



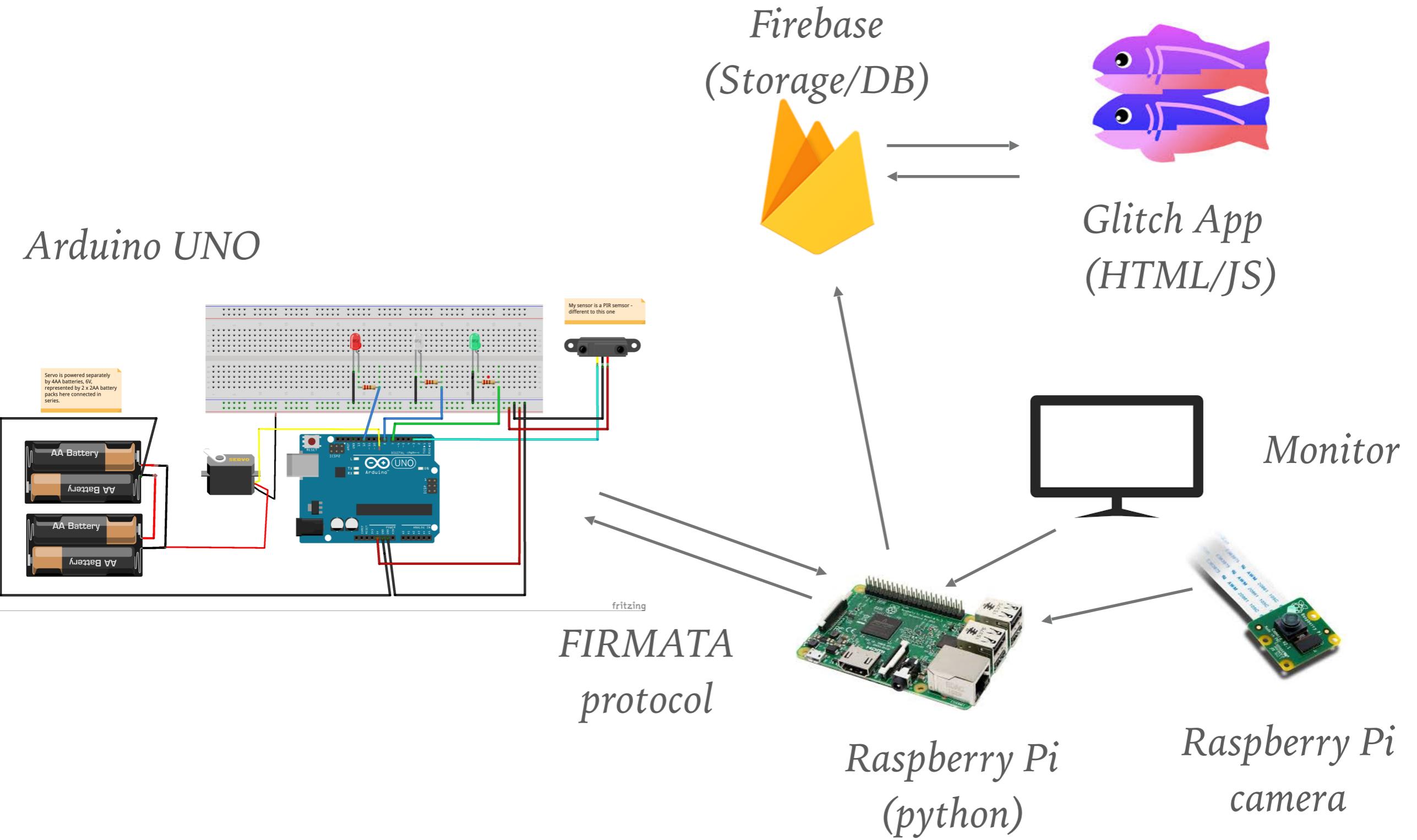
MASKIT

Aileen Drohan

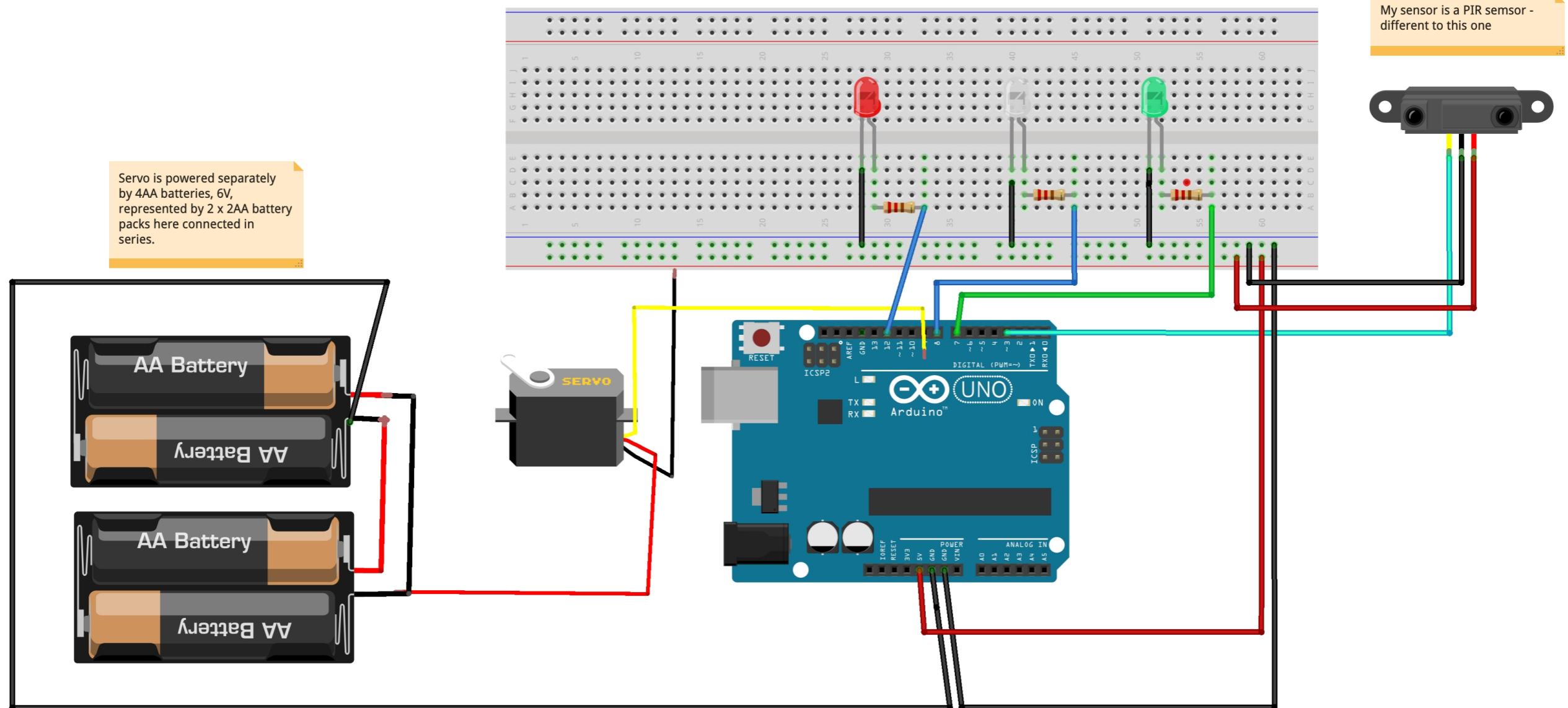
KEY UPDATES ON INITIAL PROPOSAL

- Assign the computing intensive tasks to the Raspberry Pi and the sensor controlling tasks to an Arduino.
- Use Raspberry Pi 4 with 32GB SD card and an Arduino UNO
- Communicate between the Raspberry Pi and Arduino UNO with PyFirmata Protocol
- Use IoT Platform Firebase
- In addition to the IoT business solution proposed, I would like to create a glitch app that utilises the data gathered with a creative twist, essentially forming form the beginnings of an art piece. The overarching idea would be to create a work derived from navigating and reflecting on the following: surveillance, sousveillance, data protection, data collection, data representation or misrepresentation, technological limitations. While completing this art piece is not within the scope of this project, a small experimental piece will be included, expressed through the visual output of the realtime image updates to the Glitch app.

PICTORIAL REPRESENTATION

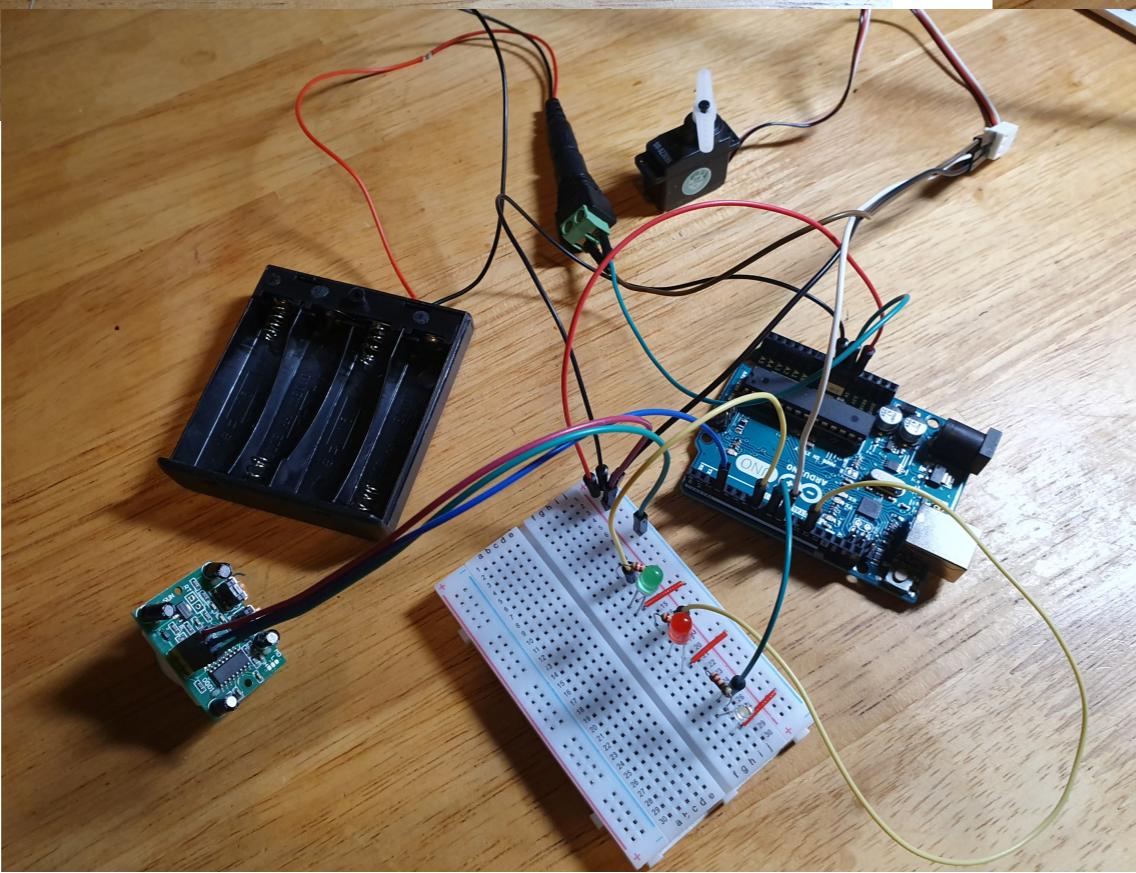
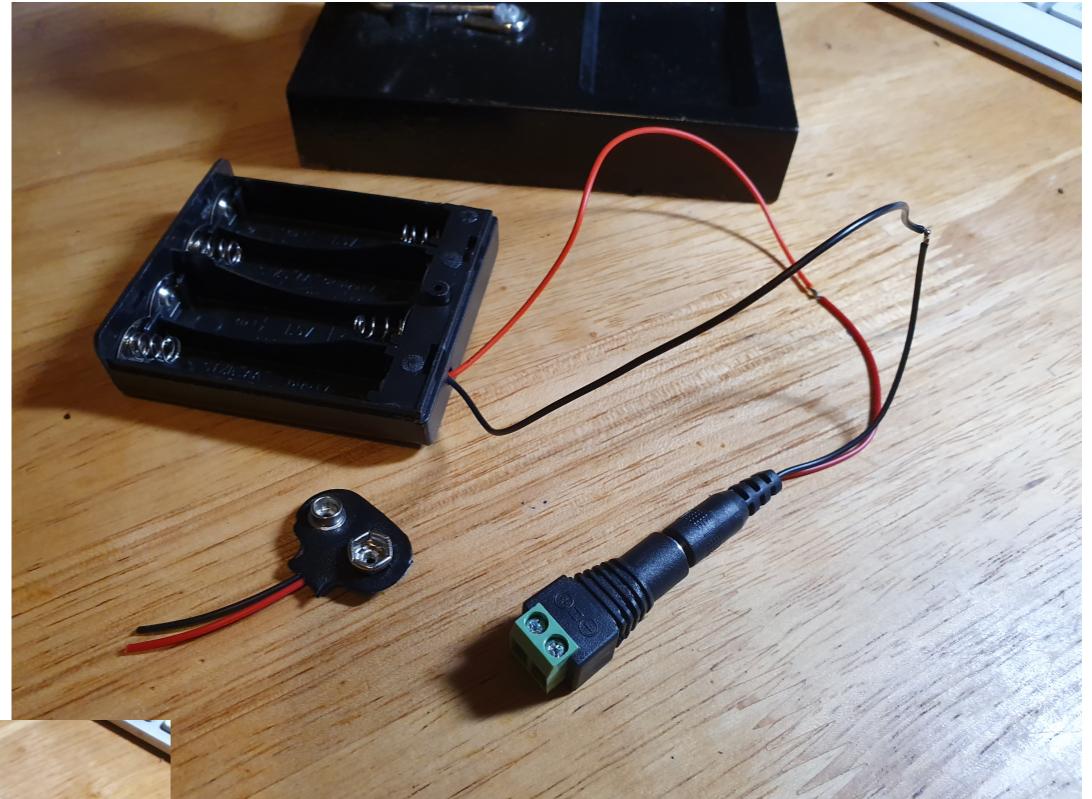
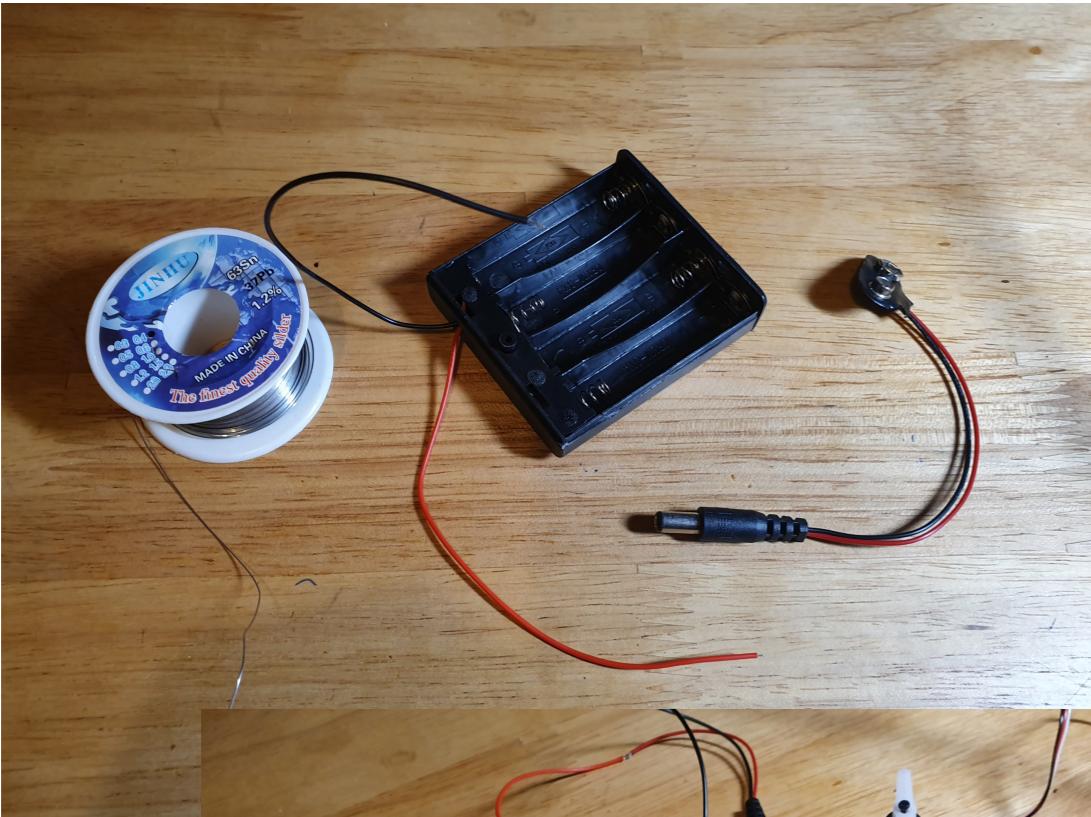


ARDUINO & COMPONENTS SETUP



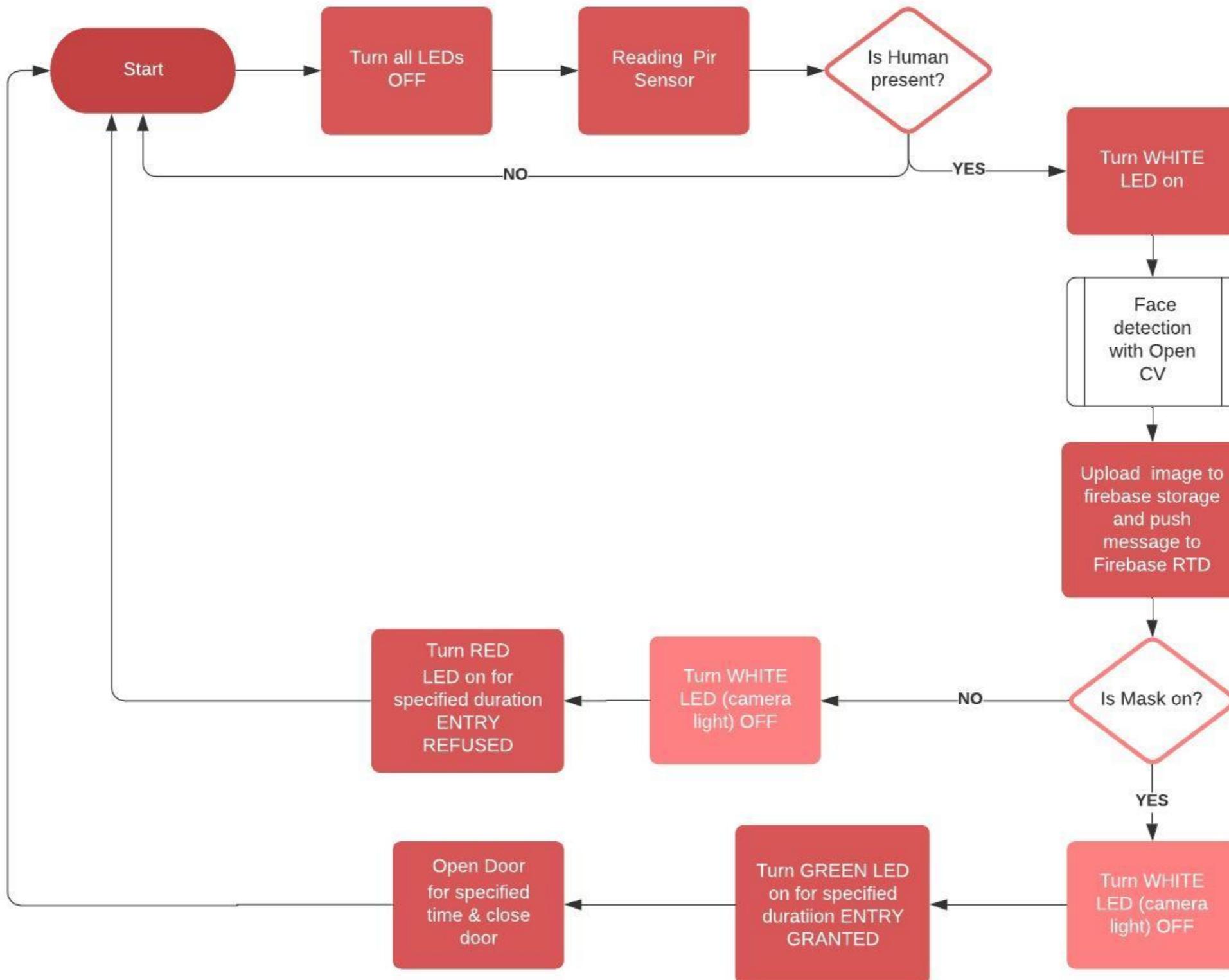
fritzing

SERVO SETUP FOR EXTERNAL POWER

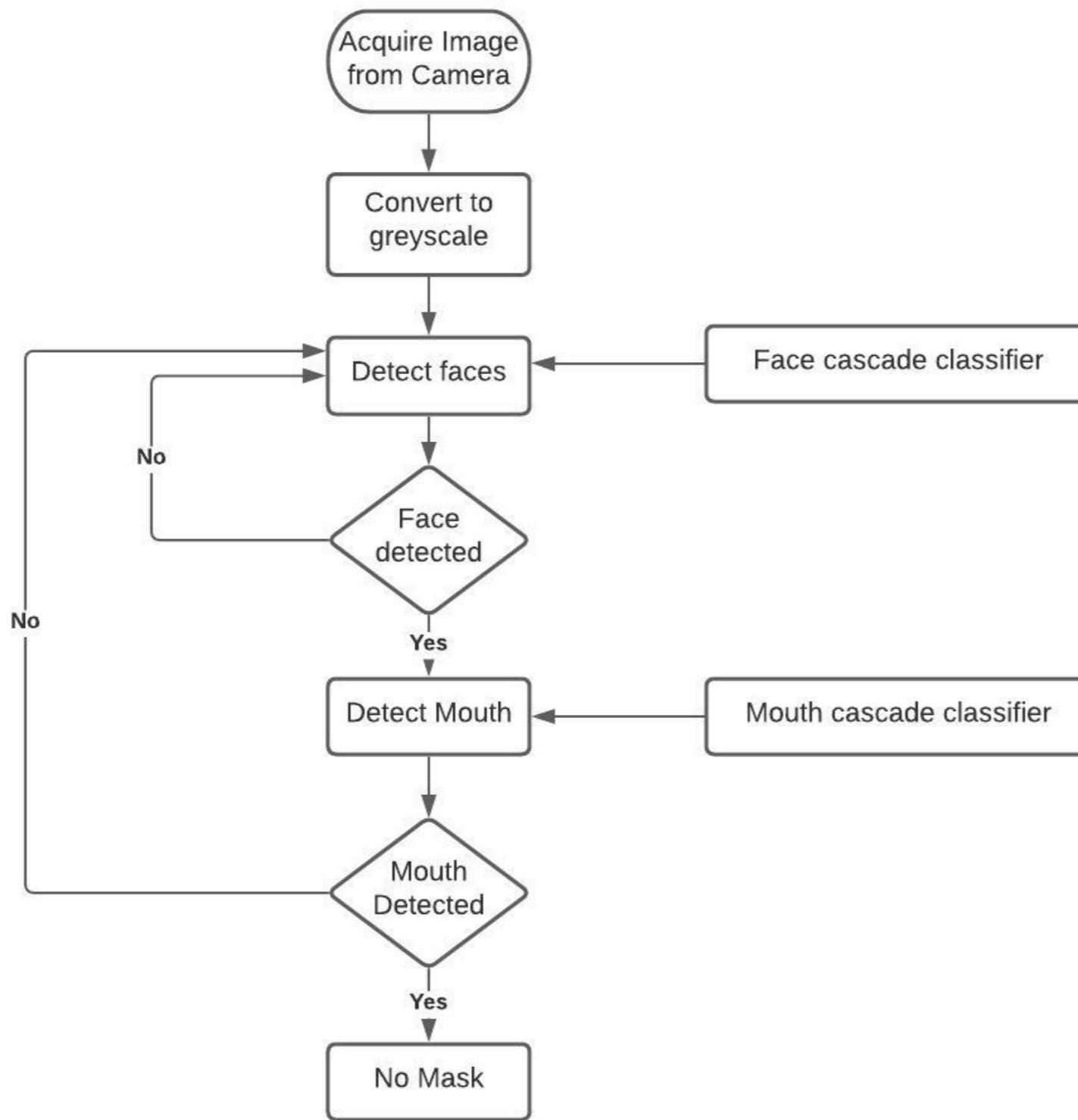


Ideally a 4xAA battery pack with a jack connector would be ideal. I actually cut one off a 9v battery clip and soldered it to the 4xAA battery holder which was then plugged in to the power adaptor block used to provide external power to the servo motor.

FLOW DIAGRAM



FACE DETECTION METHOD #1



SETUP

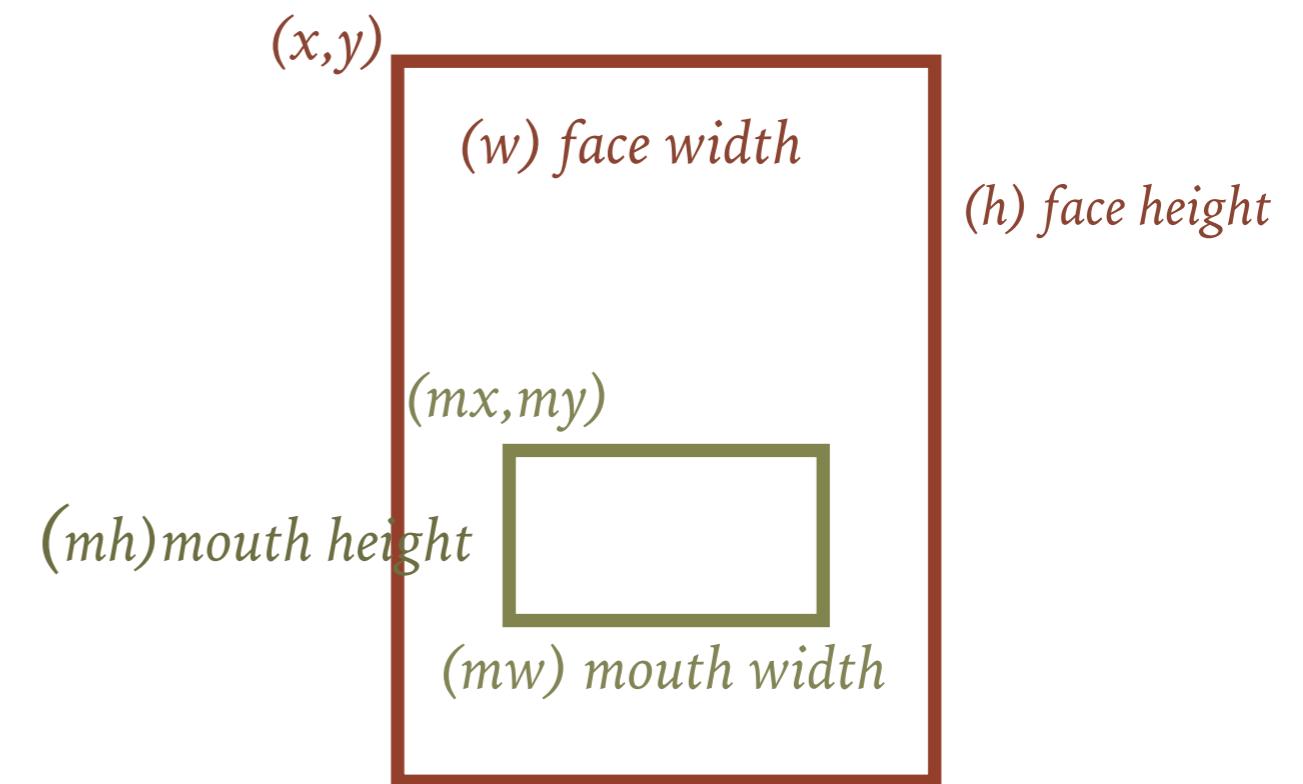
Face Mask Detection using Open CV with frontal face and mouth Haarcascade Classifiers (Viola Jones algorithm)

1. Detect facial features using the face cascade classifier (included as a default classifier with OPEN CV)
2. Detect mouth features using the mouth cascade classifier (Not included as a default classifier in OPEN CV)
3. Mouth Classifiers can be found online
e.g. <http://alereimondo.no-ip.org/OpenCV/34>

FACE DETECTION METHOD #1

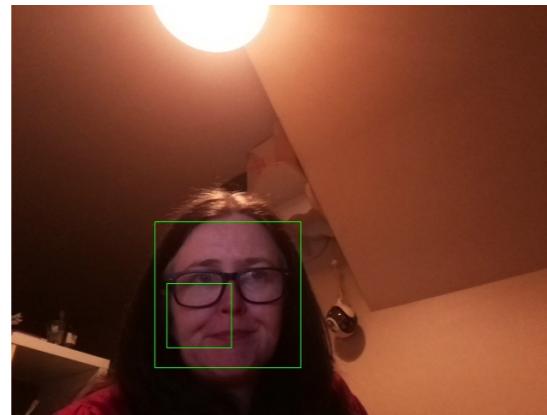
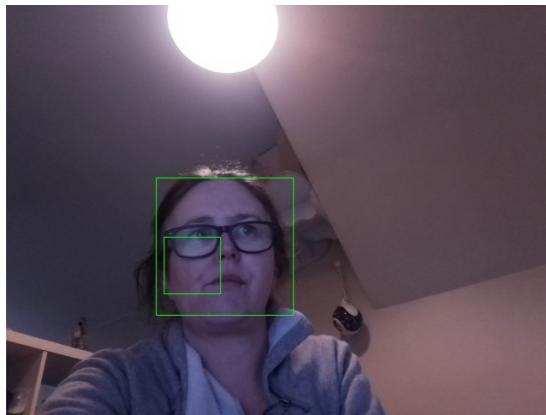
Identify Human face and mouth in each frame

Face Detected	Mouth Detected	Output
If the length of the faces list is > 0 there is a face detected.	If the length of mouth list == 0 that person is wearing a mask.	Mask on
If the length of the faces list is > 0 there is a face detected.	If the length of the mouth list is > 0 and $y < my < y + h$ there is a mouth detected.	No Mask

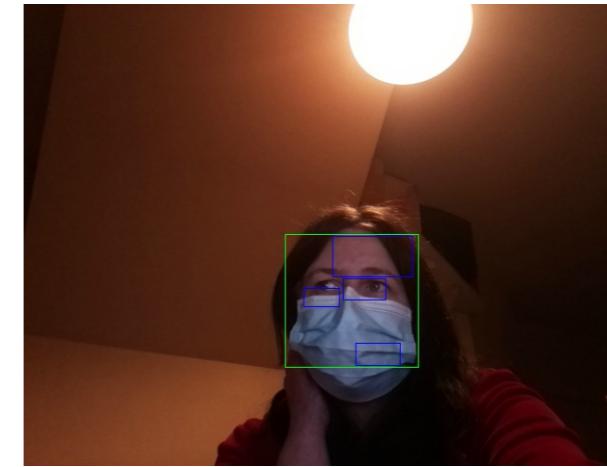
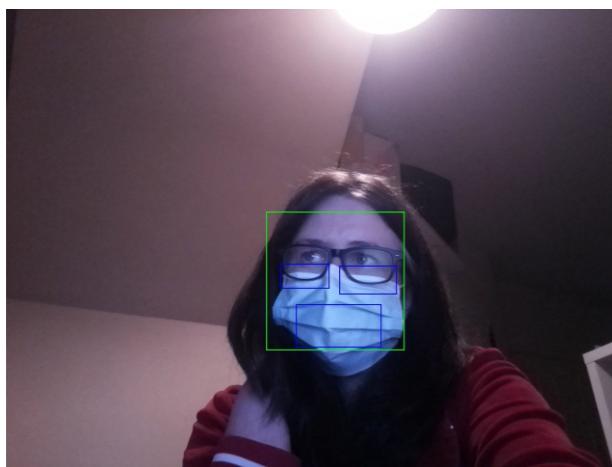


FACE DETECTION METHOD #1 - TESTING

- When wearing a mask the default frontal face hard cascade will still detect a face. False positive sometimes happen but not that frequently.



- In opencv there are no haar cascade mouth detection files included and none of the mouth detection csv files I downloaded from the internet worked.
- There is however a smile detection haar cascade included in opencv. I attempted to implement my project using the smile cascade instead of the mouth cascade. Unfortunately the smile cascade returned multiple false positives with eyes, glasses and mask areas detected as a smile.



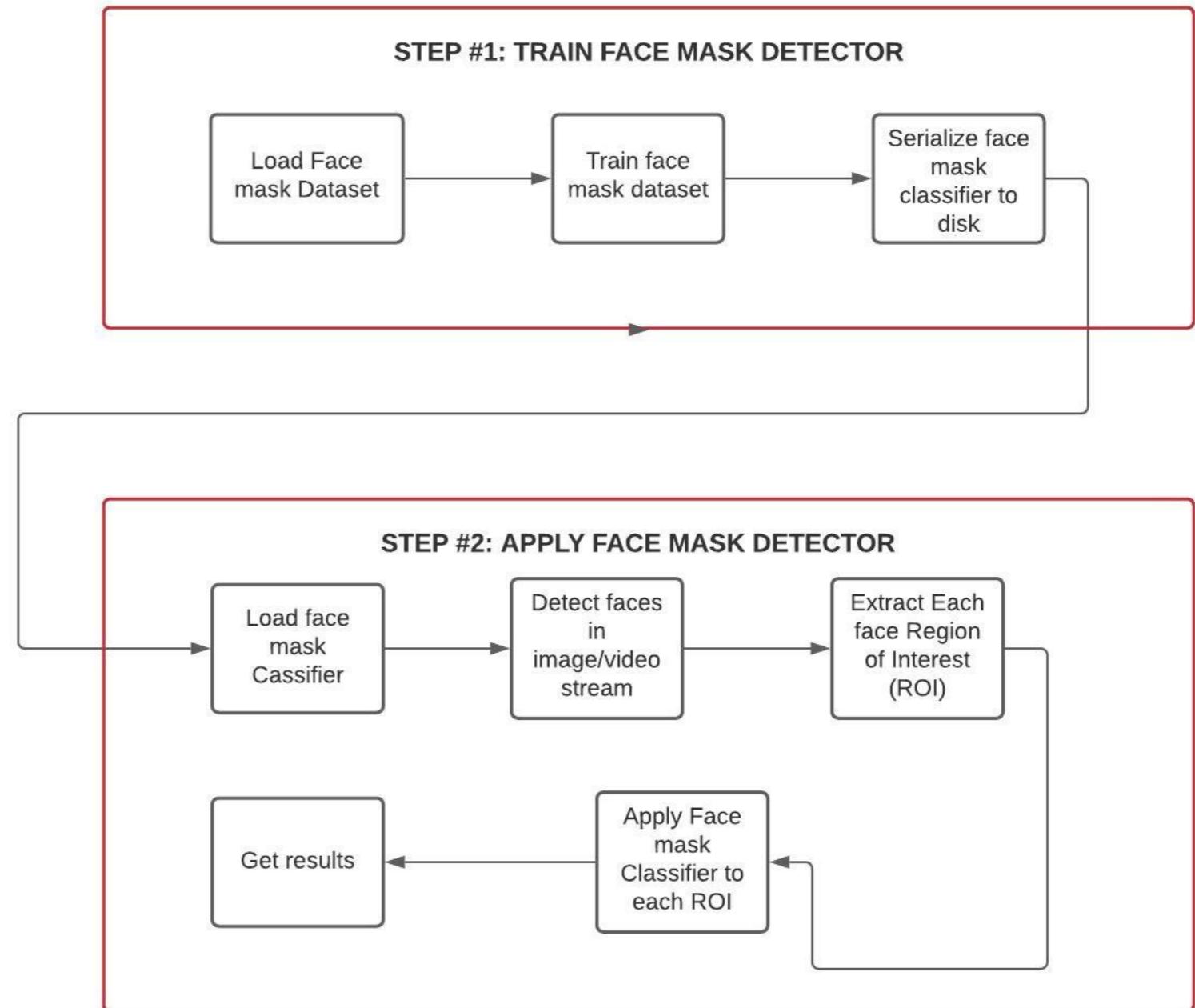
FACE DETECTION METHOD #1 – CONCLUSION

- CONCLUSION: I have found that using haar cascades is an unreliable method due to the false positive detections that occur. Because of this implementing method 1 became fruitless as in most cases a number of smiles were detected around the eyes, glasses and indeed the actual facemasks rendering it impossible to use this method to distinguish mask wearers from non-mask wearers.

FACE DETECTION METHOD #2 – UNIMPLEMENTED

Identify mask in each frame using a trained face mask detection classifier

1. *Source dataset*
2. *Train dataset*



TOOLKITS

Physical Hardware	Communication Protocols	Programming Languages	IDE's & Libraries	Cloud Platform
<ul style="list-style-type: none">• Raspberry Pi 4• Raspberry Pi Camera• Arduino Uno• Breadboard• LED's - Red, Green and White• PIR sensor• Servo Motor• Monitor• Jumper wires• 4xAA (6V) battery pack• Female DC Power adapter 2.1mm jack to screw terminal block	<ul style="list-style-type: none">• Wifi wireless protocol• Web socket protocol• Firmata protocol	<ul style="list-style-type: none">• Python• Arduino C++• JSON• HTML• Javascript	<ul style="list-style-type: none">• Arduino IDE• Open CV computer vision library• PyFirmata	<ul style="list-style-type: none">• Firebase• Glitch App