Systems and Soft Computing 6 (2024) 200088

Contents lists available at ScienceDirect

Systems and Soft Computing

journal homepage: www.journals.elsevier.com/soft-computing-letters

CNN based plant disease identification using PYNQ FPGA

Vivek Karthick Perumal [a,*], Supriyaa T [a], Santhosh P R [a], Dhanasekaran S [b]

[a] Department of ECE, Sona College of Technology, Salem, Tamil Nadu, India
[b] Department ECE, Sri Eshwar College of Engineering, Coimbatore, India

ARTICLE INFO

Keywords:
CNN
Image Processing
Plant Disease
FPGA
PYNQ

ABSTRACT

This research presents a novel approach for plant disease identification utilizing Convolutional Neural Networks (CNNs) and the PYNQ FPGA platform. The study leverages the parallel processing capabilities of FPGAs to accelerate CNN inference, aiming to enhance the efficiency of plant disease detection in agricultural settings. The implementation involves optimizing the CNN architecture for deployment on the PYNQ FPGA, considering factors such as image size and learning rates. Through experimentation, the research refines hyper parameters, achieving improved accuracy and F1 scores. Visualizations using heat maps highlight the CNN's reliance on color, shape, and texture for feature extraction in disease identification. The integration of FPGA technology demonstrates promising advancements in real-time, high-performance plant disease classification, offering potential benefits for precision agriculture and crop management. This research contributes to the growing field of FPGA-accelerated deep learning applications in agro technology, addressing challenges in plant health monitoring and fostering sustainable agricultural practices.

1. Introduction

Agricultural production is an ancient and essential means of providing sustenance and income globally. Plants play a critical role in supporting human and animal life by providing food, oxygen, and other necessities. Efforts by governments and experts to enhance food production are crucial in addressing food security challenges. However, plant diseases can significantly impact agricultural output, affecting various parts of the plant such as stems, leaves, and branches. Different types of diseases, including bacterial and fungal, can affect crops, influenced by factors like climate.

Food insecurity remains a concern for many due to insufficient crop output, influenced by factors like climate change. Early detection of plant diseases is vital in preventing large-scale crop losses. Farmers often face challenges in applying the right amount of pesticides, and misuse can harm crops and farmland. Researchers have focused on developing tools for automated disease detection, utilizing technologies like deep learning [19] and neural networks.

In this context, the study emphasizes the use of a Deep Convolutional Neural Network (CNN) [21] for identifying infected and healthy leaves and detecting illnesses in plants. This approach leverages deep learning techniques to provide precise and quick results, benefiting both small and large-scale agricultural cultivation. The CNN model is designed to accommodate both healthy and diseased leaves, using images for training and determining the output based on input leaf characteristics. This technological advancement aids farmers in efficient disease detection and management in their crops.

This paper focuses on the development and implementation of a system that utilizes an FPGA platform and convolutional neural networks (CNNs) programmed with Python for plant leaf disease detection with reference of [1]. The main objective of this system is to detect and classify common plant diseases from leaf images captured [21] in the field. The detection and diagnosis of plant diseases are crucial for maintaining agricultural sustainability as they can lead to significant yield loss and a decrease in crop quality. However, traditional methods of plant disease detection are often time-consuming and not effective for early-stage detection.

To address this issue, the proposed system utilizes the high processing capabilities of the FPGA platform and advanced machine learning techniques provided by CNNs. The system employs a highly optimized CNN model that is trained on a sizeable dataset of leaf images to achieve high accuracy with low power consumption on the FPGA platform.

In this paper, the implementation of the system on the Xilinx PYNQ development board using the Python productivity suite as the FPGA design interface. Additionally, rest of the paper evaluate the system's

* Corresponding author:
E-mail addresses: vivekmalar@gmail.com (V.K. Perumal), supriyaadurai@gmail.com (S. T), santhoshcrunch60@gmail.com (S. P R).

https://doi.org/10.1016/j.sasc.2024.200088
Received 21 September 2023; Received in revised form 13 March 2024; Accepted 1 April 2024
Available online 13 April 2024
2772-9419/

Table 1
Machine Specifications.

Hardware & Software	Characteristics
Memory	8.0 GB
Processor	Intel ® Core™ i5- 9300H CPU @ 2.40 GHz
Graphics	NVIDIA GeForce RTX 2060 6GB GDDR6

Table 2
Dataset used for Classification.

S.no	Species	Class	No.ofimages
	PEPPER BELL	HEALTHY	1478
	PEPPER BELL	BACTERIAL SPOT	997
	POTATO	HEALTHY	152
	POTATO	EARLY BLIGHT	1000
	POTATO	LATE BLIGHT	1000
	TOMATO	HEALTHY	1591
	TOMATO	TARGET SPOT	1404
	TOMATO	MOSAIC VIRUS	373
	TOMATO	YELLOW LEAF CURL VIRUS	3209
	TOMATO	BACTERIAL SPOT	2127
	TOMATO	EARLY BLIGHT	1000
	TOMATO	LATE BLIGHT	1909
	TOMATO	LEAF MOLD	952
	TOMATO	SEPTORIA LEAF SPOT	1771
	TOMATO	TWO SPOTTED SPIDER MITES	1676

performance using a benchmark dataset of plant leaf images and implemented its potential as a portable and reliable plant disease detection and diagnosis tool that can be deployed by farmers and crop consultants.

2. Literature review

Emanuel Cortes utilized Convolutional Neural Networks (CNN) [18] and Modeling Adversarial Networks, achieving successful crop species and disease status classification. Prasanna Mohanty's deep convolutional neural network displayed remarkable accuracy in detecting various crops and identifying multiple diseases, with potential applications in precision agriculture. S. Khirade's 2015 study demonstrated the effectiveness of digital image processing and Backpropagation Neural Networks in plant disease detection, emphasizing the importance of automated methods. Together, these studies contribute to the advancement of plant disease detection, showcasing the versatility and efficacy of machine learning and neural network techniques in addressing challenges within agricultural systems.

The reviewed studies collectively highlight the strides made in utilizing machine learning, particularly neural networks, for plant disease identification. K. Muthukannan and colleagues innovatively applied LVQ, FFNN, and RBFN algorithms to categorize spot infections in leaves, showcasing the effectiveness of machine learning in diagnosing plant diseases and laying the groundwork for improved crop quality in the Indian economy. Malvika Ranjan's [16] study emphasized the use of artificial neural networks, specifically focusing on early and reliable identification of cotton leaf illnesses through image data processing.

Syafiqah Ishakais [14] and collaborators utilized an adjusted contrast, segmentation, and feature extraction algorithm, employing both multilayer feed-forward Neural Networks and radial basis function networks for classifying healthy and unhealthy leaves. The superiority of the RBF network highlighted the significance of network architecture in disease classification tasks.

Srdjan Sladojevic's [13] novel approach with deep convolutional neural networks showcased a quick and painless system setup for recognizing thirteen types of plant illnesses. The precision rates achieved underscored the potential of deep learning frameworks like Coffee [15] in plant disease recognition.

Deep convolutional neural network [20] demonstrated remarkable accuracy in detecting various crops and illnesses, emphasizing the practicality of such strategies for precision agriculture. S. Khirade [17] and colleagues' use of digital image processing and backpropagation neural networks further contributed to the diverse methodologies explored in plant disease detection.

In the broader context, these advancements signify a transformative era in agriculture, enabling early and accurate detection of plant diseases, thereby enhancing crop quality, ensuring food security, and

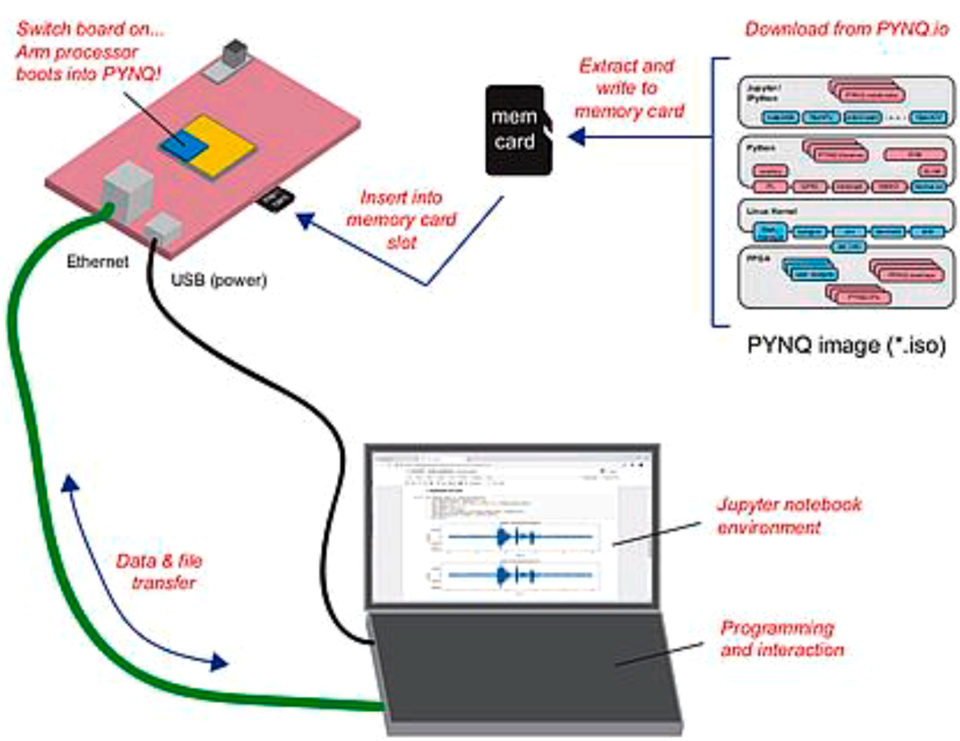

Fig. 1. Block Diagram.

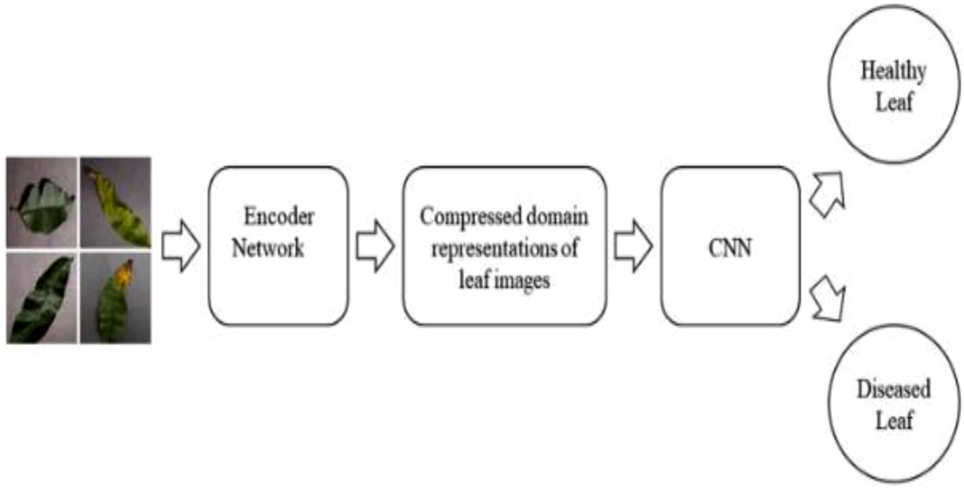

Fig. 2. Code flow diagram of image classification.

Fig. 3. Pre-processed images.

Table 3
Image Trail Information.

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

mitigating economic losses for farmers worldwide.

3. Problem statement

Agriculture is a vital sector of the Indian economy, providing employment to nearly half of the country's workforce. India is a leading producer of several crops, including pulses, rice, wheat, spices, and vegetables. The economic growth and livelihoods of farmers largely depend on the quality of their products, which is impacted by various factors such as plant growth and yield. Plant diseases pose a significant threat to crop production, leading to economic losses and ecological imbalances. Thus, it is crucial to detect plant diseases at an early stage. However, manual detection of diseases using leaf images is a

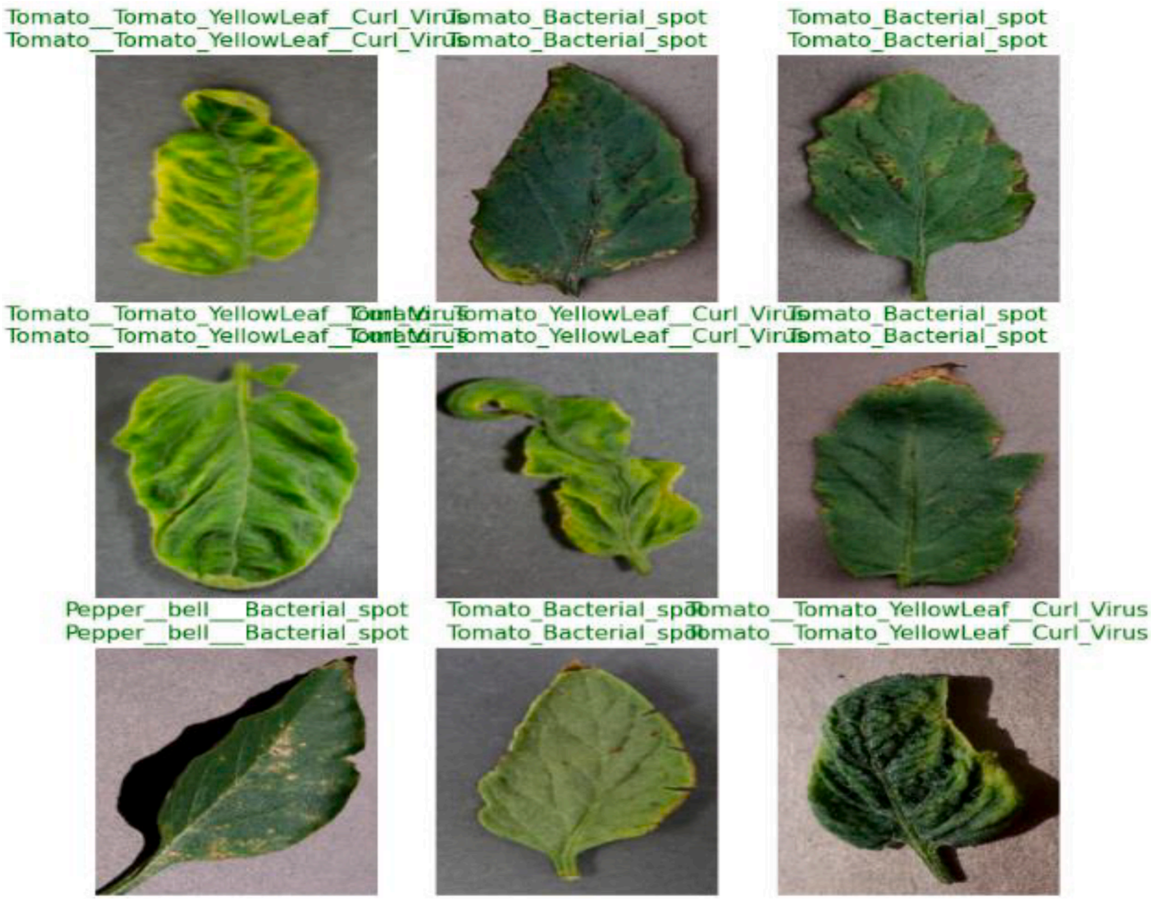

Fig. 4. Pre-processed images −2.

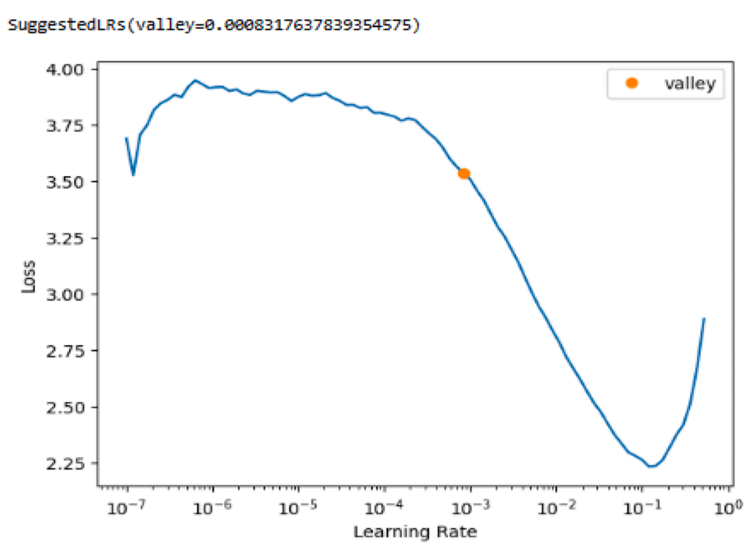

Fig. 5. learning rate vs loss rate.

cumbersome and time-consuming task. Therefore, there is a need for the development of automated disease detection techniques using computational methods. Disease symptoms are often visible in various parts of the plant, including leaves, making the development of an accurate and reliable system for disease detection and classification using images of plant leaves a top priority in the agriculture sector.

4. Proposal method

The proposed method for plant disease detection involves using an FPGA platform and convolutional neural networks (CNNs) to detect and classify common plant diseases from leaf images captured in the field with reference of [2]. The system's highly optimized CNN model is trained on a large dataset of leaf images to achieve high accuracy while maintaining low power consumption on the FPGA platform. The

epoch	train_loss	valid_loss	time
0	1.248195	2.559777	35:17
1	2.375299	1.813999	32:20
2	1.229128	0.782573	29:08
3	0.455866	0.171569	3:58:29
4	0.221892	0.134992	2:17:41

Fig. 6. Image Trails.

implementation of this system is done on the Xilinx PYNQ development board using the Python productivity suite as the FPGA design interface. The system's effectiveness is assessed using a benchmark collection of plant leaf photos, demonstrating its potential as a dependable and transportable tool for farmers and crop advisors to use for plant disease identification and diagnosis. The report will also address the challenges of deep learning for plant disease and object detection using CNNs and propose solutions accordingly.

5. Materials and methods

The section outlines the process of creating and deploying the plant disease classifier [21] based on CNN. The classification process involves three distinct phases, each addressing a specific task. Notably, all the work was carried out on a single machine, as detailed in Table 1, without the need for any additional computational resources.

The steps involved in building the classifier system include training the CNN model taking a dataset of labeled pictures of healthy and diseased plant leaves, using method such as data augmentation ortransfer learning to improve performance. Once trained, the model can be integrated into a system that receives new images of plant leaves, feeds them through the CNN, and returns the predicted classification result. The classifier system can be further enhanced by integrating it with other technologies such as drones or IoT devices to enable automated plant disease monitoring and management.

5.1. Data acquisition

The images used in this study to classify leaves of potatoes, tomatoes, and peppers were obtained from the publicly available 'Plant Village' Kaggle dataset [3], which includes a total of 20,487 images. The dataset includes a specific set of classes for each plant species, as outlined in Table 2. All images were captured under controlled conditions, which could introduce bias into the model. To test for this bias, a separate test dataset of 50 images was created by sourcing images from Google. This test dataset includes plant anatomy variations, in-field background data, and different stages of the diseases being studied (Figs. 1 and 2).

5.2. Data preprocessing

In this research, the dataset was divided into an 80 % training and a 20 % validation split. To enhance the training data, several techniques were employed, including random flipping, reflection padding mode, and zoom with crop (scale = (1.0,1.5)) with each approach having some weighted probability of being applied in each epoch in reference with [4]. However, it was observed that 'zoom with crop' inappropriately removed infected leaf areas, leading to its removal. Additionally, all images were resized by a compress function and normalized using RGB ImageNet-based statistics. Ultimately, the images' size was reduced to 224 × 224, and a sample of the pre-processed images is depicted in Fig. 3.

5.3. Classification by CNN

1) First Phase – The first phase of this study involves testing the impact of image size on the model's performance using 5 different image sizes ranging from150 × 150 to 255 × 255. Initially, pre-trained Resnet34 wt are downloaded, and all layers except for the final two are frozen. These final layers contain new weights tailored to the plant disease classification task and are trained separately without back propagating the gradients, using the 1cycle policy [5]. Once training of these layers is completed, the remaining layers are un-frozen, and a plot of learning rate vs. loss is created for fine-tuning purposes [6]. A suitable learning rate is chosen, and the model is re-run, with the resulting data recorded. This process is repeated for the remaining four image sizes, maintaining consistency in each trial, including the learning rate employed and steps undertaken. Table 3 displays the additional image sizes used during the trial.
2) Second Phase – Model Optimization-In the second phase of this study, the aim is to optimize the ResNet34 model by using the image size that was found to be most suitable in the first phase [7]. Additional augmentation settings, such as brightness changes (0.4, 0.7) and warp (0.5), are introduced to further improve the model's performance, as displayed in Fig. 4. The next step involves isolating the

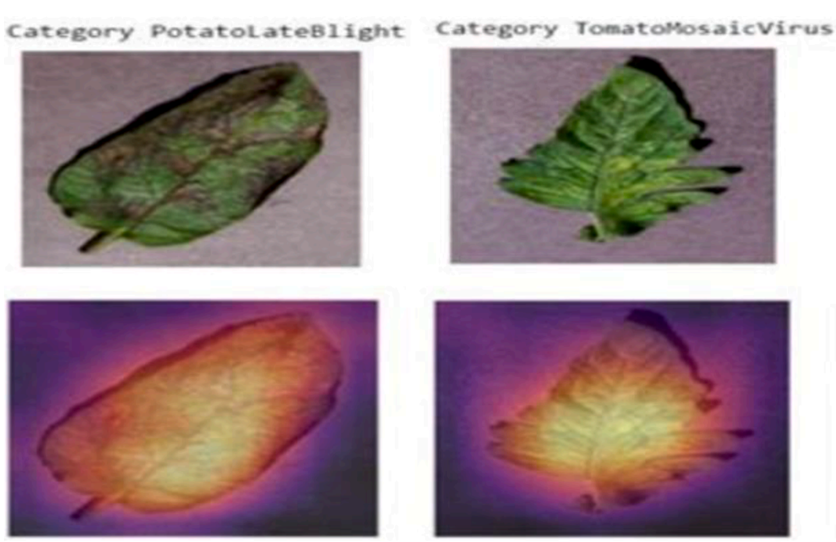

Fig. 7. Heat Map.

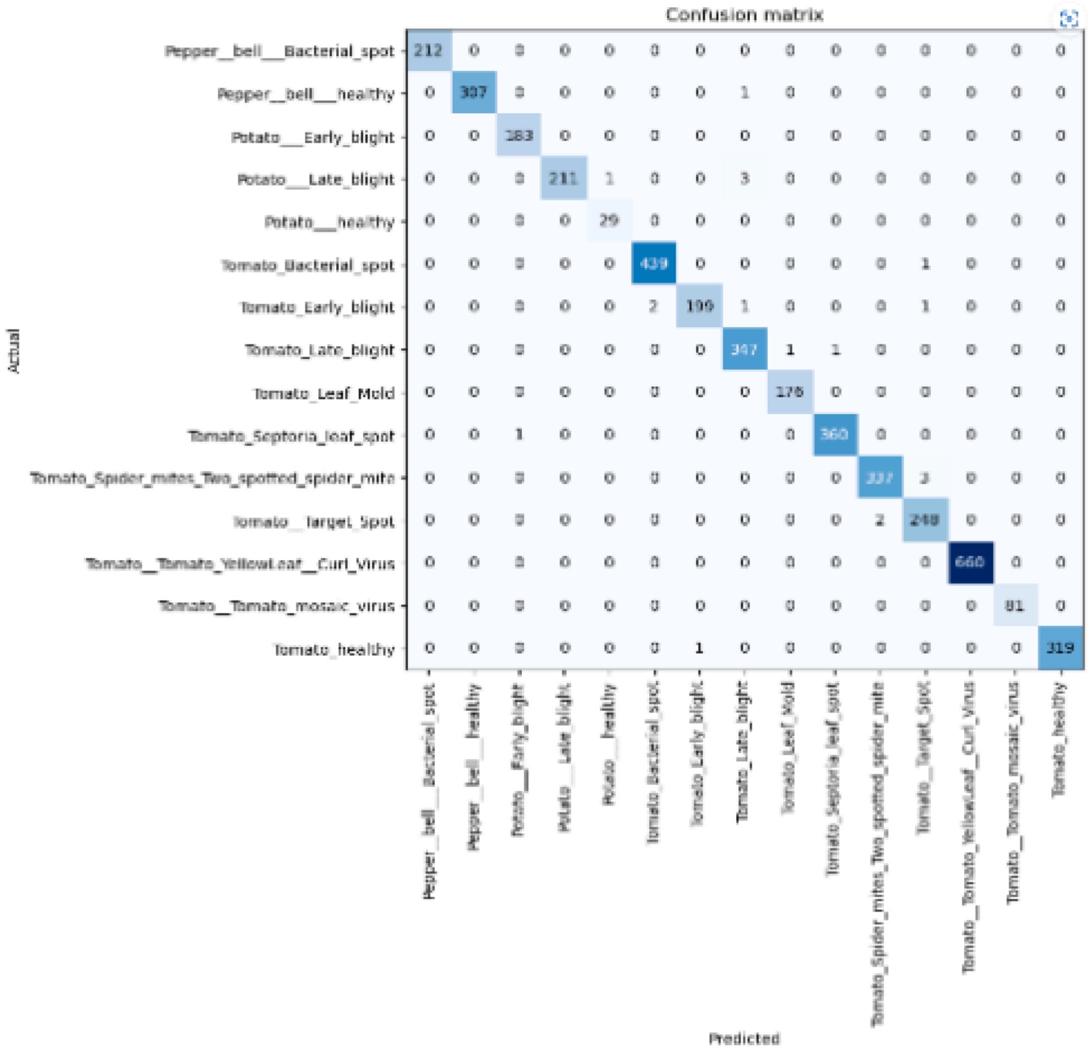

Fig. 8. Confusion Matrixes for Predicted Dataset.

```
In [29]: from PIL import Image
         img = Image.open('D:\\downloads\\plantvillage\\plant_disease\\Tomato__Tomato_mosaic_virus\\2d053972-2f30-414c-9de9-e37c98e44648_

         # Resize the image to 224x224
         img_resized = img.resize((224, 224))
         pred, pred_idx, probs = learn.predict(img_resized)
         im_t = cast(array(img_resized), TensorImage)
         # Print the predicted label and probability
         print(f"Predicted label: {pred}, probability: {probs[pred_idx]:.4f}")
         img

         Predicted label: Tomato__Tomato_mosaic_virus, probability: 1.0000
Out[29]:
```

Fig. 9. Result for the given input.

final two layers and training them at the default learning rate. Fine-tuning is then conducted by running multiple experiments with different learning rates and epochs, with the results being recorded to determine which settings are most effective in achieving optimal performance.

3) Third Phase – Visualizations-The third phase of this study focuses on generating visualizations for the validation and test datasets to aid in interpretation. Additionally, a web application is created by deploying the model, which involves storing the essential files in a GitHub repository and exporting the model as a pickle file. The GitHub repository is then connected to the unified platform, Render, to deploy the model [8]. To accomplish this task, the "Render Examples" GitHub repository is used as a guide. Overall, this phase involves creating visual aids to enhance the interpretation of the data, as well as deploying the model to create a usable web application.

6. Result

1) First Phase– During the experimentation process, the research group tested various image sizes and learning rates ranging from 1e-05 to 1e-04 for each model over four epochs. Surprisingly, the results revealed that an image size of 244 produced the highest accuracy and F1 score, contradicting the prior research suggesting that 224 × 224 image size is ideal for plant disease classification. Despite the marginal benefits of increasing the image size, the team concluded that an image size of 244 is the most optimal based on the obtained results [5]. Consequently, the group decided to employ this image size for the rest of the research work (Fig. 5).
2) Second Phase – Model Optimization-The research found that a learning rate range of 1e-05 to 1e-04 produced the best results. However, by fine-tuning this hyper parameter, there was a slight increase in accuracy (1.5 %) and F1-Score (1.3 %). Although the model appeared to be slightly under fitting on the final epoch, as shown in Fig. 6, this was corrected by systematically increasing the epochs number. At approximately the 10th epoch, a noticeable improvement in the fit of the model was observed. This resulted in an overall increasing in 2.8 % in accuracy and 3.1 % in F1-score, as displayed in Fig. 6. Overall, by fine-tuning the hyper parameters and increasing the number of epochs, the model's performance was significantly enhanced [9].
3) Third Phase – Visualizations-Through the analysis of heat maps, it was discovered that the CNN relies on color, shape, and texture to extract features of plant diseases, as shown in Fig. 3 and Fig. 4. In particular, color is a critical factor and helps differentiate similar diseases by providing an extra dimension of characterization [10]. This emphasizes the importance of RGB data in disease classification tasks, as noted earlier in the study. The CNN demonstrated effectiveness in recognizing features for all three species, including paper bell disease classes [11]. Overall, the study shows that color plays a key role in identifying plant diseases and can provide valuable insights into the inner workings of CNNs for disease classification [12].

7. Conclusion

In conclusion, the integration of Convolutional Neural Networks (CNNs) with the PYNQ FPGA platform for plant disease identification signifies a significant leap forward in agricultural technology. The research successfully optimized the CNN architecture for FPGA deployment, capitalizing on parallel processing capabilities to enhance the efficiency of disease detection. Through meticulous experimentation, hyperparameters were fine-tuned, resulting in improved accuracy and F1 scores. Visualizations using heat maps underscored the CNN's reliance on color, shape, and texture for robust feature extraction in disease identification. The utilization of FPGA technology showcases the potential for real-time, high-performance plant disease classification, promising advancements in precision agriculture and crop management. This innovative approach addresses the critical need for rapid and accurate plant health monitoring, contributing to sustainable agricultural practices. As a result, the study underscores the feasibility and efficacy of leveraging FPGA-accelerated CNNs for tackling challenges in plant disease identification in the agricultural domain (Figs. 7-9).

CRediT authorship contribution statement

Vivek Karthick Perumal: Writing – review & editing, Supervision, Project administration, Methodology, Investigation. **Supriyaa T:** Project administration, Methodology, Investigation. **Santhosh P R:** Methodology, Investigation, Conceptualization. **Dhanasekaran S:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] G. Shrestha, M. Deepsikha. "Das, N. Dey, Plant Disease DetectionUsing CNN, 2020 IEEE Applied Signal Processing Conference(ASPCON) (2020).
[2] S.I. Venieris, et al., fpgaConvNet: a Framework for Mapping Convolutional Neural Networks on FPGAs. FCCM, 2016.
[3] 'plantvillage' dataset obtain from Kaggle [Online], (2019) Available https://www.kaggle.com/datasets/emmarex/plantdisease.
[4] J. Boulent, S. Foucher, J. Théau, P. St-Charles, Convolutional neural networks for the automatic identification of plant diseases, Front. Plant Sci. 10 (2019), https://doi.org/10.3389/fpls.2019.00941. Available.
[5] C. Szegedy, 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].
[6] H.A. Atabay, A convolutional neural network with a new architecture applied on leaf classification, IIOAB J 7 (5) (2016) 226–331.
[7] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, https://doi.org/10.1145/3065386. Available.
[8] S. Jadhav, Convolutional neural networks for leaf image-based plant disease classification, IAES International Journal of Artificial Intelligence (IJ-AI) 8 (4) (2019) 328, https://doi.org/10.11591/ijai.v8.i4.pp328-341. Available.
[9] Y. Toda, F. Okura, How convolutional neural networks diagnose plant disease, Plant Phenomics 2019 (2019) 1–14, https://doi.org/10.34133/2019/9237136. Available.
[10] M. Ji, L. Zhang, Q. Wu, Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks, Information Processing in Agriculture (2019), https://doi.org/10.1016/j.inpa.2019.10.003. Available.
[11] X. Zhang, Y. Qiao, F. Meng, C. Fan, M. Zhang, Identification of maize leaf diseases using improved deep convolutional neural networks, IEEE Access 6 (2018) 30370–30377, https://doi.org/10.1109/access.2018.2844405. Available.
[12] C. Buche, S. Wallelign and M. Polceanu, "Soybean Plant Disease Identification Using Convolutional Neural Network", 2018.[Accessed 22 April 2020].
[13] Srdjan Sladojevic, Deep neural networks based recognition of plant diseases by leaf image classification, Comput. Intell. Neurosci. (2016) 2016.
[14] Syafiqah Ishak, Leaf disease classification using artificial neural network, J. Teknologi 77 (17) (2015).
[15] Wallelign, Serawork, Mihai Polceanu, and Cedric Buche. "Soybean plant disease identification using convolutional neural network." The thirty-first international flairs conference. 2018.
[16] Malvika Ranjan, Detection and classification of leaf disease using artificial neural network, Int. J. Technical Res. Appl. 3 (3) (2015) 331–333.
[17] S.D. Khirade, A.B. Patil, Plant Disease Detection Using ImageProcessing, 2015 International Conference on Computing CommunicationControl and Automation (2015) 768–771, https://doi.org/10.1109/ICCUBEA.2015.153.
[18] N. Amoura, B. Benaissa, M. Al Ali, S. Khatir, Deep neural network and YUKI algorithm for inner damage characterization based on elastic boundary displacement, in: R. Capozucca, S. Khatir, G. Milani (Eds.), Proceedings of the International Conference of Steel and Composite for Engineering Structures. ICSCES 2022 317, Springer, Cham, 2023. Lecture Notes in Civil Engineering.

[19] A.A. Movassagh, J.A. Alzubi, M. Gheisari, M. Rahimi, S.k. Mohan, A.A. Abbasi, N. Nabipour, Artificial neural networks training algorithm integrating invasive weed optimization with diferential evolutionary model, J. Ambient. Intell. Human. Comput. (2021), https://doi.org/10.1007/s12652-020-02623-6.
[20] Jafar A. Alzubi, Rachna Jain, Preeti Nagrath, Suresh Satapathy, Soham Taneja, Paras Gupta, Deep image captioning using an ensemble of CNN and LSTM based deep neural networks, J. Intell. Fuzzy Syst. (2020), https://doi.org/10.3233/JIFS-189415.
[21] Omar A. Alzubi, Jafar A. Alzubi, Mohammed Alweshah, Issa Qiqieh, Sara Al-Shami, Manikandan Ramachandran, An optimal pruning algorithm of classifier ensembles: dynamic programming approach, Neural. Comput. Appl. (2020).