Plant Disease Detection using CNN

Thesis · September 2020

DOI: 10.13140/RG.2.2.36485.99048

1 author:

Emma Harte
National College of Ireland

Plant Disease Detection using CNN

Emma Harte
School of Computing
National College of Ireland
Mayor Street, IFSC, Dublin 1
Dublin, Ireland
x18185428@student.ncirl.ie

***Abstract*— Plant disease is an ongoing challenge for smallholder farmers, which threatens income and food security. The recent revolution in smartphone penetration and computer vision models has created an opportunity for image classification in agriculture. Convolutional Neural Networks (CNNs) are considered state-of-the-art in image recognition and offer the ability to provide a prompt and definite diagnosis. In this paper, the performance of a pre-trained ResNet34 model in detecting crop disease is investigated. The developed model is deployed as a web application and is capable of recognizing 7 plant diseases out of healthy leaf tissue. A dataset containing 8,685 leaf images; captured in a controlled environment, is established for training and validating the model. Validation results show that the proposed method can achieve an accuracy of 97.2% and an F1 score of greater than 96.5%. This demonstrates the technical feasibility of CNNs in classifying plant diseases and presents a path towards AI solutions for small holder farmers.**

Keywords—deep learning, transfer learning, classification, CNN, plant disease detection, ResNet architecture, precision agriculture

I. Introduction

By 2050, global crop production must increase by at least 50% to support the predicted demand [1]. The majority of production currently occurs in Africa and Asia, where 83% of farmers are family run with little to no horticultural expertise [2, 3]. Due to this, yield losses of greater than 50%; as a result of pests and diseases are common [4].

In classifying crop diseases, the traditional method of human analysis by visual inspection is no longer feasible. The development of computer vision models offers a quick, standardised and accurate solution to this issue. Once trained, a classifier can also be deployed as an application [5]. Easy to use, all that is required is an internet connection and camera-equipped smartphone. Popular commercial apps 'iNaturalist' [6] and 'PlantSnap' [7] demonstrate how this can be executed. Both apps have attained success in not only delivering expertise to users but also in building an interactive online social community.

Each year, smartphones continue to become more accessible and affordable. In 2020 there are approximately 5 billon smartphone users in the world [8]. Of this, one billion users are located in India and a further one billion are located in Africa. According to Statista, these figures have consistently risen every year for the last decade [9]. With these facts in mind, it is believed that AI apps will play an important role in shaping the future of farming.

The use of CNNs in plant disease classification has achieved excellent results in recent years [10]. Due to the ongoing emergence of superior results, the multi-layered supervised network has become favourable among researchers [11]. Since the release of LeNet (1988), CNN structures have changed dramatically. Sophisticated functions such as ReLu nonlinearity and overlapping pooling [12], have become a prevalent feature in modern architecture. Such developments have helped to reduce training time and error rate [12]. Above all, the evolution of architecture has been a necessary demand of large and complex 21st century datasets [13].

One recent architecture; ResNet (2015) introduced further ground-breaking functions [14]. This incorporates dynamic skip connections as well as heavy batch normalization. This allows training to occur at a much higher learning rate [15]. In 2019, Wu et al., compared ResNet to VGGNet, GoogLeNet, and DenseNet, finding that ResNet produced the best results in classifying grape leaf diseases [16].

In modern research; architectures including AlexNet, LeNet and GoogleNet (2014), are commonly incorporated into the backbone of custom builds [16, 17]. Wallelign proposed such a build; based on LeNet, in his research of Soybean disease classification. The model consisted of three convolution layers, one max-pooling layer and a fully connected MLP with Relu activation and achieved a 99% accuracy rate [18].

Data pre-processing is crucially important to a model's performance. Viral, bacterial and fungal infections can be difficult to distinguish, often sharing an overlap of symptoms. These symptoms can be any measurable difference in colour, shape or function which result as the plant responds to the pathogen [19]. Because of this complexity, it is preferable to use RGB data [10, 20]. This produces clear, noise free images which may take longer than greyscale data to train, but overall are more suitable for plant disease identification models [21].

Smaller datasets or unvaried data can affect a model's reliability. This can be managed in several ways, by using techniques such as augmentation or transfer learning. Augmenting training images can not only reduce overfitting but can improve a model's overall performance [22, 18].

This can be performed by adding functions such as zoom, rotate, adding colour changes or contrast changes. The transformed images should, however, reflect the expectations of the validation dataset [18]. When inappropriately applied, a classifier's accuracy can worsen despite the extra data generated.

The method of transfer learning has also proved very successful when working with smaller datasets. This involves fine-tuning the weights of a pre-trained model. The ImageNet database is commonly used for this purpose and contains over 14 million images [23]. In 2016, Mohanty et al. exposed these benefits in a study focused on crop disease classification. Here, superior results were recorded using transfer learning (ImageNet), compared to a model built from scratch [24]. As

ImageNet contains images irrelevant to a plant specific task, it is questionable whether pre-training on a botanical database instead, may enhance performance. Current research suggests that pre-training on ImageNet may generalise better, however pre-training on a plant specific task may reduce overfitting. These statements, however, are inconclusive. Due to an absence of large botanical datasets, the topic is relatively unexplored [25]. Augmentation can also be applied to pre-trained models. Due to the knowledge already attained by such a model however, the effects are greater when applied to un-trained CNNs [26].

The quality and type of training data massively impacts the model's capabilities. When trained on imagery which contains plain background data, a classifier's accuracy becomes dependent on this composition [20]. Therefore, it is likely to be unreliable when tested with in-field photography. Many of the available plant disease datasets including, the 'PlantVilllage' dataset [10], do not contain in-field imagery. The need for such a dataset is highlighted heavily in research [24, 20].

Segmentation in this case can prove effective, by separating a leaf from its background [27]. This technique can furthermore be used in situations where the classifier requires scene awareness. For example, this may involve understanding the extent of pathogen damage around the infected tissue, as opposed to just the infected tissue [28, 29]. Segmentation is not a new concept and has been applied to disease classification tasks since the 1990s. Even at this early stage, good results were reported. Early studies were also helpful in identifying the limitations, showing that the technique could not overcome poor image quality. Thus, stressing the importance of careful data collection and pre-processing [30]. The relevance of segmentation continues into 2020. There is a great deal of research potential in combining this with specialized imagery [31].

The type of training data used also determines what stage of disease, detection is possible. For early disease detection, specific imagery must be used [32]. Chlorophyll fluorescent (CFI), infrared thermography (IRT), hyperspectral (HSI) and multispectral (MSI) imagery have specific abilities to identify symptoms which are not yet visible to the naked eye. These can be used alone or combined where appropriate [32]. For example, IRT has the exceptional ability to detect an increase in temperature. This has been successful in diagnosing crop diseases including downy mildew in roses [33] and FHB in wheat [32], days before symptoms were visible.

This topic of early detection is relatively unexplored due to the limited availability of such data. [22, 33] The technology needed to capture this specialised imagery is becoming more affordable, with a growing academic interest in the area. At this stage however, it is not an accessible tool for remote farmers. Therefore, it would be unrealistic to include it in a project intended for such users [34].

II. Contributions

This study aims to evaluate the use of a pre-trained ResNet34 model in training a plant disease classifier. Three plant species will be focused on. These include potato (*Solanum tuberosum)*, tomato (*Solanum lycopersicum*) and rice (*Oryza sativa*). For each species the model will be trained to recognize a select number of diseases or state of healthiness.

The specific goals of this research are to:

i) Determine the model's overall effectiveness in classifying diseases using both a validation and test dataset.

ii) Compare the model's accuracy when tested with various image sizes and augmentation settings.

iii) Deploy the trained model to create an easy to use web application

Due to an uneven class distribution, both the f1-score and accuracy metrics will be examined in accessing the model's performance. Once the model reaches an accuracy and F1score of greater than 80%, it will be accepted.

This research will be carried out with the needs of smallholder farmers in mind. The classifier and web application will require both a smartphone and internet connection, which as previously mentioned, continue to reach remote regions. In acknowledging the limitations of basic camera phones, the model will be tested with a variety of image sizes and augmentation settings.

III. Materials and Methods

This section describes the steps involved in creating and deploying the classifier. Classification by CNN is divided into three phases which tackle separate tasks. All work involved in this research was completed on one machine, with specifications listed in Table 1.

A. Data Acquisition

All Potato and Tomato imagery derive from 'The PlantVilllage Dataset' [35], an open-access repository which contains in total 54,323 images. All Rice imagery originates from the "Rice Diseases Image Dataset" Kaggle dataset [36]. For each species, a select number of classes are chosen, with details viewable in Table II.

All images are captured in a controlled environment. Due to this, model bias is expected. To access this, a test dataset containing 50 images, sourced from Google is also established. These images contain additional plant anatomy, in-field background data and varying stages of disease.

TABLE I. MACHINE SPECIFICATIONS

Hardware & Software	Characteristics
Memory	8.0GB
Processor	Intel(R) Core™ i5-9300H CPU @ 2.40GHz
Graphics	NVIDIA GeForce RTX 2060 6GB GDDR6
Operating system	Windows 10 Home 64

B. Data Pre-Processing

The dataset is divided into 80% for training and 20% for validation. First, augmentation settings are applied to the training data. These are generated 'on the fly', with each operation carrying a weighted probability of appearing in each epoch [37].

The settings applied include flipping (random), padding mode (reflection) and zoom with crop (scale = (1.0,1.5)). 'Zoom with crop' was later omitted after discovering that it had inappropriately cropped areas of infected leaf. Finally, all images are re-sized and normalized. Resizing is carried out using a compress function, to 150 x 150. As a pre-trained model is used, the RBG ImageNet statistics are used to normalize. A sample of the final pre-processed images is viewable in Fig.1.

C. Classification by CNN

1) Phase One – Trialling of Image size

Phase one aims to investigate the effect that image size has on model performance. In total, five images sized are tested ranging from 150 x 150 to 255 x 255.

To begin, the Resnet34 pre-trained weights are downloaded. As a default of transfer learning, all layers with the except of the final two layers are frozen. These contain new weights and are specific to the plant disease classification task. Freezing allows these layers to be disease separately trained, without backpropagating the gradients. In exactly this way, the 1cycle policy is used to train the final layers.

With this complete, the remaining layers are released. To aid the fine-tuning process, a plot displaying learning rate vs loss is generated and analysed. From this, a suitable learning is selected, and the model is run. With results recorded, the model is re-created to the additional four image sizes (Table III.). All steps remain consistent in each trial including the learning rate

TABLE III. IMAGE SIZE TRIAL INFORMATION

Trial	Image Size	No. Epochs	Learner Rate
1	150 x 150	4	1e-05 and 1e-04
2	195 x 195	4	1e-05 and 1e-04
3	224 x 224	4	1e-05 and 1e-04
4	244 x 244	4	1e-05 and 1e-04
5	255 x 255	4	1e-05 and 1e-04

TABLE II. DATASET USED FOR CLASSIFICATION

Species	Class	No. of Images
Potato	Early blight	1000
Potato	Late blight	1000
Potato	Healthy	152
Tomato	Bacterial Spot	2119
Tomato	Leaf Mold	952
Tomato	Mosaic Virus	160
Tomato	Healthy	1000
Rice	Brown Spot	523
Rice	Leaf Blast	779
Rice	Healthy	1000

2) Phase Two – Model Optimisation

Using the most suitable image size, the ResNet34 model is optimised. To further improve the model's performance, additional augmentation settings are added (Fig. 2). Operations include brightness changes (0.4,0.7) and warp (0.5).

Next, the final two layers are isolated and trained at the default learning rate. With this complete, fine tuning is performed, running multiple trials to test a series of learning rates and number of epochs.

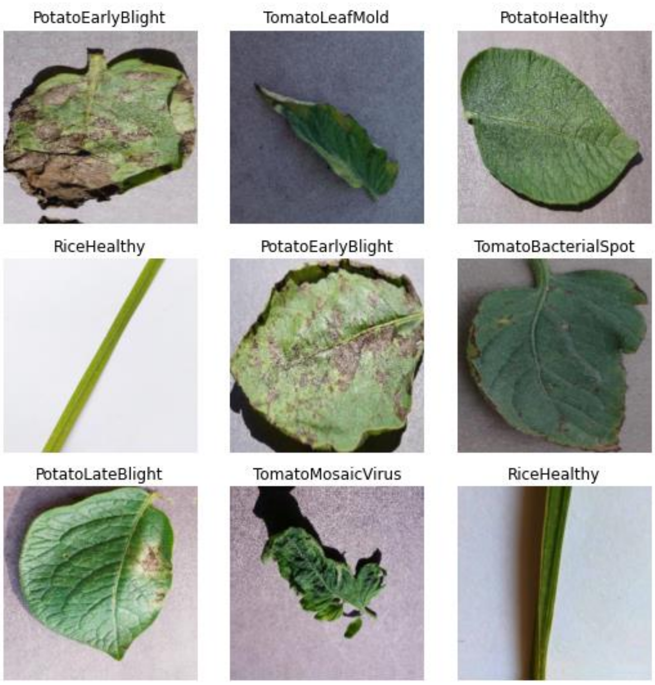

Fig. 1. Pre-processed images - Phase One *augmentation settings = flipping (random), padding mode (reflection)*

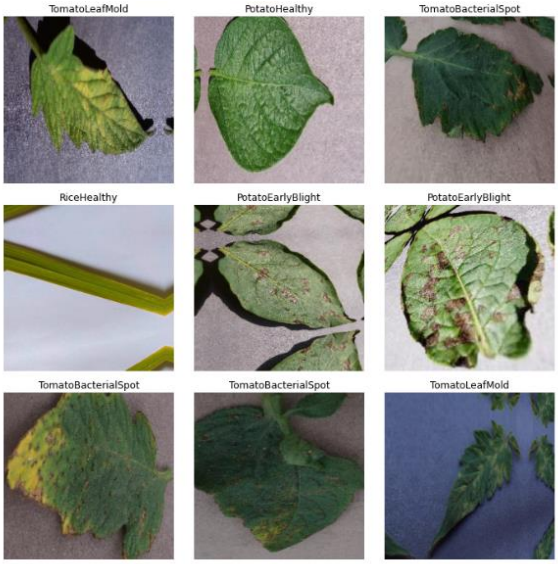

Fig. 2. Pre-processed images - Phase Two
augmentation settings = brightness changes (0.4,0.7), warp (0.5), flipping (random), padding mode (reflection)

3) *Phase Three – Visualisations*

For the purpose of interpretation, a series of visualisations are generated based on the validation and test datasets. Additionally, the model is deployed to create a web application. To achieve this, the completed essential files are stored in a GitHub repository and the model is exported as a pickle file. To deploy the model, the repository is connected to the unified platform; Render. In carrying out this task, the 'Render Examples' GitHub repository was used as a guide. [41]

IV. Project Management

All work associated with this research was carried out during a 12-week period. This project contained several challenging elements which required careful management. One of which, being that both python and image classification were unfamiliar to the researcher.

As a starting point, both a Gantt chart and RAID log were created online [38, 40]. Initially these documents were used to define the scope of the project. At this early stage, all tasks dependencies, required resources, risks and issues were also identified and discussed. Both documents were updated and reviewed regularly throughout the project.

Due to unforeseen circumstances, a two-week extension was granted. This extra time was used to execute all tasks, including the lower priority tasks which had previously been identified. The final edit of both the Gantt chart and RAID log can be viewed in the supplementary material file, which has been submitted separately to this project.

All programming tasks were carried out on the free cloud service GoogleColab which offers 25GB GPU. The only cost incurred throughout this study was model deployment on Render. The model will be deployed for one month (30/04/2020-30/05/2020), costing approximately $10. For programming support, the online Fastai [42], Render [41] and Pytorch [43] documentation was reviewed.

V. Results

1) *Phase One – Trialling of Image Size*

The results of Phase One prove that it is possible to achieve an accuracy and F1 score of greater than 90% for image sizes 155 x 155 to 255 x 255. As expected, an increase in image size not only improves feature extraction but also increases running time (Table IV.). This initial analysis produced excellent results. As previously stated, the model would be accepted if it reached an accuracy of at least 80%. Even at this early stage, results far exceed the acceptance criteria.

To achieve this result, each model was passed a range of learning rates from 1e-05 to 1e-04 and run for 4 epochs.
Overall, image size 244 produced the best results including the highest accuracy and F1score. Although literature suggests image size 224 x 224 to be suitable for plant disease classification tasks (10a), this model appears to marginally benefit from an increased image size. For these reasons, image size 244 was chosen for the remainder of this research.

2) *Phase Two – Model Optimisation*

Prior to fine-tuning, the model attained an accuracy of 0.9465 and F1 score of 0.9359.(Fig. 3) To aid fine-tuning, a plot depicting learning rate (logarithmic scale) v loss was analyzed (Fig. 4). This demonstrates a relatively low loss between learning rates 1e-06 to 1e-04. As the learning rate increases past 1e-04 however, a dramatic increase in loss is experienced. These facts considered, several trails testing learning rate were carried out.

A learning rate range of 1e-05 to 1e-04 produced the best results. By fine-tuning this hyperparameter, a slight increase in accuracy (1.5%) and F1-Score (1.3%) was accomplished. On the final epoch however, the closing training and validation values indicate that the model may be slightly underfitting (Fig. 5). To correct this, the number of epochs was increased systematically. At Approximately the 10^{th} epoch, there was an evident improvement to the fit of the mode. A final reading presented an overall improvement of 2.8% in accuracy and 3.1 % in F1-score (Fig. 6).

As stated earlier, the validation dataset consists of a very specific composition; one leaf and a plain background. For an accurate reading, akin to those stated in this section, use of the classifier should mimic this image layout.

Table IV. Results - Phase One
(4 epochs, max_lr = slice(1e-05,1e-04))

Test	Image size	Train Loss	Valid Loss	Accuracy	F1 Score	Time (hours)
1	155	0.1660	0.1222	0.9557	0.9439	2:83
2	195	0.1588	0.1150	0.9585	0.9460	3.62
3	224	0.1778	0.1256	0.9522	0.9359	4.29
4	244	0.1310	0.1153	0.9603	0.9450	5.20
5	255	0.1607	0.1249	0.9562	0.944	5.42

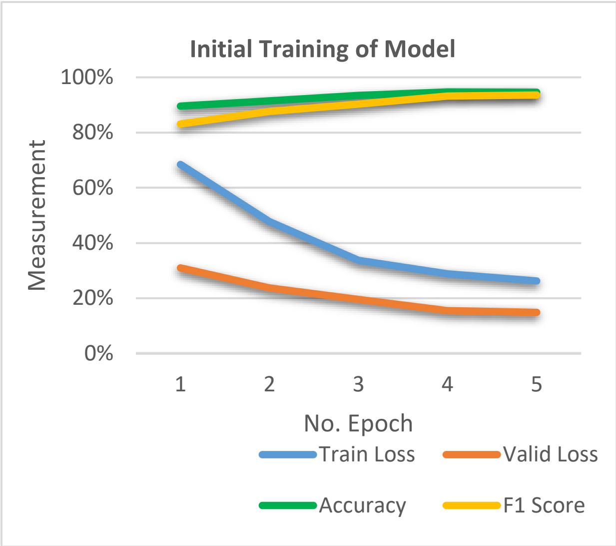

Fig. 3. Training the final layers (lr=1e-3)
Before fine-tuning the model attained an accuracy of 0.9465 and F1 score of 0.9359.

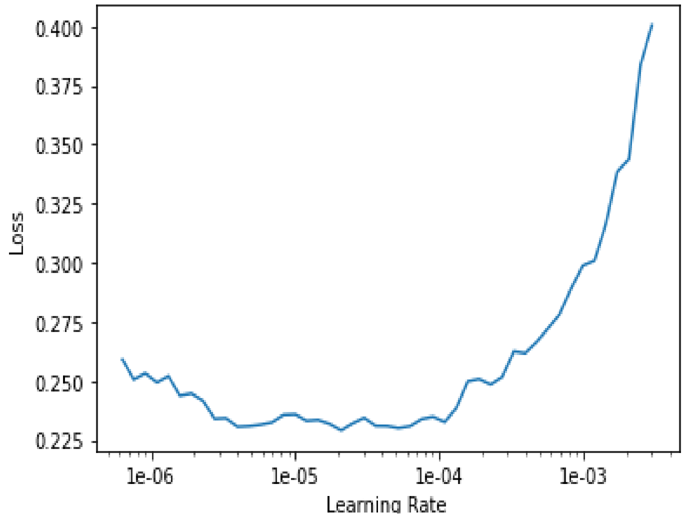

Fig. 6. Learning rate v loss
Used to guide the fine-tuning process.
As the learning rate increases past 1e-04, a dramatic increase in loss is experienced.

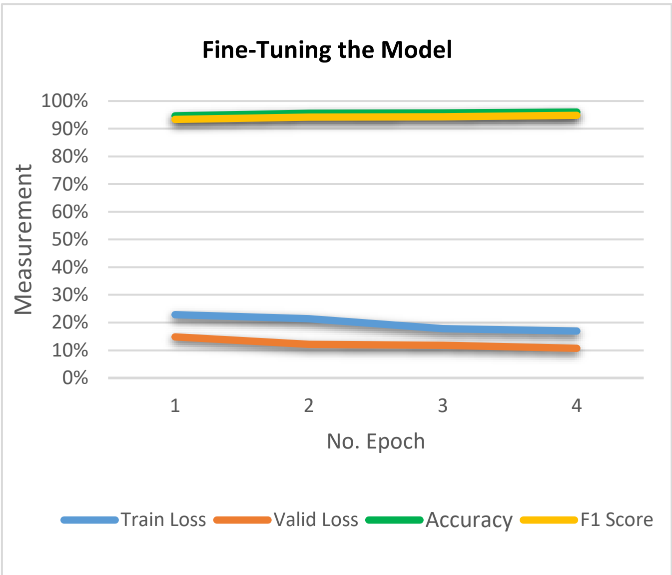

Fig. 4. Fine-tuning the model,
learning rate range = 1e-05, 1e,04, epochs = 4,
Signs of underfitting apparent

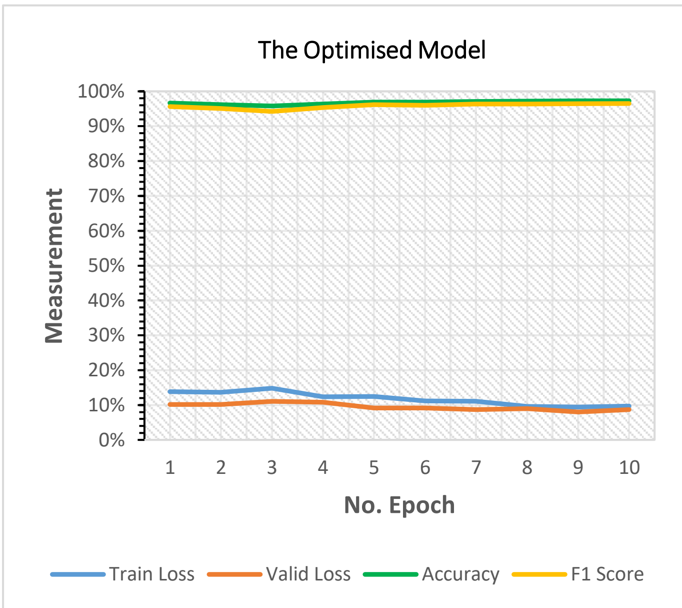

Fig. 5. The fianl optimized model,
learning rate range = 1e-05, 1e,04, epochs = 10

3) Phase Three– Visualisasations

An analysis of heat maps reveals the inner workings of the CNN. Colour, shape and texture appear to be important factors in working to extract plant disease features (Fig.7, Fig.8). Colour appears to be especially crucial, helping to clearly differentiate similar diseases, by adding an extra dimension of characterisation. This explains the importance of RGB data to disease classification tasks, as was highlighted earlier [10, 20]. For all three species, the CNN shows effectiveness in recognising features. This is also true for rice disease classes, which contain smaller, and more difficult to distinguish symptoms.

The confusion matrix presented in Fig. 10 lists the validation dataset results. Overall, no errors were recorded in any Potato or Tomato classes. Rice as a species, performed poorly, suggesting that there may be an underlying issue with the data. Rice Brown Spot was the highest misclassified class. 13.9% of these images were incorrectly classified as Healthy and a further 9.9% were misclassified as RiceLeafBlast. A clear symptom of brown spot is irregular dark spots. While this may be mistaken for similar lesions in leaf blast, there should be overlapping characteristics with healthy samples. On average, 12.65% of each Rice class were misdiagnosed.

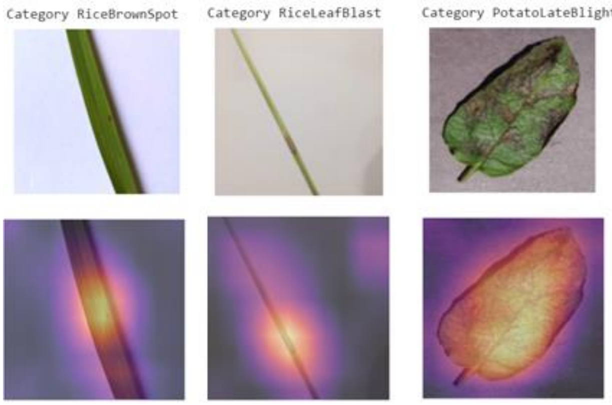

Fig. 7. Heat map example 1

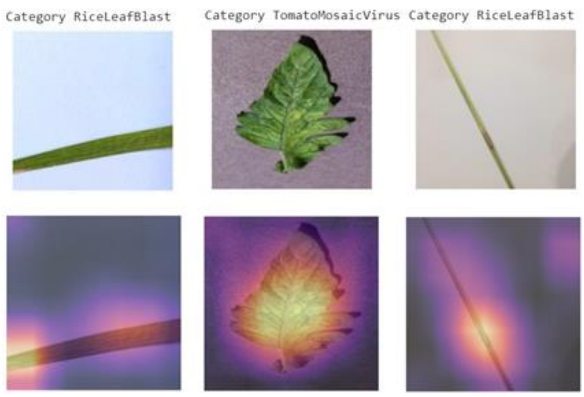

Fig. 8. Heat map example 2

To investigate this matter further, the misclassified images were plotted and sorted respective to loss (Fig. 11). A closer inspection reveals that the quality of several images is questionable. Even to the skilled eye, an accurate diagnosis based on these images would be challenging. This data may have been mislabeled or is simply a poor class representation. As such data is not beneficial to the classifier, it should not be included in the training dataset.

As expected, the model suffers a significant drop in accuracy when in-field imagery is tested. Out of 50 images, only 44% were accurately diagnosed (Fig. 12). This is due to a combination of factors; which augmentation could not overcome; including new plant anatomy and alternative background data. As the model was not trained on such data, adapting to such circumstances is extremely difficult. Diversifying the training data to include imagery which has been captured in this uncontrolled environment could stand to strengthen the model immensely. As highlighted earlier, there is a current lack of 'in field' plant disease imagery available. These results signify the importance of developing such resources.

Finally, the model was deployed on Render to create a web application (Fig. 9). This provides the user with a live disease classification service and reflects the capabilities and limitations of both the validation and test dataset, which have been discussed in this section. The application is available to use on the following link:

https://plants.onrender.com.

Fig. 9. Creation of the web application on Render

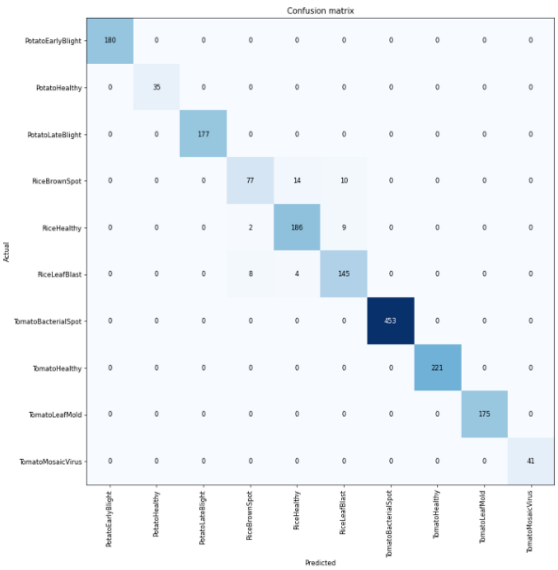

Fig. 10. Confusion matrix – validation dataset

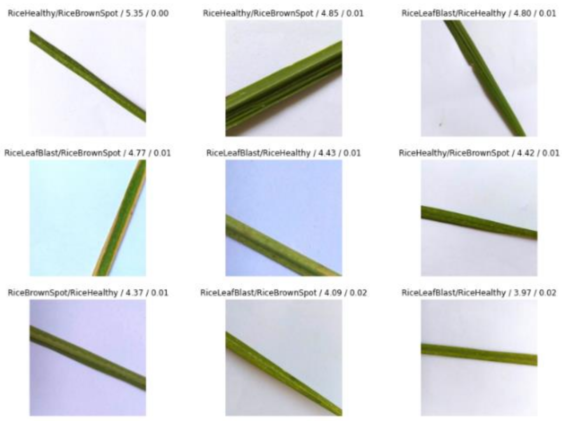

Fig. 11. Top losses plotted – validation dataset

Actual \ Predicted	PotatoEarlyBlight	PotatoHealthy	PotatoLateBlight	RiceBrownSpot	RoceHealthy	RiceLeafBlast	TomatoBacterialSpot	TomatoHealthy	TomatoLeafMold	TomatoMosaicVirus
PotatoEarlyBlight	4		1							
PotatoHealthy	1	1	1					2		
PotatoLateBlight	1		4							
RiceBrownSpot	1					4	1			
RiceHealthy					3	1		1		
RiceLeafBlast						4	1			
TomatoBacterialSpot			1				3		1	
TomatoHealthy							3	1	1	
TomatoLeafMold							2	1	2	
TomatoMosaicVirus							2		3	

Fig. 12. Confusion matrix – test dataset

VI. Conclusion

To prevent losses, small holder farmers are dependent on a timely and accurate crop disease diagnosis. In this study, a pre-trained Convolutional Neural Network was fine-tuned, and the model was deployed online. The final result was a plant disease detection app. This service is free, easy to use and requires just a smart phone and internet connection. Thus, the user's needs as defined in this paper have been fulfilled.

A thorough investigation exposes the capabilities and limitations of the model. Overall, when validated in a controlled environment, an accuracy of 97.2% is presented. This achieved accuracy depends on a number of factors including the stage of disease, disease type, background data and object composition. Due to this, a set of user guidelines would be required for commercial use, to ensure the stated accuracy is delivered. As the model was trained using a plain background and singular leaf, imitation of these features is best.

Augmentation and transfer learning in this case, proved beneficial to the model, helping the CNN to generalize more reliability. While this improved the model's ability to extract features r, it was not enough when the model was presented with 'in field' imagery. In this case, the classifier ranked an accuracy of just 44% Above all, this highlights the importance of diversifying the training dataset to include alternative background data, additional plant anatomy and varying stages of disease.

Overall, this study is conclusive in demonstrating how CNNs may be applied to empower small-holder farmers in their fight against plant disease. In the future, work should be focused on diversifying training datasets and also in testing similar web applications in real life situations. Without such developments, the struggle against plant disease will continue.

Acknowledgment

Thank you to NCI lecturer and project mentor Symeon Charalabides, and to the NCI Data Analytics Support Group; who provided unconditional supported throughout this project.

VII. Supplementary Material

- 1 - Gantt chart and RAID Log
- 2 – Charts and tables
- 3 - Python code
- 4 – Web application code
- 5 – Additional dataset information

References

[1] M. Hunter, R. Smith, M. Schipanski, L. Atwood and D. Mortensen, "Agriculture in 2050: Recalibrating Targets for Sustainable Intensification", *BioScience*, vol. 67, no. 4, pp. 386-391, 2017. Available: 10.1093/biosci/bix010.

[2] "PlantVillage", Plantvillage.psu.edu, 2020. [Online]. Available: https://plantvillage.psu.edu/. [Accessed: 31- Jan- 2020].

[3] D. Klauser, "Challenges in monitoring and managing plant diseases in developing countries", Journal of Plant Diseases and Protection, vol. 125, no. 3, pp. 235-237, 2018. Available: 10.1007/s41348-018-0145-9.

[4] A.Muimba-Kankolongo, Food crop production by smallholder farmers in Southern Africa. Elsevier, 2018, pp. 23-27.

[5] J. Boulent, S. Foucher, J. Théau and P. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases", *Frontiers in Plant Science*, vol. 10, 2019. Available: 10.3389/fpls.2019.00941.

[6] "iNaturalist", iNaturalist, 2020. [Online]. Available: https://www.inaturalist.org/. [Accessed: 30- Jan- 2020].

[7] I. Ltd., "Success-stories - PlantSnap: Training the world's largest plant recognition classifier. | Imagga Technologies Ltd.", Imagga.com, 2020. [Online]. Available: https://imagga.com/success-stories/plantsnap-case-study. [Accessed: 31- Jan- 2020].

[8] "Smartphone users worldwide 2020 | Statista", Statista, 2020. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. [Accessed: 22- Apr- 2020].

[9] K. Liakos, P. Busato, D. Moshou, S. Pearson and D. Bochtis, "Machine Learning in Agriculture: A Review", *Sensors*, vol. 18, no. 8, p. 2674, 2018. Available: 10.3390/s18082674.

[10] Y.Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease", *Plant Phenomics*, vol. 2019, pp. 1-14, 2019. Available: 10.34133/2019/9237136.

[11] Atabay, H. A. 2016b. A convolutional neural network with a new architecture applied on leaf classification. IIOAB J 7(5):226–331

[12] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017. Available: 10.1145/3065386.

[13] S. Jadhav, "Convolutional Neural Networks for Leaf Image-Based Plant Disease Classification", *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 4, p. 328, 2019. Available: 10.11591/ijai.v8.i4.pp328-341.

[14] C. Szegedy and S. Ioffe, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Arxiv.org*, 2015. [Online]. Available: https://arxiv.org/pdf/1502.03167v3.pdf. [Accessed: 10- Apr- 2020].

[15] D. Wu, Y. Wang, S. Xia, J. Bailey and X. Ma, "Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets", *arXiv.org*, 2020. [Online]. Available: https://arxiv.org/abs/2002.05990. [Accessed: 10- Apr- 2020].

[16] M. Ji, L. Zhang and Q. Wu, "Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks", *Information Processing in Agriculture*, 2019. Available: 10.1016/j.inpa.2019.10.003.

[17] X. Zhang, Y. Qiao, F. Meng, C. Fan and M. Zhang, "Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks", *IEEE Access*, vol. 6, pp. 30370-30377, 2018. Available: 10.1109/access.2018.2844405.

[18] C. Buche, S. Wallelign and M. Polceanu, "Soybean Plant Disease Identification Using Convolutional Neural Network", 2018. [Accessed 22 April 2020].

[19] Isleib, "Signs and symptoms of plant disease: Is it fungal, viral or bacterial?", MSU Extension, 2012. [Online]. Available: https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial. [Accessed: 11- Apr- 2020].

[20] P. Sharma, Y. Berwal and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation", Information Processing in Agriculture, 2019. Available: 10.1016/j.inpa.2019.11.001.

[21] Padmavathi and K. Thangadurai, "Implementation of RGB and Grayscale Images in Plant Leaves Disease Detection – Comparative Study", *Indian Journal of Science and Technology*, vol. 9, no. 6, 2016. Available: 10.17485/ijst/2016/v9i6/77739.

[22] S. Sankaran, A. Mishra, R. Ehsani and C. Davis, "A review of advanced techniques for detecting plant diseases", Computers and Electronics in Agriculture, vol. 72, no. 1, pp. 1-13, 2010. Available: 10.1016/j.compag.2010.02.007.

[23] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei L. (2009). “Imagenet: A large-scale hierarchical image database,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (IEEE).

[24] S. Mohanty, D. Hughes and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection", Frontiers in Plant Science, vol. 7, 2016. Available: 10.3389/fpls.2016.01419.

[25] S. Lee, H. Goëau, P. Bonnet and A. Joly, "New perspectives on plant disease characterization based on deep learning", Computers and Electronics in Agriculture, vol. 170, p. 105220, 2020. Available: 10.1016/j.compag.2020.105220.

[26] P. Pawara, E. Okafor, L. Schomaker and M. Wiering, "Data Augmentation for Plant Classification", 2017. Available: https://www.researchgate.net/publication/319990090_Data_Augmentation_for_Plant_Classification. [Accessed 11 April 2020].

[27] B. Škaloudová, V. Křivan and R. Zemek, "Computer-assisted estimation of leaf damage caused by spider mites", *Computers and Electronics in Agriculture*, vol. 53, no. 2, pp. 81-91, 2006. Available: 10.1016/j.compag.2006.04.002.

[28] Y. Oo and N. Htun, "Plant Leaf Disease Detection and Classification using Image Processing", *International Journal of Research and Engineering*, vol. 5, no. 9, pp. 516-523, 2018. Available: 10.21276/ijre.2018.5.9.4

[29] K. Lin, L. Gong, Y. Huang, C. Liu and J. Pan, "Deep Learning-Based Segmentation and Quantification of Cucumber Powdery Mildew Using Convolutional Neural Network", *Frontiers in Plant Science*, vol. 10, 2019. Available: 10.3389/fpls.2019.00155.

[30] C. Tucker and S. Chakraborty, "Quantitative Assessment of Lesion Characteristics and Disease Severity Using Digital Image Processing", *Journal of Phytopathology*, vol. 145, no. 7, pp. 273-278, 1997. Available: 10.1111/j.1439-0434.1997.tb00400.x.

[31] M. Hasan, S. Ullah, M. Khan and K. Khurshid, "COMPARATIVE ANALYSIS OF SVM, ANN AND CNN FOR CLASSIFYING VEGETATION SPECIES USING HYPERSPECTRAL THERMAL INFRARED DATA", ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. -213, pp. 1861-1868, 2019. Available: 10.5194/isprs-archives-xlii-2-w13-1861-2019.

[32] A. Mahlein, E. Alisaac, A. Al Masri, J. Behmann, H. Dehne and E. Oerke, "Comparison and Combination of Thermal, Fluorescence, and Hyperspectral Imaging for Monitoring Fusarium Head Blight of Wheat on Spikelet Scale", *Sensors*, vol. 19, no. 10, p. 2281, 2019. Available: 10.3390/s19102281

[33] S. Gomez, "Infection and spread of Peronospora sparsa on Rosa sp. (Berk.) - a microscopic and a thermographic approach", 2014. [Accessed 22 April 2020].

[34] J. Su et al., "Wheat yellow rust monitoring by learning from multispectral UAV aerial imagery", Computers and Electronics in Agriculture, vol. 155, pp. 157-166, 2018. Available: 10.1016/j.compag.2018.10.017.

[35] S. Mohanty, "spMohanty/PlantVillage-Dataset", *GitHub*, 2016. [Online]. Available: https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw/color. [Accessed: 26- Apr- 2020].

[36] Huy Minh Do]"Rice Diseases Image Dataset", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/minhhuy2810/rice-diseases-image-dataset. [Accessed: 26- Apr- 2020].

[37] "vision.transform | fastai", *Docs.fast.ai*, 2020. [Online]. Available: https://docs.fast.ai/vision.transform.html. [Accessed: 26- Apr- 2020].

[38] "Online Gantt Chart Software | TeamGantt", *Teamgantt.com*, 2020. [Online]. Available: https://www.teamgantt.com/. [Accessed: 27- Apr- 2020].

[39] "Practical Deep Learning for Coders, v3 | fast.ai course v3", *Course.fast.ai*, 2020. [Online]. Available: https://course.fast.ai/. [Accessed: 27- Apr- 2020].

[40] "GroupMap | Collaborative Brainstorming & Group Decision-Making", *GroupMap - Collaborative Brainstorming and Decision*

[41] "Render Examples", GitHub, 2020. [Online]. Available: https://github.com/render-examples. [Accessed: 29- Apr- 2020].

[42] "| fastai", Docs.fast.ai, 2020. [Online]. Available: https://docs.fast.ai/. [Accessed: 29- Apr- 2020].

[43] "PyTorch documentation — PyTorch master documentation", Pytorch.org, 2020. [Online]. Available: https://pytorch.org/docs/stable/index.html. [Accessed: 29- Apr- 2020]