Project # 3: Object Detection Part 1: Image Features

Instructor: V. Paúl Pauca

Jianqiu Xu (Tony)

1. Introduction

In this project, the goal is to experiment on different ways of detecting features from certain objects in given images and observe the differences in these methods. Since there are five small datasets to choose from, I focused on the palm tree datasets. The objective is for the algorithm to output “Palm” or “Not Palm” for a given image.

2. Methodology

2.1 Datasets

In the palm tree datasets, we have some images of canopies among which palm trees can be identified at various scales. For example three images are closeups showing palm trees clearly. A fourth image taken at higher altitude shows much more canopy with palm trees spread out but still identifiable. A fifth image shows the a region of the rainforest at much higher altitude but with high resolution. Palm trees are still identifiable.

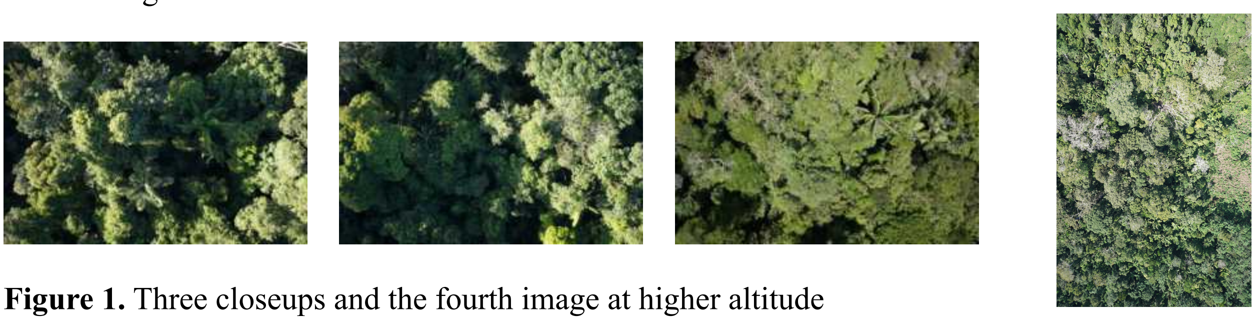


Figure 1. Three closeups and the fourth image at higher altitude

I randomly selected 10 pictures of palms and 10 pictures of non-palms. 5 of the 10 palm pictures are captured from the closeups and the other 5 are from the image taken at higher altitude (which means less resolution). The same is applied for 10 non-palm pictures.

Figure 2. Five closeup palm images (labeled palm_h_1 to 5):

Figure 3. Five higher altitude palm images (labeled palm_l_1 to 5):

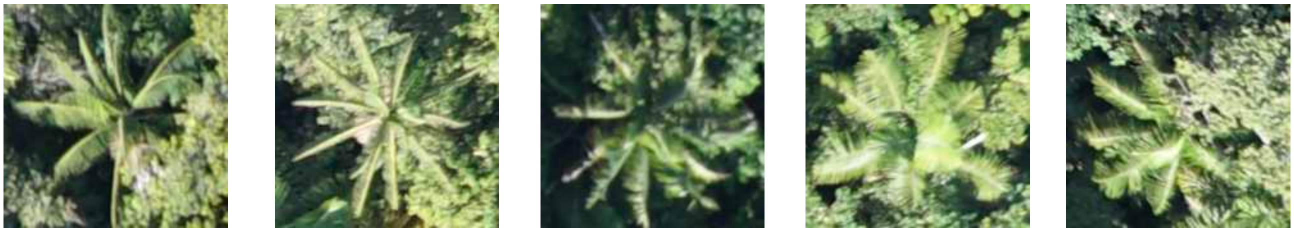


Figure 4. Five closeup non palm images (labeled non_palm_h_1 to 5):

Figure 5. Five higher altitude non palm images (labeled non_palm_l_1 to 5):

After deciding the datasets for training and testing, I explored two popular feature descriptors: HOG and LPB.

2.2 Histogram of Oriented Gradients (HOG)[1]

Histogram of Oriented Gradients is a feature descriptor used to do object detection. When we implement the HOG descriptor algorithm, we first divide the image into small connected regions called cells, and for each cell compute a histogram of gradient directions or edge orientations for the pixels within the cell. Then, we discretize each cell into angular bins according to the gradient orientation. Each cell's pixel contributes weighted gradient to its

[1] Source: https://software.intel.com/en-us/ipp-dev-reference-histogram-of-oriented-gradients-hog-descriptor

corresponding angular bin. Groups of adjacent cells are considered as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms. We take these blocks and contrast normalizes their overall responses before passing to next stage. Normalization introduces better invariance to illumination, shadowing, and edge contrast. Finally, normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

The computation of the HOG descriptor requires parameters including masks to compute derivatives and gradients, geometry of splitting an image into cells and grouping cells into a block, block overlapping, and normalization parameter.

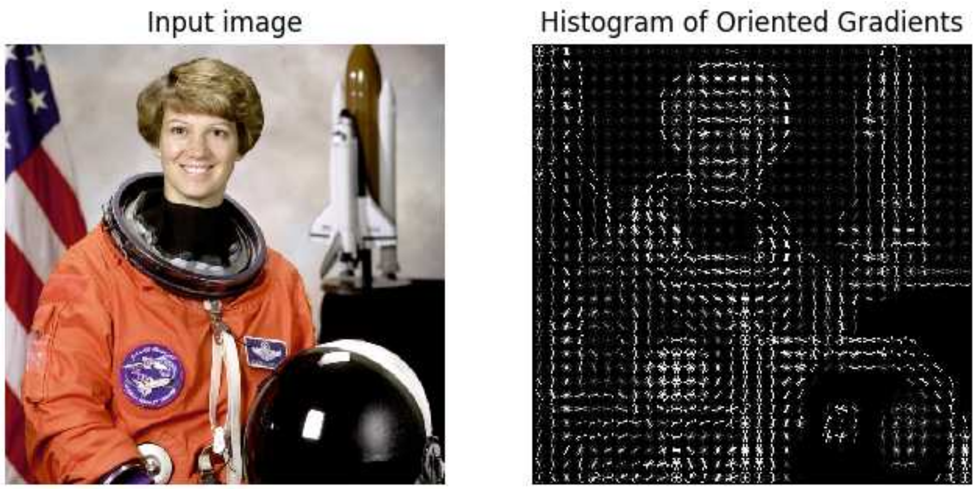


Figure 6. Example of input image and Histogram of Oriented Gradients

2.3 Local Binary Pattern for texture classification (LBP)[2]

Local Binary Pattern is a texture classifier which looks at points surrounding a central point and tests whether the surrounding points are greater than or less than the central point, which appear as binary results.

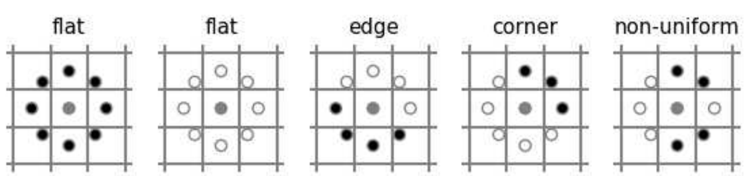


Figure 7. Local Binary Pattern

[2] Source: http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_local_binary_pattern.html

The figure shows some example results with black representing pixels that are less intense than the central pixel and vice versa. When using LBP to detect texture, it computes an array containing the local binary pattern for each pixel. The array describes how much of “flat", “edge", and “corner" pattern there is in a defined radius around each pixel. We measure a collection of LBPs over an image patch and look at the distribution of these LBPs.

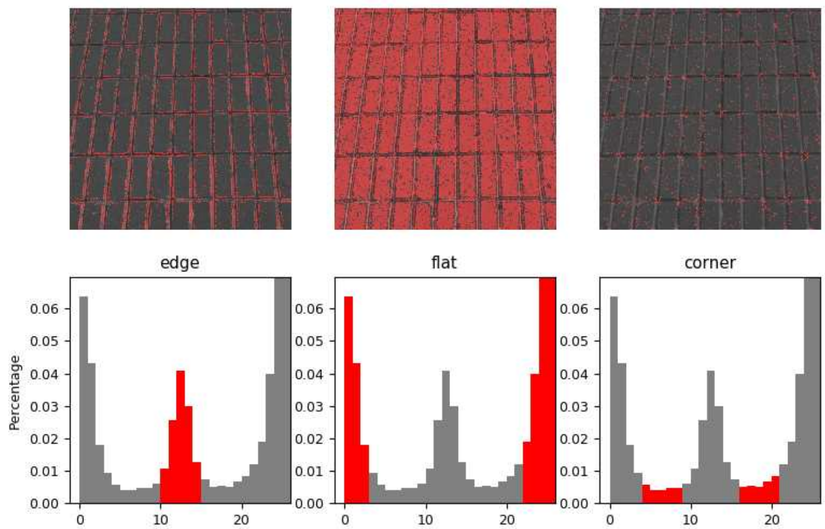


Figure 8. Flat, edge-like, and corner-like regions of the image

3. Experimental Results

3.1 Histogram of Oriented Gradients (HOG)

I implemented HOG on a total of 8 picture samples with 4 palm images and 4 non-palm images (2 low resolution & 2 high resolution for both groups).

Figure 9. HOG of Palm Tree (palm_h_1 and palm_h_2)

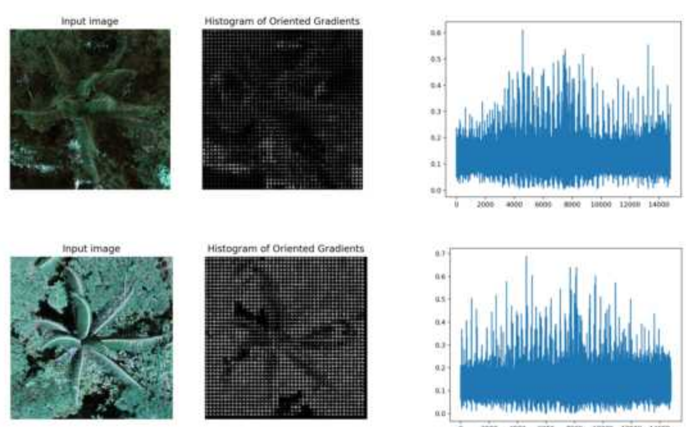


Figure 10. HOG of Palm Tree (palm_l_1 and palm_l_2)

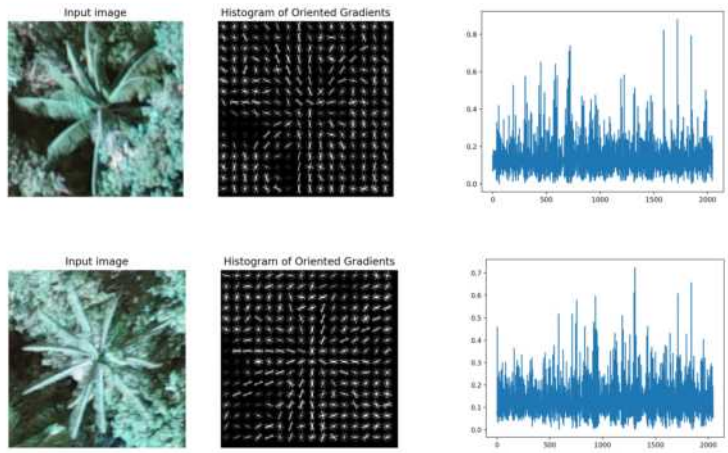


Figure 11. HOG of Non Palm Tree (non_palm_h_1 and non_palm_h_2)

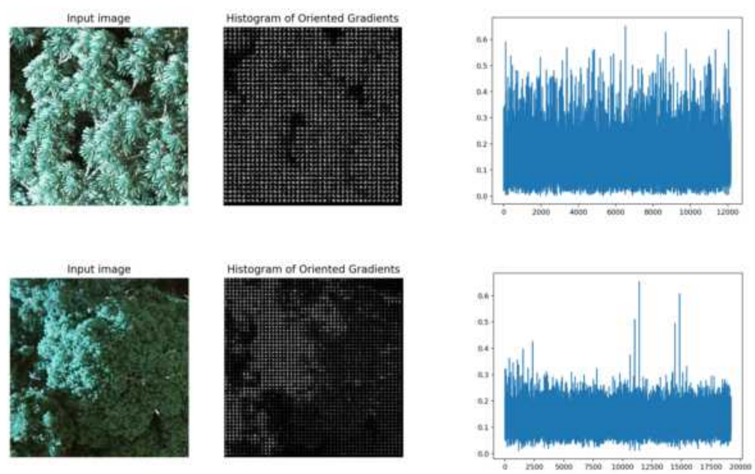
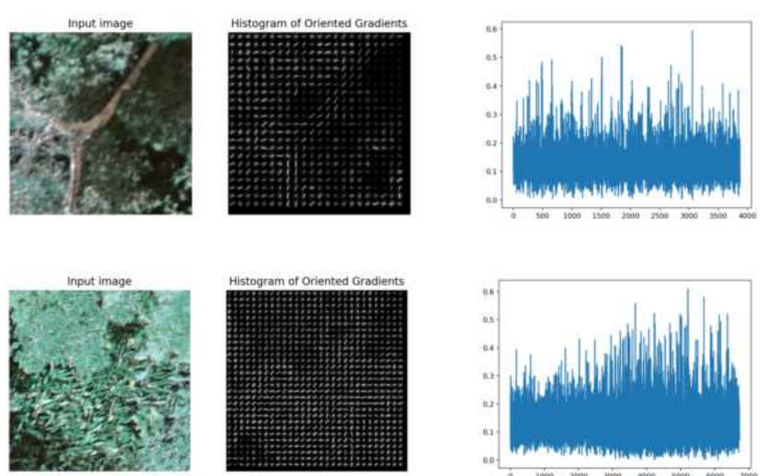


Figure 12. HOG of Non Palm Tree (non_palm_h_1 and non_palm_h_2)

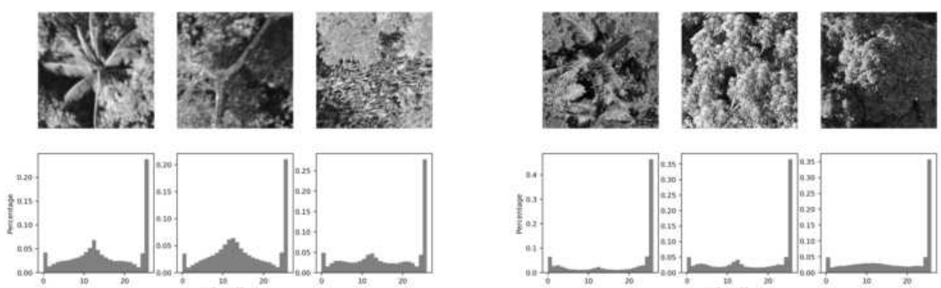


The Histogram of Oriented Gradients computes centered horizontal and vertical gradients' orientation and magnitudes. From the HOG graphs above, we can see that gradients for palm tree images all have similar pattern that horizontal and vertical gradients emit from a center point and spread into several directions, which show the palm leaves that different the palm tree from the background. For the non-palm images, they show flatter and more uniform patterns in their horizontal and vertical gradients with little noticeable edges or central point.

The graphs on the right plots the gradient magnitude (y-axis) and orientation values (x-axis). The palm tree images generally has higher maximum gradient magnitude values between 0.7 to 0.8, whereas the maximum gradient magnitude values for non palm images are at around 0.5 to 0.6. The gradient magnitude values on different orientation values are similar to each other and more evenly distributed for images without palm trees, since non-palm images usually have uniformed gradient on each blocks and have no clear direction information. The images including the palm have clear gradient towards same and different directions, and each blocks various from one another, thus the gradient magnitude for different blocks are various. The graphs also indicate that palm tree images have a relatively high magnitude of gradient for orientation in a certain range than the rest of the orientation values.

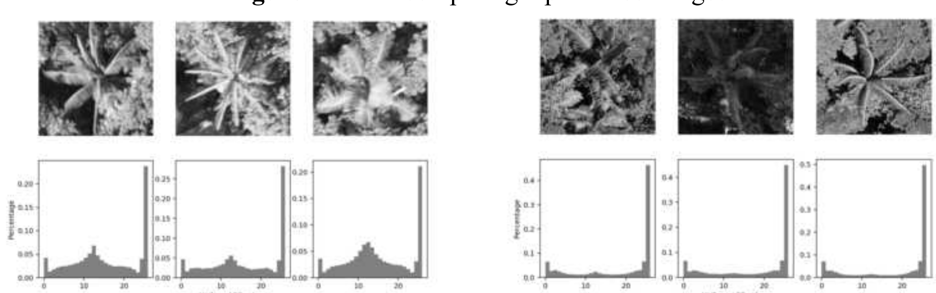
3.2 Local Binary Pattern for texture classification (LBP)

Figure 13. LBP for 1 palm image against 2 non-palm images



I first compared the local binary pattern of 1 palm tree samples with 2 non-palm tree samples (for both high and low resolution images). We can clearly see the difference of uniform LBP values between palm images and non palm images. However, it is still unclear whether palm trees all have a similar LBP patterns.

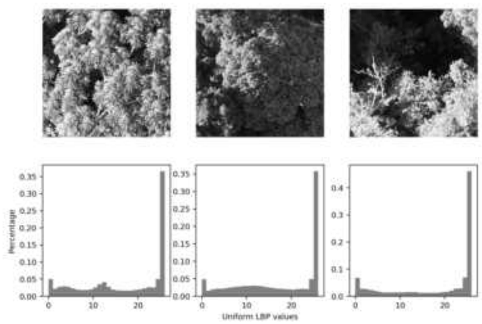
Figure 14. LBP comparing 3 palm tree images



Very nice experiments. There is still a lot of forest in the palm samples though.

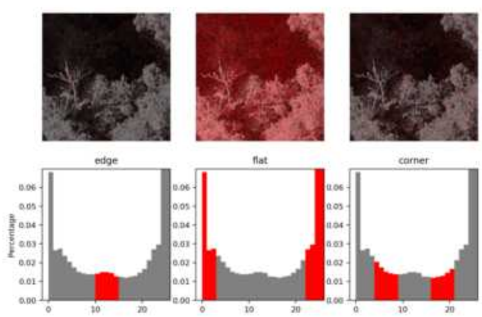
The graphs above compare the random palm tree samples in the datasets and show the LBP patterns. The palm trees all show similar uniform LBP values shown in the figure 14 above. I also compared non-palm images and observed the LBP patterns for non-palm regions. In the figure 15, I compared 3 non-palm images and they all have a unique LBP patterns, which implies that LBP classifier might be a very useful tool to achieve our objective.

Figure 15. LBP comparing 3 non-palm tree images

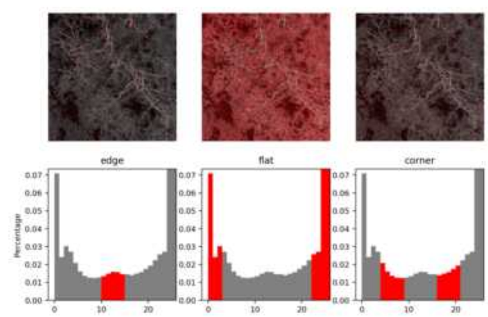


Nevertheless, by just looking at the LBP patterns of each images, it is difficult to get reliable conclusion since we cannot distinguish different parts of the LBP values and understand what each part stands for. After adjusting the code, I was able to highlight flat, edge-like, and corner-like regions of the images.

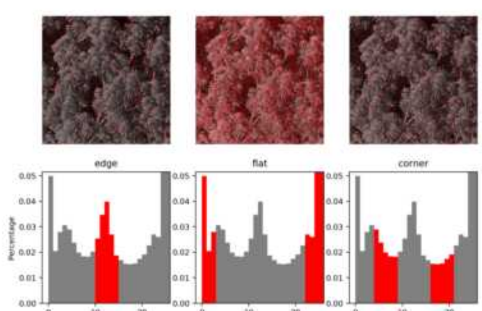
Figure 16. LBP highlights for non-palm images



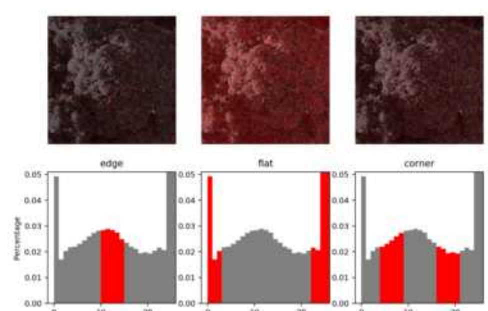
a.



b.



c.

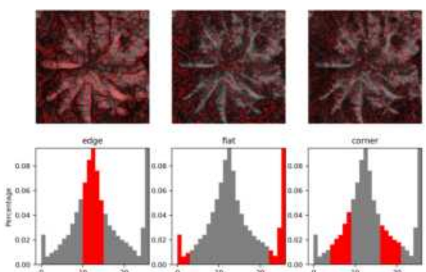


d.

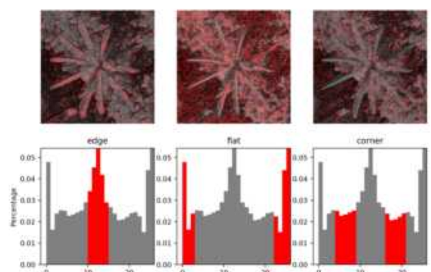
From the highlighted patterns, we can see that most non-palm images have similar percentage in edge-like and corner-like regions, since they generally have few clear edges and corners (or too many edges and corners), which is different from palm tree patterns. While non-palm images have lower percentage in corner-like and edge-like regions, some palm tree showed a high percentage in the edge-like region and a relatively lower percentage in the corner-region. This texture pattern is different from what we observed from non-palm images.

Nonetheless, not all palm images follow the pattern observed above, taking d, e, f in Figure 17 for instance. This might be because the palm tree images have included too many non palm areas so that we cannot get an accurate solution. The pattern of some non-palm image, such as Figure 16 (c), is similar to the palm tree pattern. So, in order to get more accurate and reliable solution of the texture LBP patterns, I decided to divide the current image data into even smaller patches so that non-palm texture in the palm tree images can be eliminated.

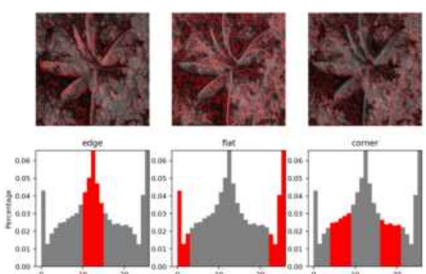
Figure 17. LBP highlights for palm images



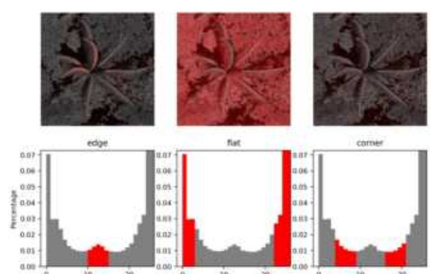
a.



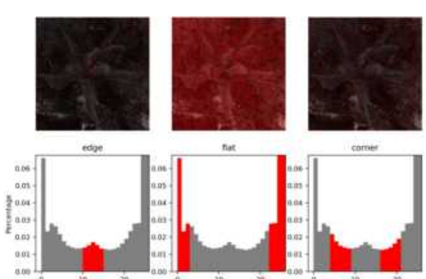
b.



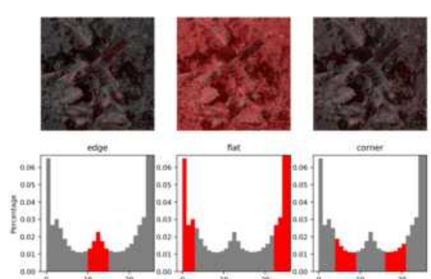
c.



d.



e.



f.

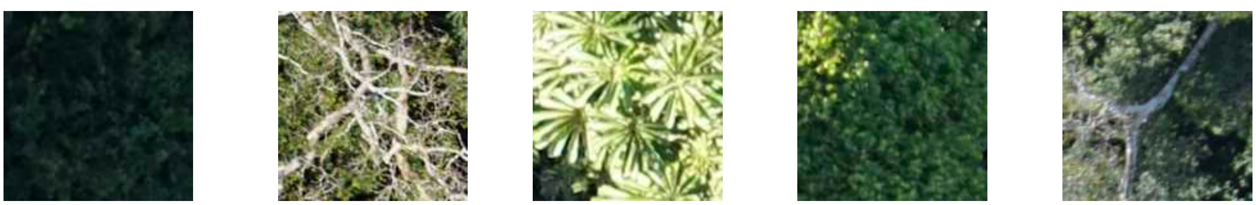
3.4 Local Binary Pattern for Smaller Patches

After comparing the LBP patterns of the palm tree images and the non-palm tree images, we are quite certain that palm trees have a unique texture that is different from its surrounding environment. However, the difference on the LBP values are not that significant when we are using the whole palm tree images, since the LBP result would be effect by the tree leaves surrounding the palm trees. To solve this issue and magnify the difference in texture, I decided to cut the original palm images further into smaller patches that only contains the palm leaves and observe the texture patterns.

Figure 18. Smaller patches contains palm leaves only (p_p1 to 5)

I also cut some smaller patches for other textures that is not from palm trees to do comparison.

Figure 19. Smaller patches contains non palm texture only (np_p1 to 5)



I implemented the same LBP texture classifier to the smaller patches and get the LBP patterns with possible "edge", "flat", "corner" been highlighted. The texture of palm leaves is quite recognizable. The LBP pattern shows a large percentage of LBP values are distributed in the "edge" region and "flat" region, while the LBP values of "corner" are relatively low and even. However, the LBP pattern for non palm texture generally has high percentage in both "edge" and "corner" region. Since they do not have clear stripes in their patterns, which leads to a high percentage of LBP values for corner-like regions. The difference is not so different when it comes to patch *np_p3*, since this type of plants also has a similar shape with the palm tree, and I have to improve the performance of the classification by finding the characteristic of this specific plant in the next stage of our project.

Great work in this project, Tony. Your experiments look promising. I'm curious about what you will get in classification. Looking into performance, say using AUC, as a function of window size might be good.

Figure 20. LBP on smaller patches contains palm leaves only (p_p1 to 5)

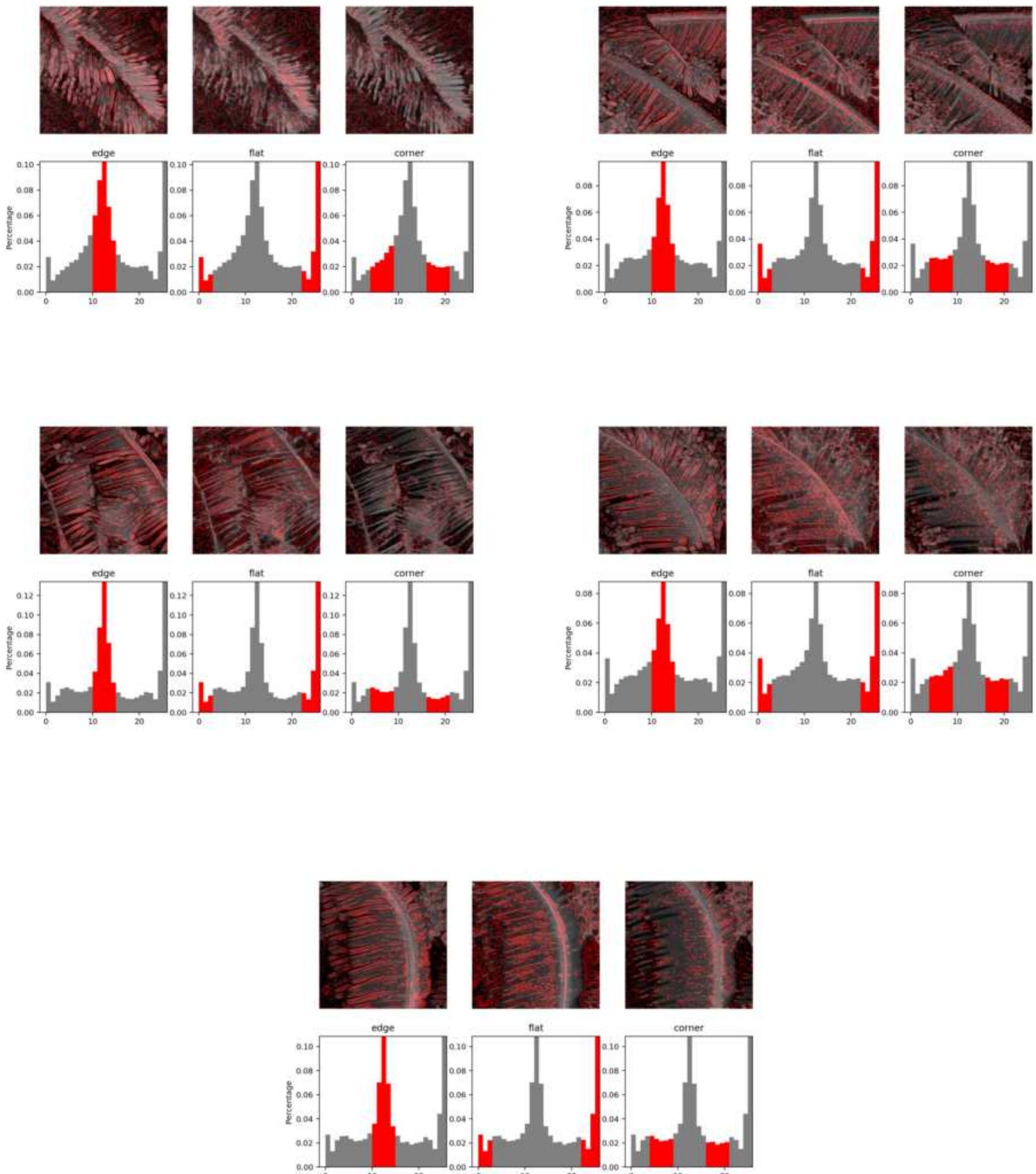


Figure 21. LBP on smaller patches contains non palm texture only (p_p1 to 5)