Different Crop Leaf Disease Detection Using Convolutional Neural Network

Ashutosh Pawar[1], Mihir Singh[2], Swapnil Jadhav[2], Vidya Kumbhar[2](✉), T. P. Singh[2], and Sahil K. Shah[2]

[1] Hardcastle Agrotech Solutions Pvt. Ltd., Pune, India
[2] Symbiosis Institute of Geoinformatics, Symbiosis International (Deemed University), Pune, India
kumbharvidya@gmail.com

Abstract. Crop diseases are a considerable danger to the crop's health, affecting the yield. Timely detection is challenging due to a lack of infrastructure in many regions of the world. Since they result in the death of plants, the loss of their product, and the global food problem, plant diseases must be investigated. Crop disease detection has been made possible by recent advancements in computer vision, deep learning, and the growing worldwide adoption of smartphones. Convolutional Neural Networks have significantly improved classifying images in the past several years. The performance of deep learning-based techniques for plant disease recognition under actual circumstances is thoroughly examined in this research. The objective was to offer some principles for conducting a more thorough and realistic examination of deep learning-based approaches for disease recognition. Sequential Architecture was used to classify 38 diseases of 14 crops on a crop leaves image dataset containing 70,295 training and 17,572 testing images. A simple convolutional neural network has been proposed that detects crop diseases seamlessly. The maximum accuracy obtained was 95% on the 14th epoch. This was accomplished by following the Sequential Model. It is a cutting-edge network that can help new researchers who desire to conduct their studies in deep learning applications with an emphasis on agriculture.

Keywords: Convolutional Neural Network · Crop Disease Detection · Image Analytics

1 Introduction

Agriculture is one industry where Artificial Intelligence (AI) is still a growing technology. Modern farming has never been more advanced owing to AI-based technology. Deep learning principles are increasingly being used to solve agricultural issues. Crop disease detection has been made possible by recent advancements in computer vision, deep learning and the growing worldwide adoption of smartphones. Convolutional Neural Networks (CNN) have significantly improved classifying images in the past several

S. Tamane et al. (Eds.): ICAMIDA 2022, ACSR 105, pp. 966–979, 2023.
https://doi.org/10.2991/978-94-6463-136-4_85

years. Deep learning has developed as a promising method that can be used for computer vision tasks and data-intensive applications [1]. It has enormous potential and, like other fields, can also be used in agriculture. CNN is particularly effective in solving the problem of plant disease detection. The best approach for Object Recognition is widely acknowledged to be CNN. Faster Region-Based Convolutional Neural Networks (Faster R-CNN), Region-based Convolutional Neural Networks (R-FCN), and single-shot Multi-box detectors (SSD) make up the neural architecture. Depending on the application, each Neural Architecture should be able to be combined with any feature exactor. Preprocessing data is crucial for models to work accurately. Viral or fungal infections on the plant leaf can be challenging since their symptoms frequently overlap [2]. Farming is greatly aided by deep learning because it provides high-quality images; AI-based technology is particularly beneficial for agriculture because it makes detecting diseases, monitoring, scanning, and evaluating crops simpler. This technology helps to track the development of the crops. Farmers can also choose whether the crops are ready for harvest. [3] has worked on a Sugarcane Disease Recognition System using Deep Learning. Their trained model has accomplished its objective by correctly identifying and categorizing sugarcane images into healthy and unhealthy groups based on the pattern of leaves. The 95% with 60 epochs was the highest documented validation accuracy throughout training. [4] proposed an application for Plant Disease Identification. To minimize the yield loss and provide farmers with video training, the research uses CNN and numerous layers of Artificial Neural Network (ANN) with which they have created their custom deep learning algorithm. [5] have classified sugarcane images collected between 2019–2020; the researchers employed machine learning and an unsupervised approach by classifying all other crops and concentrating on classifying sugarcane as a different crop [6]. Moreover, [7] has proposed an Image-Based Disease Detection System Using Deep Learning for Tomato Leaves and Paddy Crop Disease Prediction Systems, respectively. Their goal was to offer a model for identifying the disease in paddy plants and determining the likelihood that it will occur, which can aid in making important decisions about the plant's health. The accuracy obtained was 99.84% and 90% for tomato and paddy plant disease detection systems. [8] have worked on Seasonal Crop Disease Prediction and Classification Using Deep Convolutional Encoder Network. They aimed to develop a unique method employing crop leaf images and convolutional encoder networks to identify crop diseases. [9] use CNN to detect plant diseases. Simulation study and analysis are carried out on sample images to determine the time complexity and the size of the diseased area. The model has been given 15 cases, of which 12 involve sick plant leaves. The accuracy obtained was 88.80%. [10] use the VGG architecture, the accuracy for disease identification was 99.53%. The authors found that changes in the structure of the plant leaves could be used to differentiate between diseased and healthy plants using the convolutional neural network. [11] proposes employing Convolutional Neural Networks (CNN) algorithms for the most effective real-time detection of diseases that impact the plant and the affected area. The convolutional neural network model was integrated into the TensorFlow backend system to categorize plant diseases (Fig. 1).

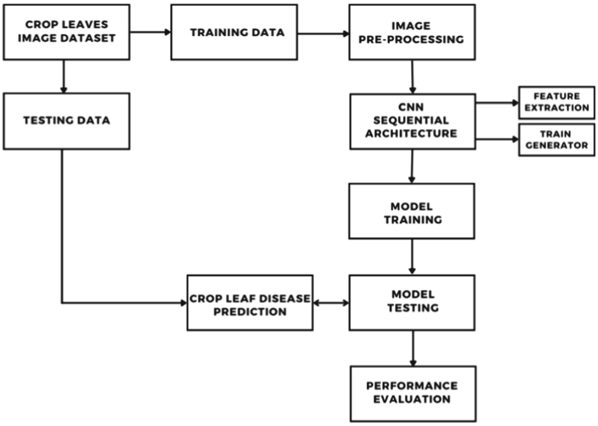

Fig. 1. Flow of Research Methodology.

2 Materials and Methods

2.1 Data Acquisition and Data Sources

The data comprises crop leaf images of 38 distinct diseases covering 14 different crops. This study used an augmented plant village dataset, that was created by researchers and organizations. The augmented datasets were integrated to build our model and give effective results on the Plant Village dataset.

Data Sources

Plant Village *(Original Dataset)* [12].

The organization Plant Village provided the primarily used dataset. The 54,303 healthy and diseased leaf images in the Plant Village collection are grouped into 38 categories by species and disease. However, the dataset has much bias. The following dataset, which researchers frequently utilized, was considered after a comprehensive literature review.

New Plant Village Dataset *(Augmented Dataset)* [13].

The Plant Village dataset was recreated by augmentation using offline techniques like image cropping, resizing, scaling, filtering etc. from the initial dataset. This dataset, broken down into 38 classes, contains 70,295 training images and 17,572 testing images of healthy and diseased crop leaves. The directory structure is preserved when the training and validation sets are split into an 80:20 proportion. Table 1 gives categorization of crop leaf images as per their types and number of diseases. Figures 2, 3, 4, 5, 6, 7, 8 and 9 gives insights about sample crop leaf images both healthy and diseased used in this research work.

2.2 Pre-processing Images

The first step of application development is to preprocess data. This step can include preprocessing steps like feature extraction, feature normalization, and data compression.

Table 1. Crop Leaf Image Dataset: 14 Crops and their Distribution amongst 38 Diseases [12]

Crop	Diseases
Apple	4
Blueberry	1
Cherry	2
Corn	4
Grape	4
Orange	1
Peach	2
Bell-Pepper	2
Potato	3
Raspberry	1
Soybean	1
Squash	1
Strawberry	2
Tomato	10

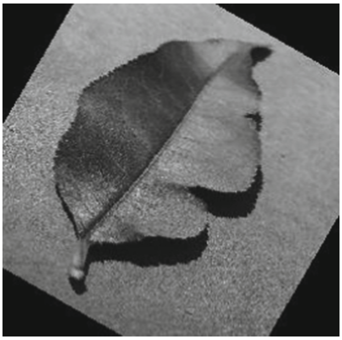

Fig. 2. Healthy Leaf (Peach)

For example, CNN requires the input images to be in an array of the same size. Each image is split into three arrays: one for passing through the network layers, one for holding scores computed by the network, and one for passing back to the starting location. The ImageDataGenerator is used to preprocess the data, which is a class in the Keras library. Our model accesses the Sequential API to improve each image as it receives new data. The model is created and trained on the training image data sets, which are scaled, and a validation spilt of 20% is set to it.

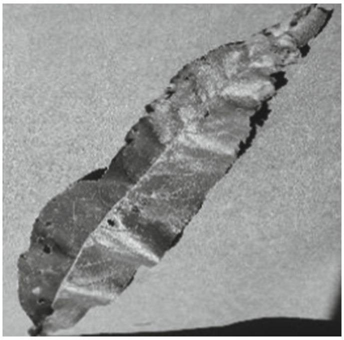

Fig. 3. Bacterial Spot (Peach)

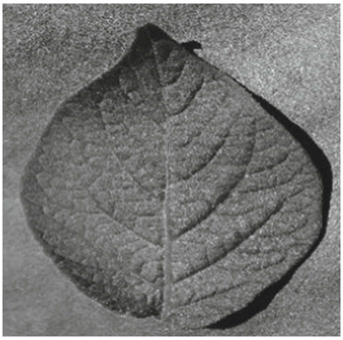

Fig. 4. Healthy Leaf (Potato)

Fig. 5. Leaf Blight (Potato)

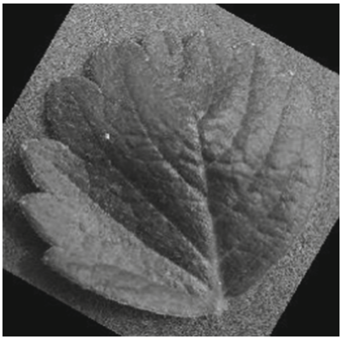

Fig. 6. Healthy Leaf (Strawberry)

Fig. 7. Leaf Scorch (Strawberry)

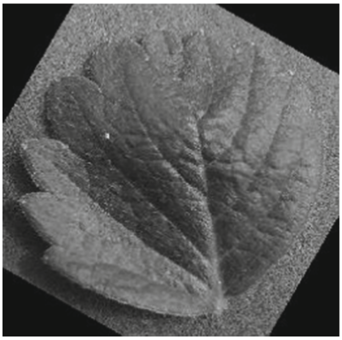

Fig. 8. Healthy Leaf (Tomato)

2.3 Data Augmentation

Data augmentation creates additional, modified data by modifying an existing dataset. It is recommended when initial data is insufficient for training or if we want to improve our model's performance. We have used the flow_from_directory function of the Keras

Fig. 9. Mosaic Virus (Tomato)

Library. The images in the data are set to 256 pixels in height and width, the recommended number of pixels [14]. The custom function creates batches of enhanced data using a directory's path and then produces batches in an infinite loop. The quantity of training samples used in a single iteration is called batch size [12].

2.4 Convolutional Neural Network (CNN)

CNN are a subset of neural networks that are particularly effective in classifying and recognizing images. Contrary to conventional methods, CNNs do not require manual feature extraction; instead, they can learn high-level features directly from the original image. It is shown that CNNs outperform conventional feature extraction techniques in recognizing plant types and diseases [15]. A sequential method is sufficient for a short stack of layers, where each layer has one input tensor and one output tensor. This strategy is put into practice using the Sequential API. The core idea of Sequential API is to arrange the Keras layers sequentially, hence the name. Most ANNs also include sequentially ordered layers, and data moves through each layer in the intended order until it reaches the output layer.

Convolution Layer. The first layer employs input images, kernels, or filters to understand the connections between features and extracts features from the input image. An input image is processed to extract 32 features [16].

ReLU Layer. Rectified Linear Unit (ReLU), refers to a non-linear operation. The neurons are activated via rectified linear activation. To bring non-linearity to CNN, ReLU is utilized [17].

The output is in the form as shown in Eq. 1.

$$f(\mathrm{x}) = \max(0, \mathrm{x}) \tag{1}$$

Table 2. Inference features for Model Compilation.

Feature	Description
Optimizer *(Adam).*	Adam optimization technique extends stochastic gradient descent with an adaptive learning rate.
Loss *(categorical_crossentropy)*	The Categorical cross-entropy was the chosen loss function. It is a widely used loss function for multi-class classification models with two or more output labels.
Metrics *(accuracy)*	It is a metric function that calculates how often predictions equal labels.

Pooling Layer. It reduces the number of parameters and saves crucial data for further model development. Various pooling forms exist such as Max Pooling and Average and Sum pooling [18].

Flattening Layer. After pooling, the next step is to convert the complete matrix into a vertical vector. The 4D array is flattened to a 1D array using the Flatten layer aligning with input layer. Some neurons are freed/relaxed by dropout [19].

Fully Connected Layer. The input layer receives the flattened vector. A model is created using these learned properties. Additionally, the outputs are categorized using an activation function such as the SoftMax or Sigmoid.

In our work, we have utilized SoftMax as disease classes are more than two resulting in multiclass classification problem. SoftMax converts a vector of numbers into a vector of probabilities. In this case, the likelihood of each value in the vector is inversely proportional to its relative scale. Additionally, after each layer, the output shape of the CNN reduces as it learns characteristics and retains relevant information [20].

While training from scratch necessitates a more extensive training set. However, very few models have been developed using a primary CNN from scratch in the past. We have trained the model on input image dataset from scratch using the **Sequential Model**. Using Kera's. It allows us to create a model layer-by-layer. Sequential architecture was used to classify 38 diseases of 14 crops on crop leaves image dataset containing 70,295 training and 17,572 testing images.

The model was compiled using the **model.compile()** function of the Keras Library. The function creates an object with training and inference characteristics made up of layers (Table 2).

The model was trained using the **model.fit()** function of the Keras' library. The function trains the model for parameters we assign based on the training dataset. Table 3 discusses the inference features set for training the model (Fig. 10).

Table 3. Inference features for Model Training.

Feature	Description
train_generator	The method of storing the training data to read the images from a NumPy array.
Epochs	Indicator for the number of passes on the entire training data (Forward & Backward Pass).
steps_per_epoch	Indicator for the total number of steps to yield the training data.
validation_data	Indicator to evaluate the loss and any model metrics.
validation_steps	Indicator for the total number of steps to yield for the validation data.

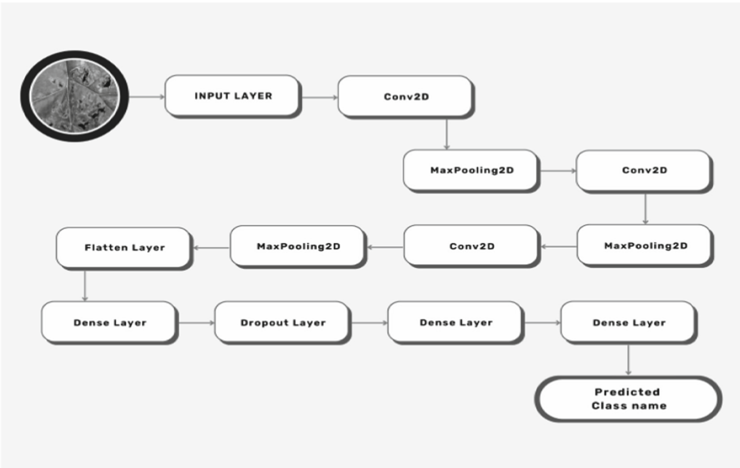

Fig. 10. Sequential Architecture for Crop Leaf Disease Detection

3 Results and Discussions

3.1 Data Preprocessing

Preprocessing was required to get images ready as input to the model. For instance, the fully connected layers of CNN needed each image to be kept in an array of the same size. Model preprocessing reduces the time required for training the model and speed up the model inference. The ImageDataGenerator class was used to preprocess the data; it is a class in the Kera's library. The images were improved in real time while the model was still learning. Any arbitrary adjustments were made to the training images as they were fed to the model. This has enhanced the model using less overhead RAM. The images were scaled, and a validation split of 20% was set to it. Data augmentation, which generates extra, changed data from the training set, was also used to increase the training artificially. Data augmentation was done to prevent overfitting and improve the model's performance. From Kera's library flow from directory technique was utilized for augmentation. The images in the dataset were set to a size of 256 × 256 pixel size. The method created batches of improved data and then produced batches in an endless loop.

3.2 Crop Disease Identification Model

The model successfully identified the potential disease the plant leaf could suffer after being tested using the testing data. The model was trained for 15 Epochs on a sample size of 2493 Images. The maximum accuracy obtained was 95% on the 14th epoch using the Sequential Architecture. A sequential method was sufficient using a short stack of layers, where each layer had one input tensor and one output tensor. This strategy was put into practice using the Sequential API. The core idea of Sequential API was to arrange the Keras layers sequentially. The model architecture consisted of three convolutional layers, three max-pooling layers, one flatten layer, three dense layers, and one dropout layer. The optimizer that was used to have an adaptive learning rate was Adam. The loss function was categorical cross-entropy, used to classify two or more output labels. The model obtained a loss of 0.18%.

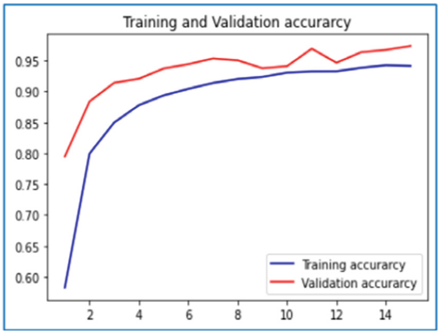

Fig. 11. Training Accuracy vs. Validation Accuracy.

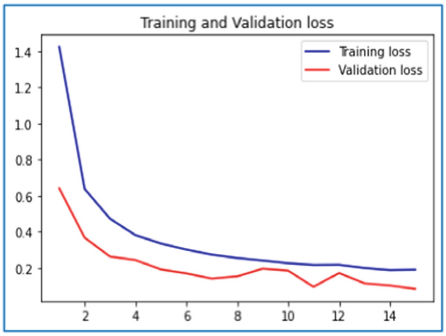

Fig. 12. Training Loss vs. Validation Loss.

Table 4. Performance Evaluation.

Parameter	Value
Loss	0.1896
Accuracy	0.9525
Validation Loss	0.0836
Validation Accuracy	0.9736

Table 5. Evaluation Metrics for Model Training.

Parameter	Value
Loss	It gives a notion about performance of model in terms of predicted results. More is the loss, model will be less accurate and perform badly.
Accuracy	It is a ratio of number of classifications a model correctly predicts over the total number of predictions made
Validation Loss	It focuses on "How well our model fits new data."
Validation Accuracy	To evaluate model, this accuracy is calculated on the unseen dataset (not used during training).

3.3 Accuracy and Validation Plots

Figure 11 indicates that our model trained well and increased accuracy during training over the number of epochs. Higher accuracy was obtained during the 14th epoch, which was 95.25%.

Figure 12 indicates that the model loss dropped significantly during training. This is a good sign, as the lesser the loss, the better our model will perform on test data.

Tables 4 shows present results obtained during performance evaluation of the model while Table 5 discusses the terms used while evaluating the performance.

3.4 Model Predictions

Figure 13 shows results obtained for sample crops using test dataset. It shows identified class labels by the model after training it successfully.

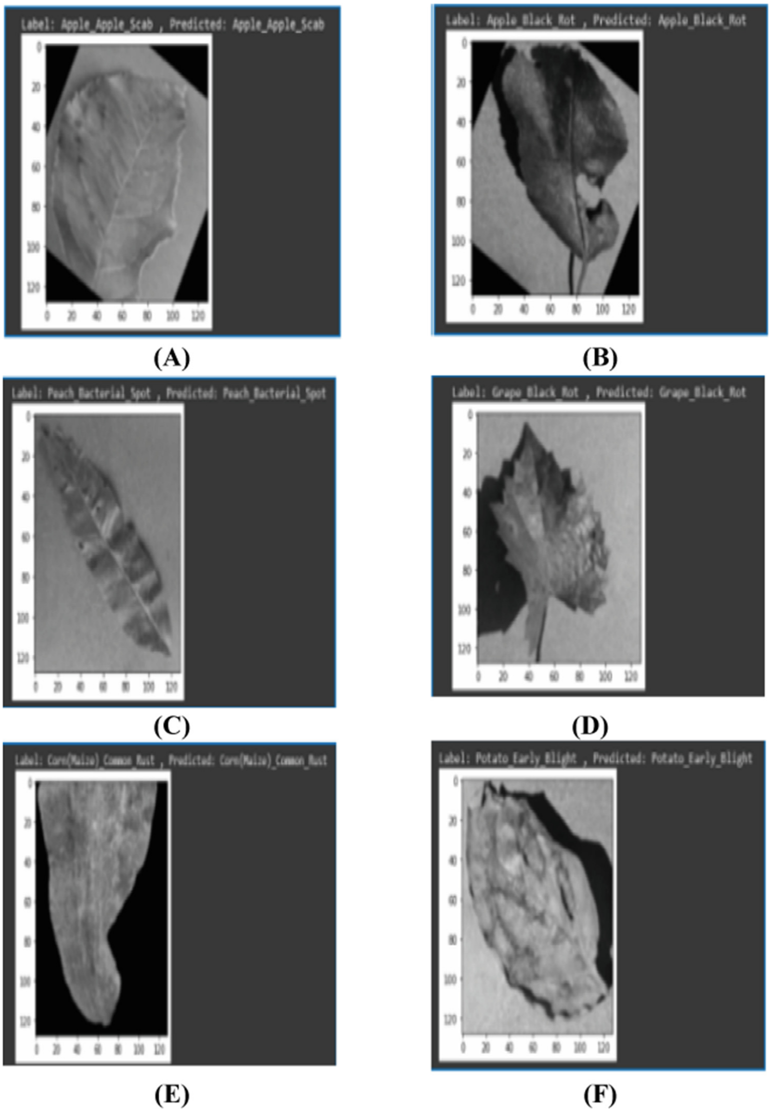

Fig. 13. **[A to F]** Prediction of Crop Diseases with Identified Class Name.

4 Conclusion

The performance of deep learning-based techniques for plant disease recognition under actual circumstances is thoroughly examined in this research. The current study offers principles for conducting a more thorough and realistic examination of deep learning-based approaches for disease recognition. A simple convolutional neural network has been used that detects crop diseases seamlessly. It is a cutting-edge network that can help new researchers who desire to conduct their studies in deep learning applications with an emphasis on agricultural practices. The literature research revealed that new crop disease detection studies were concentrated on transfer learning due to its advancement. Transfer learning techniques, however, need GPUs, which shorten the model training period. The model successfully identified the potential disease the plant leaf could suffer after being tested using the testing data. We observed that there is a substantial scope to utilize potential of CNN for automated and accurate crop disease identification. The images used for training could have been captured under different lighting conditions, which might have impacted the appearance of a crop leaf and made it more challenging for deep-learning models to identify diseases correctly. A diversified dataset will improve the generalizability of the model and enable it to make precise predictions on new, unseen data. If the size of input dataset is increased, better results can be achieved. This research provides future directions to use CNN in agricultural practices to improvise the food production and minimize the yield losses.

References

1. M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, p. 1, Dec. 2015, doi: https://doi.org/10.1186/s40537-014-0007-7.
2. A. Kumar, S. Sarkar, and C. Pradhan, "Recommendation System for Crop Identification and Pest Control Technique in Agriculture," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, Apr. 2019, pp. 0185–0189. doi: https://doi.org/10.1109/ICCSP.2019.8698099.
3. S. V. Militante, B. D. Gerardo, and R. P. Medina, "Sugarcane Disease Recognition using Deep Learning," in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Yunlin, Taiwan, Oct. 2019, pp. 575–578. doi: https://doi.org/10.1109/ECICE47484.2019.8942690.
4. M. Munnangi, "Crop: Plant Disease Identification Using Mobile App.," *Towards Data Science*. https://towardsdatascience.com/crop-plant-disease-identification-using-mobile-app-aef821d1a9bc
5. J. Y. Lee *et al.*, "Mapping Sugarcane in Central India with Smartphone Crowdsourcing," *Remote Sens.*, vol. 14, no. 3, p. 703, Feb. 2022, doi: https://doi.org/10.3390/rs14030703.
6. B. A. M. Ashqar and S. S. Abu-Naser, "Image-Based Tomato Leaves Diseases Detection Using Deep Learning," vol. 2, no. 12, 2018.
7. School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, India. *et al.*, "Paddy Crop Disease Prediction- A Transfer Learning Technique," *Int. J. Recent Technol. Eng. IJRTE*, vol. 8, no. 6, pp. 1490–1495, Mar. 2020, doi: https://doi.org/10.35940/ijrte.F7782.038620.
8. A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, "Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network," *Circuits Syst. Signal Process.*, vol. 39, no. 2, pp. 818–836, Feb. 2020, doi: https://doi.org/10.1007/s00034-019-01041-0.
9. G. Shrestha, Deepsikha, M. Das, and N. Dey, "Plant Disease Detection Using CNN," in *2020 IEEE Applied Signal Processing Conference (ASPCON)*, Kolkata, India, Oct. 2020, pp. 109–113. doi: https://doi.org/10.1109/ASPCON49795.2020.9276722.
10. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, Feb. 2018, doi: https://doi.org/10.1016/j.compag.2018.01.009.

11. S. Y. Yadhav, T. Senthilkumar, S. Jayanthy, and J. J. A. Kovilpillai, "Plant Disease Detection and Classification using CNN Model with Optimized Activation Function," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, Jul. 2020, pp. 564–569. doi: https://doi.org/10.1109/ICESC48915.2020.9155815.
12. David. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, [Online]. Available: http://arxiv.org/abs/1511.08060
13. S. Mohanty, "Plant Village Dataset." https://github.com/spMohanty/PlantVillage-Dataset
14. Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks: Deep learning-based crack damage detection using CNNs," *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, May 2017, doi: https://doi.org/10.1111/mice.12263.
15. J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agric.*, vol. 173, Jun. 2020, doi: https://doi.org/10.1016/j.compag.2020.105393.
16. L. Alzubaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, 2021.
17. W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units." arXiv, Jul. 19, 2016. Accessed: Dec. 14, 2022. [Online]. Available: http://arxiv.org/abs/1603.05201
18. M. Nitti, M. Murroni, M. Fadda, and L. Atzori, "Exploiting Social Internet of Things Features in Cognitive Radio," *IEEE Access*, vol. 4, pp. 9204–9212, 2016, doi: https://doi.org/10.1109/ACCESS.2016.2645979.
19. T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks." arXiv, Jan. 31, 2021. Accessed: Dec. 14, 2022. [Online]. Available: http://arxiv.org/abs/2102.00554
20. I. A. Kandhro *et al.*, "Impact of Activation, Optimization, and Regularization Methods on the Facial Expression Model Using CNN," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Jun. 2022, doi: https://doi.org/10.1155/2022/3098604.