



# MASKIT

---

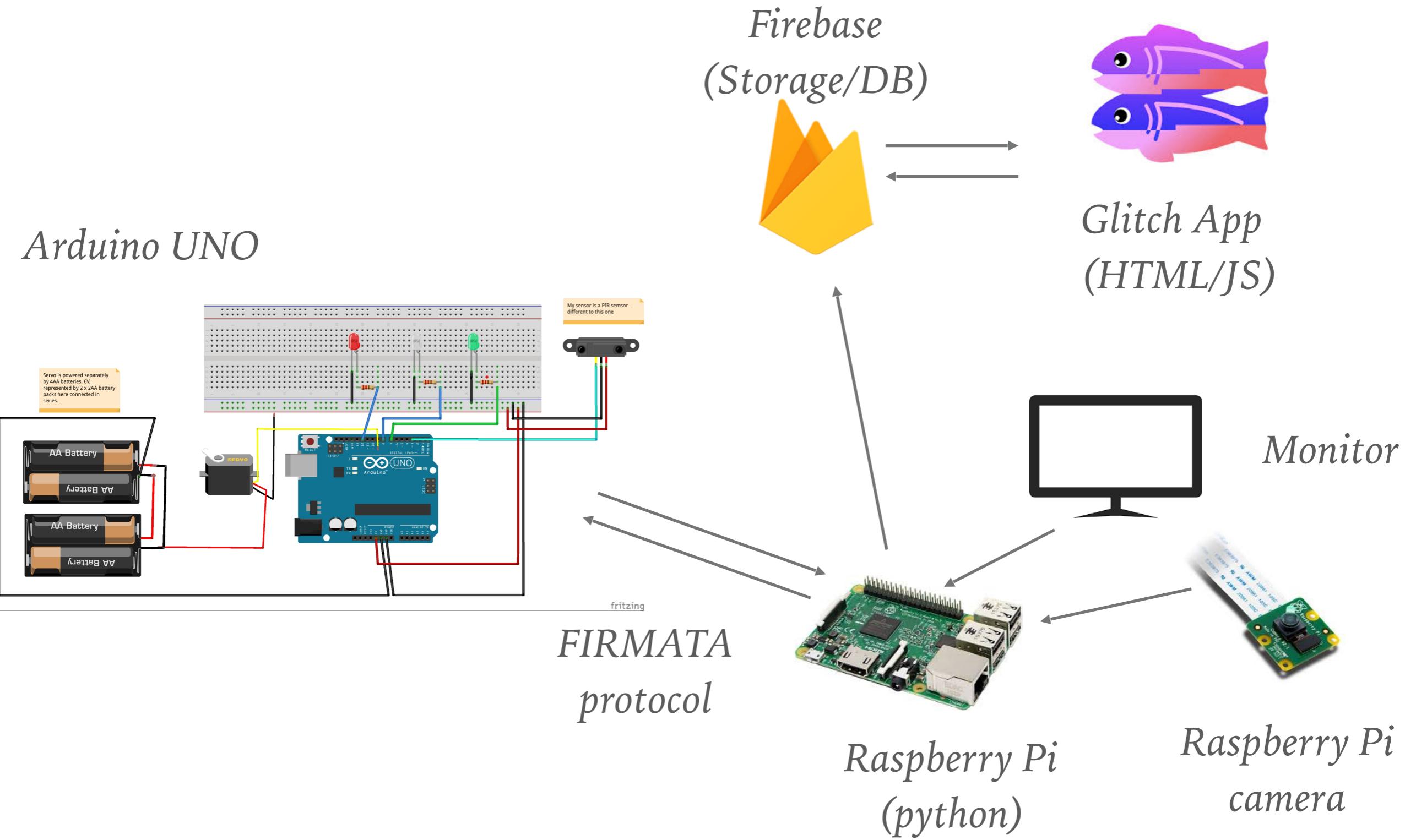
*Aileen Drohan*

# 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 to 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

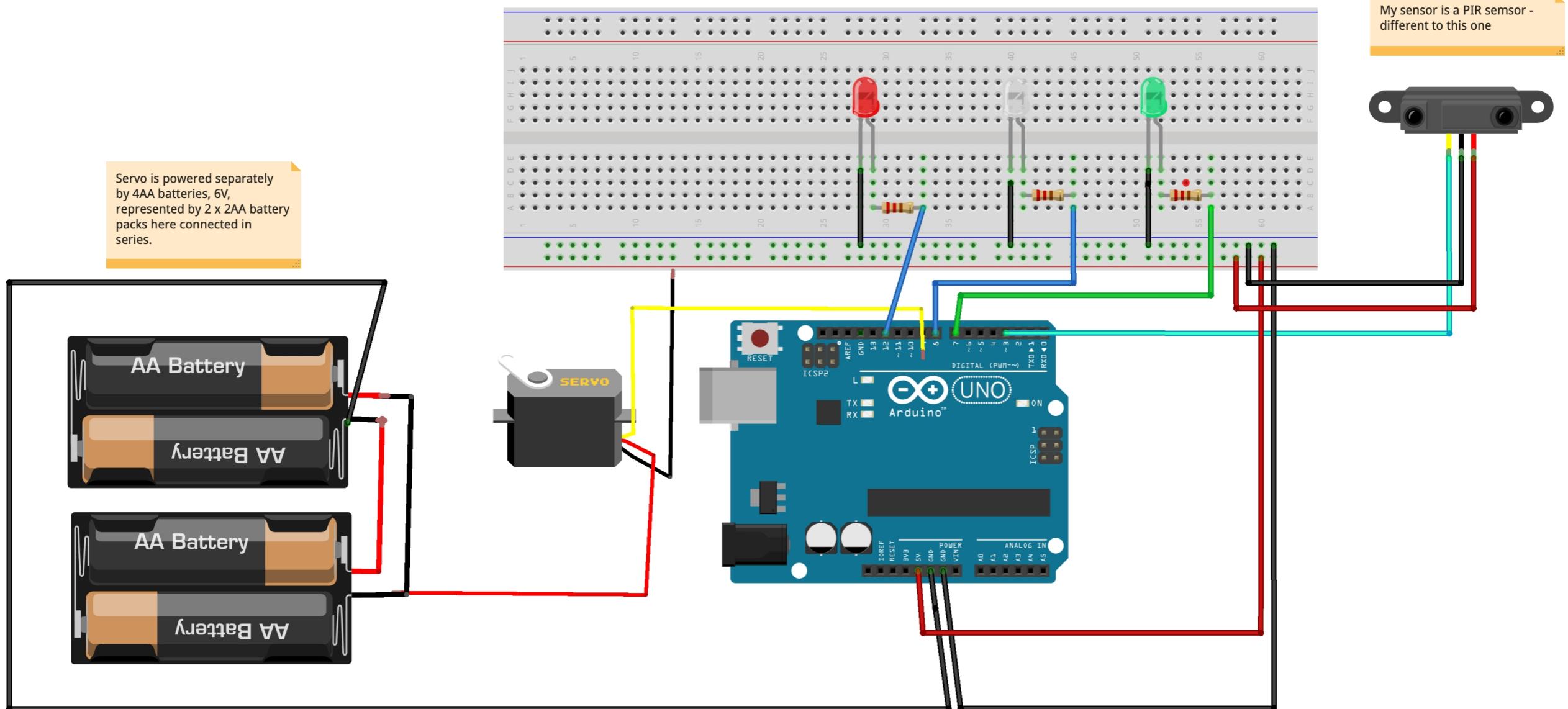


# TOOLKITS

---

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

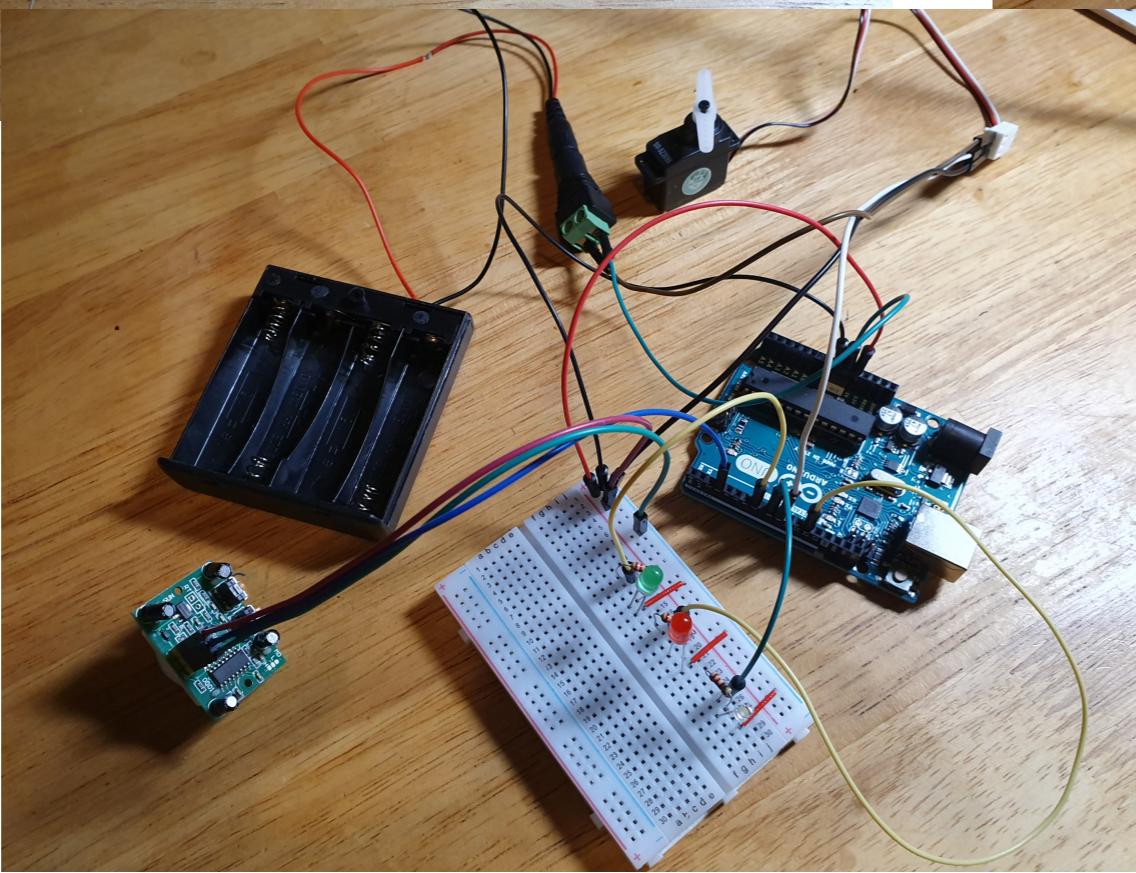
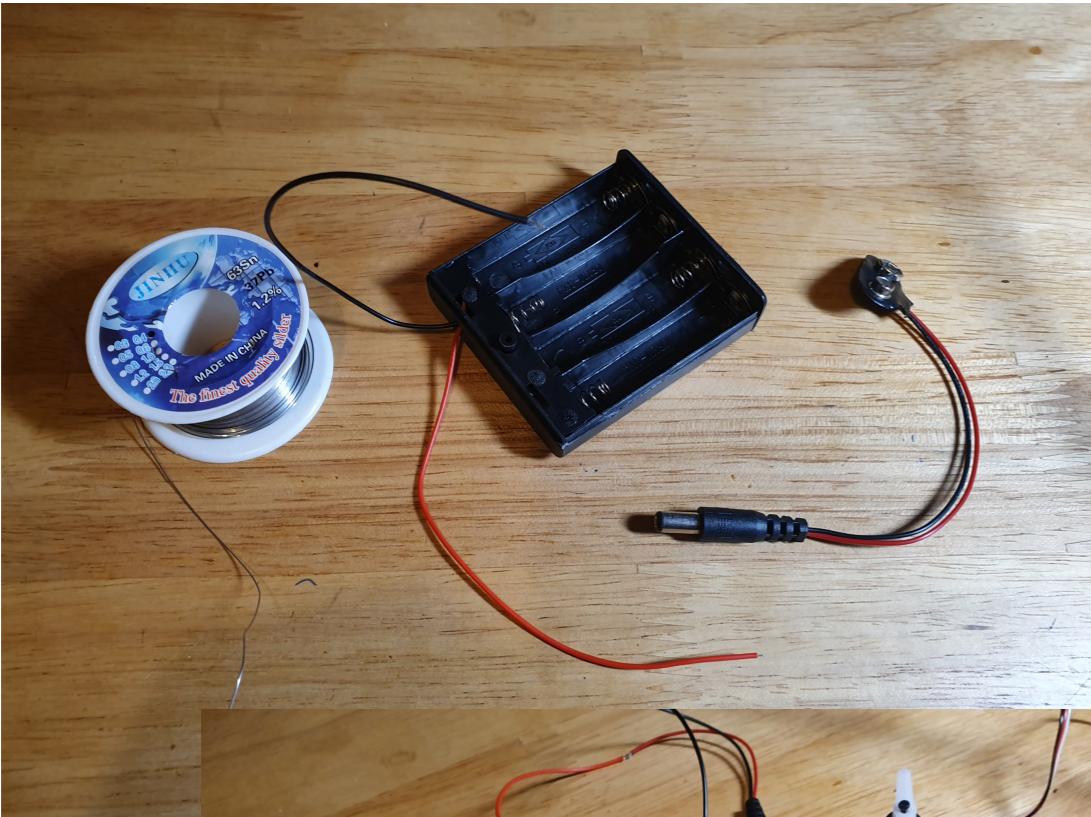
# ARDUINO & COMPONENTS SETUP



fritzing

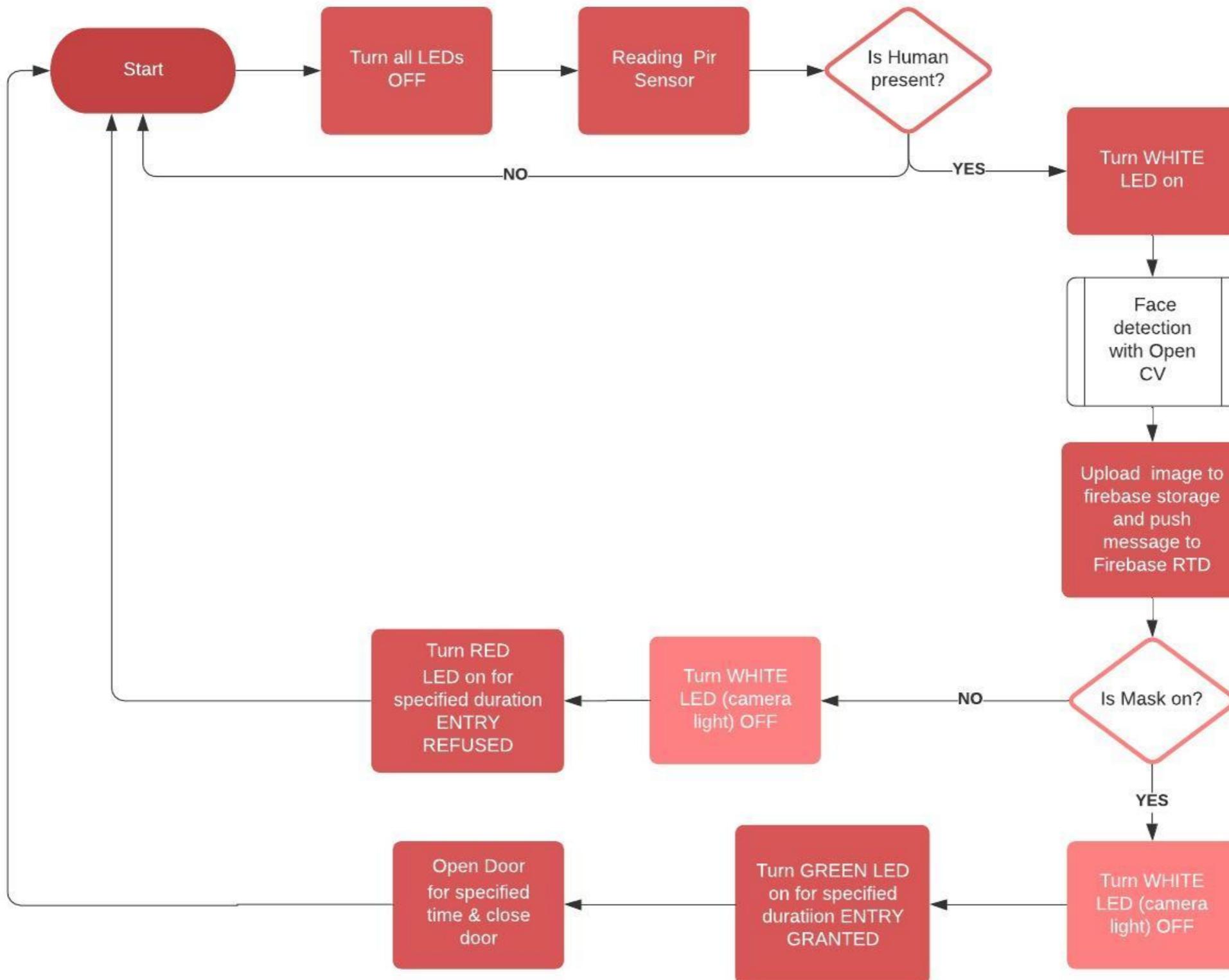
# SERVO SETUP FOR EXTERNAL POWER

---



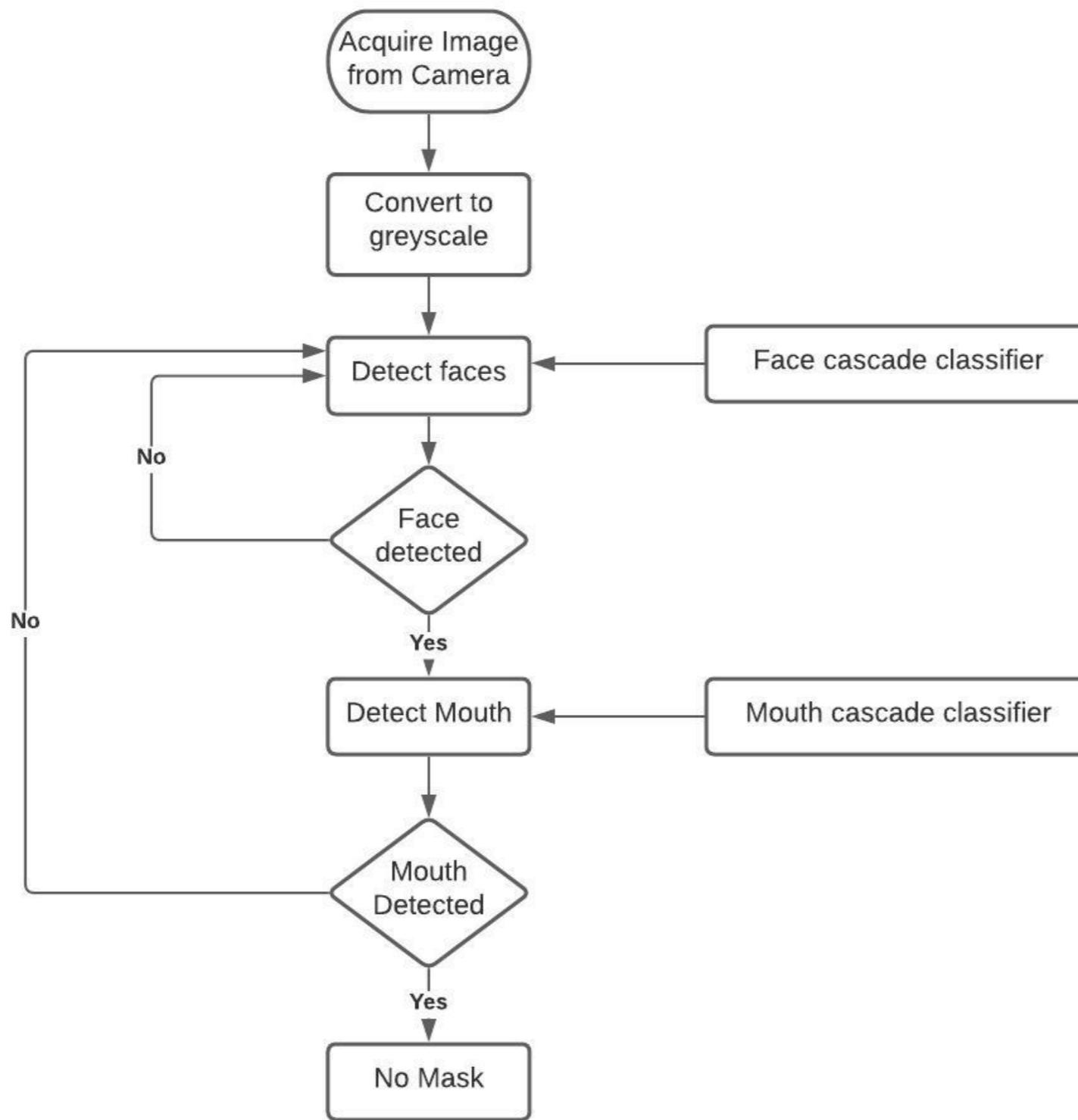
Ideally a 4xAA battery pack with a jack connector would be ideal. I actually cut one off a 9v battery clip and soldered it to the 4xAA battery holder which was then plugged in to the power adaptor block used to provide external power to the servo motor.

# FLOW DIAGRAM



# FACE DETECTION METHOD #1

---



## SETUP

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

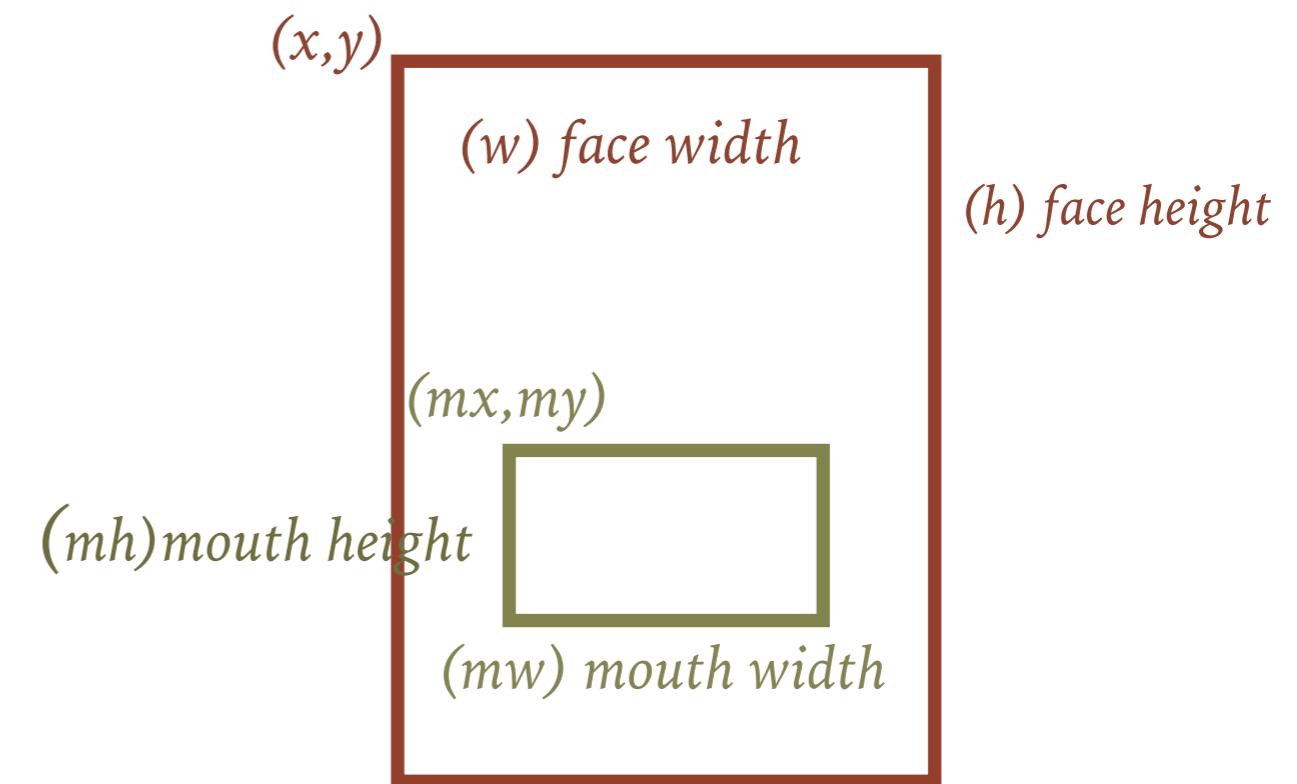
1. Detect facial features using the face cascade classifier (included as a default classifier with OPEN CV)
2. Detect mouth features using the mouth cascade classifier (Not included as a default classifier in OPEN CV)
3. Mouth Classifiers can be found online  
e.g. <http://alereimondo.no-ip.org/OpenCV/34>

# FACE DETECTION METHOD #1

---

*Identify Human face and mouth in each frame*

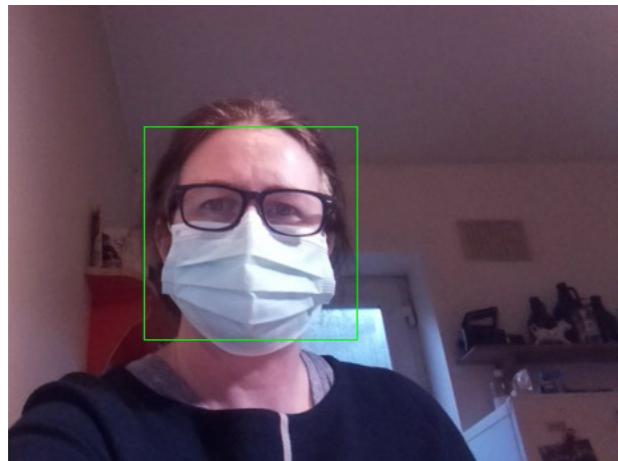
Face Detected	Mouth Detected	Output
If the length of the faces list is > 0 there is a face detected.	If the length of mouth list == 0 that person is wearing a mask.	<b>Mask on</b>
If the length of the faces list is > 0 there is a face detected.	If the length of the mouth list is > 0 and $y < my < y +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



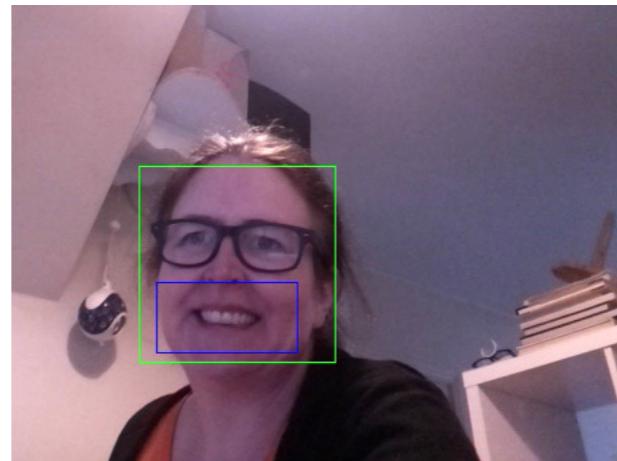
# FACE DETECTION METHOD #1 - TESTING

---

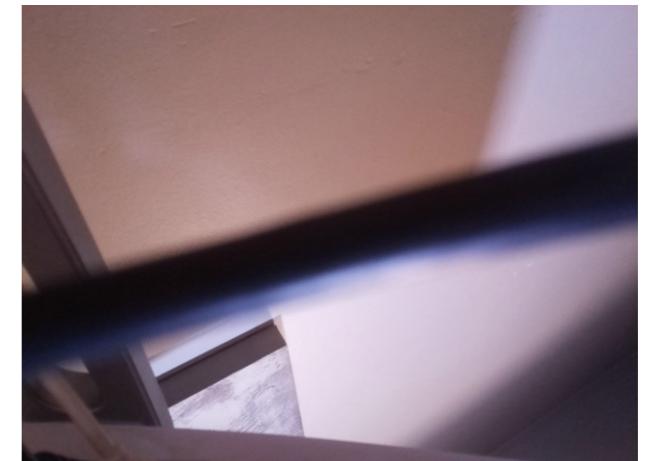
- In opencv there are no haar cascade mouth detection files included and none of the mouth detection xml files I downloaded from the internet worked. There is however a smile detection haar cascade included in opencv. I attempted to implement my project using the smile cascade standing in place of a trained mouth cascade to test the idea.



Face detected  
No Smile detected  
Mask On  
Door access granted



Face detected  
Smile detected  
Mask Off  
Door access refused

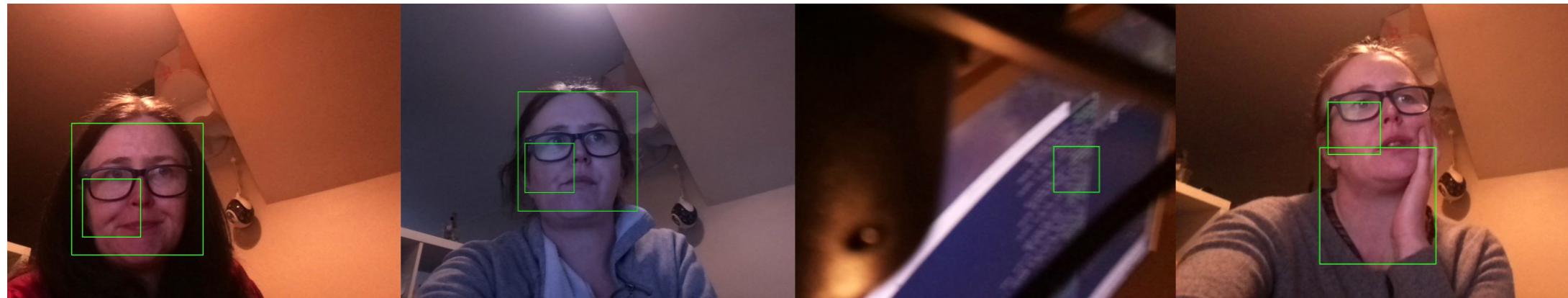


No Face detected  
Door access refused

# FACE DETECTION METHOD #1 – CONCLUSION

---

- CONCLUSION: I have found that using haar cascades isn't a reliable method due to false positive detections that can occur.



*Examples of face positives with frontal face detection*

- The next port of call would be to implement Face Detection method 2 with a trained face mask dataset

# FACE DETECTION METHOD #2 – UNIMPLEMENTED

---

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

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

