Marianne Bjorner / Carter Sifferman

Computer Vision (CS 639) Midterm Project Report

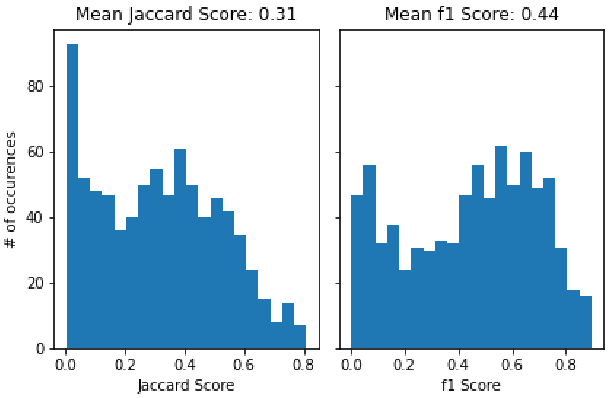
Project Github Repository

Project Website

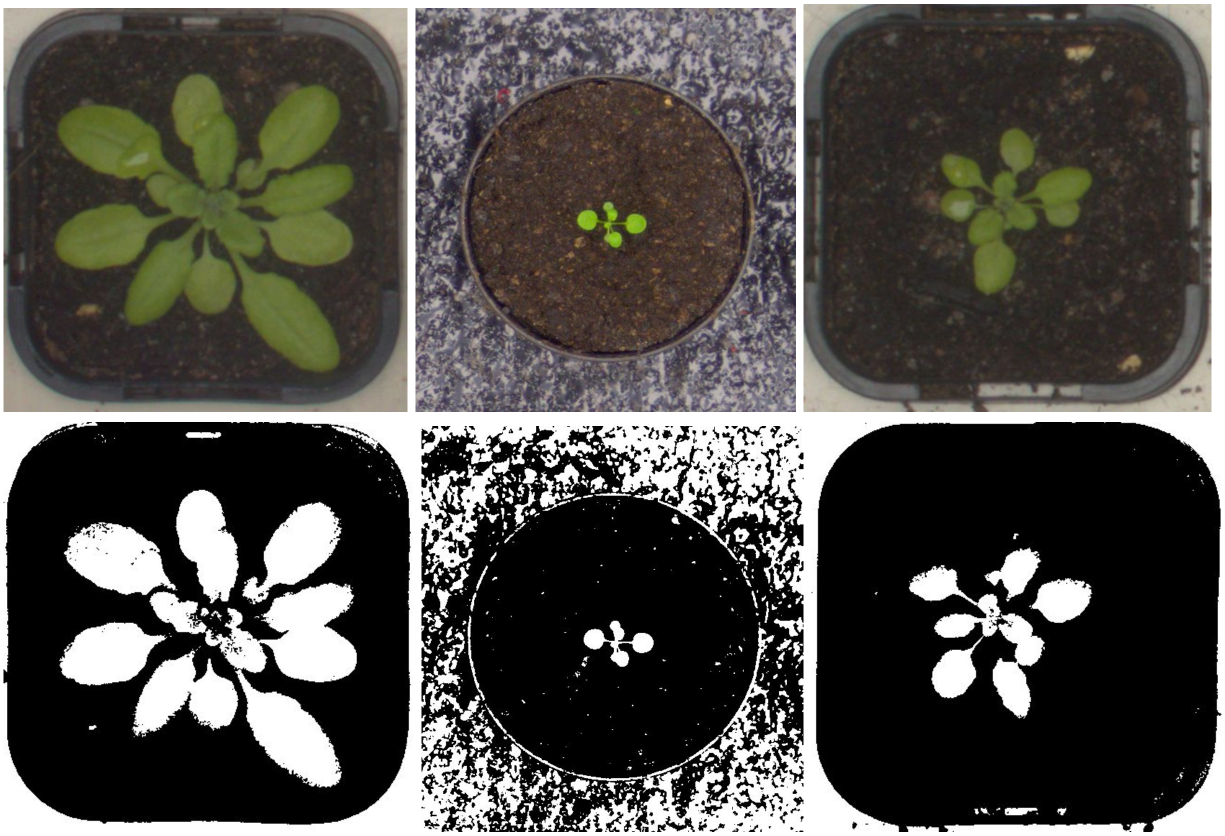
Summary of current progress

Summary: We have set up a framework of Python scripts to run segmentation methods and evaluate their results on the CVPPP2017 dataset. We have also started adapting an existing segmentation algorithm, PlantVision, to work on our RGB images rather than fluorescent images. To create our own challenge dataset, we have gathered and manually labeled 50 overhead plant images. Going forward, we have selected another pixel-color based supervised segmentation method to implement.

Evaluation: Our evaluation script evaluate_segmentation.py evaluates the accuracy of resulting masks against the ground truth provided in CVPPP2017 or our own dataset. It calculates both the Jaccard score and Dice Coefficient (F_1 score), a measure used in CVPPP's own evaluation suite, for each image. The evaluation script saves these results to a text file, as well as plots a histogram as shown below:

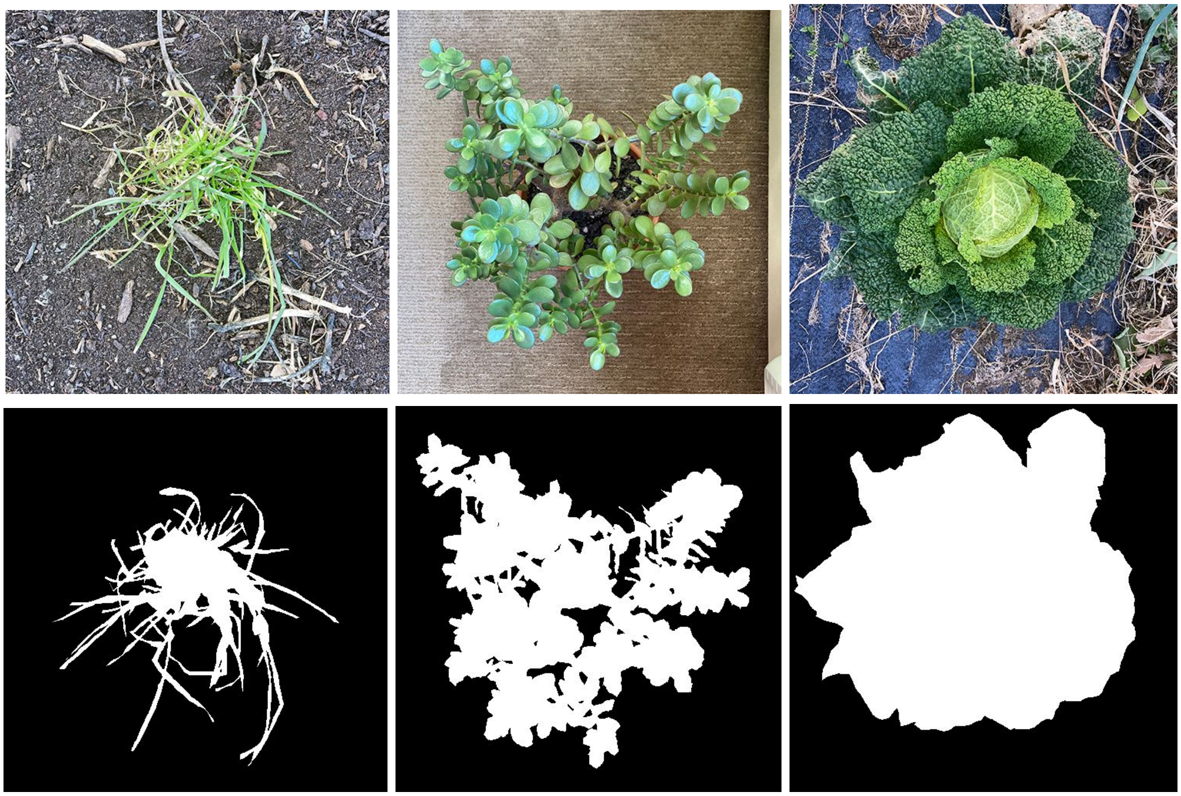


Segmentation: We have written a segmentation script, segment.py, which reads in data from CVPPP2017 or our own dataset, performs segmentation, and saves the resulting images in a format which can be read by our evaluation script. We have implemented one meaningful segmentation algorithm so far, green_channel_thresh, which simply returns pixels which have a green value above 100. We have used this to test the rest of our code so far, and it will be useful as a bare minimum baseline. Now that we have the framework written, we can very easily add more segmentation algorithms just by writing a method in segment.py. Some examples of simple green thresholding are shown below. Given how simple thresholding is, we find these results surprisingly passable, and we think we can improve them with a decision boundary (see "New algorithm to implement" in Timeline/Adjustments section).



Adaptation of PlantVision: We have written a script, runFluorescentMethod.m, that runs an altered version of PlantVision's entrance file MultiLeafTracking.m on the CVPPP2017 dataset. As the original Matlab implementation of PlantVision is designed for use on closely cropped fluorescent images, several adjustments to both the thresholding and input images are needed in order to gather the best resulting masks.

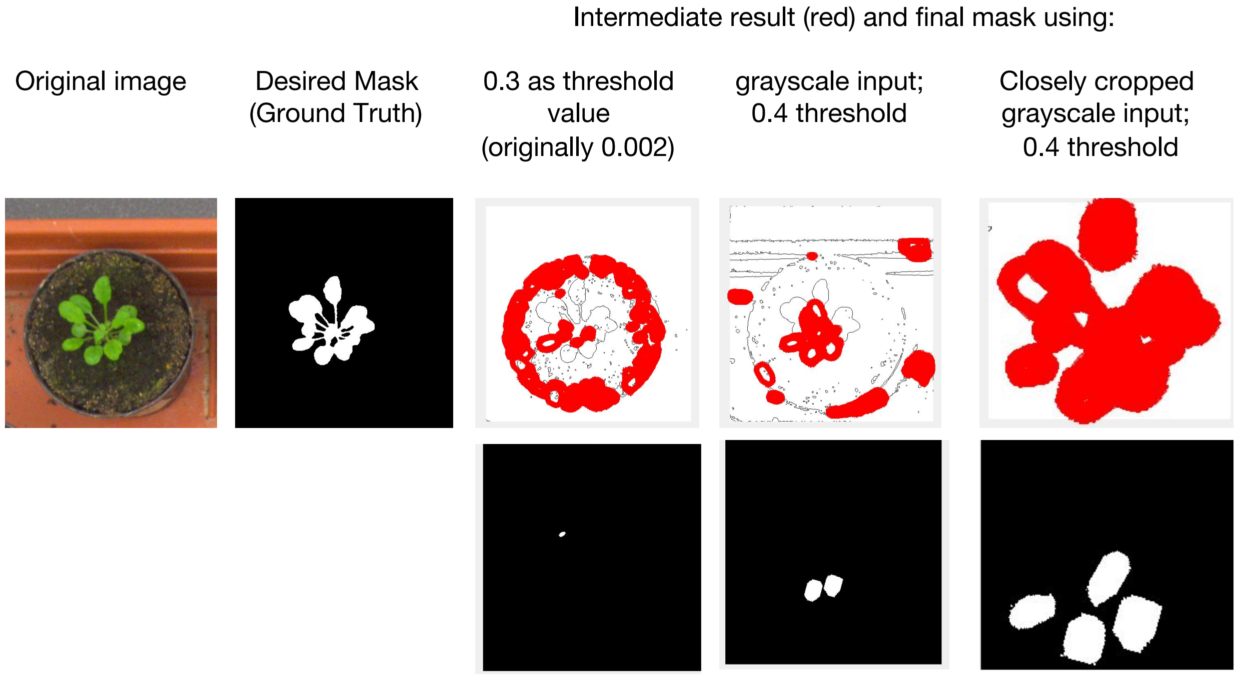
Dataset: We have collected and annotated a total of 47 overhead images of plants. Of these images, 9 are of potted plants indoors and the remaining are of outdoor plants. In contrast to existing plant segmentation datasets which focus on plants with rosette structures, ours includes succulent plants as well as broad leafed plants, crops, and grasses. Some photos in our dataset also present some additional challenges, including heavy shade, various background textures, and leaves that aren't green (e.g. purple). Because our dataset is relatively small, we see it being most useful for evaluation of algorithms that have been trained on a larger dataset such as CVPPP, or as a final step to transfer learning or a similar process which requires little data. Some examples of the range of our dataset, and their binary masks are shown below. Link to our dataset on Google Drive



Challenges

Dataset: Carter did not have great success growing his own plants, so instead we have taken some pictures of plants that both of us already have, as well as plants we have found outside around Madison. Fortunately this has helped toward our original goal of creating a more diverse and challenging dataset.

PlantVision Matlab Adaptation: The original Matlab code for image segmentation provided by PlantVision is optimized for fluorescent images, where the input data have largely already been segmented from the background. The images intended for use on this algorithm are closely cropped, grayscale, and low resolution (~50x50 pixels). Leaf templates are provided that match the intended input plants. Together, this means that an ideal adaptation would not only require a significant amount of preprocessing for plant images, but would also only be suitable for the plant types that match the leaf templates provided. Unchanged, when given the RGB images in the CVPPP2017 LSC dataset, PlantVision results in an undifferentiated binary mask. This improves when the alignment parameters' threshold values are changed, but significant alterations to input images and other parameters are necessary for ideal results, as the leaf detection method is particularly sensitive to background noise. Examples of these alterations are shown below.



Intermediate result (red) and final mask using:

Original image

Desired Mask (Ground Truth)

0.3 as threshold value (originally 0.002)

grayscale input; 0.4 threshold

Closely cropped grayscale input; 0.4 threshold

Timeline / Adjustments

New algorithm to implement: We plan on training a regression algorithm to find a decision boundary between plant and non-plant pixels in RGB space on the CVPPP dataset. This would act as a smarter version of the "simple green thresholding" method that we have implemented so far. Instead of checking if green > 100, we can use the decision boundary. We haven't seen anybody else implement this exactly, the idea is based on the image segmentation lectures we watched in class.

Because of the challenges listed above we have made some adjustments to our expectations and timeline, as shown below:

Adjusted Timeline

Completed	Most of dataset collected and annotated Evaluation script Github Repo and Website
Nov. 15	Complete dataset with 75 total images annotated PlantVision better adapted for use with RGB images
Dec. 1	Implement RGB decision boundary, PlantVision, and at least one other segmentation algorithm and compare performance and generalizability from CVPPP to our dataset.
Dec. 10	Presentation day. Presentation, final report, and website completed