From Lab to Field: An Empirical Study on the Generalization of Convolutional Neural Networks towards Crop Disease Detection

Felipe A. Guth, Shane Ward, and Kevin McDonnell

Abstract — **Due to complex feature abstraction and learning power, CNNs have been the most successful machine learning algorithms for image classification tasks. The objective of this work was to evaluate the potential of convolutional neural networks (CNNs) for extracting underlying complex features and recognize these patterns towards the task of detecting healthy and diseased crop plants. The generalization of these algorithms was assessed on different situations of training and testing scenarios using images from controlled lab conditions and real field environments. Results have shown that when presented with sufficient data variability in training, englobing images with similar conditions faced in testing, the deep learning architectures delivered accurate results of over 90%. In contrast, the same architectures were not able to generalize the accuracy of training towards the detection of new unseen images that were not extracted in the same settings as the ones from the training set, delivering, in this case, a general accuracy of around 50%. The deployment of practical automated support systems for disease detection depends on the provision of robust datasets for training CNNs which contemplate the spectral variability conditions found in numerous crop cultivation environments encountered in diverse field sites across the globe.**

Key words — **Convolutional Neural Networks, Crop Disease Detection, Deep Learning.**

I. Introduction

The occurrence of plant disease is a risk for global scale production from high tech farms that employ the last technologies in crop assessment and production to smallholder farmers whose livings rest on healthy crops. Moreover, as much as 80% of the agricultural production in developing countries is generated by small farms [1]. The occurrence of crop biotic stress, weeds, diseases and insects, are responsible for approximately 40% of the world food losses [2]. In precision agriculture, the phenotyping of diseases and biotic stress in crops has been the focus of studies and applications of machine learning throughout the last decades [3], [4].

The literature shows the use of different techniques, sensing technologies and analytical algorithms for disease detection. Classical approaches involved near-infrared [5], multispectral [6], [7], hyperspectral [8] and visual spectrum data [9]. These works of disease detection in crop leaves, using visible light range images, have been historically centered on the pre-processing of images and extraction of features with the best characteristic representation capacity of the sample's singularities. However, these applications are dependent on particular conditions such as illumination and background segmentation, in order to have satisfactory results [10].

Machine learning have brought a revolution to agriculture applications [11], more specifically to the assessment of diseases and biotic stress [12], thanks to the advent of Deep Convolutional Neural Networks (CNN's) [13] and GPU processing [14]. CNN's [15] are algorithms that deploy several layers of nonlinear data processing capable of extracting underlying patterns from the input data, turning classical approaches that rely mainly in precise handcrafted feature extraction unnecessary [16].

In the crop disease detection domain [12] a number of deep learning techniques have been demonstrating considerate potential in image classification. These algorithms have been proposed towards the classification of diseases in crops such as apples [16], tomato [17], tea [18], aubergine [19] and wheat [20. The works of [12], [21] have demonstrated the potential of CNN's towards crop disease detection by using two of the most popular architectures: AlexNet [22] and GoogLeNet [23] using as input 54.306 images of the PlantVillage [24] dataset that are distributed into 26 diseases types across 14 crop plants species. In an innovative approach, rather than resizing images to a reduced size and training a standard forward model end-to-end, DeChant *et al*. [25] created a pipeline of several CNNs that had prediction combine into heat maps that were fed into a final CNN trained to classify the presence of leaf blight in maize plants. The work of Barbedo [10] made use of data augmentation and transfer learning strategies in order to increase a plant disease dataset considering 14 species of plants and 79 diseases that were effectively classified with the use of CNN's that reported an increase of 12% on accuracy if compared to a previous classical approach. Rahman et *al.* [26] have successfully tested the accuracy of the CNN VGG-16 [27] towards the detection of diseases and pests in rice plants which was used as a benchmark to be compared to more memory efficient CNN's architectures targeting mobile devices use.

In this research study, three state-of-the-art CNN architectures: VGG16, Inception-V3 and ResNet-152 were

Submitted on March 07, 2022.
Published on March 28, 2023.
F. A. Guth, University College Dublin, School of Biosystems and Food Engineering, Ireland.
(e-mail: felipe.guth@ucdconnect.ie)
S. Ward, University College Dublin, School of Biosystems and Food Engineering, Ireland.
(e-mail: shane.ward@ucd.ie)
K. McDonnell, University College Dublin, School of Agriculture and Food Science, Ireland.
(e-mail: kevin.mcdonnell@ucd.ie).

trained for the task of detecting plant disease based on standard RGB images collected in controlled lab and real field conditions. A dataset of 12 classes of diseased and healthy plants was built and extended from the public available dataset PlantVillage [24]. More specifically, this work focused on assessing the generalization capabilities of CNNs for learning clear patterns from lab conditions to be detected again on new and more complex field conditions while avoiding overfitting.

II. Materials and Methods

A. Dataset

For the development of this work, a dataset of images extracted from the Plant Village portal was created [24]. This portal offers open access to more than 50,000 digital images of different cultures under both healthy and diseases conditions. The objective of this portal is to help solve the problem of open data in the area of disease detection in crops through the use of ML algorithms, especially those of Deep Learning that require a large amount of data.

In this study, images selected from controlled conditions (lab) were separated from the ones acquired under field conditions for the same classes. In this way, it was built two distinct datasets, the first one containing only RGB pictures from a lab environment were obtained under a controlled environment with uniform illumination and uniform background, while the second one contains both lab and field conditions images. Fig. 1 depicts the different types of crop/diseases contained on the dataset. In total, 12 classes of diseased or healthy were subdivided accordingly to the numbers brought by Table I.

B. Classes Imbalance and Data Augmentation

In the training phase, neural networks apply the backpropagation of error algorithm that entails calculating errors produced by the model on the training dataset and updating the model weights in proportion to those errors. The drawback of this technique of training is that samples from each class are handled in the same way, which for imbalanced datasets implies that the model is fitted in a much more significant way for classes with larger number of samples.

There are different methods to handling the imbalance of class imbalance with the backpropagation algorithm that alter the default calculations in a way that the model is allowed to pay more attention to examples from the minority class than the majority class in datasets with a severely skewed class distribution. Another way of resolving this problem, is simply acquiring more data, or synthetically generating more data samples with data augmentation processes to normalize the data distribution between the dataset classes.

For the training of deep learning algorithms in classification tasks, it is essential to deploy a dataset that comprises a sufficient number of elements, as well as in-class variability. This aspect fosters the production of a robust system provided with generalization aptitude, in order to classify, in a correct way, unseen samples of the same class in different conditions through the detection of key patterns previously learned.

Image augmentation processes are widely adopted practices in machine learning applications that require many samples for training and evaluation. This process of dataset expansion is based both on the purpose of providing enough elements for the algorithm convergence, as for avoiding overfitting, by introducing variability into the distribution of the data samples. This technique was applied to the original images after the separation between the training and testing sets. The images were assigned to the two sets in a random mode, without any special treatment for differentiation of the sample's characteristics.

Fig. 1. A sample of each class of crop/disease on the dataset; the images follow the order of Table I, where a sample from lab environment is followed by a sample of the same class captured on real field environments.

TABLE I: Description of The Classes System Classes Set Alongside The Original Number of Elements and The Extended Number Obtained After The Process of Data Augmentation, Which Powered The CNN Training and Testing

Class	# original images			# training set (images augmented)		# testing set		
	Lab	Field	Lab +Field	Lab	Lab +Field	Lab	Field	Lab +Field
Apple black roat	280	431	711	3960	11880	28	43	71
Apple healthy	713	1123	1836	4192	12576	71	112	183
Blueberry healthy	586	1149	1735	4102	12306	58	115	173
Corn grey leaf spot	191	1266	1457	3938	11814	19	127	146
Corn common rust	519	1095	1614	3932	11796	52	110	162
Corn healthy	445	4005	4450	3915	11745	44	401	445
Grape healthy	171	442	613	3917	11751	17	44	61
Pepper – bell healthy	577	1454	2031	4039	12117	58	145	203
Potato early blight	420	2747	3167	3940	11820	42	275	317
Soybean healthy	1917	4318	6235	4200	12600	192	432	624
Strawberry leaf scorch	390	3006	3396	3930	11790	39	301	340
Tomato early blight	405	2174	2579	3935	11805	41	217	258
Total	6614	23210	29824	48000	144000	661	2321	2982

The implementation of the image augmentation process was coded using a Python script. Spatial augmentation operations were performed for determined classes images, involving cut, translate and scale operations. Example of such transformations are shown in Fig. 2, where a sample class is presented in its original state along its subsequent transformations. The listed augmentation operations were chosen to simulate different perspectives and points of view also found in natural environments.

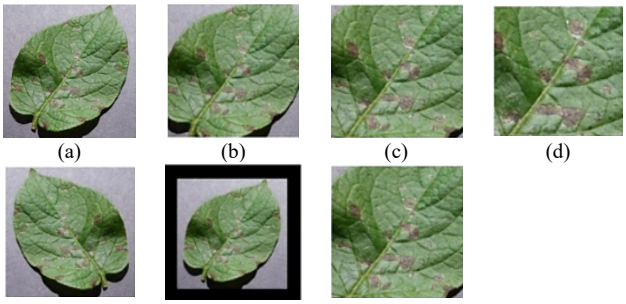

Fig. 2. The data augmentation process generates sample variability: a) Original image; b) Crop 1; c) Crop 2; d) Crop 3; e) Flip; f) Scale 0.8; g) Scale 1.6.

Different data-augmentation strategies were applied for the two training datasets of the current work. For the lab only dataset, the images were simply augmented to generate a number of samples that was close to the class with larger number of images. For the 'lab+field' dataset, for each class the number of lab images were augmented to match the number of field images (which were always greater) until a certain number of interclasses approximation.

In this way, the overall distribution of classes was not skewed as also the samples for each class were keep proportional between lab and field conditions. In the cases were, the number of lab conditions images inside a class match the number of field condition samples but the number of total samples of the class was still under the targeted number for the overall classes' distribution, data augmentation was applied in an equal way for both cases (lab and field images) until the number was reached. Every class was assigned to meet a given sub-random number of images that was close to an assessed target, given the number of original images, that was around 4.000 for the lab only dataset and 12.000 for the 'lab+field' dataset. Table I shows the distribution in between classes for both datasets.

C. Convolutional Neural Networks Architectures

Three CNNs were used to evaluate the potential of classification of crop diseases giving the dataset categories. The accuracy of these algorithms has been demonstrated on applications such as the ImageNet challenge [28] which assesses the performance of these networks proposals that are trained on millions of images for the classification of 1000 different classes. In Table II, it is shown the three CNNs accuracy results for top-1 and top-5 classification.

1) VGG-16

VGG CNN [27] was designed targeting the reducing of number of parameters in the convolution layers and improving of training time when compared to previous CNNs implementations such as AlexNet [22].

VGG has different architectures using 12, 16 or 19 layers. The crucial aspect of this CNN is that it uses fixed kernels in a way that that all the conv kernels are of size 3 × 3 and maxpool kernels are of size 2 × 2 with a stride of two.

The concept behind applying fixed size kernels lies on the fact that all the varying size convolutional kernels utilized in Alexnet (11 × 11, 5 × 5, 3 × 3) can be reproduced by using multiple 3 × 3 kernels as building blocks. The replication is in terms of the receptive field covered by the kernels. This results in a reduced number of trainable parameters which are directly correlated to faster learning and a lowered risk of network overfitting. Fig. 3 brings the complete architecture of the VGG-16 network which was used in this work.

1) Inception v3

Inception-v3 CNN [23] is an expanded network of the widespread GoogLeNet [29] which has accomplished satisfactory results even when deploy a lesser number of parameters if compared to other state-of-the-art architectures. Subsequent to GoogLeNet, Inception-v3 proposed an inception model which concatenates numerous distinct sized convolutional filters into a large new filter.

TABLE II: CNNs Architectures and Properties

Model	Size MB	Top-1 Acc.	Top-5 Acc.	Params	Depth	Salient feature
VGG 16	528	0.713	0.901	138.357.544	23	Fixed-size kernels
Inception v3	92	0.779	0.937	23.851.784	48	Wider parallel kernels
ResNet152	244	0.779	0.943	60.344.232	152	Shortcut connections

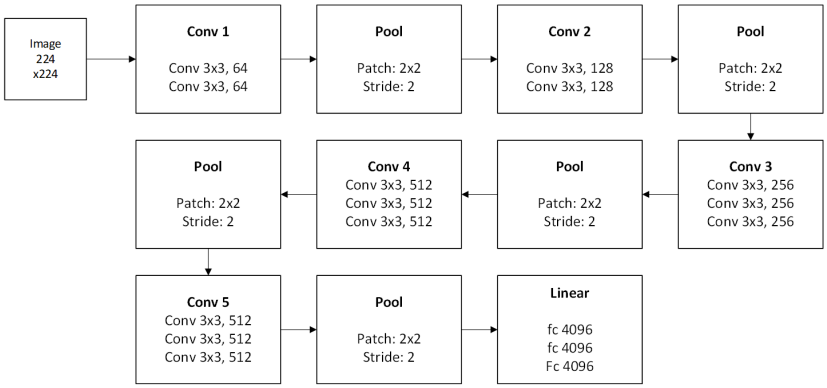

Fig. 3. The VGG-16 CNN architecture.

Larger kernels are favored for more global features that are dispersed across a large area of the image, on the other hand, smaller kernels deliver satisfactory results in identifying area-specific features that are spread across the image frame. The idea of the inception network is proposed to this idea for effective recognition of such a variable-sized feature, applying kernels of different sizes. Instead of merely running deeper in terms of the quantity of layers, this CNN also goes wider. Several kernels of different sizes are employed within the same layer.

The development of this kind of design also reduced the number of parameters to be trained and in this manner decreases the computational complexity. Inception v3 was also used in the current work, its basic architecture is shown in Fig. 4.

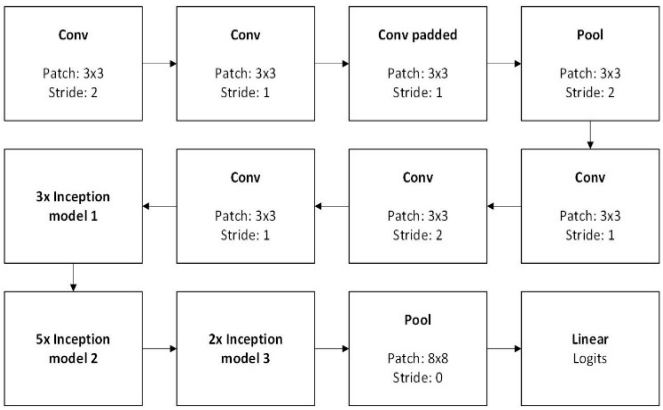

Fig. 4. Basic architecture of Inception v3.

2) ResNet-152

Residual networks (Resnet) [30] were planned as a group of several deep neural networks with analogous structures but distinct depths. The training of neural networks is based on the calculation of the error on prediction (loss) and correction of weights using the back-propagation algorithm. The issue with this is as CNN grow deeper, the derivative when back-propagating to the initial layers becomes practically insignificant in value in a problem called vanishing gradient.

Resnet proposed new building blocks, deploying a structure called residual learning unit to assuage the degradation of deep neural networks. This unit's structure is a feedforward network with a shortcut connection which adds new inputs into the network and generates new outputs. The main merit of this unit is that it produces better classification accuracy without increasing the complexity of the model. In this study, the Resnet-152 was selected as it has achieved the best accuracy among the Resnet network options. Fig. 5 illustrates the basic architecture of Resnet152.

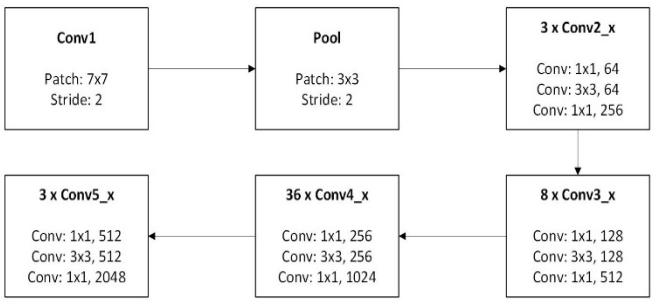

Fig. 5. Basic architecture of Resnet-152.

D. Fine Tuning

Deep Learning CNNs require a large number of elements for each class in order to extract the class particular patterns and learn the underlying features of the data to derive the correct classification model. Fine-tuning strategies enable the concept of transfer learning. The basic idea of this technique is to adapt a neural network that has already learned to recognize patterns of objects in a given problem into a new approach. In practical terms, this means restarting the weights learned by the neurons in the last layers and maintaining those of the first layers. This is based on the conclusion that the features captured by the first layers of the neural network are more generic (edges detectors, contours, and regions of colour) that are useful for most classification tasks. Whereas, while progressively advancing, more specific patterns of abstractions of the dataset classes are represented.

The number of layers restarted in a new CNN application depends primarily on the size of the new dataset and its similarity to the dataset used in the previous application. The bigger the size of the new dataset, the higher is the number of layers that can be restarted. The more similar the new dataset is with the previous one used to train the deep neural network, the lesser is the number of layers that need to be restarted.

For the training of the deep CNNs deployed in this study, the last layers made of fully connected linear layers were restarted to be fully retrained from start, while the other layers were unfrozen to update previously learn weights in a progressive way. The fine tuning of the transferred models to the new datasets occurred by slowly unfreezing the gradients of the convolutional layers as the network was trained, starting from the lowest level layers, and working its way to the top layers as training proceed. This approach supports the prevention of overfitting specially in CNNs with a high number of parameters.

E. Training Parameters and Experimental Setup

The mentioned CNNs were deployed using the PyTorch deep learning framework [31]. PyTorch is an open-source machine learning library based on the Torch library, utilized for applications such as computer vision and natural language processing, it is free and open-source software released under the Modified BSD license. PyTorch is a library for Python programs that simplifies constructing deep learning projects. It emphasizes flexibility and allows deep learning models to be coded in Python with strong GPU acceleration. The three CNNs of this study were started with PyTorch's pre-trained models on the ImageNet dataset in a fine-tuning approach. 90% of the original image samples were used for training (of which 10% for validation) couple with data augmentation approaches, while 10% of images were allocated for testing. At this stage, the accuracy of the networks was computed accordingly to the classification results of these unseen images.

The use of GPU has made applications of deep learning viable with performance far superior from CPUs based models. Deep learning comprises a massive volume of matrix multiplications and other operations that may be parallelized and thus sped up on GPU-s. A single GPU board may contain thousands of cores whereas a CPU normally has no more than 16 cores. Even tough GPU cores are slower in processing than CPU cores, they can overcome the difference in the

processing given the large number and faster memory given that the operations can be parallelized. In this work, a NVIDIA Titan RTX with 24 GB of memory, 4608 CUDA cores, 576 tensor cores and 72 RT cores has been utilized to train the deep residual network and produce the final model.

TABLE III: CNNs Training Parameters

Parameter	Value
Epoch	80
Batch Size	64
Dropout	0.5
Learning Optimizer	SGD
Learning Rate	0.0005
Wight Decay	0.0005

The networks were trained for a maximum of 80 epochs, Table III, in a fashion that a snapshot would be saved for every iteration when the error of the network was the best compared of the previous one. In this way, by the end of the training a snapshot of the weights with the best evaluation performance was saved for the final testing stage where the overall accuracy of classes was computed.

III. Results and Discussion

After the training of the three CNNs presented on section 2.3, the accuracy of each class was computed for the unseen images separated from the initial dataset that made part of the testing set. For each CNN of the three studied architectures, Table 4 presents the classes accuracy classification, an overall accuracy giving all the scenarios tested and in the last column an average figure that considers the overall accuracy of all CNNs for a given class. The five types of scenarios evaluated were:

- Lab (trained) & Lab (test): CNNs were trained only using images from lab-controlled environments and tested with unseen images from the sabe lab controlled settings.
- Lab (trained) & Field (test): CNNs were trained with images from lab conditions and tested with unseen images from field non-controlled environments. This setting offered the opportunity to CNNs prove their remarkable feature extraction robustness and generalization to classify the same crop/diseases but in much more challenging situations from what they were trained.
- Lab+Field (trained) & Lab (test): CNNs were trained with a similar number of images from lab and field conditions and were tested with images only from lab conditions. This setting was proposed to make a comparison with the results were the CNNs were only trained and tested with images from lab conditions to evaluate what would be the effect of having images with greater variability of background, illumination, and other spatial properties for the same classes.
- Lab+Field (trained) & Field (test): CNNs were trained with similar numbers of images from lab and field conditions and trained only with images from field environments. This setting was proposed to observe what would be the improvement in generalization for complex field environments predictions if compared to the case were only images from lab were used to train the network.
- Lab+Field (trained) & Lab+Field (test): CNNs were trained with a similar number of images from lab and field environments and tested with unseen samples also from lab and fields conditions. This setting was proposed to compare the CNNs accuracy results for the case where they were trained and evaluated with images with a similar environmental variability to the other cases that had images only from lab or field conditions.

By analysing the AVG accuracy that considers all results of the CNNs, it can be observed that the lowest classification accuracy obtained overall was 71.1% for tomato early blight and the maximum overall accuracy was 90.6% for grape healthy while the overall average class accuracy was of 83.1%. Examining the individual results of the CNNs it can be observed that in most classes the best accuracy results were achieved by the ResNet-152 CNN followed by Inception v3 and lastly VGG. This rank was expected as it compares to the results obtained in other benchmarks evaluations such as ImageNet. Nevertheless, is worth mentioned that as the VGG has the larger number of parameters and such as result it is more prone to overfitting the data. In this way, the accuracy results offer a good indication that the incremental unfreeze of layers of the network weights during training have worked in avoiding this problem. In case of overfitting, the VGG network would be expected to have better accuracy in most classes of the lab accuracy classification, where all the images are usually similar.

As expected, the best accuracy results were achieved in the scenario where the CNNs were trained and tested with images acquired in lab-controlled conditions. There is generally little variability in these cases, specially related to background, which makes for an easy detection of classes for the networks. In this case, in many classes instances the accuracy of the networks was equal to 100%. When the CNNs were still only tested with images from field but trained with a more complex setting, using both lab and field images, the accuracy performances decreased in average 7% for VGG, 1.5% for Inception and 1.1% for Resnet. These results have demonstrated that, even when presented with greater variability in training, adding images from non-controlled environments, the CNNs were still capable of offering good accuracy results thanks to robust learnt patterns.

The worst accuracy results were obtained in a demanding test over the generalization capability of the CNNs for learning features from one controlled environment to be used in a more complex setting. When trained only with images from lab conditions and tested only with images from field conditions, the CNNs had an average classification accuracy of 51.3%. In this case, classes such pepper bell healthy, potato early blight, soybean healthy and tomato early blight had accuracy results under 40% or even 30% of accuracy. These results compare for the ones obtained from [12] which tested a CNN trained with lab acquired images for classifying images for the same classes in field conditions. In this images that were not collected under controlled conditions, the CNN model had accuracies of 41.1 to 54.5% when classifying disease in images of a prespecified species. In images collected from natural settings, many nuisance elements contribute to make the task challenging for a network trained on artificially controlled images, including lighting variations, shadows, and exposed soil.

TABLE IV: TEST SET ACCURACY RESULTS PERFORMANCE FOR THE THREE CNN ARCHITECTURES DEPLOYED

Class		Lab (trained) & Lab (test) Acc.	Lab (trained) & Field (test) Acc.	Lab+Field (trained) & Lab (test) Acc.	Lab+Field (trained) & Field (test) Acc.	Lab+Field (trained) & Lab+Field (test) Acc.	Combined AVG Acc.	AVG CNN's Acc.
Apple black roat	VGG-16	0.929	0.349	0.893	0.791	0.859	0.764	0.829
	Inception v3	0.964	0.581	0.929	0.860	0.915	0.850	
	ResNet-152	0.964	0.628	0.964	0.884	0.930	0.874	
Apple healthy	VGG-16	0.930	0.429	0.901	0.786	0.945	0.798	0.835
	Inception v3	0.958	0.491	0.887	0.848	0.962	0.829	
	ResNet-152	0.972	0.580	0.944	0.920	0.978	0.879	
Blueberry healthy	VGG-16	0.948	0.478	0.931	0.887	0.919	0.833	0.855
	Inception v3	0.983	0.487	0.966	0.957	0.977	0.874	
	ResNet-152	0.966	0.470	0.948	0.939	0.971	0.859	
Corn grey leaf spot	VGG-16	0.789	0.262	0.842	0.786	0.762	0.688	0.734
	Inception v3	0.842	0.405	0.895	0.802	0.810	0.751	
	ResNet-152	0.842	0.413	0.947	0.825	0.794	0.764	
Corn common rust	VGG-16	0.962	0.642	0.942	0.917	0.888	0.870	0.906
	Inception v3	0.981	0.725	0.981	0.963	0.957	0.921	
	ResNet-152	0.981	0.734	0.981	0.972	0.969	0.927	
Corn healthy	VGG-16	0.955	0.398	0.909	0.823	0.896	0.796	0.838
	Inception v3	0.977	0.473	0.955	0.880	0.944	0.846	
	ResNet-152	1.000	0.498	0.977	0.930	0.964	0.874	
Grape healthy	VGG-16	0.882	0.886	0.824	0.909	0.918	0.884	0.927
	Inception v3	0.941	0.909	0.941	0.955	0.951	0.939	
	ResNet-152	1.000	0.932	0.941	0.955	0.967	0.959	
Pepper – bell healthy	VGG-16	0.931	0.214	0.914	0.876	0.926	0.772	0.818
	Inception v3	0.966	0.338	0.948	0.924	0.936	0.822	
	ResNet-152	0.983	0.379	0.983	0.966	0.980	0.858	
Potato early blight	VGG-16	0.857	0.310	0.881	0.814	0.823	0.737	0.775
	Inception v3	0.905	0.347	0.905	0.854	0.864	0.775	
	ResNet-152	0.929	0.434	0.905	0.905	0.899	0.814	
Soybean healthy	VGG-16	0.938	0.448	0.922	0.870	0.915	0.818	0.845
	Inception v3	0.953	0.497	0.927	0.903	0.944	0.845	
	ResNet-152	0.979	0.524	0.964	0.933	0.955	0.871	
Strawberry leaf scorch	VGG-16	0.949	0.677	0.923	0.877	0.926	0.870	0.899
	Inception v3	1.000	0.737	0.949	0.900	0.956	0.908	
	ResNet-152	1.000	0.770	0.949	0.910	0.971	0.920	
Tomato early blight	VGG-16	0.659	0.240	0.756	0.829	0.775	0.652	0.711
	Inception v3	0.780	0.327	0.854	0.871	0.826	0.732	
	ResNet-152	0.829	0.378	0.805	0.880	0.857	0.750	

In contrast, for the next setting where the CNNs were trained with both lab and field images and then again tested only with natural images from field, the accuracy results had a significant improvement. VGG results were improved by 40.3% on average, while Inception 36.7% and 35.7% for ResNet. This significant enhancement in accuracy results, once more supports the fact that deep learning solutions are heavily dependent on the complexity of the problem being solved [32]. For the case of disease detection, background and illumination variation in the images produced a considerate impact on results. It was noted that field images that had less background interference produced better results than the ones with a cluttered background. This observation has shown that despite of its successful implementation in image analysis tasks, CNNs are still dependent on data variability and scale. For disease detection in crops, the results have exposed that these algorithms would not have a suitable performance if presented with images of the same diseases but trained in a somewhat different environment.

Deep Learning Convolutional Networks are notable in part for their capacity of learning complex abstract map of features through their numerous layers in modern architectures which enables the learning and generalization of patterns, especially when trained with high quality images. This mechanism is expected to give more importance for recurrent features such as disease spots and ignore non important background, scale and lightning variations. Nonetheless, the results presented in this study reenforce the fact that the CNNs must be presented with enough data variability for complex problems such as disease classification. Another option is the adoption of image pre-processing methods, as background segmentation and image enhancing techniques in order to generate more robust models according to the results presented by [10]. However, this choice would result in undesired extra processes that might even require manual segmentation of images.

More evidence about the importance of natural data variance was shown with the last setting where the CNNs were both trained and tested with images from both lab and field. In this case, the increased variability in training samples fostered the adequate adjusting of the CNNs weights that were consistently more accurate in the classification results. In this setting VGG had an overall classification accuracy of 87.9%, Inception 92% and ResNet 92.9%. These accuracy results are closely related to the ones related by Amara and around 5 to 7% less accurate from the figures reported by [12], [21], [33]. In this perspective, the slighter less accuracy results may be related to the current work concerns around the overfitting in training of the neural nets, where the fine-tuning of layers occurred progressively. Also, another factor is related to a small number of images in field conditions that were acquired through web searches in order to extend the original field dataset in an organic way. These images were not repeatedly acquired in the same environment, as it was for the most cases of classes in the original database.

Images of healthy plants were particularly difficult to classify in field conditions. The classes of apple healthy, blueberry healthy, corn healthy, grape healthy, pepper-bell healthy and soybean healthy were constantly between themselves miss-classified in field conditions. In the same

line, the diseases of potato and tomato early blight were also miss-classified among themselves in non-controlled conditions. This aspect on correct classification of the disease/health although in the wrong crop category, supports the findings of [34] that stated that the learning of CNNs differ according to the methodology implemented and that they do not essentially focus on the part of images affected by the disease, being the disease spots. The authors proposed a new more intuitive method that considers diseases independently of crops was introduced and showed to be more effective than the classic crop-disease pair approach. This finding consequently promotes future research to reconsider the current de facto paradigm of crop disease categorization of images into crop-disease pairs.

IV. Conclusions

The current work proposed the testing of the generalization abilities of three state-of-the-art deep learning CNNs towards the learning and detection of image patterns of plant disease. In this disease classification context, different scenarios were evaluated where the models were trained with images from only lab or lab and field conditions and subsequently were evaluated in classification of the same diseases but with different test sets that were at times from different environmental conditions not seen during training. Special attention was also given in relation to the overfitting of the neural nets. The CNNS were successful in the classification of the crop-disease pairs when provided with images samples for training that covered the same situations presented on the test stage. It can be stated that these algorithms are suitable for the automated detection and diagnosis systems of plant diseases by having as input simple RGB images from leaves, when presented with enough data. This fact is supported by the analysis of results where the CNNs were trained with a controlled set of data and then tested with real field conditions images for the same category. In this case, the accuracy results showed that the neural nets were not effective for generalizing patterns learnt from high quality images in training towards new complex real field situations.

Therefore, the development of a final product would have to contemplate a broader spectrum of images englobed in a careful designed dataset that would be able to contemplate the large variability of crop conditions in different locations with large numbers of individual image samples. This constraint is fundamental for the development of a practical tool for disease detection and classification, despite the significant improvement brought by deep learning CNN algorithms for image analysis tasks.

Acknowledgment

The authors sincerely thank the Brazilian National Council for Scientific and Technological Development (CNPq) for its support through the research grant 205321/2014-3.

Conflict of Interest

Authors declare that they do not have any conflict of interest.

References

[1] IFAD U. Smallholders, food security and the environment. Rome: *International Fund for Agricultural Development*. 2013; 29.
[2] Oerke E-C, Dehne H-W. Safeguarding production—losses in major crops and the role of crop protection. *Crop protection*, 2004;23(4):275–85.
[3] Behmann J, Mahlein A-K, Rumpf T, Römer C, Plümer L. A review of advanced machine learning methods for the detection of biotic stress in precision crop protection. *Precision Agriculture*, 2015;16(3):239–60.
[4] Li L, Zhang Q, Huang D. A review of imaging techniques for plant phenotyping. *Sensors*. 2014;14(11):20078–111.
[5] Izsó E, Bartalné-Berceli M, Gergely S, Salgó A. Off-Line Detection of "Pannon Wheat" Milling Fractions by Near-Infrared Spectroscopic Methods. *International Journal of Agricultural and Biosystems Engineering*. 2015;9(6):655–8.
[6] Dhau I, Adam E, Mutanga O, Ayisi K, Abdel-Rahman EM, Odindi J, *et al.* Testing the capability of spectral resolution of the new multispectral sensors on detecting the severity of grey leaf spot disease in maize crop. *Geocarto International*. 2018;33(11):1223–36.
[7] Franke J, Menz G. Multi-temporal wheat disease detection by multi-spectral remote sensing. *Precision Agriculture*, 2007;8(3):161–72.
[8] Liu Y, Pu H, Sun D-W. Hyperspectral imaging technique for evaluating food quality and safety during various processes: A review of recent applications. *Trends in Food Science & Technology*, 2017 Nov 1;69:25–35.
[9] Camargo A, Smith JS. Image pattern classification for the identification of disease causing agents in plants. *Computers and electronics in agriculture*, 2009;66(2):121–5.
[10] Barbedo JGA. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems engineering,* 2016;144:52–60.
[11] Kamilaris A, Prenafeta-Boldú FX. Deep learning in agriculture: A survey. C*omputers and Electronics in Agriculture,* 2018 Apr 1;147:70–90.
[12] Mohanty SP, Hughes DP, Salathé M. *Using Deep Learning for Image-Based Plant Disease Detection.* Front Plant Sci [Internet]. 2016 [cited 2021 Jul 10];7. Available from: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full.
[13] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015 May;521(7553):436–44.
[14] Raina R, Madhavan A, Ng AY. Large-scale deep unsupervised learning using graphics processors. In: *Proceedings of the 26th annual international conference on machine learning*. 2009: 873–80.
[15] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989;1(4):541–51.
[16] Rawat W, Wang Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 2017;29(9):2352–449.
[17] Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors,* 2017;17(9):2022.
[18] Karmokar BC, Ullah MS, Siddiquee MK, Alam KMR. Tea leaf diseases recognition using neural network ensemble. *International Journal of Computer Applications*, 2015;114(17).
[19] Krishnaswamy Rangarajan A, Purushothaman R, Pérez-Ruiz M. Disease classification in aubergine with local symptomatic region using deep learning models. *Biosystems Engineering*, 2021 Sep 1;209:139–53.
[20] Schirrmann M, Landwehr N, Giebel A, Garz A, Dammer K-H. Early Detection of Stripe Rust in Winter Wheat Using Deep Residual Neural Networks. *Frontiers in Plant Science*. 2021;12:475.
[21] Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, 2016;2016.
[22] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems,* 2012;25:1097–105.
[23] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2016. p. 2818–26.
[24] Hughes D, Salathé M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:151108060,* 2015.
[25] DeChant C, Wiesner-Hanks T, Chen S, Stewart EL, Yosinski J, Gore MA, *et al.* Automated identification of northern leaf blight-infected

maize plants from field imagery using deep learning. *Phytopathology*, 2017;107(11):1426–32.

[26] Rahman CR, Arko PS, Ali ME, Khan MAI, Apon SH, Nowrin F, *et al.* Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*. 2020;194:112–20.

[27] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014;

[28] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015;115(3):211–52.

[29] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. p. 1–9.

[30] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2016. p. 770–8.

[31] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, *et al.* Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:191201703,* 2019.

[32] Shin H-C, Roth HR, Gao M, Lu L, Xu Z, Nogues I, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging,* 2016;35(5):1285–98.

[33] Brahimi M, Boukhalfa K, Moussaoui A. Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence,* 2017;31(4):299–315.

[34] Lee SH, Goëau H, Bonnet P, Joly A. New perspectives on plant disease characterization based on deep learning. *Computers and Electronics in Agriculture,* 2020;170:105220.