

Soybean disease identification using original field images and transfer learning with convolutional neural networks

Noah Bevers ^{*}, Edward J. Sikora, Nate B. Hardy

Department of Entomology and Plant Pathology, Auburn University 301 Funchess Hall Auburn University, AL 36849-5413, USA



ARTICLE INFO

Keywords:
Convolution
Transfer learning
Augmentation
Soybean
Disease diagnostics
Image classification

ABSTRACT

Meeting the growing demand for soybeans will require increased production. One approach would be to reduce yield loss from plant diseases. In the U.S., soybean diseases account for approximately 8–25% of average annual yield loss. Early and accurate detection of pathogens is key for effective disease management strategies and can help to minimize pesticide usage and thus boost overall productivity. Recent advancements in computer vision could move us towards that goal by making disease diagnostics expertise more readily accessible to everyone. To that end, we developed an automated classifier of digital images of soybean diseases, based on convolutional neural networks (CNN). For model training and validation, we acquired more than 9,500 original soybean images, representing eight distinct disease and deficiency classes: (1) healthy/asymptomatic, (2) bacterial blight, (3) Cercospora leaf blight, (4) downy mildew, (5) frogeye leaf spot, (6) soybean rust, (7) target spot, and (8) potassium deficiency. To make training more efficient we experimented with a variety of approaches to transfer learning, data engineering, and data augmentation. Our best performing model was based on the DenseNet201 architecture. After training from scratch, it achieved an overall testing accuracy of 96.8%. Experimenting with full or partial freezing of core DenseNet201 model weights did not improve performance. Neither did a deliberate effort to increase the diversity of subject backgrounds in the digital images. Models performed best when trained on datasets composed exclusively of images of soybean leaves still attached to the plant in the field; conversely, mixing in images of detached leaves on simple backgrounds reduced performance. On the other hand, data augmentation to increase representational parity across disease classes provided a substantial performance boost. Our development experience may provide useful insights for researchers considering how to best build and analyze datasets for similar applications.

1. Introduction

Over the last 60 years, soybean (*Glycine max* (L) Merrill) has become the world's premier oilseed crop and one of the five most important food crops worldwide (Savary et al., 2019). The land area devoted to soybean production has increased by 2.7 % per year since 1964; a growth rate higher than that for any other major crop (Hartman et al., 2011; Kalaitzandonakes et al., 2019). Soybeans are usually processed into meal or oil (Ali, 2010). The meal is high in protein and is used primarily as feed for livestock, in particular, poultry and swine; it is also commonly used in aquacultural production systems (Hartman et al., 2011). Soybean oil is mostly used for human consumption, but it is also used for a variety of industrial applications including the formulation of inks, paints, detergents, and lubricants (Kalaitzandonakes et al., 2019). Use in biodiesel production may also soon be significant

(Kalaitzandonakes et al., 2019). But the most likely increases in demand will come from the challenge of keeping growing human populations fed. Meeting these future demands for soybean will require creative solutions to current production limitations. Here, we focus on production losses caused by pathogens.

Across the board, plant pathogens cause hundreds of billions of dollars in crop losses per year (Fried et al., 2017). Soybean farms are no exception; in the United States (Hartman et al., 2016), and abroad (Kalaitzandonakes et al., 2019; Oerke, 2006; Wrather et al., 2010), soybean diseases account for approximately 8–25 % of the average annual yield loss (Bradley et al., 2020; Hartman et al., 2016; Savary et al., 2019). Across the USA, an estimated \$45 USD per acre is lost annually due to the presence of 28 soybean diseases and pathogens; resulting in approximately \$3.9 billion USD in annual yield losses (Bradley et al., 2020). Added to these direct costs are the costs of

* Corresponding author.

E-mail addresses: nzb0054@auburn.edu (N. Bevers), sikorej@auburn.edu (E.J. Sikora), nbh0006@auburn.edu (N.B. Hardy).

pesticide application and concomitant environmental degradation. Moreover, the liberal application of pesticides hastens the evolution of resistance in pest populations, and thus threatens the long-term sustainability of soybean farming. For example, in several regions of the USA, heavy reliance on quinone outside inhibitor (QoI) fungicides for the control of frogeye leaf spot disease in soybean has led to the development of several QoI fungicide-resistant isolates of the disease's causal agent, *Cercospora sojina* (Zeng et al., 2015). This has rendered QoI fungicides useless in some areas. The cost of developing a new pesticide for regulated usage can be up to \$286 million USD (Kalaitzandonakes et al., 2019). Soybean farming would be more sustainable and better able to meet global food demands if soybean pesticide applications were more precise.

Early and accurate detection of pathogens is crucial for applying effective management strategies and for developing coordinated disease and resistance monitoring programs at local, regional, national, and global scales (Sikora et al., 2014). Early detection can also help to minimize pesticide usage which may ultimately slow the development of pesticide resistance. Many soybean diseases display progressive visual symptoms on the leaves of the plant such as discoloration, blotching, spotting, and lesion formation. Some of these diseases can be diagnosed by the trained eye of an experienced farmer, extension agent, or professional plant pathologist (Miller et al., 2009). But soybean producers may lack such expertise, and in some cases even an expert's eye is not enough. Thus, there is a need to improve the speed and accuracy of disease diagnostics. That need could be met with new technologies (Miller et al., 2009).

Recent advancements in deep learning and computer vision have the potential of democratizing disease diagnostics expertise and thereby increasing the precision and economy of pest control. In the last decade, a variety of computational modeling methods have been applied in attempts to automate crop disease diagnostics from digital images, including K-Nearest Neighbor classifiers (Shrivastava and Hooda, 2014), color transformation and histogram approaches (Garcia et al., 2015; Gui et al., 2015), support vector machines (Shrivastava and Singh, 2017), and more recently, deep learning with convolutional neural networks (CNNs) (Karlekar and Seal, 2020; Mohanty et al., 2016; Sagar and Dheeba, 2020; Sahu et al., 2021; Wu et al., 2019; Yu et al., 2022). CNN-based approaches have proven the most successful (Lecun et al., 2015).

Artificial neural networks are *meta*-models composed of many computational nodes, each of which is (or is akin to) a logistic regression model. Nodes are arranged in layers, and each node in a layer is connected to all nodes in each adjacent layer. More specifically, each node receives output signals from nodes in the upstream layer, integrates those inputs in a way that depends on activation weights and bias values assigned by the model, and then outputs a signal to nodes in the downstream layer. Such models can be trained to learn the rules – which can be complex and non-linear – for making certain predictions or classifications from data, using an algorithm known as backpropagation to optimize a set of activation weights for the connections between nodes in the network. All artificial neural networks have an input layer (which takes a data vector), an output layer (which gives predictions or classifications), and one or more hidden layers that map inputs to outputs (Lecun et al., 2015). CNNs are a type of artificial neural network in which there can be several distinct types of hidden layers, among which are special convolutional layers (Albawi et al., 2017). A convolutional layer applies a convolution to the outputs of the layer upstream, with convolution being the technical term for what is essentially a filter that accentuates certain data features and makes it easier for subsequent layers to extract their meaning. For example, a convolution might sharpen the vertical edges of a digital image, or increase the light–dark contrast across all image pixels (Albawi et al., 2017; Lecun et al., 2015). CNNs have been used for speech recognition, object segmentation, and visual object recognition and classification (Lecun et al., 2015). There have also been many applications in agriculture, for example, for weed identification and control (Du et al., 2021), detection of fraud in the sale

of ginger and turmeric powder (Jahanbakhshi et al., 2021a, 2021b), fruit detection and counting (Vasconez et al., 2020), and automatic sorting of carrot fruit for waste management (Jahanbakhshi et al., 2021c). In summary, CNNs are a type of artificial neural network that are good at solving problems that involve ‘seeing,’ that is parsing and interpreting a grid of light intensity values, and have great potential for solving problems in agricultural production (Kim, 2017).

Nevertheless, the use of CNNs has some drawbacks. In particular, the typical development process requires such large inputs of training data and computational resources, that it can be infeasible for some applications, and where it is possible, can have significant detrimental effects on the environment (Dhar, 2020; Henderson et al., 2020; Lacoste et al., 2019). Thus, increasing the efficiency of CNN development has emerged as a research priority.

So, here our goal was to develop a CNN-based automated classifier of digital images of soybean diseases. Moreover, we aimed to find ways to increase the efficiency of CNN model training, so as to expand its feasibility for niche applications in agriculture. In particular, we explored what kinds of training images are most useful, so that we could train well with a relatively small training datasets (Lacoste et al., 2019). To further boost training efficiency, we experimented with data-augmentation and transfer learning approaches. To be clear, this is not the first attempt to develop an automated classifier of images of soybean disease symptoms. Previous work has been based on publicly available image databases – in particular the Plant Village dataset – which are of limited scope (disease classes) and depth (images per disease class), with only a few hundred images available for model training across all soybean disease categories (Garcia et al., 2015; Hughes and Salathe, 2015; Karlekar and Seal, 2020; Shrivastava and Hooda, 2014). Furthermore, much of the available training data consists of low-resolution images with little variation in composition: most are of dis-articulated leaves on simple backgrounds (Barbedo, 2018). Hence, our first objective was to obtain new training data that better reflect the diversity and prevalence of soybean diseases in the southeastern US, and better reflect the types of images that farmers and agricultural inspectors are apt to obtain. A related objective was to systematically vary training image compositions (specifically image backgrounds) to gain insights into how to engineer training data for efficient training. Although our primary aim for this study was to optimize classification accuracy from a relatively small training data set, we also consider some previously-proposed approaches for reducing computational resource consumption (Strubell et al., 2020).

2. Materials and methods

2.1. Overview

Over the course of two seasons of soybean production, we collected over 9,500 original field images of soybean leaves in eight distinct disease categories. We explored the effects of varying image backgrounds, and we also experimented with data augmentation techniques. After training data sets were assembled, we explored the effects of varying model architectural parameters, and in particular we leveraged transfer learning methods (Shin et al., 2016). With transfer learning, models that have been trained on large data sets to solve one type of classification problem are adapted to a related type of classification problem (Tan et al., 2018). Such transferred models can apply the higher-level, more abstract lessons they have learned about seeing something in general to the problem of telling a few specific things apart. The upshot is that they work well when training data sets are relatively small. For the general problem of digital image classification, several high-performing models are available (Krizhevsky et al., 2012; Shin et al., 2016), all of which have been trained on the ImageNet dataset, which contains over fourteen million well-annotated images, representing over twenty thousand image classes (Fei-Fei et al., 2010; Krizhevsky et al., 2012). We explored the application of several of these high-performing image classifier

models, namely, VGG16 (Simonyan and Zisserman, 2015), ResNet50 (He et al., 2016), Xception (Chollet, 2017), DenseNet201 (Huang et al., 2017), and EfficientNetB0 (Tan and Le, 2019).

2.2. Data collection

Soybean leaves were photographed during the 2020 and 2021 Alabama (southeastern USA) growing seasons in the months of August, September, and October. Examples of images can be seen in Fig. 1. During the 2020 season, all photographs were captured using a Canon EOS 7D Mark II Digital SLR Camera at the EV Smith Agricultural Research Station in Tallahassee, Alabama, USA. The primary method was to walk through rows of planted soybean fields and capture images of a single soybean leaf showing symptoms of a known disease with the leaf still attached to the plant. The leaves photographed were simply the ones ‘stumbled upon’ in a known diseased area. We attempted to photograph leaves at various heights in the canopy in proportion to how disease symptoms were distributed. At the end of each such walk, 10–20 leaves of each represented disease category were detached from the plant and then immediately photographed while laid flat on the ground in trimmed grass or on a white surface (car hood).

The diseases photographed were limited to those that naturally occurred in the field. During 2020, we captured 4,181 images across five disease categories, namely, healthy (asymptomatic) plants, bacterial blight, Cercospora leaf blight, frogeye leaf spot, and soybean rust.

During the 2021 season, we captured 5,467 original images. To the five disease categories represented in the 2020 data set, we added two new diseases – downy mildew and target spot – along with a nutrient deficiency, namely potassium. All images of the potassium deficiency were taken at Cullars Rotation in Auburn, AL. Images of target spot were from Brewton Agricultural Research Unit in Brewton, AL. All other disease categories were from the EV Smith Research station. We used a slightly modified version of the data acquisition process we had used the year prior. For about 70 % of cases, the same Digital SLR camera was used, but for the other 30 % we used a Motorola Moto Z2 Play smartphone (Motorola, Inc. in Chicago, Illinois). This was done to better approximate the types of images that might be taken by users of the application in the field, and to add variety to the dataset which might

help the CNN distinguish between signal and noise. Another modification was to increase the proportion of images of detached leaves on varying backgrounds. Approximately 30 % of images were of a single detached leaf in 2021. About half of the detached leaves were photographed with the leaf laid flat on the ground in trimmed grass and the other half laid flat on a white table. We also deliberately varied light conditions, with a close to even mix of subjects in full sun and in shade.

Across both seasons we amassed a total of 9,648 original images collected across eight disease/deficiency categories. To these original images, we added an average of 80 images per disease category from extension resources and the publicly available “Image Database of Plant Disease Symptoms (PDDB)” (Barbedo et al., 2016). See Table 1 for count details and Fig. 1 for sample images.

2.3. Data preprocessing and augmentation

The dataset of images used for the training and testing of CNN models included 10,722 total images. A Python script was used to randomly apportion the data into a training set and a testing set. Specifically, ~85 % (9,113) of the images were used to train models, and ~15 % (1609) were used to test them. The testing set was further subdivided in a validation set of 533 images (33 % of the testing set), used to compare the performance of different model architectures, and an assessment set of 1076 images (66 % of the testing set) used to assess the accuracy of the optimized model. These ratios are fairly standard, and tend to strike a balance between model over-fitting and under-fitting (Karlekar and Seal, 2020).

CNN models tend to learn best when each classification category has roughly the same number of images (Li et al., 2018). Image augmentation is a common practice used to artificially increase the number of images in a data set by applying various transformation to original images, such as rotating, zooming, reflecting, adjusting brightness, and adding noise. In addition to increasing parity across classes, data augmentation can also be used to simply increase the overall volume of training data.

In our data set, bacterial blight was represented by roughly half as many images as the other categories. To increase parity, we augmented the bacterial blight set by horizontally reflecting each original image. On



Fig. 1. Sample Images. Examples of the photographed soybean leaves for all eight disease categories. Starting from left to right, top to bottom: bacterial blight, cercospora leaf blight, downy mildew, no disease, potassium deficiency, soybean rust, target spot. Note some cropping and size adjustment was used here for display purposes.

Table 1

Image Collection. Displays the total number of images for each disease category that was photographed for each season. Bacterial blight was the only category that used the addition of augmented images prior to model training. Total images used across both training and testing can be seen in the far-right column.

| Disease Category | 2020 Original Images | 2021 Original Images | Total Original Images | Additional Augmented Images | Extension Images | Images Used in Models |
|------------------------|----------------------|----------------------|-----------------------|-----------------------------|------------------|-----------------------|
| Bacterial Blight | 484 | 0 | 484 | 484 | 104 | 1072 |
| Cercospora Leaf Blight | 1163 | 435 | 1598 | 0 | 38 | 1636 |
| Downey Mildew | 0 | 652 | 652 | 0 | 104 | 756 |
| Frogeye Leaf Spot | 495 | 1045 | 1540 | 0 | 110 | 1650 |
| Healthy (Asymptomatic) | 908 | 724 | 1632 | 0 | 31 | 1663 |
| Potassium Deficiency | 0 | 1034 | 1034 | 0 | 49 | 1083 |
| Soybean Rust | 1131 | 496 | 1627 | 0 | 127 | 1754 |
| Target Spot | 0 | 1081 | 1081 | 0 | 27 | 1108 |
| Total | 4181 | 5467 | 9648 | 484 | 590 | 10,722 |

top of that, image augmentation was performed across the training data set during all model training using the Keras ImageDataGenerator class (Chollet, 2015). This library class acts as a template of sorts and allows the models to use a new transformation of each original image during each epoch of the training process. After some initial experimentation, all CNN models were trained using ImageDataGenerator settings that allowed for width and height shifts [0.2] (fraction of total width/height), brightness adjustment [0.9,1.1] (range for picking a shift value from), rotation [30] (degree range for random rotations), and vertical and horizontal reflection. Another image pre-processing step was to standardize image size. This size was $224 \times 224 \times 3$ (RGB) for all models except for Xception, which used $299 \times 299 \times 3$. Image normalization was performed to make sure that there was uniform data distribution in the range of [0,1]. This was done automatically within ImageDataGenerator by rescaling image pixel values by the number of channels, i.e., 1/255 (Chollet, 2015).

Other possible pre-processing techniques such as object segmentation or grayscaling were not implemented, following previous work (Ferentinos, 2018; Mohanty et al., 2016) demonstrating that they do not tend to improve final model performance for foliar disease diagnostics.

2.4. Model design

2.4.1. Training

Training a CNN is a special case of supervised machine learning. We provide the model with labeled training data – here, a set of classified images – and the model learns rules for classification that it can then apply to unlabeled data. In CNNs, this learning process is made more efficient by some elements of the model architecture. Concretely, feature

extraction is performed via convolutional layers, feature reduction is then performed by pooling layers, and then classification is done with a softmax layer (Chen et al., 2020; Wang et al., 2018).

Training and testing of the models was implemented using Tensorflow (Abadi et al., 2016) and Keras (Chollet, 2015) in a Linux environment on a shared computer cluster using an NVIDIA T4 Tensor Core GPU. Following the transfer-learning approach, baseline CNN architectures with weights pre-trained on the ImageNet dataset (Fei-Fei et al., 2010) were slightly modified (see next section for details) and then re-trained on our soybean image dataset. We compared the performance of several general-purpose CNN architectures while varying model parameters and data augmentation techniques.

2.4.2. Model comparisons and parameters -Base models from scratch

The general template for the CNN architectures we implemented is shown in Fig. 2. The base models (VGG16, ResNet50, Xception, DenseNet201, and EfficientNetB0) initiated with pre-trained ImageNet weights and their respective layers were used for feature extraction. The last, fully-connected layer of each model was dropped. In its place we added architectural elements following the work of Gianchandani et al. 2020 (Gianchandani et al., 2020). This comprised four new layers: (1) a global average pooling layer to reduce the number of parameters, (2) a fully connected layer of 128 neurons with a relu activation function and dropout of 0.2, (3) another fully connected layer of 64 neurons with relu activation and a dropout of 0.3, and (4) a fully connected dense with softmax activation for eight-class classification of the different soybean disease categories. Gianchandani et al. (2020) show that this additional architecture used in combination with an ensemble modeling technique increases the generalizability of the models.

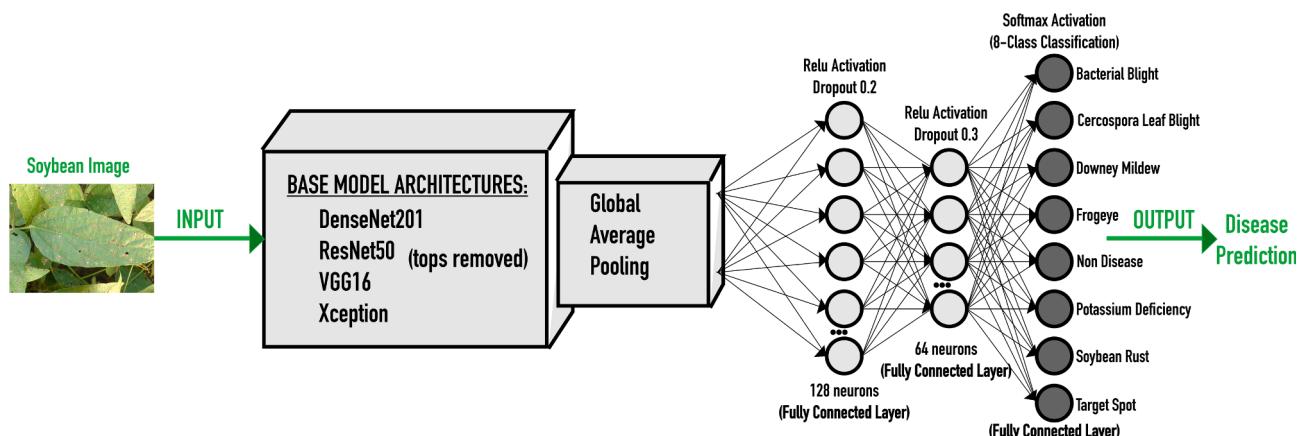


Fig. 2. Convolutional Neural Network Model Architecture. A simplified graphic/template for all model architectures and standardized parameters. Within the base model 'box' is, in reality, each respective base model's full architecture without the last layer. All models are initiated with the pre-trained ImageNet weights. For the 'scratch' models, all layer weights, including those inside the base model box are allowed to update (i.e., they are not frozen). For the additional DenseNet201 transfer learning models, the layers inside the box are frozen for a certain number of epochs.

There are two main approaches to transfer learning. In the first, which we refer to as the ‘frozen-core’ approach, the pre-trained, core fully-connected model weights are fixed, and only newly appended fully-connected layers of the network are trained on a new data set. In the second, known as ‘training from scratch’ the pre-trained layer weights are not fixed, but rather are used as a starting point, and all model weights are updated while training on new data. In other words, the entire model is trained. Each of the CNN base models we considered have quite different and complex architectures. So unfreezing/freezing only parts of each model makes it very hard to then compare model performance metrics in a standardized way. A previous study investigating CNN model performance trained with over 900 ferrograph images found that DenseNet and ResNet base model architectures had higher accuracy when ‘trained from scratch’ (Peng and Wang, 2020). We began with that approach.

After experimenting with various hyperparameter values to optimize accuracy on partial datasets, we standardized them across model experiments in order to allow for fair comparison. The hyperparameters used across all base models were: Learning rate: 0.0001, Optimizer = Stochastic Gradient Descent, Momentum = 0.9, Loss = Categorical Cross-entropy, Batch size = 32, Epochs = 100. By keeping the learning rate small, we optimize model accuracy at the cost of longer training times; this is in keeping with our overall aim of learning as much as possible from a small training dataset. The use of stochastic gradient descent as our optimization method follows research which has shown it to have high generalization performance in comparison to other adaptive optimization methods (Keskar and Socher, 2017). A value of 0.9 for momentum has been shown to help strike a balance between over- and under-fitting (Smith et al., 2018), and a batch size of 32 is generally regarded to be a good default value that takes advantage of the speed-up of matrix operations (Bengio, 2012).

2.4.3. Alternative Fine-Tuning strategies

After comparing the performance of the different base model architectures trained from scratch, we decided to further investigate the potential of transfer learning with other fine-tuning strategies. For this, we used the architecture which performed best when trained from scratch, namely, that with DenseNet201 as the base. First, we tried the frozen-core approach. For this, we kept learning rate at 0.0001, but allowed for more training time, setting the number of training epochs to 150. With this approach, we sought to fully leverage the pre-trained ImageNet weights.

Next, we implemented a fine-tuning strategy with partial freezing. Initially, we again froze the core layers and only allowed the added fully-connected layers to update. For this stage, we increased the learning rate to 0.001 and trained for 90 epochs. Then, we unfroze all layers of the model, lowered the learning rate back to 0.0001, and trained for a further 10 epochs (for a total of 100 epochs). Thus, this approach can be seen as combination of the frozen-core and from-scratch approaches.

2.4.4. Model sensitivity to variation in training data

We also wanted to get a sense for how model performance was affected by variation in the training data. In particular, we were curious about the impact of adding the augmented (horizontally flipped) bacterial images on model performance, as well as the impact of the mix of subject backgrounds, that is, whether attached or severed leaves were photographed. For the former, we removed the 484 additional bacterial blight augmented images from the training dataset and retrained from scratch the architecture using the DenseNet201 base model.

To quantify the effect of different subject backgrounds, we curated three different training datasets consisting of varying proportions of attached and severed leaf images. To reiterate, severed leaves were detached from the plant and photographed on either the ground in trimmed grass or on a white background of a table top or car hood. The three datasets consisted of (100 % attached/ 0 % severed), (75 % attached/ 25 % severed), and (50 % attached/ 50 % severed).

In order to maximize the variety of proportions of attached and severed leaves that we could analyze, each of the image databases contained 4500 original images. As for the main analyses, a Python script was used to randomly apportion the data into a training set and a testing set with a 85:15 split. For these comparisons, as an expediency we used a ResNet50 base model trained from scratch implementing the same hyperparameters as previously described for scratch models. Models based on ResNet50 performed nearly as well as those based on DenseNet201 but were trained more efficiently.

2.5. Performance metrics

Model performances were tested using 1076 previously unseen test images and were evaluated using quantitative metrics derived from a confusion matrix. These metrics include accuracy, precision, recall, and F-1 scores. Accuracy is often considered the model’s overall performance and calculates the correctly predicted labels by the equation:

$$\text{Accuracy} = \frac{\text{TruePositives} + \text{TrueNegatives}}{\text{TruePositives} + \text{TrueNegatives} + \text{FalsePositives} + \text{FalseNegatives}} \quad (1)$$

Precision is used to evaluate the exactness of the predictions by comparing the correctly predicted number of images in a disease category with the total number predicted in that category by the model:

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (2)$$

Recall uses a ratio for the number of correctly predicted images in a disease category compared to all actual observations of that category. A higher recall implies better ability to predict both positive and negative observations:

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (3)$$

The F1-score is an integrative measure that combines precision and recall:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Loss and accuracy plots were generated using *ggplot2* (Wickham, 2011) in R studio (R Core Team, 2019).

FLOPs, or floating point operations, is a measure of the total number of operations that the model has to perform to make a classification. More operations will require more computing power. FLOPs for each model architecture was calculated using the *keras-flops* python package (Tokusumi, 2020).

Inference latency is the amount of time (seconds) that it takes the model to process one image for a batch size of one. In other words, how long does it take the model to make a disease prediction when given one image of a diseased soybean. This is important when considering deployment of models for their real time use by end-users. Once deployed for real world usage, the models will not be running on a GPU, so the inference latency of each model architecture was calculated for comparison using a 4 core CPU and a python script implementing the TensorFlow Graph function (see RoboCrop github repository).

3. Results

3.1. Model performance comparisons

The relative performances of transfer-learning architectures trained from scratch and based on VGG16, Xception, ResNet50, DenseNet201, and EfficientNetB0, as well as the additional fine-tuning and augmentation models using DenseNet201 are summarized in Table 2. The highest overall testing accuracy, 96.8 %, was achieved with the base model architecture of DenseNet201 trained from scratch (Fig. 3). This

Table 2

Model Performance Comparison. Test evaluation of each of the base model architectures implemented. All performance calculations done using a GPU as described in the methods section with the exception of ‘average inference latency’, which was calculated using a 4 core CPU.

| Base Model Architecture | Accuracy | Precision | Recall | F1 | Secs | Secs | Total | Trainable Parameters | Non-trainable Parameters | FLOPS (G) | Avg Inference |
|--|----------|-----------|--------|------|------|--------|------------|----------------------|--------------------------|-----------|-----------------------|
| | | | | | | | | | | | Latency (secs/sample) |
| DenseNet201 (Scratch) | 96.8 | 96.6 | 97.1 | 96.8 | 29.3 | 8398 | 18,576,648 | 18,347,592 | 229,056 | 8.63 | 0.2088 |
| ResNet50 (Scratch) | 95.9 | 95.9 | 96.4 | 96.1 | 20.0 | 5783 | 23,858,760 | 23,805,640 | 53,120 | 7.75 | 0.1070 |
| EfficientNetB0 (Scratch) | 94.2 | 94.1 | 94.3 | 94.2 | 13.7 | 3913 | 4,222,315 | 4,180,292 | 42,023 | 0.795 | 0.0764 |
| Xception (Scratch) | 92.4 | 93.0 | 91.9 | 92.4 | 37.1 | 10,621 | 21,132,528 | 21,078,000 | 54,528 | 16.8 | 0.1437 |
| VGG16 (Scratch) | 90.8 | 91.0 | 90.9 | 90.4 | 34.5 | 9868 | 14,789,128 | 14,789,128 | 0 | 30.7 | 0.2464 |
| DenseNet201 | 96.7 | 95.9 | 97.1 | 96.4 | 29.0 | 7974 | 18,576,648 | 18,347,592 | 229,056 | 8.63 | 0.2030 |
| (No Extra Augmentation) | | | | | | | | | | | |
| DenseNet201 (Freeze first 90, unfreeze last 10 epochs) | 94.9 | 95.5 | 94.9 | 95.1 | 12.8 | 3718 | 18,576,648 | 18,347,592 | 229,056 | 8.63 | 0.1801 |
| DenseNet201 (Freeze core- all 150 epochs) | 84.6 | 85.2 | 86.0 | 85.4 | 11.1 | 3270 | 18,576,648 | 254,664 | 18,321,984 | 8.63 | 0.1905 |

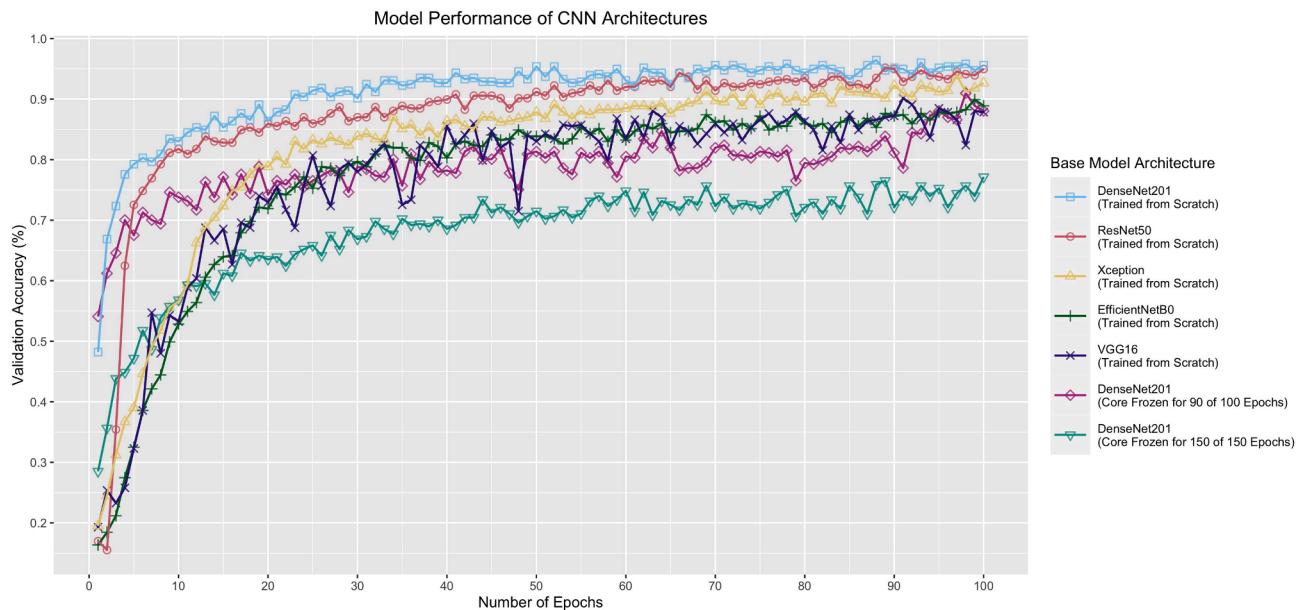


Fig. 3. Accuracy Comparison. Validation accuracy of each model while trained through 100 epochs. Note that the last 50 epochs of the DenseNet201 (Core Frozen for 150 of 150 epochs) base model are not shown in this image, but it maintains the lowest overall validation accuracy.

model was closely followed by the model based on ResNet50, and then next by the EfficientNetB0 and Xception architectures. The DenseNet201 from-scratch model also had the highest precision, recall, and F1-score metrics. The EfficientNetB0 model was the fastest to train for the from-scratch models. EfficientNetB0 also had significantly fewer parameters than any other model architecture, used by far the least amount of compute power (FLOPs), and had the lowest inference latency for making model predictions. The DenseNet201 models that implemented full or partial freezing of the core model weights both decreased the amount of total training time but diminished the overall model performance. The partial-freezing approach of DenseNet201 (first 90 epochs frozen, last 10 epochs unfrozen) had the shortest total training time.

The training and validation loss for the from-scratch DenseNet201 model can be seen in Fig. 4 where both plots follow a similar trend with

approximately similar values; training and validation are simultaneously and continuously maximizing accuracy while minimizing loss. This is desirable and it appears that any slight over-fitting that may be occurring between training and validation did not significantly impact the model’s ability to make correct predictions and maintain high accuracy (96.8 %) on unseen test data.

Fig. 5 shows a confusion matrix for the DenseNet201 from-scratch model (Cipriano, 2018). The model’s worst precision was bacterial blight. It predicted bacterial blight as the disease 118 times when there were only 108 actual bacterial blight images. The model’s worst recall was for soybean rust. The model correctly predicted 163 out of 175 soybean rust images (93.14 %). The model had the best performance on potassium deficiency, which it correctly identified in all 109 cases.

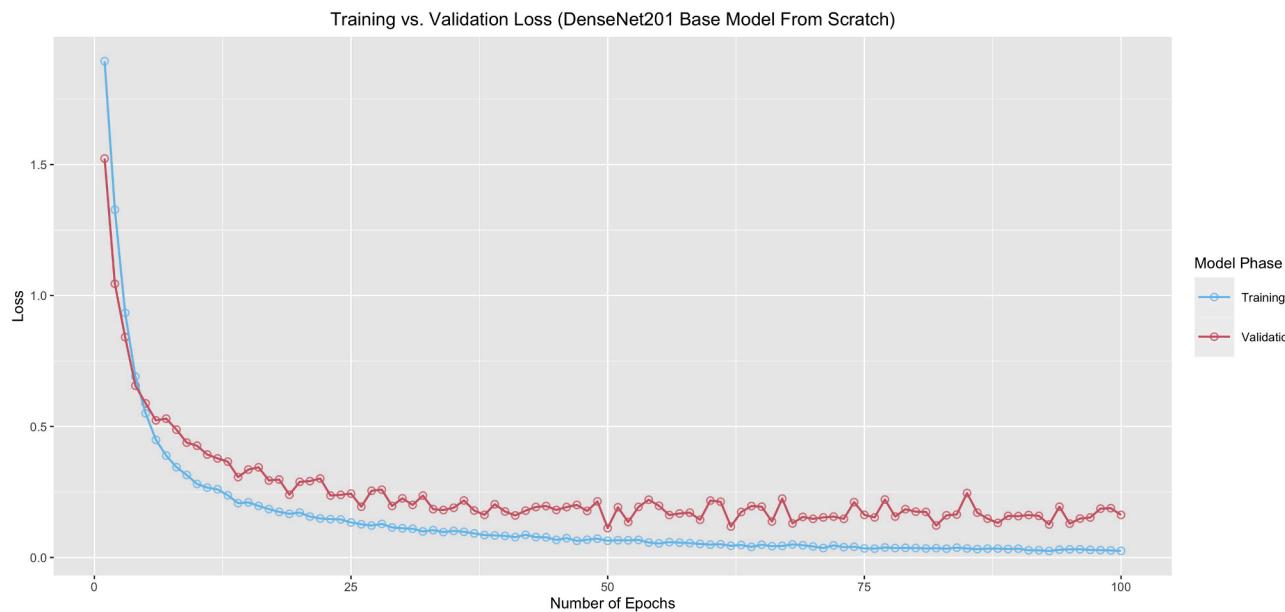


Fig. 4. DenseNet201 Training Loss vs Validation Loss. Minimization of the loss function through 100 epochs for the training dataset compared to the validation dataset for DenseNet201 base model trained from scratch.

| Confusion Matrix | | | | | | | | |
|------------------------|---|---|--|---|---|---|---|--|
| Predicted | | | | | | | | |
| | Actual | | | | | | | |
| Bacterial Blight | 106 9.85% | 3 0.28% | | 1 0.09% | | 6 0.56% | 2 0.19% | 118 89.83% 10.17% |
| Cercospora Leaf Blight | | 153 14.22% | | | | 3 0.28% | | 156 98.08% 1.92% |
| Downy Mildew | 1 0.09% | | 76 7.06% | 1 0.09% | | 1 0.09% | 1 0.09% | 80 95.00% 5.00% |
| Frogeye | | 2 0.19% | | 163 15.15% | | 2 0.19% | 1 0.09% | 168 97.02% 2.98% |
| Non-Disease | | 3 0.28% | | 164 15.24% | | | 1 0.09% | 168 97.62% 2.38% |
| Potassium Deficiency | | | | | 109 10.13% | | | 109 100% 0.00% |
| Soybean Rust | | 3 0.28% | 1 0.09% | | 1 0.09% | | 163 15.15% | 168 97.02% 2.98% |
| Target Spot | 1 0.09% | | | 1 0.09% | | | 107 9.94% | 109 98.17% 1.83% |
| Column Total | 108 98.15% 1.85% | 164 93.29% 6.71% | 77 98.70% 1.30% | 166 98.19% 1.81% | 165 99.39% 0.61% | 109 100% 0.00% | 175 93.14% 6.86% | 112 95.54% 4.46% |
| Row Total | | | | | | | | 1076 96.75% 3.25% |

Fig. 5. Confusion Matrix for DenseNet201 Base Model (Trained from Scratch). The performance of trained models was tested using 1076 previously unseen test images. Model predicted categories for each test image can be read on the row values of the left (y-axis) as compared to the actual disease categories on the column values of the bottom (x-axis). The green diagonal boxes are filled with the correct number of predictions for that category. White boxes filled with red numbers are the number of incorrect predictions. The green numbers in the black boxes display the precision and recall scores for each disease category. Precision scores are the green numbers within the black far-right column. Recall scores are within the black bottom right row. Overall model performance is in the blue bottom right box. 1041 correct predictions out of 1076 total test images; 96.75% accuracy.

3.2. Model sensitivity to variation in training data

Removing the additional augmented bacterial blight images had slightly negative effects on the overall model performance using the

scratch DenseNet201 base model architecture (as seen in Table 2). The effect was more pronounced on the bacterial blight category specifically. Removal of the additional augmented images resulted in a precision decrease for bacterial blight images from 89.83 % to 82.86 % and a

recall decrease from 98.15 % to 96.67 %. The model trained without the additional augmented images was not as good at being able to predict or recognize bacterial blight.

Comparisons of model performances depending on the mix of attached and severed leaf images are summarized in Table 3. As a reminder, these models used a ResNet50 base model architecture to save on computing time. The best performance was from the dataset that included only attached leaf field images (100 % Attached / 0 % Severed). In fact, the trend appears to be that overall model performance slightly decreased as more severed leaf images were included in the datasets.

We have deployed our soybean foliar disease classifier, which we call Robocrop, as a freely-available web application (<http://sickbeans.skullisland.info>), built using the Flask framework (Ahmad, 2021). Codes used for model specification and training are available in a Github repository (<https://github.com/nzb0054/RoboCrop-CNN-WebApp>).

4. Discussion

Our goal was to develop an application to make soybean disease identification more accessible and hopefully reduce the excessive use of pesticides and thereby forestall the evolution of pesticide resistance. Our approach was to train a CNN, and to explore ways of making the training process more efficient. We began by building a new training data set, and by considering aspects of image composition that could affect the efficiency with which a CNN model could extract meaningful feature data. Then we optimized various data augmentation, and transfer-learning approaches. Several high-performing, general-purpose computer vision networks are available for this purpose. We found that, for soybean foliar disease classification, the best performing base model (in terms of test accuracy) was DenseNet201, using the from-scratch training approach. Using that model, we were able to develop an application that distinguishes between eight soybean disease/deficiency classes with an overall accuracy of 96.75 %. We were able to achieve this level of performance with a training dataset that, while significantly larger than what had previously been available for soybean foliar diseases, was still quite feasible to assemble. Hence, we find that the state of deep learning technology has advanced to the point where high-performance automated plant disease diagnostics is quite a reachable goal. With freely-available general-purpose computer vision models it would seem possible to develop custom computer vision applications with only about 1000 images of each class. Of course training data acquisition and model training still take substantial investments, but once the model is trained, its application is rather simple (Mohanty et al., 2016). With the Robocrop web application, users can submit a picture of a diseased soybean leaf in the field, and get an almost instant diagnosis, along with links to further extension resources for diagnostics confirmation and management options.

Some caveats are in order. We sought to collect a comprehensive set of images, portraying the symptoms of soybean diseases that are common in our area, under a variety of field conditions. Nevertheless, given logistical constraints, our training data set still has strong spatial and temporal biases. The model can only be expected to classify diseases and deficiencies that are represented in the training data. Regardless of what data are input, the model always does the same thing: assigns

identification probabilities to each of the classes it was trained to recognize. In most cases, if a user submits an image of something that the model has not been trained to recognize, it will simply assign low probabilities to each of its disease and deficiency classes, and we have configured Robocrop to return diagnoses only when the assignment probability is greater than 80 %. Nevertheless, if images are submitted from places and times that fall outside of the current training dataset, we cannot rule out that Robocrop might systematically assign high probabilities to incorrect diagnoses. We will get a better sense for the robustness of the current model as it is deployed more broadly over space and time, and we expect further improvements in its usefulness and accuracy as more training data become available. Robocrop stores user-submitted images. Our plan is to vet these images, and then use them to iteratively extend and refine the model (Shin et al., 2016).

While assembling our training data, we deliberately sought to acquire images of subjects in a variety of settings, that is, with different backgrounds. We put special emphasis on photographing a mix of attached leaves still in the field and severed leaves placed on an alternative background. We expected that this would help the CNN learn to disregard spurious background information and hone in on subject features. Contra to this expectation, we found that models perform better when they are trained exclusively on images of attached leaves. Conversely, increasing the proportion of severed leaf images reduces model performance. It is possible that this is an artifact of reducing the total size of the training dataset so that we could manipulate the ratios of attached and severed leaf images. It is also possible that there is genuinely useful information in the background of an image about the health state of a subject, for example, the overall structure of the canopy or the contrast of leaf colors. If the inferred positive effects of attached leaf images are reliable, this would make further data collection easier, and best practices for obtaining query images more convenient. It is less work to take a picture of an attached leaf.

Our development experience suggests that researchers planning to develop similar diagnostic applications should consider data augmentation as a way of increasing parity among classes. Our top-performing model has the most difficulty (in the sense of the lowest precision) recognizing bacterial blight. Simply put, the model occasionally predicted bacterial blight as the disease category when that was not the case. CNNs typically perform best with large, balanced datasets (Li et al., 2018) and bacterial blight had the least representation of any category. We attempted to remedy this by doubling the bacterial blight images used in the original training set through data augmentation (horizontal flipping). This technique appears to have been effective. Omitting these additional, augmented images diminished overall model performance and caused a noticeable decline in the precision and recall scores for the bacterial blight disease category. Experimenting more deeply with the further addition of augmented original images for all disease categories might allow for further increases in model performance. In sum, it appears that training data limitations underlie some of the shortcomings in model performance, and hence additional data should continue to improve its performance.

In our experiments with frozen-core approaches to model training, we found that in comparison to from-scratch training, model performance was somewhat diminished, but the training time was considerably reduced. In our case, with relatively few training data and other sustainability challenges to address – namely, increasing and safeguarding the productivity and soybean farms – we decided to optimize classification accuracy, and mostly relied on from-scratch training. But it may be useful in future work to explore more fully the potential of freezing or partially freezing core network layers to reduce the computational cost of model training. Likewise, EfficientNetB0 was by far the fastest from-scratch model to train and was comparable to the frozen Densenet201 training. EfficientNetB0 was also the least computationally expensive and had the shortest inference latency time. Again, in our case, where we are catering to a niche agricultural client base, we do not anticipate application use rates that would demand high-

Table 3
Model Performance of Differing Proportion of Attached and Severed Images. All models implemented an identical base model ResNet50 architecture from scratch.

| Images Used in Dataset (ResNet50 Models) | Accuracy | Precision | Recall | F1 |
|--|----------|-----------|--------|-------|
| 50 % Attached / 50 % Severed | 95.18 | 95.51 | 95.93 | 95.60 |
| 75 % Attached / 25 % Severed | 96.48 | 96.54 | 96.53 | 96.50 |
| 100 % Attached / 0 % Severed | 97.58 | 97.66 | 97.75 | 97.69 |

throughput computing optimizations. But for high-throughput applications, the EfficientNewBO architecture could be an integral part of increasing development sustainability and usability.

In conclusion, with mostly off-the-shelf neural network architectures, and with original digital images of diseased soybean plants taken by a single researcher over the course of two seasons, we were able to develop an automated classifier of soybean foliar diseases that are common in the southeastern US that appears to be correct in more than 95 % of the cases. We also found additional evidence for how specific base-model architectures and frozen-core training approaches can reduce the computational cost of model training. For this particular application, with few training data, the computation cost of thorough accuracy optimization was not prohibitive. Overall, we found that the keys to achieving high model accuracy from a relatively small training data set where image composition, data augmentation, and from-scratch transfer learning.

5. Data archiving

Our soybean disease image library is deposited with Data Dryad and openly available at <https://doi.org/10.5061/dryad.41ns1rnj3>. The dataset is named 'Auburn Soybean Disease Image Dataset (ASDID)' (Bevers et al., 2022).

CRediT authorship contribution statement

Noah Bevers: Data curation, Methodology, Formal analysis, Writing – original draft, Visualization. **Edward J. Sikora:** Data curation, Conceptualization, Writing – review & editing, Visualization, Funding acquisition. **Nate B. Hardy:** Conceptualization, Methodology, Supervision, Writing – review & editing, Visualization, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Respective links are mentioned within manuscript. Additional image data will be shared via a Dryad repository upon acceptance of the manuscript.

Acknowledgements

This work was supported by the Alabama Soybean Producers Association and the Alabama Agricultural Experiment Station (AAES).

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.
- Ahmad, N., 2021. Flask Web Applications in Python [WWW Document]. URL <https://github.com/noumannnahmad/Flask>.
- Albawi, S., Mohammed, T.A.M., Alzawi, S., 2017. Understanding of a Convolutional Neural Network. *Ieee* 16.
- Ali, N., 2010. The soybean: botany, production, and uses. CABI. DOI: 10.1079/9781845936440.0000.
- Barbedo, J.G.A., 2018. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* 153, 46–53. <https://doi.org/10.1016/j.compag.2018.08.013>.
- Barbedo, J.G.A., Koenigkan, L.V., Santos, T.T., 2016. Identifying multiple plant diseases using digital image processing. *Biosyst. Eng.* 147, 104–116. <https://doi.org/10.1016/j.biosystemseng.2016.03.012>.
- Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science.* 437–478. https://doi.org/10.1007/978-3-642-35289-8_26.
- Bevers, N., Sikora, E.J., Hardy, N.B., 2022. Auburn soybean disease image dataset (ASDID). Dryad, Dataset. <https://doi.org/10.5061/dryad.41ns1rnj3>. Submitted for publication.
- Bradley, C., Allen, T.W., Sisson, A.J., Bergstrom, G.C., Bissonnette, K.M., Bond, J., Byamukama, E., Chilvers, M.I., Collins, A.A., Damicone, J.P., Dorrance, A.E., Dufault, N.S., Esker, P.D., Faske, T.R., Fiorellino, N.M., Giesler, L.J., Hartman, G.L., Hollier, C.A., Isakeit, T., Jackson-Ziems, T.A., Jardine, D.J., Kelly, H.M., Kemerait, R.C., Kleczewski, N.M., Koehler, A.M., Kratochvil, R.J., Kurle, J.E., Malwick, D.K., Markell, S.G., Mathew, F.M., Mehl, H.L., Mehl, K.M., Mueller, D.S., Mueller, J.D., Nelson, B.D., Overstreet, C., Padgett, G.B., Price, P.P., Sikora, E.J., Small, I., Smith, D.L., Spurlock, T.N., Tande, C.A., Telenko, D.E.P., Tenuta, A.U., Thiessen, L.D., Warner, F., Wiebold, W.J., Wise, K., 2020. Soybean yield loss estimates due to diseases in the United States and Ontario, Canada, from 2015 to 2019. *Plant Heal. Prog.* 21, 238–247. <https://doi.org/10.1094/PHP-01-21-0013-RS>.
- Chen, J., Chen, J., Zhang, D., Sun, Y., Nanehkaran, Y.A., 2020. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* 173, 105393. <https://doi.org/10.1016/j.compag.2020.105393>.
- Chollet, F., 2015. Keras.
- Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-Janua, 1800–1807.* 10.1109/CVPR.2017.195.
- Cipriano, W., 2018. Pretty Print Confusion Matrix.
- Dhar, P., 2020. The carbon impact of artificial intelligence. *Nat. Mach. Intell.* 2, 423–425. <https://doi.org/10.1038/s42256-020-0219-9>.
- Du, Y., Zhang, G., Tsang, D., Jawed, M.K., 2021. Deep-CNN based Robotic Multi-Class Under-Canopy Weed Control in Precision Farming 2273–2279.
- Fei-Fei, L., Deng, J., Li, K., 2010. ImageNet: Constructing a large-scale image database. *J. Vis.* 9, 1037–1037. 10.1167/9.8.1037.
- Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* <https://doi.org/10.1016/j.compag.2018.01.009>.
- Fried, G., Chauvel, B., Reynaud, P., Sache, I., 2017. Impact of Biological Invasions on Ecosystem Services. Decreases in Crop Production by Non-Native Weeds, Pests, and Pathogens. 83–101. <https://doi.org/10.1007/978-3-319-45121-3>.
- Garcia, J., Barbedo, A., Godoy, C.V., 2015. Automatic Classification of Soybean Diseases Based on Digital Images of X Sbi Agro.
- Gianchandani, N., Jaiswal, A., Singh, D., Kumar, V., Kaur, M., 2020. Rapid COVID-19 diagnosis using ensemble deep transfer learning models from chest radiographic images. *J. Ambient Intell. Humaniz. Comput.* <https://doi.org/10.1007/s12652-020-02669-6>.
- Gui, J., Hao, L., Zhang, Q., Bao, X., 2015. A new method for soybean leaf disease detection based on modified salient regions. *Int. J. Multimed. Ubiquitous Eng.* 10, 45–52. 10.14257/ijmuae.2015.10.6.06.
- Hartman, G.L., Rupe, J.C., Sikora, E.J., Domier, L.L., Davis, J.A., Steffey, K., 2016. PART IV: Soybean Disease and Pest Management Strategies., in: Compendium of Soybean Diseases and Pests, Fifth Edition. APS Publications, pp. 167–173. 10.1094/9780890544754.005.
- Hartman, G.L., West, E.D., Herman, T.K., 2011. Crops that feed the World 2. Soybean-worldwide production, use, and constraints caused by pathogens and pests. *Food Secur.* 3, 5–17. <https://doi.org/10.1007/s12571-010-0108-x>.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-Decem, 770–778.* 10.1109/CVPR.2016.90.
- Henderson, P., Hu, J., Romoff, J., Brunsell, E., Jurafsky, D., Pineau, J., 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Mach. Learn. Res.* 21, 1–25.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-Janua, 2261–2269.* 10.1109/CVPR.2017.243.
- Hughes, D.P., Salathe, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics.
- Jahanbakhshi, A., Abbaspour-Gilandeh, Y., Heidarbeigi, K., Momeny, M., 2021a. Detection of fraud in ginger powder using an automatic sorting system based on image processing technique and deep learning. *Comput. Biol. Med.* 136, 104764. <https://doi.org/10.1016/j.combiomed.2021.104764>.
- Jahanbakhshi, A., Abbaspour-Gilandeh, Y., Heidarbeigi, K., Momeny, M., 2021b. A novel method based on machine vision system and deep learning to detect fraud in turmeric powder. *Comput. Biol. Med.* 136, 104728. <https://doi.org/10.1016/j.combiomed.2021.104728>.
- Jahanbakhshi, A., Momeny, M., Mahmoudi, M., Radeva, P., 2021c. Waste management using an automatic sorting system for carrot fruit based on image processing technique and improved deep neural networks. *Energy Reports* 7, 5248–5256. <https://doi.org/10.1016/j.egyr.2021.08.028>.
- Kalaitzandonakes, N., Kaufman, J., Zahringer, K., 2019. The Economics of Soybean Disease Control. CABI.
- Karlekar, A., Seal, A., 2020. SoyNet: Soybean leaf diseases classification. *Comput. Electron. Agric.* 172 <https://doi.org/10.1016/j.compag.2020.105342>.
- Keskar, N.S., Socher, R., 2017. Improving Generalization Performance by Switching from Adam to SGD.
- Kim, P., 2017. Convolutional Neural Network. MATLAB Deep Learn. 121–147 <https://doi.org/10.1007/978-1-4842-2845-6>.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 56, 84–90.

- Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T., 2019. Quantifying the Carbon Emissions of Machine Learning.
- Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–444. <https://doi.org/10.1038/nature14539>.
- Li, S., Song, W., Qin, H., Hao, A., 2018. Deep variance network: An iterative, improved CNN framework for unbalanced training datasets. *Pattern Recognit.* 81, 294–308. <https://doi.org/10.1016/j.patcog.2018.03.035>.
- Miller, S.A., Beed, F.D., Harmon, C.L., 2009. Plant disease diagnostic capabilities and networks. *Annu. Rev. Phytopathol.* 47, 15–38. <https://doi.org/10.1146/annurev-phyto-080508-081743>.
- Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7, 1–10. <https://doi.org/10.3389/fpls.2016.01419>.
- Orke, E.C., 2006. Crop losses to pests. *J. Agric. Sci.* 144, 31–43. <https://doi.org/10.1017/S0021859605005708>.
- Peng, P., Wang, J., 2020. How to fine-tune deep neural networks in few-shot learning?. R Core Team, R.F. for S.C., 2019. R: A Language and Environment.
- Sagar, A., Dheeba, J., 2020. On Using Transfer Learning For Plant Disease Detection. *bioRxiv* 2020.05.22.110957. 10.13140/RG.2.2.12224.15360/1.
- Sahu, S.K., Pandey, M., Geete, K., 2021. Classification of Soybean Leaf Disease from Environment effect Using Fine Tuning Transfer Learning. *25*, 2188–2201.
- Savary, S., Willocquet, L., Pethybridge, S.J., Esker, P., McRoberts, N., Nelson, A., 2019. The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* 3, 430–439. <https://doi.org/10.1038/s41559-018-0793-y>.
- Shin, H.C., Roth, H.R., Gao, M., Liu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Trans. Med. Imaging* 35, 1285–1298. <https://doi.org/10.1109/TMI.2016.2528162>.
- Shrivastava, S., Hooda, D.S., 2014. Automatic Brown Spot and Frog Eye Detection from the Image Captured in the Field. *Am. J. Intell. Syst.* 4, 131–134. <https://doi.org/10.5923/j.ajis.20140404.01>.
- Shrivastava, S., Singh, S.K., 2017. Soybean plant foliar disease detection using image retrieval approaches. *Multimed Tools Appl.* 26647–26674 <https://doi.org/10.1007/s11042-016-4191-7>.
- Sikora, E.J., Allen, T.W., Wise, K.A., Bergstrom, G., Bradley, C.A., Bond, J., Brown-Rytlewski, D., Chilvers, M., Damiconi, J., DeWolf, E., Dorrance, A., Dufault, N., Esker, P., Faske, T.R., Giesler, L., Goldberg, N., Golod, J., Gómez, I.R.G., Grau, C., Grybauskas, A., Franc, G., Hammerschmidt, R., Hartman, G.L., Henn, R.A., Hershman, D., Hollier, C., Isakeit, T., Isard, S., Jacobsen, B., Jardine, D., Kemerait, R., Koenning, S., Langham, M., Malwick, D., Markell, S., Marois, J.J., Monfort, S., Mueller, D., Mueller, J., Mulrooney, R., Newman, M., Osborne, L., Padgett, G.B., Ruden, B.E., Rupe, J., Schneider, R., Schwartz, H., Shaner, G., Singh, S., Stromberg, E., Sweets, L., Tenuta, A., Vaiciunas, S., Yang, X.B., Young-
- Kelly, H., Zidek, J., 2014. A coordinated effort to manage soybean rust in North America: A success story in soybean disease monitoring. *Plant Dis.* 98, 864–875. <https://doi.org/10.1094/PDIS-02-14-0121-FE>.
- Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* 1–14.
- Smith, S.L., Kindermans, P.J., Ying, C., Le, Q. V., 2018. Don't decay the learning rate, increase the batch size. *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.* 1–11.
- Strubell, E., Ganesh, A., McCallum, A., 2020. Energy and policy considerations for modern deep learning research. *AAAI 2020 - 34th AAAI Conf. Artif. Intell.* 1393–13696. 10.1609/aaai.v34i09.7123.
- Tan, M., Le, Q. V., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. *36th Int. Conf. Mach. Learn. ICML 2019* 2019-June, 10691–10700.
- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C., 2018. A Survey on Deep Transfer Learning. *Int. Conf. Artif. neural networks 1–10*.
- Tokusumi, 2020. Keras-flops Calculator.
- Vasconez, J.P., Delpiano, J., Vougioukas, S., Auat Cheein, F., 2020. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Comput. Electron. Agric.* 173, 105348 <https://doi.org/10.1016/j.compag.2020.105348>.
- Wang, S.H., Phillips, P., Sui, Y., Liu, B., Yang, M., Cheng, H., 2018. Classification of Alzheimer's Disease Based on Eight-Layer Convolutional Neural Network with Leaky Rectified Linear Unit and Max Pooling. *J. Med. Syst.* 42, 1–11. <https://doi.org/10.1007/s10916-018-0932-7>.
- Wickham, H., 2011. Ggplot2. *Wiley Interdiscip. Rev. Comput. Stat.* 3, 180–185. <https://doi.org/10.1002/wics.147>.
- Wrather, A., Shannon, G., Balardin, R., Carregal, L., Escobar, R., Gupta, G.K., Ma, Z., Morel, W., Ploper, D., Tenuta, A., 2010. Effect of Diseases on Soybean Yield in the Top Eight Producing Countries in 2006. *Plant Heal. Prog.* 11, 29. <https://doi.org/10.1094/php-2010-0102-01-rs>.
- Wu, Q., Zhang, K., Meng, J., 2019. Identification of Soybean Leaf Diseases via Deep Learning. *J. Inst. Eng. Ser. A* 100, 659–666. <https://doi.org/10.1007/s40030-019-00390-y>.
- Yu, M., Ma, X., Guan, H., Liu, M., Zhang, T., 2022. A Recognition Method of Soybean Leaf Diseases Based on an Improved Deep Learning Model. *Front. Plant Sci.* 13, 1–23. <https://doi.org/10.3389/fpls.2022.878834>.
- Zeng, F., Arnao, E., Zhang, G., Olaya, G., Wullschleger, J., Sierotzki, H., Ming, R., Bluhm, B.H., Bond, J.P., Fakhoury, A.M., Bradley, C.A., 2015. Characterization of quinone outside inhibitor fungicide resistance in *Cercospora sojina* and development of diagnostic tools for its identification. *Plant Dis.* 99, 544–550. <https://doi.org/10.1094/PDIS-05-14-0460-RE>.