# Confidence-Driven Hierarchical Classification of Cultivated Plant Stresses

Logan Frank[1] Christopher Wiegman[2] Jim Davis[1] Scott Shearer[2]

[1] Department of Computer Science and Engineering
[2] Department of Food, Agricultural, and Biological Engineering
Ohio State University
{frank.580, wiegman.3, davis.1719, shearer.95}@osu.edu

## Abstract

*The application of convolutional neural networks (CNNs) and deep learning to different domains has become increasingly popular in the last several years. In particular, such models have been used in the agriculture domain to identify plant species, identify plant stresses, and estimate crop yields. Although there has been much success in applying these techniques to the agriculture domain, these works contain many shortcomings that are hindering their chance for adoption in practice (e.g., lack of domain knowledge, predicting only specific stress types, etc.). We address issues of previous works for the task of plant stress identification by applying a hierarchical classification approach employing confidence as a means to determine the specificity of a classification. This work is a collaboration between computer science and agricultural engineering experts.*

## 1. Introduction

In modern agriculture, plant stress identification is a critical task for protecting the crop through the growing season. A stress is defined as an external condition that adversely affects the growth, development, or productivity of plants (example images of plant stresses are shown in Fig. 1). Currently, annual yield losses due to disease (a subset of stresses) in North America are estimated at 11% in soybeans and ranges from 2% to 17% in corn [18, 30]. Total corn production for the U.S. and Ontario, Canada from 2012 to 2014 was almost 54 billion bushels valued at $244 billion. If the 2% loss estimate is realized, this results in losing more than 1 billion bushels of grain costing almost $5 billion in revenue [29]. Total soybean production from 2010 to 2014 in the U.S. was 17.2 billion bushels valued at over $209 billion. With the estimated 11% yield loss, this equates to over 1.9 billion bushels of yield loss, costing farms more than $23 billion [1]. These numbers are widely regarded as understated considering they do not account for additional costs associated with misdiagnosis, field scouting, and diagnostic fees (such as laboratory or soil testing). When coupled with the unpredictability of climate change and modern trends in cultivation techniques, disease pressure is a growing issue impacting farms across North America [29, 1, 32, 39, 27].

(a) (b) (c)

Figure 1: Example plant stresses. (a) Bacterial spot in tomato plants, (b) Phosphorus deficiency in corn plants, and (c) Stalk rot in corn fields.

Rapid and accurate classification of plant stresses is critical for effective management, enabling precise application of remedial measures to only the affected areas of a field, reducing both the negative impact on yield and the environment by minimizing the total application amount of chemicals (*e.g.*, fertilizer, fungicides, pesticides, etc.) [15, 23, 37, 41, 3]. Current monitoring methods involve human experts scouting fields in person, using visual disease estimation along with microscopic, serological, and microbiological techniques. However, these methods often fail to provide results in a timely manner [24, 31, 17, 4, 26, 36]. In the time span it takes to receive results from a test, a stress could have spread, causing further yield and revenue loss. Furthermore, despite the need for human experts, their number is diminishing and the ability to scale crop monitoring capabilities to meet modern demand is increasingly challenging [27].

Automatic visual classification of stress symptoms using CNNs has shown substantial promise in the past few years due to their native ability for representational learning

[22, 7]. Numerous studies have also been published evaluating the application of CNNs to this task in various forms, such as comparing architectures, datasets, data preprocessing methods, and more [28, 38, 14, 21, 2, 40, 16, 13, 12, 11, 35]. Despite this intensive study, there are still many shortcomings that prevent large-scale adoption of deep learning models in commercial applications. Such shortcomings include the lack of incorporated domain knowledge, the limited ability to expand functionality, rigid approaches providing only specific class labels from training, and the lack of a confidence value for predictions.

Crop stress identification poses a unique challenge partly due to the difficulty of building a training dataset, as a domain expert is necessary for data annotation [20, 5, 6]. Furthermore, not only is there substantial variation in the appearance of symptoms for one stress, there can also be visual overlap in symptoms between different stresses as well as multiple stresses present in a single image. Thus exclusive utilization of visual classification can be insufficient [5].

The ability to generate more general/broader labels for confusing or novel cases can still be useful for treatment consideration. For example, consider an image where a fungal stress and an insect-related stress are present. Instead of committing to one specific stress, we could classify the example with the general label of 'biotic stress'. Along with each prediction, some notion of confidence would also be helpful. Whether it be a farmer needing a medium-level of confidence to aid in determining if they should spray a fungicide or an insurance company requiring a high-level of confidence to help determine if they should send an adjuster to a field to assess crop damage, having this confidence association could greatly increase the acceptance and usability of deep learning models in agriculture.

We apply the confidence-based hierarchical approach of [8] in a collaborative effort between domain experts in computer science and agricultural engineering to address these needs. Given a trained base classifier, the approach starts with an initial prediction from the base classifier and analyzes it in a hierarchical manner using posterior probabilities to meet a user-defined threshold of confidence, generalizing the label as needed. We anticipate this approach to be useful in addressing current desires for generalized labels along with associated confidence values.

## 2. Related Work

Multiple techniques have been proposed for reasoning and classifying with hierarchical representations to increase performance, efficiency, or accuracy. However very few have been specifically applied towards the agricultural domain. In [10], a visual tree is constructed using a hierarchical clustering algorithm on visual features (texture, color, shape, and structure) extracted using SIFT features into visual bag-of-words, color histograms, color moments, and histograms of curvatures. Then SVM classifiers for every label in the visual tree are jointly learned. In [25], a small hierarchy is enforced in the network architecture where they first have a series of shared convolutional and pooling layers then multiple branches following the same convolution, pool, and fully-connected layer architecture. One of the branches serves as a selector for predicting the species of the plant (apple, tomato, etc.). Based off the plant species prediction, a stress is predicted from that species' branch. In both [10] and [25], their final output will always be a flat classification (a terminal class label in their hierarchy) and no confidence is provided with the output.

Though multiple techniques exist for performing hierarchical classification, we chose to adopt the approach of [8] for its ability to generalize output labels with a confidence guarantee on every prediction. In [8], a posterior probability is separately computed for each node/label in the tree conditioned on its (uncalibrated) softmax value, where an equivalent softmax value for a non-terminal label is computed as the sum of softmax values from its terminal descendants. The posterior for each label is modeled non-parametrically using a normalized histogram. Inference is conducted in a bottom-up fashion using the argmax-selected label (a terminal label in the hierarchy) from the base classifier until meeting a confidence threshold.

These exist alternatives, such as [9], where the output logit for each terminal node/label in a hierarchy is computed by its respective SVM classifier (one SVM classifier per terminal label) and posterior probabilities are estimated using Platt scaling [34]. Non-terminal posteriors are computed by summing the descendent terminal label posteriors. Their approach is formulated on the maximization of a reward function given a specified overall accuracy on the validation set. Their final predictions are selected from the node/label in the hierarchy that yields the maximum expected reward. However, their approach has the possibility of "flipping" initially correct predictions to incorrect labels (labels that do not lie on the path from the ground truth label to root node) while [8] does not have this issue. Additionally, in [9], meeting a confidence threshold is guaranteed only on the validation set while [8] is guaranteed to always meet the confidence threshold on any test input. It is for these reasons that we are applying the approach of [8] and will conduct our analysis using this approach. In addition to our analysis experiments, we will compare the approaches of [8] and [9].

To the best of our knowledge, this is the first time a hierarchical classification approach with confidence guarantees and the ability to classify an example using a non-terminal label has been applied to the agriculture domain.

## 3. Hierarchical Classification with Confidence

In this work, we apply the hierarchical classification approach presented in [8]. Given a pre-trained base classifier, this method consists of an estimation procedure for computing the posterior distributions at every terminal and non-terminal label in a hierarchy and an inference procedure for determining the final confident prediction label for some test example. It should be noted that they used a validation set to model the posteriors as to not overfit to the training data.

### 3.1. Estimation

For a validation example $x$ with ground truth label $l$, the base classifier softmax value $s$ corresponding to $l$ is extracted then quantized into index $s_q$. This index is used to increment the *positive* histogram bin $H_l^+[s_q]$. Then for each ancestor label $a$ of $l$, the softmax value for $a$ is found by summing the softmax values of all terminal descendant labels of $a$. This aggregated softmax value is then quantized into $s_q$ and used to increment the *positive* histogram bin $H_a^+[s_q]$.

*Negative* histograms are then computed for each label $d$ that is not $l$ or an ancestor of $l$. Still using the same validation example $x$, the corresponding softmax values for $d$ are summed, quantized into $s_q$, and used to index and increment the *negative* histogram bin $H_d^-[s_q]$.

After all validation examples have been processed, the positive and negative likelihood distributions for each node are computed by L1-normalizing their respective histograms using

$$P(s_q|l) = H_l^+ / ||H_l^+||_1 \quad (1)$$

$$P(s_q|\neg l) = H_l^- / ||H_l^-||_1 \quad (2)$$

Priors $P(l)$ and $P(\neg l)$ are computed for each node/label in the hierarchy and can be used to incorporate domain knowledge (as will be discussed later). For our experiments, we used equal priors for the terminal nodes/labels.

Finally, Bayes' Rule (with a two-class context) is used to compute the posterior probability distribution $P(l|s_q)$ for label $l$ from its corresponding prior and likelihood distributions

$$P(l|s_q) = \frac{P(s_q|l)P(l)}{P(s_q|l)P(l) + P(s_q|\neg l)P(\neg l)} \quad (3)$$

### 3.2. Inference

The procedure begins with estimating the posterior probability of the base classifier's initial argmax-selected label $l$ (a terminal label in the hierarchy) using its corresponding softmax value $s$. The softmax value is quantized into $s_q$ and indexed into the posterior distribution $P(l|s_q)$. If the retrieved posterior value is below the confidence threshold $T$, the immediate parent of $l$ is examined. The parent's softmax value is computed by summing the softmax values of its terminal descendants, then the parent label's aggregated softmax value is quantized and used to index into its posterior distribution. This process continues up the ancestral path until either a sufficiently confident label is found (confidence meeting $T$) or the root node label of the hierarchy is reached (having 100% confidence).

## 4. Experiments

In this section, we examined the approach of [8] on three crop stress datasets. We will describe each dataset used in our experiments and their respective base classifier and hierarchy. Then we will describe each of the metrics used in our evaluation and finally present a series of experiments and their results.

### 4.1. Datasets & Base Classifiers

We evaluate the approach on Tomato, Corn, and Soybean datasets that contain healthy and stressed examples of plant leaves. The Tomato dataset is a subset of the PlantVillage dataset [20] while the Corn and Soybean datasets were recently collected by the agricultural engineering collaborators on this paper. The corn and soybean imagery were collected in stressed crop fields and were verified by our institution's agricultural extension educators and plant pathologists to guarantee accuracy. Example images from all three datasets are shown in Fig. 2 and the details of each dataset will be described below.

The **Tomato dataset** consists of (256x256) RGB images of tomato plant leaves in a controlled environment, with a single leaf centered on a constant background (either a table surface or black). It consists of 10 classes (9 stresses and healthy) and 15,892 total images (split into 11,204 train, 3,092 validation, and 1,596 test images).

For this dataset we trained a simple CNN with 3 convolution layers (8, 16, and 16 channels) and one fully-connected layer. We used a LeakyReLU activation function (to prevent any possibility of stagnant gradients) followed by a 3x3 max-pooling layer after each convolution layer and a global average-pooling layer before the fully-connected layer. This network was trained for 30 epochs and the networks weights from the training epoch that produced the lowest cross entropy loss value on the validation set were selected. This resulted in a class-balanced accuracy of 82.1% (computing the accuracy for each class then averaging). We note a larger CNN trained to a better accuracy level could also be used here. We chose to employ a smaller CNN for this dataset to emphasize the benefits of the hierarchical inference procedure.

The **Corn dataset** consists of RGB images of corn plant leaves in a natural setting, a corn field with a noisy background, captured with an 18MP camera at ground level (re-

sulting in images of size 3456x5184 or 5184x3456). In each instance, the area of interest (the plant stress the instance is annotated as) is at the center of the image. The dataset consists of 11 classes (10 stresses and healthy) and 8,911 total images (split into 6,232 train, 1,793 validation, and 886 test images). It should be noted that one of the stress class labels is actually a combination of two stresses (which will be discussed in Sect. 4.2).

For this dataset we trained a modified ResNet-18 network [19] for 30 epochs with final network weights selected as before. The network was adapted from PyTorch's [33] implementation from GitHub with all ReLU activations switched in favor of LeakyReLU activations. This resulted in a network with a class-balanced accuracy of 68.8%.

The **Soybean dataset** consists of RGB images of soybean plant leaves in a natural setting, a soybean field with a noisy background, captured with an 18MP camera at ground level (resulting in images of the same size as corn). Again, the area of interest (the plant stress the instance is annotated as) is at the center of the image. The dataset consists of 6 classes (5 stresses and healthy) and 6,635 total images (split into 4,642 train, 1,335 validation, and 660 test images). As with the Corn dataset, one of the stress class labels in this dataset is also a combination of two stresses (which will also be discussed later). For this dataset we trained a similar modified ResNet-18 network [19] which resulted in a class-balanced accuracy of 80.0%.

Since the plant leaf of interest is centered in every instance within the Corn and Soybean datasets, each image was center cropped (to be square) then resized to 345x345 (one-tenth resolution). Tomato images faced no cropping or resizing operations.

All base classifiers were implemented using Python 3.7 and PyTorch 1.2 and trained on a single NVIDIA Titan GPU using stochastic gradient descent with momentum (0.9) and a step learning rate scheduler (initially 0.001 then multiply by 0.1 every 5 epochs). All three datasets faced similar data augmentation schemes during training: random horizontal flip, random vertical flip, random rotation at 90 degree increments, gamma jitter, and brightness scale. In testing, the images were only cropped and resized (if necessary).

All three of our datasets face varying degrees of class imbalance. To address this, we implemented a simple replication scheme during training to balance all classes. Every batch is guaranteed to have the same number of examples from each class and each example from the same class is guaranteed to be different from the others in the batch. We used batch sizes of 20, 11, and 6 for Tomato, Corn, and Soybean, respectively. In our experiments we trained on a balanced dataset but tested on an imbalanced dataset.

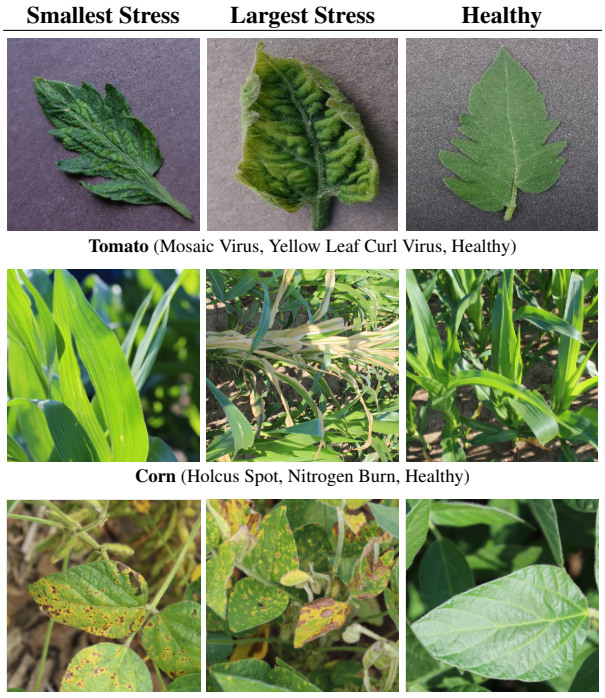


Figure 2: Example images from each of the datasets. Smallest and largest stress correspond to the stress class with the least and most examples, respectively. Corn and soybean images are center-cropped to make square.

## 4.2. Plant Stress Relational Trees

A plant stress relational tree will be employed for each dataset to define the relationships between terminal label plant stresses and more generalized stress categories. The root node of each tree is given the 'Unknown' label. Each tree is constructed using agricultural domain knowledge based on existing diagnostic and management practices to govern the relations. Although we construct a tree for tomato, corn, and soybean production systems, trees can be tailored to fit any food production system (the general practices utilized in planting, maintaining, and harvesting food crops). Our plant stress relational trees are designed to be application-focused where the stresses are grouped according to basic management strategies and testing practices. Each level of the tree aids the end user, such as a farmer, in making some decision regarding additional diagnostic tests or management actions. Information provided by the levels gets increasingly generic moving from the terminal nodes in the hierarchy to the root. While our plant stress relational trees are application-focused, these trees could also be purely taxonomic, phylogenetic, or other depending on the context. Our plant stress relational trees are shown in Figs. 3-5. The terminal classes are shown in bold at the bottom with the other labels being the non-terminal generalized plant stress categories.

In the Tomato dataset, the stress categories 'Hemi-

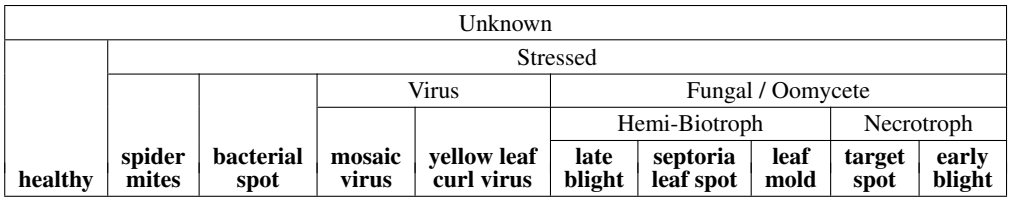


Figure 3: Tomato stress relational tree.

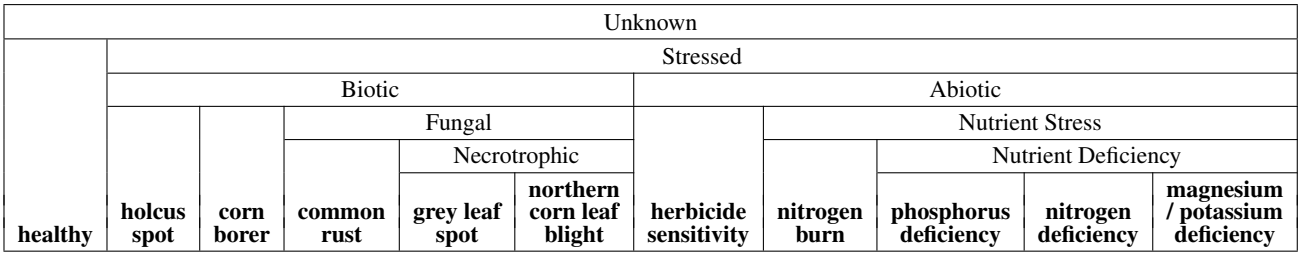


Figure 4: Corn stress relational tree.

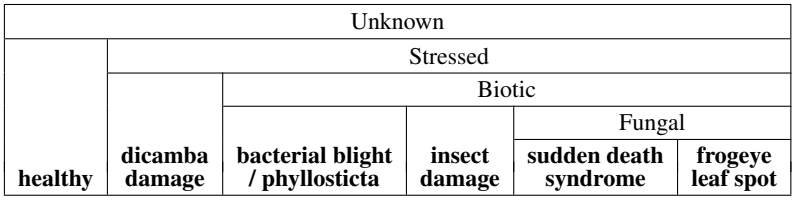


Figure 5: Soybean stress relational tree.

Biotroph' and 'Necrotroph' are often difficult to differentiate visually. Thus we joined these two non-terminal labels to the parent label 'Fungal / Oomycete'. We included this super-class because, although a completely descriptive stress label would not be provided, the user would still know all tests and treatments that are associated with 'Virus' and not 'Fungal / Oomycete' could be removed from consideration and further analysis and testing would be required to diagnose the specific stress.

In the Corn dataset, the 'magnesium/potassium deficiency' class contains images of just magnesium deficiency, images of just potassium deficiency, and images containing both. We included this joint terminal class for evaluating the approaches' ability to address the issue of classifying imagery with multiple stress symptoms.

In the Soybean dataset, the bacterial blight and phyllosticta stresses are indistinguishable using just their visual symptoms. Even the foremost expert in soybeans pathology was unable to provide a definite classification using only images. Thus these two stresses were combined to 'bacterial blight / phyllosticta'.

### 4.3. Metrics

To evaluate the approach, we included metrics from [8] and [9]. These metrics give credit for predictions that lie on the correct ancestral path from the ground truth up to, and including, the root [9] (no partial credit is given to predictions that are off the upward path from the ground truth). Some of the metrics are based on the sets of originally *correct* (C) and originally *incorrect* (IC) base classifier predictions (given the ground truth and argmax of the logits for test examples). We report the following:

- **C-Persist** [8] is the fraction of initially *correct* predictions of the base classifier that remain at the terminal level in the hierarchy.
- **C-Withdrawn** [8] is the fraction initially *correct* predictions of the base classifier that are assigned to the root ('Unknown').
- **C-Soften** [8] is the fraction of initially *correct* predictions of the base classifier that are generalized to a non-root and non-terminal label.
- **IC-Remain** [8] is the fraction of initially *incorrect* predictions of the base classifier that remain at an incorrect non-root label.
- **IC-Withdrawn** [8] is the fraction of initially *incorrect* predictions of the base classifier that are assigned to the root ('Unknown').
- **IC-Reform** [8] is the fraction of initially *incorrect* predictions of the base classifier that are generalized to a correct, non-root label.

- **avg-sIG** corresponds to the depth of the generalizations in terms of Information Gain (IG), as similarly used in [9]. The scaled IG (sIG) for a correct prediction at node $\mathcal{N}_i$ is

$$\text{sIG}(\mathcal{N}_i) = (\log_2|\mathcal{T}| - (\log_2(|\downarrow(\mathcal{N}_i)|)) \;/\; \log_2|\mathcal{T}| \quad (4)$$

  where $\mathcal{T}$ is the set of all terminal labels in the hierarchy and $\downarrow(\mathcal{N}_i)$ is the set of all terminal labels that are descendants of $\mathcal{N}_i$. If $\mathcal{N}_i$ is a terminal node: $\downarrow(\mathcal{N}_i) = \mathcal{N}_i$. Therefore when a correct prediction is at the terminal level (most precise), the gain is sIG = $(\log_2|\mathcal{T}| - (\log_2 1)) \;/\; \log_2|\mathcal{T}|$ = 1. When a prediction is withdrawn to the root ('Unknown'), the gain is sIG = $(\log_2|\mathcal{T}| - \log_2|\mathcal{T}|) \;/\; \log_2|\mathcal{T}|$ = 0. The sIG is 0 by default for any incorrect prediction. We compute the average sIG across all test examples to get **avg-sIG**.
- **Accuracy** is the fraction of post-inference classification results that are correct, where any label on the path from the ground-truth to the root node is considered a correct label (as used in [9]).

To address data imbalance in the test set, we modify the metrics to become class-balanced by computing the metric for each class then averaging all classes.

### 4.4. Evaluation

Results at 50%, 80%, 85%, 90%, and 95% confidence thresholds for the three datasets are provided in Tables 1-4. We also provide the performance of the base classifiers (with no hierarchical inference).

Evaluation results for the Tomato dataset are shown in Table 1. We can see that at 50% confidence, 19% of initially incorrect predictions have already been reformed to a correct non-root label and a large majority (91%) of initially correct predictions remained at the terminal level. Large improvements are seen moving up to 80% confidence with 67% of initially incorrect predictions being reformed to a valid label and 71% of initially correct predictions remaining at the terminal level. We also see a class-balanced accuracy improvement of 8.6% over the score for 50% confidence. For all confidence levels, almost all predictions (class-balanced) remained at the non-root labels.

Results for the Corn dataset are shown in Table 2. At 50% confidence, 89% of initially correct predictions still maintained their terminal status while 21% of initially incorrect predictions have already been reformed to a correct non-root label. With an 80% confidence threshold, 70% of initially correct predictions remained at the terminal level with 24% at a softened, non-root label. Additionally, only 39% of initially incorrect predictions remained with 21% being withdrawn and 40% being reformed to a correct non-root label. Even at higher confidences, the IC-Withdrawn score does not increase much while IC-Reform continues to increase. At 80% confidence, we also see that the overall class-balanced accuracy has increased by almost 20% from the base classifier.

| | Tomato | | | | | |
|---|---|---|---|---|---|---|
| | Base | 50% | 80% | 85% | 90% | 95% |
| C-Persist | 1.0 | .91 | .71 | .66 | .60 | .44 |
| C-Withdrawn | - | .00 | .01 | .02 | .02 | .03 |
| C-Soften | - | .09 | .28 | .32 | .38 | .53 |
| IC-Remain | 1.0 | .81 | .29 | .23 | .22 | .11 |
| IC-Withdrawn | - | .00 | .05 | .05 | .05 | .08 |
| IC-Reform | - | .19 | .67 | .71 | .73 | .80 |
| avg-sIG | - | .78 | .65 | .61 | .58 | .49 |
| % Valid (¬root) | 100 | 99.8 | 98.6 | 97.6 | 97.5 | 96.6 |
| Accuracy | 82.1 | 86.0 | 94.6 | 95.8 | 96.1 | 98.2 |

Table 1: Tomato class-balanced hierarchical results.

| | Corn | | | | | |
|---|---|---|---|---|---|---|
| | Base | 50% | 80% | 85% | 90% | 95% |
| C-Persist | 1.00 | .89 | .70 | .49 | .46 | .41 |
| C-Withdrawn | - | .01 | .06 | .13 | .13 | .13 |
| C-Soften | - | .10 | .24 | .37 | .41 | .46 |
| IC-Remain | 1.00 | .77 | .39 | .27 | .27 | .13 |
| IC-Withdrawn | - | .02 | .21 | .26 | .26 | .26 |
| IC-Reform | - | .21 | .40 | .48 | .48 | .61 |
| avg-sIG | - | .65 | .56 | .45 | .44 | .40 |
| % Valid (¬root) | 100 | 98.6 | 87.9 | 80.9 | 80.9 | 80.9 |
| Accuracy | 68.8 | 74.7 | 87.7 | 92.4 | 92.4 | 95.8 |

Table 2: Corn class-balanced hierarchical results.

| | Soybean | | | | | |
|---|---|---|---|---|---|---|
| | Base | 50% | 80% | 85% | 90% | 95% |
| C-Persist | 1.00 | .92 | .87 | .87 | .76 | .33 |
| C-Withdrawn | - | .04 | .05 | .07 | .18 | .18 |
| C-Soften | - | .05 | .07 | .06 | .06 | .49 |
| IC-Remain | 1.00 | .59 | .38 | .34 | .28 | .13 |
| IC-Withdrawn | - | .09 | .20 | .23 | .29 | .29 |
| IC-Reform | - | .32 | .43 | .43 | .43 | .58 |
| avg-sIG | - | .76 | .72 | .72 | .63 | .40 |
| % Valid (¬root) | 100 | 96.3 | 93.0 | 91.9 | 82.1 | 82.1 |
| Accuracy | 80.0 | 81.5 | 84.5 | 85.1 | 85.5 | 98.7 |

Table 3: Soybean class-balanced hierarchical results.

We see particularly interesting results for the Soybean dataset as shown in Table 3. This classifier tends to be highly confident in all of its initially correct predictions, having more predictions remain at the terminal level at 90% confidence than the Tomato and Corn datasets. We also see 43% of initially incorrect predictions being reformed to correct non-root labels and an avg-sIG score of 0.63 which means the predictions are tending to reside deeper in the tree (potentially around the 'Biotic' label). We also see the class-balanced accuracy increases by about 13% from 90% confidence to 95% confidence.

An explanation for the high C-Persist values for soy-

beans could be that four of the classes achieved a base classifier individual class accuracy of over 90%. Thus for these four classes, the network was able to identify discriminant features that made it confident in its initial prediction. This is validated further by the lower C-Withdrawn values. The predictions are highly confident without being completely withdrawn to the root node. These high initial confidences could also explain the accuracy increase from 90% to 95% confidence. The Soybean dataset also contained one particularly difficult class ('frogeye leaf spot') that only achieved 12% base classifier class accuracy. Such low class accuracy could be attributed to the small lesions as symptoms of the stress, which would be difficult for a CNN to recognize.

Overall, as the confidence threshold increases, we see more initially correct predictions generalized from their terminal label and more initially incorrect predictions reformed to correct labels. Additionally, as the confidence threshold increases we naturally see a decrease in avg-sIG. We also see that the class-balanced accuracy increases as confidence increases. The inverse relationship between avg-sIG and class-balanced accuracy shows the accuracy-specificity trade-off made when selecting a confidence threshold.

We show some test predictions across the datasets at 90% confidence in Fig. 6 (note the corn and soybean images were square center cropped to better fit on the page). In the generalized examples, the base classifier actually predicted the correct labels, though they were deemed unreliable and generalized by the approach. The C-Withdrawn examples were also classified correctly by the base classifier, but were again not confident or not standard instances. The IC-Reform examples were incorrectly classified by the base classifier, but were generalized to a reasonable, and correct, label (in both cases the lowest common ancestor in the tree was selected). Lastly, the IC-Withdrawn images were incorrectly classified by the base classifier and subsequently generalized to the root node label ('Unknown').

Although being able to classify a plant as healthy versus 'Stressed' provides value for determining if action is needed, the greatest value comes when we can provide a more specific stress label or category. Thus we evaluate how well each dataset experiment maintains stresses at a "specified" label (i.e., a label that is a descendant of 'Stressed'). To evaluate this, we compute a specified stress metric (**SpecStress**) as the percentage of correctly predicted stresses $S_s$ that remain at a label that is a descendent of the general plant stress label ('Stressed'). We call any label that is a descendent of 'Stressed' a "specified stress" because it still provides useful, more narrowed information about what stress is present in the image. We report the results of this experiment in Table 4.

We can see across these experiments that there were more stresses at a "specified" label than generic 'Stressed'. This provides substantial value if we can successfully pinpoint an example to one category of stress. If successful, and the prediction is correct, we have the ability to rule out numerous tests or treatments because they are unnecessary for the provided label (this will be discussed further in Sect. 5). As expected, as the confidence threshold increases, more predictions are generalized further up the tree, pushing more predictions to the 'Stressed' label.

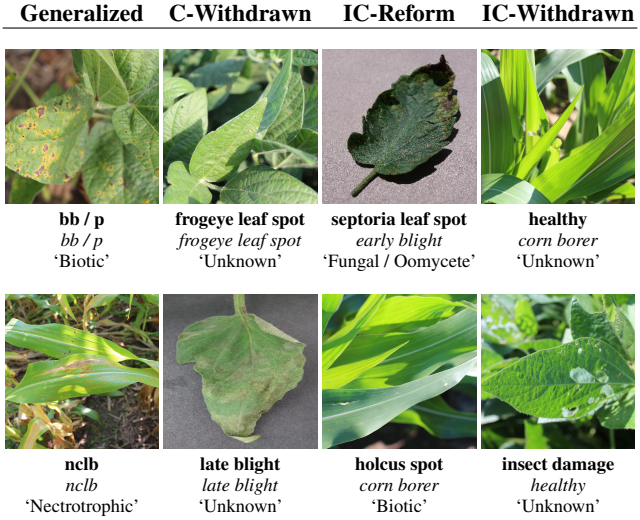


Figure 6: Example classification results at 90% confidence from the three datasets (**Ground truth** / *Base Classifier* / 'Final Label'). Labels bb/p and nclb correspond to 'bacterial blight / phyllosticta' and 'northern corn leaf blight', respectively.

| | Confidence | | | | | |
|---|---|---|---|---|---|---|
| Dataset: | Base | 50% | 80% | 85% | 90% | 95% |
| Tomato | 100 | 94.6 | 76.5 | 72.8 | 71.5 | 59.3 |
| Corn | 100 | 99.8 | 82.5 | 80.4 | 80.4 | 60.2 |
| Soybean | 100 | 99.8 | 98.5 | 98.5 | 98.5 | 98.5 |

Table 4: SpecStress values for all three datasets.

## 4.5. Hierarchical Comparison

We additionally compared to the related algorithm of [9] and report the results in Table 5 for the specified metrics at a 90% confidence threshold. For [9], the learned parameters (on the validation set) were $\lambda_{90\%} = 0.398,\ 0.578,\ 1.011$ for Tomato, Corn, and Soybean, respectively. All learned parameters were found using confidence $\epsilon = 90\%$ and $\tilde{\epsilon} = 0.001$.

For this comparison, we added two additional metrics. **C-Corrupt** is the fraction of initially *correct* predictions of the base classifier that are relabeled to an incorrect label off the ancestral path of ground truth. Lower proportions are desired. Note C-Corrupt will always be 0 for [8], but not [9]. The average of the final posterior values of test predictions

| | Tomato | | Corn | | Soybean | |
|---|---|---|---|---|---|---|
| | [8] | [9] | [8] | [9] | [8] | [9] |
| C-Persist | .60 | .52 | .46 | .65 | .76 | .75 |
| C-Withdrawn | .02 | .01 | .13 | .03 | .18 | .02 |
| C-Soften | .38 | .47 | .41 | .32 | .06 | .24 |
| C-Corrupt | .00 | .00 | .00 | .01 | .00 | .00 |
| IC-Remain | .22 | .48 | .27 | .71 | .28 | .42 |
| IC-Withdrawn | .05 | .02 | .26 | .03 | .29 | .10 |
| IC-Reform | .73 | .50 | .48 | .26 | .43 | .48 |
| avg-sIG | .58 | .52 | .44 | .55 | .63 | .67 |
| avg-Post | .98 | .66 | .97 | .73 | .96 | .89 |
| % Valid (¬root) | 97.5 | 98.9 | 80.9 | 97.6 | 82.1 | 97.3 |
| Accuracy | 96.1 | 89.4 | 92.4 | 71.7 | 85.5 | 93.4 |

Table 5: Comparing the approaches [8] (our approach) and [9] at 90% confidence.

(**avg-Post**) are also presented. A value that meets the given confidence threshold, but does not exceed the threshold by a large margin, is desired (i.e., lower C-Withdrawn).

We see in 5 that [8] will correct significantly more initially incorrect predictions than [9]. The main benefit of [8] is that there is an actual confidence guarantee for every test prediction. This is reflected in the avg-Post scores where the values for [9] tend to fall substantially below the confidence threshold of 90%. The approach of [9] provides no such guarantee on any test prediction, making the approach unreliable for our purposes. This also implies that many of the predictions are being forced to remain deeper in the tree, valuing specificity at the trade-off of accuracy. Unlike [8], the approach of [9] introduces C-Corrupt errors. Although the C-Corrupt for [9] in the Tomato dataset is reported as .00, there are still initially correct predictions that were "flipped" to incorrect labels (very small fraction). Noticeable corruption errors were incurred in the Corn dataset and no corruption errors were introduced in the Soybean dataset. To reiterate, the approach of [8] will only consider ancestors of the base predictions for the final classification label. This prevents the possibility of changing any originally correct (but unconfident) prediction to an incorrect one, which could happen once we begin to consider non-ancestral nodes as possible options. While considering non-ancestral nodes could reduce the IC-Withdrawn value (and potentially increase the IC-Reform value), the possibility of a non-zero C-Corrupt value is unacceptable as this immediately results in the loss of confidence of the end user (*e.g.*, farmer).

## 5. Discussion & Relevance

A contribution of the proposed approach is the ability to output the root node label ('Unknown') or general stress category label ('Stressed'). Although non-descriptive, these "worst case" output labels still provide the user (*e.g.*, a producer/farmer or farm manager) with useful information. The stress in the image could be one not seen in that area before and therefore not included in the training set or a plant resistance reaction that has yet to be documented. Regardless of the reason for withdrawing a prediction to the root node or 'Stressed', the user knows that properly classifying this stress will most likely require additional expertise or testing.

Another contribution of this approach is having the capability to classify an example at any softened (non-terminal and non-root) label. Any time an example is softened, we are still able to exclude numerous tests and treatments from consideration on how to properly assess and manage a stress. This is best demonstrated in the context of the corn stress relational tree. If an example was labeled as 'Nutrient Deficiency', the user (*e.g.*, a certified crop advisor) is able to narrow their focus on activities that directly diagnose nutrient deficiencies such as soil testing and not expend time or resources on other diagnostic approaches such as those that would be used for biotic stresses. This narrowing of possibilities allows for saving time, money, and effort on effectively treating a plant stress.

As briefly mentioned in Sect. 3, one method we have introduced for incorporating domain knowledge into the approach is by allowing the ability to hand-tune class priors for individual stresses. An agricultural engineering expert is best able to assess using additional information such as epidemiological models, weather data, and cultivation practices to determine the likely stresses to appear. Given a set of visually similar stresses, an expert can tune the priors to place emphasis on a stress that is more likely to exist at a given time. Thus external knowledge can be directly injected into the approach via the use of the priors, providing an advantage over existing approaches.

## 6. Conclusion

In this work, we presented an approach in the agricultural domain to hierarchically classify various crop stresses with a specified confidence level. Shortcomings in previous agricultural applications with CNNs included being unable to 1) generalize prediction labels and 2) make predictions with an associated confidence. Therefore we applied the Bayesian approach of [8] that models posteriors at every node/label in a hierarchy. Inference starts from an initial hypothesis and generalizes the label until meeting a particular confidence. We evaluated the approach on Tomato, Corn, and Soybean plant stress datasets at various confidence thresholds. Results showed this method provides a wide range of generalized labels and corrects many of the initially incorrect predictions, providing useful information to properly diagnose a stress, which can lead to a quick and effective treatment. We believe this approach has potential in future agricultural systems, such as being deployed on drones for efficient and effective surveillance and analysis of crop fields.

## References


[1] T. W. Allen, C. A. Bradley, A. J. Sisson, E. Byamukama, M. I. Chilvers, C. M. Coker, A. A. Collins, J. P. Damicone, A. E. Dorrance, N. S. Dufault, et al. Soybean yield loss estimates due to diseases in the United States and Ontario, Canada, from 2010 to 2014. *Plant Health Progress*, 18(1):19–27, 2017.

[2] H. A. Atabay. Deep residual learning for tomato plant leaf disease identification. *Journal of Theoretical & Applied Information Technology*, 95(24), 2017.

[3] R. Balodi, S. Bisht, A. Ghatak, and K. H. Rao. Plant disease diagnosis: Technological advancements and challenges. *Indian Phytopathology*, 70(3):275–281, 2017.

[4] J. G. A. Barbedo. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems engineering*, 144:52–60, 2016.

[5] J. G. A. Barbedo. Factors influencing the use of deep learning for plant disease recognition. *Biosystems engineering*, 172:84–91, 2018.

[6] J. G. A. Barbedo. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153:46–53, Aug 2018.

[7] J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles. Convolutional neural networks for the automatic identification of plant diseases. *Frontiers in plant science*, 10, 2019.

[8] J. Davis, T. Liang, J. Enouen, and R. Ilin. Hierarchical semantic labeling with adaptive confidence. In *International Symposium on Visual Computing*, pages 169–183. Springer, 2019.

[9] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3450–3457. IEEE, 2012.

[10] J. Fan, N. Zhou, J. Peng, and L. Gao. Hierarchical learning of tree classifiers for large-scale plant species identification. *IEEE Transactions on Image Processing*, 24(11):4172–4184, 2015.

[11] K. P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[12] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9):2022, 2017.

[13] A. F. Fuentes, S. Yoon, J. Lee, and D. S. Park. High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Frontiers in plant science*, 9:1162, 2018.

[14] S. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar. An explainable deep machine vision framework for plant stress phenotyping. *Proceedings of the National Academy of Sciences*, 115(18):4613–4618, 2018.

[15] N. K. Gogoi, B. Deka, and L. C. Bora. Remote sensing and its use in detection and monitoring plant diseases: A review. *Agricultural Reviews*, 39(4):307–313, 2018.

[16] K. Golhani, S. K. Balasundram, G. Vadamalai, and B. Pradhan. A review of neural networks in plant disease detection using hyperspectral data. *Information Processing in Agriculture*, 5(3):354–371, 2018.

[17] J. L. Green, J. Capizzi, and O. Maloy. A systematic approach to diagnosing plant damage. *Cooperative Extension Service, Oregon State University, Corvallis, Oregon. Ornamentals Northwest Newsletter*, 13(6):1–32, 1990.

[18] G. L. Hartman, J. C. Rupe, E. J. Sikora, L. L. Domier, Jeff A Davis, and Kevin Lloyd Steffey. *Compendium of soybean diseases and pests*. Am Phytopath Society, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] David Hughes, Marcel Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015.

[21] A. Kamilaris and F. X. Prenafeta-Boldú. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3):312–322, 2018.

[22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[23] A.-K. Mahlein. Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. *Plant disease*, 100(2):241–251, 2016.

[24] A.-K. Mahlein. Present and future trends in plant disease detection. *Plant Disease*, 100(2):1–11, 2016.

[25] Z. Mao, J. Chen, and M. Yang. Multi-branch structure for hierarchical classification in plant disease recognition. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 528–538. Springer, 2019.

[26] F. Martinelli, R. Scalenghe, S. Davino, S. Panno, G. Scuderi, P. Ruisi, P. Villa, D. Stroppiana, M. Boschetti, L. R. Goulart, et al. Advanced methods of plant disease detection. a review. *Agronomy for Sustainable Development*, 35(1):1–25, 2015.

[27] S. A. Miller, F. D. Beed, and C. L. Harmon. Plant disease diagnostic capabilities and networks. *Annual review of phytopathology*, 47:15–38, 2009.

[28] S. P. Mohanty, D. P. Hughes, and M. Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[29] D. S. Mueller, K. A. Wise, A. J. Sisson, T. W. Allen, G. C. Bergstrom, D. B. Bosley, C. A. Bradley, K. D. Broders, E. Byamukama, M. I. Chilvers, et al. Corn yield loss estimates due to diseases in the United States and Ontario, Canada from 2012 to 2015. *Plant health progress*, 17(3):211–222, 2016.

[30] G. P. Munkvold, Donald G. White, et al. *Compendium of corn diseases*. Am Phytopath Society, 2016.

[31] D. Myers, C. M. Ross, and B. Liu. A review of unmanned aircraft system (UAS) applications for agriculture. In *2015 ASABE Annual International Meeting*, page 1. American Society of Agricultural and Biological Engineers, 2015.

[32] E.-C. Oerke. Crop losses to pests. *The Journal of Agricultural Science*, 144(1):31–43, 2006.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[34] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[35] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes. Deep learning for image-based cassava disease detection. *Frontiers in plant science*, 8:1852, 2017.

[36] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis. A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, 72(1):1–13, 2010.

[37] E. C. Tetila, B. B. Machado, N. A. de Souza Belete, D. A. Guimarães, and H. Pistori. Identification of soybean foliar diseases using unmanned aerial vehicle images. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2190–2194, 2017.

[38] Y. Toda, F. Okura, et al. How convolutional neural networks diagnose plant disease. *Plant Phenomics*, 2019:9237136, 2019.

[39] M. Vilà, P. E. Hulme, et al. *Impact of biological invasions on ecosystem services*, volume 12. Springer.

[40] G. Wang, Y. Sun, and H. Wang. Automatic image-based plant disease severity estimation using deep learning. *Computational intelligence and neuroscience*, 2017, 2017.

[41] J. S. West, C. Bravo, R. Oberti, D. Lemaire, D. Moshou, and H. A. McCartney. The potential of optical canopy measurement for targeted control of field crop diseases. *Annual review of Phytopathology*, 41(1):593–614, 2003.