(G52IIP) COMP2005 Coursework 1 Report

By Justin Ng

1. Introduction

This report will describe and explain the steps and methods used to achieve the image processing result in MATLAB. The steps in the methodology describe and explain the process I used for each image sample after loading each image in. I will also be evaluating the strengths and weaknesses to my methodology and features of my obtained results in the evaluation section of this report.

2. Methodology

This section of the report will be describing and explaining each step of the process and approach I used to achieve the final binary images for each provided sample image. Note that this is after reading the three provided images in and storing them as variables.

— Step 1: Isolating colour channel

To start with, the image's RGB channels need to be filtered to show only the green (G) channel with the MATLAB code `im(:,:,2)`, where *im* is the image passed to the main function. This is so that the green parts of the image are represented as brighter pixels compared to the background. This shows as a grayscale-like image which highlights the brightest parts of the image (the plant) so that I can process them further.

I chose to isolate the green colour channel as opposed to converting the image to grayscale as the parts of the image which are green will show up brighter than if I converted the images to grayscale. Isolating

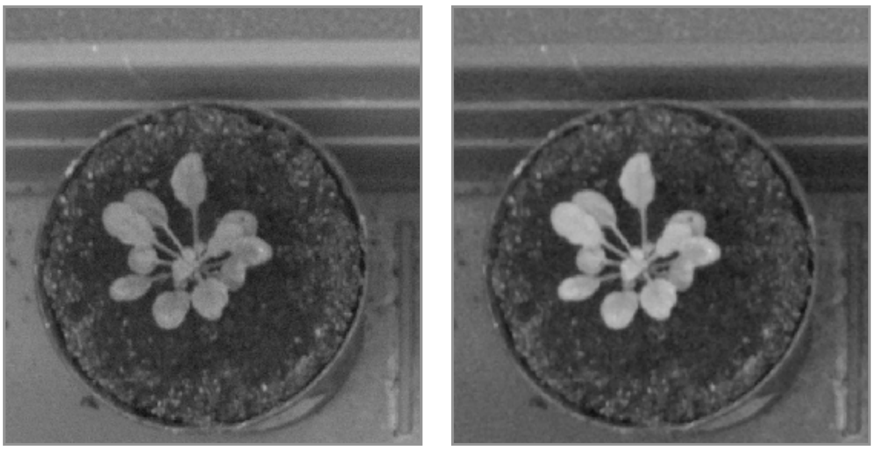

Figure 1: Image converted to grayscale on the left, Image with isolated green channel on the right

the green channel is more specific to the task of finding plant-based parts of the image. For instance, if there was a bright red object in the image, grayscale would highlight that too, whereas green isolation wouldn't.

Figure 1 shows the difference between the grayscale converted image against the green channel isolated image - the green channel isolated image highlights the plant more than the grayscale image, without also brightening the background.

Step 2: Contrast adjustment

After converting the image to a single colour channel, I then increased the contrast of the image by multiplying the image by a constant factor (in this case *2*). This is to reduce the impact of the shadows on any leaves during image binarising. By making the shadows lighter, it helps pick up the leaves with dark shadows on them which would otherwise be left out of the binary image - which is especially useful on the last image.

Figure 2a: Green channel image without contrast adjustment on the left, Green channel image with contrast adjustment on the right

Figure 2b: Binarised image without contrast adjustment on the left, Binarised image with contrast adjustment on the right

As visible in *Figure 2b,* The contrast adjusted binary image picks up slightly less shadow information which helps provide a more complete plant image.

Step 3: Noise removal

After contrast adjustment, I then applied median filtering to reduce the amount of noise in the images. I chose to use median filtering over average filtering as it provides a better result by preserving sharp edges instead of blurring them which the average filter tends to do.

This step is done to reduce the amount of noise in the image, making it slightly easier to process further on.

Step 4: Determining threshold

To find the ideal threshold for each image, I used *Otsu* thresholding with the MATLAB function `graythresh( )` to determine the appropriate threshold for each image between 0 and 1. This is important to determine which pixels should be classified as black and white in the binarising stage.

I also tried using a fixed threshold of 0.3 which provided a clearer plant outline in the binarising stage. I chose to use *Otsu* thresholding as it is able to provide a dynamic threshold based on the image which would be more useful if this was attempted on other images outside the three provided samples.

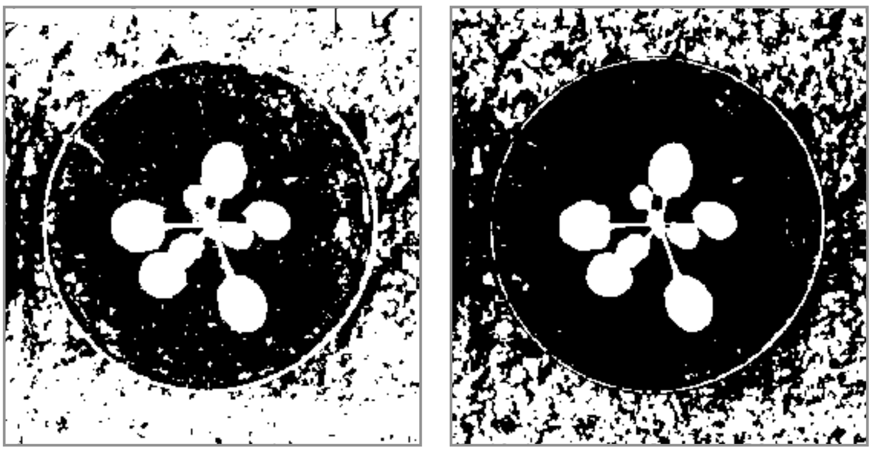

Figure 3: Binarised image with fixed threshold on the left, Binarised image using Otsu thresholding method on the right

Step 5: Binarising the image

This step converts the single colour channel image into a binary image where each pixel is considered either full black or full white. This is especially important for segmentation later on in the process.

I binarised the image using the MATLAB function ***`imbinarize()`*** and passed the image from the previous stages and the *Otsu* determined threshold in as parameters.

This produced the output of an image with black and white parts as determined by the threshold level in order to identify lighter areas - signifying the plant bits of the image.

Step 6: Binary image processing

This step aims to remove artefacts that are part of the background and not the plant.

The first part of this step involves extracting the largest *N* connected objects in the binary image where I have determined the smallest value of *N* that works with the provided image samples is 3.
This helps discard smaller parts of the binary image that are not part of the plant and are not connected.
This was achieved with the MATLAB function ***`bwareafilt()`*** and providing the function with the binary image and the value of *N* as parameters.

The second part of this step involves extracting connected objects of the binary image with circle-like properties only.
This helps to remove some of the larger parts of the background that were picked out in the first part of the step. By filtering out objects based on their *eccentricity*, we can identify vaguely circular connected objects in the binary image as plants.
This was achieved with the MATLAB function ***`bwpropfilt()`*** using the *eccentricity* attribute as the property to filter the objects by.

The results of these steps and this approach amount to the final binary images and are presented in the results section below.

3. Results

These are the binary image results of the three provided images of the plants after processing them through the described steps in the methodology section above. The white sections of the binary image identify the plant in the image.

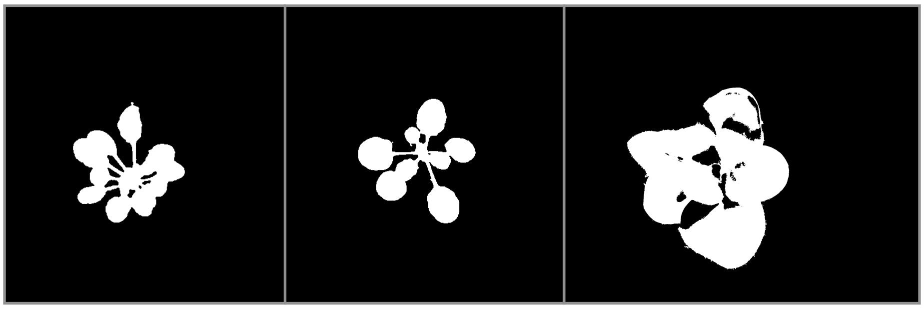

Figure 4a *Figure 4b* *Figure 4c*

4. Evaluation

This section of the report will be addressing & evaluating the strengths and weaknesses of my methodology and approach.

In my approach to solving this problem, I discovered my methodology had several areas of weaknesses, one of which is not being able to fully extract the third plant *(Figure 4c)* in the binary image. Shadows, while reduced, are still present and marked as not part of the plant when it is.

Whilst I have taken several steps, outlined in my methodology, to reduce the impact of the dark shadows on the leaves, my approach does not entirely mark the third plant appropriately, but does mark the outline and most of its contents as being a plant.

An attempt to correct this could relate to filling in the holes of the connected objects in the binary image using the MATLAB function ***`imfill()`***, however, this may cause issues with the other provided images as the parameter needs to work with all three images, and may cause the plant stems of the other images to be missed.

One strength of my solution, is that the background and its contents are almost entirely removed from the binary image. The plant is well highlighted with only small parts of the third plant (*Figure 4c*) being missed. A reason

as to why the plants are relatively well separated from the background is due to the *Otsu* thresholding method used. This method is dynamic for each image as it determines the weighted sum of all the possible threshold values in each image. This would allow for my method to work better outside of the three given sample images.
Another reason as to why the plants are well separated is due to the detection and removal of connected components in the background of each binary image, leaving the plant intact.

Figure 5a: Artefacting in the binary image results around the leaf edges

Another weakness to my approach is that there appears to be some artefacting around the leaf edges (visible in *Figure 5a*), particularly noticeable in the third image. Whilst this isn't a serious issue, it could be improved - one method of doing so might be to perform some further binary image processing to remove non-circular connected objects around the leaves.

Overall, I think that my approach was successful in identifying the plants in the images, and has both its strengths and weaknesses in separating the plant from the background.