Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

ECEN 404 Bi-Weekly 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
  - Nutrient Solution
  - Day of Growth

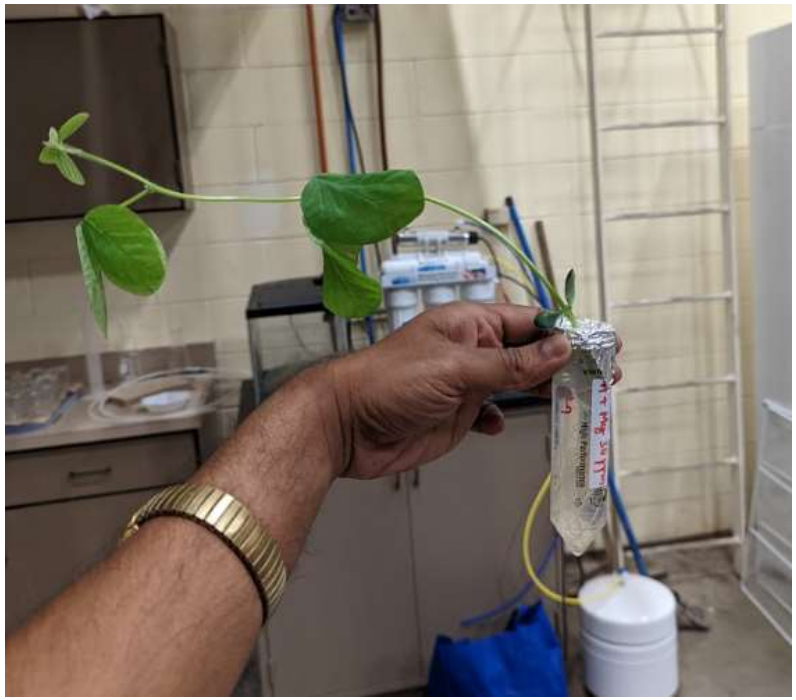
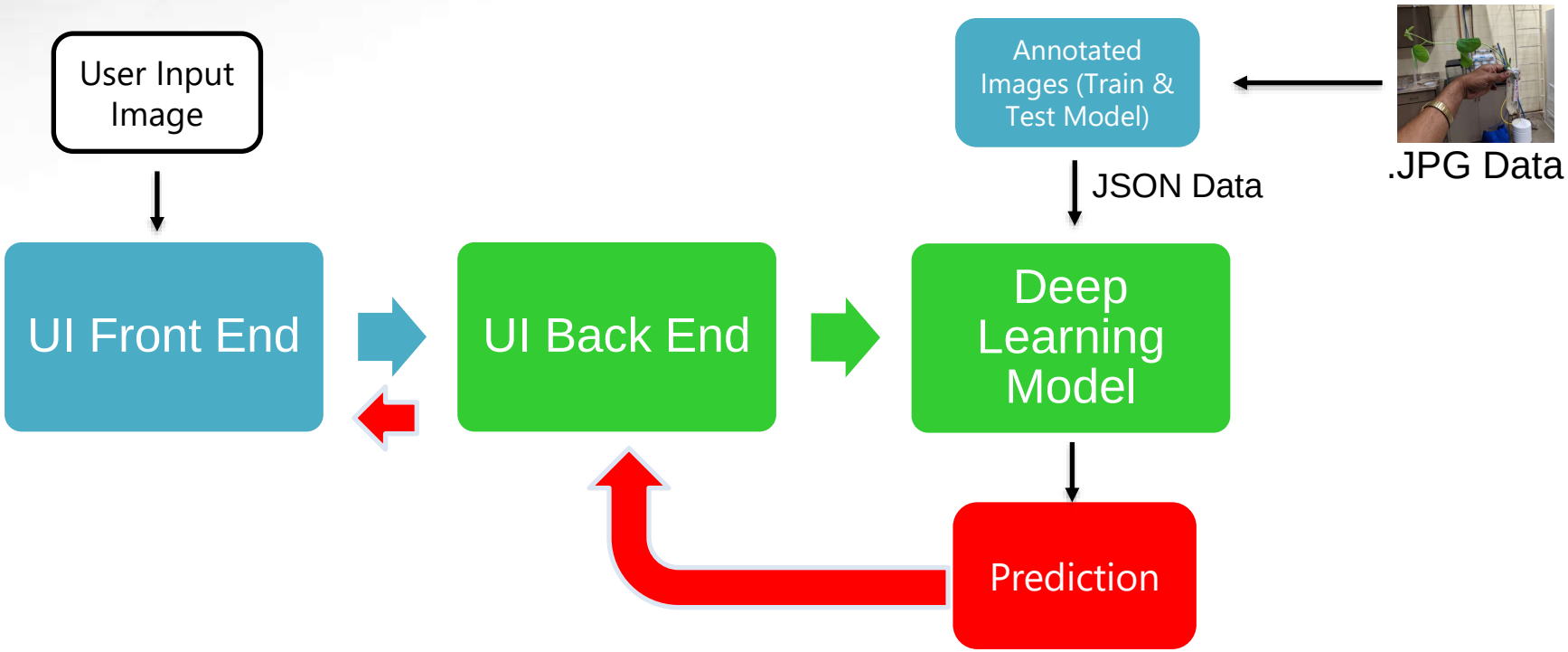


Image 1. Sample Data Image

System Diagram

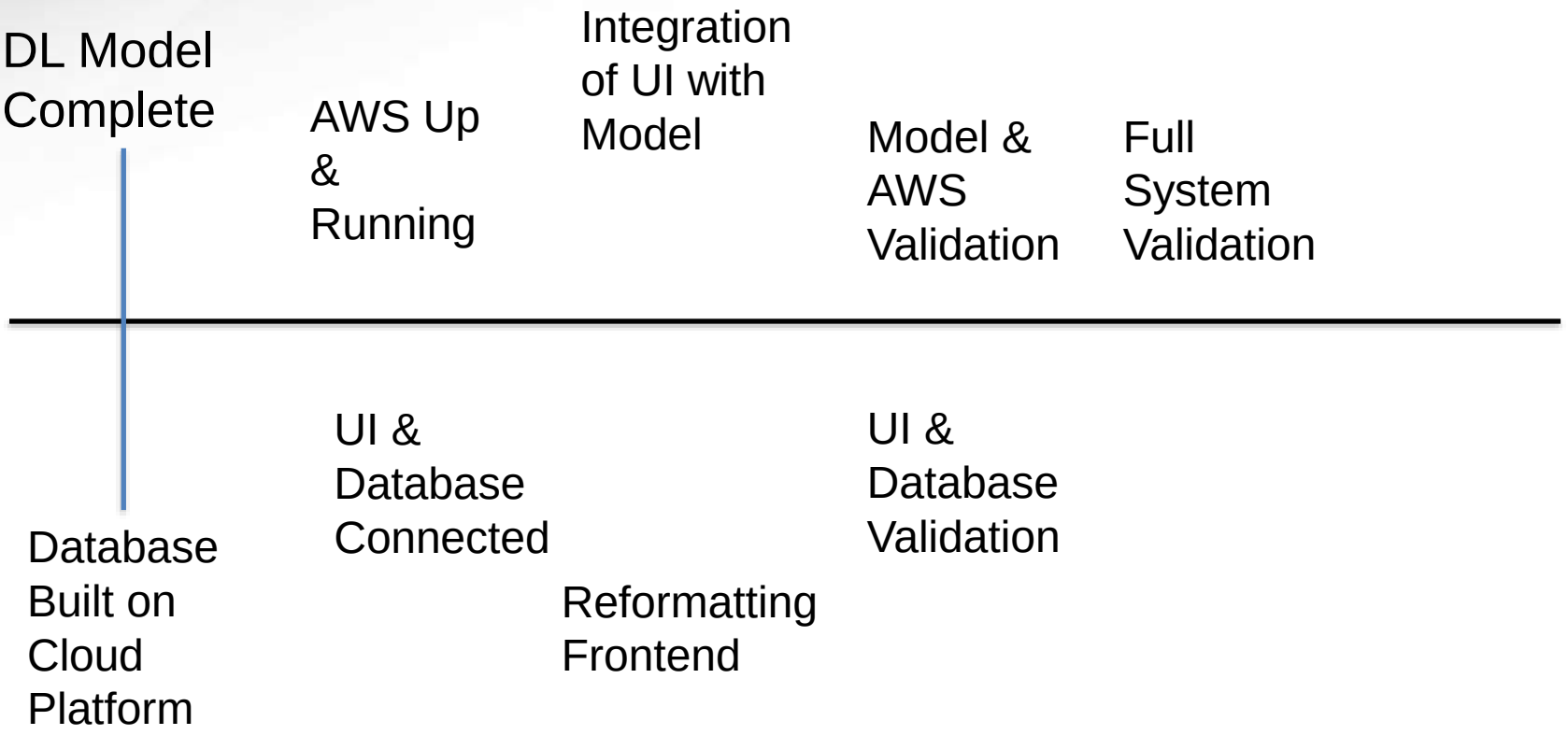


Samuel He
Mary Hughes

Major Project Changes for 404

- Adding a database subsystem (Samuel)

Project Timeline



Mary Hughes

Deep Learning Model

- For 403, we used a deep learning model trained by our sponsor using the data set that was annotated earlier this semester by Samuel.
- Due to time restraints, this model output a range of days for day of growth and was not able to identify the nutrient solution.

- For 404, I will be training our own deep learning model.
- The primary goal of this model is to accurately predict the exact day of the growth cycle.
- We would like to be able to achieve identifying the nutrient solution, but we are unsure if this will be possible due to the amount of data that is available to us.

Subsystem: Deep Learning Model

Accomplishments since 403 20 hours of effort	Ongoing progress/problems and plans until the next presentation
Determined the ideal split of training and testing data out of the dataset we were given. Researched various libraries and functions within Tensorflow & Keras for building our own CNN and DL model	Currently implementing the research I did Completing the deep learning model this week

Mary Hughes

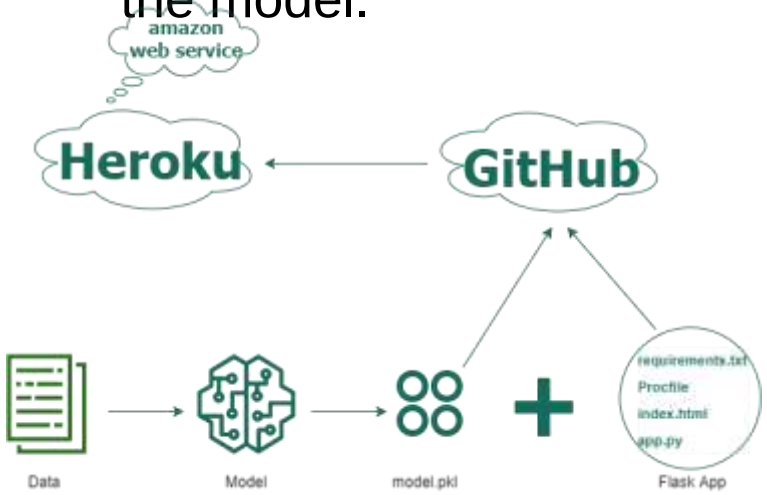
Communication with Model

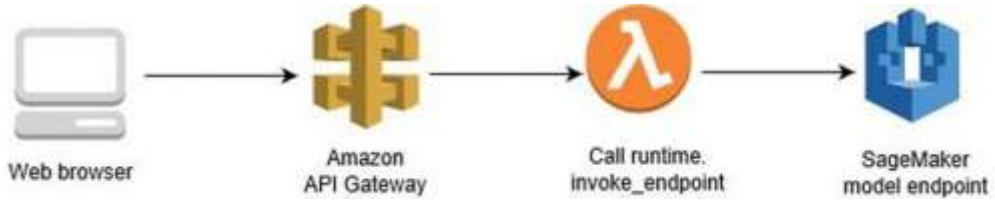
Used for 403 Demo: Heroku

- Use Heroku to host both the front end and backend (separately).
- Built backend in Flask, used GET and POST HTTP requests to communicate with the model.

Goal for 404: AWS

- Deploy Model
- Create REST API
- Write Lambda function/event handler
- Verify IAM roles, permissions, and policies





Mary Hughes

Communication with Model (Flask Back End)

```python
@app.route("/predict", methods=['GET','POST'])
def user_upload():
    if request.method == 'POST':
        #need to get image from POST request
        f = request.files["image"]
        print(request.files)
        # #create img_path to call model
        basepath = os.path.dirname(__file__)
        img_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
        f.save(img_path)
        # #call model
        pred = model_predict(img_path)
        pred = pred.tolist()
        output = output_statement(pred)
        os.remove(img_path)
        return {"message": output["message"], "accuracy": output["accuracy"]}

    elif request.method == 'GET':
        response = {}
        response["MESSAGE"] = "Soybean Prediciton API is running!"
        return response
```

Samuel He

User Interface

Finished:

Deployed End to End Connections ~40 hours

- Back End:
  - CORS permissions to prevent unwanted access

Access-Control-Allow-Origin

-Dependencies:

-Heroku dependencies and buildpack fixes

-Tensorflow is picky

```
absl-py==1.3.0
astunparse==1.6.3
cachetools==5.2.0
certifi==2022.9.24
charset-normalizer==2.1.1
click==8.1.3
colorama==0.4.6
Flask==2.2.2
flatbuffers==22.11.23
gast==0.4.0
google-auth==2.14.1
google-auth-oauthlib==0.4.6
google-pasta==0.2.0
grpcio==1.51.1
gunicorn==20.1.0
h5py==3.7.0
idna==3.4
itsdangerous==2.1.2
Jinja2==3.1.2
keras==2.10.0
libclang==14.0.6
Markdown==3.4.1
MarkupSafe==2.1.1
numpy==1.23.5
oauthlib==3.2.2
```

```
opt-einsum==3.3.0
packaging==21.3
protobuf==3.19.6
pyasn1==0.4.8
pyasn1-modules==0.2.8
pyparsing==3.0.9
requests==2.28.1
requests-oauthlib==1.3.1
rsa==4.9
six==1.16.0
tensorboard==2.10.0
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
tensorflow-cpu==2.10.0
tensorflow-estimator==2.10.0
tensorflow-io-gcs-filesystem==0.28.0
Pillow==9.2.0
termcolor==2.1.1
Flask-Cors==3.0.10
touch==2020.12.3
typing_extensions==4.4.0
urllib3==1.26.13
values==2020.12.3
Werkzeug==2.2.2
wrapt==1.14.1
```

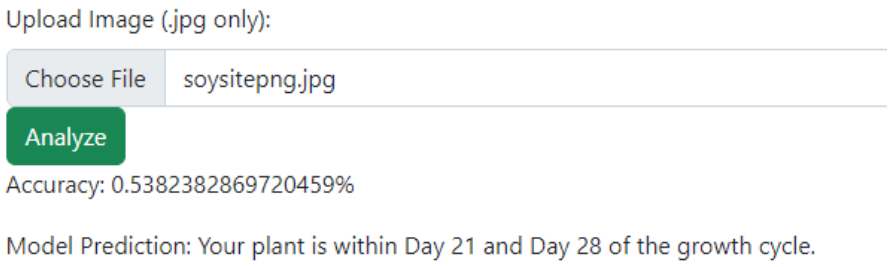
Samuel He

User Interface (front end)

In Progress/To do:

UI Styling: ~1 hours

- ReactJS (Web Framework)
- Bootstrap (CSS Library)

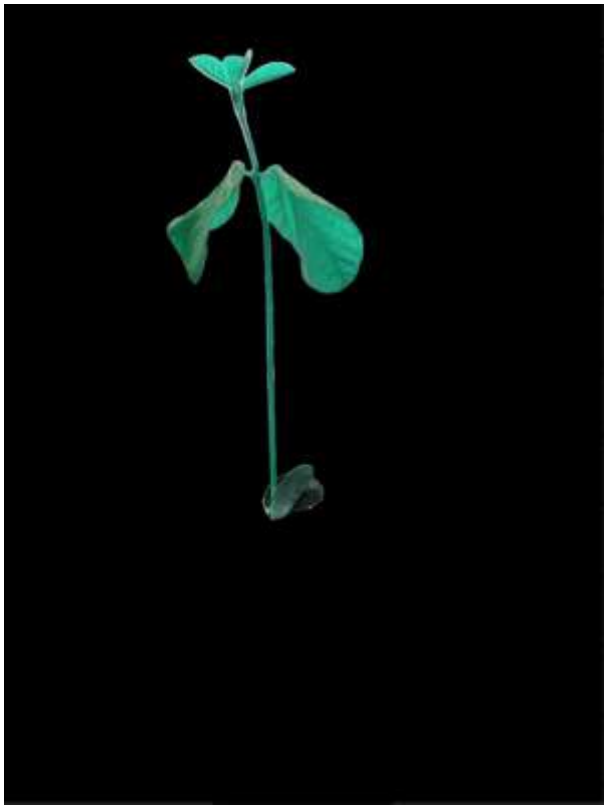


Samuel He

Image Processing

- Image annotation – CVAT (Computer Vision Annotation Tool)
  - Using segmentation and classification for feature extraction purposes

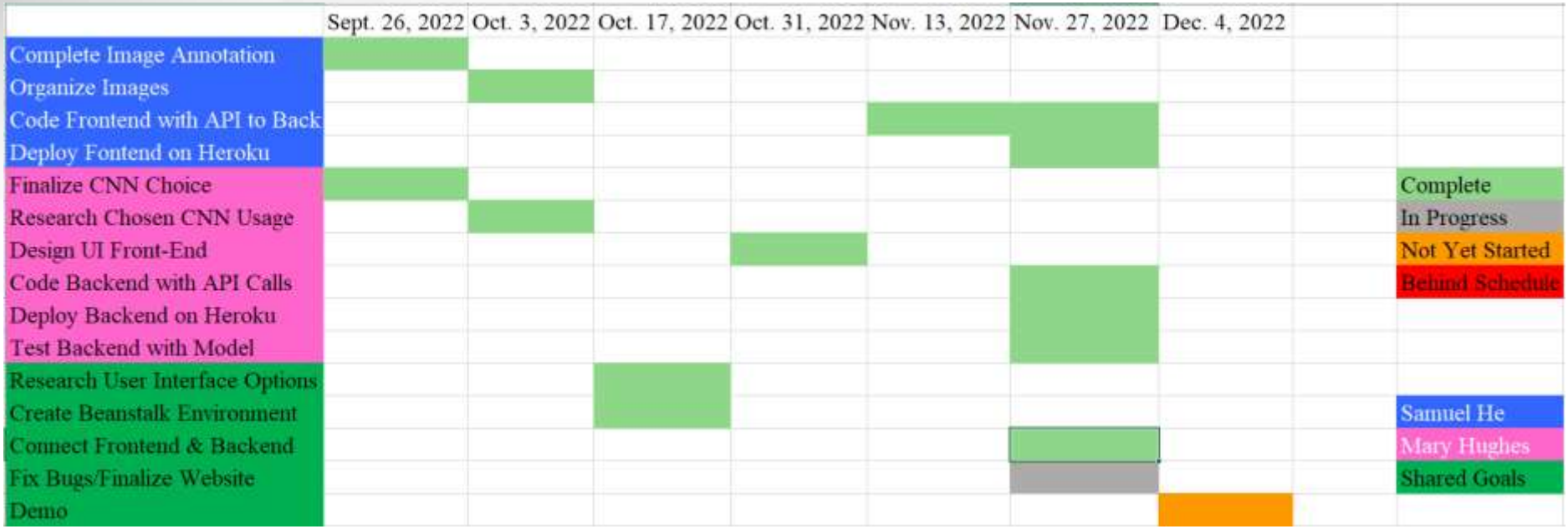
CVAT segmentation



Future Goals & Plans

- 404:
  - Training and testing our own deep learning model from scratch
  - Potentially expanding the scope of the model
  - Perfecting web app
    - Security
    - Testing
    - Styling

Execution Plan



Validation plan

Paragraph #	Test Name	Success Criteria	Status	Responsible Engineers
3.2.1.3	UI Front End Functionality	The User Interface works as expected, users can upload an image, webpage interacts as intended.	COMPLETE	Samuel He
3.2.1.3	Input Delivery to Back End	Input is successfully being delivered to the backend from the front end of the UI.	COMPLETE	Samuel He
3.2.1.3	UI Backend Communication with Model	The User Interface Back End API calls work as expected, and can return a prediction in a 3rd party testing platform.	COMPLETE	Mary Hughes
3.2.3.2.1	UI Output Delivery	An output is being delivered in the correct format to the UI.	COMPLETE	Mary Hughes
3.2.1.3	UI Readability	UI design is clean and understandable, easy to use.	COMPLETE	Shared
3.2.1.1	Day of Growth Identification (404)	The deep learning model is correctly identifying the day of growth of an input.	UNTESTED	Mary Hughes
3.2.1.2	Nutrient Solution Detection (404)	The deep learning model is correctly identifying the nutrient solution of an input.	UNTESTED	Mary Hughes
3.2.5.1.1	Application Failure Detection (404)	Internal testing properly identifies when the application fails to communicate with the deep learning model.	UNTESTED	Samuel He
3.2.5.1.1.1	Model Failure Detection (404)	Application correctly detects if the model has given a valid input to the UI.	UNTESTED	Samuel He
N/A	Full System Demo	The application and deep learning model process input as expected and deliver correct output to the UI.	UNTESTED	Shared

Questions?