

## 0.1 RaspberryPi

### 0.1.1 Objective and Functionality

The RaspberryPi is the backbone of the project, it serves as the MQTT broker and merges the collected data from both the Biometric Sensors and the camera it has attached to provide a prediction of who may the user interacting with system be, sending that final prediction to the Monitor System. Its responsibilities include:

- Subscribing to and processing biometric data from the ESP-01 module attached to biometric sensors via MQTT.
- Processing data from the webcam connected to the RaspberryPi via usb and running it through a neural network to get a prediction of who is the user in the picture.
- Run the biometric data obtained through MQTT and the result of the neural network through a decision tree to get a final prediction of who is the user interacting with the system.
- Sending the final result to the 'rpi/prediction' topic so the System Monitor can show the result.

### 0.1.2 Project Definition and Milestones

The development of the RaspberryPi system involved the following milestones:

1. Setting up an MQTT broker to handle all the IoT devices communications.
2. Establishing MQTT communication to receive biometric data and send predictions.
3. Training a neural network to work with the webcam input and predict which user is in front of it.
4. Training a classification tree to get a final prediction using both biometric data and neural network prediction.

### 0.1.3 Achieved milestones, execution order, priority, and dependencies

1. **Milestone 1: Setting up an MQTT broker** - *Priority*: High. Fundamental for all the data transmission. - *Dependencies*: Basic WiFi setup. - *Execution Order*: First, as it is crucial for data reception.
2. **Milestone 2: Establishing MQTT communication to receive and send data** - *Priority*: Medium. Important for working with real data. - *Dependencies*: Successful MQTT setup. - *Execution Order*: Second, building upon established communication.

3. **Milestone 3: Training neural network** - *Priority:* High. Essential for effective prediction. - *Dependencies:* Functional hardware setup (webcam). - *Execution Order:* Third, getting a first prediction.
4. **Milestone 4: Training a classification tree** - *Priority:* Medium. Important for a more accurate prediction. - *Dependencies:* Functioning MQTT communication and neural network. - *Execution Order:* Fourth, finalizing the machine learning prediction.
5. **Milestone 5: Sending the results** - *Priority:* High. Essential for system effectiveness. - *Dependencies:* Functioning MQTT communication and classification tree. - *Execution Order:* Fourth, last step towards getting a prediction shown to the user.

#### 0.1.4 Hardware setup

The hardware setup for the RaspberryPi comprises:

- A RaspberryPi 4 with WiFi capabilities.
- A logitech webcam with usb connection.

#### 0.1.5 Software Implementation

The software, written in python has the key functions:

- Receiving biometric data from 'sensor3/galvanic' and 'sensor3/heart' topics.
- Triggering a neural network prediction through the webcam obtained footage.
- Run the result from the neural network and the biometric data through a classification tree.
- Publish the final prediction to the 'rpi/prediction' topic.

The operating system (RaspberriPi OS Lite) is responsible of connecting to WiFi and running MQTT broker. So basically it has to:

- Run a service for MQTT broker that launches on startup.
- Establish WiFi connection.
- Run a service that initializes the prediction pipeline when a user is detected.

**NOTE:** The data collection for training the classification tree has also been done on the RaspberryPi as well as the training itself, code for replicating this steps can be found on the project repository.

### 0.1.6 Testing

Testing has been mainly focused on the machine learning aspect, since getting accurate results was non trivial. By generating artificial data (data augmentation) it was possible to finally achieve a more convincing accuracy. Regarding MQTT, setting up the broker was a very straight forward process with no trouble.

### 0.1.7 Dedication Time

Approximately 30 hours were dedicated to developing the RaspberryPi system, using most of the time on the machine learning aspect.

### 0.1.8 Challenges and Solutions

- **Hardware Challenges:** Collecting footage from the usb webcam.
- **Software Challenges:** Getting an accurate prediction from the model. Using a foundational model trained for object detection wasn't working very well for telling faces apart so the solution provided is to use a representative item for each member of the group which is what is placed in front of the camera, this way the neural network can tell us apart easily.

### 0.1.9 Hardware and Software Integration

The only hardware used on this setup was a webcam connected through usb to the RaspberryPi.

### 0.1.10 MQTT Topics

- **sensor3/heart:** This is the topic used to collect heart rate biometric data.
- **sensor3/galvanic:** Provides galvanic resistance data, and together with the first topic, this two provide the full biometric data needed.
- **rpi/prediction:** Topic used to publish the result of the prediction.

### 0.1.11 Conclusion

The development on the RaspberryPi system was a very interesting experience, it was the first time working with machine learning for some of us and it was very challenging to get an accurate prediction, but after some research and testing it was possible to get a good result. The MQTT broker setup was very straight forward and it was easy to get it working.