Project Requirement

1) Identify the Person

2) Get the Prescription/Medicine they needed

3) Giving the medicine

Person Identifying

Facial Recognition

We can use a Raspberry Pi or other single-board computer with a camera module to run a face recognition algorithm. This approach will give you more processing power and flexibility to implement more advanced face recognition algorithms.

For the face recognition algorithm, we can use OpenCV, which is a popular open-source computer vision library that has built-in face recognition capabilities. We can train the algorithm using a dataset of images of known individuals, and then use it to recognize faces in real-time.

- Capture images of people's faces using the camera module.
- Pre-process the images to enhance quality and detect faces.
- Extract facial features from the detected faces.
- Compare the extracted features with the trained dataset to identify the person.
- If the person is identified, trigger the action

To ensure that the system can handle crowded places, we may need to implement additional features, such as:

- Face tracking: to track the movement of individuals and ensure that the system doesn't misidentify people.
- Occlusion handling: to handle situations where people's faces are partially occluded by others.
- Multi-face detection: to detect and recognize multiple faces in a single image.

In terms of the system requirements,

- A camera module with a high-resolution sensor (e.g., 720p or higher).
- A processing unit with sufficient power to run the face recognition algorithm (e.g., Raspberry Pi or other single-board computer).
- A power supply to power the system.
- A speaker or display to provide feedback to the user.

For the specific requirement of identifying a person when they stand in front of the device for at least 2 seconds, we can implement a timer that triggers the face recognition algorithm only when the person has been detected for a minimum of 2 seconds. This will help reduce false positives and ensure that the system is more accurate.

Raspberry Pi facial recognition system can identify a person from a given picture using a camera. We would need to train the system with images of the person we want to recognize, and then it can identify that person when they appear in front of the camera. The system can be set up to continuously analyze the video feed from the camera and compare the faces it detects to the trained dataset. When it finds a match, it can trigger an action, such as displaying the person's name or performing a specific task.

To accomplish this, we can use a Raspberry Pi along with a camera module and a face recognition library like OpenCV. We would first need to detect faces in the images and then extract facial features, which can be represented as a set of numbers, often called encodings. Once you have the encodings for the images in your dataset, you can compare the encodings of the detected faces in the video feed to the dataset to identify the person.

To detect a specific person using facial recognition on a Raspberry Pi, we need to train the system using images of that person. This process is called "enrollment" or "training".

- Collect images: Gather a set of images of the person you want to detect. These images should show the person's face from different angles, with varying lighting conditions, and with different facial expressions.
- Pre-process images: Resize the images to a consistent size, convert them to grayscale, and apply any necessary normalization or filtering to enhance the quality of the images.
- Extract facial features: Use a facial recognition library like OpenCV or face recognition to extract facial features from each image. These features are typically represented as a set of numbers, called an "encoding" or "face embedding".
- Create a dataset: Store the extracted facial features and corresponding labels (e.g., the person's name) in a dataset.
- Train the model: Use the dataset to train a facial recognition model, which learns to recognize patterns in the facial features and associate them with the corresponding labels.

Once we trained the model, we can use it to detect the person in new, unseen images or video streams. The system will compare the facial features extracted from the new images to the trained dataset and return a match if the similarity is above a certain threshold.

Some important considerations when training a facial recognition system:

- Quality of images: The quality of the training images has a significant impact on the accuracy of the system. Use high-quality images with good lighting, resolution, and focus.
- Variety of images: Use a diverse set of images to train the system, including different angles, lighting conditions, and facial expressions.
- Number of images: The more images you use for training, the better the system will be at recognizing the person. However, using too many images can lead to over fitting, where the system becomes too specialized to the training data and fails to generalize well to new images.
- Labeling: Ensure that the labels associated with each image are accurate and consistent.

Alternative way

- We can use IP camera/smart camera and send the data to a system for identifying the person and sending command to dispense medicine to device. Like DDS system, Computing will take place in external system (PC) .We can set communication to the device using cables (rj45/DB9) or any other. The system will identify the person using the image/video shared in the camera and can be used to send data to the device for medicine dispensing .The system can also use to find any faults and other misbehavior, And also be used to display status of the available medicine
- We can also add RFID tags to get patient details 13.65mhz /125khz RFID cards
- Display entry for new person
- Computer Vision with Object Detection: Use computer vision with object detection algorithms to detect the patient's presence. The algorithm can detect the patient's body or face and trigger an output when they are within a certain range