Visualizing Feature Maps for Model Selection in Convolutional Neural Networks

Sakib Mostafa
University of Saskatchewan
sakib.mostafa@usask.ca

Debajyoti Mondal
University of Saskatchewan
dmondal@cs.usask.ca

Michael Beck
University of Winnipeg
m.beck@uwinnipeg.ca

Christopher Bidinosti
University of Winnipeg
c.bidinosti@uwinnipeg.ca

Christopher Henry
University of Winnipeg
ch.henry@uwinnipeg.ca

Ian Stavness
University of Saskatchewan
ian.stavness@usask.ca

Abstract

Convolutional neural networks (CNN) are increasingly being used to achieve state-of-the-art performance for various plant phenotyping and agricultural tasks. While constructing such CNN models, a common problem is overparameterization, which may lead to a model becoming overfit on a training dataset. This problem is particularly relevant for plant datasets with limited variation and/or small samples sizes. Inspection of the loss and accuracy curves is a common way to detect overfitting in a CNN model, but it provides little insight into how the model could be improved. There are several reasons contributing to the overfitting of a CNN model; however, in this paper, we aim at explaining overfitting in a CNN classification model by analyzing the features learned at various depths of the model. We use three plant phenotyping datasets in our experimental studies. Our comparative analysis between the visualizations of the feature maps obtained from overfit and balanced models reveals that the image background often influences an overfit model's behavior. Researchers with limited deep learning domain knowledge often attempt to build deeper layer models with the hope of improving performance. Using Guided Backpropagation, we show how the pairwise similarity matrix between the visualization of the features learned at different depths can be leveraged to pave a new way to potentially select a better CNN model by removing redundant layers.

1. Introduction

The ability of deep learning to extract complex features from a large amount of data [28] has motivated experts from agriculture [8, 48] (i.e., precision agriculture, crop breeding, plant phenotyping) to adopt deep learning approaches. However, plant image datasets often have different characteristics as compared to general image datasets, such as small sample sizes, limited variation, highly self-similar foreground objects, and highly simplified backgrounds. Therefore, when used as a "black box" [44, 32], deep learning models for plant datasets may perform poorly. To improve the trustworthiness of models and to design them effectively for the unique challenges of specialized datasets, many recent studies have focused on explaining the learning and prediction of deep learning models [29, 44].

Convolutional neural networks (CNN) are the most widely used type of deep learning models in image-based plant phenotyping. They can learn features adaptively from the images in different spatial resolutions. However, when designing a CNN model, a common phenomenon is model overfitting. An overfit model can approximate or memorize the training in a variety of ways to predict the output and thus fails to generalize to unseen examples in the testing data [34]. A common way to evaluate a model's performance and detect overfitting is by inspecting accuracy and loss curves [16, 18], but this does not provide insight into what and how much a model is learning nor does it allow a practitioner to understand which features or part of the image contributed to the model's prediction.

To explain the learning of CNN models, researchers have proposed different feature-map visualization techniques [5, 26, 42, 35, 40]. Although feature-map visualization approaches have been successful in explaining which image features are learned by a model, to the best of our knowledge, such visualizations have yet to be used to detect or explain the behavior of overfit CNN models. Furthermore, there have been few attempts to analyze overfitting in CNN models used for image-based plant phenotyping tasks that is understandable by a researcher without the deep learning models' domain knowledge.

In this study, we focus on the plant phenotyping task of plant species classification, which is relevant in digital agriculture, e.g. precision herbicide application [47], and is a popular task for employing CNN models [12, 4]. We examine the features learned by the intermediate layers of CNN classifiers to understand overfit models' behavior and the contribution of images' background in overfitting. To examine how the CNN models learn in various conditions (overfit or balanced), we use Guided Backpropagation (GBP) [42] to visualize the features being learned at different layers of the CNN models. In addition to examining the model behavior, we explore whether the GBP-based feature visualizations could be leveraged to detect overfitting and provide guidance to select a CNN model with appropriate depth.

There are two main contributions of this study. First, we visualize the intermediate layers of different CNN models to investigate whether there is a difference in the learned features of an overfit model and a balanced model (i.e., a model which neither overfit nor underfit). Our experimental results with plant classification datasets show that the image background features may have more influence on model prediction for overfit models than the balanced models. Second, we propose a novel SSIM-based evaluation technique that relates overfitting to the depth of the model and provides an intuitive way to understand the differences between overfit and balanced models. Here SSIM refers to a measurement of the similarity between two feature map visualizations. Our analysis shows that in a model with a large number of convolutional layers, the features learned in the initial layers are more diverse than the features learned in the deeper layers. We discuss how this phenomenon may help detect potential overfitting in the CNN models or to select a better model by estimating an appropriate model depth.

2. Related Studies

Visualization for CNN models. A rich body of research has examined how changing the intensity of the pixels may change a CNN classifier's prediction [42, 41, 51]. This gave rise to deconvolutional networks (Deconvnet) [51] that provide insight into the function of a CNN classifier's intermediate layers by modifying the model's gradient and displaying the visual patterns in the input image that generated the activations. Zhou et al. [52] proposed class activation mapping (CAM), where convolutional and average max-pooling layers replace the fully connected layers in a CNN image classifier, which helps achieve class-specific feature visualization. Gradient-weighted Class Activation Mapping (Grad-CAM) proposed by Selvaraju et al. [39] is an extension of the CAM that creates a class-specific heatmap of the objects in the input image contributing to the prediction by using the weights and activations of a trained CNN model.

There have been several attempts that deviate from deconvolutional networks. Simonyan et al. [41] used the gradient of a CNN model's output with respect to the input image's pixel intensities to generate saliency maps. In a perturbation-based forward propagation approach, Zintgraf et al. [53] analyzed the difference in prediction by marginalizing each input patch and creating saliency maps for each instance of a CNN image classifier. In PatternNet [24], the authors trained a linear signal estimator on top of a non-linear neural network to visualize the relation between the neural network model's signal and attributed pattern.

Explainable Models in Plant Phenotyping. Several recent studies leverage deep learning for plant phenotyping tasks [33, 45, 20]. However, explainable deep learning models in plant phenotyping still remain to be an active field of research with lots of scope for improvement [8]. Ghosal et al. [17] proposed a novel framework to identify the stress regions on a plant leaf. They visualized feature maps in various layers that detected the stress regions and the output of the framework was the summation of the feature maps. Nagasubramanian et al. [31] used a saliency map based visualization technique to detect the hyperspectral wavelengths that is responsible for the models' performance. They trainied DenseNet-121 to classify stress levels in Soybean leaf and used different visualization techniques to visualize the parts of the leaf that contributed to the models' decision. They showed that even when the model misclassifies an image, it still detects the correct stress region.

Dobrescu et al. [10] showed that the model always looks at the leaves in the image in the CNN-based plant classifier. Dobrescu et al. extended their work, and in [11] they used layerwise relevance propagation and GBP to explain the learning of intermediate layers of the CNN model counting the leaves in an image. The authors showed that only the object's edges contribute to the model prediction, and covering the object's area does not affect the results. Escorcia et al. [13] studied the visualization of the leaf features and found the existence of attribute-centric nodes, which, rather than learning attributes, learns to detect objects. In a more recent work Lu et al. [25] proposed an explainable leaf counting framework. They used guided upsampling and background suppression to improve models' performance. However, their explanation was limited to the visualization of the instances that was responsible for the count.

Overfitting in Deep Learning. Toneva et al. [43] explained the learning of the CNN models in terms of forgetting patterns, where at some point during the training, the model correctly predicts an example, but eventually, it is misclassified. They experimented with several benchmark datasets and empirically proved that some examples are forgotten more frequently than others. Omitting such examples from the training dataset does not affect the model's performance. Arpit et al. [3] studied overfitting in deep learning models by examining the model's performance on random labels and true labels. They found that overfitting depends on the model architecture, optimization process, and data itself. They also concluded that deep learning models initially tend to learn patterns rather than memorize input samples and corresponding labels. Feldman [14] took a different approach and demonstrated that when there are numerous instances of rare examples in the dataset, the deep learning models must memorize the labels to achieve state-of-the-art performance. Feldman and Zhang [15] showed that along with memorizing outliers, the deep learning models also memorize training examples and if there are testing examples similar to it and hence overparameterized models perform extraordinarily.

Nagarajan and Kolter [30] showed that the weight norms of the model increase with the number of training examples. Due to the weight norms, the bounds increase with the increase of training examples of small batch sizes, and the generalization error decreases. Rice et al. [36] proposed that overfitting affects the model's performance in an adversarial network and observed that early stopping outperforms other methods. Salman and Liu [37] claimed that overfitting is caused due to the continuous update of a deep learning model's gradient and scale sensitiveness of the loss function. They also proposed a consensus-based classification algorithm for limited training examples.

In this paper, we complement these results by systematically analyzing the visualizations of the features learned by an overfit CNN classification model.

3. Technical Background

Guided Backpropagation. The GBP is a gradient-based visualization technique that visualizes the gradient with respect to images when backpropagating through the Relu activation function [42]. GBP allows the flow of only the positive gradients by changing the negative gradients' values to zero. This allows visualizing the image features that activate the neurons. Let f be the feature map of any layer l then the forward pass is $f_i^{l+1} = Relu(f_i^l, 0)$. Since GBP only allows the flow of positive gradients, the backward pass of the GBP is $R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot (R_i^{l+1})$, where R is an intermediate result on the calculation of the backpropagation for layer l. The final output of the GBP is an image of the same dimension as the input, displaying the features of the input image that maximized the activation of the feature maps. A major advantage of GBP is that it works for both convolutional layers and fully connected layers. Figure 1 shows some examples of the visualization generated by GBP for the Weedling dataset using ResNet-50 [21]. The grey color in the output of the GBP images (Figure 1) represents that the features in those positions of the input image do not contribute to the prediction.

Structural Similarity Index Measure (SSIM). The SSIM is a measurement of the similarity between two images [22]. To measure the SSIM, the image is divided into different windows of the same shape, and the similarity (based on mean and variance) of different windows are averaged to calculate the final SSIM. An SSIM value ranges from 0 to 1, where a value of 0 indicates that the images are very dissimilar, and an value of 1 represents that the image is highly similar. We used the SSIM function available in the python library skimage to compute the SSIM values [46]

4. Hypothesis

In a CNN, it is expected that the convolutional layers will learn features from the foreground objects in images that are being classified. The background features are considered irrelevant, and often these features are not consistent in the images. One of the expected behaviors of an overfit model is that it extracts some features from the background of the image. As a result, it performs well for the training images but fails to classify the testing images due to the absence of the features that were present in the training set. Although this is widely believed, no formal exploration has been done in the plant phenotyping context. We thus examine the following hypothesis, which will potentially help elucidate overfitting behavior of a model from the feature visualizations of its different layers.

H1: An overfit model learns from the background of the images.

Models with a large number of layers have a very high representational capacity, and therefore tend to overfit on training sets with small number of samples. In such a case, features learned in deeper layers may not be useful for learning the task due to over capacity in the model. So we explored the following hypothesis.

H2: In a model with a large number of convolutional layers, the feature visualizations obtained from the shallow layers are more diverse than those from the deeper layers. Furthermore, the diversity of the feature visualizations at a deeper layer is larger in a balanced model compared to those in an overfit model.

5. Methodology

We use the GBP approach to visualize the features learned by the intermediate layers of a CNN (e.g., see Figure 1). For every layer, GBP creates an RGB image with the same shape of the input image representing the learned features. Figure 2 depicts pairwise SSIM matrices for ResNet-50 model on different datasets, i.e., each entry (i, j) denotes the SSIM value between the GBP visualizations obtained for the ith and jth convolutional layer of ResNet-50. Here a darker red indicates higher SSIM. From the color-coding, we can observe that the pairwise SSIM is much lower at the initial layers compared to the layers at a deeper layer. This inspired us to find a way to separate the initial (dissimilar) layers from the later (similar) layers. Let $L_1, L_2, ..., L_n$ be the GBP visualization for different convolutional layers of a CNN model with n convolutional layers. The intuition is that the number k, where $1 \leq k \leq n$, with the best separation between $\{L_1, \ldots, L_k\}$ and $\{L_{k+1}, \ldots, L_n\}$ would

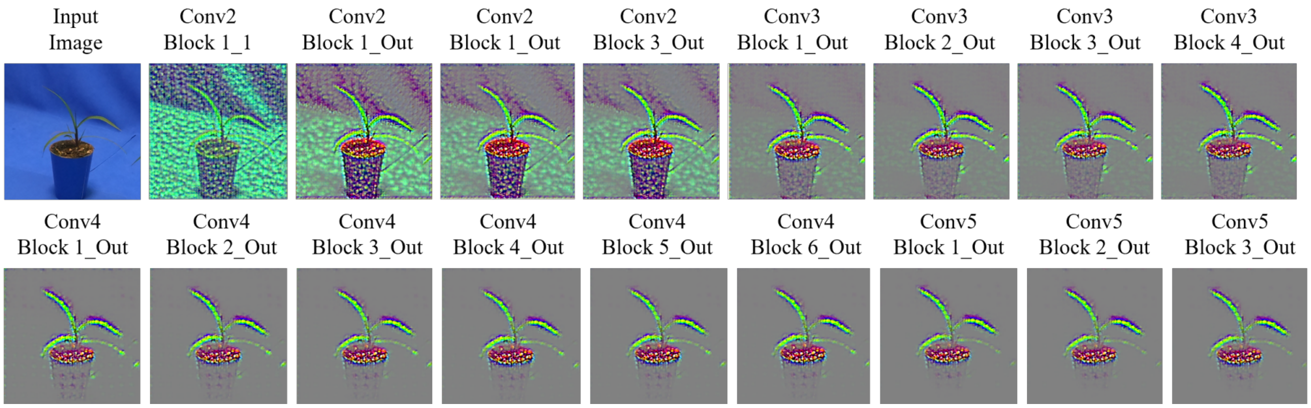


Figure 1. Visualization of the learning of the intermediate layers of ResNet-50 using GBP for Barnyard Grass of the Weedling dataset [6, 7]

suggest a reasonable depth for the model to have a good performance.

Given a number k (i.e., a cut position), we first define a *SSIM cut value* $\mathcal{C}_k$ to obtain an estimation of how good the cut is for the value k. We define $\mathcal{C}_k$ to be the mean pairwise similarity between $\{L_1, \ldots, L_k\}$ and $\{L_{k+1}, \ldots, L_n\}$:

$$\mathcal{C}_k = \frac{1}{k(n-k)} \sum_{i=1}^{k} \sum_{j=k+1}^{n} s_{i,j},$$

where $s_{i,j}$ is the SSIM between L_i and L_j. In the rest of the paper, we will refer to the function $\mathcal{C}_k$ with respect to k as the *SSIM cut curve*. If the hypothesis **H2** holds, then one can expected the SSIM curve to have a sharp positive slope for low values of k, whereas the slope would flatten for higher cut positions. The cut position where the SSIM cut curve starts to flatten rapidly (elbow of the curve) is expected to provide us with the required depth estimation.

We can observe this phenomenon better by examining the rate of change, as follows. Let M_i be the sum of the SSIM values of L_i with all other layers. Then $\mathcal{C}_k$ can be rewritten as $\mathcal{C}_k = \frac{1}{k(n-k)} \left(\sum_{i=1}^{k} M_i - \sum_{i=1}^{k} \sum_{j=1}^{k} s_{i,j} \right)$. If the curve appears to be flat around the middle cut positions, i.e., when $k \approx (n-k)$, then $\Delta\mathcal{C}_k = \mathcal{C}_{k+1} - \mathcal{C}_k = 0$. In other words, we will have $\Delta\mathcal{C}_k \approx M_{k+1} - 2\sum_{i=1}^{k} s_{i,k+1} = 0$, and hence $\sum_{i=1}^{k+1} s_{i,k+1} = \frac{1}{2} M_{k+1}$. Thus the similarity of L_{k+1} with the earlier layers $\{L_1, \ldots, L_k\}$ will be equal to its similarity with the rest of the layers $\{L_{k+1}, \ldots, L_n\}$.

6. Datasets

The use of deep learning in plant phenotypic tasks are gradually gaining popularity [1, 45, 38, 2], and the dataset plays a vital role as it contains a large amount of noise representing the real-world scenarios. Manually measuring the plant traits is a time-consuming process, which is also prone to error. Image-based automated plant trait analysis

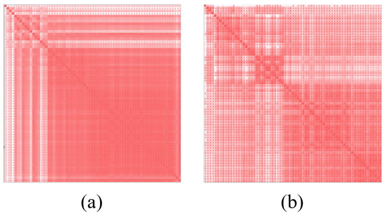

(a) (b)

Figure 2. SSIM matrix ($s_{i,j}$) generated with GBP images for (a) Barnyard Grass of the Weedling dataset (b) Apple leaf of the Plant Village dataset using ResNet-50. A darker red indicates higher SSIM.

using deep learning can help overcome these drawbacks [1]. However, most of the studies explaining the deep learning models use benchmark datasets (e.g., MNIST [9], Fashion-MNIST [49], and so on), and very few studies have attempted to explain the learning using a plant dataset [11]. Analyzing the results using plant datasets instead of the benchmark datasets will help us understand how the deep learning models are performing on dataset where the processing of the data can not be controlled and much more noise is present in the background of the images.

We used three plant datasets: Weedling dataset [6, 7], Plant Village dataset [27], and Plant Seedling dataset [19] which are commonly used for creating deep learning models for plant phenotyping tasks. For all the dataset, 80% of the available was used for training and 20% for testing.

In the Weedling dataset, there are RGB images of Barnyard Grass, Bean, Canola, Dandelion, Soybean, Canada Thistle and Smartweeds, and Wheat, which are taken in a controlled environment from different distances and angles. Researchers used the EAGL-I system to capture the images by attaching a camera to a robotic arm and placing the plants in front of a blue screen. The system automatically captured, cropped, and labeled the images to generate the dataset representing a single plant in every image. This dataset consists of 80407 training and 23535 testing images, of which 34666 are publicly available at this time at Dryad [7].

The Plant Village dataset was created by taking RGB images of nine classes of leaves, i.e., Apple, Blueberry and Cherry, Corn, Grape, Orange, Peach and Pepper, Soybean, Strawberry and Squash, and Tomato. We randomly selected training and testing images and ended up with 28693 training images and 6892 testing images.

The Plant Seedling dataset has 4739 RGB images belonging to twelve species at several growth stages. We randomly selected 3761 training and 978 testing images.

7. Deep Learning Models

ResNet-50: In this study, we used the ResNet-50 model with random weight initialization and adam optimizer as the optimization function. We also replaced the top layer of the model with a fully connected layer with Softmax activation function and neurons representing the number of classes for the classification. We trained the model for 100 epochs and only used the model with the highest testing accuracy.

2-Conv-ResNet: Keras ResNet-50 model is an implementation of the architecture proposed by He et al. [21], where the authors used five convolutional blocks. However, we also used a smaller version of the ResNet-50, where we only used the layers in Conv1 and Conv2_x (see Table 1, He et al. [21]). Apart from discarding the convolutional blocks Conv3_x, Conv4_x, and Conv5_x, the rest of the architecture remained the same. We used this 2-Conv-ResNet to see whether decreasing the depth helps avoid overfitting.

ResNet-50-10% and 2-Conv-ResNet-10%: In an attempt to create overfit models for this study, we trained the ResNet-50 and 2-Conv-ResNet on 10% training data for the Weedling and Plant Village dataset; but we left out the Plant Seedling dataset due to its small size.

Shallow CNN: Along with the ResNet-50, we also used two shallow CNN models for our experiments: one with 6 convolutional layers and the other with 13 convolutional layers, which we named **Shallow CNN, 6 Layers** and **Shallow CNN, 13 Layers**, respectively. In the shallow CNN models, we only used a combination of convolutional layers and avoided using the residual connection. The purpose of these models is to examine whether the observations obtained from the comparative analysis between ResNet-50 and 2-Conv-ResNet also hold for shallow CNN models.

For the shallow CNNs, we used categorical cross-entropy as the loss function, random weight initialization, and adam optimizer to optimize the models. Similar to ResNet models, we trained shallow models for 100 epochs with a minibatch size of 16, and chose the model with maximum testing accuracy. While training the shallow CNNs on the Weedling dataset, we resized the images to 512×512. For the other datasets, the size of the images was 224×224, as it is required for the ResNet models. We used varying zoom range, image flipping, distorting images along an axis (shear angle) for data augmentation, and added an additional batch of augmented images during each epoch. The model architecture and more details of the shallow CNN models are in the supplementary document[1].

The training and testing accuracy of different models on the different datasets is in Table 1. If the difference between training and testing accuracy was more than 10%, we considered the model an overfit model; otherwise, we considered it a balanced model. For the Weedling and Plant Village dataset, both the ResNet-50-10% and 2-Conv-ResNet-10% were overfit. All the CNN models for the Plant Seedling dataset were overfit except the shallow CNN with 13 convolutional layers, which had a very poor accuracy indicating the model was not optimized for the classification.

Table 1. Performances of different models for various datasets.

Dataset Name	Model Name	Training Accuracy (%)	Testing Accuracy (%)
Weedling	ResNet-50	98.70	96.70
	ResNet-50-10%	99.89	50.70
	2-Conv-ResNet	99.88	95.53
	2-Conv-ResNet-10%	99.89	52.10
	Shallow CNN, 6 Layers	94.00	89.60
	Shallow CNN, 13 Layers	96.23	95.45
Plant Village	ResNet-50	98.59	98.04
	ResNet-50-10%	87.99	77.93
	2-Conv-ResNet	99.25	99.17
	2-Conv-ResNet-10%	90.91	82.57
	Shallow CNN, 6 Layers	98.26	96.46
	Shallow CNN, 13 Layers	96.96	96.46
Plant Seedling	ResNet-50	91.26	81.90
	2-Conv-ResNet	85.16	68.75
	Shallow CNN, 6 Layers	90.51	76.79
	Shallow CNN, 13 Layers	68.41	69.22

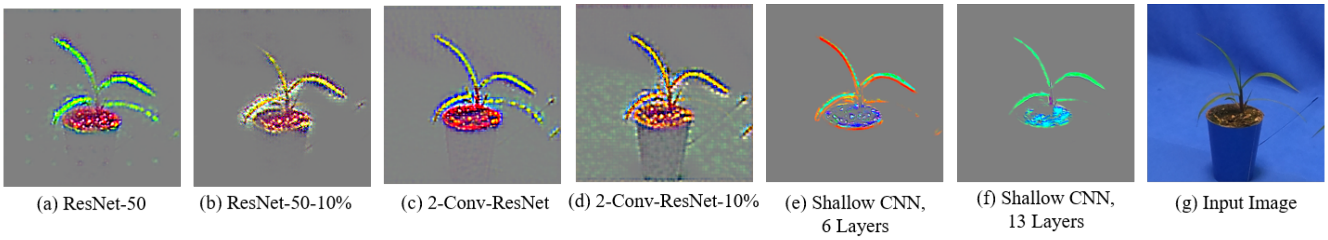
8. Result and Discussion

8.1. Learning of Intermediate Layers

A CNN model is expected to perform better when it extracts features from the foreground of the images [50]. Hence we first examined GBP visualization of the features being learned by the intermediate layers in various models.

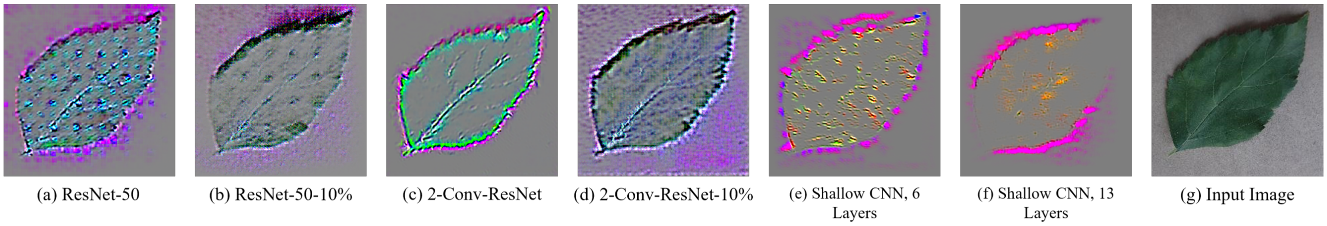
Figure 3 shows the GBP visualization of the last convolutional layer of different CNN models for the Barnyard Grass of the Weedling dataset. From Table 1, we can see that both the ResNet-50-10% and 2-Conv-ResNet-10% are

[1] https://github.com/SakibMostafa/CVPPA_PID_0041.git



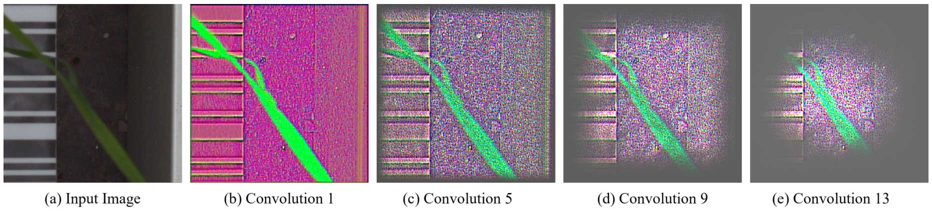
(a) ResNet-50 (b) ResNet-50-10% (c) 2-Conv-ResNet (d) 2-Conv-ResNet-10% (e) Shallow CNN, 6 Layers (f) Shallow CNN, 13 Layers (g) Input Image

Figure 3. GBP visualization of the last convolutional layer of different CNN models for the Barnyard Grass of the Weedling dataset.



(a) ResNet-50 (b) ResNet-50-10% (c) 2-Conv-ResNet (d) 2-Conv-ResNet-10% (e) Shallow CNN, 6 Layers (f) Shallow CNN, 13 Layers (g) Input Image

Figure 4. GBP visualization of the last convolutional layer of different CNN models for the Apple leaf of the Plant Village dataset.



(a) Input Image (b) Convolution 1 (c) Convolution 5 (d) Convolution 9 (e) Convolution 13

Figure 5. GBP visualization of the convolutional layers of the Black grass of the Plant Seedling dataset for Shallow CNN, 13 Layers.

overfit models. Figure 3 reveals that an overfit model may fail to extract relevant features. In particular, the 2-Conv-ResNet-10% learned features from the image background. For ResNet-50-10%, the background's contribution is not prominent, but it could not properly extract features from the top left leaf and bottom-right leaf. Although the model did not extract all the relevant features, it still achieved 99.89% training accuracy, which indicates that the model might be relying on the features in the background of the training data. However, due to the absence of the background features in the Figure 3 (b), hypothesis **H1** remains inconclusive.

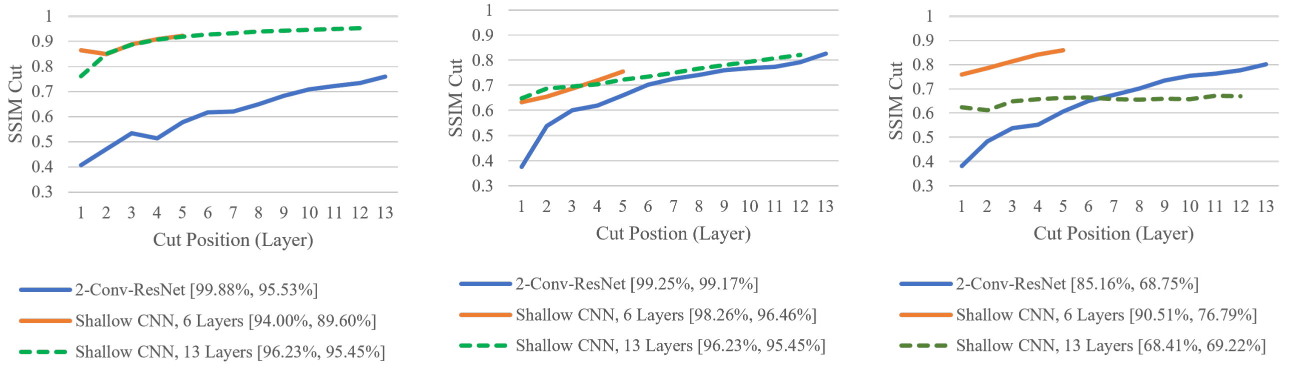
Figure 4 shows the GBP visualization of features learned by the last convolutional layer of different CNN models for the Plant Village dataset. The background of the images in this dataset consists of a grainy texture, e.g., see Figure 4. Both ResNet-50-10% and 2-Conv-ResNet-10%, which are overfit models, extracted features from such grainy pixels. This supports the observation of the Weedling dataset. The 2-Conv-ResNet had the highest classification accuracy. Analyzing the output shows that the model extracted features from the edge of the leaf, and very little contribution of the background is present. This is consistent with the finding of Dobrescu et al. [21], where covering the leaf areas did not affect the performance of regression neural networks.

For both datasets, the Shallow CNN with 6 Layers and Shallow CNN with 13 Layers were balanced models. They achieved reasonable performance, and the influence of background features appeared to be smaller when compared with the overfit models.

8.2. Contribution of Model Depth to Performance

When designing a CNN model, a common practice is to increase the depth of the model to achieve better performance. In this experiment, we studied whether increasing the depth of the model helps learn better features. For every class in a dataset, we randomly selected an image from the testing set and calculated the SSIM cut values for the images (see Section 5). Next, for every layer of a CNN model, we averaged the SSIM cut values over all the images. Thus for every model, we ended up with an SSIM cut curve.

The SSIM cut curve resembles the 'elbow method', commonly used in cluster analysis [23] to choose the number of clusters that optimize the clustering cost. For the SSIM cut curve, the elbow of the curve is a point when moving the cut position more to the right no longer improves the SSIM cut value significantly. Figure 7 shows the SSIM cut curves of ResNet-50 for different datsets. Initially, every SSIM cut curve shows a sharp positive slope, which indicates the feature visualizations for the initial layers are very dissimilar



(a) Weedling (b) Plant Village (c) Plant Seedling

Figure 6. Comparison of the Cut Position (Layer) VS SSIM cut curve for the 2-Conv-ResNet, Shallow CNN, 6 Layers, and Shallow CNN, 13 Layers for different datasets. The value in the legend of the chart indicate the training and testing accuracy of the model.

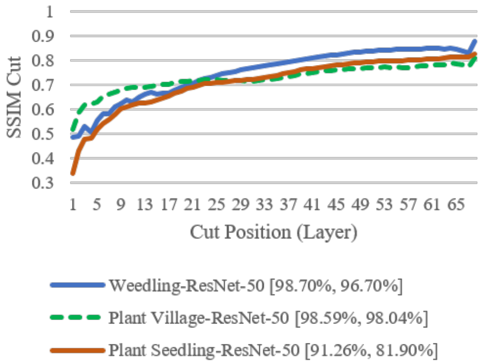


Figure 7. Comparison of the Cut Position (Layer) VS SSIM cut curve for the ResNet-50 models for different datasets. The value in the legend of the chart indicate the training and testing accuracy of the model.

from the rest of the layers. The slope becomes flatter with the increase in cut position, which supports hypothesis **H2**.

To evaluate whether the SSIM cut curve's elbow point could be used as a guide for selecting the depth of the model, we examined the performances of truncated ResNet-50 (i.e., 2-Conv-ResNet) for the same datasets. We observed that (Table 1) 2-Conv-ResNet achieved similar performance when compared with ResNet-50 for the Weedling dataset and even better performance for the Plant Village dataset. For the Seedling dataset, both the ResNet-50 and 2-Conv-ResNet remained overfit.

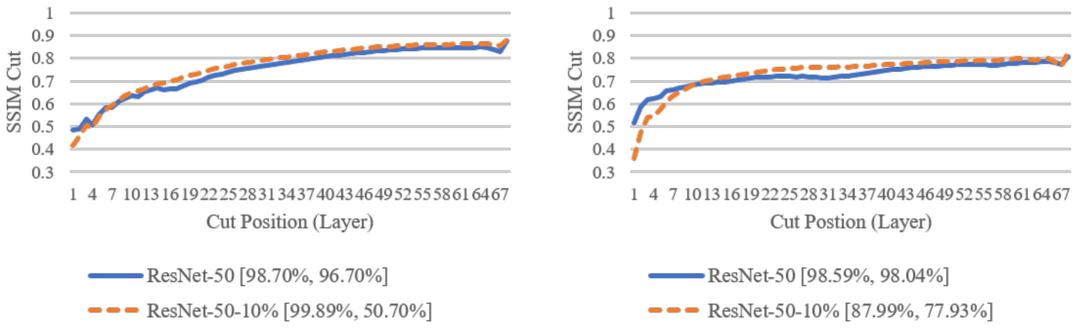
To examine whether shallow models could achieve high performances (as we have seen above for 2-Conv-ResNet), we compared the SSIM cut curve for the 2-Conv-ResNet, Shallow CNN with 6 Layers, and Shallow CNN with 13 layers (Figure 6). For the Weedling and Plant Village datasets, the Shallow CNN models achieved comparable performance to the ResNet-50 models. Furthermore, the Shallow CNN models with 6 layers performed similarly to CNN models with 13 layers. In both cases, we observed a steady increase in the SSIM cut value. For the Seedling dataset, the Shallow CNN with 13 layers performed poorly and relied on the background features (Figure 5). The Shallow CNN with 6 layers was overfit, but its training and test accuracy were higher than Shallow CNN with 13 layers.

To examine the diversity of the feature visualizations between balanced and overfit models (**H2**), we compared SSIM cut curve of ResNet-50, and ResNet-50-10% models for the Weedling and Plant Village dataset (Figure 8). For both the datasets, ResNet-50 models were balanced and ResNet-50-10% models were overfit. In both cases, the SSIM curve of the overfit model had a sharper positive slope initially, which suggests an earlier elbow point. The similar trend can also be seen for the per class analysis in Figure 9.

In summary, our experimental results suggest that some overfit models learn additional features from the image background, which provides some evidence towards **H1**. However, we also found overfit models where the contribution of the image background was not clearly visible in GBP visualizations. Hence **H1** remains inconclusive. Our analysis of the SSIM curve shows that the GBP visualizations of the initial convolutional layers of a model are much more diverse than the GBP visualizations for the deeper layers (**H2**). Furthermore, the features learned by balanced models at deeper layers are more diverse (i.e., has smaller SSIM cut values) than overfit models. The rate of change and the SSIM cut curve's elbow point can potentially provide some insight into whether a model could be designed with a smaller depth.

9. Limitation and Future Work

The results of the visual analysis is subject to human perception and interpretation. Quantifying the contribution of the background to asses its impact can be a good way to move forward. Also, including more datasets and examin-



(a) Weedling (b) Plant Village

Figure 8. Comparison of the Cut Position (Layer) VS SSIM cut curve for the ResNet-50, and ResNet-50-10% models for different datasets. The value in the legend of the chart indicate the training and testing accuracy of the model.

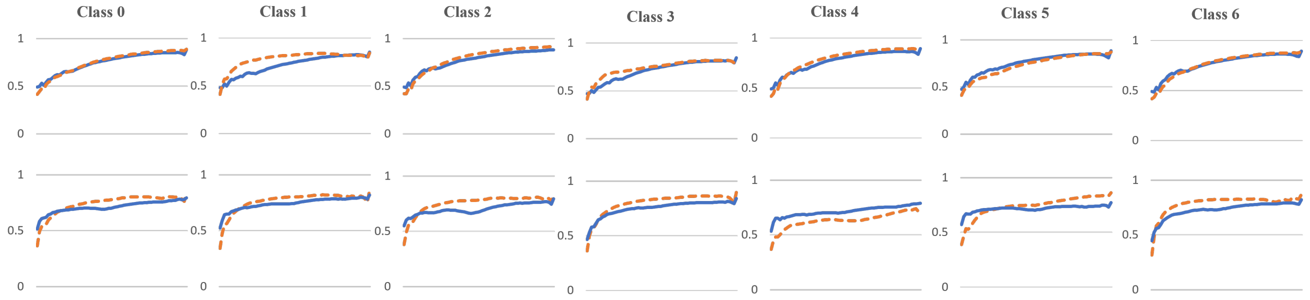


Figure 9. Comparison of the Cut Position (Layer) VS SSIM cut curve for different classes of the ResNet-50 (blue), and ResNet-50-10% (orange) models for (top row) Weedling and (bottom row) Plant Village dataset.

ing other CNN models could further strengthen our results. It would also be interesting to explore other feature-map visualization techniques in this context. In our SSIM cut curve analysis, the elbow point may not always correspond to a sharp elbow or be identified unambiguously in practice, which is a commonly known limitation of elbow heuristics [23]. We also envision to run a user study involving deep learning experts, where one can show the output of different models by hiding the labels and record their opinions to see whether it is possible for a domain expert to detect an overfit model by only observing the GBP visualization of the intermediate layers. Due to the presence of the residual connection in the ResNet models, there might be a possibility that it may avoid overfitting and the influence similarity of the GBP visualizations of various layers. Hence it would be interesting to investigate the contribution of the residual connections in an overfit model's performance. Another direction of the future experiments would be to investigate the SSIM cut curve at different granularity of the depth of the model.

10. Conclusion

In this paper, we explained the overfitting in a CNN model for plant phenotyping by visualizing the intermediate layers' learning. We used guided backpropagation to visualize the learning of the intermediate layer of different CNN models. We used four different models on three different plant classification datasets, and our experiments showed that an overfit model sometimes relies on the background of the images. We proposed a novel SSIM cut based analysis to measure the similarity among the features learned in the intermediate layers of a CNN. Our SSIM cut curve revealed that in a more complex model, the shallow layers learn more diverse features as compared to the deeper layers and that a more distinct transition between these regimes is noticeable for overfit models. The SSIM cut curve method can help detect a potential overfit condition or inform a practitioner that a shallower model may be more appropriate for training with a particular dataset. We believe our study contributes to better understanding of the behaviour of overfit CNN models and provides new directions for creating metrics to detect and avoid model overfitting in plant phenotyping tasks.

Acknowledgement

This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund.

References

[1] Shubhra Aich, Anique Josuttes, Ilya Ovsyannikov, Keegan Strueby, Imran Ahmed, Hema Sudhakar Duddu, Curtis Pozniak, Steve Shirtliffe, and Ian Stavness. Deepwheat: Estimating phenotypic traits from crop images with deep learning. In *2018 IEEE Winter conference on applications of computer vision (WACV)*, pages 323–332. IEEE, 2018. 4

[2] Shubhra Aich and Ian Stavness. Leaf counting with deep convolutional and deconvolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2080–2089, 2017. 4

[3] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017. 2

[4] Muhammad Azfar Firdaus Azlah, Lee Suan Chua, Fakhrul Razan Rahmad, Farah Izana Abdullah, and Sharifah Rafidah Wan Alwi. Review on techniques for plant leaf classification and recognition. *Computers*, 8(4):77, 2019. 1

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 1

[6] Michael A Beck, Chen-Yi Liu, Christopher P Bidinosti, Christopher J Henry, Cara M Godee, and Manisha Ajmani. An embedded system for the automated generation of labeled plant images to enable machine learning applications in agriculture. *Plos one*, 15(12):e0243923, 2020. 4

[7] Michael A. Beck, Chen-Yi Liu, Christopher P. Bidinosti, Christopher J. Henry, Cara M. Godee, and Manisha Ajmani. Weed seedling images of species common to manitoba, canada. *PLOS ONE*, 15:1–23, 12 2020. 4, 5

[8] Akshay L Chandra, Sai Vikas Desai, Wei Guo, and Vineeth N Balasubramanian. Computer vision with deep learning for plant phenotyping in agriculture: A survey. *arXiv preprint arXiv:2006.11391*, 2020. 1, 2

[9] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 4

[10] Andrei Dobrescu, Mario Valerio Giuffrida, and Sotirios A Tsaftaris. Leveraging multiple datasets for deep leaf counting. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2072–2079, 2017. 2

[11] Andrei Dobrescu, Mario Valerio Giuffrida, and Sotirios A Tsaftaris. Understanding deep neural networks for regression in leaf counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 4

[12] Mads Dyrmann, Henrik Karstoft, and Henrik Skov Midtiby. Plant species classification using deep convolutional neural network. *Biosystems engineering*, 151:72–80, 2016. 1

[13] Victor Escorcia, Juan Carlos Niebles, and Bernard Ghanem. On the relationship between visual attributes and convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1256–1264, 2015. 2

[14] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020. 2

[15] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *arXiv preprint arXiv:2008.03703*, 2020. 3

[16] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019. 1

[17] Sambuddha Ghosal, David Blystone, Asheesh K Singh, Baskar Ganapathysubramanian, Arti Singh, and Soumik Sarkar. An explainable deep machine vision framework for plant stress phenotyping. *Proceedings of the National Academy of Sciences*, 115(18):4613–4618, 2018. 2

[18] Scott Gigante, Adam S Charles, Smita Krishnaswamy, and Gal Mishne. Visualizing the phate of neural networks. *arXiv preprint arXiv:1908.02831*, 2019. 1

[19] Thomas Mosgaard Giselsson, Mads Dyrmann, Rasmus Nyholm Jørgensen, Peter Kryger Jensen, and Henrik Skov Midtiby. A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. *arXiv preprint*, 2017. 4

[20] Anirban Jyoti Hati and Rajiv Ranjan Singh. Artificial intelligence in smart farms: Plant phenotyping for species recognition and health condition identification using deep learning. *AI*, 2(2):274–289, 2021. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5

[22] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 3

[23] David J Ketchen and Christopher L Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458, 1996. 6, 8

[24] Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598*, 2017. 2

[25] Hao Lu, Liang Liu, Ya-Nan Li, Xiao-Ming Zhao, Xi-Qing Wang, and Zhi-Guo Cao. Tasselnetv3: Explainable plant counting with guided upsampling and background suppression. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–15, 2021. 2

[26] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017. 1

[27] S. P. Mohanty. Plant village. `https://https://github.com/spMohanty/PlantVillage-Dataset`, 2018. 4

[28] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019. 1

[29] Sakib Mostafa and Debajyoti Mondal. On the evolution of neuron communities in a deep learning architecture. *CoRR*, abs/2106.04693, 2021. 1

[30] Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 3

[31] Koushik Nagasubramanian, Sarah Jones, Asheesh K Singh, Soumik Sarkar, Arti Singh, and Baskar Ganapathysubramanian. Plant disease identification using explainable 3d deep learning on hyperspectral images. *Plant methods*, 15(1):1–10, 2019. 2

[32] Seong Joon Oh, Bernt Schiele, and Mario Fritz. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 121–144. Springer, 2019. 1

[33] Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. Deep learning for multi-task plant phenotyping. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2055–2063, 2017. 2

[34] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999. 1

[35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 1

[36] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020. 3

[37] Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networks. *arXiv preprint arXiv:1901.06566*, 2019. 3

[38] Hanno Scharr, Massimo Minervini, Andrew P French, Christian Klukas, David M Kramer, Xiaoming Liu, Imanol Luengo, Jean-Michel Pape, Gerrit Polder, Danijela Vukadinovic, et al. Leaf segmentation in plant phenotyping: a collation study. *Machine vision and applications*, 27(4):585–606, 2016. 4

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[40] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016. 1

[41] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[42] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1, 2, 3

[43] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 2

[44] F-Y Tzeng and K-L Ma. *Opening the black box-data driven visualization of neural networks*. IEEE, 2005. 1

[45] Jordan R Ubbens and Ian Stavness. Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks. *Frontiers in plant science*, 8:1190, 2017. 2, 4

[46] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. 3

[47] Martin Weis, Christoph Gutjahr, Victor Rueda Ayala, Roland Gerhards, Carina Ritter, and Florian Schölderle. Precision farming for weed management: techniques. *Gesunde Pflanzen*, 60(4):171–181, 2008. 1

[48] Yang WENG, Rui ZENG, ChenMing WU, Meng WANG, XiuJie WANG, and YongJin LIU. A survey on deep-learning-based plant phenotype research in agriculture. *Scientia Sinica Vitae*, 49(6):698–716, 2019. 1

[49] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 4

[50] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020. 5

[51] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

[52] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[53] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017. 2