[Team 23]Project 1.1:Leaf Wilting Detection in Soybean

Rajshree Jain
rjain27@ncsu.edu

Sai Shruthi Madhuri Kara
skara2@ncsu.edu

Sruthi Talluri
stallur2@ncsu.edu

1 METHODOLOGY

(1) Image Analysis - Color Intensity:
As we can see in the figure below, we have done the image analysis to track the evolution of the intensity of a particular color across the stack. The rationale behind performing this analysis is to understand how the color of the leaf is changing so that we can detect the leaf wilting in the Soybean plant.

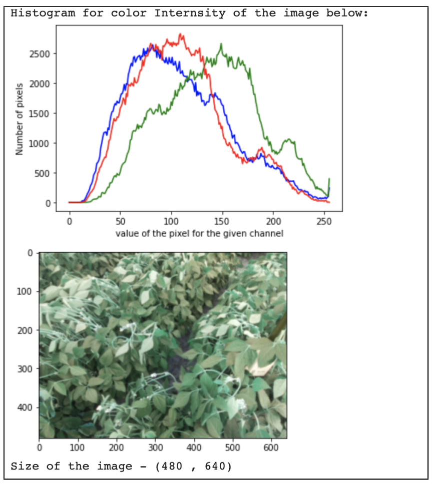

Figure 1: RGB Colour Intensity of Image

(2) Random Forest Classification
It is ensemble algorithm. Ensemble algorithms are those which combines more than one algorithms of same or different kind for classifying objects. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object. Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. Irregular timberlands RF or arbitrary choice woodlands are a gathering learning strategy for characterization, relapse and different errands, that work by developing a huge number of choice trees at preparing time and yielding the class that is the method of the classes (in arrangement) or mean forecast (in relapse) of the individual trees. RF is an improvement over the decision tree algorithm as it corrects habit of over fitting in decision trees to their training set.

(3) Feature Extraction and VGG19:
We have built a 3 layered convolutional neural network as a part of our attempt to detect the wilting in the images of the Soybean plant. We have utilized the pre-trained VGG19 model on Image Net for feature extraction by resizing the leaves of the Soybean plant image to 80 by 80 by 3 pixels 3 denoting the RGB color channels of the Soybean plant images.
The output from the model has been squeezed into the shape of (2440, 1, 2, 2, 512) and has been fed into the first layer of our 3-layer CNN. The second layer which is the dense hidden layer of the neural network has 1024 neurons. Since, we have 5 classes to classify the images into, we need 5 neurons at the output layer which has been taken care of. The structure of the neural network is as in the table below.For a glimpse, we have included the convolution layer shape at the end of each block instead of showing all the layers along with what is happening in the max-pooling layer as well.
We will be performing rigorous hyper-parameter tuning on the input size of the image that will be fed into the pre-trained VGG19 model as well as the number of neurons in the hidden layer in the later stages. We could observe the performance improvement before and after the feature extraction process.

Layer	**Output Shape**	**Params**
InputLayer	(None, 80, 80, 3)	0
Conv2D_End_Block1	(None, 80, 80, 64)	36928
MaxPooling2D	(None, 40, 40, 64)	0
Conv2D_End_Block2	(None, 40, 40, 128)	147584
MaxPooling2D	(None, 20, 20, 128)	0
Conv2D_End_Block3	(None, 20, 20, 256)	590080
MaxPooling2D	(None, 10, 10, 256)	0
Conv2D_End_Block4	(None, 10, 10, 512)	2359808
MaxPooling2D	(None, 5, 5, 512)	0
Conv2D_End_Block5	(None, 5, 5, 512)	2359808
MaxPooling2D	(None, 2, 2, 512)	0
Dense(Relu)	(None, 1024)	525312
Dense(Softmax)	(None, 5)	5125

2 MODEL TRAINING AND SELECTION

2.1 Model Training

(1) Oversampling Data
We observed that the data has class imbalance. One approach to address this imbalanced datasets is to over-sample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE in short

(2) Tensors in Image Classification:
We have converted all our input leaves images of the Soybean plant into the tensors as it can be used like a matrix

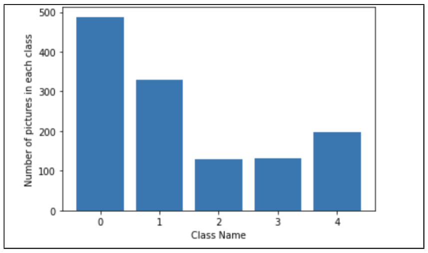

Figure 2: Leaf Wilting Image Class Distribution Before Sampling

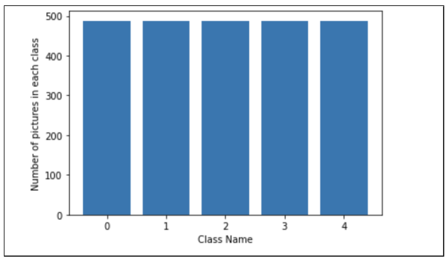

Figure 3: Leaf Wilting Image Class Distribution After Sampling

to perform operations like manipulating the image - data augmentation etc. This will be extremely beneficial while predicting the class a particular input image belongs to.

2.2 Model Selection

The approach proposes, usage of Random Forest Model and a Neural Network VGG19 model. For training both the models we have considered our image size to be 80*80. This image size, is one of the parameters that can be changed. In further steps, the algorithm will include a hyper parameter tuning on image size. Changing the image size, will lead to change in pixels. This impacts the performance of the models. Determining an appropriate image size leads to a better performing Neural network model.

Further it is also important to consider choosing an appropriate Optimiser that can help in understanding the loss accurately. Currently, the neural network model works with "rmsprop" optimizer. However, a better way would be to perform an analysis of which optimizer would perform better.

The Batch size and number of epochs is also a good way to determine which parameters and models perform better. Number of epochs, determines the number of times training data set will be propagated through the network. The batch_size defines, number of instances that will be propagated through the model at a point of time.

For now we have tried different types of algorithms, and preprocessing approaches like VGG19 with feature extraction, over sampling and performed early stopping procedure to analyse the epoch at which we get the best accuracy. With the help of this we could maintain the fact that VGG19 model would be a good model for our problem and feature extraction and SMOTE are methods that would definitely help. Evaluation section involves and points to plots, diagrams and table that state the same.

Class	Value	Value	Value	Value	Value
0	132	7	0	0	0
1	44	51	0	0	0
2	26	12	8	3	0
3	9	3	1	25	1
4	7	2	0	2	50

Table 1: Confusion Matrix for Random Forest

3 EVALUATION

The metric that has been used to compare different models and understand the correctness of models classification on the test data is F1 score. A measurement that considers both precision and recall to compute the score. The F1 score can be interpreted as a weighted average of the precision and recall values, where an F1 score reaches its best value at 1 and worst value at 0.

The approach involves implementation of two different models that are Random Forest and VGG19 deep learning model. In order to improve the classification prediction on test data, techniques like Over Sampling and Feature extraction have been implemented.

In order to determine the correctness of predictions for validation data in Random Forest, a confusion matrix as in Table 1 and Table 3 has been considered. This matrix can be used to calculate Precision, Recall and F1 score for all the classes. We can observe that the F1 score for all the classes has improved in Table 3, which lead us to understand that Oversampling would be a good step to get better classification results.

Also, it can be observed that the f1 score of all the combined classes has also increased from 0.64 to 0.93 when we used Random Forest with oversampling from Table 5.

In case of VGG19 we can see from Figure 4 and Figure 5 that as the validation Loss decreases(not considering the last epoch due to the early stopping procedure in figure 4 and figure 5), the validation accuracy is increasing and validation loss is decreasing. The model has been trained on different epochs in order to ensure that there is no over fitting.

It can be observed from Table 5, that although the validation accuracy for VGG19 is not as high as Random Forest, yet the F1 score is more, which means that the neural network method helped in solving the classification problem better.

As we can see from table 5, in case of results for test data the F1 score for Random Forest without oversampling was 0.12. On oversampling the data and implementing Random Forest the F1 score value improved to 0.14. Further, the VGG19 model on oversampled data with Feature extraction the F1 score improved to 0.38.

As observed from the table, the F1 score(Our evaluation metric) of VGG19 model with feature extraction and oversampling is the best. We will consider it as our base model for further evaluation.

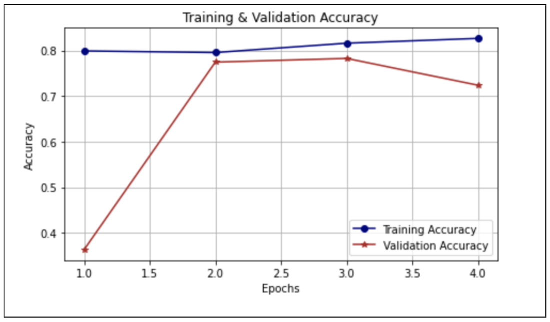

Figure 4: Validation Accuracy After Feature Extraction for VGG19

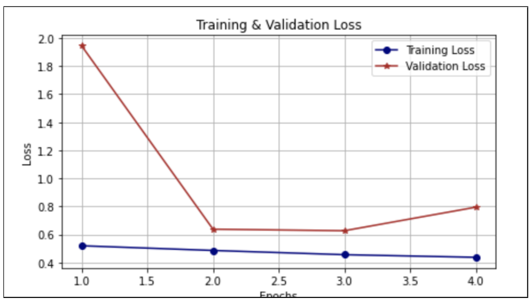

Figure 5: Validation Loss After Feature Extraction for VGG19

Class	Precision	Recall	F1 Score
0	0.61	0.95	0.74
1	0.68	0.54	0.60
2	0.89	0.16	0.28
3	0.83	0.64	0.72
4	0.98	0.82	0.89

Table 2: Class wise Precision, Recall and F1 Score for Random Forest

Class	Value	Value	Value	Value	Value
0	129	15	1	3	0
1	26	129	0	0	0
2	1	2	150	1	0
3	0	0	1	132	0
4	0	0	0	1	141

Table 3: Confusion Matrix for Random Forest with Oversampling

Class	Precision	Recall	F1 Score
0	0.83	0.87	0.85
1	0.88	0.83	0.86
2	0.99	0.97	0.98
3	0.96	0.99	0.98
4	1	0.99	1.00

Table 4: Class wise Precision, Recall and F1 Score for Random Forest with Oversampling

Model	Accuracy	Loss	Recall	Precision	Val. F1Score	F1Score
RF	0.69	-	0.79	0.62	0.64	0.12
RF - Sampling	0.93	-	0.93	0.93	0.93	0.14
VGG19	0.78	0.62	0.80	0.74	0.76	0.382

Table 5: Accuracy, Loss, Precision, Recall Values for Validation Data, F1 Score for Validation Data and Test Data

COLLABORATION

We have used the online Google Colab collaboration platform to merge and implement our code. We had several meetings and online discussion sessions through Zoom Links.

REFERENCES

[1] https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2
[2] https://keras.io/api/metrics/
[3] https://towardsdatascience.com/simple-guide-to-hyperparameter-tuning-in-neural-networks-3fe03dad8594
[4] https://medium.com/@franky07724_57962/using-keras-pre-trained-models-for-feature-extraction-in-image-clustering-a142c6cdf5b1#:~:text=Let's%20consider%20VGG%20as,model%20with%2019%20weight%20layers.