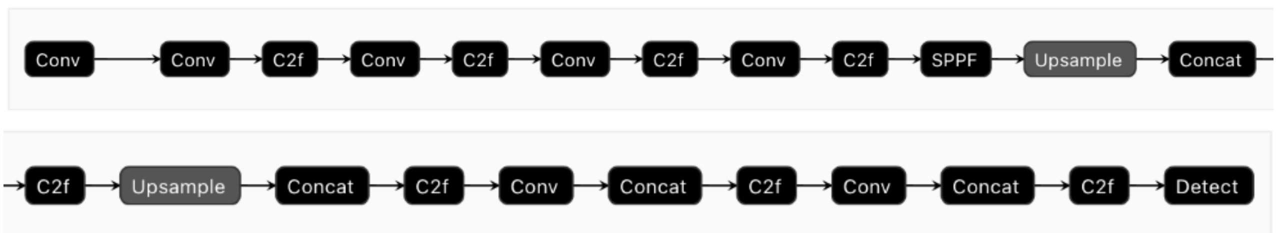


We extracted relevant frames from the video footage and applied image processing techniques to enhance the visibility of bee movements. This involved converting 30 percent of the images to grayscale and adding bounding boxes tightly around the bees to emphasize the bee's shape and orientation. The reason for using some grayscale and some full-color images is due to the fact that our camera also supports full-color images. This means that during daylight hours or non-dark conditions, full-color images would be taken rather than grayscale IR images.

# 4. Model Choice/Design

4.1. TensorFlow Lite Model
Ultimately, we chose a TFLite model because it can be deployed onboard a Raspberry Pi, allowing for onboard data processing with minimal internet connectivity. This is optimal for remote areas or large fields. We experimented with several different custom and pre-built models beforehand, including a custom TensorFlow u-net, a YOLOv8 nano model, and finally, a YOLOv5 small model that was converted to an fp16 TFLite model, which was quantized into an int8 TFLite model.

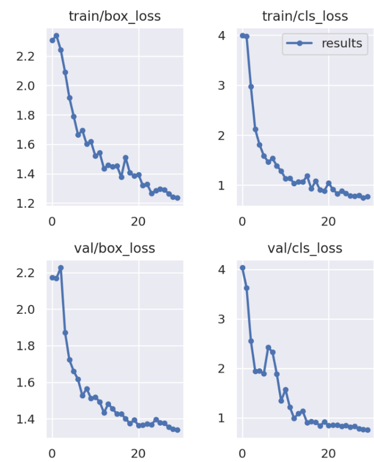
YOLOv5 model architecture:




4.2. Detection techniques
On the techniques side, we experimented with several techniques to identify bees in a hive and detect the vector direction their heads are facing. First, we attempted to use a u-net to detect the bee as a convex hull before using the shape to conduct vector direction, but the technique proved to be too computationally expensive and difficult to implement. From there, we attempted to train a YOLO model on images labeled with their tertiary cardinal direction (N, NW, NNW, WNW, etc.), but the labeling proved to be too labor intensive, and the model also had a difficult time distinguishing certain cardinal directions (i.e., NW and NNW). As such, we decided to train a YOLO model purely for the detection of bees and the creation of bounding boxes. Then, using a custom-written algorithm that used Hough line detection to find the area in a bounding box with the densest amount of lines, signifying it is the part of the image that includes a bee, and then finding the average vector direction of these lines to determine the vector direction the bees are facing. In the end, YOLOv5 was specifically chosen for bee detection because of the built-in package to convert from .pt to .tflite.

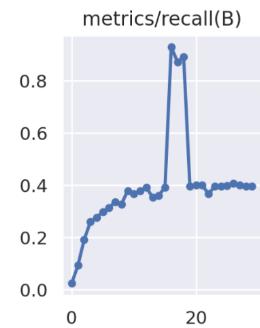
4.3. Raspberry Pi Zero/4
The Raspberry Pi is preferred over Arduino modules due to its convenience and range of data transmission over WIFI, enabling future OTA updates and long-range data transmission to beekeepers for daily or hourly updates. While we tested on the Raspberry Pi 4 due to short-term hardware constraints, in our production version, we look to use a Raspberry Pi Zero (possibly boosted with a Coral TPU) for the compact form factor.

# 5. Model Training and Evaluation Results




5.1. Model Training
We trained a deep-learning model using the pre-processed images to detect bee dances. The model was designed to recognize the waggle dance and other dance types performed by bees within the hive.

5.2. Model Evaluation
The model was evaluated on its ability to detect bees accurately. Our model demonstrated promising results in detecting waggle dances and other dance types, which can be used to optimize hive placement and foraging efficiency. Please refer to the figures to see the relatively low training and validation class loss ("cls_loss"), which we believe is the more important metric in assessing the model since slight abnormalities in the size of the bounding box ("box_loss") are less important than the correct identification of a bee actually existing in a given bounding box (measured by class loss). In addition, we also had an 82.3% recall rate for classification on the validation dataset. It was very difficult to create a metric and system for analyzing the accuracy of the vector algorithm because of its custom nature, so we were unable to collect any quantitative data. Despite this, our model was qualitatively observed to be accurate for a majority of situations when using our demo Raspberry Pi/camera setup to test a video from an iPad, despite some glare from the backlight and the pixelated nature of the images.




# 6. Deployment/Hardware Details

6.1. Raspberry Pi Deployment
We deployed our quantized, efficient model on the Raspberry Pi using a .py script that leveraged cv2 to stream images from the camera, which were then put through the bee detection algorithm to detect bounding boxes. Then, only the parts of the image bounded by bounding boxes were analyzed for bee vector direction. Finally, the bounding box and bee vector direction were rendered using cv2.

6.2. IR Camera and Lighting Setup

We used a wide-lens IR Camera with IR light attachments to capture high-quality images of bee dances inside the hive. The IR camera is capable of capturing images in low-light conditions, making it suitable for observing bees within the dark environment of the hive. The IR light attachments ensure that the bees are not disturbed by visible light while their dances are being recorded.

# 7. Challenges/Future Work

7.1. Challenges

Some of the challenges encountered during the project included creating an effective model to identify multiple bees and their orientation direction, as well as determining the best hardware and techniques for capturing images inside the beehive, as has been discussed extensively in previous sections already (challenges finding the correct model as well as technique for bee and vector detection). On the hardware setup side, ensuring the visibility of bee movements without disturbing the bees or damaging the hive structure was also a significant challenge.

7.2. Future Work

Future work includes field testing and validation of the system in real-world beekeeping scenarios to assess its performance and practicality. We plan to deploy the system in multiple apiaries and gather feedback from beekeepers to refine and improve the model and overall system.

7.3. Model Refinement and Additional Features

We also plan to refine our model to detect when bees are vibrating for better waggle dance analysis, as this can provide more accurate information about food source locations. Due to the difficulty of creating a metric and system for analyzing the accuracy of the vector algorithm because of its custom nature, we will be revisiting techniques that can offer better accuracy in vector labeling, like using a convex hull bounding polygon around the bees. Given the tight bounding of a convex hull bounding polygon around the bee, it would be relatively easy to detect the direction the bee was facing based on the longest "chord" in the polygon.

Additionally, we look to make our model even more efficient by using the knowledge distillation technique to downgrade the necessary resolution of our images (e.g., from 1080p to 720p or 480p) in order to further improve training speed. In order to do this, we will use the current TFlite model to generate the training labels for the student model, which will then be trained with the downgraded resolution images rather than our original 1080p images. On the hardware side, we will be leveraging the compact but powerful Coral TPU to increase the hardware performance of our Raspberry Pi.

Feature-wise, we aim to explore other features that can be extracted from bee dances, such as the quality of food sources or the presence of threats, to provide more comprehensive insights to beekeepers.

After consulting with beekeepers, they suggested we implement a simple yet valuable feature, count the number of varroa in the hive. Implementing a feature to count Varroa mites in a beehive is crucial for maintaining colony health and ensuring honey production. These parasitic mites, identifiable by their smaller size and contrasting color compared to honeybees, pose a significant threat to bee colonies. By monitoring mite populations, beekeepers can detect infestations early, protect colony health, and safeguard honey production, pollination services, and the sustainability of the beekeeping industry. This a niche sector that we can have a significant impact on since there were no widely available commercial products or applications that incorporate an automated feature to accurately count Varroa mites in beehives.

7.4. Dissemination and Presentations

We plan to present our work at conferences and share our findings with the beekeeping community to raise awareness of the potential benefits of our automated bee dance detection system. By disseminating our research, we hope to encourage further development and adoption of similar technologies within the beekeeping industry.

# 8. Conclusion

This project has demonstrated the potential of an automated system for detecting bee dances to optimize hive placement and improve foraging efficiency. By using a TensorFlow Lite model deployed on a Raspberry Pi 4 and an IR camera, we have developed a low-power, resource-constrained system suitable for remote areas or large fields with minimal internet connectivity. Our solution has the potential to revolutionize beekeeping practices, promote sustainable agriculture, and enhance pollination and crop yields. Future work will focus on refining the model, field testing, and validating the system in real-world beekeeping scenarios, as well as disseminating our research and findings to the broader beekeeping community.