Assessment of Herbicide Efficiency on Treating Invasive Species

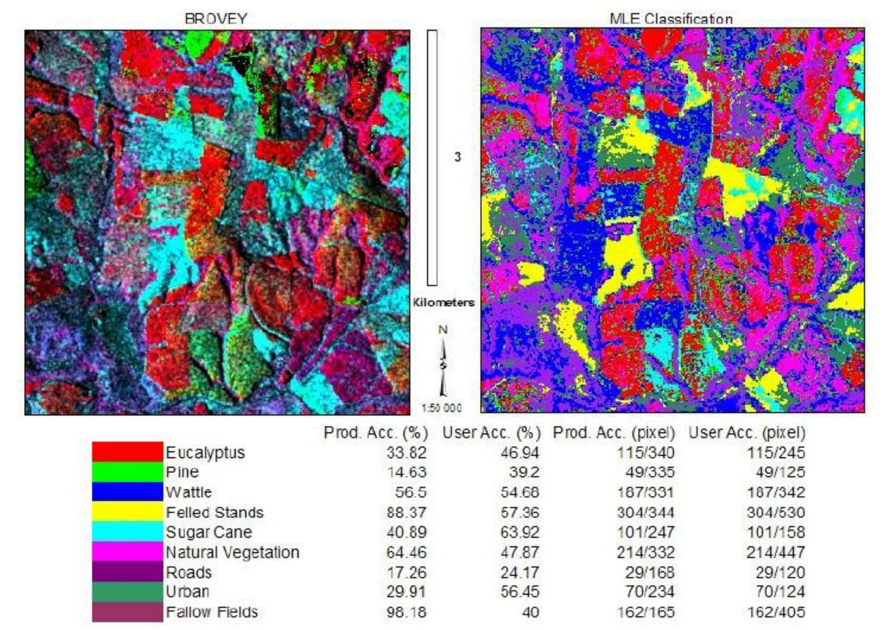
CS29: Jack McConnell and Mason Hall
with partner Bogdan Strimbu

Introduction

- Reed Canary grass is an invasive species present in Oregon
- Herbicides are the only practical method of treatment
- New EPA regulations restrict use of many herbicides
- An ongoing project is underway to test herbicides within regulations

Introduction

- Our task was to analyze data provided from the team testing these herbicides
- Our product was not a program, but the results of this data analysis
- The actual analysis was in the form of image classification

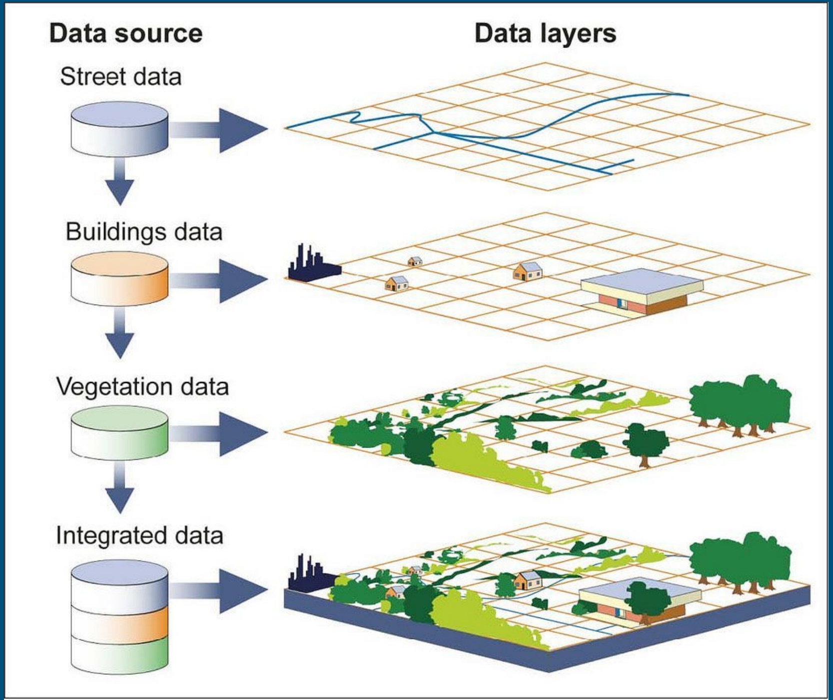


		Prod. Acc. (%)	User Acc. (%)	Prod. Acc. (pixel)	User Acc. (pixel)
🟥	Eucalyptus	33.82	46.94	115/340	115/245
🟩	Pine	14.63	39.2	49/335	49/125
🟦	Wattle	56.5	54.68	187/331	187/342
🟨	Felled Stands	88.37	57.36	304/344	304/530
🟦	Sugar Cane	40.89	63.92	101/247	101/158
🟪	Natural Vegetation	64.46	47.87	214/332	214/447
🟪	Roads	17.26	24.17	29/168	29/120
🟩	Urban	29.91	56.45	70/234	70/124
🟪	Fallow Fields	98.18	40	162/165	162/405

Example of image classification (not our data)

Our Methods: QGIS

- QGIS: A Geographic Information System
- Stores visual information alongside other data (in our case, reflectance on 4 wavelengths)
- Used both to manipulate images and to analyze them

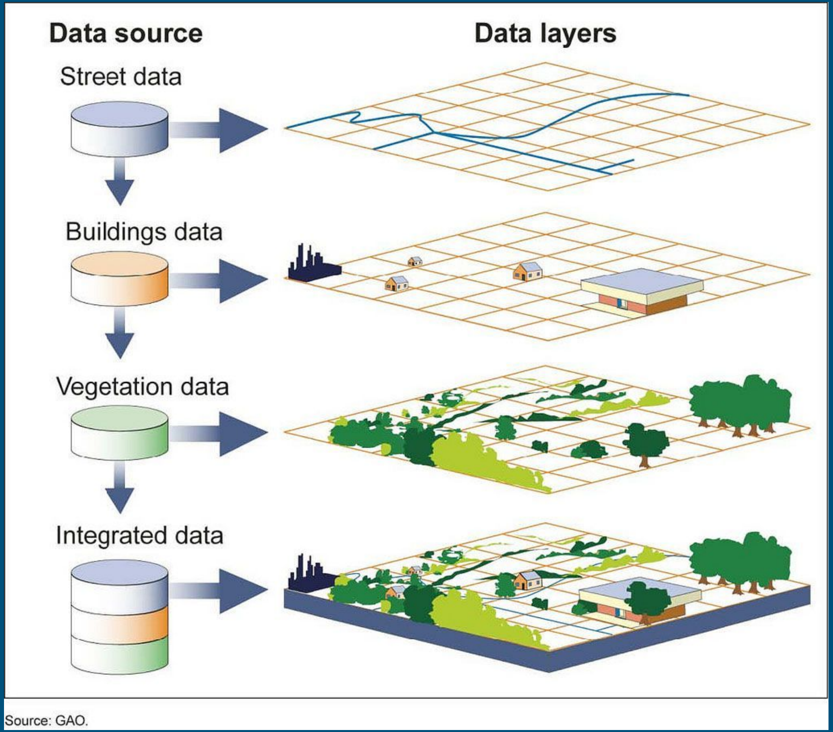


Source: GAO.

Example of Raster Images (.tif) (not our image)

Our Methods: QGIS

- We first used this to crop the overall image into individual treatment plots
- Later we also used QGIS to gather statistics on each plot's reflectance data

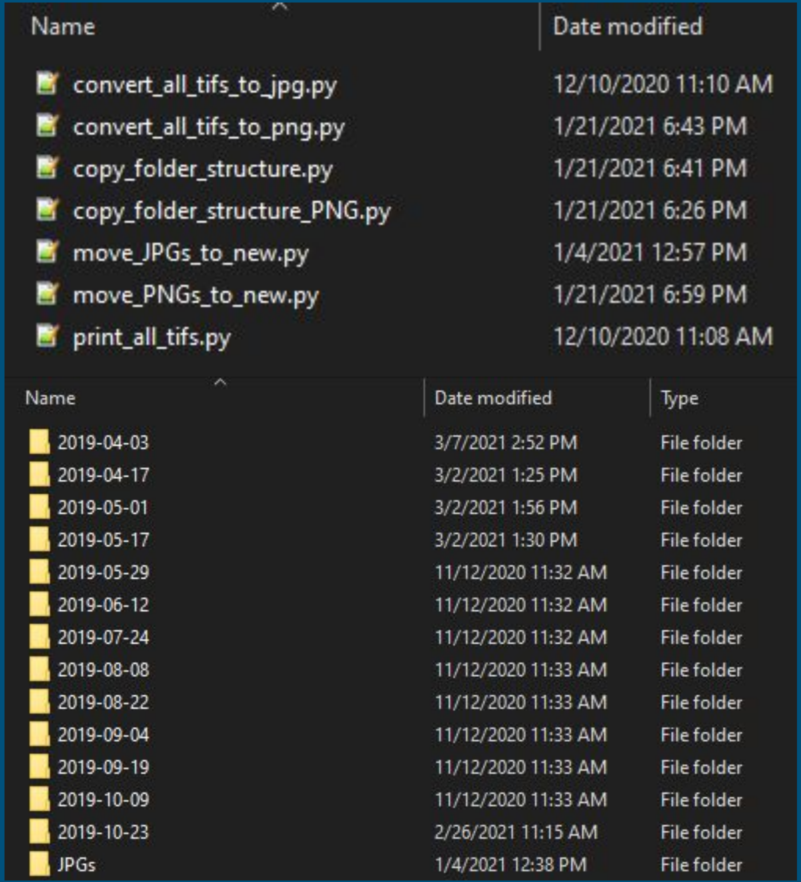


Source: GAO.

Example of Raster Images (.tif) (not our image)

Our Methods: Python

- Thousands of images
- Each image needed to be both organized and manipulated
- Python scripts were used for both tasks
- Later we used Python to implement MLE (more details on that later)

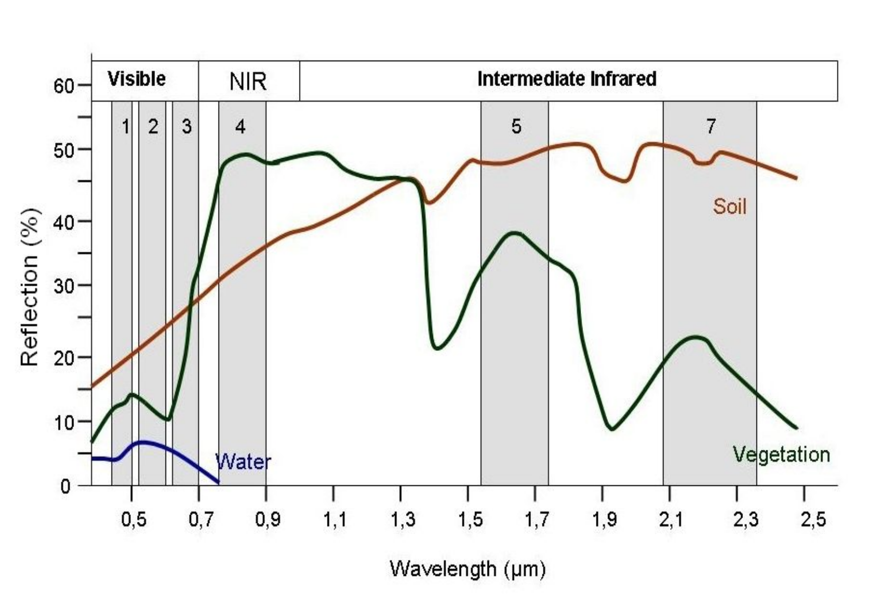


Name	Date modified
convert_all_tifs_to_jpg.py	12/10/2020 11:10 AM
convert_all_tifs_to_png.py	1/21/2021 6:43 PM
copy_folder_structure.py	1/21/2021 6:41 PM
copy_folder_structure_PNG.py	1/21/2021 6:26 PM
move_JPGs_to_new.py	1/4/2021 12:57 PM
move_PNGs_to_new.py	1/21/2021 6:59 PM
print_all_tifs.py	12/10/2020 11:08 AM

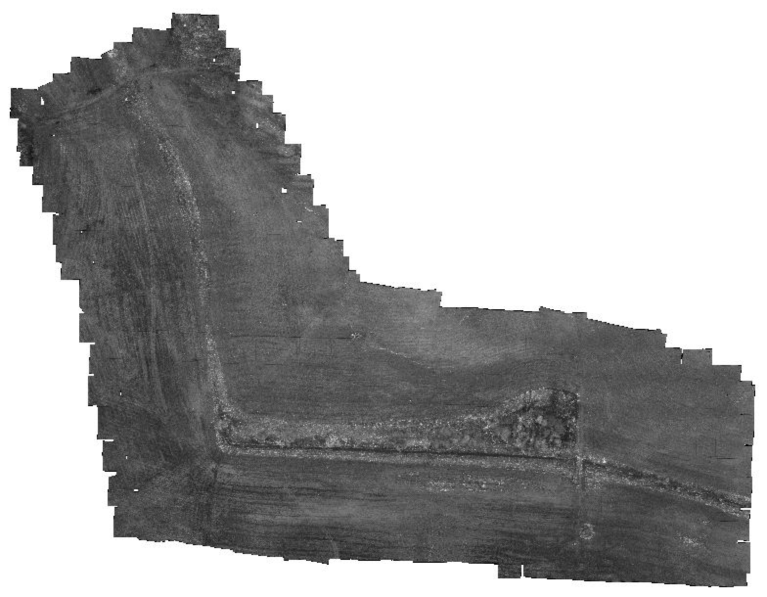
Name	Date modified	Type
2019-04-03	3/7/2021 2:52 PM	File folder
2019-04-17	3/2/2021 1:25 PM	File folder
2019-05-01	3/2/2021 1:56 PM	File folder
2019-05-17	3/2/2021 1:30 PM	File folder
2019-05-29	11/12/2020 11:32 AM	File folder
2019-06-12	11/12/2020 11:32 AM	File folder
2019-07-24	11/12/2020 11:32 AM	File folder
2019-08-08	11/12/2020 11:33 AM	File folder
2019-08-22	11/12/2020 11:33 AM	File folder
2019-09-04	11/12/2020 11:33 AM	File folder
2019-09-19	11/12/2020 11:33 AM	File folder
2019-10-09	11/12/2020 11:33 AM	File folder
2019-10-23	2/26/2021 11:15 AM	File folder
JPGs	1/4/2021 12:38 PM	File folder

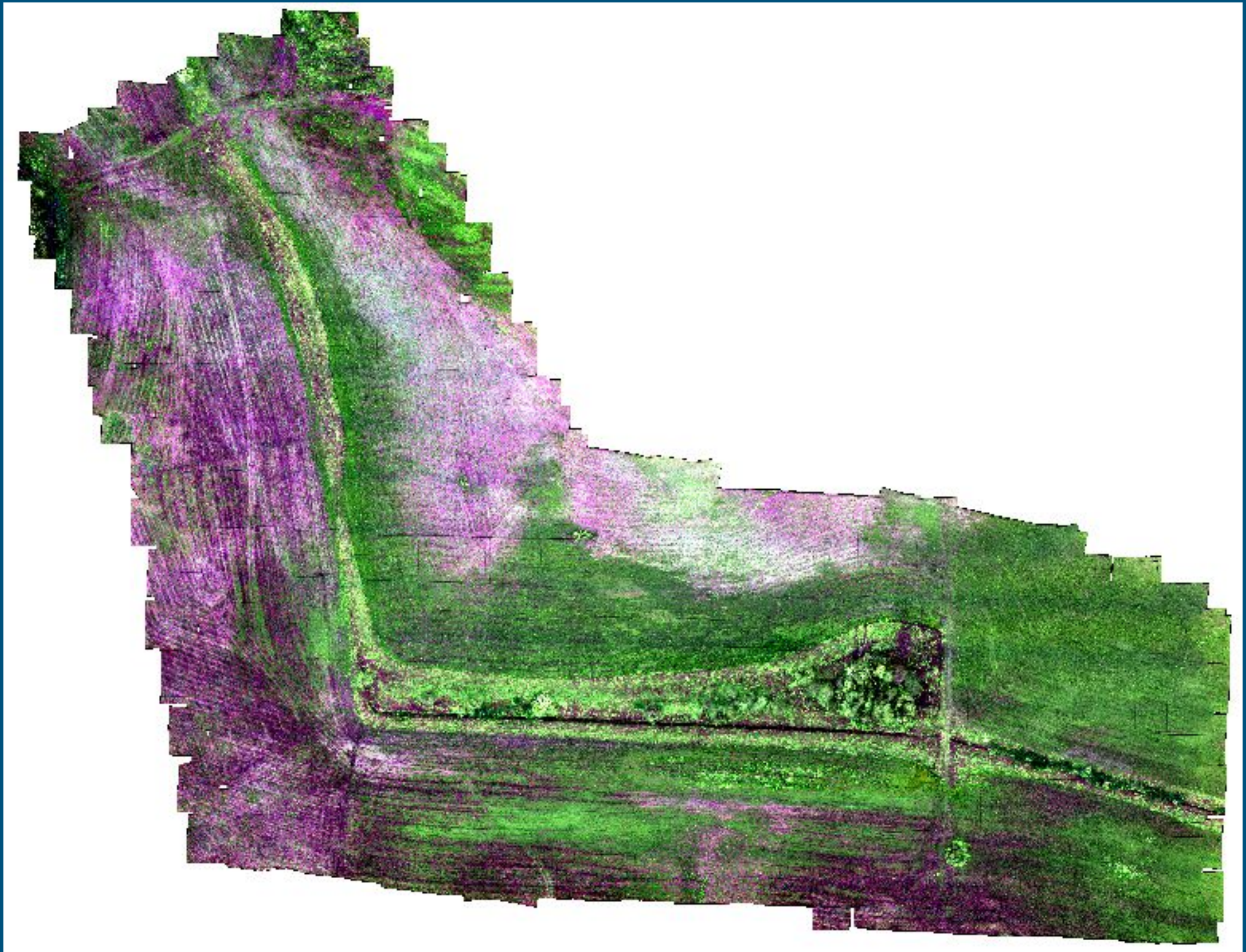
The Technique: Spectral Signature Analysis

- Our .tif flyover images were taken in 4 wavelengths
- Each wavelength has a reflectance value for each pixel
- Each pixel’s four values create a “spectral signature”
- This signature is used to classify each pixel



Example of Different Spectral Signatures (Not our image)



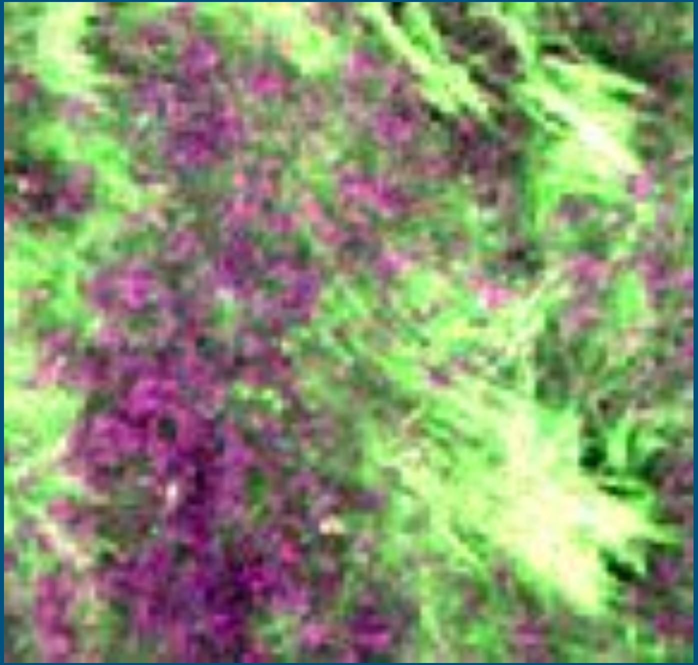


Our Methods: MLE

- Maximum Likelihood Estimation is a method used to estimate which of several classes an object is actually in
- MLE uses multiple variables to place the objects of interest into one of several classes
- Provided images that captured 4 bands (4 different wavelengths of light), our MLE implementation used 4 variables
- To accomplish this, sections of images known to be in certain categories were identified

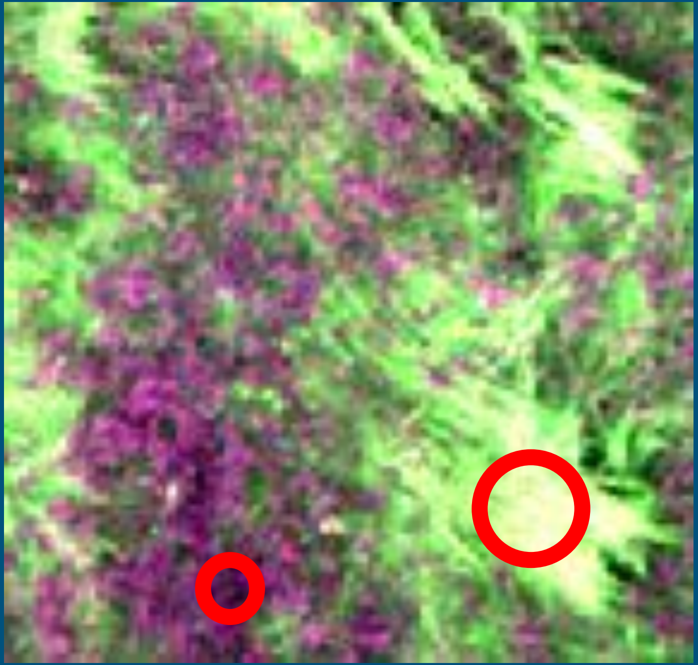
Our Methods: MLE

- Known objects in each class are identified, known as regions of interest (ROIs)
- Statistics of all 4 variables within each ROI are recorded
- These statistics are used to estimate which class every pixel in the image would most likely fall into



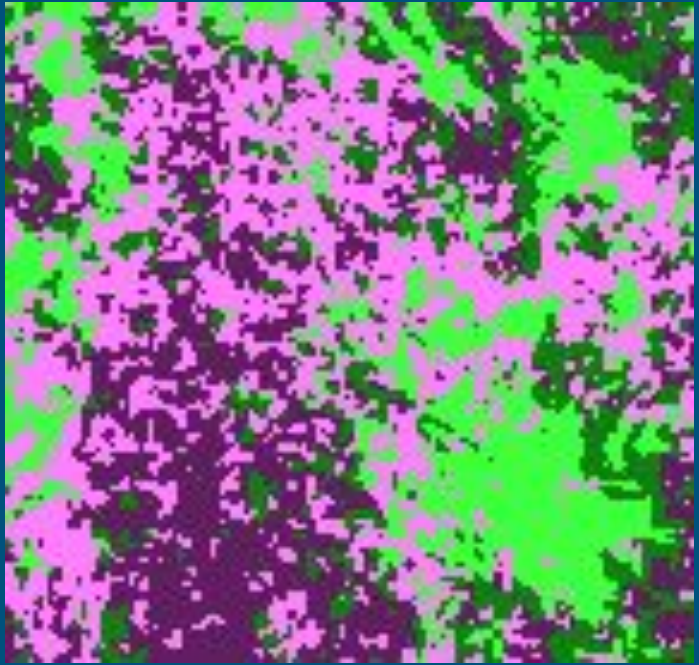
Our Methods: MLE

- Known objects in each class are identified, known as regions of interest (ROIs)
- Statistics of all 4 variables within each ROI are recorded
- These statistics are used to estimate which class every pixel in the image would most likely fall into



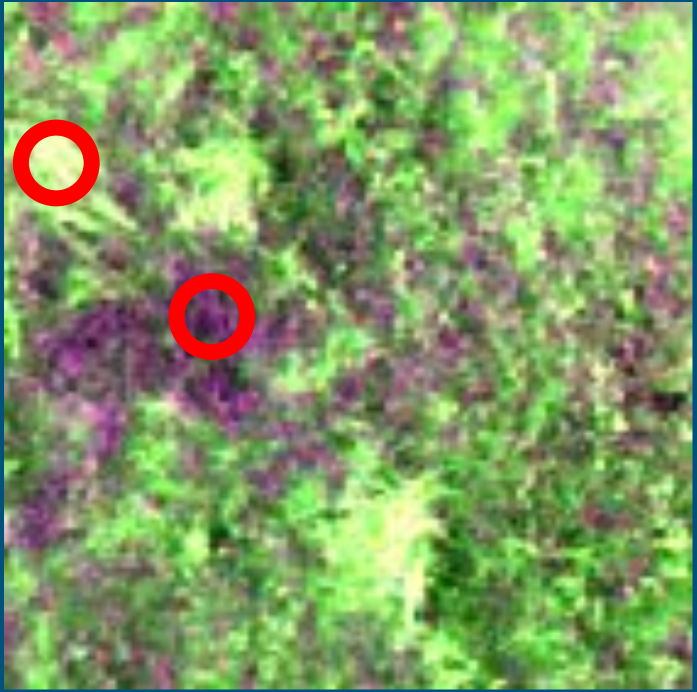
Our Methods: MLE

- Known objects in each class are identified, known as regions of interest (ROIs)
- Statistics of all 4 variables within each ROI are recorded
- These statistics are used to estimate which class every pixel in the image would most likely fall into



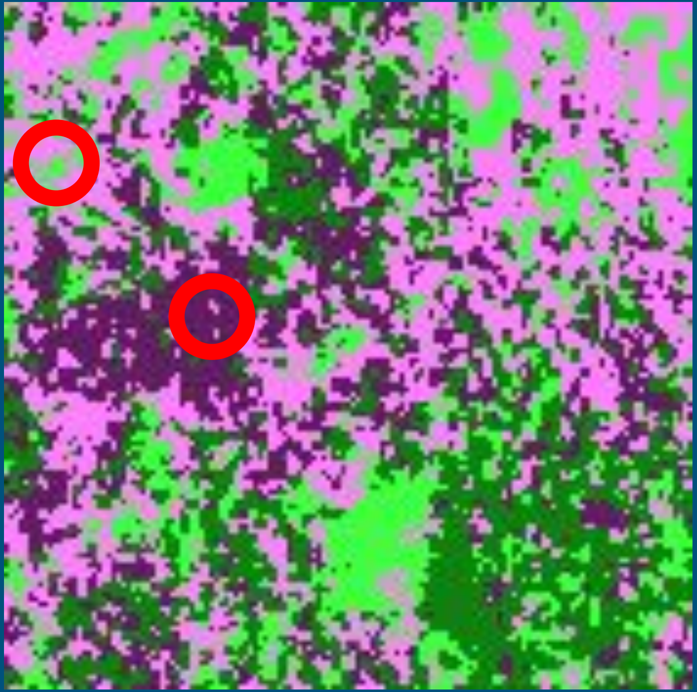
Our Methods: Confusion Matrix

- For each class, identify regions that should have been placed in that class
- Check how many pixels in these regions were actually placed into that class
- With some math, this yields an estimate on the overall accuracy



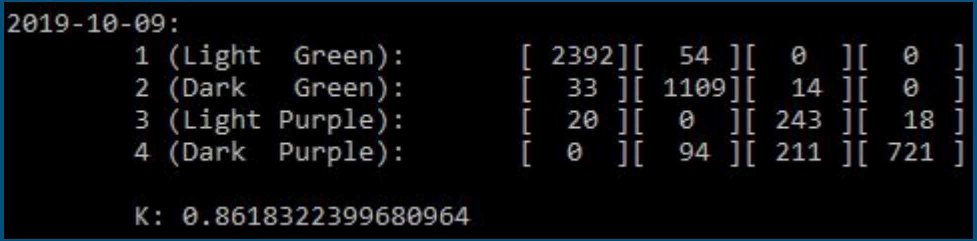
Our Methods: Confusion Matrix

- For each class, identify regions that should have been placed in that class
- Check how many pixels in these regions were actually placed into that class
- With some math, this yields an estimate on the overall accuracy



Our Methods: Confusion Matrix

- For each class, identify regions that should have been placed in that class
- Check how many pixels in these regions were actually placed into that class
- With some math, this yields an estimate on the overall accuracy

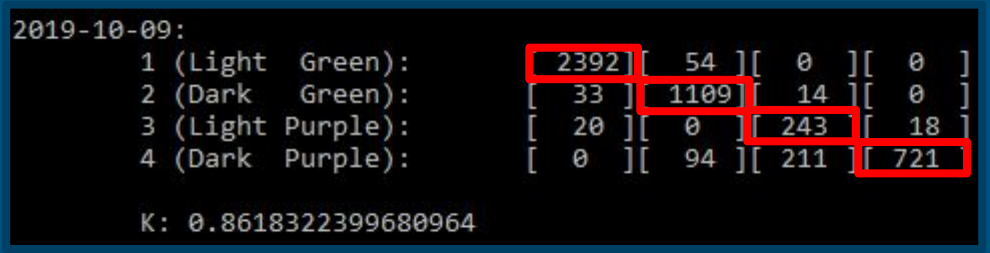


```
2019-10-09:
        1 (Light  Green):       [ 2392][  54 ][  0  ][  0  ]
        2 (Dark   Green):       [  33 ][ 1109][  14 ][  0  ]
        3 (Light Purple):       [  20 ][  0  ][ 243 ][  18 ]
        4 (Dark  Purple):       [  0  ][  94 ][ 211 ][ 721 ]

        K: 0.8618322399680964
```

Our Methods: Confusion Matrix

- For each class, identify regions that should have been placed in that class
- Check how many pixels in these regions were actually placed into that class
- With some math, this yields an estimate on the overall accuracy



```
2019-10-09:
        1 (Light  Green):        [ 2392][  54 ][  0  ][  0  ]
        2 (Dark   Green):        [  33 ][ 1109][  14 ][  0  ]
        3 (Light Purple):        [  20 ][  0  ][ 243 ][  18 ]
        4 (Dark  Purple):        [  0  ][  94 ][ 211 ][ 721 ]

        K: 0.8618322399680964
```

Overview of Methods: Flowchart

.TIF Images
of Entire Area

Shapefile of
Individual Plots

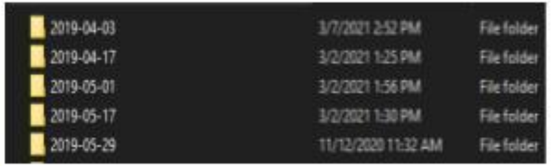
QGIS

QGIS

Cropping by an area
designated by the shapefile

Cropped .TIF Images
of Each Plot

Python
Scripts

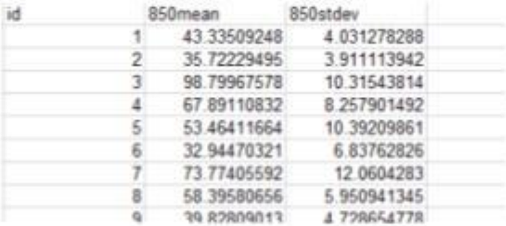


```
2019-04-03    3/7/2021 2:52 PM      File folder
2019-04-17    3/2/2021 1:25 PM      File folder
2019-05-01    3/2/2021 1:56 PM      File folder
2019-05-17    3/2/2021 1:30 PM      File folder
2019-05-29    11/12/2020 11:32 AM   File folder
```

All Plots Organized
By Plot and Date

QGIS

“Raster Statistics” Tool



id	850mean	850stdev
1	43.33509248	4.031278288
2	35.72229495	3.911113942
3	98.79967578	10.31543814
4	67.89110832	8.257901492
5	53.46411664	10.39209861
6	32.94470321	6.83762826
7	73.77405592	12.0604283
8	58.39580656	5.950941345
9	39.82809013	4.728654778

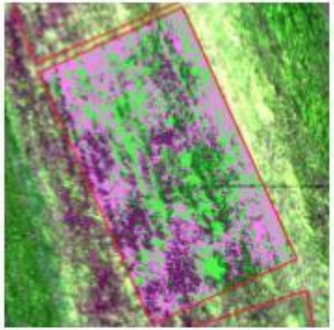
Mean and Std
Deviation for Each
Image Class

MLE

MLE

Maximum
Likelihood
Estimation

% Of Each Class In
Each Plot



Verify

Confusion
Matrix

Final Results for
Analysis

Demo: QGIS

Demo: MLE

Our Github

Thank you!
Any questions?