A Post-processing Fusion Framework for Deep Learning Models for Crop Disease Detection

Subodh Bansal¹, Anuj Kumar²

¹Department of Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Sector 25, Chandigarh
²Department of Computer Science and Applications, Panjab University, Sector 14, Chandigarh
Subodhbansal991@gmail.com

Abstract. The paper proposes a Post-processing Fusion Framework (PFF) for crop disease detection. The framework combines the prediction matrices produced by the multiple deep learning models and increases the overall classification accuracy of the system. The model is tested on the Plant Village dataset covering 14 plants and 26 diseases in 54,305 images, divided into 38 classes. Five deep learning models are used in the classification stage. Further, the PFF is applied to the results of the classification stage. Post-processing enhanced the efficiency of the system and achieved an accuracy of 99.33%. It secures 99.99%, 99.61% and 99.42% scores for top-5 accuracy, Mean Reciprocal Rank and Mean Average Precision values, respectively. Additionally, the classification rate of the approach is less than one second per test image. The high speed and accuracy make the technique highly suitable for real cultivation early warning systems.

Keywords. Post-Processing, Deep Convolutional Neural Networks, Deep Learning, Image Classification, Plant disease detection.

1. Introduction

Crop loss due to diseases is a significant cause for the reduction in crop yield. Every year, over 20% of the agriculture output is lessened due to pathogenic infections caused to crops [1]. It results in a reduction in the availability of food, thus making it a significant challenge in achieving the Goal-2: ‘Zero Hunger’ set by the United Nations Organisation [2]. Therefore, a mechanism for mitigating the same is the need of the hour.

Crop loss due to diseases can be prevented or reduced by its prompt detection and correct identification and early treatment. So, a crop disease detection and classification system (CDDS) is required. The field of computer vision and artificial intelligence can be used for the development of the system.

Many researchers have made efforts to develop effective CDDSs. Several approaches ranging from the traditional pattern matching to the modern-day state-of-the-art deep learning (DL) models have been applied for its development. The steps involved in conventional and modern approaches are listed in figure 1(a) and 1(b), respectively.
In the traditional approaches, the main focus of the researchers was to handpick the best features (symptoms) for detection of the disease. Whereas in the modern approaches, testing every possible orientation of the black box DL models to maximise the classification accuracy is the focus. Still, in majority of the researches, regardless of the approach, work is done on the stages starting from pre-processing to classification. The post-processing stage is often neglected.

In the present investigation, a Post-processing Fusion Framework (PFF) is proposed to increase the accuracy of the DL models after the classification stage. It is implemented on the Plant Village [3] dataset. For validation of the framework, five DL models are used. The paper contains an in-depth literature review of the existing CDDS approaches, in section 2. Section 3 presents the proposed PFF. The experimentation of the investigation is detailed in section 4. The obtained results are discussed in section 5. Finally, the conclusions drawn, and the possible future directions of the research field are discussed in section 6.

2. Related work

The efforts made in the field of computer vision and machine learning for the development of CDDS are summarised in this section. First, the traditional methods are discussed followed by the modern methods.

2.1. Traditional methods

S. Kai et al. [4] proposed an approach for crop disease detection using Grey Level Co-occurrence Matrix (GLCM) [5] for feature extraction and Back Propagation Neural Networks (BPNN) [6]. The researcher applied the technique on a self-collected dataset of maize leaf images and achieved an accuracy of 98.00%. Z. Zhang et al. [7] experimented with variety of combinations of Genetic Algorithms (GA) [8] and Support Vector Machines (SVM) [9]. Experimentations on a self collected dataset concluded in an accuracy of 90.25%.

B. Jiang et al. [10] experimented on water stress for Maize using Super-Pixel (SP) and GLCM for feature extraction and SVM for classification. Application of the methodology on a Self constructed dataset of 1297 leaf mages (in black background), was able to achieve an accuracy of 98.97%. X. E. Pantazi et al. [11] implemented Local Binary Patterns (LBPs) for feature extraction and One-Class Classification (OCC) technique for classification. The researcher identified downy mildew, Powdery mildew and black rot diseases from crop leaf images.
2.2. Modern methods

The parallel computation power of computers have increased a lot in the recent years, especially with the introduction of computationally powerful Graphical Processing Units (GPU) having hundreds of computing cores. This has helped the field of Artificial Intelligence to achieve seen exponential advancement. Execution of the computationally expensive deep learning models (DL) (once thought to be just theoretical concepts) is now possible [12]. Following are some of the advancements in the same area of crop disease detection.

G. Wang et al. [13] used (Video Geometry Group) VGG-16, VGG-19 [14], Inception-v3 [15], and Resnet [16] DL models for detection of Black Rot disease in Apple leaves, in the Plant Village dataset achieving an accuracy of 93.1%. M. Brahim et al. [17] used AlexNet [18] and GoogLeNet [19] models for multiple diseases of tomato leaves achieving accuracy, precision, recall and F1 score of 99.18%, 99.53%, 98.53% and 98.52%, respectively. K. Yamamoto et al. [20] applied structurally constrained Recurrent Neural Networks [21] on the same dataset achieving the accuracy of 95%.

J. Gwan et al. [22] used VGG-A DL model on radish leaf images captured using Unmanned Aerial Vehicle (UAV) for the detection of Fusarium wilt and scored an accuracy of 97.40%. J. Lu et al. [23] used VGG-FCN for disease detection for Wheat leaves on Wheat Disease Database 2017 (WDD2017) achieving accuracy of 97.95%. C. Dechant [24] developed a CNN model for the detection of Northern Corn Leaf Blight in Maize leaves and correctly identified the same, 96.70% of times.

A. Fuentes et al. [25] applied various (AlexNet, ZFNet, VGG-16, GoogLeNet, ResNet-50, ResNet-101, ResNetXt-101 [26], Faster RCNN [27], R-FCN [28] and SSD) DL models on a self extracted dataset of 5000 images covering 9 diseases of tomato plant leaves from Korea, achieving an average precision of 83.06%. A. C. Cruz et al. [29] developed a modified DL X-FIDO Detector for O. europea L. model for olive quick decline syndrome disease of Olive plant, achieving an accuracy value of 98.60%.

A. Ramcharan et al. [30] worked on cassava plant leaf disease detection using transfer learning with Inception v3 and ImageNet DL models on a self extracted dataset of 11,670 images achieving an accuracy of 93%. B. Liu et al. [31] worked with self captured dataset of 1053 images of Apple crop using AlexNet, GoogLeNet, VGG-16 and ResNet-20 DL models achieving classification accuracy of 97.62%. J. Amara et al. [32] worked with Banana crop’s self captured dataset containing 147 images of healthy and infected leaves. Further experimentation with LeNet model resulted in achievement of accuracy, precision, recall and F1-score of 98.61%, 98.67%, 98.60% and 98.64% respectively.

S. Ghoosal et al. [33] worked on Soybean using self-captured dataset of 25,000 images with AlexNet DL model, achieving classification accuracy of 94.31%. M. Sibiya et al. [34] conducted experiments with CNN using an aggregation approach for Maize leaves for detection of fungal diseases from Plant Village Dataset [3] and achieved the success rate of 92.85%. J. C. A. Barbedo [35] used DL approaches on individual lesions and spots of diseased leaf instead of whole leaf which increased the accuracy by 12%.

R. V. D. Vijver et al. [36] worked on high-resolution canopy reflectance images of 32 potato plants. The researcher used spectral classifiers like partial test squares discriminant analysis (PLS-DA) and SVM to detect the presence of the early blight disease caused by Alternaria solani. The experiment achieved an accuracy of 92%. W. Kim et al. created a disease monitoring system for onion farms for the detection of downy mildew disease. The images were used to train a DL model developed by B. Zhou et al. [37] and achieved an accuracy of 87.20%.

3. Post-processing fusion framework

From the modern techniques discussed in the previous section, it is seen that majority of the researches end the experimentation on the classification stage, and the post-processing stage is often neglected. Therefore a post-processing technique is proposed. The working and architecture of the technique are discussed further.

The classification stage consists of one or more DL models. A DL model is a neural network having multiple hidden layers. Its last layer is called the output layer. The output layer consists of nodes equal
to the number of classes present in the test dataset. For each test image, the output layer produces a percentage score at every node in the output layer. This percentage score depicts the confidence of the DL model in the test image belonging to the corresponding class. The node having the maximum prediction score is considered as the final prediction of the model.

The percentage scores list obtained at the output layer of a DL model can be written as a list of key-value pair (class name/node, percentage score). The list of key-value pair is known as the prediction matrix.

After running many simulations of the existing DL models on the standard annotated crop disease datasets such as the Plant Village dataset and analyzing the prediction matrices produced, some useful findings are noticed:

i. Many a times, a test image misclassified by one DL model is accurately classified by another.

ii. A DL model achieving higher classification accuracy also shows misclassifications that are correctly classified by the inferior models.

iii. In majority of the cases, when the prediction value is compared, the model showing the correct classification of the test image often score better than the best prediction score of model, misclassifying the image.

All the above findings are analyzed, and a general equation (1) is formulated. The formula fuses the outputs of the existing DL models to achieve better accuracy that is not possible using DL models in isolation. The technique is named Post-processing Fusion Framework or PFF in short.

\[
P_{fm} = \max \text{Prob}(DP_1, DP_2, DP_3, \ldots, DP_n)
\]  

(1)

In equation 1, \(P_{fm}\) signifies the prediction score of the experimentation after using PFF on the test image and \(DP_n\) depicts the highest prediction score of the participating DL models.

4. Experiment
The steps of the experimentation are illustrated as a flowchart in figure 2 and listed ahead.

Figure 2: Methodology
Step 1: Input: the dataset is obtained and prepared for the experimentation.
Step 2: Normalization: the images of the dataset are resized and cropped for the experimentation.
Step 3: Split: the dataset is split into training and testing sub-sets.
Step 4: Augmentation: the training subset is augmented to reduce the risk of over-fitting.
Step 5: Classification: DL models are trained and tested on the training and testing subsets, respectively and the prediction matrices are prepared.
Step 6: Post-processing: the prediction matrices obtained from the classification step are post-processed using PFF.

4.1. Dataset
An open access dataset containing 54,305 annotated images provided by D. P. Hughes et al. [3] is used. It covers 14 plant species and 26 diseases. Overall, the dataset is classified in 38 classes.

4.2. Dataset normalization
The dataset is normalized, i.e. resized and cropped to 256x256 pixels. Table 1, shows that the DL models considered for the experiment use images of size 256x256 or a little less, hence it is the most optimal size. Further, the images can be reduced to this size without much loss of necessary information. The steps used for normalization are:

- Step 1: Detection of the smaller dimension, length or width
- Step 2: Reduction of the smaller dimension to 256 pixels while maintaining on aspect ratio
- Step 3: Centring of the image to 256x256 pixels canvas, chopping the remaining area.

4.3. Training and testing datasets
The image dataset is then split into two subsets using the 80:20 approach. 80% images per class are used for neural network training and the remaining 20% for the testing purpose. The pictures are selected at random for the subsets. The numbers of images in the training and testing subsets are 43,456 and 10,849, respectively.

4.4. Dataset augmentation
After normalization, the training subset images are subjected to augmentation. Augmentation is performed to obtain better results, on a relatively small dataset. Additionally, it reduces the risk of over-fitting of the model on the training set. In the present experiment, each image is augmented into 29 images, increasing the total number of training images from 43,456 to 12,60,224. The augmentation strategy has four main elements:

- Image rotation – it is helpful to mimic alternate angles from which an image can be captured.
- Scaling – it is helpful in mimicking different distances from which an image can be captured.
- Brightness – it is helpful in mimicking varying lighting conditions in which an image can be captured, like over and underexposure especially during noon and dusk, respectively.
- Clipping – it is helpful in mimicking the event when the area of interest is only a small part of the actual image.

Additional pre-processing does not improve the classification performance in case of DL models [38]. Hence, it is not considered. Further, as the DL models using Convolutional Neural Networks automatically identifies the significant and insignificant features in images, so the complicated step of segmentation is also avoided.

4.5. Model training
After augmentation, the next step is neural network training. Five prominent deep neural networks are selected for the purpose: (i) AlexNet [18], (ii) NIN [39], (iii) ZFNet [40], (iv) LeafNet [41], and (v) SqueezeNet [42]. These models are trained, tested and implemented using Caffe DL framework using
python programming language on a machine with NVIDIA GTX1080Ti GPU card, using CUDA parallel programming platform on Linux machine (Ubuntu 16.04 operating system).

The input size of the images required for each neural network is shown in table 1. So the images of the training subset are cropped accordingly.

### Table 1: Neural network input image size

| Neural Network | Input Image Size |
|----------------|------------------|
| AlexNet [18]   | 227x227x3        |
| NIN [39]       | 224x224x3        |
| ZFNet [40]     | 225x225x3        |
| LeafNet [41]   | 256x256x3        |
| SqueezeNet [42]| 227x227x3        |

The hyper-parameters of the DL models are optimized using the random search approach. The hyper-parameters used in the final experimentation are listed in table 2. The learning rate starts from $1.5625 \times 10^{-2}$ and is annealed by a factor of one half after every one-tenth of the total number of mini-batches, i.e. 7,876 images, and rested at $3.0518 \times 10^{-5}$.

### Table 2: CNN parameters and solver training hyper-parameters

| Parameter             | Value          | Parameter             | Value          |
|-----------------------|----------------|-----------------------|----------------|
| Epoch Size            | 1260224        | Gamma ($\gamma$)      | 0.5            |
| Epochs                | 2              | Weight Decay          | 0.0002         |
| Batch Size            | 32             | Base Learning Rate ($\alpha$) | $1.5625 \times 10^{-2}$ |
| Batches / Epoch       | 78764          | Learning Rate Decrease Policy | Step          |
| Momentum ($\mu$)      | 0.5            | Step Size             | 7876           |

Training of all the five DL models is performed on the augmented training dataset and the models weight biases are recorded and cached.

### 4.6. Post-processing fusion framework

The test images are passed one by one from the trained DL models and the probability matrices are recorded. Further, the fusion framework is applied and the final result i.e. the final prediction is obtained and recorded. The results are discussed in the following section.

### 5. Experimentation Results

The results of the experiment are recorded at two levels: (a) after classification, and (b) after PFF. The performance of the methodology is evaluated using four performance metrics: top-1 accuracy ($O_1$), top-5 accuracy ($O_5$), Mean Reciprocal Rank (MRR) [43] and Mean Average Precision (MAP) [44]. The results obtained are discussed further.

#### 5.1. Obtained results

At level $a$, i.e. after classification, the individual trained models are validated on the test sub-dataset. At level $b$, the trained model’s outputs are fused using PFF. The results obtained from the both levels are recorded and tabulated in table 3 and plotted in figure 3.

### Table 3: Performance of CNN model architectures for classification on test dataset

| Level | Model     | $O_1$ (Value %, no. of Iterations) | $O_5$ (Value %, no. of Iterations) | MRR (Value %, no. of Iterations) | MAP (Value %, no. of Iterations) | Misclassifications |
|-------|-----------|-----------------------------------|-----------------------------------|----------------------------------|----------------------------------|-------------------|
| $a$   | AlexNet   | 96.00 (63008)                     | 99.95 (78760)                     | 97.85 (63008)                    | 96.87 (55132)                   | 434               |
| $a$   | LeafNet   | 96.24 (78762)                     | 99.86 (70884)                     | 97.92 (78762)                    | 97.03 (78762)                   | 408               |
| $a$   | NIN       | 94.14 (78760)                     | 99.90 (78760)                     | 96.80 (70884)                    | 95.40 (78760)                   | 636               |
| $a$   | SqueezeNet| 89.66 (78760)                     | 99.44 (78760)                     | 93.98 (78760)                    | 91.61 (78760)                   | 1121              |
| $a$   | ZFNet     | 96.01 (78760)                     | 99.94 (55132)                     | 97.88 (55132)                    | 96.92 (55132)                   | 433               |
| $b$   | PFF       | 99.33 (78760)                     | 99.99 (55132)                     | 99.61 (78760)                    | 99.46 (78760)                   | 70                |

It can be clear from the table 3 that the LeafNet DL model scores the highest average top-1 accuracy ($O_1$) of 96.24% at 78762 iterations. The best score in average top-5 accuracy ($O_5$) is achieved by AlexNet, with a score of 99.95% at 78760 iterations. The best Mean Reciprocal Rank (MRR) and
Mean Average Precision (MAP) are again achieved by LeafNet at 97.92% and 97.03%, respectively at 78762 iterations. After implementation of PFF, the scores at all the performance metrics increases. The value of $O_1$ increases from 96.24% (LeafNet) to 99.33%; $O_5$ increases from 99.95% (AlexNet) to 99.99%; MRR increases from 97.92% (LeafNet) to 99.61%; and finally, the value of MAP increased from 97.03% (LeafNet) to 99.46%. Additionally, the fusion framework results in decreasing the number of misclassifications from 3.76% (408 images) to 0.67% (70 images).

Further, after investigating the misclassified images, the following reasons may be stated for these misclassifications:

i. Similar background noise in multiple images, like the hand of the image capturer.
ii. The picture has small Region of Interest (ROI).
iii. Various diseases have similar symptoms, even confusing the human eye.

Figure 3: Performance Measurement over Number of Iterations: (a) Average Top 5 accuracy, (b) Average Top 5 accuracy, (c) Mean Reciprocal Rank, (d) Mean Average Precision and (e) Number of misclassifications
5.2. Importance of number of iterations

The number of iterations is an important metric to keep the space and time complexity in check. Using an excessive number of iteration may waste time in model training, this time can be utilised for optimisation of the model or its use in the field. Therefore, keeping the number of iterations in check is an important parameter.

In this experiment, the number of training images are 43,456, and are augmented to 12,60,224. The batch size is kept at 32 (optimised using the random search algorithm). The maximum number of iterations possible are 78,762 (epoch size * number of epochs/batch size, 12,60,224 * 2/32) per epoch. For comparison of iterations, the trained models are screenshot at every one-tenth of the total number of iterations, i.e. at every 7876 iterations. The results are plotted in various graphs, shown in figure 3. The results show a consistent trend in all the five graphs (3(a) to 3(e)), i.e. for all four performance metrics (O1, O5, MRR and MAP). The performance initially enhances with an increase in the number of iterations. But, after reaching a threshold (47,256 for AlexNet, 55,132 for NIN, 39,380 for LeafNet, 63,008 for SqueezeNet, and 55,132 for PFF), the increase in the number of iterations became insignificant and the performance stagnates.

5.3. Comparison with existing approaches

The results of the existing methods and the present experimentation are shown in table 4. From the results, it is evident that the proposed approach outperformed the existing methods by a fair margin.

| Year | Diseases covered                                                                 | Accuracy | Reference |
|------|----------------------------------------------------------------------------------|----------|-----------|
| 2020 | Apple scab, Bacterial spot, Black rot, Cedar apple rust, Cercospora leaf spot, Common rust, Early blight, Esca (Black Measles), Gray leaf spot, Haunglongbing (Citrus greening), Late blight, Leaf blight (Isaropsis Leaf Spot), Leaf Mold, Leaf scorch, Northern Leaf Blight, Powdery mildew, Septoria leaf spot, Spider mites, Target Spot, Tomato mosaic virus, Tomato Yellow Leaf Curl Virus | 99.33%   | This study |
| 2020 | Early Blight                                                                    | 92%      | [36]      |
| 2020 | Downy mildew                                                                   | 87.20%   | [37]      |
| 2019 | Common rust, Northern leaf blight                                               | 88%      | [45]      |
| 2019 | Nitrogen deficiency, phosphorous deficiency, potassium deficiency                | 90%      | [46]      |
| 2019 | Northern Corn Leaf Blight, Common Rust, Grey Leaf Spot                           | 92.85%   | [34]      |
| 2018 | Drought stress                                                                  | 98.97%   | [10]      |
| 2017 | Northern Leaf Blight                                                            | 96.7%    | [24]      |
| 2015 | Exserohilum Turcicum, Brown spot, Grey spot, Curvarularia Lunata, Round spot and Corn rust | 92.82%   | [7]       |
| 2015 | Leaf blight, brown spot, grey leaf spot, rust spot, Curvarularia leaf spot, small spot | 94.46%   | [47]      |

6. Conclusion and future scope

In this work, various DL models are investigated, hyper-parameters are optimized, and a novel Post-processing Fusion Framework (PFF) is proposed. The PFF achieved the classification accuracy of 99.33% on the Plant-Village dataset, using the 'training from scratch' approach. The previous best accuracy in the same category on the same dataset was achieved by Mohanty et al. [38] in 2016 at 98.37%. Thus an increase of 0.96% accuracy is achieved. Therefore, the proposed approach appears to be very promising to be used in real-cultivation conditions. Hence, it can be concluded from the results that the post-processing stage of the experimentation for crop disease detection plays a vital role in the efficient and effective classification. It should not be excluded while carrying similar works.

Further, the proposed post-processing technique is simple to use and is computationally less expensive. The approach helps to overcome the limitations of individual DL models by selecting the correct classifications from the outputs of each participating model. It enhances the overall accuracy and robustness of the system. Hence, such frameworks are quite useful in the field.

6.1. Future suggestions

Future research may focus on the experimentation of the proposed framework using techniques like transfer learning. Moreover, the initial learning of transfer learning approach may be executed on the
augmented Plant-Village [3] dataset produced as a by-product of the present investigation. Plant disease dataset should be used instead of popular image-net [48] for pre-training for crop disease detection. It may prove to be more successful for this specified task as the pre-trained model will be quite proficient in the domain and will require very less fine-tuning. Thus, it may result in the realization of enhanced results on a relatively small dataset, like digipathos [49]. Further, the final model may be combined with a pesticide suggestive system and implemented as a smartphone app. The resulting system may assist the farmers with real-time remedial measures for crop security against diseases. Thus, it will help farmers in maximizing crop yield and minimizing crop loss due to diseases. Consequently, improving the overall crop-yield and resulting as a positive step towards a very healthy agrarian future.

References
[1] R. N. Strange and P. R. Scott, “Plant Disease: A Threat to Global Food Security,” Annu. Rev. Phytopathol., vol. 43, no. 1, pp. 83–116, 2005, doi: 10.1146/annurev.phyto.43.113004.133839.
[2] “Goal 2: Zero Hunger – United Nations Sustainable Development.” https://www.un.org/sustainabledevelopment/hunger/ (accessed Aug. 18, 2020).
[3] D. P. Hughes, M. Salathé, and M. Salathe, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” CoRR, vol. abs/1511.0, 2015, doi: 10.1111/1755-099X.12257.
[4] S. Kai, Z. Liu, H. Su, and C. Guo, “A research of maize disease image recognition of corn based on BP networks,” in Proceedings - 3rd International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2011, 2011, vol. 1, pp. 246–249, doi: 10.1109/ICMTMA.2011.66.
[5] B. Sebastian, A. Unnikrishnan, and K. Balakrishnan, “GREY LEVEL CO-OCCURRENCE MATRICES: GENERALISATION AND SOME NEW FEATURES,” Int. J. Comput. Sci. Eng. Inf. Technol., vol. 2, no. 2, 2012, doi: 10.5121/ijcset.2012.2213.
[6] M. Cîlîmîkovic, “Neural Networks and Back Propagation Algorithm.”
[7] Z. Zhang, X. He, X. Sun, L. Guo, J. Wang, and F. Wang, “Image Recognition of Maize Leaf Disease Based on GA-SVM,” 2015, doi: 10.3303/CET1546034.
[8] S. Mirjalili, “Evolutionary Algorithms and Neural Networks,” Soft Comput. Intell. Syst., vol. 780, pp. 111–136, 2019, doi: 10.1007/978-3-319-93025-1.
[9] B. Schlkopf, A. J. Smola, and F. Bach, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, 2018.
[10] B. Jiang, P. Wang, S. Zhuang, M. Li, Z. Li, and Z. Gong, “Detection of maize drought based on texture and morphological features,” Comput. Electron. Agric., vol. 151, pp. 50–60, Aug. 2018, doi: 10.1016/j.compag.2018.03.017.
[11] X. E. Pantazi, D. Moshou, and A. A. Tamouridou, “Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers,” Comput. Electron. Agric., vol. 156, no. July 2018, pp. 96–104, Jan. 2019, doi: 10.1016/j.compag.2018.11.005.
[12] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[13] G. Wang, Y. Sun, and J. Wang, “Automatic Image-Based Plant Disease Severity Estimation using Deep Learning,” Comput. Intell. Neurosci., 2017, doi: 10.1155/2017/2917536.
[14] A. Z. Karen Simonyan, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” CoRR, vol. abs/1409.1, pp. 1–14, 2015.
[15] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision.”
[16] J. S. Kaiming He, Xiangyu Zhang, Shaoqing Ren, K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” CoRR, vol. abs/1512.0, pp. 1–12, 2015, doi: 10.1016/j.corr.2015.02.026.
[17] M. M. M. Brahami, B. Kamel, T. H. Bounediene, K. Boukhalia, and A. Moussauoi, “Deep Learning for Tomato Diseases: Classification and Symptoms Visualization,” Appl. Artif. Intell., vol. 31, no. 4, pp. 299–315, 2017, doi: 10.1080/08989055.2017.1315516.
[18] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Commun. ACM, vol. 60, no. 6, p. 2017, 2012.
[19] A. R. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, “Going Deeper with Convolutions,” CoRR, vol. abs/1409.4, pp. 1–9, 2014, doi: 10.1002/jetc.4820.
[20] K. Yamamoto, T. Togami, and N. Yamaguchi, “Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture,” Sensors (Switzerland), vol. 17, no. 11, pp. 1–13, 2017, doi: 10.3390/s17112557.
[21] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and A. Ranzato, “LEARNING LONGER MEMORY IN RECURRENT NEURAL NETWORKS.”
[22] J. G. Ha et al., “Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial
vehicles," J. Appl. Remote Sens., vol. 11, no. 04, pp. 04262 (1–14), 2017, doi: 10.1117/1.JRS.11.042621.

[23] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, “An in-field automatic wheat disease diagnosis system,” Comput. Electron. Agric., vol. 142, no. 1, pp. 369–379, 2017, doi: 10.1016/j.compag.2017.09.012.

[24] C. Dechant et al., “Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning,” Phytopathology, vol. 107, no. 11, pp. 1426–1432, Nov. 2017, doi: 10.1094/PHYTO-11-16-0417-R.

[25] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition,” Sensors (Switzerland), vol. 17, no. 9, 2017, doi: 10.3390/s17092022.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated Residual Transformations for Deep Neural Networks,” Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.05431.

[27] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497.

[28] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object Detection via Region-based Fully Convolutional Networks.” Accessed: Aug. 20, 2020. [Online]. Available: https://github.com/daijifeng001/r-fcn.

[29] A. C. Cruz, A. Luvisi, L. De Bellis, and Y. Amapatzidis, “X-FIDO: An Effective Application for Detecting Olive Quick Decline Syndrome with Deep Learning and Data Fusion,” Front. Plant Sci., vol. 8, no. October, pp. 1–12, Oct. 2017, doi: 10.3389/fpls.2017.01741.

[30] R. Ramachandran, D. C. Rajeev, S. G. Krishnan, and P. Subathra, “Deep learning in neural networks: An overview,” Neural Networks, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[31] B. Liu, Y. Zhang, D. J. He, and Y. Li, “Identification of apple leaf diseases based on deep convolutional neural networks,” Symmetry (Basel), vol. 10, no. 1, 2018, doi: 10.3390/sym10010011.

[32] J. Amara, B. Bouaziz, and A. Algergawy, “A Deep Learning-based Approach for Banana Leaf Diseases Classification,” in BTW, 2017, pp. 79–88.

[33] S. Ghosal, D. Blystone, A. K. A. K. Singh, B. Ganapathysubramanian, A. K. A. K. Singh, and S. Sarkar, “An explainable deep machine vision framework for plant stress phenotyping,” Proc. Natl. Acad. Sci., vol. 115, no. 18, pp. 4613–4618, 2018, doi: 10.1073/pnas.1716991115.

[34] M. Sibiya and M. Sumbwanyambe, “A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks,” AgrisEngineering, vol. 1, no. 1, pp. 119–131, Mar. 2019, doi: 10.3390/agriengineering10010009.

[35] J. G. Arnal Barbedo, “Plant disease identification from individual lesions and spots using deep learning,” Biosyst. Eng., vol. 180, no. 2016, pp. 96–107, Apr. 2019, doi: 10.1016/j.biosystemseng.2019.02.002.

[36] R. Van De Vijver et al., “In-field detection of Alternaria solani in potato crops using hyperspectral imaging,” Comput. Electron. Agric., vol. 168, no. March 2019, p. 105106, Jan. 2020, doi: 10.1016/j.compag.2019.105106.

[37] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization.” Accessed: Aug. 20, 2020. [Online]. Available: http://cnnlocalization.csail.mit.edu.

[38] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using Deep Learning for Image-Based Plant Disease Detection,” Front. Plant Sci., vol. 7, no. September, pp. 1–10, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[39] M. Lin, Q. Chen, and S. Yan, “Network In Network,” CoRR, vol. abs/1312.4400, 2013, [Online]. Available: http://arxiv.org/abs/1312.4400.

[40] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” CoRR, vol. abs/1311.2, p. 2013.

[41] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, “LeafNet: A computer vision system for automatic plant species identification,” Ecol. Inform., vol. 40, no. December 2016, pp. 50–56, 2017, doi: 10.1016/j.ecoinf.2017.05.005.

[42] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” CoRR, vol. abs/1709.0, pp. 1–14, 2017.

[43] H. Goëau et al., “The imageCLEF Plant Identification Task 2013,” in Proceedings of the 2Nd ACM International Workshop on Multimedia Analysis for Ecological Data, 2013, no. 1, pp. 23–28, 2011, doi: 10.1145/2509896.2509902.

[44] H. Goëau et al., “Plant Identification in an Open-world (LifeCLEF 2016),” in CLEF 2016 - Conference and Labs of the Evaluation Forum, Sep. 2016, no. LifeCLEF, pp. 428–439, [Online]. Available: https://hal.archives-ouvertes.fr/hal-01373780.

[45] A. S. Deshapande, S. G. Giraddi, K. G. Karibasappa, and S. D. Desai, “Fungal disease detection in maize leaves using haar wavelet features,” in Smart Innovation, Systems and Technologies, 2019, vol. 106, pp. 275–286, doi: 10.1007/978-3-030-14272-4_27.

[46] N. Leena and K. K. Saju, “Classification of macronutrient deficiencies in maize plants using optimized multi class support vector machines,” Eng. Agric. Environ. Food, vol. 12, no. 1, pp. 126–139, 2019, doi: 10.1016/j.eaef.2018.11.002.

[47] J. Luo, S. Geng, C. Xiu, D. Song, and T. Dong, “A curvelet-SC recognition method for maize disease,” J. Electr. Comput. Eng., vol. 2015, Jan. 2015, doi: 10.1155/2015/164547.

[48] O. Russakovsky et al., “ImageNet Large Scale Visual Recognition Challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[49] A. Johannes et al., “Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case,” Comput. Electron. Agric., vol. 138, pp. 200–209, 2017, doi: 10.1016/j.compag.2017.04.013.