University of Colorado
Boulder

Embedded Interface Design
[ECEN 5783]

Superproject Final Report

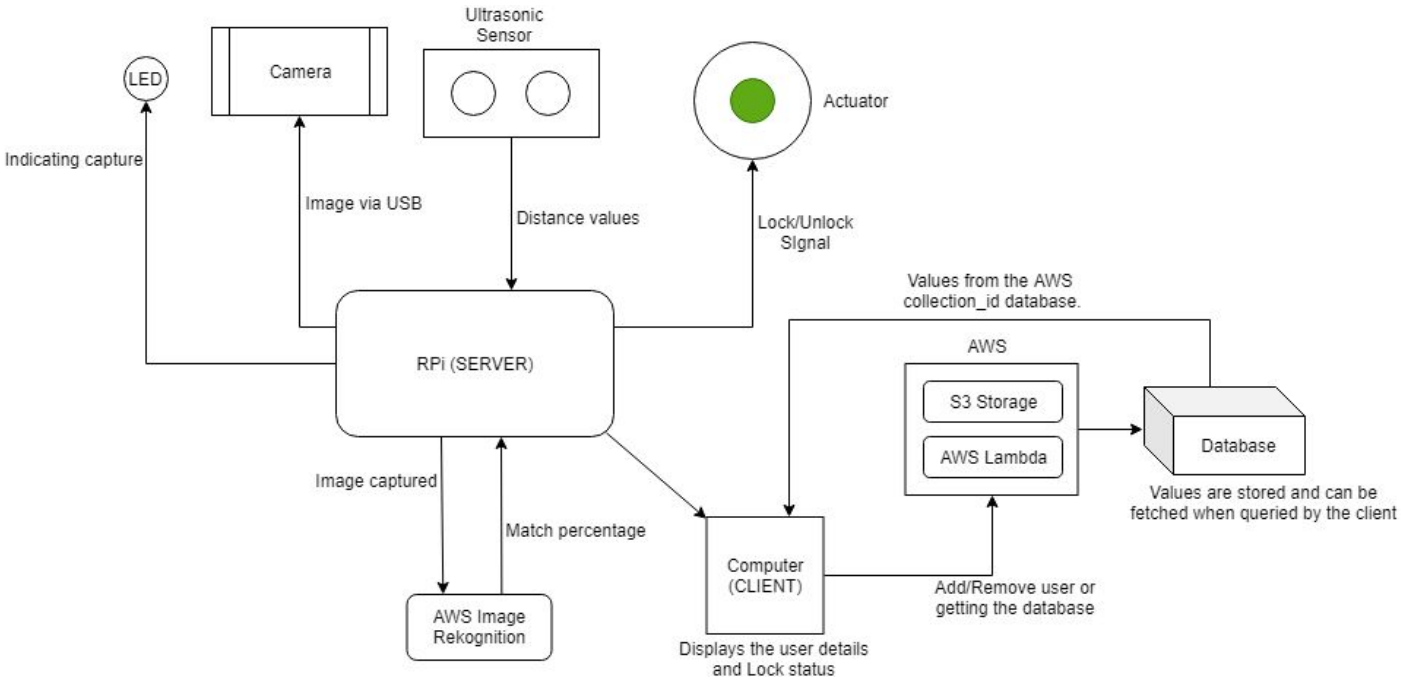
FACE RECOGNITION BASED DOOR LOCKING SYSTEM

Team:
Hardik Senjaliya
Isha Sawant

Table of Contents

Final System Architecture Diagram

Architecture Diagram:



Statement:

Link to Git repository.

- The functionality of the Server:
    - Our server is the Raspberry Pi module, interfaced with a camera, an ultrasonic sensor, a servo motor, a LED and connected to the internet.
    - The server runs a Tornado Web Server on it to send status to the web client and also receives messages from the web client running remotely on a personal computer.
    - The ultrasonic sensor keeps tracking the distance of the user from the camera and once the user enters the minimum specified range (60 cm), it captures the image, which will be indicated by the toggling of the LED.

- The captured image of the user will be used by Amazon Image Rekognition service to identify the user.
- If the user is present in our system database, then the user will be granted access to the door. The servo motor interfaced with the server fulfills the role of the door in our prototype.
- Also whether the user has been given access or not, the appropriate message will be sent to the web client via WebSocket running with the Tornado Web Server, locking the door automatically after a few significant moments to block unauthorized people from entering.
- In the event of an emergency, the server receives messages to lock or unlock the door as well.

- The functionality of the Client:

  - The WebClient runs on the personal computer with WebSockets.
  - The purpose of the web client is to monitor the status of the door.
  - After successfully logging into the systems, the user can monitor the status of the door.
  - The client receives the message from the server to update the status of the door and by whom the door was accessed.
  - The most important features of the web client is adding a new user and removing an existing user from the system.
  - The client UI also allows the admin to view the current database entries authorized to unlock the door.

Project Deviation Statement

- We added a few features, hence broadening the scope of this prototype:

  - To automate the capturing of an image using the camera, we added one more sensor. The added Ultrasonic sensor detects the distance of the user from the camera and once the user comes into the specified range, the camera clicks the picture and processes it to give or deny the door access.
  - Admin rights with the help of login username and password makes the client end more secure and does not allow everyone to access the database.

    - Add User: this feature adds a new user to our existing system. From the HTML client, we can select a photo of the new user to be added and it will be added to the Amazon S3 storage and then to the collection of all the Face IDs of current users.
    - Remove User: this feature removes an existing user from our system. From the web-based client, we can take the name of the user to be removed and as an input and the photo of the user will be removed from the Amazon S3 storage as well as from the collection of all the Face IDs of the users.

- We proposed to use the Rotating Door Lock motor to demo the door locking and unlocking feature. However, the specific motor we ordered requires positive and negative voltage to work properly in the clockwise and anticlockwise direction. Thus we ended up using a servo motor.

Third-Party Code usage Statement

- We have used the AWS Image Rekognition service for our project. We referred the user guide for AWS Image Rekognition which has many code snippets available for learning the concepts and usage. We have used those snippets and modifies to meet the requirements of our project.
- Here's the link to those snippets: AWS adding a face, deleting a face, finding a face, creating a collection and listing the collection.

Project Observation Statement

- Initially we thought that, adding a new user or removing an existing user from the system would be difficult and would take time to complete in the given time frame of the project. Just as we thought, though adding these features required lots of research and time, we were able to implement it fully.
- Working on this system definitely went better than expected as we had AWS to our rescue. We did not have to get deep into the concepts of machine learning, OpenCV, etc, hence helping a lot to get done with major functionalities.
- We gained a perspective to prototyping and only adding necessary features to the system, hence making it better and more user-market oriented.