Weekly Status Report 3
Week 3
Name: Joe Lanzi

Activities and Accomplishments

- Finished the facial recognition capable of identification of persons with facial coverings with over 99% accuracy
- Finished a local Website in Html5, CSS3, and JavaScript
- Created a local Webserver in C capable of running html, css, and javascript which fully integrated all the software together in a local storage device
- Created a full software handler capable of automating all task to completion
- Created a network capable of wide world access

Problems

- Facial Mask eliminates simple methods and causes expensive computations that must be solved with dedicated GPU's locally or cloud-based computing

Plans

- Improve software and wait for the hardware to be done

Individual Hours Worked This Week on the Project

(Note: This is the total number of hours you actually worked on Capstone. Eg, if you have a team of 4 people, you worked together for 2 hours this week. **Individually, you worked 2 hours/4 =0.5 hour this week. You put 0.5 here.)**

I worked on the project for **25 hours**

The group met for a total of **3 hours/5 people = 0.6**

$\boxed{25.6}$

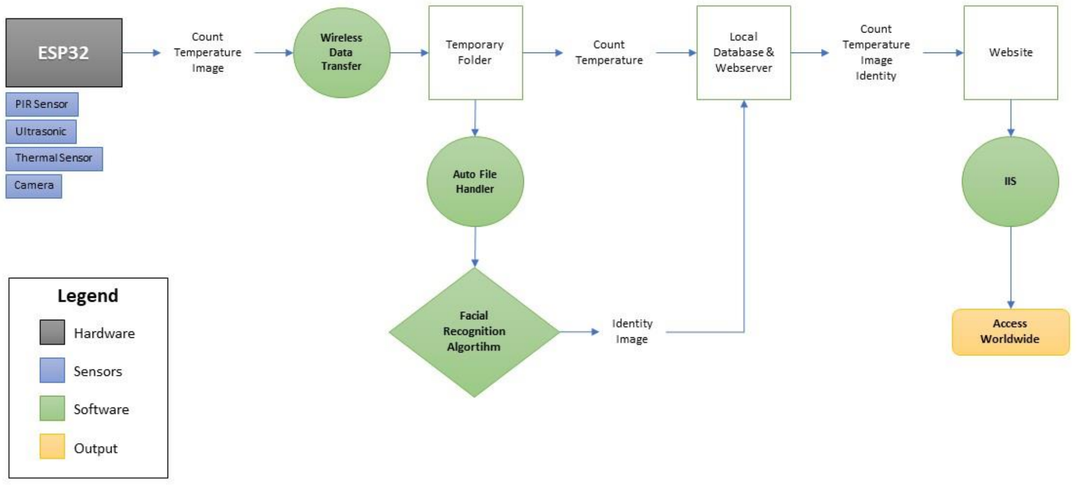
Discussion

The report below includes <u>only my own</u> work for the third week of Spring 2021.

Two weeks ago - I've informed the group to start working on the power of the whole system including calculations, integrate all hardware systems together, and start collecting testing data.

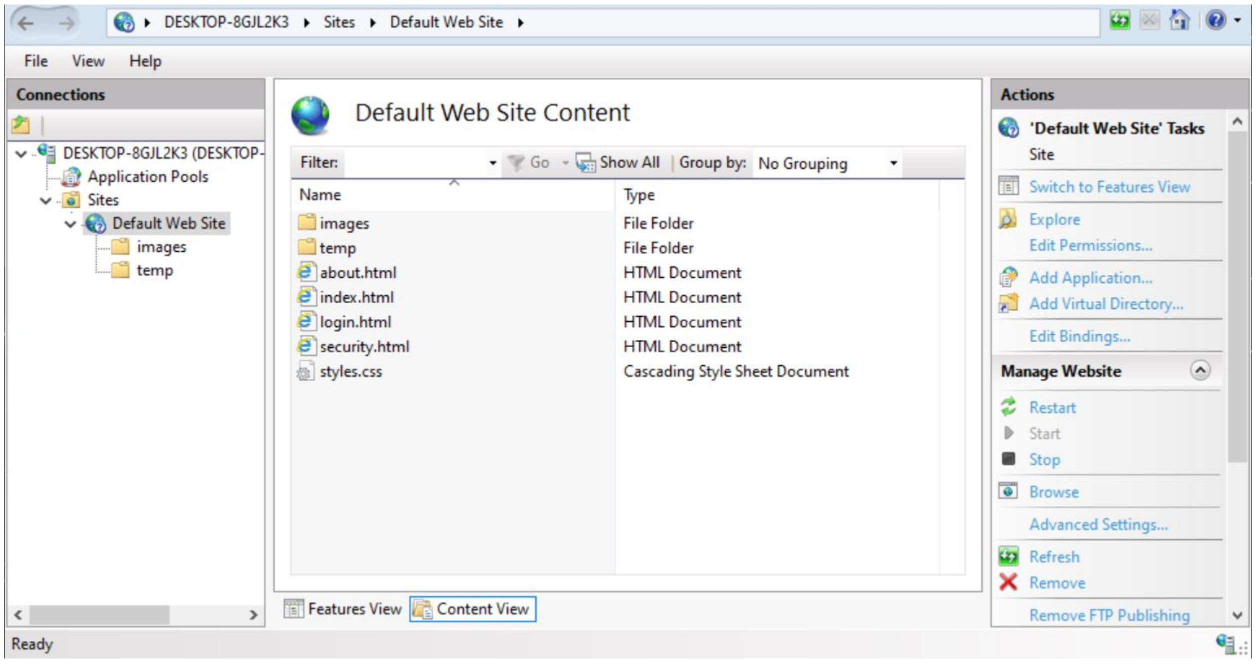
The power calculation for the whole system is incomplete. Hardware is divided to PIR + Ultrasonic & Thermal Sensor + Camera, not fully integrated yet

System



The image above is a brief summary of the entire project.

The software is comprised of the green nodes. Since the Facial Recognition, Webserver, and the Websites were completed, the next step was to create a Database that is capable of connecting all the software together.

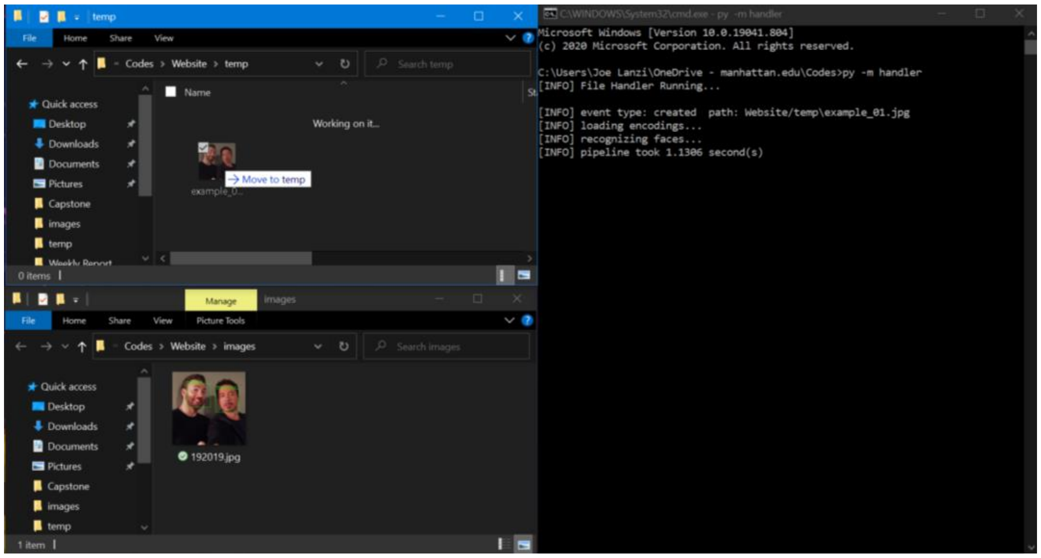


Name	Type
images	File Folder
temp	File Folder
about.html	HTML Document
index.html	HTML Document
login.html	HTML Document
security.html	HTML Document
styles.css	Cascading Style Sheet Document

For the sake of time and simple integration, a local database was configured using PHP for SQL server. This way, we can easily configure the webserver locally and host the website in a local computer. With this database, we can input all the data from the ESP32 here and output all results with HTML scripts into the website.

Once a database was completed, we now need a program that can automatically handle files. This handler or “brain” should be capable of taking the input files from the ESP32, run the facial recognition algorithm on the images, and update the rest of the information together to be displayed to the website.

For this handler, I used a Python API to monitor file system events for the database. This was applied by using a simple modified FIFO queue technique. Which means, that if the files being monitored have new items created, the observer can enqueue the file name, location, and type being inputted into the database’s monitored folders and do what the users specify instructions based on file types. For this case, the system handler monitors a folder where the data is being inputted into, then if the file type is that of an image (.jpg), it will run the facial recognition software automatically and outputs the identity of a person in the image into the database. Whereas if the file type is that of an excel documents (.cv2 or .xlsx), it will then update the database info with the new inputs from the ESP32.



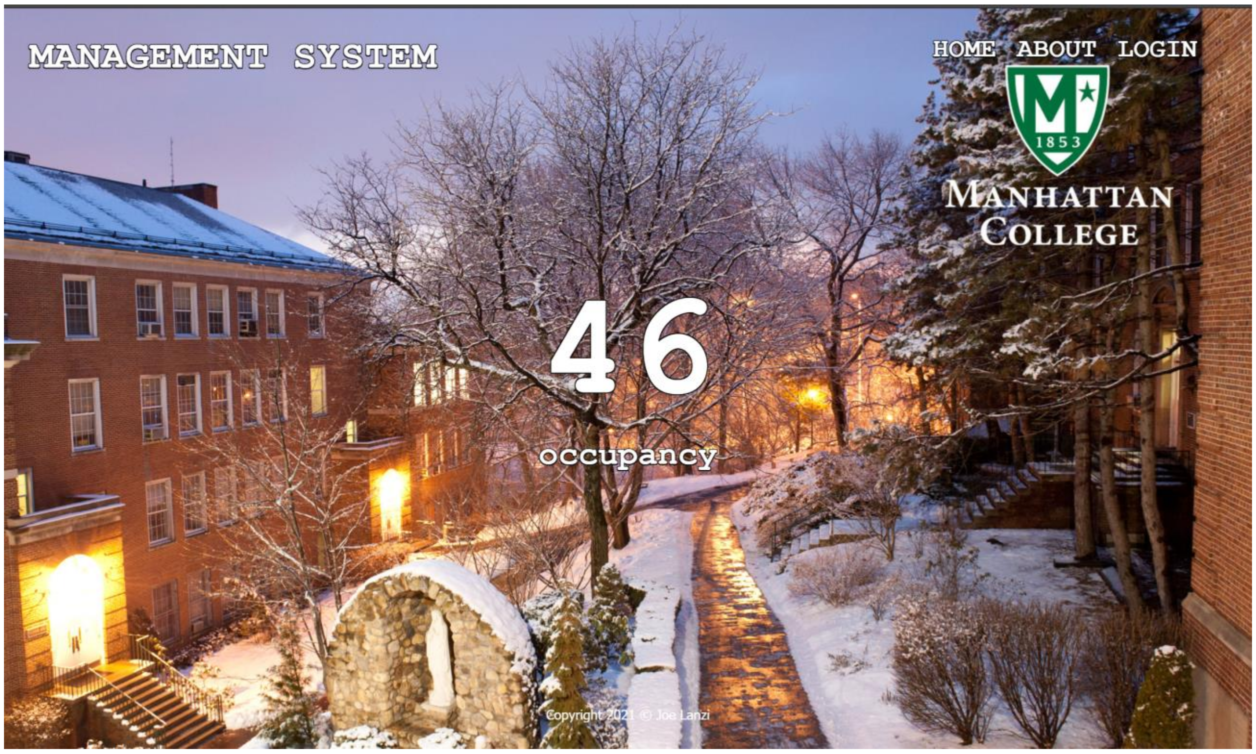
```
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Joe Lanzi\OneDrive - manhattan.edu\Codes>py -m handler
[INFO] File Handler Running...

[INFO] event type: created  path: Website/temp\example_01.jpg
[INFO] loading encodings...
[INFO] recognizing faces...
[INFO] pipeline took 1.1306 second(s)
```

As a simple demonstration, the top left folder called “temp” is a temporary folder where new data inputted in. When the system handler, on the right, detects a new item in the “temp” folder, it will then automatically run the facial detection algorithm and outputs the results in the database on the bottom left side.

To conclude, the system handler acts as the “brain” of the entire software to integrate all parts together taking inputs and getting necessary outputs. The system handler only has an execution time of just over 1 second to classify identities and outputs into the website.

Finally, now that all the software nodes are integrated together, the website should then be accessible worldwide in any devices.

For this, I use an IIS (Internet Information Services). IIS runs in the Microsoft.NET platform on the Windows OS. The IIS acts as the webserver for hosting out HTML web application. The IIS handles all the request using a request-response pattern, where a Client sends a request to the webserver and receive back with the HTTP. With this, we can easily host a website that can be accessed anywhere in the world with a simple IP address.


MANAGEMENT SYSTEM
HOME
ABOUT
LOGIN
1853
MANHATTAN COLLEGE
46
occupancy
Copyright 2021 © Joe Lanzi