# Real-Time Computer Vision Bird Feeder

**Author**: Tyler Bisk (tjb274) **Advisor**: Hunter Adams (vha3)


## A Single Solution to Many Problems

This project demonstrates the feasibility of low-cost object recognition in real-time, for which there are many applications.

Computer vision can be employed in agriculture, traffic monitoring, medicine, and more. This bird feeder provides easy access to data as close as the the backyard.

The solution found in this project can be reused to solve all these other related problems.

The bird feeder acts as a proxy to all object classification problems. The hardware and software can be easily adapted to be applicable to any situation.

## Bird Detection and Recognition in Real-Time


When a new frame is captured, it is compared to a weighted average of the previous frames to detect motion. If motion is detected, object recognition is performed.

Object recognition is done in OpenCV using a pretrained model called You Only Look Once (YOLO). If the model returns the label “bird” the system starts to record a video of the bird which is saved and emailed to the user in a later step.

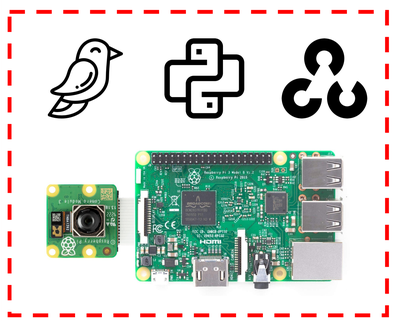
## Alerting the User and Saving Videos

When a bird is recognized, the system begins to record subsequent frames by adding them to a buffer. When the bird flies away, the recording ends and the buffer is encoded to an MP4.

The saved video is immediately sent as an attachment in an email to the user, providing the label and timestamp of the recognition.


Detected a bird at 12:03:38 PM on October 16, 2023

## System Overview



A Raspberry Pi 3 with a Camera Performs Computer Vision Object Recognition on Birds in Real Time Using OpenCV in Python.

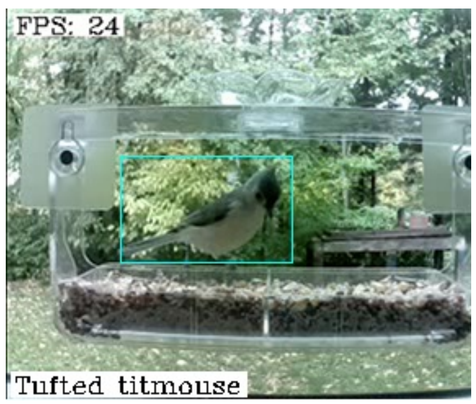
Video feed is broadcast over the internet to be viewed worldwide.

An automatic email containing a video gets sent whenever a bird is detected.

## Identifying the Species of Bird

Upon recognizing a bird in the frame in the previous step, the frame is sent to a custom API that takes advantage of Google Lens’ ability to recognize the exact species of bird in an image.

The custom API, written in JavaScript, uploads the frame to GitHub before querying google for the image that was just uploaded. Then, the API scrapes the label if nonempty and returns it to the main Python loop.

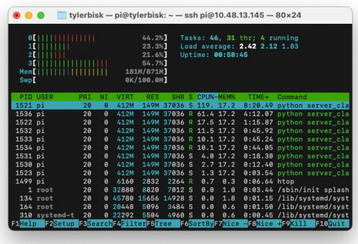



When Google Lens returns the species of bird, the frame and its label is updated, displayed, and emailed to the user.

## Conclusion

The goal of making a low-cost high-performance system that was easy to use and operates in real-time was met. Code optimization, strict memory management, and leaning on Google Lens for heavy computation yielded a Computer Vision system that cost under $60 and is over 90% accurate.

Shown below, the $35 Raspberry Pi 3 comes with 1 GB of RAM and has computation and memory to spare when running the code.




The goal of finding the balance between speed, accuracy, and cost was achieved.

*Thank you to my advisor, Hunter Adams, for offering his guidance, support, and wealth of knowledge to me throughout both this year-long project and my entire college career.*