# Leaf Wilting Detection in Soybean Competition Project Team 9

Jeffrey Barahona
*North Carolina State University*
Raleigh, United States
jabaraho@ncsu.edu

Bavithra Dhamodaran
*North Carolina State University*
Raleigh, United States
bdhamod@ncsu.edu

Krishnachaitanya Pullakandam
*North Carolina State University*
Raleigh, United States
kpullak@ncsu.edu

## I. Introduction and Motivation

Due to the changing climate, droughts are becoming more common, affecting crop yields and potentially increasing water stress in crops including soybeans. To address this, soybean breeders are working to create more drought-resistant species of soybeans to reduce the load on irrigation systems in regions where it is cost and resource prohibitive to irrigate more frequently[1]. The most widely used indicator of drought tolerance/sensitivity in soybean is leaf wilting during periods of water stress.Breeders collect this data in the most low-tech way imaginable -— walking the field with a notebook and writing down ratings to indicate how wilted each field plot looks [2].A machine learning approach can be used to tackle this problem and automate parts of this process, improving the efficacy of the soybean breeders efforts. In this project, we used transfer learning techniques and weighted average ensembling to automate the severity rating of water stress in soybeans crops.

## II. Dataset

### A. Main Competition Data

The dataset is composed of images of soybean crops at various times of the day at different stages of water stress. The labels for this dataset are the ratings of the severity of the plant's water stress and are stored in a separate CSV file.The image dimensions were $640 \times 480$. This dataset was highly skewed towards healthy plants with rating of zero and few plants with ratings of 4. The distribution is shown in figure 1

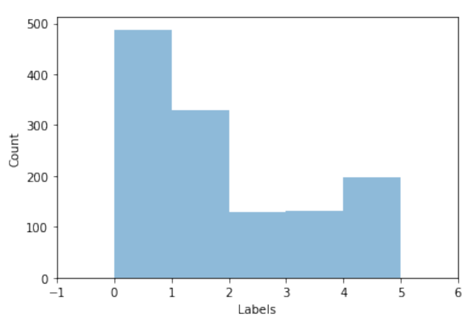


Fig. 1. Class Histogram

### B. Extra Credit Data

This dataset is a subset of the main competition data with an additional csv file containing weather and environmental conditions at the time of image annotation.

## III. Methodology, Training, and Hyperparameters

In this section, we will provide an overview for the methods used to train the model and to select the hyperparameters. Each subsection will contain implementation specific information concerning each stage of the process.

To train the model, the dataset was divided into a testing set, containing the new images provided for C2 with an equal distribution of images between the classes, and set used to select hyperparameters. The hyperparameter set data was shuffled and split into five sections used for five fold cross validation. Each section was independently augmented to prevent the introduction of transformations of the same image between different folds and potentially artificially increase validation accuracy and the chance of overfitting. The testing data was also augmented using the same procedure.

The augmented data was visually inspected to ensure that the model would not receive inputs that were too distorted or unrealistic. Then the model architecture was iteratively developed by training models until they overfit and adding more layers after each iteration until no further loss or accuracy improvements could be obtained by increasing the model's complexity. Then, the training hyperparameters were fine tuned using trial and error. Grid Search and Random Search could not be implemented to select hyperparameters due to the time limits of Google Colaboratory.

### A. Data Augmentation

Each fold is augmented by randomly cropping ten randomly sized sections of each image then passing the images through Keras's ImageDataGenerator class to generate various transformations of the images, including rotation, shearing, zooming, horizontal flipping, and translations. Using this process. Each class was augmented equally to balance the dataset and resized to $244 \times 244$.

To augment the data, the Keras ImageDataGenerator class was used to generate data in real time with the runtime provided by Google Colaboratory. Colaboratory provides a Tesla K80 GPU and a 2 core Intel Xeon processor with 25

GB of RAM and 12GB of VRAM. This allowed for fast data generation and fast training times.

Fig. 2. Unaltered Soybean Image

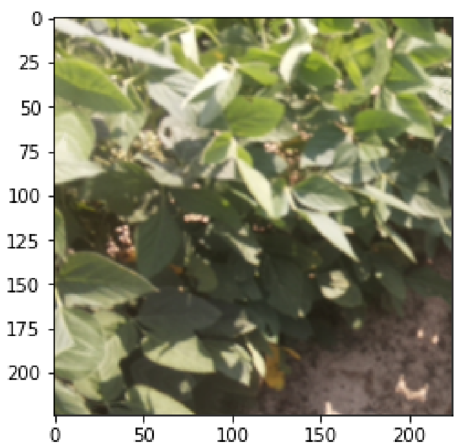


Fig. 3. Example of Data Augmentation

### B. Model Architecture

The model developed uses Resnet101 as a base for transfer learning with a single Dense layer at the top with a softmax activation. For the final prediction, four of these models were trained, and their predictions were averaged together to form the final output.The output fully connected layer uses the softmax activation function which outputs a multinoulli distribution which is useful for multi-class classification in addition to its behavior as a maximum likelihood. estimator[3].

Resnet101 was chosen by using 5 fold cross validation and comparing the results to other pre-trained models provided by Keras. We tested VGG19, VGG16, Resnet50, InceptionResnetV2, and we found that the confidence intervals for these models were lower or overlapped with the lower bounds of Resnet101 for the data provided.

### C. Training Hyperparameters

The model was configured for train using the Adam optimization algorithm and categorical cross-entropy loss function. The Adam optimizer uses a single learning rate and updates weights iteratively based on the training data hence it is suitable for our requirement. The categorical cross-entropy loss works well with multi-class classification as its outputs a probability over the classes for each image.

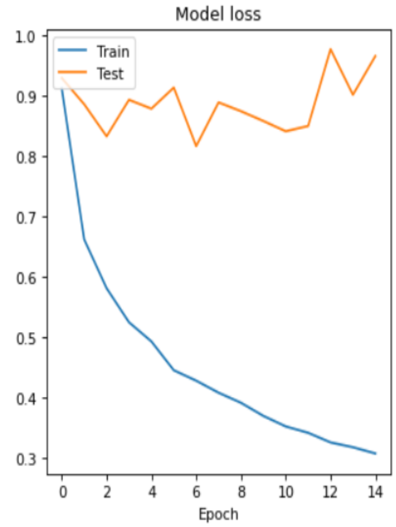


Fig. 4. Model Loss

## IV. Evaluation

TABLE I
CONFUSION MATRIX

| Dehydration Severity | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 374 | 130 | 33 | 3 | 0 |
| 1 | 60 | 440 | 32 | 2 | 6 |
| 2 | 67 | 107 | 284 | 74 | 8 |
| 3 | 15 | 71 | 134 | 229 | 91 |
| 4 | 1 | 47 | 7 | 7 | 264 |

The model was trained with a batch size of 32 over 14 epochs and 166 steps per epoch. A model with an validation accuracy of 66% was obtained. To better evaluate the performance of this model, we examined the precision, recall, and F1 scores of each model classification type. Additionally, we generated a confusion matrix to show more explicitly how the model performed with each class type. Tables III and I show how well the model performed for each class.

TABLE II
CLASSIFICATION STATISTICS

| Dehydration Severity | Precision | Recall | F1 Score |
|---|---|---|---|
| 0 | 0.72 | 0.69 | 0.71 |
| 1 | 0.55 | 0.81 | 0.66 |
| 2 | 0.56 | 0.53 | 0.54 |
| 3 | 0.73 | 0.42 | 0.54 |
| 4 | 0.81 | 0.85 | 0.83 |

Using cross validation, we determined that the model has an average loss of 0.7695 with a standard deviation of 0.1179. The 95% confidence interval for the model is $[0.5337, 1.005]$. The high variance of the model loss indicates that it is highly

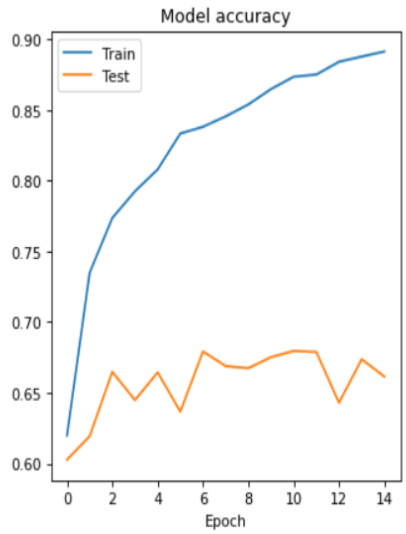


Fig. 5. Model Accuracy

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:             annotation   R-squared (uncentered):                   0.657
Model:                            OLS   Adj. R-squared (uncentered):              0.652
Method:                 Least Squares   F-statistic:                              130.1
Date:                Sun, 12 Apr 2020   Prob (F-statistic):                   1.05e-135
Time:                        22:54:08   Log-Likelihood:                         -854.23
No. Observations:                 620   AIC:                                      1726.
Df Residuals:                     611   BIC:                                      1766.
Df Model:                           9
Covariance Type:            nonrobust
============================================================================================
                               coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------
hours                       -0.0414      0.018     -2.294      0.022      -0.077      -0.006
air_temperature              0.3460      0.047      7.435      0.000       0.255       0.437
humidity                  -375.9897     43.131     -8.717      0.000    -460.693    -291.286
pressure                 -3.536e-05   1.02e-05     -3.458      0.001   -5.54e-05   -1.53e-05
longwave_radiation           0.0073      0.003      2.556      0.011       0.002       0.013
convective_precipitation    -0.6591      0.161     -4.095      0.000      -0.975      -0.343
potential_energy            -0.0010      0.000     -5.566      0.000      -0.001      -0.001
potential_evaporation       -1.4167      0.304     -4.667      0.000      -2.013      -0.821
precipitation                0.1317      0.034      3.925      0.000       0.066       0.198
==============================================================================
Omnibus:                       63.042   Durbin-Watson:                   1.571
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               85.105
Skew:                           0.769   Prob(JB):                     3.31e-19
Kurtosis:                       3.965   Cond. No.                     1.11e+08
==============================================================================
```

Fig. 6. Feature Importance

sensitive to the data points provided. This is an indicator that this model may have difficulty generalizing on the data. On the testing set, this model achieved 60% accuracy and 0.7906 RMSE loss.

## V. Extra Credit

### A. Model Specification

From the extra credit feature set, we chose the most relevant features for our classification task based on a p-value analysis (set a p-value threshold of 0.05), chose the following 9 features for modeling with respect to extra feature set: Hours, Air Temperature, Humidity, Pressure, Long wave radiation, Convective Precipitation, Potential Energy, Potential Evaporation, and Precipitation

In order to make use of the extra credit features, we used the below approach; we flattened out the final layer of the pre-trained model (resnet101 in our case) and concatenated the flattened layer with the extra credit features. After concatenation used a few dense and dropout layers before the output layer. The drop out layers are used with a drop out rate set to 0.5 to make the models more robust.

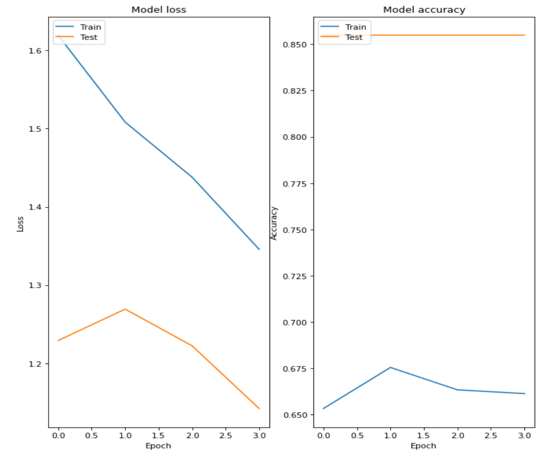


Fig. 7. Training Curves

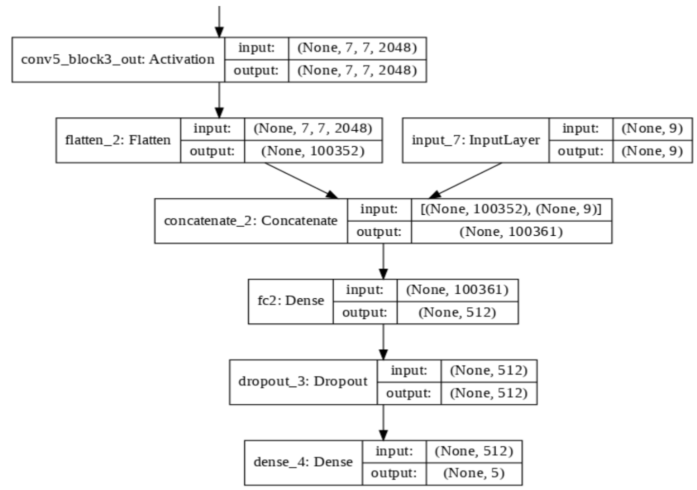


Fig. 8. Concatenated Model

### B. Evaluation

From the statistics collected, it appears that the model has a higher accuracy with the validation set, but it also has higher loss.

TABLE III
TRAINING STATISTICS

| Training Loss | Training Accuracy | Validation Loss | Validation Accuracy |
|---|---|---|---|
| 1.3579 | 0.6512 | 1.1662 | 0.8548 |

## References

[1] S. M. Pathan, J.-D. Lee, D. A. Sleper, F. B. Fritschi, R. E. Sharp, T. E. Carter, R. L. Nelson, C. A. King, W. T. Schapaugh, M. R. Ellersieck, H. T. Nguyen, and J. G. Shannon, "Two soybean plant introductions display slow leaf wilting and reduced yield loss

under drought," *Journal of Agronomy and Crop Science*, vol. 200, no. 3, pp. 231–236, 2014, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jac.12053, ISSN: 1439-037X. DOI: 10.1111/jac.12053. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jac.12053 (visited on 03/04/2020).

[2] C. A. King, L. C. Purcell, and K. R. Brye, "Differential wilting among soybean genotypes in response to water deficit," 2009, ISSN: 0011-183X. [Online]. Available: https://pubag.nal.usda.gov/catalog/748322 (visited on 03/04/2020).

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.