Competition Project: Phase 1

Machine leaning approaches towards classifying  
 visual water-stress found in soybean images

Abhishek Ranjan Singh *Electrical and Computer Engineering*  
*North Carolina State University*Raleigh, NC, USA  
arsingh3@ncsu.edu

Adam Christopher Watts  
*Wilson College of Textiles  
North Carolina State University*Raleigh, NC, USA  
[acwatts4@ncsu.esu](mailto:acwatts4@ncsu.esu)

Sankalp Singh Gaharwar  
*Computer Science*   
*North Carolina State University*Raleigh, NC, USA  
ssgaharw@ncsu.edu

# Data

## Data

A training set consisting of 1025 agricultural soybean photos each with a resolution of 640 X 480 pixels was provided. The training set consisting of 5 classes listed in Table 1. Visual inspection of the photos revealed a wide range of exposure levels with some photos being underexposed and rather dark while others overexposed and rather bright.

##### Table 1. Soybean plant classifications and descriptions.

|  |  |
| --- | --- |
| **Categorical Number** | **Classification Description** |
| 0 | No Wilting |
| 1 | Leaflets folding inward at secondary pulvinus, no turgor loss in leaflets or petioles |
| 2 | Slight leaflet or petiole turgor loss in upper canopy |
| 3 | Moderate turgor loss in upper canopy |
| 4 | Severe turgor loss throughout canopy |

## Augmentation

Image augmentation was performed by using Keras ImageDataGenerator class to synthetically extend the training set to 3075. The first 1025 images were the original training set while the remaining 2050 consisted of 2 complete rounds of augmentation using the parameters found in Table 2. The parameters and values were chosen to emulate the variation found in the provided data set.

##### Table 2. Keras ImageDataGenerator parameters

|  |  |
| --- | --- |
| **Parameter** | **Value / Boolean** |
| Rescale | 1.0/255 |
| Shear Range | 0.1 |
| Zoom Range | 0.25 |
| Horizontal Flip | True |
| Rotation Range | 25 |
| Fill Mode | ‘Wrap’ |

## Image Pre-Processing

Color….. do we need color….. Maybe do grayscale… only?

Histogram equalization

# Machine Learning Models

## Convolution Neural Networks (CNN)

The first CNN was constricted based on to the LeNet-5 architecture using Tensorflow with the Keras API [1], [2]. The input dimensions in the original LeNet-5 were only 32 X 32 greyscale images compared to the 640 X 480 RGB images used here. Activation functions for this project were modified to Relu instead of the original tanh. The input layer was connected to three, 2-Dimensional convolution layers, followed by being flatten down to two fully dense multilayer perception (MLP) of 16 and 8 neurons respectively for a total of 5 layers. The final output layer was a Stride sizes for all convolution layers were set to 1 with the Kernels decreasing from 11, 5, and finally 3 for respectively for the 3 convolution layers. The number of filters increased for each subsequently deeper convolutional layer with 64, 128, and 256 filters respectively. A maximum pooling layer with a pool size of 2 were placed in between each convolutional layer. 50% dropout was performed between each dense MLP.

Stochastic gradient descent (SGD) with used as the optimization algorithm with a batch size of 16. Training was performed on the training images without any pre-processing or augmentation. A validation split of 15% was utilized with a batch size of 16 without shuffling. Total number of trainable params were 1.7 million.

Poor stability was noticed with the training leading to exploding gradients. The optimization algorithm was altered from SGD to ADAM which lead to stable learning [3]. Although, stability was achieved, the training and validation accuracy failed to improve with each epoch. Training accuracy and validation accuracy was stagnant at 46.84% and 51.95% respectively. The CNN architecture was modified to as shown in Table 3 to improve accuracy.

Accuracy didn’t improve until dropout was completely removed leading to overfitting. Although removing dropout led to overfitting of the training data, it demonstrated that high dropout on can severely impede model learning even when the architecture was drastically changed to almost 80 million parameters. Dropout was restored prior to the Softmax output layer with a 0.20 value, resulting in a training and validation accuracy of 96.10% and 73.38% respectively at 15 epochs.

##### Table 3. CNN architecture for second attempt. C, S, F, D, O stand for convolution, subsampling, flatten, dense MLP, and dropout layer respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layer** | **Output Size** | **Filters** | **Kernel / Pool** | **Padding** |
| 1 | 640 X 480 | - | - | - |
| C1 | 640 X 480 | 64 | 11 X 11 | SAME |
| S2 | 320 X 240 | - | 2 X 2 | - |
| C3 | 320 X 240 | 128 | 3 X 3 | SAME |
| C4 | 320 X 240 | 128 | 3 X 3 | SAME |
| S5 | 160 X 120 | - | 2 X 2 | - |
| C6 | 160 X 120 | 256 | 3 X 3 | SAME |
| **Layer** | **Output Size** | **Neurons** | **Dropout** | **Activation** |
| F7 | 1,228,800 | - | - | - |
| D8 | 64 | 64 | - | Relu |
| O9 | 64 | - | 0.50 | - |
| D10 | 32 | 32 | - | Relu |
| O11 | 32 | - | 0.50 | - |
| D12 | 10 | 5 | - | Softmax |

Augmented images with their histograms equalized were then added to the training and data. It was hypothesized that the larger data set would improve accuracy as the CNN would have a larger data set to learn from over the same number of epochs. However, both training and validation accuracy deteriorated and could not improve above 50-60%. Batch normalization was added between each dense layer which speed up convergence but didn’t improve accuracy [4]. Furthermore, shuffling of dataset was also performed to minimize the chances that the CNN would “memorize” the data. The augmented images were then removed from the training set to further diagnose the poor accuracy results.

4th attempt: Max pooling of 4 prior to Dense layer. Very large dense MLP 4K – 2K, batch size up to 21 (more GPU memory)

5th Attempt: optimization with dropout from 25 to 50%, epochs set to 18

# Results

……………………………………………………

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Ha, “Gradient-Based Learning Applied to Document Recognition,” p. 46, 1998.

[2] F. Chollet, *Keras*. 2015.

[3] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Jan. 2017.

[4] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Incorporated, 2019.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**