

Zero to Web App ⚡

Rapid Customized Web Interfaces for your ML Applications with Streamlit

Michelle Yung

Software Engineer
Data Science Institute, University of Arizona

myung@arizona.edu

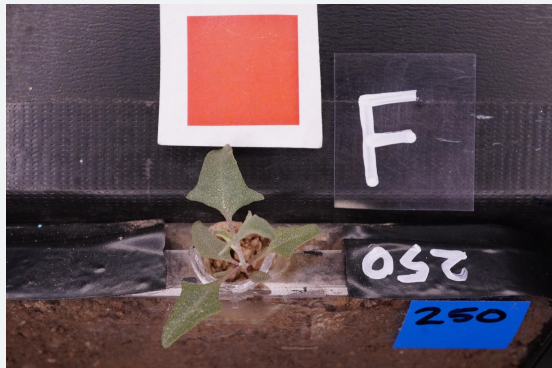


RESEARCH, INNOVATION & IMPACT
Data Science Institute



Superfund
Research Center

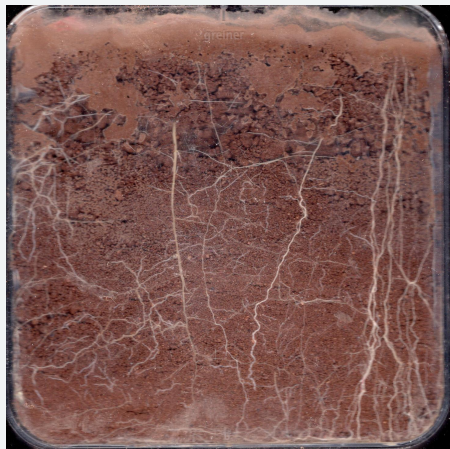
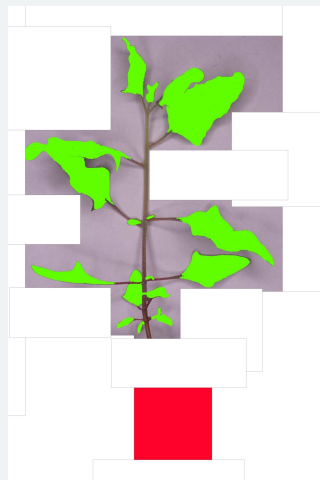
Current workflow



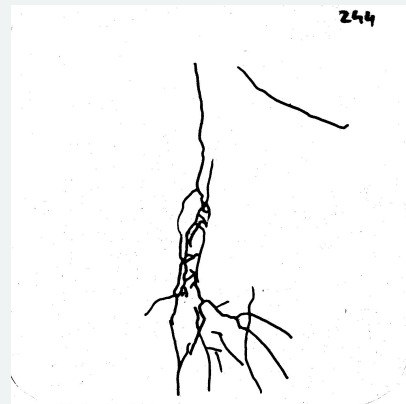
Images are renamed,
organized by date



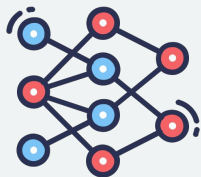
Leaves are
manually colored
using ImageJ



Roots are
manually traced
with sharpie and
scanned

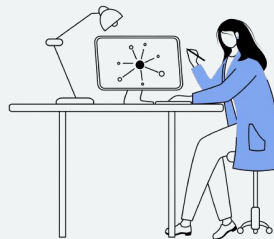


How can we automate the process?



Machine Learning

Train machine learning model to detect leaves for us!



Streamlit App

We have a model, now what? How can we share this?

Allows ML models to be shared, researchers can run model on new data independently

Streamlit

“Low-code” solution for building web applications

MyApp.py

```
import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```

<https://streamlit.io>

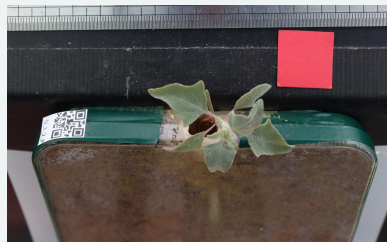
My App • Streamlit

My first app

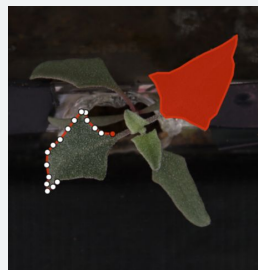
Hello world!



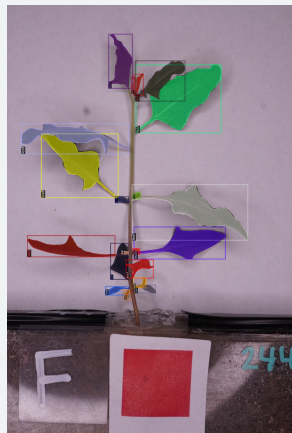
New, improved workflow using ML



Add unique QR codes for each plant



Annotate images



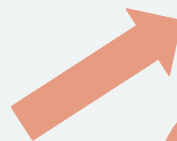
Train machine learning model to detect leaves



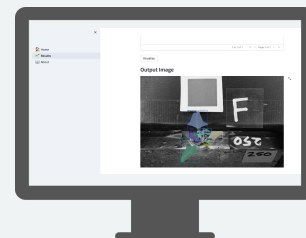
Streamlit

Build Streamlit app

(Hosted on CyVerse as VICE app)



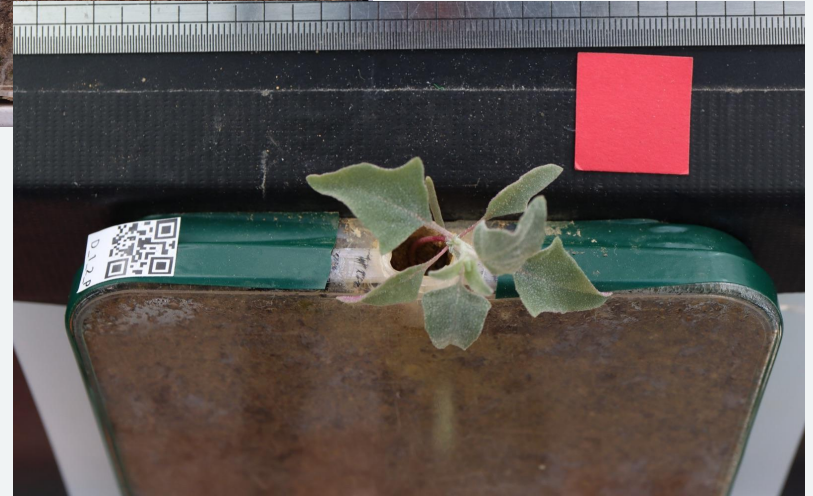
Researcher uploads new data to CyVerse



Researcher can use Streamlit app to run ML model on new data

QR codes

- Automatically identify each plant
- Can use this information to rename files automatically or attach metadata



Annotate images

- Open source software for labeling data for machine learning
- Allows many people to collaborate on labeling data
- Supports many types of data: audio, text, time series, and images



Label Studio

<https://labelstud.io>

#497 AD Admin #twLOU created, 6 minutes ago 1/1 [Icons: Grid, Undo, Redo, Close, Delete, Full Screen]

Select label and click the image to start

qr 1 leaf 2 red-square 3

```
"categories": [
  {
    "id": 0,
    "name": "leaf"
  },
  {
    "id": 1,
    "name": "qr"
  },
  {
    "id": 2,
    "name": "red-square"
  }
],
"annotations": [
  {
    "id": 0,
    "image_id": 0,
    "category_id": 2,
    "segmentation": [
      760.5318672147781,
      3206.977875578077,
      1819.8860175343202,
      3223.433862379157,
      1821.943015884456,
      2209.3336757625852,
      768.7598606153183,
      2209.3336757625852
    ]
  },
  {
    "bbox": [
      760.5318672147781,
      2209.3336757625852,
      1061.411148669678,
      1014.1001866165716
    ],
    "ignore": 0,
    "iscrowd": 0,
    "area": 1062513.5788337104
  },
  {
```

Machine Learning Model

- Use labeled data to “train” an ML model
- We are using: Mask-RCNN model w/ Detectron2 (object detection library built on Pytorch)

Final output is .pth file
(pytorch ML model weights)

The screenshot shows a Jupyter Notebook environment. On the left is a file explorer for the directory `/work/output/`. It contains a table of files:

Name	Last Modified
balloon_val_c...	10 days ago
coco_instanc...	3 days ago
dataset_dicts...	10 days ago
events.out.tfe...	10 days ago
events.out.tfe...	10 days ago
events.out.tfe...	10 days ago
events.out.tfe...	10 days ago
events.out.tfe...	10 days ago
events.out.tfe...	3 days ago
events.out.tfe...	3 days ago
instances_pre...	3 days ago
last_checkpoint	3 days ago
metrics.json	3 days ago
model_00049...	3 days ago
model_final.pth	3 days ago

The main area shows a Jupyter Notebook with a code cell [12] that loads and visualizes dataset information:

```
[12]: leaf_metadata = MetadataCatalog.get("train")
dataset_dicts = DatasetCatalog.get("train")

print(leaf_metadata)

for d in random.sample(dataset_dicts, 3):
    img = cv2.imread(d["file_name"])
    visualizer = Visualizer(img[:, :, ::-1], metadata=leaf_metadata)
    out = visualizer.draw_dataset_dict(d)
    plt.imshow(out.get_image()[:, :, ::-1])
    plt.show()
```

Below the code, a warning message is displayed:

```
WARNING [08/26 05:12:34 d2.data.datasets.coco]:
Category ids in annotations are not in [1, #categories]
```

Then, a message indicates that 152 images were loaded:

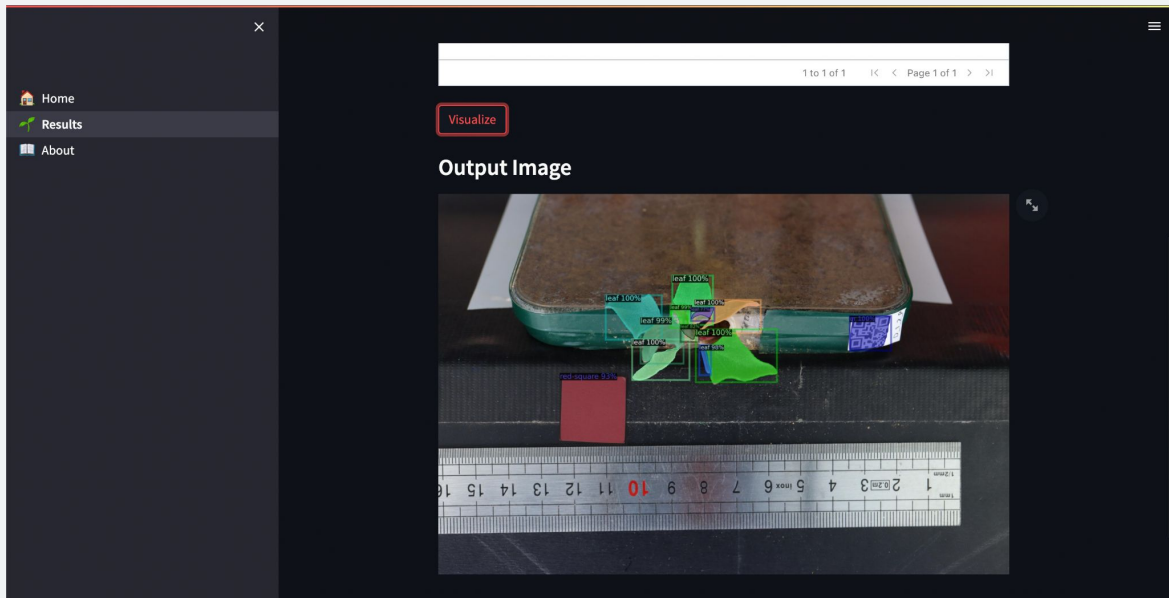
```
[08/26 05:12:34 d2.data.datasets.coco]: Loaded 152 images
Metadata(name='train', json_file='/home/jovyan/work/dataset_dicts/train.json', r_type='coco', thing_classes=['leaf'], thing_dataset_images=152)
```

The visualization shows a photograph of a blue smartphone with a small green leaf-like object on its back. A blue bounding box is drawn around the object. The image is overlaid on a ruler with a coordinate system on the left (0 to 2000) and bottom (0 to 3000).

The bottom status bar shows: Simple 0 s 0 Python 3 (ipykernel) | Busy

So we have a model.. now what?


Streamlit



```
#----- STREAMLIT INTERFACE -----

st.title('Leaf Segmentation')
st.header('Files')

# walk through directory to display files in table
file_names = []
dirs = []

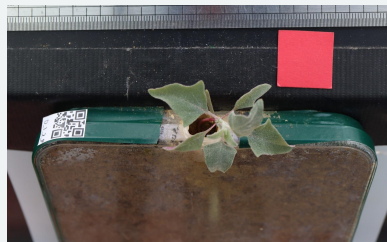
for root, dirs, files in os.walk(base_path + "data"):
    for file in files:
        filename=os.path.join(root, file)
        file_names.append(filename)

# set up AgGrid
df = pd.DataFrame({'File Name' : file_names})
gd = GridOptionsBuilder.from_dataframe(df)
gd.configure_pagination(enabled=True)
gd.configure_selection(selection_mode="single", use
gd.configure_column("File Name", headerCheckboxSele

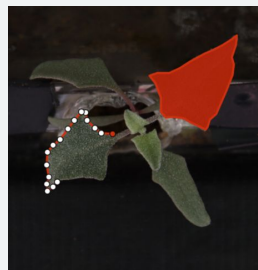
# display AgGrid
file_table = AgGrid(df, fit_columns_on_grid_load=True
update_mode=GridUpdateMode.SELECTION_CHANGED)

run = st.button('Run')
```

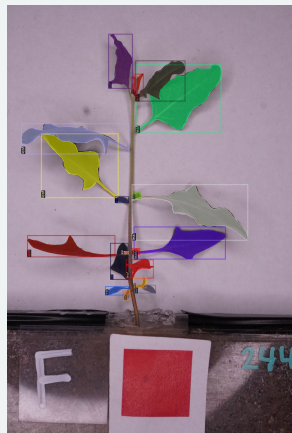
New, improved workflow using ML



Add unique QR codes
for each plant



Annotate images



Train machine
learning model to
detect leaves

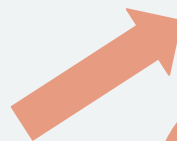
Model
weights



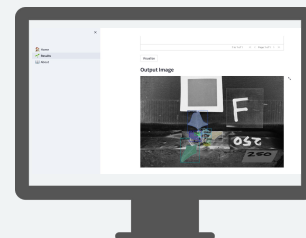
Streamlit

Build Streamlit app

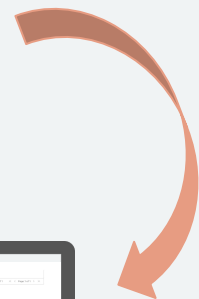
(CyVerse VICE app)



Researcher
uploads new
data to
CyVerse



Researcher can use Streamlit app
to run ML model on new data



Where to host your app?



- VICE App: launch it when you need it
- Can easily share your app with collaborators
- Built-in authentication
- Connects directly to data store
- Cost: Subscription model (free basic tier)



- App always available
- Can deploy unlimited public apps, but only one private app
- Previously \$250/month, no paid options available currently

Resources

Code

- Streamlit app and dockerfile for VICE app:
<https://github.com/michellito/streamlit-leaf>

DMAC Team



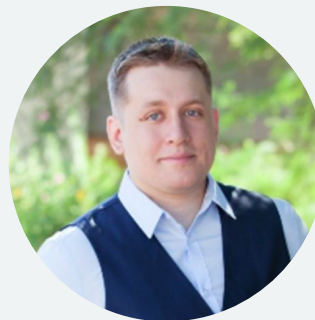
Dr. Aikseng Ooi

Principal Investigator



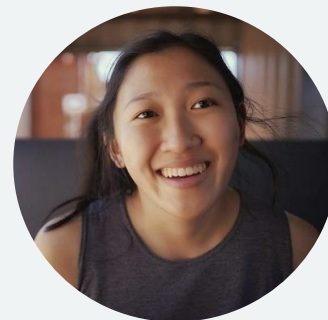
Nirav Merchant

Co-Investigator



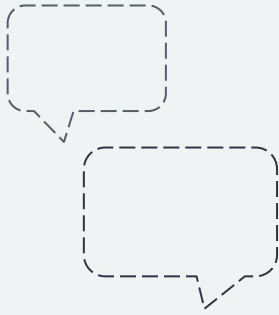
Anthony Vicenti

Statistician &
Computational Biologist

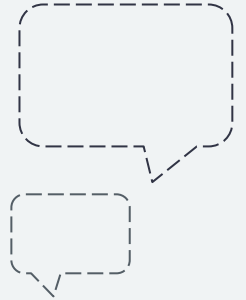


Michelle Yung

Software Engineer



What questions do you have?



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**