Computer Vision Project Fall 2022

POISON IVY

Miloni Sangani Rasika Sasturkar Harshil Patel

ABSTRACT

Poison ivy is a small plant that can grow as a vine or small shrub trailing along the ground or climbing on low plants, trees or poles. These are found everywhere in forests, fields, wetlands and along streams, road sides, and even in urban environments, such as, parks and backyards. Usually people fail to recognize them, and run away from all the similar looking plants. Therefore, our goal is to design an algorithm that will isolate and recognize poison, given the image is taken by the human. We perform semi-supervised object recognition on the images to classify poison ivy leaves from the other similar leaves that people usually confuse with. Decision tree classifier is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. So, we use this classification to identify the poison ivy leaves.

INTRODUCTION

At Rochester Institute of Technology, there is a lot of vegetative cover on and around the campus, and right in the heart of it, there's also a forest. There are various kinds of plants and trees, including poisonous ones like "Poison Ivy." Sadly, not that many students and staff members are able to recognize poison ivy correctly. We thus developed a semi-supervised object identification classifier that preprocesses the plant picture clicked by users and determines whether the plant is poison ivy or not in order to swiftly and easily identify poisonous plants. To prepare the data for the classifier, we employed a variety of image processing techniques.

We employed a variety of preprocessing procedures, such as cropping the image to put the leaf in the center and scaling it to a preset resolution. using techniques for contrast enhancement to adjust the image's exposure. As a channel may be utilized to distinguish green color correctly, we later transformed the image to LAB color space. A gaussian filter was also used to denoise the picture. The leaf edges were then detected using a sobel filter, and the resultant picture was then equalized using a histogram. Finally, to separate the leaf from its background during preprocessing, we employed the Mahalanobis classifier. In order to determine if the plant is poison ivy or not, the classifier was given the image at last.

METHODOLOGY

Pre-processing

We have taken the following image to show the results throughout this report:

Each attempt was a trial and error. We tried to change our approach a little and try new techniques and concepts to isolate the poison ivy in the image.

Our Approach:

All the failed attempts in every checkpoint helped us to make some decisions like the color space, cropping the center of the image, using sobel and gaussian filters. We now try to focus more on retaining the shape context,strong edges, modal color and leaf textures/veins.

We decided to work on this algorithm keeping some assumptions in mind. These proved really helpful for our implementation. They include:

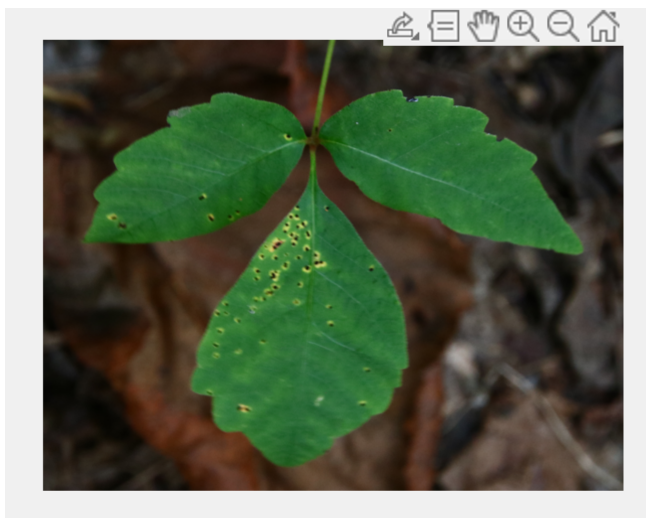
- The image of the leaves is taken by a human.
- The leaves are in the center of the image.
- The outsides of the image are not the plant.

Procedure:

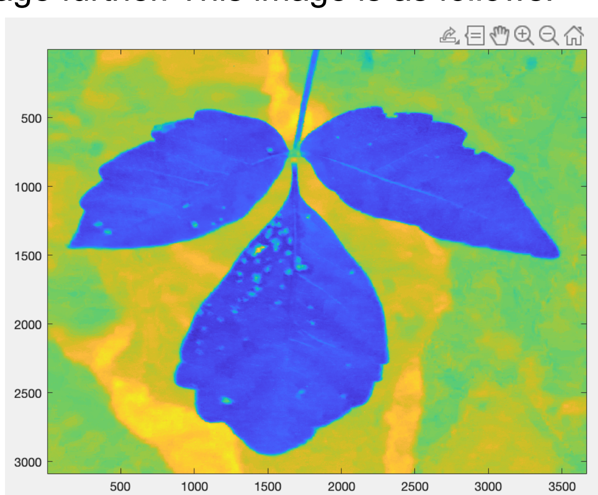
1. We started by converting the image into a double image. This would make it easier to perform operations on the image.

2. Resize the image - the images may be taken with different resolutions of the camera. So, we decided to resize the image to a standard size (in our case, it is resized to [600, 900]). It has given multiple advantages to our image processing. The bigger the image is, the more pixels that there are on target. However, the more pixels you have to process the slower the processing and development goes.

3. We cropped the center of the image, as the poison ivy leaves are always located in the center of the image. This was a good step towards the background noise reduction as after we crop the image, we have less noise to reduce. The resized image after cropping can be seen below:



4. As done in previous checkpoints, we used a- channel of La*b* color space to isolate the image from the background and used the grayscale version to process the image further. This image is as follows:

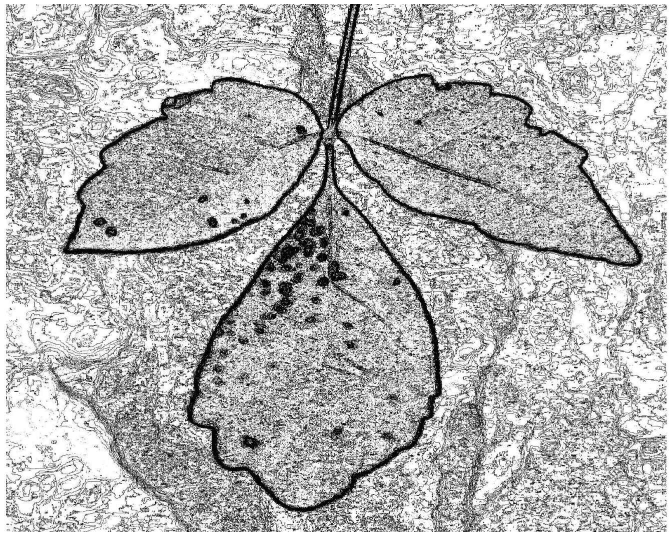


5. There were two types of filters applied to our image:
    a. We applied a Gaussian filter to sharpen the image. This will help us get the pronounced edges/veins of the leaves. It can be seen from the following image:

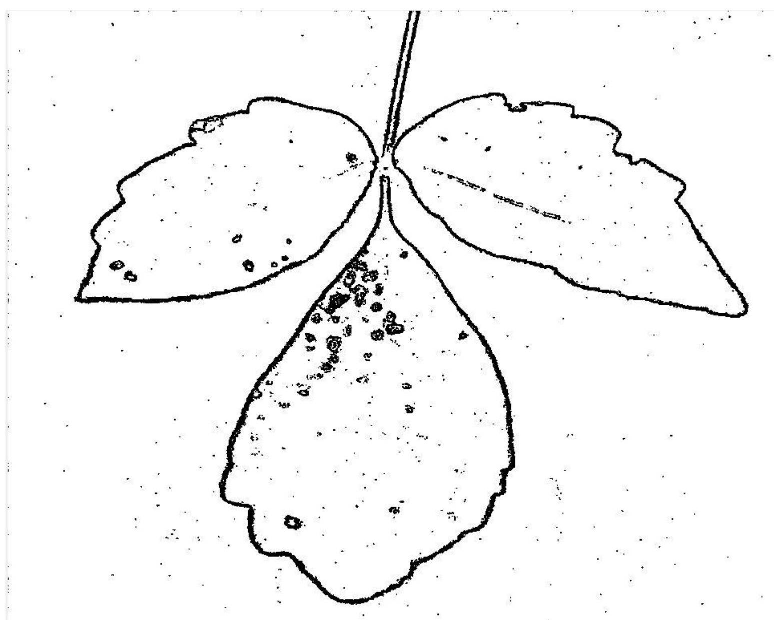
    b. We applied a Sobel filter using the imfilter() function. This gave us the filter magnitudes for x and y directions which will gradually give us the final edge magnitude. We got the cutoff value from the edge strength histogram which was helpful to determine the edge strengths of the background as compared to the edge strength of the center leaves.

6. To determine the modal color of the leaves was a bit challenging for us. We then decided to use Mahalanobis distance to separate the leaf color from the background color. In this way, we got the modal color of the leaves in the center.

7. We performed operations to add these things (edge strengths and modal color of the leaves) which gave us the following image:



8. After performing morphology operations on the previous output image, we were able to extract the final shape of the image. This can be seen in the following image:



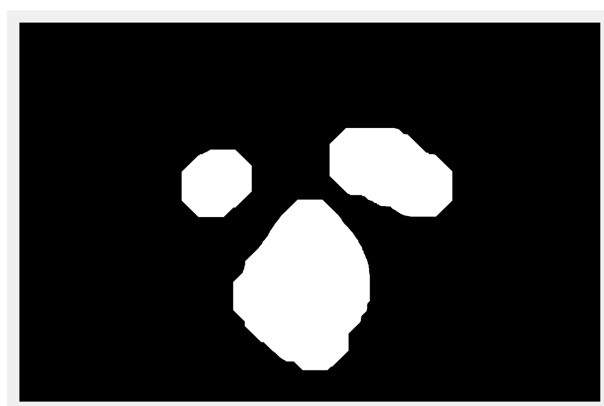
Classifier

We are still working on pre-processing the image and haven't decided which classifier will be utilized in the method. However, we have been considering employing decision tree classification because it is a straightforward classification made based on several aspects of the input image. However, before choosing one of them, we will take into account a number of classifications methods.

OUTPUT RESULTS AND DISCUSSIONS

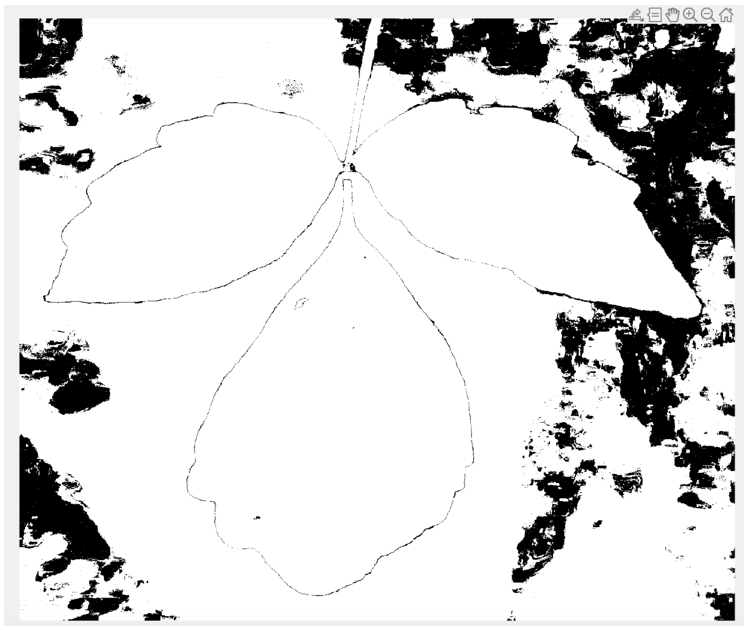
Initial approaches and shortcomings:

In the beginning, our approach was limited by our knowledge of working more with binary images and performing Otsu's method that chooses a threshold that minimizes the intraclass variance of the thresholded black and white pixels. So, we decided to convert the RGB image to a grayscale image and perform imbinarize() on it. The leaves were isolated from the background but the noise was still very prevalent in the result. To fix that issue, we considered performing morphological operations and connected components to reduce the noise. However, this led to a result with just three blobs (leaves). We were not able to retain any information about the leaf shape, edges, veins or the branch connecting the three ivy leaves. This clearly did not work and it helped us realize that binarizing the image was not a good idea. So, in the next attempt we tried working with different color space models like RGB and L*a*b.



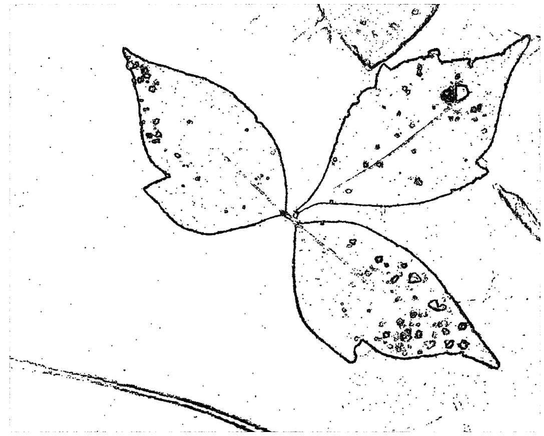
In the next checkpoint, we made an important decision of always cropping the center region of the image before starting the process of isolating the poison ivy from the background. This is essential because we are certain that the poison ivy is the subject of the images provided in the data set and the subject is always positioned in the middle.

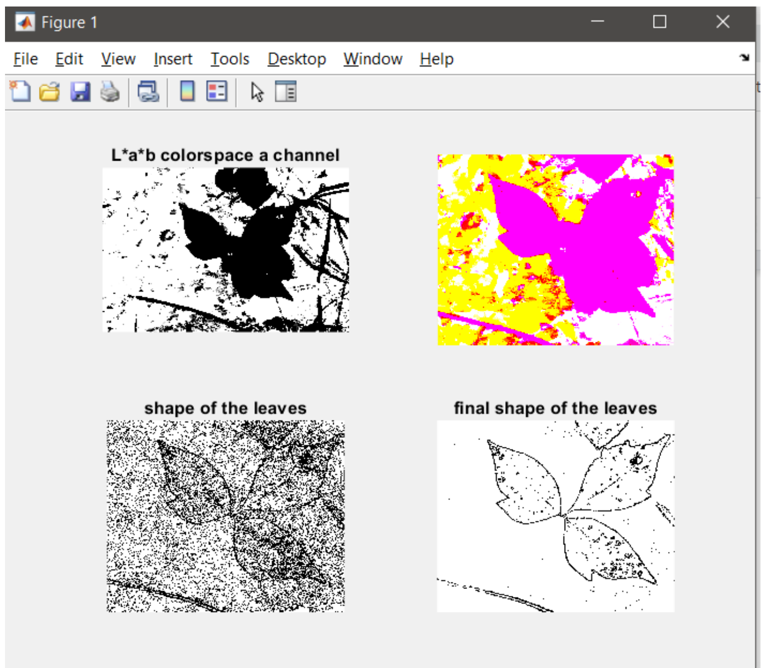
Then, we converted image into L*a*b colorspace and realized that the a- channel were the green pixels which we are interested in. Now, our objective was to separate the green pixels from the non green pixels. On doing some exploratory data analysis and seeing the histogram plot of image we realized that a certain bin range had a high frequency of green pixels. We tried to achieve this using a masking operation. We separated those ranges of pixels from the whole image using a mask where we only consider pixels in bin range -2.5 to 0.5. Applying inversion and setting the pixel to 1 for those pixels we get this image:



Final Results of our implementation till now for some of the images:

We can see here that poison ivy edges have been well detected from the image. The shape of the edges retrieved are a very accurate and true representation of the original shape.
We also managed to retain the leaf texture information, like the dark spots, veins and the branch connecting the three leaves. More importantly, using Mahalanobis classifier to ignore the background non green pixels was pretty helpful.





Images with under-exposure / over-exposure / too many green leaves and pixels in the entire image:

As we can see in the image below, we could not isolate the poison ivy from the background very well as there were too many green pixels in the image. The threshold for a pixel to be a foreground one could be modified a little to fix this issue later when we want to improve the results for such noisy images

CONCLUSION

It could be stated that morphology does not prove useful in object recognition but can be utilized to minimize noise in intermediate results after completing many tests and utilizing various methodologies for preprocessing the picture that best matches for training the model. Although it worked effectively for counting the number of leaves by converting the picture to binary form, even employing linked components cannot categorize leaves since it does not take the texture and shape of the leaf into account.

Differentiating between foreground and background is made significantly easier by choosing the right color space and channel to work on. It appears that our set of photos works best in the a* channel of the L*a*b color space. Moving forward, the leaf sobel filter has shown to be highly beneficial for detecting edges since it allows us to compare edge magnitude and set a threshold that excludes edges with weak signals. We used Mahalanobis distance to separate the leaf from the rest of the backdrop. K mean clustering can be used in place of manually choosing pixels for the background and foreground of the Mahalanobis classifier.

Finally, decision tree classification appears to be the best classification technique to categorize the photos as poison ivy or non poison ivy after preprocessing the images.