Service Industry Sentience

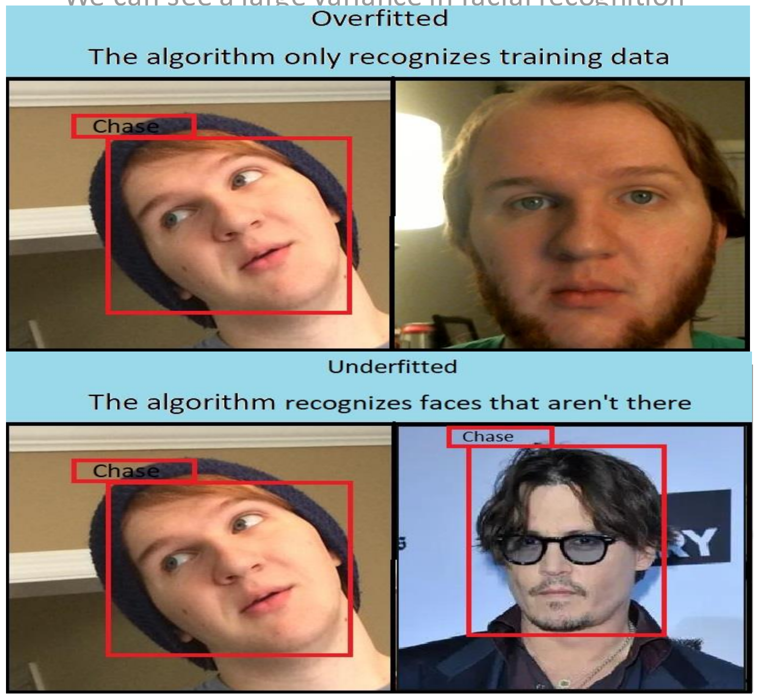
Charles Peery, associated with the College of Computing and Software Engineering, Kennesaw State University.

Introduction/Purpose:

In the face of COVID-19 many businesses have had to go contactless, opting for online payments and delivery options. Smaller businesses often can't afford to make the large changes necessary to implement these systems.
With this research I sought out whether a simple and cost-effective solution could be created for small businesses using facial recognition.

Overfitted Vs. Underfitted

We can see a large variance in facial recognition


Overfitted
The algorithm only recognizes training data
Chase
Underfitted
The algorithm recognizes faces that aren't there
Chase
Chase

Underfitted:
The prediction is too far from the data to make accurate predictions

Overfitted:
This model Is too molded to the training data to be useful for accurate predictions

Proper Weights: A properly trained model can correctly read faces, even with variance such as glasses or facial hair

COCO Dataste: cocodataset.org
OpenCV: opencv.org
YOLOV3: pjreddie.com/darknet/yolo
Darknet: github.com/AlexeyAB

Method Experimental Design:

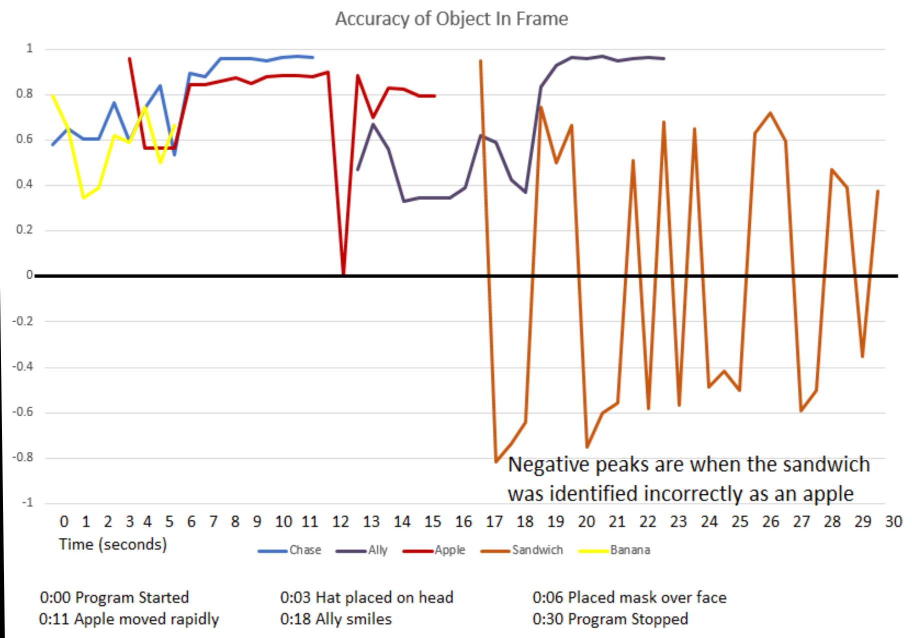
Images of food items were downloaded from Google, pictures were taken of both particpants. Each image was manually labeled, then the images and labels were zipped and uploaded to Google Drive.

Training was done using Google Colab, Darknet, and YOLOv3

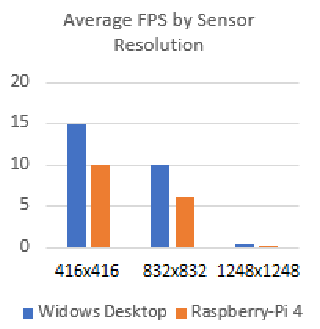
I tried using all completely free options to keep the cost of the completed product down.

Results: After a twelve-hour training period

Confidence: A score assigned to an object representing the likelihood it is that object.



Even with extensive training, models can make mistakes based on environmental variables



The raspberry-pi 4 framerate is too low to be viable with this implementation.

Future Directions

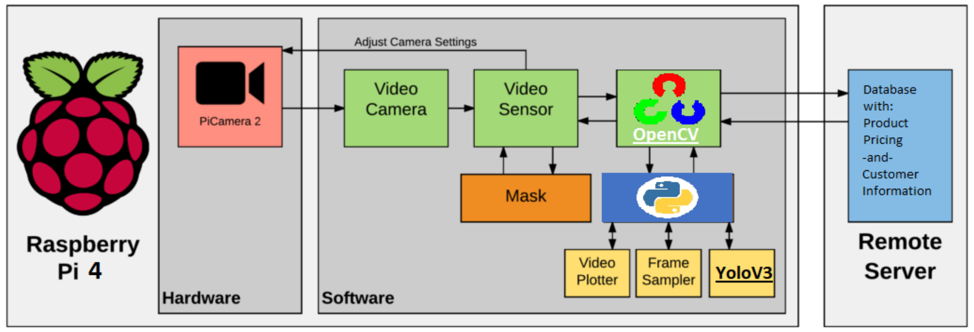
My approach to handling the training and transactions was to place all the necessary objects into a single folder, both faces and food items.

This led to a large amount of overhead data processing that slowed both the training and execution of the electronic point of sale.

To ease the load, I'd like to see two separate detection matrices, one for faces, and one for products. This would allow two separate detectors to be running on the same processor via multi-threading. It would dramatically increase the performance of this system.

Also, due to time restrictions I only tested YOLOv3, but YOLO fast may be a better solution for this system as it allows for faster framerate capture.

Summary/Graphical Diagram



The web camera passes an image into the video sensor, which is then captured by CV2 and processed by the Darknet algorithm for YOLOv3.

The image, post processing has boxes assigned to each object above a certain confidence threshold, these are the objects which are processed by the Electronic Point of Sale.

Once the transaction is processed, via an excel spreadsheet simulating a remote server, it modifies inventory data and checks out the customer.