Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

ECEN 404 Final Presentation
Team 57: Deep Learning for Hydroponic Soybean Growth

Team members: Samuel He, Mary Hughes

Sponsor: Sambandh Dahl, Krishna Gadepally

Project Summary

- **Problem Statement:**

  Researchers take time to track the solution and day of growth of a hydroponically grown plant

- **Solution**

  Deep learning model and user interface that tracks Day of Growth and outputs other growth data that may be useful to the user

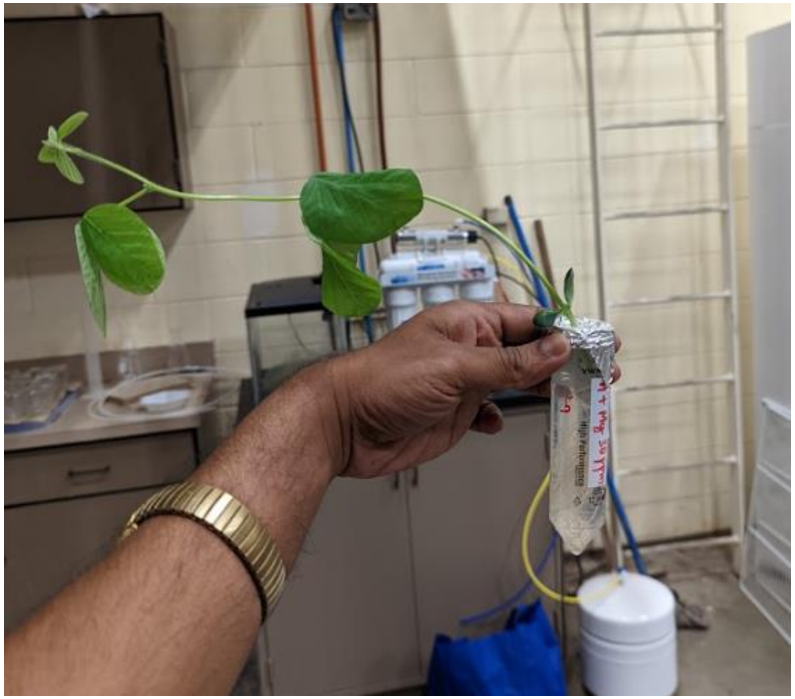
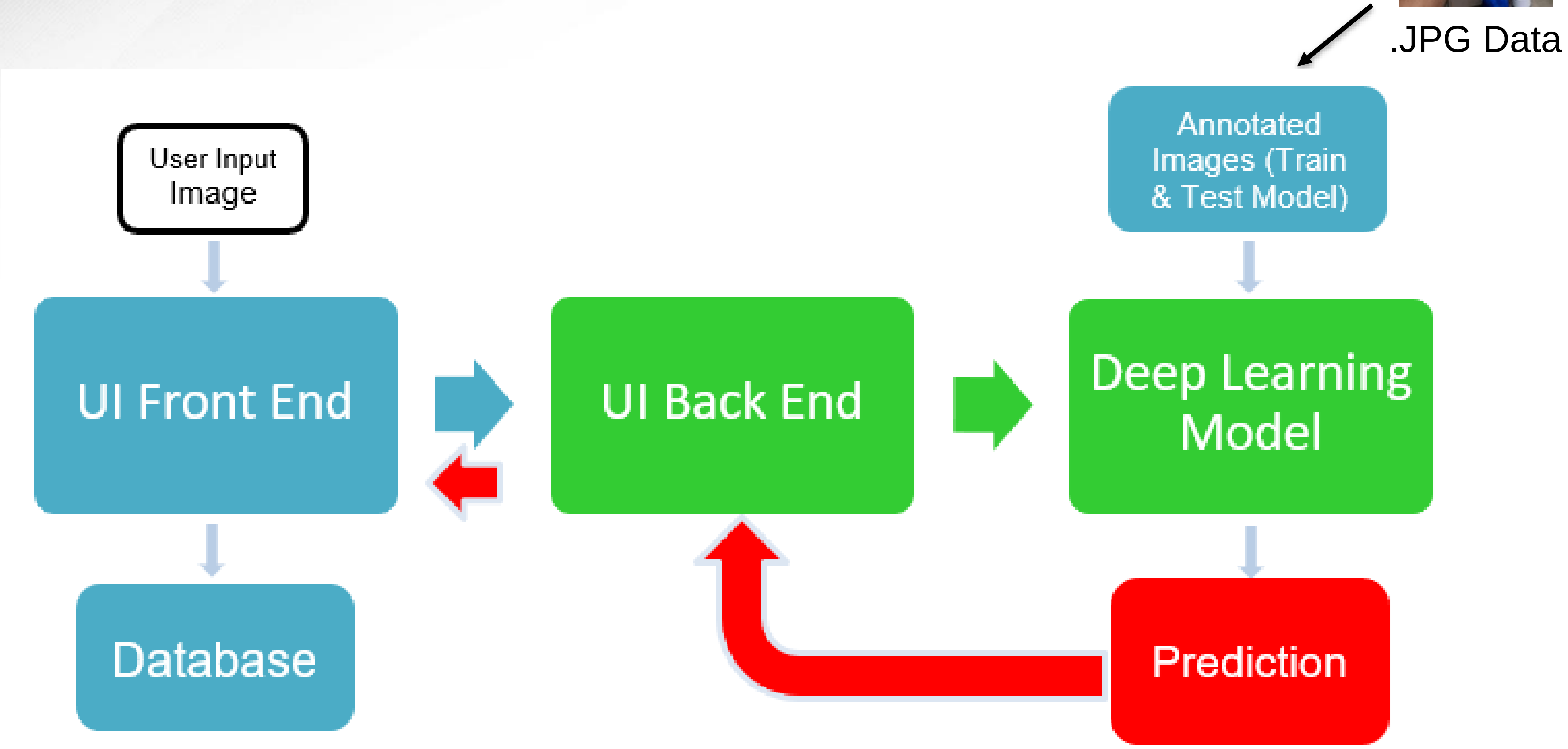


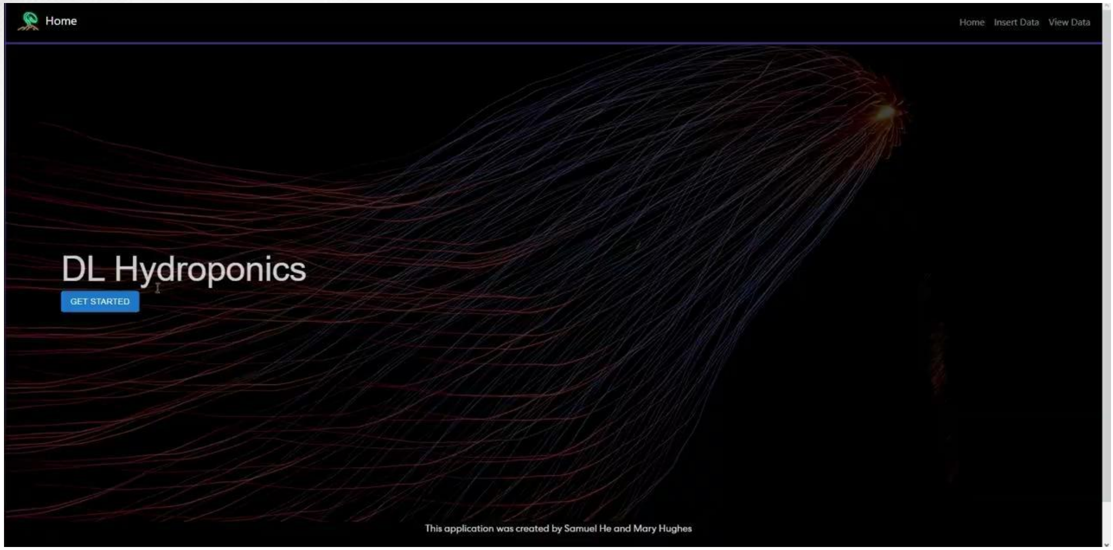
Image 1. Sample Data Image

System Diagram



Samuel He
Mary Hughes

Integrated Project Video



This application was created by Samuel He and Mary Hughes

Deep Learning Model & Data Analysis

Mary Hughes

Challenges:

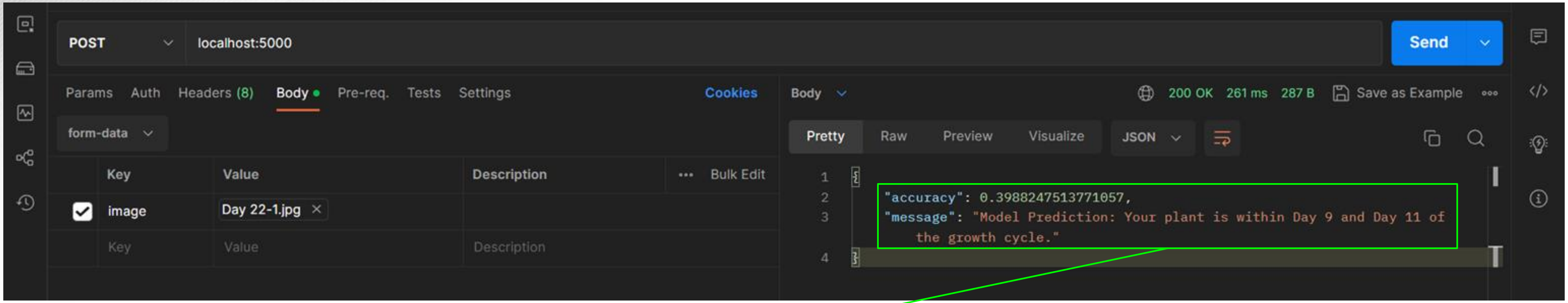
- Dataset is small and biased
- Cloud deployment with AWS has a lot of intricacies
- Defining custom metrics caused issues with deployment because of version control issues

Solutions:

- Implemented data augmentation techniques to expand the dataset without acquiring more data
- Deploying on (other platform) instead
- Eliminated custom metrics from final model file

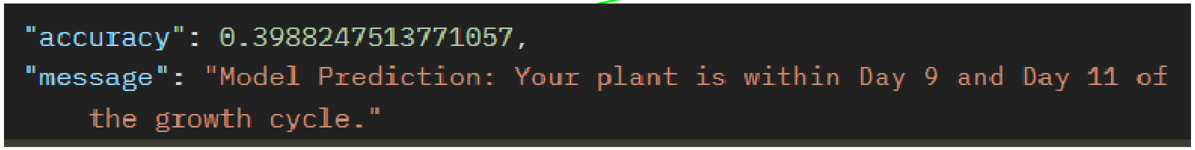
Mary Hughes

Subsystem: DL Model & Data Analysis

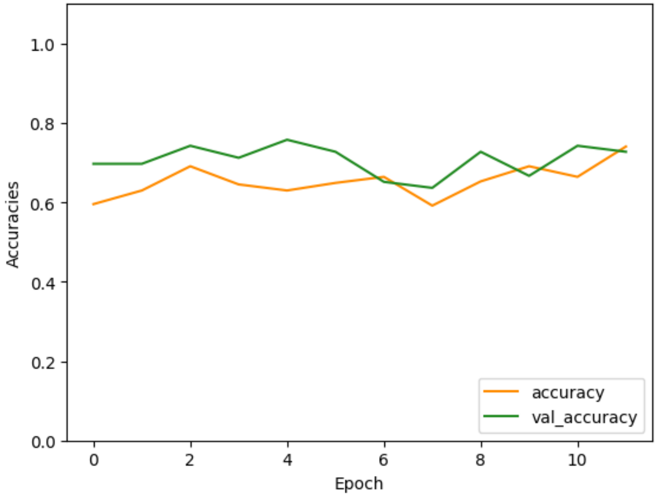


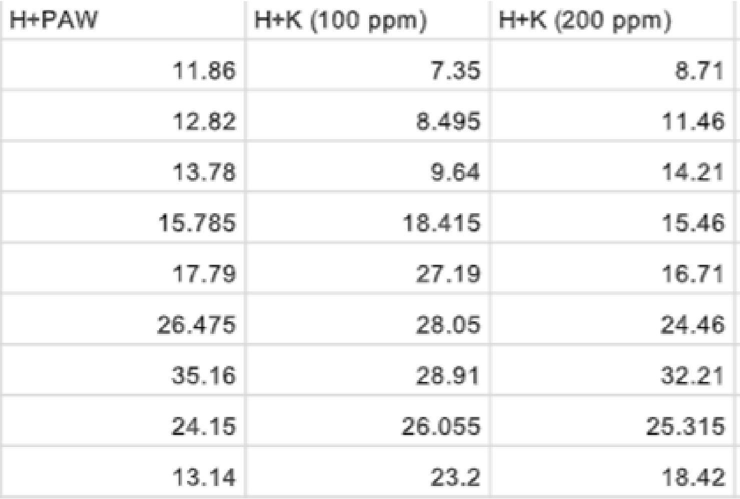
	Key	Value	Description
☑	image	Day 22-1.jpg	
	Key	Value	Description

```
{
"accuracy": 0.3988247513771057,
"message": "Model Prediction: Your plant is within Day 9 and Day 11 of
    the growth cycle."
}
```



```
"accuracy": 0.3988247513771057,
"message": "Model Prediction: Your plant is within Day 9 and Day 11 of
    the growth cycle."
```





H+PAW	H+K (100 ppm)	H+K (200 ppm)
11.86	7.35	8.71
12.82	8.495	11.46
13.78	9.64	14.21
15.785	18.415	15.46
17.79	27.19	16.71
26.475	28.05	24.46
35.16	28.91	32.21
24.15	26.055	25.315
13.14	23.2	18.42

User Interface & Backend API

Samuel He

Challenges:

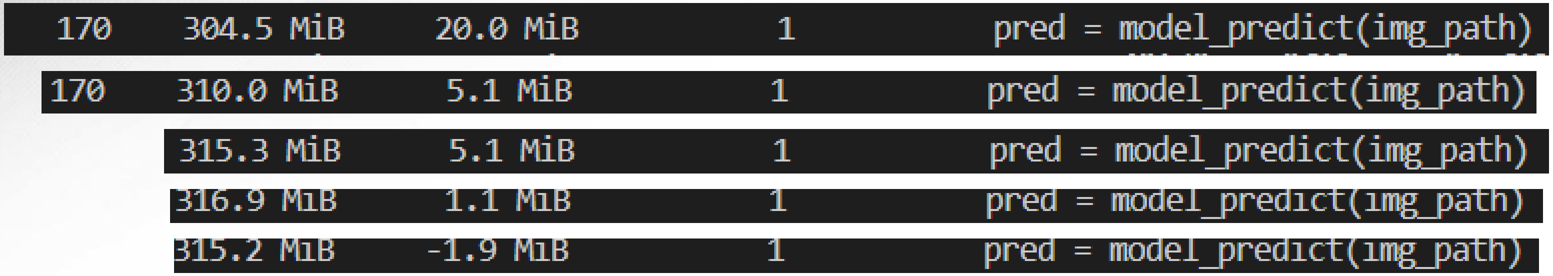
- Cloud deployment has limited resources (512MB RAM)
- User Interface had to adapt to the amount of data the user wanted to enter.
- Lag time when initially accessing the UI, or when initially querying the APIs

Solutions:

- Divided tasks into the Database and Prediction API
- Used React’s dynamically generated form structures instead of static ones.
- Added a service to the API and UI to constantly ping them to prevent idling.

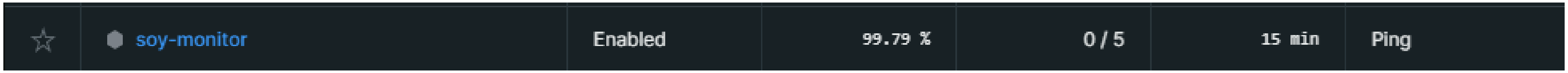
User Interface & Backend API Validation:

API RAM Usage Monitoring (needed to be under 512MB):

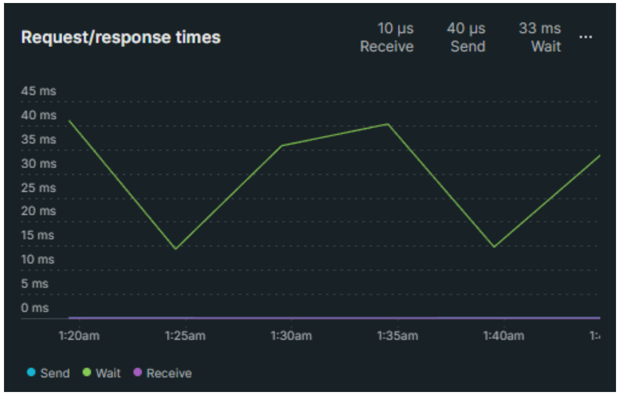


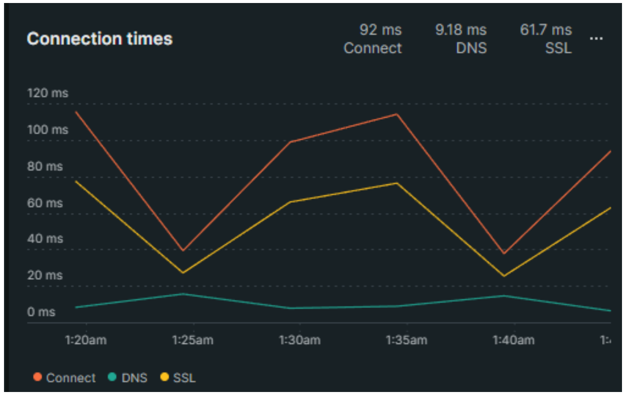
```
   170    304.5 MiB    20.0 MiB         1         pred = model_predict(img_path)
 170    310.0 MiB     5.1 MiB         1         pred = model_predict(img_path)
        315.3 MiB     5.1 MiB         1         pred = model_predict(img_path)
        316.9 MiB     1.1 MiB         1         pred = model_predict(img_path)
        315.2 MiB    -1.9 MiB         1         pred = model_predict(img_path)
```

App Response Time monitoring for front-end



☆ ● soy-monitor | Enabled | 99.79 % | 0 / 5 | 15 min | Ping





Integrated System Results

- The UI is able to take in multiple images as input.
- The UI successfully passes images to the model, and intermediate apps request a prediction for each input.
- The deep learning model is able to make a prediction about the day of growth.
- The UI outputs the prediction, and the accuracy of the prediction, in an easy-to-read format for the user.
- This information is stored in the database, which is displayed in a separate tab, so the user can go back to previously processed images.

Conclusions

- Major changes: the deep learning model does not correctly predict the day of all inputs. A database was added to the system, along with the additional data analysis. The system documents have been changed accordingly.
- Almost fully integrated: data analysis data needs to be added to the database. Validation to be completed for full system, along with input delivery time, failure response, and readability on different devices. All of this will be wrapped up in lab on Wednesday.

Thanks & Gig ‘Em!