Monitoring System for Determining the Infestation Rate of Bees by Mites

By: Oksana Lyhuta

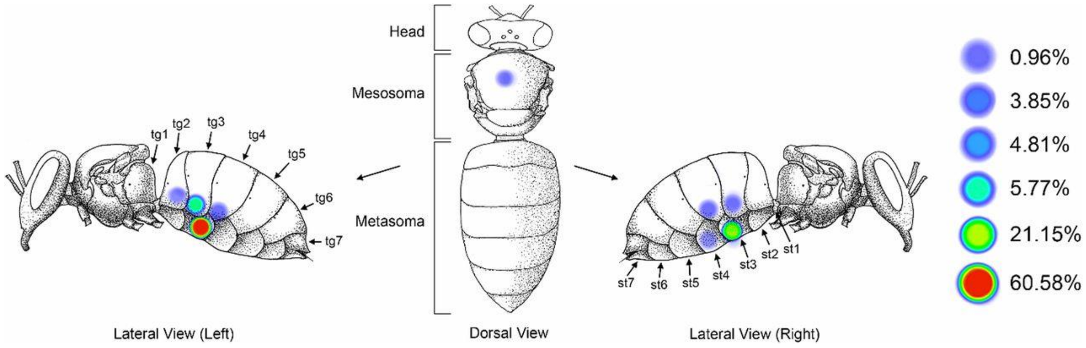
Problem statement

- Honey bees contribute nearly $20 billion to the value of U.S. crop production. Meanwhile, beekeepers continue to lose up to 45 percent of their hives annually. Among the myriad of threats to their hives, beekeepers must continually monitor for the different pests. In the European Union it is endemic, being the only beekeeping disease that requires systematic treatment of bee colonies in order to keep parasitization rates.

- This project aims at protecting bee health by creating a solution that leverages suitable cameras and sensors, allowing users to assess the health and strength of a bee hive and possibly monitoring bee induced activity. One promising avenue for varroa detection is through the use of deep learning techniques applied to images of honey bees.

- By training neural networks to recognize the unique characteristics of varroa mites, researchers and beekeepers could quickly and accurately identify infested colonies and take appropriate action to treat and manage the infestation. The development of such methods has the potential to improve the way beekeepers monitor and protect their hives, ultimately helping to protect the future of honey bees and the vital ecosystems they support.

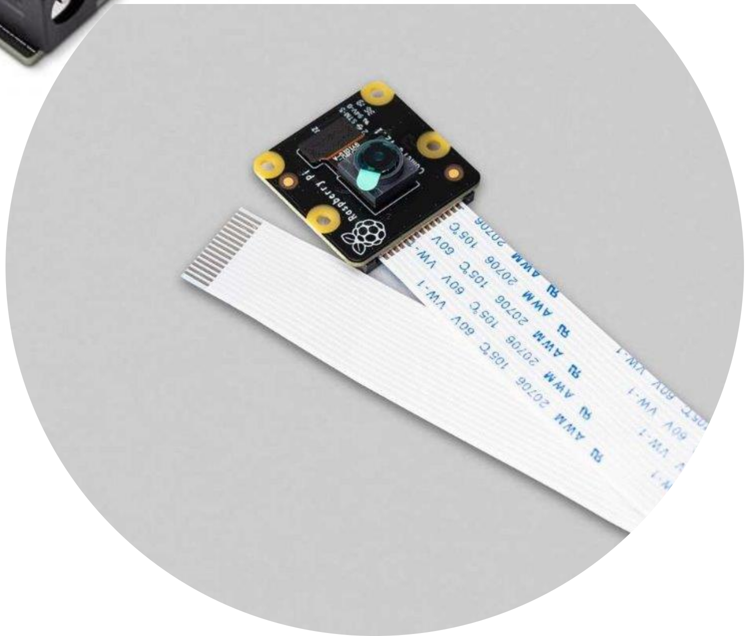
- The parasitic mite Varroa destructor is the greatest single driver of the global honey bee health decline. Better understanding of the association of this parasite and its host is critical to developing sustainable management practices.
- Varroa mites are a major culprit. They are visible to the naked eye and look somewhat like a tick. They feed on the hemolymph of adult bees and the developing brood.
- Varroa mites are super effective vs. honey bees. They drain body fluids directly and transmit deadly virus
- Could be detected and identified visually, but it's still a challenging task, since the size of parasites is too small, and location might differ



Equipment

NVIDIA Jetson Developer Kit has been chosen as the main hardware platform. It can perform constant monitoring and streaming in the different types of environment. NVIDIA® Jetson Nano™ Developer Kit is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing.

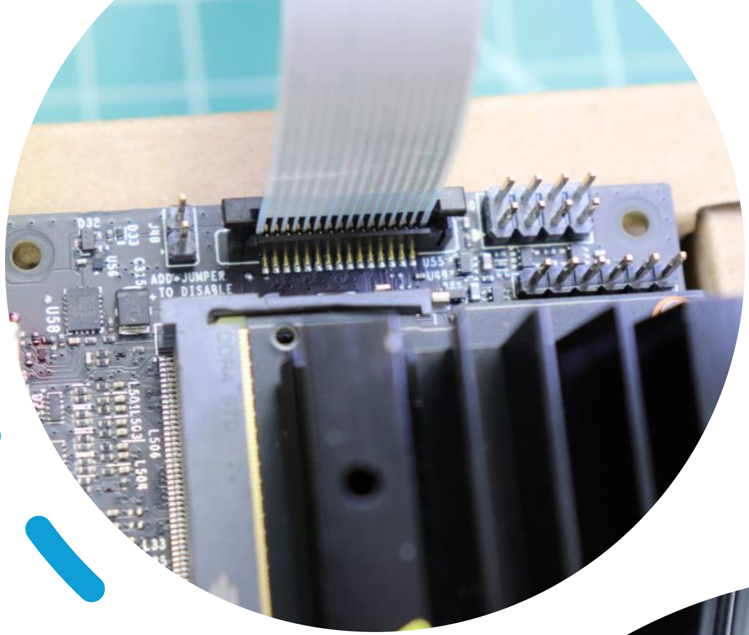
For video capturing we would need cameras that are both compatible with the Jetson Nano's hardware and capable of capturing high-quality video in the conditions of a bee hive. The **Raspberry Pi Camera Module 2 NoIR** here is using (gives the ability to see in the dark with infrared lighting). The camera works with all models of Raspberry Pi that have a CSI connector. Starting with L4T 32.2.1 / JetPack 4.2.2, GStreamer support is built-in to OpenCV.

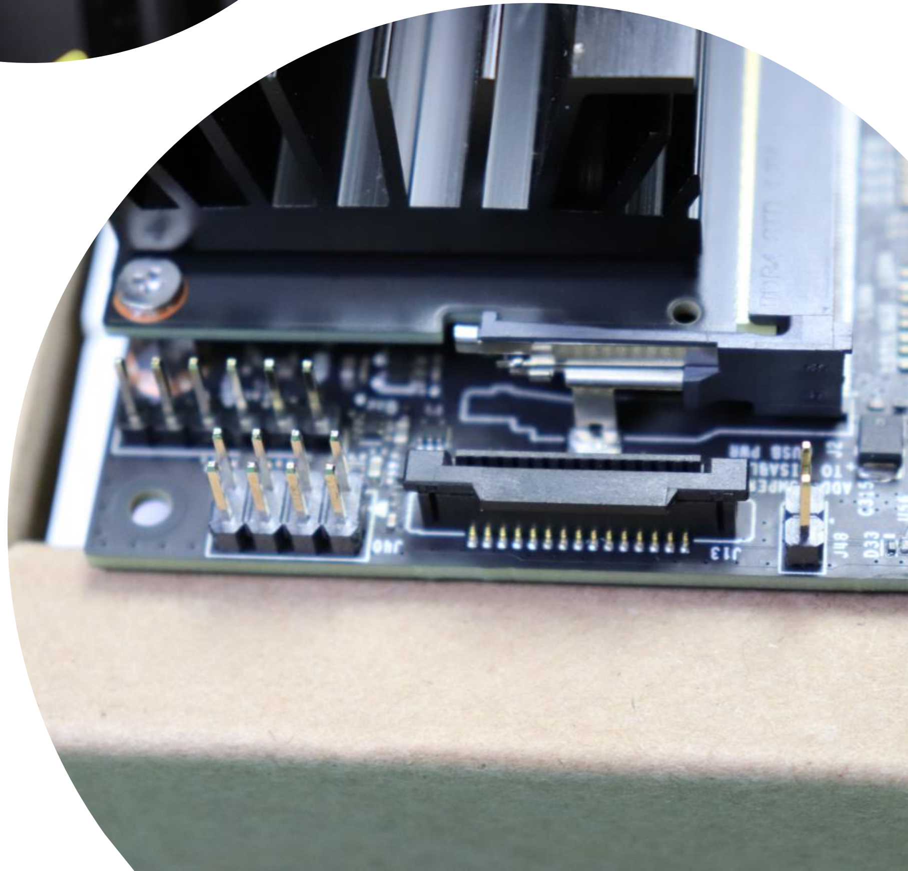


Instructions

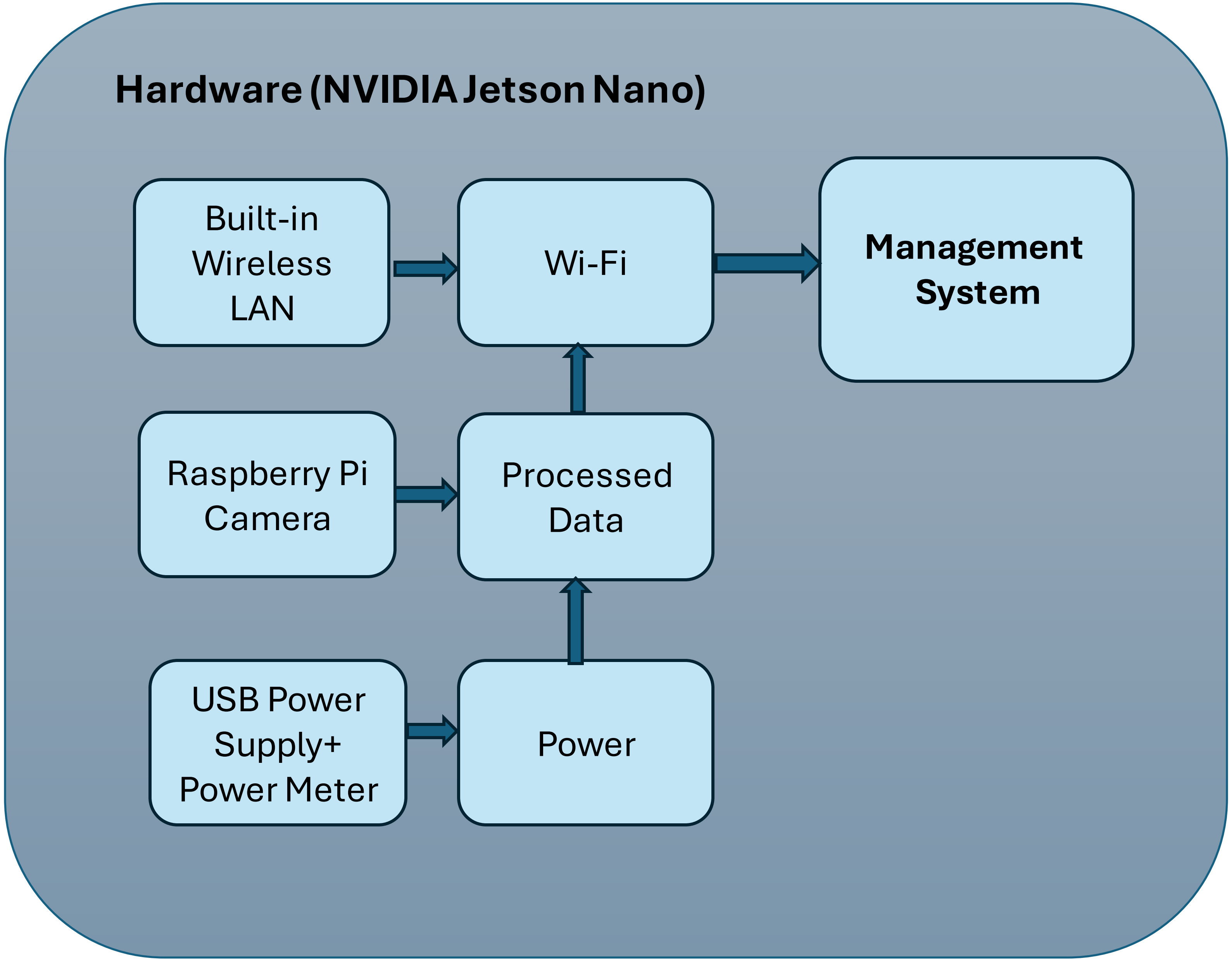
To connect a Raspberry Pi NoIR Camera Module v2 to a NVIDIA Jetson Nano, follow these steps:

- Ensure Compatibility: The Raspberry Pi Camera Module v2 is compatible with the NVIDIA Jetson Nano. It uses a Sony IMX-219 imager and is available in different versions, including the NoIR version which is suitable for night vision applications when paired with infrared lighting
- Connect the Camera: The Raspberry Pi Camera Module v2 has a ribbon connector. Connect this to the Jetson Nano using the CSI-2 camera connector. The Jetson Nano has a compatible connector for the Raspberry Pi camera, making the connection straightforward
- Driver Installation: The Jetson Nano Developer Kit comes with device drivers for the IMX 219 sensor, which is used in the Raspberry Pi Camera Module v2. These drivers are already installed, so you don't need to install them separately. The camera is configured to work out of the box





Schema


Hardware (NVIDIA Jetson Nano)
Built-in Wireless LAN
Wi-Fi
Management System
Raspberry Pi Camera
Processed Data
USB Power Supply+ Power Meter
Power

- Each frame provided by the camera (or video file) will be processed to identify the bees in the image. The found bee positions will then be used to reconstruct the bee movements and paths using Kalman filters. The paths are then used to count the bees entering or leaving the hive. Each detected bee will then be cut from the image, rotated and forwarded to a neural network for classification.
- The neural network performs simple classification tasks to identify varroa mite infected bees. The results can also be visualized. The neural network runs in a separate process and the results may be too late to visualize them, as the bee may have already left the filmed area. But that depends on the performance of the used system.

Algorithms and approaches

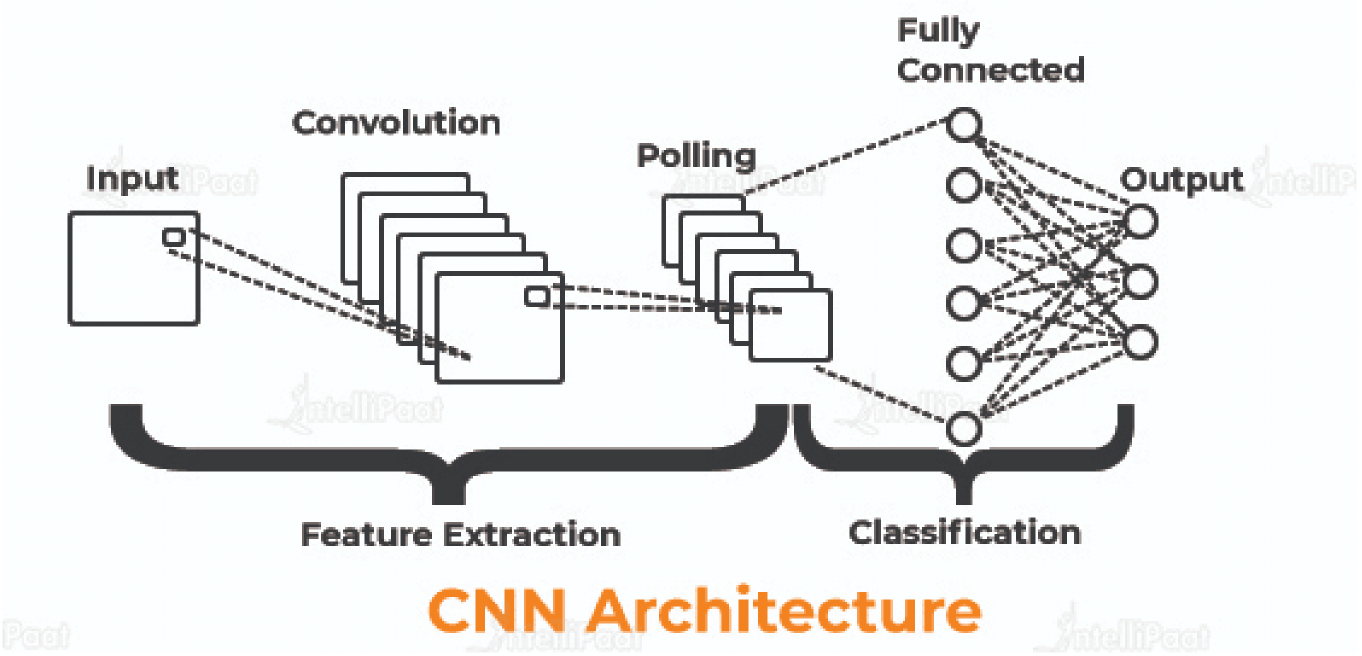
The defined approach is a part of the pre-trained **CNN** model, and **TensorFlow-based CNN algorithm** for classifying images of bees and detecting mites, which after the use of **Kalman filters**, trying to perform simple classification tasks to identify varroa mite infected bees.

The model can be trained over the images, collected by the gathering from the video monitors and then annotated by hand using the LabelImg tool. Other commonly used annotation tools are Labelbox, ImgAnnotation and Computer Vision Annotation Tool, all of which provide an XML annotated output. Meanwhile for providing more accurate results the model of the project has been trained over the predefined **TensorFlow dataset (bee-dataset),** that already contains contains images and a set of labels that expose certain characterisitics of that images, such as varroa-mite infections.

Convolutional Neural Network (CNN) is a type of deep learning algorithm specifically designed for image processing and recognition tasks. Compared to alternative classification models, **CNNs** require less preprocessing as they can automatically learn hierarchical feature representations from raw input images. They excel at assigning importance to various objects and features within the images through convolutional layers, which apply filters to detect local patterns. The weights of the convolutional kernel are shared between the neighbouring layers in the network. The features of a layer are then constructed from previous layers receptive field with the shared weights.

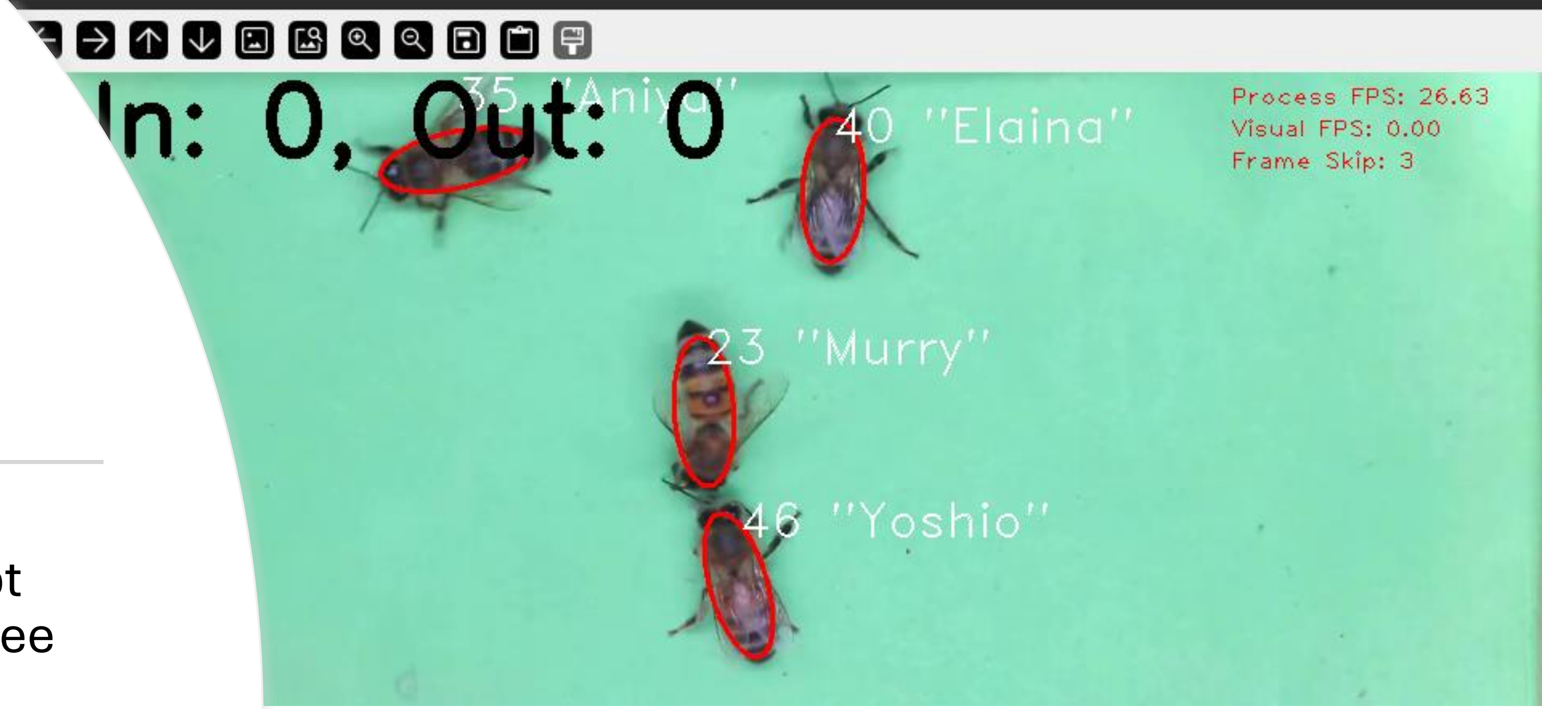
Structure of a typical CNN

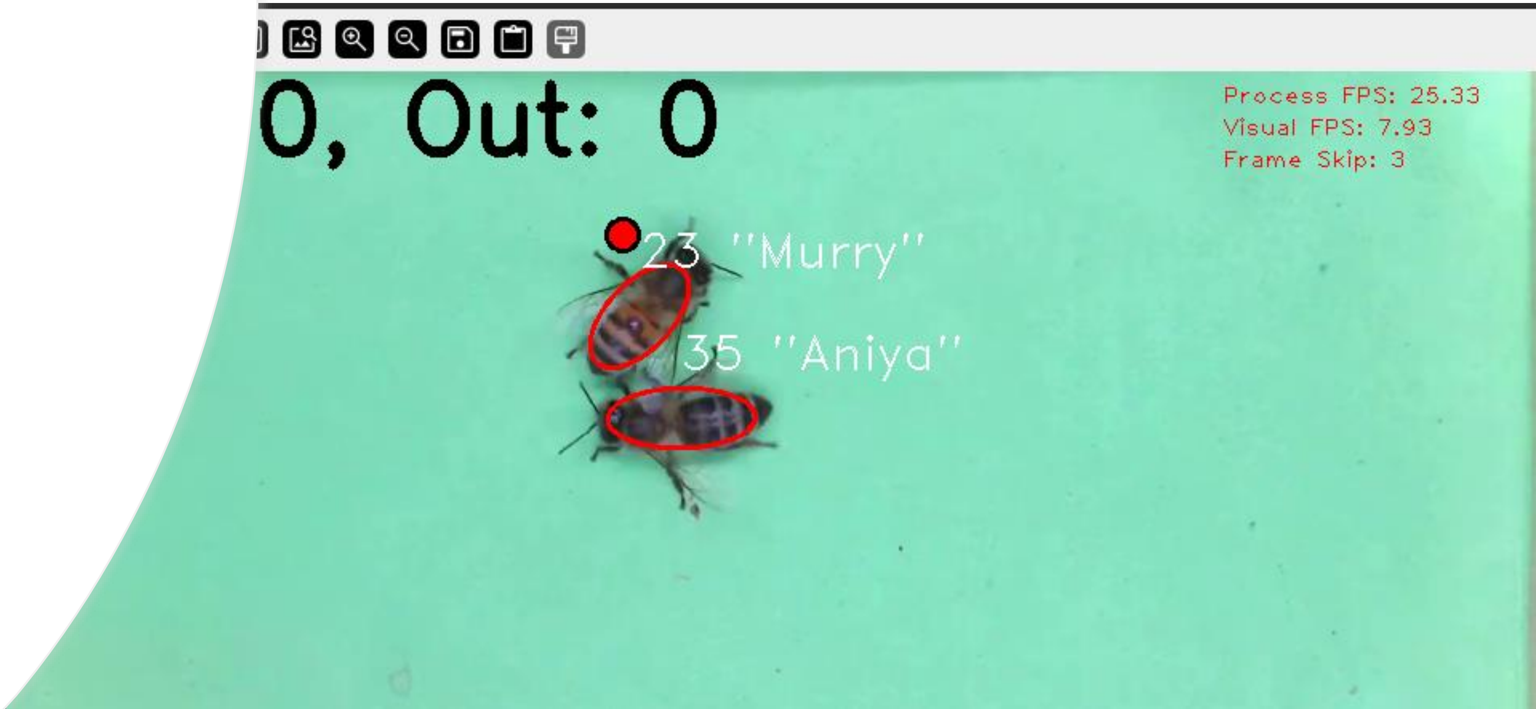
A CNN is built upon multiple different layers. It begins with the input layer followed by multiple convolution and pooling layers. The final layers consist of a fully-connected network and output layer. The most important parameters tuned during training are weights and biases for the corresponding connections between layers. The parameters are tuned with forward propagation and back propagation. CNN training is classified as supervised learning so it needs labeled data as input to the network.



Testing and analyzing results

Results can be tested by running a script over some samples of the video. Each bee is tracked and identified by applying a name, if the varroa is detected, the infected bee will be marked with a red tick.





Conclusion

The proposed system, based on the CNN algorithm, was able to identify and catch varroa mites on the bees.

Meanwhile, some important factors to be considered when discussing the inference results. Since the model has been trained over the shared and prepared set of images for testing, the results may vary when it is prepared over the images/videos, captured in the initial environment.

Most image classifiers are trained with tens or even hundreds of thousands of images to gain confidence in the model. So collecting data for getting better results is crucial here.

Resources

- https://www.extension.iastate.edu/smallfarms/managing-varroa-mites-honey-bee-colonies
- https://beeaware.org.au/archive-pest/varroa-mites
- https://github.com/JetsonHacksNano/CSI-Camera
- https://www.raspberrypi.com/products/pi-noir-camera-v2/
- https://developer.nvidia.com/embedded/jetson-nano-developer-kit
- https://www.pilot-things.com/en/iot-help/lorawan-protocol
- https://www.tensorflow.org/datasets/catalog/bee_dataset
- https://github.com/opencv/opencv
- https://www.tensorflow.org/tutorials