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

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

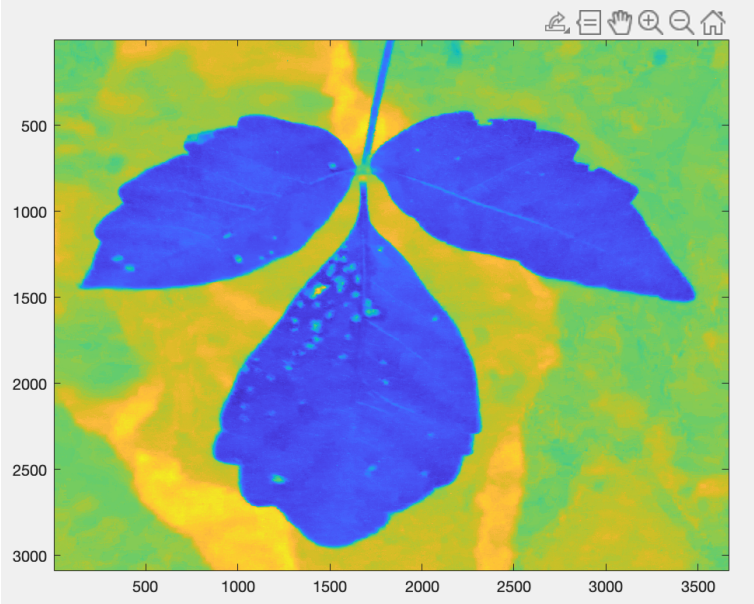
We decided to work on this algorithm keeping some assumptions in mind. These proved really helpful for our implementation. They include:

- 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.

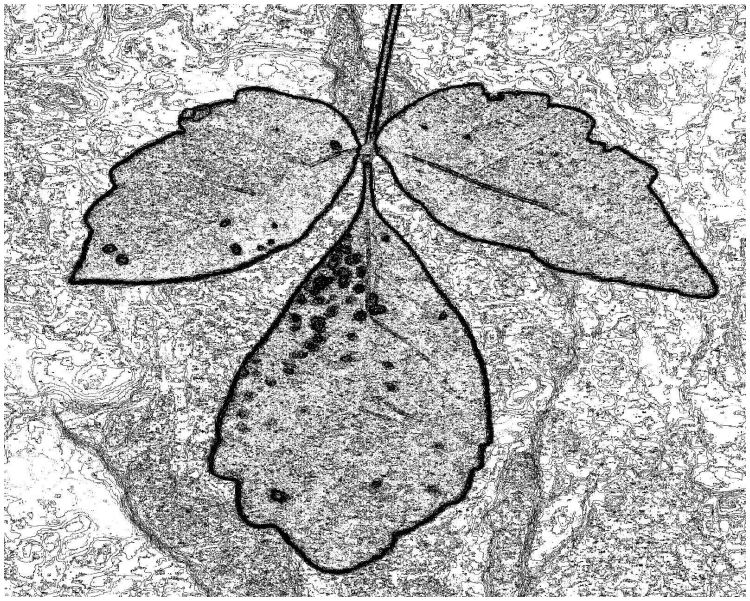
A. Pre-processing

Our images contain center leaves which can be Poison Ivy or some other leaves in the foreground and all types of leaves or branches or soil or other things in the background. We want to isolate this foreground plant from the background which will require some kind of pre-processing on the image. This part encapsulates the noise cleaning part of the imaging chain. We tried to remove the background as much as possible to get a cleaned image. So, we pass this cleaned image to the build classifier program.

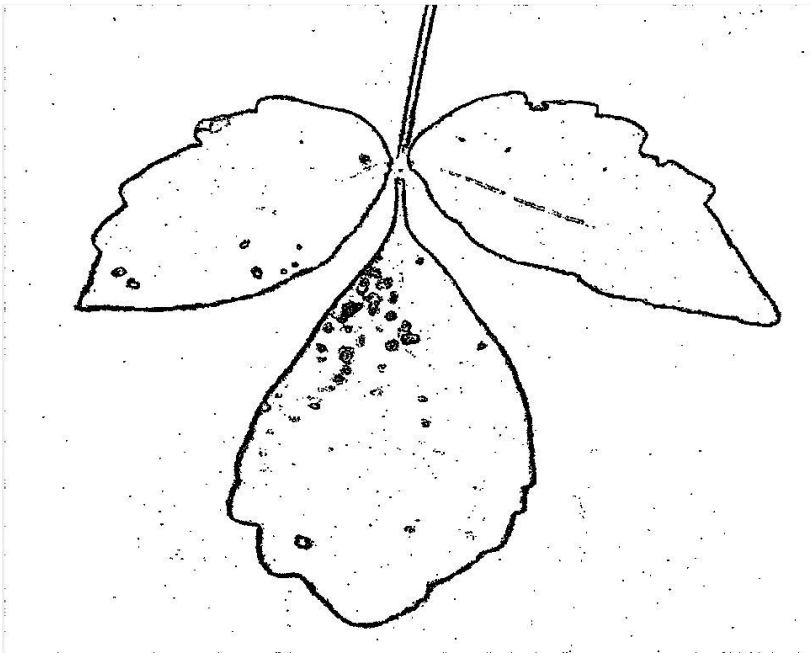
We are given a RGB image, so we start with converting our image into a double image. This would make it easier to perform operations on the image. Now, we know that the center plant is always located in the center of the image. So, we cropped this center. It turns out to be a good step towards the background noise reduction as after we crop the image, we have less noise to reduce. Now, we have converted this image into grayscale image using a- channel of La*b* color space and used this grayscale version to process the image further. The grayscale version of the cropped image can be seen below:



Furthermore, we want to find strong edges near the center leaves. First, we applied a Sobel filter using the imfilter() MATLAB function. This gave us the filter magnitudes for x and y directions which gradually gave 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. To determine the modal color of the leaves, we used Mahalanobis distance to separate the center leaf color from the background color. After getting the edge strength and modal color, we performed operations to add these two which gave us the following image:



The final step of pre-processing was to perform some morphological operations to extract the final shape of the image. We used a strel of disk shape and performed the closing operation using the imclose() MATLAB function. This gave us a cleaned image containing the center plant and most of the background noise removed. One last thing before passing the image to the classifier, we want to convert the image into a binary image. This makes it easier to perform further operations in the classifier program. The final image after complete pre-processing is:



B. Extract features

Feature selection is an important aspect that should be considered before building a decision tree classifier. The decision tree fits the data better if the feature selection process is well executed. We use the regionprops() MATLAB function to return properties and measurements for each 8 connected components in the binary image of the leaf. We pass the cleaned image produced after preprocessing to this function.

Out of the 4 properties we extracted from this function, we have chosen two properties/features which are related to the Area of each connected component identified in the image. It returns the number of pixels covered by each connected component. And the other two properties are lengths of major axis and minor axis. Then, we get rid of all components that are less than a particular threshold scalar value in order to eliminate measurements of the small dirt and noise.

The updated feature matrix contains information about only the features that help us to significantly differentiate between poison ivy and non poison ivy leaves. We also use the Convex Area property of the regionprops method which returns the area of the convex hull over each component in the image. We then used the ratio of each component's area to its convex hull's area. The idea is that this ratio is pretty much the same for all poison ivy leaves and it is different from that of non poison ivy leaves which can again be a good basis of differentiation. We returned this matrix which can be used in the classifier program. The final matrix is:

```
Feature matrix:
   1.0e+06 *

    9.3849    0.0050    0.0044    0.0000
    0.9214    0.0015    0.0008    0.0000
    1.5888    0.0017    0.0013    0.0000
    0.0057    0.0004    0.0000    0.0000
    1.0201    0.0018    0.0008    0.0000
```

C. Classification

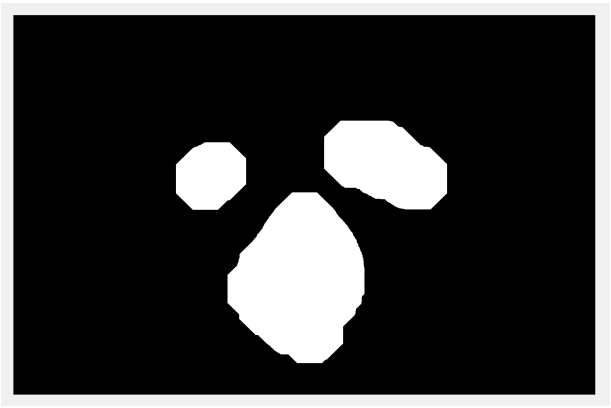
We have achieved the process of Classification using a Decision Tree. It involves two steps: Fitting the decision tree and running the fitted Classifier on the test data set. We have been provided with a collection of both Poison Ivy and Non poison ivy leaves. We kept 80% of the data for fitting the tree and the rest 20% for testing the built classifier. We used the MATLAB 'fitctree' fit binary tree decision for binary classification which receives the feature matrix obtained after feature selection.The pre processed image after noise removal is used to get the features and it uses these features to fit the decision tree and classifies each image as either Poison Ivy (class label = 1) or Non Poison Ivy (class label = 2).

The built decision tree is then used by the Classifier to classify the test images as either poison ivy or non poison ivy. The classifier performs the same pre processing and feature extraction sub routine which was performed before building the decision tee. This ensures that we have a standard format of images received by the classifier. The final step is to predict a label for each image and keep a track of the accuracy for the test data.

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 as seen from following images:

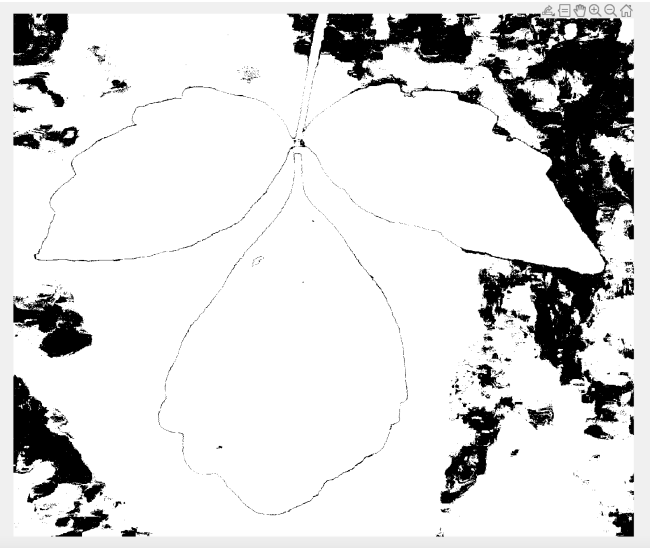


So, in the next attempt we tried working with different color space models like RGB and L*a*b.

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:



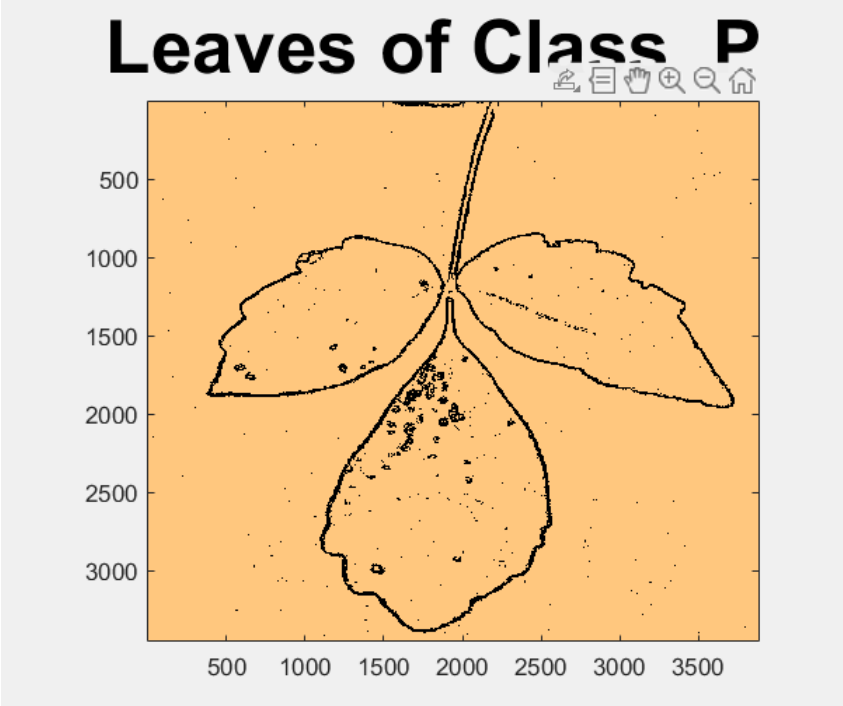
Final results:

We obtained a clean image after pre-processing and extracted features of the same. The displayed results of an image containing Poison Ivy leaves are:

Feature Table after tossing out dirt components

Area	MajorAxisLength	MinorAxisLength	ConvexArea
9.3849e+06	5024.7	4383.7	1.34e+07
9.2143e+05	1517.2	792.19	9.8763e+05
1.5888e+06	1703.6	1279.1	1.8454e+06
5700	391.51	23.393	7672
1.0201e+06	1777.8	770.1	1.1149e+06

The program classifies this into Class P:

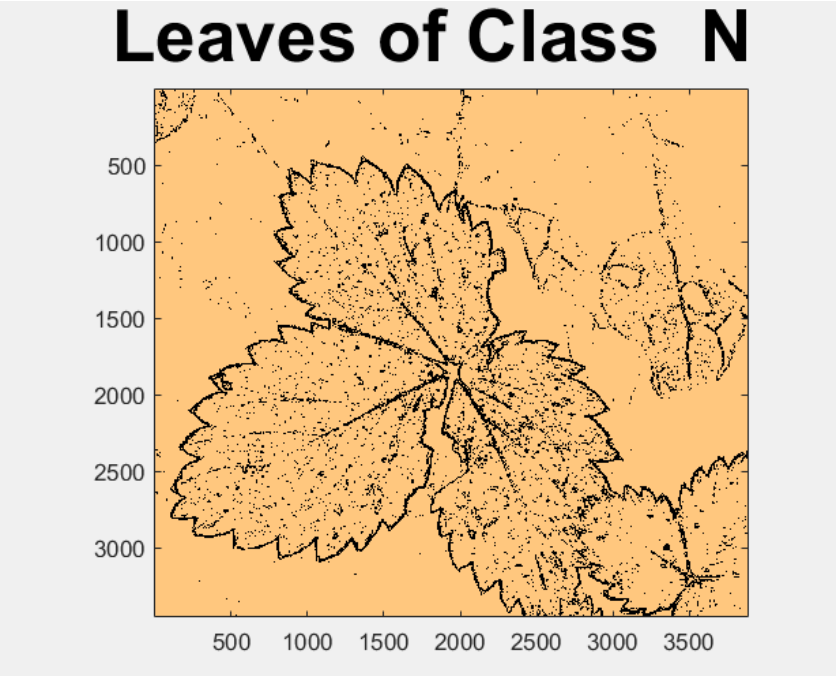


The displayed results of an image containing Non-Poison Ivy leaves are:

Feature Table after tossing out dirt components

Area	MajorAxisLength	MinorAxisLength	ConvexArea
1.2323e+07	4532.8	4020.9	1.34e+07
5335	160.07	50.132	7590

The program classifies this into Class N:



The dataset of images provided to the decision tree is labeled which helps us to calculate the training accuracy as well. A confusion matrix is printed after we predict the labels for each image. This training accuracy is around 91% for the set of images provided.

```
confusion_matrix_on_training_data = [ ...
      76    13 ;
      18   220 ;
];
```

After building the classifier, we loaded this tree in the run_classifier program. It contained all the fitted features for poison ivy and non-poison ivy classes. To test this classifier, we created a validation data of 11 images (includes 6 poison ivy images and 5 non-poison images). We found the following results:

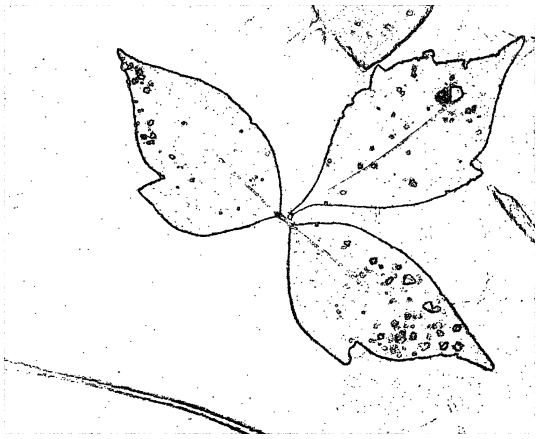
```
Number of Non Poison ivy leaves detected:
     4

Number of Poison ivy leaves detected:
     7
```

From this, we found the accuracy of our classifier on the test data is around 90%.

Additional discussion:

Using LAB color space - We transformed the original RGB image into the LAB color space because we saw that the ‘a’ channel was giving much better results when computing the Mahalanobis distance as compared to the green channel of the RGB color space. The veins, texture and edges of the image were more crisp and clearly visible in the LAB color space.

Preprocessing - We can see here that the center plant 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. Preprocessing for an another image can be seen from the following images:



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. This returned a noisy image without removing any significant background noise.

Resize the image - The images may be taken with different resolutions of the camera. So, we wanted to resize the image to a standard size (like [600,900]). It could have given multiple advantages to our image processing. The processing and development of the process would have been faster. However, we were unable to achieve this as after cropping the resized image, the filters were focussing on weird regions which were not helpful in restoring the shape of the image.

CONCLUSION

It could be stated that morphology does not prove useful in poison ivy isolation 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. Noise reduction through various filters does enhance the image to perform better operations on it.

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 detection using 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. It was found that Mahalanobis classifier detects edges better as compared to hough transform. We used Mahalanobis distance to separate the leaf from the rest of the background. K mean clustering can be used in place of manually choosing pixels for the background and foreground of the Mahalanobis classifier.

Before building a classifier the image has to be changed into a features matrix to be passed to the classifier. In order to select important features, we studied the various properties of the regionprops method like the Major Axis Length, Minor Axis Length, Convex Hull, Area and Extrema among others.

After converting the input image into a feature matrix, classification of the input image was done using a decision tree classifier. Even convolutional neural networks can be used to build a classifier. But as the training and testing data set was relatively smaller in size and it is less complicated to build a decision tree, we used decision tree classification. We split the labeled data set into 4:1 for training and testing the classifier.

The accuracy on the test data set was 90%. We think that a larger data set of the leaves would have helped us to make better evaluations of our decision tree model and selected features. The chances of overfitting and underfitting would be less with more data.

We learnt that cleaning the image thoroughly before building a decision tree and using appropriate features is far more important in the process of building the decision tree.
Throughout the project, we made sure to use the concepts learned in class to isolate the poison ivy leaves from the background. We tried thresholding, morphology, k-means segmentation to cluster the leaves and background, hough transform for leaf edge detection, Mahalanobis distance estimation and edge strength calculations for finding shape and color context. We really enjoyed the process of playing around with different techniques and eliminating possibilities when they did not seem fit for achieving our goal.

We are really proud of the fact that we were able to isolate the leaves very well from the background using the techniques we learnt in class and perform classification on the same. We sincerely appreciate the help and guidance from Dr Kinsman and Dr Niu. The demonstration code and insights provided were very helpful and really helped us understand the project in its entirety.