CNN Image Classification of Corn Leaf Diseases

August 2021

Matt Haberkorn
Phoenix College Bioscience Dept.
matt.haberkorn@phoenixcollege.edu
matt.a.haberkorn@gmail.com

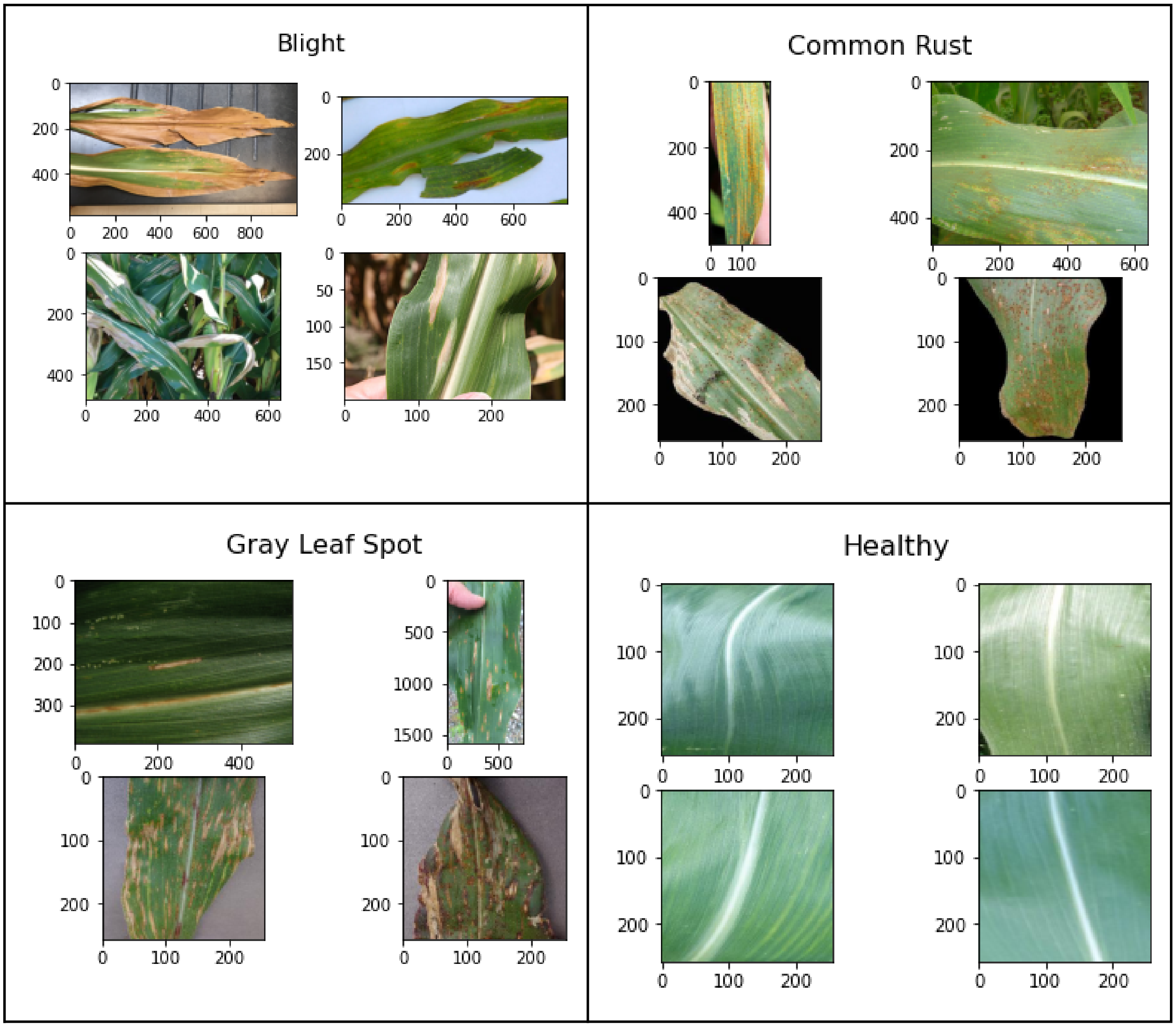
Introduction

Corn is a globalized crop that produces more grain by weight annually than any other crop worldwide. Its high importance can be found in both developed and developing nations globally and is essential not only as an agricultural commodity but also in its numerous industrial applications. As the world population and economy has grown, corn has continued to play an essential role in the feeding of billions and contributing to economic growth essential to supporting the population. One of the challenges to increasing corn productivity to support a growing population and economy is the identification and control of leaf diseases. Correct and early identification of corn leaf diseases is becoming increasingly important to the assurance of improved crop yields and prevention of disease spread. In some leaf disease outbreaks, overall yield can be reduced by up to 50% and early identification of these diseases could prevent disease spread and result in increased yield. The purpose of this project is to produce a Convoluted Neural Network (CNN) that can ID photos of diseased plant leaves to aid farmers in proper identification of disease so the best course of crop management can be determined, thus increasing yield and preventing further spread of the disease.

Data

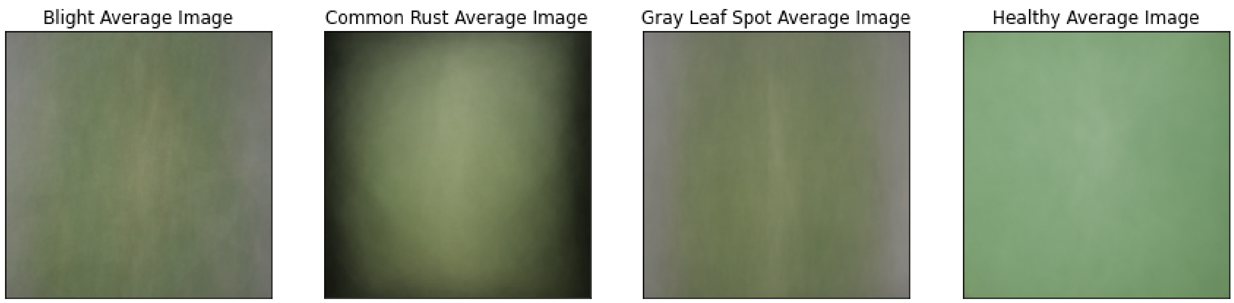
Data for this project was found on the Kaggle “Corn and Maize Leaf Disease” project website (https://www.kaggle.com/smaranjitghose/corn-or-maize-leaf-disease-dataset). The dataset contained a total of 4,188 images. Of these images 1,306 were common rust, 574 gray leaf spot, 1,146 blight and 1,162 were healthy images. A standard method of taking photographs was not used for this image library and images contain a variety of backgrounds to no backgrounds, differing proportions of the leaves to background in the images, differing image shape and size, differing number of leaves, and differing levels of infections. Images from the different categories had different methods of photographing and certain categories had consistent backgrounds, or lack of backgrounds, that were absent in other image categories. This property of the image library may prove to be a problem in modeling, and may possibly result in models classifying backgrounds rather than leaves.

There are four categories of leaves, healthy and three diseased leaf image categories. The first mentioned disease, common rust, occurs when corn leaves are infected by the fungus *Puccinia sorghi,* typically in high humidity environments when night time temperatures are between 65 and 70 degrees Fahrenheit. Gray leaf spot is caused by infection of *Cercospora zeae-maydis* when temperatures remain above 80 degrees Fahrenheit for 12 plus hours in a high humidity environment. Blight is an infection by *Setosphaeria turcica* when leaves are persistently wet for 6 plus hours and temperatures are between 64 and 81 degrees Fahrenheit. By identification of leaf diseases and understanding disease growth conditions, a proper remediation plan can be determined by farm managers.

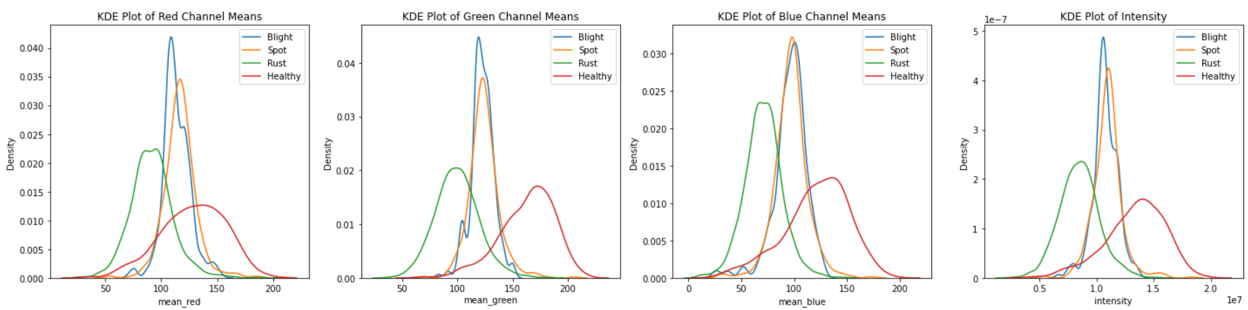


The above images of corn leaves are the first four images within each of the four leaf categories.

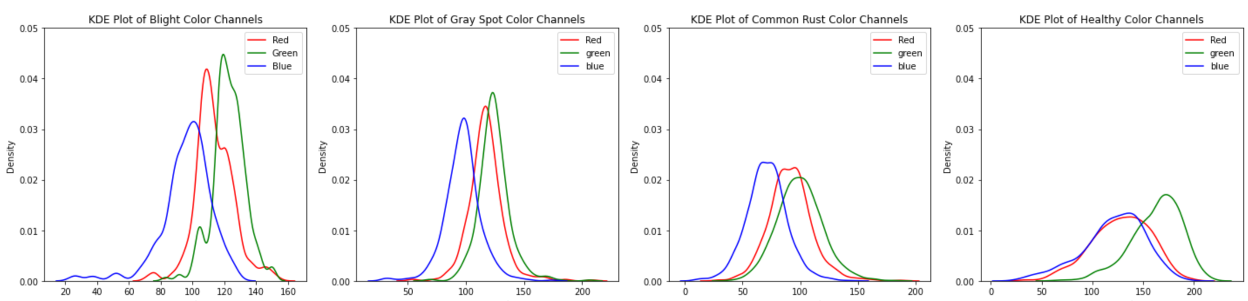
Image Averaging



The first 300 images in each of the leaf categories were averaged (see above). The average images reveal different intensities of green and brown which may be useful for modeling. Healthy leaves appear to have a slightly more intense green than the other images. The slightly different colored edges of the images, specifically the common rust average image, are a reflection of the non-standard way of photographing images that were found in the library and is a problem for modeling being the model may detect backgrounds rather than the actual corn leaves.



The above set of graphs demonstrate a difference between mean red, green, and blue values as well as image intensity for the four categories of images. It appears that rust and healthy have the most differences while spot and blight are the most similar.



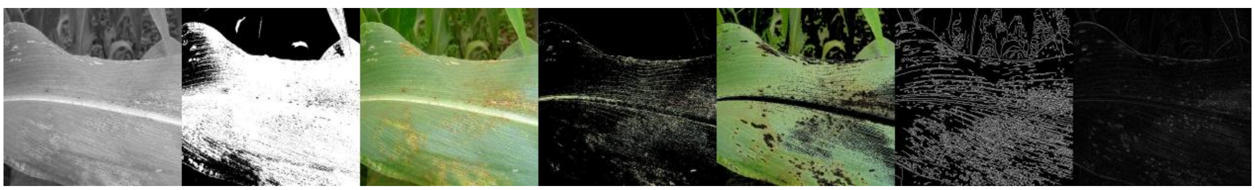
The above set of graphs demonstrate the mean values for RGB for each of the image categories. All four of the graphs have the same scale and demonstrate differences between the four categories in RGB values.

Image Processing

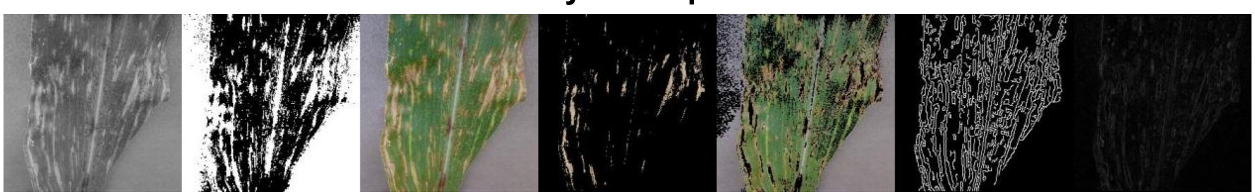
All images were then processed with various methods in Python with the desire that processed images will cause leaf diseases to be better identified by CNN. The final image to be processed by CNN included the original RGB, grayscale, binary, brown filter, green filter, Sobel edges and Canny edges. As part of this, the brown and green filters were developed through the creation of a function that filtered the predetermined RGB values. RGB values were determined through a process of color picking and trial and error (color picker can be found at https://imagecolorpicker.com/). These images were transformed to 180 by 180 by 3 pixels and then concatenated together to a final image shape of 180 by 1260 by 3 pixels.

Blight

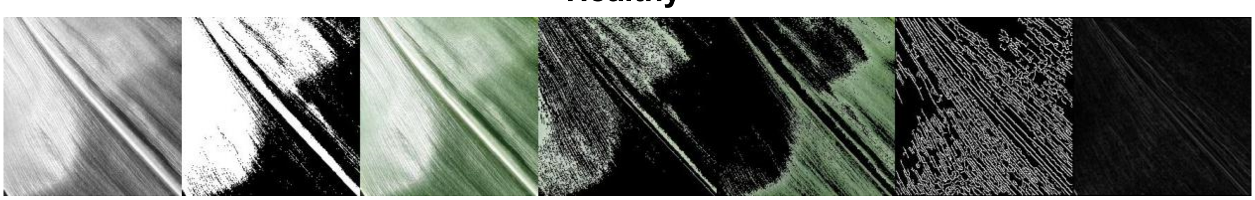
Common Rust



Gray Leaf Spot



Healthy



The above four series of images are processed images from the original RBG image and concatenated together to form a 180 by 1260 by 3 pixel image for CNN processing. From left to right, each sequence of images contains grayscale, binary, RGB, brown filter, green filter, Canny edges and Sobel edges.

Convolutional Neural Network (CNN) Modeling

On Kaggle, I researched previous attempts on this same dataset and found CNN modeling by others has resulted in 95.5% validation accuracy and 0.163 validation loss of corn leaf disease classification. In their attempts, they did not use any image processing or filterings such as I have here, however they did used image augmentation(this model can be found on Kaggle https://www.kaggle.com/milan400/0-00001adam-cornmaizeleaf-vgg16).

Simple RGB Model

The first CNN developed utilized the original RGB images to form a baseline for this project to compare with CNN's developed for the processed images. A simple CNN was developed for the original RGB images with random horizontal and vertical flip image augmentation. This model was called the Simple RGB Model and can be seen in figure 1 below. This particular model resulted in a 85.3% validation accuracy and 0.360 validation loss (See figure 2).

```
Model: "Simple RGB Model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_5 (Rescaling)      (None, 180, 1260, 3)      0
_________________________________________________________________
sequential_6 (Sequential)    (None, 180, 1260, 3)      0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 178, 1258, 32)     896
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 89, 629, 32)       0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 87, 627, 32)       9248
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 43, 313, 32)       0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 41, 311, 32)       9248
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 20, 155, 32)       0
_________________________________________________________________
flatten_5 (Flatten)          (None, 99200)             0
_________________________________________________________________
dense_11 (Dense)             (None, 128)               12697728
_________________________________________________________________
dense_12 (Dense)             (None, 4)                 516
=================================================================
Total params: 12,717,636
Trainable params: 12,717,636
Non-trainable params: 0
```

Figure 1. CNN developed for RGB images.

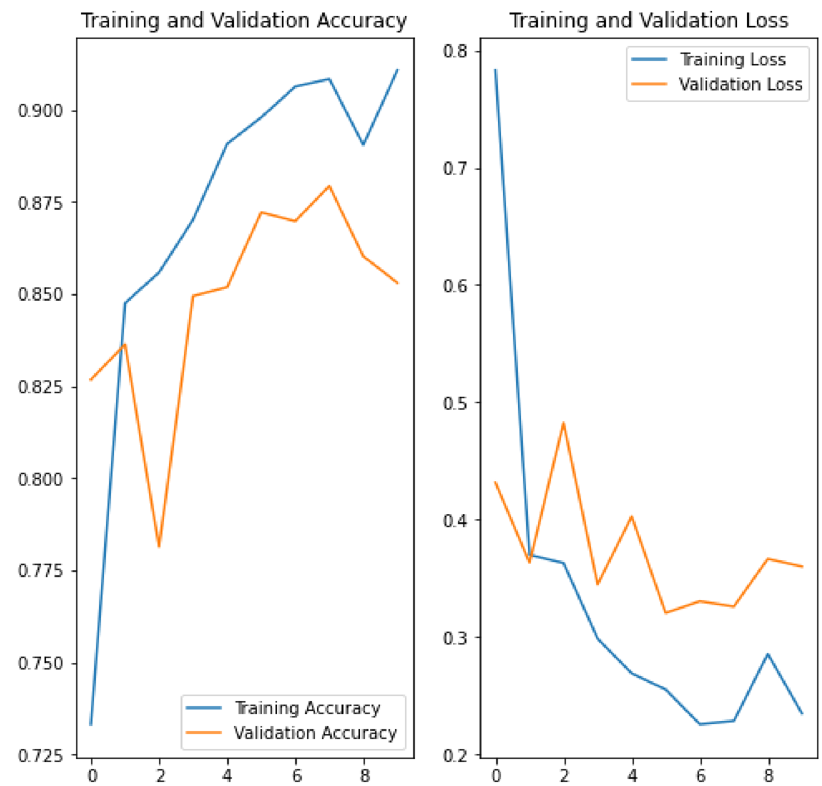


Figure 2. CNN Accuracy and loss for Simple RGB Model seen in figure 1.

Simple Processed Image Model

A second CNN model, very similar to the Simple RGB Model was developed, also with random horizontal and vertical flip image augmentation. This model, called "Simple Processed Image Model" can be seen below in Figure 3. Validation accuracy was significantly improved for this model to 95.9% with a validation loss of 0.203 (See figure 4).

```
Model: "Simple Processed Image Model"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
rescaling_8 (Rescaling)     (None, 180, 1260, 3)      0
_________________________________________________________________
sequential_11 (Sequential)  (None, 180, 1260, 3)      0
_________________________________________________________________
conv2d_19 (Conv2D)          (None, 178, 1258, 32)     896
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 89, 629, 32)      0
_________________________________________________________________
conv2d_20 (Conv2D)          (None, 87, 627, 32)       9248
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 43, 313, 32)      0
_________________________________________________________________
flatten_8 (Flatten)         (None, 430688)            0
_________________________________________________________________
dense_17 (Dense)            (None, 128)               55128192
_________________________________________________________________
dense_18 (Dense)            (None, 4)                 516
=================================================================
Total params: 55,138,852
Trainable params: 55,138,852
Non-trainable params: 0
```

Figure 3. CNN developed for processed images.

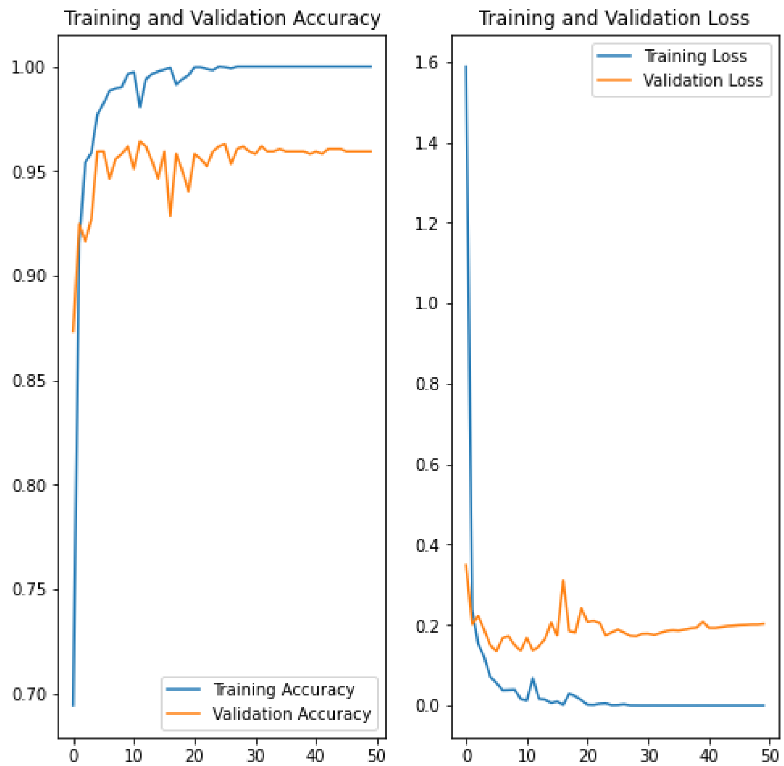


Figure 4. CNN Accuracy and loss for Simple Processed Image Model seen in figure 3.

A Different Approach: Transfer Learning

A third model was developed utilizing transfer learning from the VGG16 base model. For CNN, transfer learning utilizes a model trained on another set of images and applies it to a new set of images. This model was found to work best without image augmentation and resulted in a validation accuracy of 97.6% and a 0.119 validation loss (See figure 6 below). The input and output layers were trainable while the VGG16 layers were not trainable and retained the default Imagenet weights of the neural network. The VGG16 Model can be seen in figure 5 below.

```
Model: "VGG16 Model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 180, 1260, 3)]    0
_________________________________________________________________
vgg16 (Functional)           (None, 5, 39, 512)        14714688
_________________________________________________________________
global_average_pooling2d (Gl (None, 512)               0
_________________________________________________________________
dense_8 (Dense)              (None, 4)                 2052
=================================================================
Total params: 14,716,740
Trainable params: 2,052
Non-trainable params: 14,714,688
```

Figure 5. CNN model using the VGG16 for transfer learning.

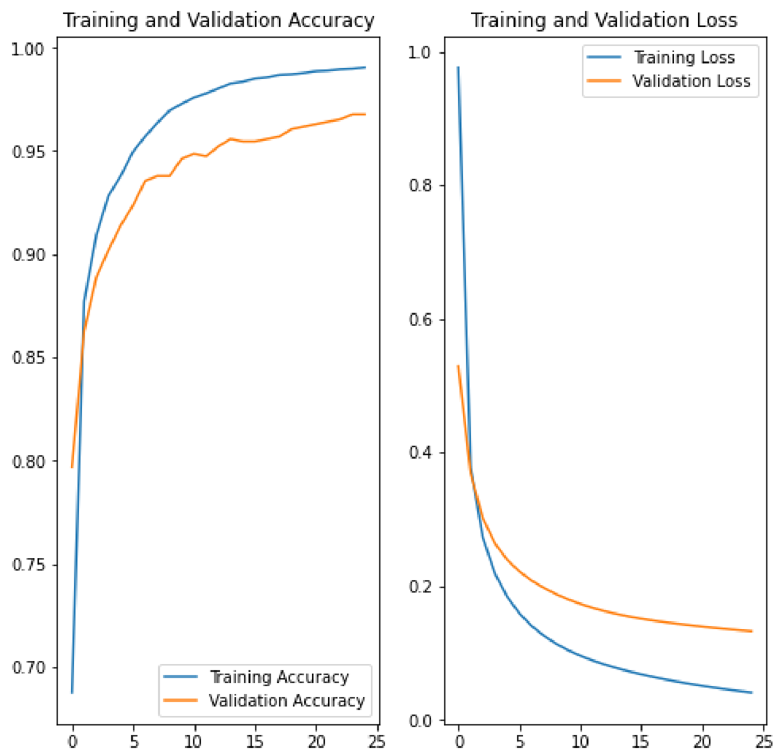
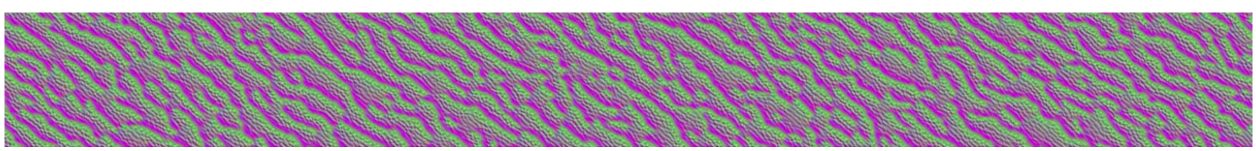


Figure 6. CNN Accuracy and loss for transfer learning model seen in figure 5.

Visualizing VGG16 Filters



One way to better understand how CNN works is to view the different filters contained within the model. The above image is a filter taken from the VGG16 block 2 convolutional layer 2 utilizing the Imagenet weights, and resembles some sort of diagel lines. The rectangular shape of the filter is a reflection of the input image size of 180x1260 pixels for the VGG16 model. Patterns and shapes of these filters are complex representations of images taken from the real world. While it is not always known what each individual filter is representing or detecting, the patterns are the general means by which CNN identifies and categorizes images.

Other Models

Comparison Kaggle Model

For comparison sake, a VGG16 model found on Kaggle is included in the overview of models table found below. This model is included due to it also utilizing VGG16 transfer learning so that it can be compared to the VGG16 Model developed in this project. In the Kaggle model however, images were not processed, image augmentation was used and the VGG16 model was trainable. Results for the VGG16 model used in this project were about 1.7% more accurate.

Random Forest
Random Forest modeling was also carried out on images statistics which included mean red, green and blue values; standard deviation of red, green and blue values; and intensity which is the sum of the pixels in each color channel. It was found that a Random Forest model using a max depth of 9, auto features, and 200 estimators produced an accuracy of 86.6%.

Overview of Models

Model Name	Image Augmentation	Transfer Learning	Validation Accuracy	Validation Loss
VGG16 Model	No	VGG16 (not trainable, input and output layers trainable)	97.6%	0.119
Simple Processed Image Model	Random horizontal and vertical flip	No	95.9%	0.203
Comparison Kaggle Model	Random height shift, zoom, rotation	VGG16 (All layers trainable)	95.5%	0.163
Random Forest	No	No	86.6%	
Simple RGB Model	Random horizontal and vertical flip	No	85.3%	0.360

Conclusion
In this project, feeding processed images such as, binary, grayscale, brown and green filtered, and edge detection, into CNN models produced greater accuracy than models that simply used similar RGB images. The Simple RGB Model was a base model that was fed unprocessed RGB images and utilized a simple CNN that produced 85.3% accuracy. When a similar CNN utilized the processed images, accuracy increased to 95.9%. Transfer learning with the VGG16 model with ImageNet weights utilizing the processed images increased accuracy to 97.6% and resulted in the most accurate model produced in this project. A similar model found on Kaggle utilized VGG16 also, but only on RGB images, and produced a slightly lower model accuracy of 95.5%. In these instances, image processing has been demonstrated to result in significantly

more accurate CNN models. Transfer learning has also been demonstrated to be a valuable tool in increasing CNN image classification accuracy.

Future Work

Future work for this project could involve further work on developing a CNN utilizing transfer learning from the VGG16 model. Additionally, seven images were utilized for the processed image models. It is assumed that some of these processed images contribute more to model accuracy than others and some may increase model noise. By testing which of the processed images are most valuable to increasing accuracy and which ones potentially produce model noise, and utilizing only the most important processed images, higher model accuracy should be able to be achieved. Image processing could also be utilized to remove unwanted backgrounds in images, such as those seen with the average images. Due to the diversity of backgrounds, it is difficult to simply remove backgrounds, however, simply cropping images should result in the elimination of or decreased influence of backgrounds and increase the proportion of the leaf relative to the size of the image.

To view code for this project:

https://github.com/haberkornm/Convolutional-Neural-Network-Of-Corn-Leaf-Disease-Images