Sebastian Zdarowski
DSC 481 Digital Image Processing
Final Report
November 21, 2019

Does Color Impact Classification when using a Convolution Network

Abstract:

This paper studies convolution neural networks ("CNN") and the affect that leaf color has on a CNN. There will be 5 different classes for the CNN to predict using a subset of the Leafsnap dataset. Around 270 images will be segmented using automatic mean thresholding to create binary images, and a mask/filter will be applied for color images to separate green leaf pixels from the background. Ranges for these color values will be determined by building a 3D grid for the highest quality image in the leaf class. It has been found that the CNN doesn't respond well to color images when given the task of classification as opposed to the binary image classification. It appears that by adding 3 color channels to the image, the CNN misclassifies similar images that share the same color range. Leaf images that contain a unique color range, have a higher percent chance of being properly classified when compared to the classes that share similar color ranges.

Introduction:

Plants are present within our daily lives, but do we really know what kind of plants they are just by looking at them? Sure, you are able to identify popular common plants, but plants come in a wide array of species, types, shapes and sizes. The goal of the project is to be able to determine if adding color to the leaf classification problem could prove beneficial. Leaf classification is a popular topic amongst computer vison scientists. If we were able to identify plant species based on the images alone, it would allow a person without a background in botany to determine which species is what. From the papers that I will discuss in the background, when other researches were running their research they seem to exclude the color of the leaves from their analysis even though many plants come in different shades of green. Could it prove to be beneficial to a Neural Network ("NN") when it comes to classification? Would the additional information allow the NN to classify these leaves with a higher accuracy rating? Within this project, I will be using the Leafsnap dataset to provide clean lab images to work on. When considering this project, other datasets were considered but they only provided pre segmented images, which already had half of the work completed. From the Leafsnap dataset, I will take 5 different classes of leaves for a total of 270 images.

Background:

The Paper "*Leafsnap: A Computer Vision System for Automatic Plant Species Identification"[1]* was one of the primary inspirations for this project. The paper primarily focused on creating a mobile application that has the ability to identify different plant species using automatic visual recognition. The system called Leafsnap is a computer vison system and was used to identify different tree species from photographs of their leaves. The dataset itself contained over 184 different tree species native to the northeastern part of North America. The methods used within this paper are classifying, segmenting, extracting and comparing the features within the dataset. The researchers employed 4 different methods on classifying leaves based on their images. The researchers created a system that was able to identify leaves based on a mobile picture, which converted into a mobile application. The researchers didn't post accuracy ratings, so it's hard to determine if the experiment was fully successful.

In the paper "*Deep – Plant Identification with Convolutional Neural Network*"[2] the researchers proposed a NN; specifically a convolutional NN to be able to identify and classify leaves based on their images and the features associated with each image. They used this approach to avoid the black box problem that is commonly associated with neural networks. A visualization technique was also applied based on the deconvolutional networks ("DN") to make sense of all the features that the convolutional neural network ("CNN") was using. In the first part of the paper the researchers propose the CNN model to automatically learn the features representation for plant categories. The second part of the paper focused on the DN to visualize which feature seemed the most important. Shape and venation were the most important features to consider when classifying leaves. This project will focus primarily on shape due to this paper, as the CNN was able to obtain an accuracy rating of 99.6 % based on shape alone.

Segmentation will also be a big part of this project. The paper that worked primarily on binary images was *"Leaf Shape Identification Based Plant Biometrics"[3].* The goal of this paper was to use leaf shapes, to identify different plants, using binary images that are segmented from their backgrounds to visualize what shape the leaf was and how to apply a proper classification for an image with the shape alone. The difference in this paper is that it requires user input to map the base points and reference points of each image. Based on those points, a leaf can be extracted from its background to provide the leaf shape. Binary image process was done manually within this paper. Users mapped out anchor points for the leaves and performed a successful binary segmentation. Within this project, mean thresholding will be applied on the grayscale leaf images to properly segment the image into binary.

Color is another important aspect of this experiment, and the paper that dealt with that was *"Leaf Classification Using Shape, Color, and Texture Features"[4].* The paper primarily focused on leaf classification using shape color and texture features to identify different plant species. Most methods to identify plant species with color are widely ignored in leaf classification, due to color not being an important part of identification. The paper sets a point out to see if by adding color, would it be able to increase the performance of a classifier. The difference between this project and mine, is that the researches used a Probabilistic NN instead of a convolution NN. In addition, feature extraction was performed on the dataset, which allowed the researches to use not only color, but shape, venation and texture of different leaves. The final NN that was built contained an input layer, a pattern layer, a summation layer and finally an output layer to provide the results of the experiment. Without the features providing information, the PFT model got a performance rating of 74.69% accuracy. By adding in a combination of shape, color, vein and texture, the model was able to obtain a 93.75% accuracy rating. The experiment ran by these researchers shows that color could be an important part of classification when it comes to identifying different plant leaves.

Methods:

Data: The data was obtained from LeafSnap website, which hosts over 10,000 images of different tree species. They provided segmented and unsegmented images. Due to this being an image processing class, the non segmented images were worked on. 5 different tree species were identified to be worked on. Leaf images shown below with count. In total there is around 270 images to be worked on for this project.


(A)

(B)

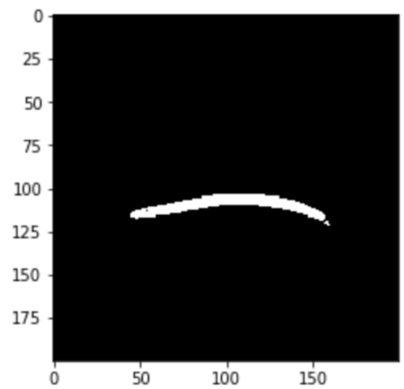
(C)

(D)

(E)

Abies_Concolor(A)(51 images) , Acer_Ginnala(B)(31 Images), Acer_Palmatum(C)(92 Images), Acer_Rubrum(D)(45 Images) and Asimina_Triloba(E)(49 Images)

Binary Segmentation: For this project binary segmentation was done in order to get a binary image for the leaves within the dataset. Before that could be done, grayscale had to be applied to the images using a package within python labeled CV2.RGB2Gray(), so that the background could be easily identified from the leaf images. In theory, the white background should have a higher pixel value than the leaf images themselves. As well as the background containing mostly white, when applying mean thresholding to the image, the white pixels should be counted as the main value within the image due to their high



value and how prominent they are within the image itself and all the other pixels should be less than the mean. This is a form of automatic thresholding, the idea is to separate the image into two parts, background and the foreground. Within Python we are able to do this by multiplying the 2 axis of the images and obtaining the mean of the image. The formula to the right shows a simpler version of the python code that was used to generate the mean thresholding algorithm.

$$J(x,y) = \begin{cases} 0 & \text{if } I(x,y) < T \\ 1 & \text{otherwise.} \end{cases}$$

Figure 1

Once the algorithm was completed, it was set into a loop and for each class of leaves the mean was computed and with that the background was segmented from the image itself.

Color Segmentation: In order to obtain the leaf images excluding the white background, color segmentation had to be performed on the dataset. The technique applied here was more of a manual calculation of the HSV ("Hue, Saturation, Value") space of the class of leaves. First, the image had to be processed using another package from CV2.RGB2HSV in order to transform the colored images from the RGB space to the HSV space. Once that was completed, to map out the approximate values that each class of leaf will fall into, a 3D grid was created in order to visualize how the HSV values fall within the leaf image. This was done by using matplot lib library within python. Within Figure 2 on we can see that the "green" hue values are falling between 40 and 90, saturation is falling between 105 to 255, and the

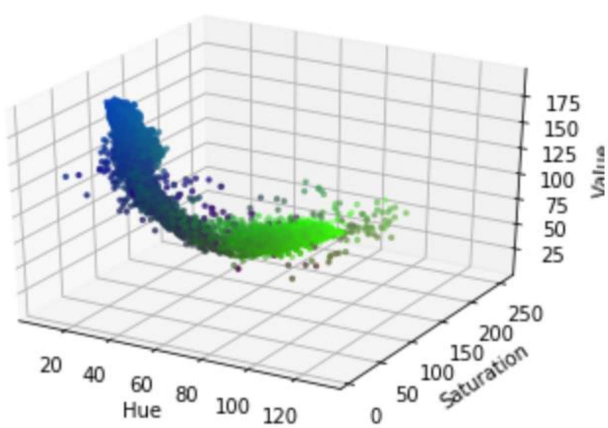


Figure 3

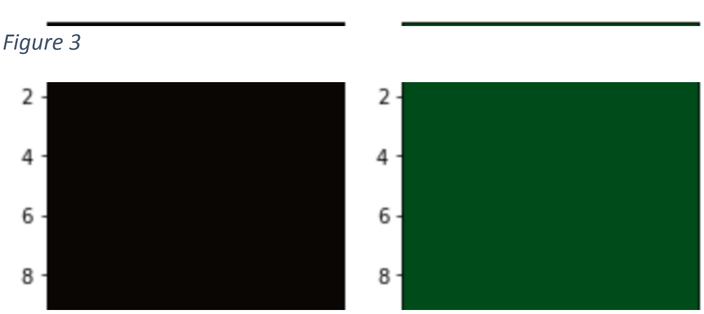


Figure 2

Values falling between 0 to 55. These plots were created for the 5 different classes, to capture the different shades of green present within each class of leaf. In order to pick out the best leaf to use, it was decided that the leaf had to have the clearest background of each set to find the values to use for the thresholding. Once the values were obtained for each individual class, a lower range and upper range were created (Figure 3). After the ranges are configured for each class, a mask would be created to apply on each individual image within each class as seen in figure 4. The image to the left would be considered the mask and the image to the right would be the color segmented image.

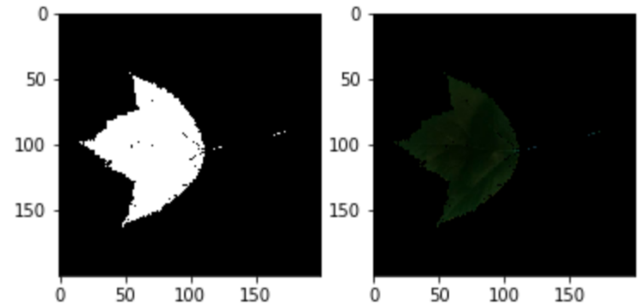


Figure 4

Convolution Network: A convolution Network will act as the classifier for this project. Convolution Networks are a popular computer vison neural network model that is used to classify images. The NN does this by looking at a small patch of the image("pixels") like a 5x5 window will usually look at 25 pixels to learn what the values are within those pixels and then move on to the next patch. The moving on to the next patch is called the stride. For example, a stride of 2, means that our 5x5 window would move over 2 pixels at a time until the filter created 5x5 spans that go through the entire image. Within the project for the Binary Classifier, there are 2 present convolution filters within the classifier; one that contains 32 convolutions by a window size of 9x9 and the second that contains 64, with a window size of 9x9. The window sizes are larger for this Convolutional NN, because the images are 200x200. If the window sizes were smaller, the network would take a longer time to train. Color CNN will be using one Convolution filter with a size of (32,9x9), due to the color model not responding well to additional convolution layers. Keras will be primarily used as the package to run the NN. When training and testing the NN, the dataset itself would be split into 70% testing and 30% training.

<u>Results:</u>

Binary Segmentation: Most of the images were properly segmented using the automatic mean thresholding technique described within the methods section. As seen in Figure 5, here are results regarding the binary segmentation for each of the 5 classes. Most of the leaf image was captured within the segmentation technique. As shown, the background was identified as the highest value within the images, so it turned to black, while the leaves had a lower value than the mean so they turned white.

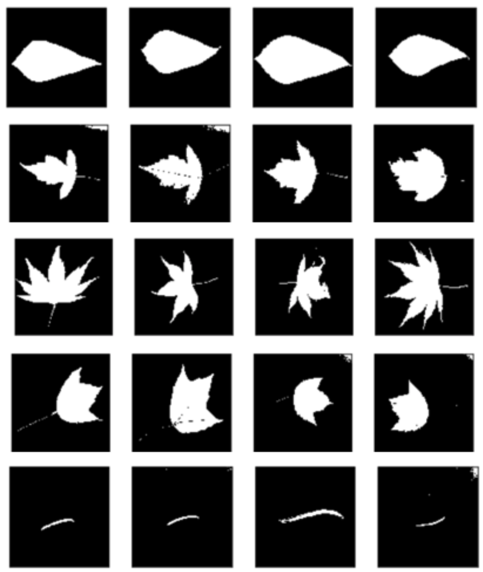


Figure 5

There are some noticeable white pixels in the corner of a few of the images, but the entire leaf was captured so it would be an acceptable segmentation. However, not all of the leaves were perfectly segmented within the dataset as there were a few instances of the leaf merging with the background. This issue occurred with the darker images within the dataset itself. All the brightly colored images were segmented without any issue, but for some of the darker colored images, parts of the leaf identified as the background. Besides a few of these problem images, the rest turned out well. The class that was affected the worst by this was the Acer_Palmatum, as the images were taken with darker lighting. The best class turned out to be Acer_Rubrum, because the images taken of these leaves seemed to be taken with the appropriate lighting.

Figure 6

Color Segmentation: Segmentation for color turned out fairly well with the HSV gird chart estimation method. As you can see figure 7, all classes were segmented by the different shades of green present within the leaf image. Classes Asimina_Triloba and Acer_Ginnala responded best to the transformation. The HSV grid charts for these classes were very helpful as these leaves tended to have brighter values which helped in the charting out of the HSV values for these leaves. Acer_Palmatum and Abies_Concolor had some purple pixels mixed

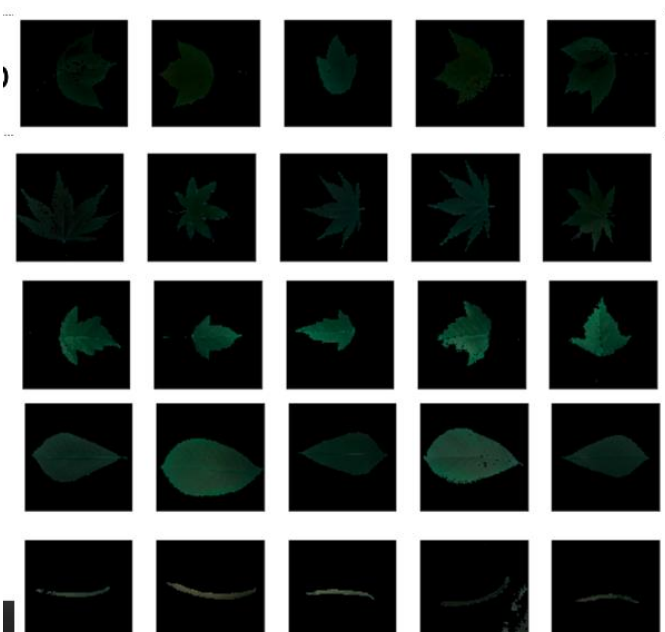


Figure 7

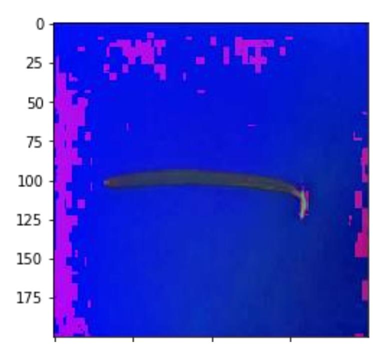


Figure 8

in within the leaf ranges. Therefore, it was harder to capture the leaf entirely as the background was transformed into blue/purple pixels. This was because the leaves themselves contained some of similarly colored pixels within the image. As seen in figure 7, some of the pixels were not captured for the class Acer_Palmatum because there were black pixels located within the leaf itself. Same with the Abies_Concolor class as there was noise detected on some of the corners of the images.

Convolution Network: As stated in the methods section, the convolution network was used within this project as

	Binary Classification		Color Classification	
	Training	Testing	Training	Testing
Model 1	96.65%	77.52%	17.32%	19.10%
Model 2	99.44%	85.39%	18.99%	19.10%
Model 3	-	-	19.01%	19.11%
Model 4	-	-	30.17%	29.21%
Model 5	-	-	32.96%	34.83%

Figure 9

the classifier, to see if the NN, could pick up on the image shape itself and identify it from different leaf species. A total of 7 CNNs were created for this project to classify with the highest accuracy. Within the table you will see the Runs of each model for the training accuracy and the testing accuracy for the dataset. Binary Classification only required 2 models to obtain a favorable classification rate. The model building process was done by first building out a simple model that only contained 1 Conv parameter (32,3x3 window size). The model consisted of 7 layers, with the total amount of parameters being 20,073,733. It obtained a 77.52% accuracy rating when it came to the testing set, so another model was built to achieve a higher rating. The second model parameters were listed in figure 10. By adding a second Conv filter to the model, the NN was able to predict the classes with an acceptable classification rate. The color convolution network is where the classification was not successful. After running serval models, the best score possible on the NN turned out to be 34.83%. It seems that by adding the 3 RGB channels within the NN, prevented the NN from being able to differentiate between the different colors. Instead it focused on the similar color ranges between the leaves which would attribute to the misclassification of these different leaf species. Figure 11 will show the full parameters of this model. At first it was thought that having a similar network that worked for binary segmentation would work for the color classification, but the model didn't perform as well with

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_13 (Conv2D)           (None, 192, 192, 32)      2624
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 184, 184, 64)      165952
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 92, 92, 64)        0
_________________________________________________________________
dropout_11 (Dropout)         (None, 92, 92, 64)        0
_________________________________________________________________
flatten_7 (Flatten)          (None, 541696)            0
_________________________________________________________________
dense_11 (Dense)             (None, 128)               69337216
_________________________________________________________________
dropout_12 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_12 (Dense)             (None, 5)                 645
=================================================================
Total params: 69,506,437
Trainable params: 69,506,437
Non-trainable params: 0
_________________________________________________________________
```

Figure 10

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_29 (Conv2D)           (None, 198, 198, 32)      896
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 99, 99, 32)        0
_________________________________________________________________
dropout_33 (Dropout)         (None, 99, 99, 32)        0
_________________________________________________________________
flatten_16 (Flatten)         (None, 313632)            0
_________________________________________________________________
dense_29 (Dense)             (None, 64)                20072512
_________________________________________________________________
dropout_34 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_30 (Dense)             (None, 5)                 325
=================================================================
Total params: 20,073,733
Trainable params: 20,073,733
Non-trainable params: 0
_________________________________________________________________
```

Figure 11

Species (Class Accuracy)	Binary	Color
Ginalla	90.00%	12.7%
Palmatum	90.63%	0.0%
Rubrum	60.00%	100.0%
Abies	86.67%	12.6%
Tribola	100.00%	46.3%

Figure 12

additional conv filters. Parameters were tweaked, filter sizes changed from 3x3 through 9x9, but it made little impact on the classification. Since the complicated models were not working as well for the color classification, a simpler model proposed in figure 11 was used and that was the best model score for color. Within figure 12, the results are present for each species of leaf in the best possible model for Binary and Color segmentation. The classes Palmatum, Ginalla, Abies and Tribola performed very well within the binary segmentation model. It seemed the class Rubrum didn't perform as well, and when looking at the prediction results, the model chose to place 40% of Rubrum class within the Palmatum class. This would make sense as the shapes of these plants are similar, perhaps if there were more examples of the rubrum class, prediction rates would be higher. On the color model, it seemed that all Palmatum images were classified as Rubrum images. Within figure 7, these classes were the first two rows, Rubrum being the first row and Palmatum being the second. Within the images, you can see that these leaves are sharing a similar color range which could explain why the NN model decided to classify these images as the same.

Conclusion:

Does color impact classification? It seems that it does. But not in the way the project was initially proposed. With the extra pixel information and additional color channels the assumption was made that color could perhaps help a Convolution NN classify leaf images. Running multiple NN models and tweaking parameters for the color model didn't seem to improve the overall classification rate for those leaves. In the future if the project would be attempted again, I think that by adding venation features and extracting curvature features with the images provided in the project, classification would be more specific. Feature extraction as noted in the background research papers, still proves to be an important feature that must be considered when classifying different leaf species. Another thing to consider was the size of the dataset, if there was more time to process 10,000 images for the color segmentation, it could possibly make a difference. The task itself seems pretty monumental considering the HSV grid range technique applied in the methods section. Color doesn't seem to be important to a NN, in fact it seems to hinder it when it comes to leaf classification.

Reference:

[1]Kumar, Neeraj, et al. “Leafsnap: A Computer Vision System for Automatic Plant Species Identification.” *Computer Vision – ECCV 2012 Lecture Notes in Computer Science*, 2012, pp. 502–516., doi:10.1007/978-3-642-33709-3_36

[2]Lee, Sue Han, et al. “Deep-Plant: Plant Identification with Convolutional Neural Networks.” *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, doi:10.1109/icip.2015.7350839.

[3]Hossain, Javed, and M. Ashraful Amin. “Leaf Shape Identification Based Plant Biometrics.” *2010 13th International Conference on Computer and Information Technology (ICCIT)*, 2010, doi:10.1109/iccitechn.2010.5723901.

[4]Kadir, Abdul, et al. “Leaf Classification Using Shape, Color, and Texture Features.” *Leaf Classification Using Shape, Color,and Texture Features*, July 2011, pp. 225–230.