Leaf Wilting Detection in Soybean Crops

Shubham Miglani
Dept. of ECE
North Carolina State University
Raleigh, USA
smiglan@ncsu.edu

Tianji Gao
Dept. of ECE
North Carolina State University
Raleigh, USA
tgao5@ncsu.edu

Mohammad Sareeb Hakak
Dept. of ECE
North Carolina State University
Raleigh, USA
mhakak@ncsu.edu

I. METHODOLOGY

A. *Data Acquisition*

The data provided for this project is in the form of images with labels defined separately. The labels were sorted to match the images. Google Colab was the choice of the working environment due to the GPU support and high RAM availability. Keras built on TensorFlow was the preferred framework of implementation. Image loader in Keras was used to load the images. Since, the image size is large, the RAM got exhausted continuously. To tackle this, we used the Image data generator feature in Keras which loads the images in batches. This makes sure that the RAM is managed properly.

However, the Keras Image data generator expects labeled images to be available in certain folder hierarchy, the folders being subdivided into different classes. So, the images were divided into 5 classes and split into training (702 images) and validation (194 images) sets. Also, the images were resized to a square image of 124x124.

B. *Scheduled Plan/ Approach*

The proposed plan of work for the project was divided into baseline implementation and subsequent fine tuning. The initial idea was to implement a CNN which makes sense in working with images and due to various implementations of CNNs on soybean plant disease detection [2] which have a similar pattern of images as in our task.
After hyperparameter tuning, the optimized CNN is chosen as the baseline for the next part whereby this model would be used along with a pre-trained network like resnet or vgg16 or a concatenation of few [1]. A second network will be built on top of the pre-trained model and then the optimized CNN weights will be loaded. After that, the network will be trained and fine-tuned to hopefully get a better solution. This is planned to be implemented in Project C2.

II. MODEL TRAINING AND HYPER-PARAMETER SELECTION

The initial structure of our neural network was based on the crop disease detection paper [2].

For hyperparameter selection, the variables which we chose to optimize are the dropout value, the number of neurons in the dense layer, learning rate and feature map size. First, the number of neurons were varied, the best value was selected and updated for the next hyperparameter and so on. The initial hyperparameter values are shown as in Table1.

Table 1
Initial Hyperparameter Values

Hyperparameter	**Default value**
Neuros in Dense Layer	512
Feature Map Size	32
Learning Rate	0.001
Dropout Probability	0.2
Activation Function	ReLU

The range of hyperparameters checked and the best values are shown in Table

Table *2*
Range of Hyperparameter Values

Hyperparameter	**Range Checked**	**Best Value**
Neuros in Dense Layer	16,32,64,128,256,512,1024	**32**
Learning Rate	0.001,0.0015,0.003,0.0001,0.0005	**0.001**
Feature Map Size	8,16,24,32,48,64	**32**
Dropout Probability	0.1,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.50	**0.2**

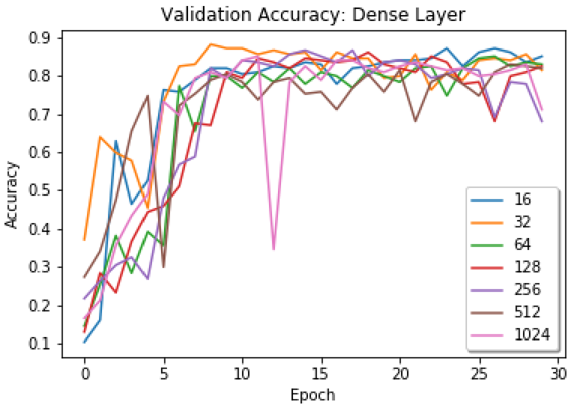
It was also noticed that the validation accuracy was dropping after some epochs and there was not a constant reduction, so the maximum validation accuracy was recorded as well to be used along with early stopping.

A. *Number of neurons in the hidden layer*

Firstly, the number of neurons in the dense layer were varied. The best value chosen was for 32 neurons.

Table 3
Validation Accuracy for change in neurons in dense layer

Number of neurons	**Validation accuracy (After 30 Epochs)**	**Max Validation accuracy**
16	0.8505	0.8711
32	0.8144	**0.8814**
64	0.8298	0.8505
128	0.8247	0.8608
256	0.6804	0.8659
512	0.8195	0.8298
1024	0.7113	0.8505

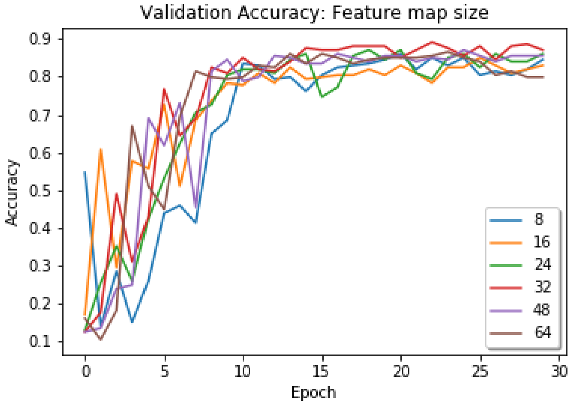


B. *Feature Map Size*

For the three convolution layers, the feature maps for the second layer was 2 times in the first layer and the feature maps in the third layer was 2 times the second layer.

Table 4
Validation Accuracy for change in Feature map Size

Feature Map Size (Layer 1)	**Validation accuracy (After 30 Epochs)**	**Max Validation accuracy**
8	0.8453	0.8608
16	0.8298	0.8505
24	0.8608	0.8711
32	0.8556	**0.8917**
48	0.8556	0.8711
64	0.7989	0.8659



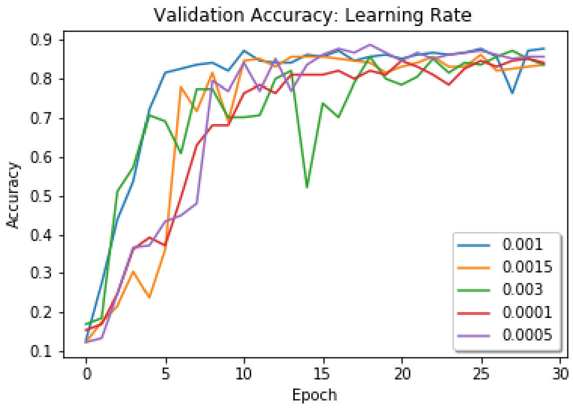
Based on the data, 16 feature maps are chosen in the first layer.

C. *Learning rate for Adam Optimizer*

Thirdly, the learning rate was varied. The curve for validation accuracy is as shown:

Table 5
Validation Accuracy for change in Learning Rate

Learning Rate	**Validation accuracy (After 30 Epochs)**	**Max Validation accuracy**
0.001	0.8762	**0.8762**
0.0015	0.8350	0.8608
0.003	0.8350	0.8711
0.0001	0.8402	0.8505
0.0005	0.8556	0.8865

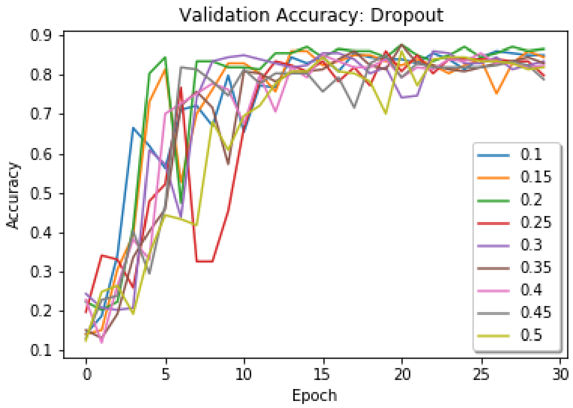


Although the max validation accuracy at learning rate of 0.0005 is higher, 0.001 is chosen as it had more consistent results. The same validation accuracy could not be reproduced consistently for 0.0005.

D. *Dropout Probability*

Table 6
Validation Accuracy for change in Dropout

Dropout	**Validation accuracy (After 30 Epochs)**	**Max Validation accuracy**
0.10	0.8505	0.8608
0.15	0.8453	0.8608
0.20	0.8659	**0.8762**
0.25	0.7989	0.8608
0.30	0.8350	0.8606
0.35	0.8298	0.8762
0.40	0.8247	0.8556
0.45	0.7886	0.8505
0.50	0.8195	0.8608



After the structure was finalized, it was noticed that there was significant over-fitting with high training accuracy and less accuracy on validation data. Dropout had already been included and L2 regularization was introduced but it didn't help much. Max norm was decided to be used based on [5] along with dropout and the max norm values were fine-tuned and Batch Normalization was also used.

The final structure of our network is:

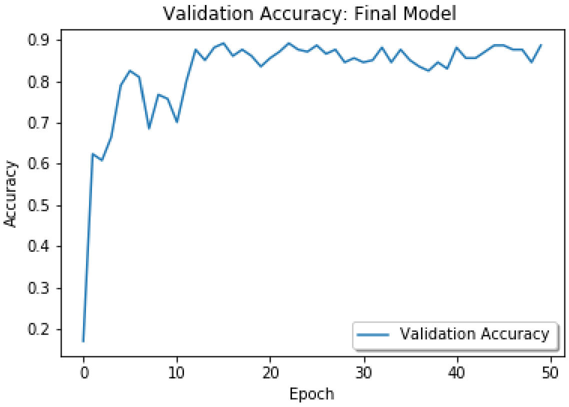
Table 7
Final Network Structure

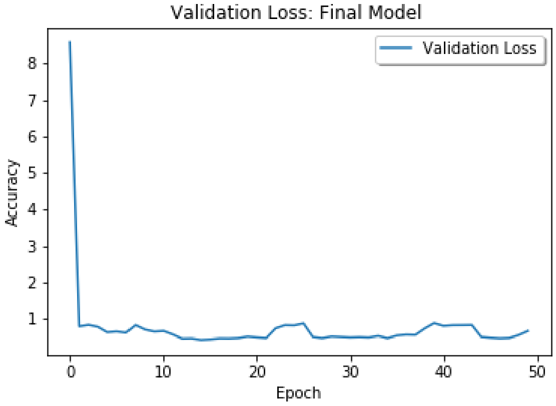
Layer type	Output Shape	Parameters
Conv2D	(None,128,128,32)	896
Batch Normalization	(None,128,128,32)	128
Max Pooling 2D	(None,63,63,32)	0
Dropout	(None,63,63,32)	0
Conv2D	(None,63,63,64)	32832
Batch Normalization	(None,63,63,64)	256
Max Pooling 2D	(None,31,31,64)	0
Conv2D	(None,31,31,128)	8320
Batch Normalization	(None,31,31,128)	512
Max Pooling 2D	(None,15,15,128)	0
Dropout	(None,15,15,128)	0
Flatten	(None,28800)	0
Dense	(None,32)	921632
Batch Normalization	(None,32)	128
Dropout	(None,32)	0
Dense	(None,5)	165

III. EVALUATION

A. *Final values*

The network chosen had a loss of **0.4252** and validation accuracy of **0.8814.** Early stopping was used to get the weights of the network with minimum loss.
The learning curves for the final model are as shown:





B. Error Metrics

The distribution of images and their error is as shown in Table8.

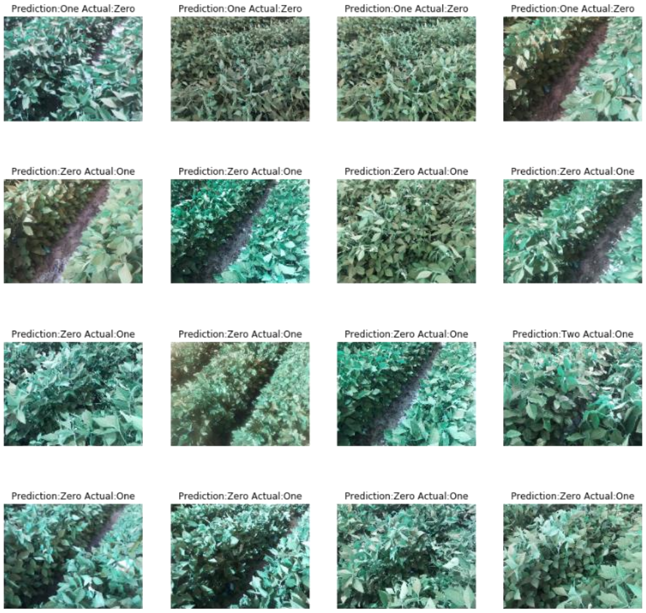
Table 8
Error Distribution

Class	Input Data Distribution	Training Set	Validation Set	Error
0	437	341	96	4
1	154	121	33	12
2	113	89	24	4
3	116	91	25	2
4	76	60	16	1

Out of 12 images being misclassified in Class1, 11 are classified as 0. This can be attributed to the fact that most of the images in the training set are of Class 0. Also, Class 1 and Class 0 are adjacent to each other.

Many factors were found for the errors. Some images were out of focus, some were blurred at the edges and few had more percentage of soil than the plants.

Some of the incorrectly identified images are as shown:


Prediction:One Actual:Zero
Prediction:One Actual:Zero
Prediction:One Actual:Zero
Prediction:One Actual:Zero
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Two Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One
Prediction:Zero Actual:One

REFERENCES

[1] B. Zhong, Q. Ge, B. Kanakiya, R. Mitra, T. Marchitto, E. Lobaton, “A Comparative Study of Image Classification Algorithms for Foraminifera Identification,” IEEE Symp. Series on Computational Intelligence (SSCI), 2017.

[2] Serawork Wallelign, Mihai Polceanu, Cedric Buche, “Soybean Plant Disease Identification Using Convolutional Neural Netowrk” Artificial Intelligence Research Society Conference (FLAIRS-31).

[3] King AC, Purcell LC, and Brye KR (2009) Differential wilting among soybean genotypes in response to water deficit. Crop Science 49:290-298. doi: 10.2135/cropsci2008.04.0219

[4] Pathan, SM et al. (2014) Two soybean plant introductions display slow leaf wilting and reduced yield loss under drought. Journal of Agronomy and Crop Science 200:231-236. doi: 10.1111/jac.12053

[5] Srivastava, Nitish, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15, no. 1 (2014): 1929-1958.