Team E7: IntelliRack

Doreen Valmyr, Ryan Lin, Surafel Tsadik

Use Case

Attendees check in personal items such as backpacks and coats at events

Some Problems:

- Attendants needed to man station
- Time-consuming
- Disorganization

 Image from https://cptevntsvcs.com/coat-check/

Design Requirements

Facial Detection and Recognition:

- Detect faces within 0.5 meters, within 5 seconds
- 95% accuracy for recognizing faces

Item deposit/retrieval:

- Detect an item has been added or removed within 1 second
- Once user face has been matched, display user's item position with LED lights within 1 second

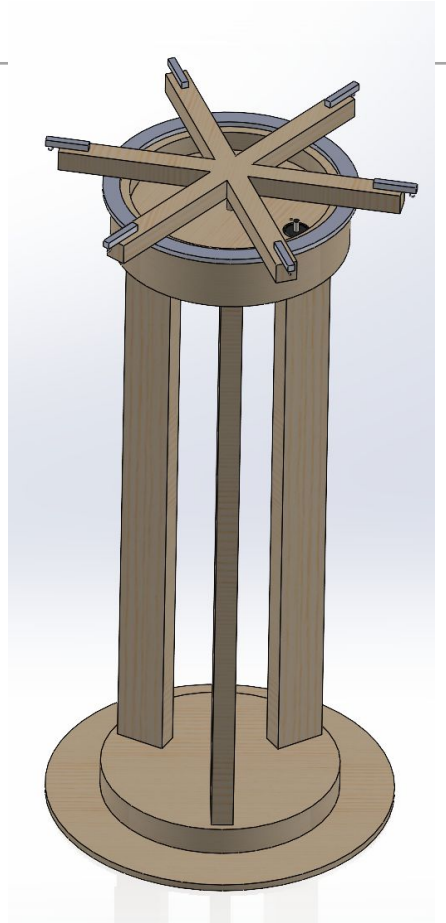
Item/Coat Stand Integrity:

- Be able to handle weights of 25 pounds on each hook and worse case 150 pounds on the entire stand

Solution Approach

Hardware item stand built with:

- **Plywood** - lightweight and strong
- **18” Ring Bearing** - smooth rotation and support
- **Load cells** with **hooks** - weight detection
- **LEDs** - Notify users of open hook
- **Arduino** - Ease of use, interfacing with components
- **RF transceiver module** - Communication between components
- **Stepper Motor** - Strong enough to rotate rack



Solution Approach

Software Program/Application:

- Built using Django
- Logs and displays which users have an item checked in
- Processes user inputs and controls hardware stand
- Balanced-rack algorithm - ensures placement of items on rack for best weight distribution
- Joins other components of system together: Communicates with small arduino through serial port and calls openCV methods

Solution Approach

Facial Recognition Algorithm

- OpenCV
- Use face bounding box to detect user distance from hardware stand
- Scan and store face in database for recognition later
  - MySQL

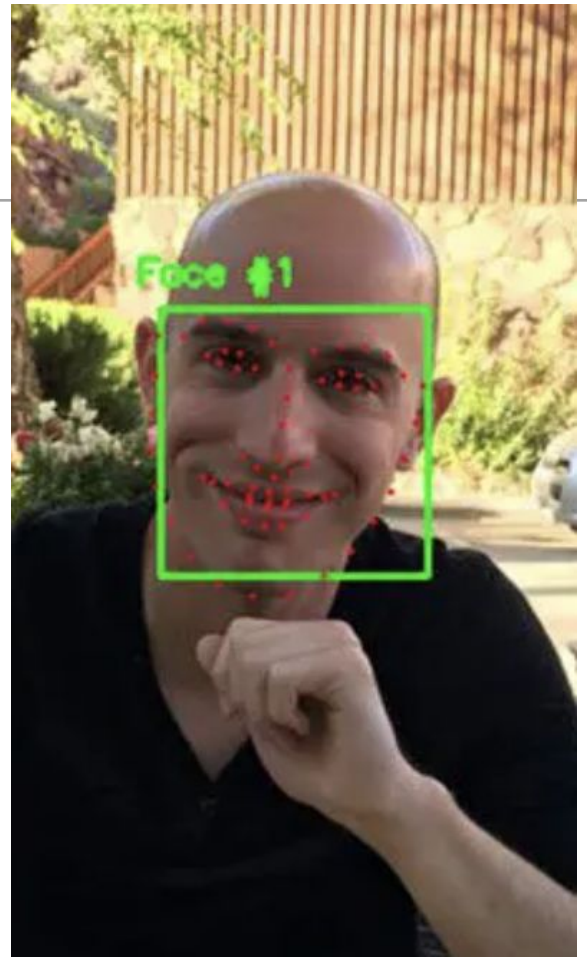
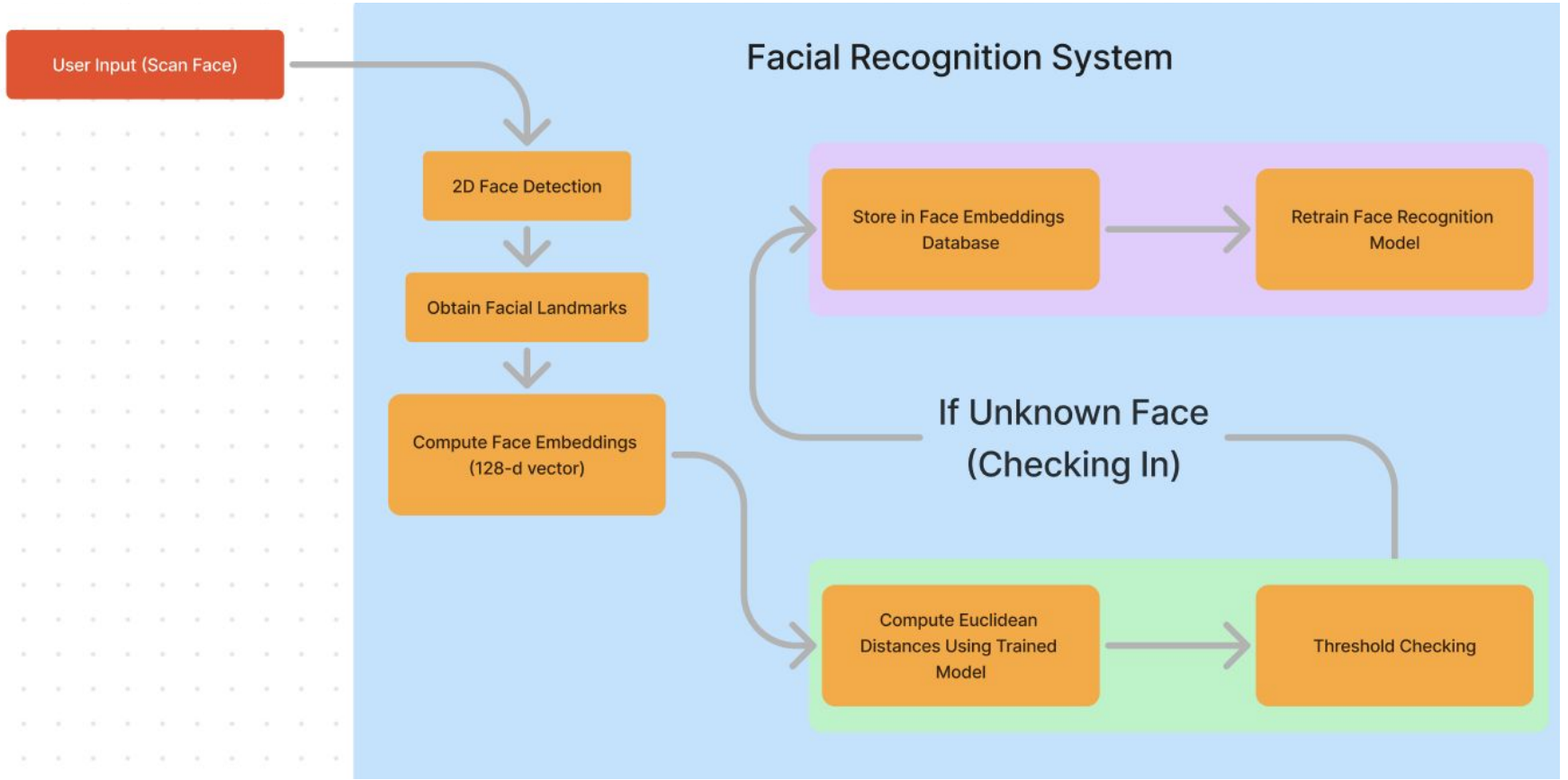
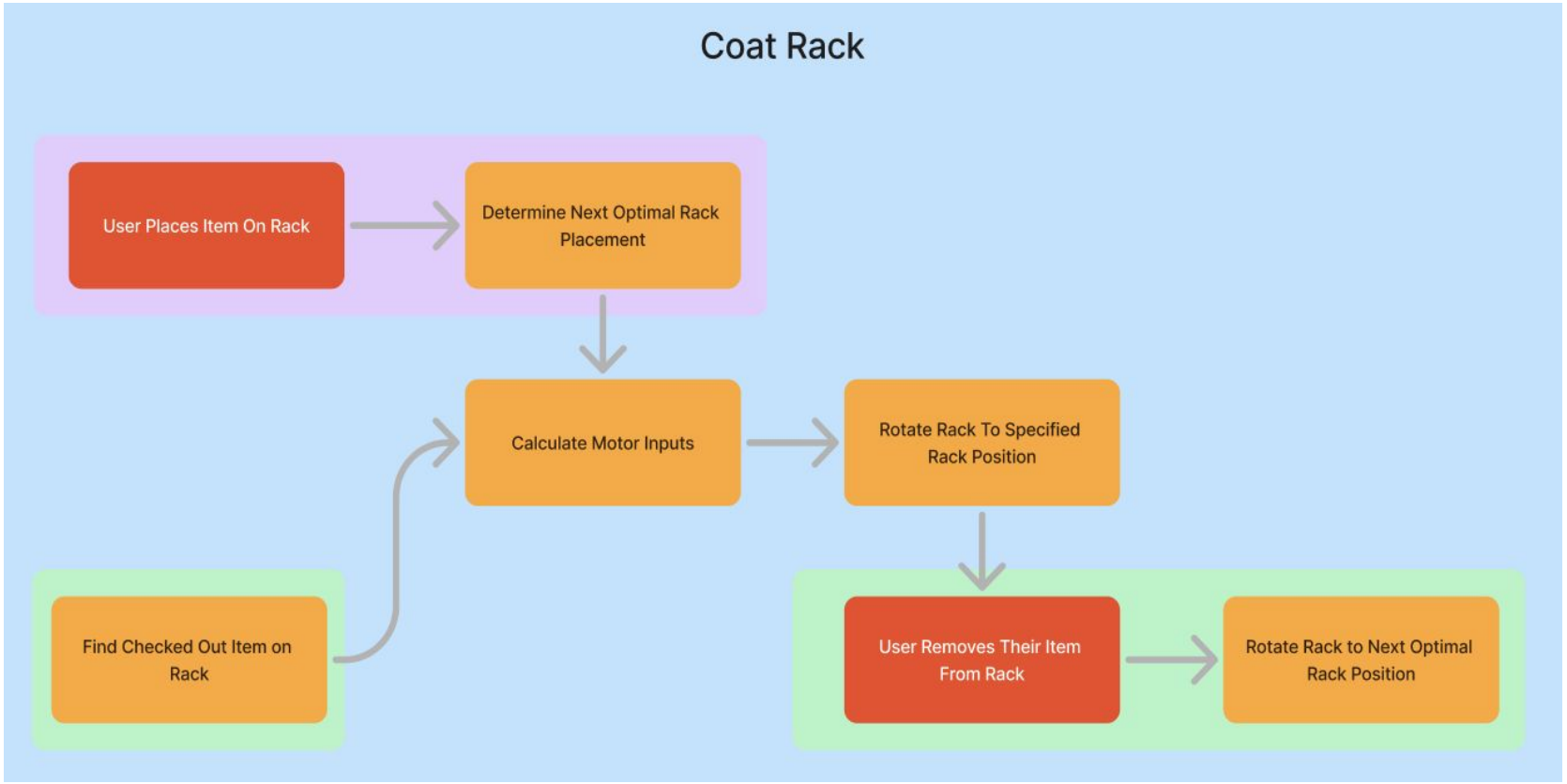


 Image from https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

System Diagram



System Design



Implementation Plan

- Facial recognition tools from OpenCV
- Team is designing and assembling rack on our own
  - Using PID control for precise control of speed and position of motor
  - Some pre-assembled components:
    - Ring Bearing
    - Stepper Motor
    - Load Cells
- Collect sensor data with Arduino Mega
- Communicate all data wirelessly (with NRF24L01 transceiver) to software application
- Communicate actuation settings from application to hardware

Testing and Verification

Detect faces within 0.5 meters, within 5 seconds, with 95% accuracy:

- Detecting Faces: stand at 2m, 1m, and 0.5m away from camera (should only work at 0.5m)
- Accuracy Test: Recognize Different/Similar faces, with 20+ volunteers

Detect item placement/removal and display item position within 1 second:

- Remove and place items on hooks at least 20+ times, should quickly display item or open slot positions within 1 second 90% of the time

Handle weights up to 25 lbs/hook, 150 lbs total on rack:

- Weight Test: Varying weights, weight imbalances
- Max weight (150lb) and rotation test, one-sided imbalance tests with 75lb

Risk Factors and Mitigation

Risk Factors:

- Accurate facial recognition
- Robust rotating rack system

Mitigation Plans:

- Limit the types of items users can place on the rack to solely lightweight items likes coats rather than backpacks
- If weight detected is over threshold pounds, flash LED red to notify the user
- Use facial recognition libraries other than OpenCV

Schedule

Construction and electronics of item stand - Ryan and Doreen
Lead on software - Surafel (support from Ryan and Doreen)
Research and deployment of facial recognition - Team Effort