Leaf Segmentation

Introduction to Image Processing and Computer Vision - Project 1
Report

Noman Noor

January 2023

Contents

1 Introduction

Techniques of Image Processing are quite useful for fields of Computer Vision and AI. The goal of this project is to perform one such technique, namely segmentation, on the provided dataset.

2 Problem Description

2.1 Dataset Description

The dataset we need to operate on is a part of the PLantVillage Dataset's. More specifically, "Grape___Esca_(Black_Measles)", the part of the PlantVillage Dataset dealing with Grape leaves with Esca (Black Measles) disease.

The relevant section of the dataset has unsegmented, colored photos for 1383 such leaves. For each of these photos, there are also corresponding grayscale, and segmented variants. The colored photos ar. However, we as the input for our segmentation algorithm, and the pre-segmented variants of each of those are compared with our the result of our segmentation algorithm to measure the accuracy of the implemented segmentation algorithm. The current version of the solution does not deal with the grayscale variants at all.

It is to be noted though, that the segmented versions of the leaves in the dataset are sometimes incorrect (to various extents).

2.2 Task Description

We need to find a suitable mask for leaves of Grape vine leaves with Esca (Black Measles) and use that to segment/extract out only the leaf from any given image. The calculation of the accuracy of the results is also to be estimated from existing/ground truth, pre-segmented images.

(a) Input Image (b) Expected output (segmented)

Figure 1: Showing the expected input and output of the segmentation part.

3 Description of the Segmentation Implementation

3.1 The Algorithm

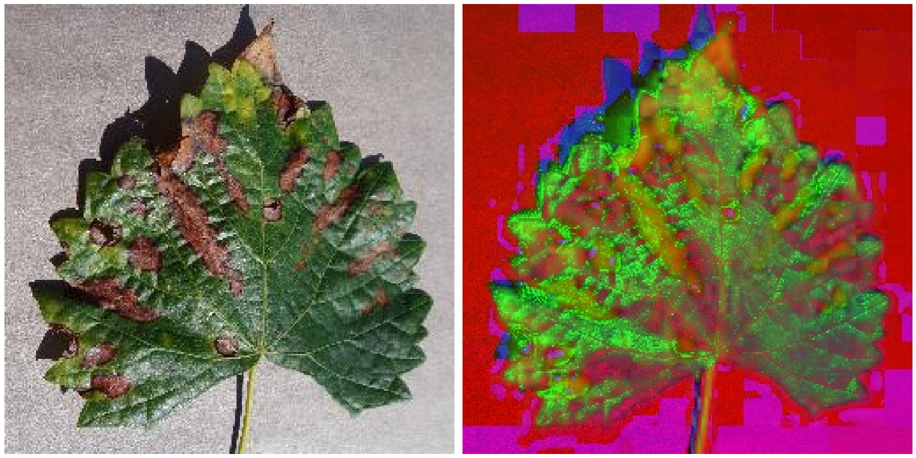
To demonstrate how the solution works, let's look at the following pseudocode of the algorithm.
For each image, we do the following:

Algorithm 1 Leaf Segmentation

```
procedure GET MASK FOR SEGMENTATION
    Input
        image    the input image
    hsv ← HSV representation of image
    mask ← binary mask generated by filtering all elements
        in the range of leaves' healthy/non-diseased parts in hsv
    maskDisease ← binary mask generated by filtering all elements
        in the range of leaves' diseased parts in hsv
    mask ← combination (bitwise or) of mask and maskDisease
    mask ← median blur applied to mask
    conts ← find contours in mask
    mask ← draw contours from conts on mask
    Output
        mask    the final mask returned for segmentation
procedure APPLY MASK TO INPUT IMAGE
    Input
        image    the input image
    mask ← output of Procedure: Get Mask for Segmentation
    segImg ← bitwise and operation on image and mask
    Output
        segImg  Result from segmentation of input image (using mask)
```

3.2 Step-by-step explanation

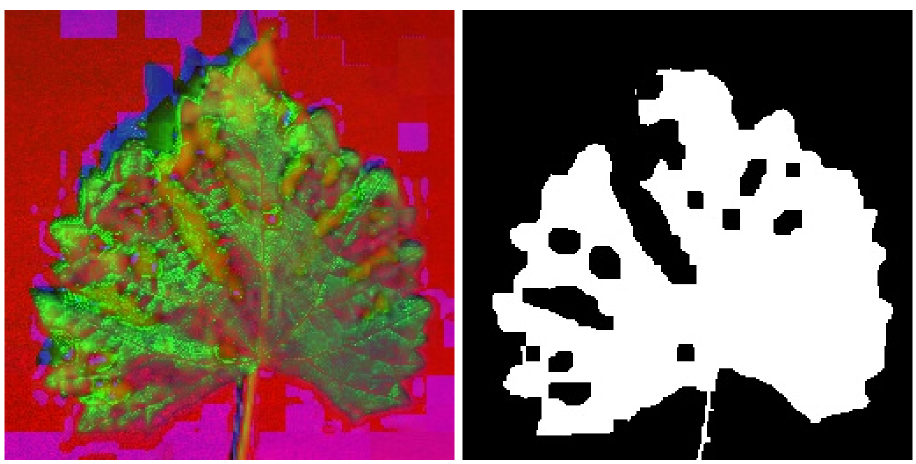
Let's go through how the steps of how algorithm is applied to an example image in order to visualize how the algorithm functions. The approach taken in the accompanied solution is to convert the image to HSV, and then determine filtering which of the Hue, Saturation, and Value channels' can be used to get rid of the unneeded ranges (ranges not part of the expected color ranges of leafs or their diseases).



(a) Input Image (b) HSV representation

Figure 2: Converting input image to HSV colorspace representation.

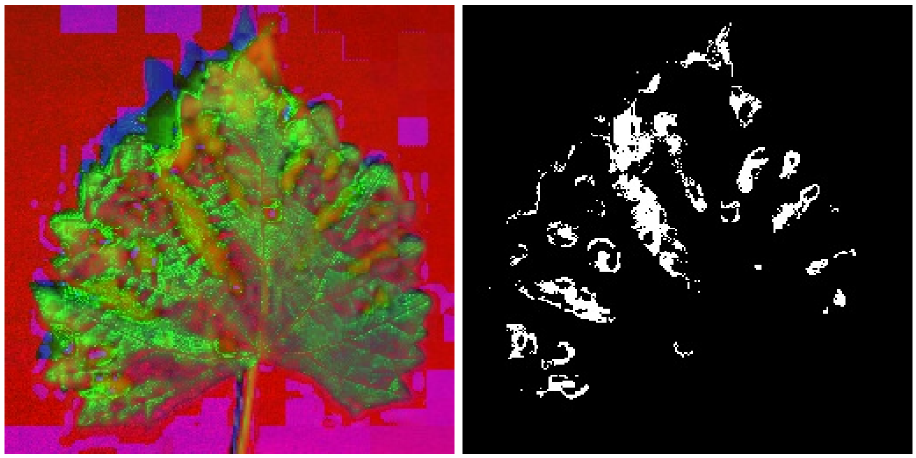
Then, we make a mask from the HSV ranges of the non-diseased part (ranges of green).



(a) HSV representation of Input image (b) Mask for uninfected part.

Figure 3: Getting mask for healthy/green parts of the leaf.

After that, similarly filter to get the mask for only the diseased parts.

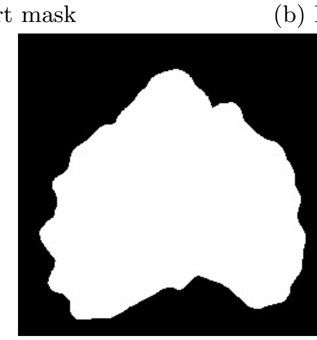


(a) HSV representation of Input image (b) Mask for the diseased part

Figure 4: Getting mask for diseased part.

Then, we combine the masks and apply some post-processing like median blur and contour detection to get the final mask.

(a) Uninfected part mask (b) Diseased part mask

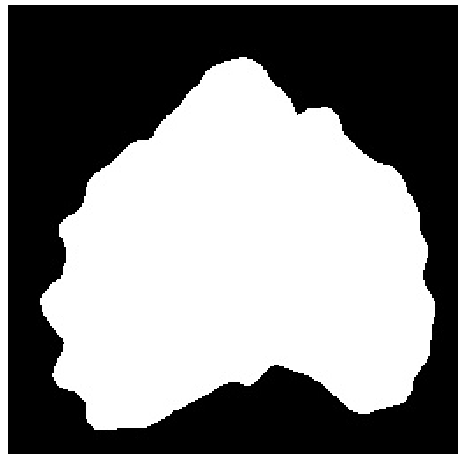


(c) Combination of the masks, with post processing.

Figure 5: Getting final mask

Finally, we use the mask obtained to get a segmented picture of the leaf.

(a) Input image



(b) Mask for the diseased part

Figure 6: Getting mask for diseased part.

Figure 7: Output segmented leaf image

4 Assessment of the Results

The following figures showcase the segmented output images alongside the input image and the expected segmented leaf result, in order to be able to compare these with a human eye.

Figure 8: Input Image

Figure 9: Segmented output

Figure 10: Ground truth: pre-segmented Image

While we could have humans judge these images, it is fairly subjective and dependant on various uncontrollable factors. As such, numeric indicators/metrics are often used to measure the quality of results. As such, we will use metrics such as IoU and Dice coefficient to compare our results to the ground truth (in our case, pre-segmented images). However, one thing to note is that numeric metrics may not always score results in a way that humans would agree with.

4.1 Overall - The big picture

The following tables provide us the overall metrics for the accuracy of our solution.

Table 1: Dice for the whole set

Number of Images	1383
Average Dice	**84.03%**
Median Dice	**84.25%**

Table 2: IoU for the whole set

Number of Images	1383
Average IoU	**78.16%**
Median IoU	**78.52%**

4.2 Examples

4.2.1 Best 5 Results (according to Dice)

Let's take a look at the five best results according to the Dice coefficient.

Fifth Best

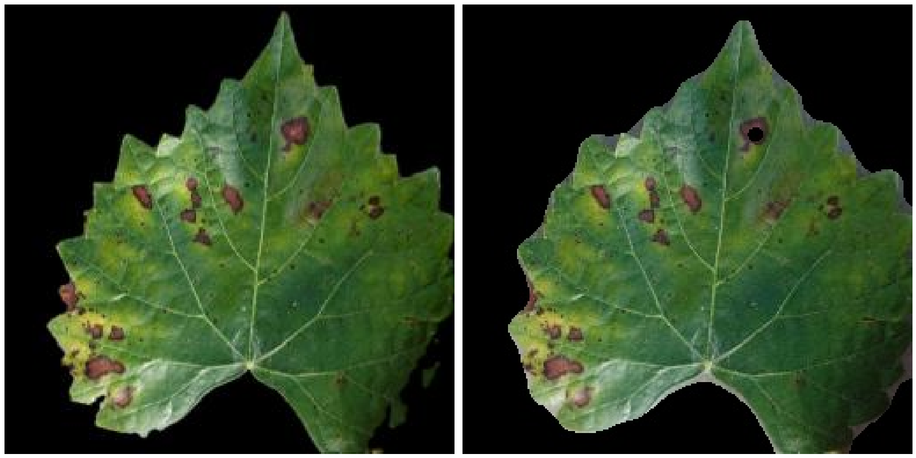
(a) Ground Truth (b) Program Result

Figure 11

IoU	**84.6%**
Dice	**88.97%**

Comments: It can be clearly seen that the images are pretty much identical except some minor lost details at the top tip of the leaf and also inclusion of some of the background at boundaries.

Fourth Best



(a) Ground Truth (b) Program Result

Figure 12

IoU	**84.84%**
Dice	**89.15%**

Comments: Similar to the above image, the truth and result/output are pretty much identical here as well, with the similar exception of some minor lost details at the top tip of the leaf and also inclusion of some of the background at boundaries.

Third Best

(a) Ground Truth

(b) Program Result

Figure 13

IoU	**84.91%**
Dice	**89.23%**

Comments: Once again, the images are quite indentical, but this time the differences are not only the usual background appearing at edges, but also some of the shadows.

Second Best

(a) Ground Truth (b) Program Result

Figure 14

IoU	**85.60%**
Dice	**89.63%**

Comments: Just like the previous image, here, the images are mostly the same, except the shadows show up quite a bit on our results and some of the background at some edges as well.

First Best / The Best

(a) Ground Truth

(b) Program Result

Figure 15

IoU	**87.09%**
Dice	**90.31%**

Comments: These are almost exactly the same except for the loss of the tip of the top of the leaf on the result, and the inclusion of background at some of the edges.

4.2.2 Worst 5 Results (according to Dice)

Fifth Worst

(a) Ground Truth (b) Program Result

Figure 16

IoU	**65.35%**
Dice	**75.64%**

Comments: It can be seen that our segmentation isn't actually quite as bad as it may have seemed from the metrics, but that the Ground Truth itself is a worse segmentation than the result of our program. As such, the metrics for the accuracy of our result ended up being low and misleading.

Fourth Worst

(a) Ground Truth (b) Program Result

Figure 17

IoU	**65.32%**
Dice	**75.17%**

Comments: Here, clearly both the ground truth and the result of the program are bad segmentations. Both of them have holes/missing pixels/information in the leaf. Our output does have more holes than the ground truth. Our output is also missing a part of the tip of the leaf (towards the top). There's also the presence of shadows at certain parts and also the presence of the background near of the edges.

Third Worst

(a) Ground Truth (b) Program Result

Figure 18

IoU	**65.10%**
Dice	**75.07%**

Comments: While our output suffers some of the same problems faced in many of the prior examples (to some extent), the ground truth is clearly worse by a large margin. As such, the metrics for the accuracy of our result ended up being low and misleading.

Second Worst

(a) Ground Truth (b) Program Result

Figure 19

IoU	**53.75%**
Dice	**67.59%**

Comments: While our output suffers some of the same problems faced in many of the prior examples (to some extent), the ground truth is clearly worse by a large margin and hence lead to this result's score being so low. As such, the metrics for the accuracy of our result ended up being low and misleading.

First Worst / The Worst

(a) Ground Truth (b) Program Result

Figure 20

IoU	**52.86%**
Dice	**66.34%**

Comments: Once again, while our output suffers some of the same problems faced in many of the prior examples (to some extent), the ground truth is clearly worse by a large margin. As such, the metrics for the accuracy of our result ended up being low and misleading.

5 Potential Improvements to the Program

One could envision various improvements to the current solution, including the following.

- Adding some type of filling algorithm for filling in any holes left in the leaf masks to fix some of the problems caused by such issues as can be seen in some of the examples above.
- Figuring out better suited HSV ranges for each of the channels, for both the disease part and the non-diseased/green part.
- Decreasing the amount of background that's still visible around the edges somehow.
- Including naturally withered parts of the leaves in the filters (because the color ranges are different for that than either diseased or the healthy parts).

- Try to preserve the "branch" (petiole) connected to the leaves.
- Trying to preserve the top tip of the leaves.

6 Conclusion

Depending on the purpose and required accuracy, the solution could be satisfactorily sufficient. However, there have been some problems that the solution suffers, some of which faced have been described along with the results in the section about analysis of results (more specifically, the subsections for best and worst five results). Regarding these problems, and other issues, there are still some improvements that can be made (some of which can be found in the appropriately titled section, which can be found prior). It is noteworthy that one of the *"issues"* our analysis suffers from is that the analysis of results was held back by the fact that a fair number of the ground truth, pre-segmented images were fairly inaccurate. Yet, it should still suffice to some extent when accompanied by human monitoring/moderation.

Regardless of the issues and problems, or even the accuracy of results, the program achieves it's main goal of segmenting Grape leaves with Esca (Black Measles) disease fairly well.