A Novel CNN Model for Fruit Leaf Disease Detection: A Lightweight Solution for Grapes, Figs, and Oranges

Dalya Anwar[*1]
[1] Department of Computer Science, Salahaddin University, Erbil, Iraq
Emails: dalya.anwar@su.edu.krd

Abstract

Plant diseases are considered a real threat to food security due to the losses incurred by individuals and countries. Early detection is one of the real solutions that can help reduce the size of these losses, but early detection is still bleeding. This study presents the development of a Convolutional Neural Network (CNN) model for classification with a new architecture and optimal performance suitable for real-time applications for the detection of fruit diseases (figs, oranges, grapes). The developed CNN model balanced accuracy and FLOPs using Squeeze-Excitation (SE) and adaptive-average pool layers. After implementing new data developed from Iraqi farms, the CNN model achieved optimal performance compared to the most famous models such as VGG16, ResNet, EfficientNet, and AlexNet.

Keywords: Plant Disease Detection; Convolutional Neural Networks (CNN); Fruit Leaves; Deep Learning in Agriculture; Real-Time Detection

1. Introduction

Plant diseases are among the key factors influencing global food security, as they cause significant harm to agricultural crops and adversely impact both local and global production levels [1, 2]. Symptoms of plant diseases often appear on leaves, stems or roots in the form of spots, erosion or discoloration, etc., which requires early detection methods to prevent major losses.. [3]. Early detection is a strategy for plant diseases and is a step of pivotal importance to prevent outbreaks, as proactive treatment, especially in large scale farms [4,5,6].

Visual inspection is one way to detect plant diseases early, but with humans becomes more challenging [7]. The clear progress in automation, especially with mobile robots and drones [8, 9], makes artificial intelligence in general and computer vision in particular a fertile environment for implementation due to its high ability to extract patterns and accuracy and speed in detection using deep learning, whether it is classification or object detection tasks [10,11,12]. Many deep learning models for plant disease classification have been developed by researchers, including VGG16, ResNet, AlexNet, and EfficientNet [13]. These models have certain drawbacks, especially with their computational requirements. For example, ResNet18 takes over 11 million parameters and more than 3.3 billion floating-point operations (GFLOPs). ResNet34 uses over 21 Million parameters and 6 Billion FLOPs, while ResNet50 requires more than 25 Million parameters and 7 Billion FLOPs [14]. VGG16 is even more demanding, with over 138 Million parameters and 27 Billion FLOPs [15], and AlexNet requires over 61 Million parameters and 1.2 Billion FLOPs [16]. While EfficientNet is more optimized, it still consumes more than 5 Million parameters and 0.7 GFLOPs [17]. The high number of parameters directly affects memory usage, while the GFLOPs directly affect speed and energy efficiency, making these models less ideal for systems with limited resources.

Many deep learning models have presented an acceptable accuracy in classification tasks, but they still bleeding-edge due to their high demand for computational resources, including memory, energy, and execution time. These limitations can significantly affect their overall performance, mostly when deployed in real-world applications with constrained resources such as limited RAM, GPU, or CPU power [18, 19]. The contributions of this study can be summarized as follows:

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

- Developing a new model for a dataset of diseased fruits for three types of fruits (grapes, figs, oranges) suitable for both classification and object detection tasks in addition to segmentation mixed from farms (current reality) and closed studio.
- Developing a CNN model for classification with a new architecture based on achieving the best accuracy with the least possible FLOPs and testing it on the developed fruits dataset.
- Implementing cutting-edge classification models on the developed fruits dataset and comparing them with the developed CNN model.

The second section presents a review of the pertinent literature. In the third section, an in-depth analysis of the architecture of state-of-the-art deep learning models is provided. The fourth section presents the developed dataset and its details, including relevant details. The fifth section highlighted the results, a comparison and discussion of the findings. Finally, the study concludes in the sixth section, which also includes forward-looking perspective directions.

2. Literature Review

Though their availability and scope are still somewhat restricted, researchers have made strides in creating image datasets for the detection of plant diseases. The analysis of those datasets, particularly for the identification of fungal plant diseases, has made extensive use of deep learning models. Using the PlantVillage and PlantDoc datasets, for example, studies comparing different classification models demonstrated the EfficientNet model's capabilities. The EfficientNet model was notable for its fast execution speeds and efficient memory usage, as well as its accuracy of over 98 percent. Due to these characteristics, the EfficientNet model was able to outperform other well-known models, including VGG16, DenseNet121, ResNet151, and ResNet50. [17].

Older models such as VGG16 and AlexNet were tested on a smaller subset of the PlantVillage dataset, which contained approximately 13,000 images representing seven diseases. VGG16 achieved an accuracy of around 97.29%, while AlexNet slightly outperform VGG16 with 97.43% [20]. Although both models delivered high accuracy, they also revealed shortcomings. VGG16, for example, required large computational resources, while AlexNet struggled with generalization and produced insufficient results when applied to more datasets that are complex. Other studies, explored VGG16's performance on larger datasets with a collection of 43,810 images from PlantVillage, the model achieved 44.54% accuracy, while on a smaller subset of 15,915 images, it achieved a much higher accuracy of 95% [21].

In the meantime, machine-learning techniques like Support Vector Machines (SVM) have also been used to classify plant diseases. With a considerably smaller dataset of 1,882 photos from PlantVillage, the SVM model's accuracy was approximately 94%. Despite their computational efficiency, SVM models often perform poorly when handling large datasets or more intricate image patterns, highlighting their shortcomings in comparison to contemporary deep learning models [22].

Object detection, which involves both classification and localization, has been applied to detecting plant diseases by predicting leaf disease such as powdery mildew, leaf mold fungus,blight, and Tomato Mosaic Virus (ToMV). Moreover, the Faster R-CNN model was improved by replacing its original VGG16 backbone with ResNet and integrating a K-means algorithm to enhance the average precision of bounding box localization. These modifications resulted in a 2.71% boost in accuracy, showing the model's performance [23].

Furthermore, the YOLO model, one of the most popular methods for object detection, achieved an acceptable accuracy of 93% when applied to plant disease detection [24]. In another study, researchers compared state-of-the-art object detection techniques with an improved YOLO model, which used the DarkNet53 architecture. This improved YOLO-DarkNet53 model was used to detect Magnaporthe oryzae and detect pathogen spores in microscopic images, achieving an average accuracy of 98.0%. This highlights the strong performance of the model in predicting and classifying plant pathogens [25]. Table 1. Presents the summary of well-known deep learning classification models.

Table 1: Summary of state-of-the-art classification models.

Model	Accuracy	Computational Cost	Memory Use
ResNet18	Moderate	Moderate	Moderate
ResNet50	High	High	High
ResNet150	Very High	Very High	High
VGG16	Moderate	High	High

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

AlexNet	Low-Moderate	Moderate	Moderate
EfficientNet	Very High	Low	Low

3. Deep Learning

CNNs serve as a fundamental framework for deep learning-based classification models. Although their specific implementation may vary, CNNs have consistently shown remarkable performance in extracting complex patterns, especially in computer vision.

EfficientNet is a neural network architecture that is perfect for applications that need low computational costs because it strikes a balance between accuracy and efficiency. It employs a compound expansion technique that uniformly modifies the network's depth (layers), width (channels), and resolution to enhance performance and use fewer resources. Mobile Inverted Bottleneck (MBConv) blocks come after a (3 × 3) convolutional layer that extracts important features at the beginning of the architecture. These blocks maintain performance while lowering computational complexity by using independent depth-level convolutions. A projection phase lowers dimensionality, depth-level convolutions process spatially, and an expansion phase boosts representation capacity in each block. The purpose of EfficientNet blocks is to record both local and global features. In order to extract intricate patterns, deeper blocks use larger kernels (5 × 5), whereas early blocks use smaller kernels (3 × 3) for fine detail. By recalibrating channel importance, squeeze-excitation (SE) layers in each block enhance focus on key features. By guaranteeing a smooth gradient flow, residual connections also help with training. Low parameters and FLOPs are maintained while achieving state-of-the-art accuracy on benchmarks such as ImageNet thanks to the architecture's intricate scaling strategy [26]. Figure 1. Illustrates the EfficientNet architecture and their hyperparameter.

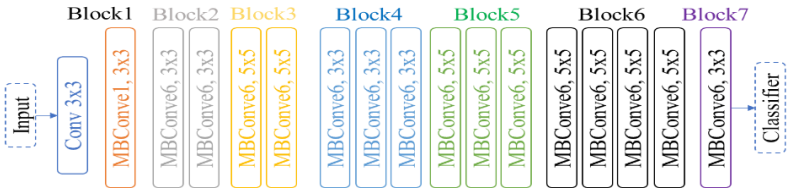

Figure 1. EfficientNet architecture model.

The architectures of two well-known CNNs, AlexNet (Figure 2b) and VGG16 (Figure 2a), are depicted in Figure 2. A ReLU activation follows each cascaded layer of 3x3 tiny convolutional filters, which form the foundation of VGG16's deep, homogeneous architecture. These layers preserve a recurring pattern throughout the network while extracting features. To decrease the spatial dimensions of the feature maps without sacrificing significant information, MaxPooling layers are positioned in between the convolutional blocks. The feature map at the network's end is compressed using an Adaptive Average Pooling layer after being shrunk to [1,512,7,7]. For classification, the output is subsequently flattened and sent to fully connected layers. The architecture of VGG16 is robust and dependable because it captures hierarchical features with tiny filters in every layer. As an early stage of CNN development, AlexNet (Figure 2b) has a simpler architecture than VGG16. Larger convolutional kernels (11 × 11 and 5 × 5) and fewer layers are used in the first layers to extract features from the 300 × 300 × 3 input image. While the feature depth increases and the spatial dimensions decrease as the data moves through the network, MaxPooling layers aid in reducing the size of the data. In the final step, a fixed-size feature map is created using an Adaptive Average Pooling layer and fed into fully connected layers for classification. during AlexNet's debut [27, 28].

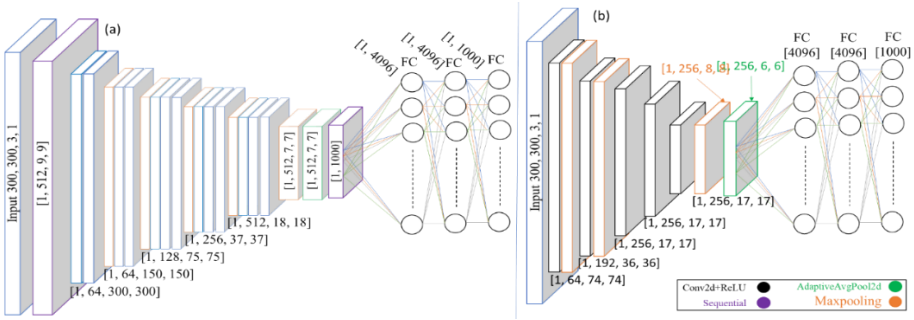

Figure 2. Models' architecture: a) VGG16. b) AlexNet.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

The structural design and distinctions between these two popular network configurations are illustrated through the architecture of ResNet18 and ResNet50. An input layer processing 300 × 300 × 3 images is the first step in ResNet18. Batch normalization and ReLU activation functions are then combined with a sequence of convolutional layers. In order to facilitate smoother gradient flow and mitigate the vanishing gradient issue during training, max-pooling layers progressively reduce the spatial dimensions while the remaining connections avoid the two-layer blocks. A fixed-size feature vector is generated by the network's adaptive average pooling layer at the end, and it is then fed into a fully connected layer for classification. Bottleneck blocks are used to create the 50-layer deeper architecture that ResNet50 depends on. Each bottleneck block has three 1 × 1, 3 × 3, and 1 × 1 convolutional layers that are intended to increase the network's capacity for feature extraction while also increasing computational efficiency. Four phases of bottleneck blocks make up the architecture, each of which reduces the spatial dimensions and adds more feature maps. ResNet50 concludes with a fully connected layer for final classification, just like ResNet18, after an adaptive mean pooling layer. ResNet50 is made for more complex, computationally demanding applications, whereas ResNet18 is better suited for simpler tasks, as evidenced by the difference in depth and block structure [29]. These designs are visually depicted in Figure 3(a) for ResNet18 and Figure 3(b) for ResNet50.

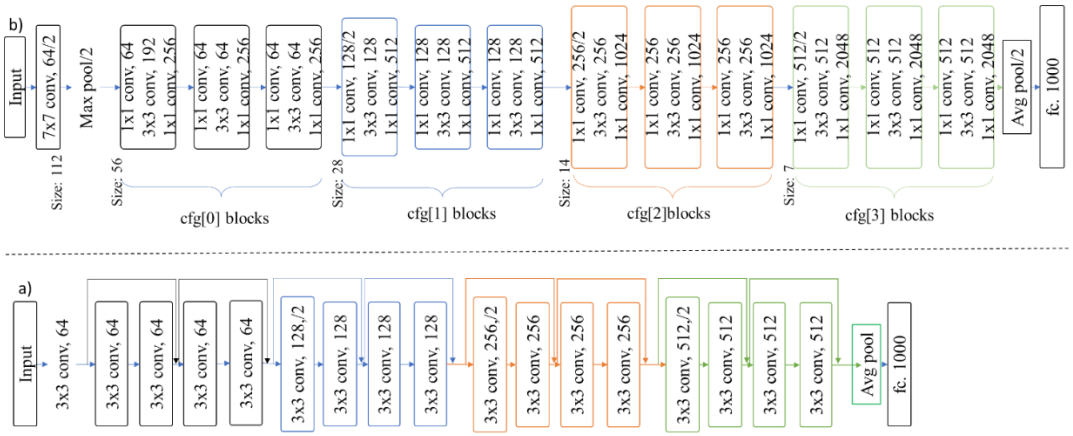

Figure 3. ResNet Model architecture: a) ResNet18. b) ResNet50.

The suggested CNN architecture is especially made to tackle the difficulties of classifying plant diseases while consuming the least amount of memory, computing power, and computational expense. The model incorporates a number of essential elements to achieve this, which streamline feature extraction and boost efficiency. It begins by extracting key spatial features from the input image using a 3x3 convolutional layer. Batch normalization, which normalizes the input distributions to stabilize the learning process, and a ReLU activation function, which adds nonlinearity to the network so it can learn intricate patterns at low computational cost, come next. Separate depth convolution units are used in place of conventional convolutions, which significantly increases the model's efficiency. The convolution process is divided into two stages by these units: point convolution (1×1), which aggregates data from multiple channels, and depth convolution, which extracts spatial features independently from each channel. Without sacrificing accuracy, the computational complexity is significantly decreased by separating these operations, leading to fewer parameters and FLOPs..

SE modules are added to help the model better focus on key elements. These modules assist the network in giving the most instructive channels priority. Initially, the feature map statistics are summarized during the compression phase by combining spatial data using global mean pooling. A tiny neural network that learns the significance of each channel during the excitation phase then recalculates the feature maps. This makes predictions more accurate by enabling the model to highlight pertinent features while suppressing less instructive ones. In addition, the architecture incorporates global context attention blocks, which are crucial for comprehending connections between various input image regions. These building blocks allow the model to recognize subtle patterns that are frequently present in many plant leaves by capturing long-range dependencies and extracting global contextual information. For the detection of plant diseases, where symptoms may manifest in small, dispersed areas, this is especially crucial.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

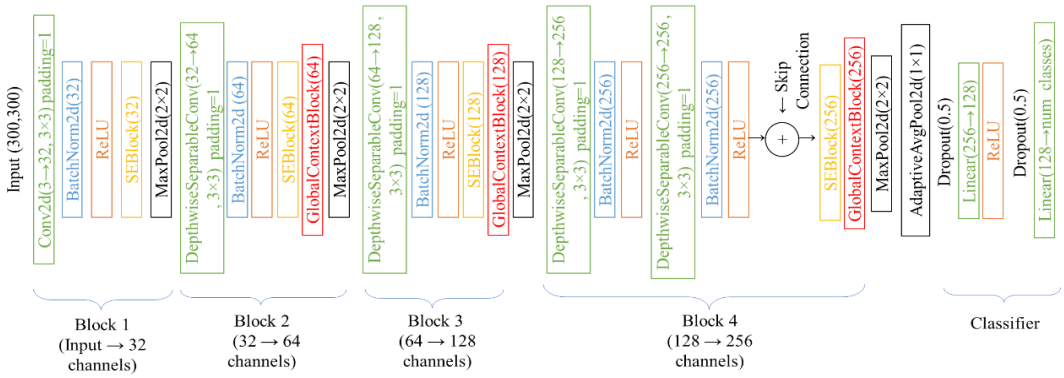

Figure 4. Proposed CNN architecture Model.

To enhance the information flow between layers, the model also uses skip connections. In order to ensure smoother gradient propagation during training, these connections aid in resolving the vanishing gradient issue. Additionally, they speed up the training process by making it simpler for the model to learn both shallow and deep features. The architecture increases the number of channels while decreasing the feature map dimensions to maximize computational resources. This method makes sure that global feature extraction in deeper layers and local feature extraction in earlier layers are balanced. An OneCycleLR learning rate scheduler is utilized for efficient training. Over 50 epochs, the learning rate peaks at a maximum learning rate of 0.01 after rising and then falling in a symmetric curve. This method improves convergence and helps the model avoid being trapped in local minima. A number of regularization strategies are used to lessen overfitting and enhance generalization. In order to prevent overfitting, dropout randomly disables neurons, batch normalization controls activation distributions, and data augmentation adds variations to the training data such as flips, rotations, and scaling—to strengthen the model against unknown input variations. Figure 4. demonstrating the CNN architecture that has been optimized for the classification of plant diseases.

4. Fruit dataset

A crucial phase in the development of AI is the dataset preparation procedure, which calls for exacting attention to detail. Due to the dearth of publicly available data on agricultural crops, information was gathered directly from Iraqi orchards, concentrating on fruits like oranges, figs, and grapes. To improve the dataset's quantity and quality, this collection was carried out in phases. Figure 5. Depicts the Iraqi fruit dataset samples. In the initial stage, all of the images were taken from public orchards; there were 100 photos in each category, for 300 photos. A thorough filtering procedure was applied to these photos. Nevertheless, early training attempts with this dataset showed that some classification models had instabilities, and some models were overfitting. In order to solve this, the dataset was expanded in the second phase to include 150 images in each category, for 450 images. The models showed difficulties in generalization, suggesting limited adaptability to unseen data, even though this larger dataset stabilized the training processes.

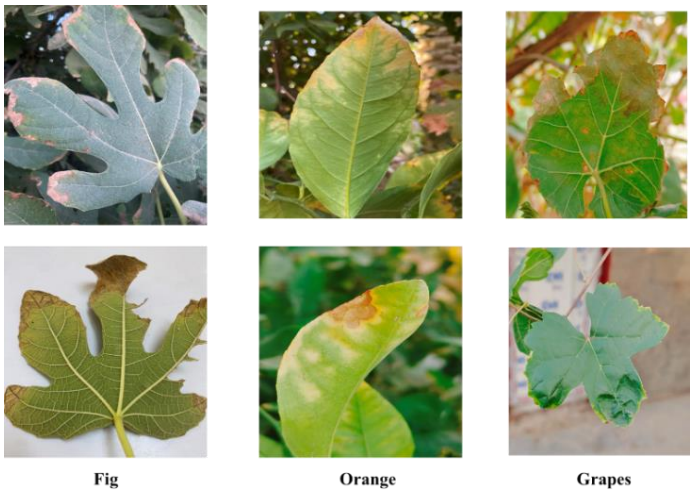

Figure 5. Iraqi Fruit disease dataset.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

In order to enhance performance even more, the third stage added more photos taken in a controlled studio setting, bringing the total to 565. Even though the results were better because of this expansion, the models' instability and fluctuations persisted throughout training. The fourth stage involved the use of data augmentation techniques, such as 90-degree image rotation and saturation adjustments to improve clarity by 10%. The dataset was significantly increased by these methods, yielding 1,380 images in total. Model performance significantly improved as a result of the larger and more varied dataset, as the models demonstrated increased stability and robustness throughout training.

5. Experimental Results

All models in this study were trained with identical hyperparameters to guarantee a fair comparison. Eighty percent of the dataset was used for traditional training, and twenty percent was used for validation. Given the variations in their architectural depth and complexity, the training procedures for ResNet18, ResNet34, and ResNet50 exhibit acceptable learning behaviors. ResNet18 exhibits consistent learning without obvious overfitting, as evidenced by the training and validation loss curves' steady decline over all 100 epochs. Interestingly, with the aid of regularization strategies like dropout or data augmentation, the validation loss consistently stays lower than the training loss, suggesting that the model performs well when applied to unseen data. The model's strong generalization capabilities are further supported by the corresponding accuracy curves, which demonstrate a gradual improvement with the validation accuracy marginally surpassing the training accuracy. ResNet34's training process, on the other hand, shows a quicker initial drop in loss during early epochs, indicating that the deeper network can advance more quickly. By the 60th epoch, both the training and validation losses have stabilized at values near zero, indicating successful model learning. With training and validation accuracies convergent at approximately 96 percent, the accuracy curves demonstrate nearly flawless performance. The benefits of the ResNet34 architecture's deeper depth are highlighted by this behavior.

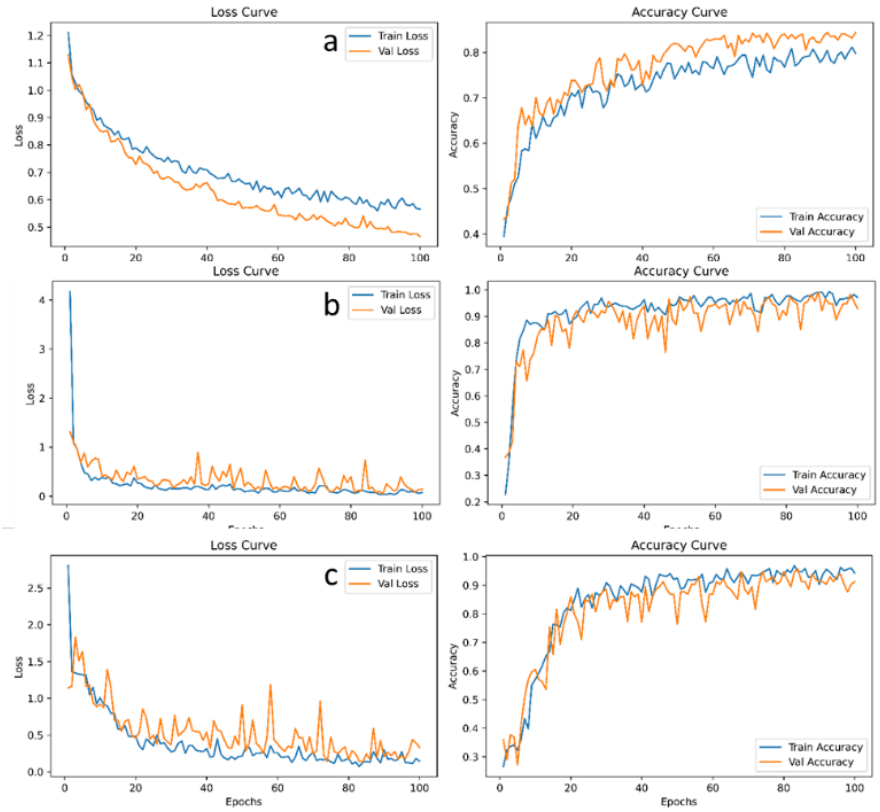

Figure 6. Training process for fruits disease classification: a) ResNet18. b) ResNet34. c) ResNet50.

Significant variations between VGG16, AlexNet, and EfficientNet's training runs highlight their distinct architecture and capacities for learning. A significant initial drop in loss is seen in the training curves for VGG16, indicating successful learning in the early stages. However, during the middle training phases (epochs 50–70), validation loss spikes are visible, suggesting unstable times and a propensity for overfitting. By the last epochs, the training indicates a certain level of stability and validation losses converge to smaller values. The model learns well, according to the accuracy curves for VGG16, but the validation accuracy varies greatly, suggesting uneven generalization throughout the dataset. AlexNet exhibits more consistent training behavior in contrast. Indicating a

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

more balanced, albeit less accurate, learning process, the initial loss drop is quick, but stabilization happens sooner with fewer validation losses. With training and validation accuracy moving in close proximity to one another, its accuracy curves exhibit a consistent upward trend. In contrast, EfficientNet's training curves are more consistent and effective than those of the other models. With little difference between training and validation losses, its loss curves rapidly converge in the first few epochs, suggesting strong generalization. Accuracy also rapidly increases, reaching high values at an early stage and preserving minimal variance in the agreement between the training and validation curves. Illustration 7. depicts.

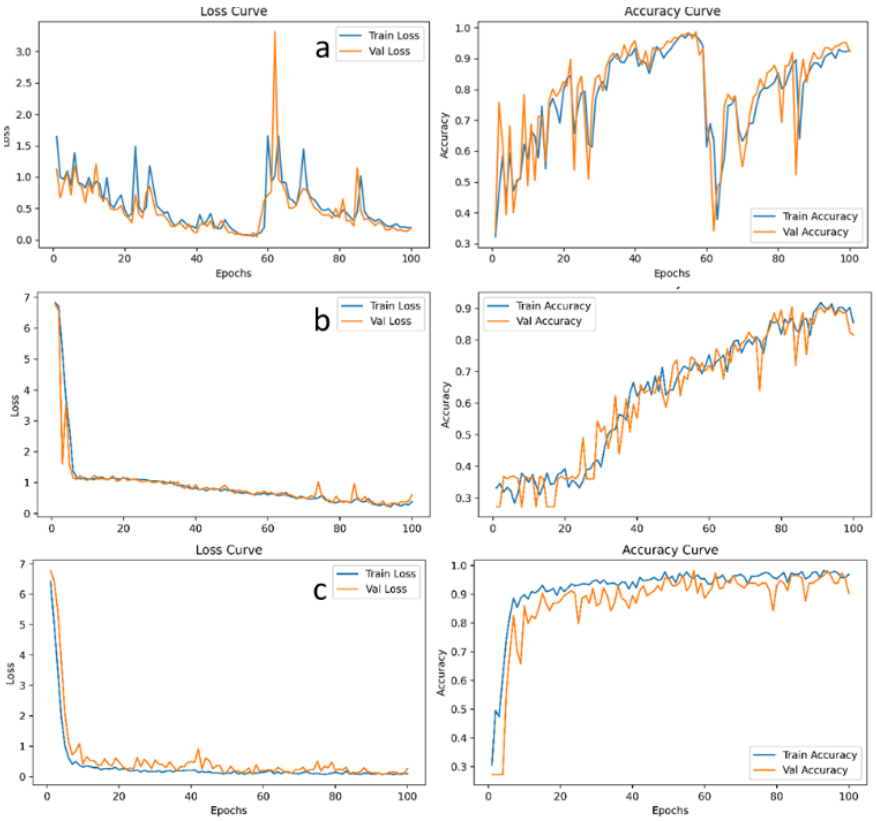

Figure 7. Training process for fruits disease classification: a) VGG16. b) AlexNet. c) EfficientNet.

The suggested CNN's training procedure for classifying fruit diseases, as illustrated in Figure 8, displays the evolution of loss, accuracy, and learning rate over 50 epochs. The loss curve shows that both training and validation losses significantly decreased over time, demonstrating the model's capacity for efficient learning. Due to noise in the validation data, the validation loss varies in the early epochs but stabilizes as training goes on. Both training and validation accuracies reach high levels as the epochs go on, and the accuracy curve similarly shows a quick improvement during the early epochs. The strong agreement between the validation and training accuracies suggests that the model performs well when applied to new data. Plotting the learning rate shows a dynamic schedule, with the learning rate beginning relatively high to enable faster convergence at first and then progressively decreasing to enhance the model afterwards. This method guarantees successful model convergence and aids in learning process optimization.

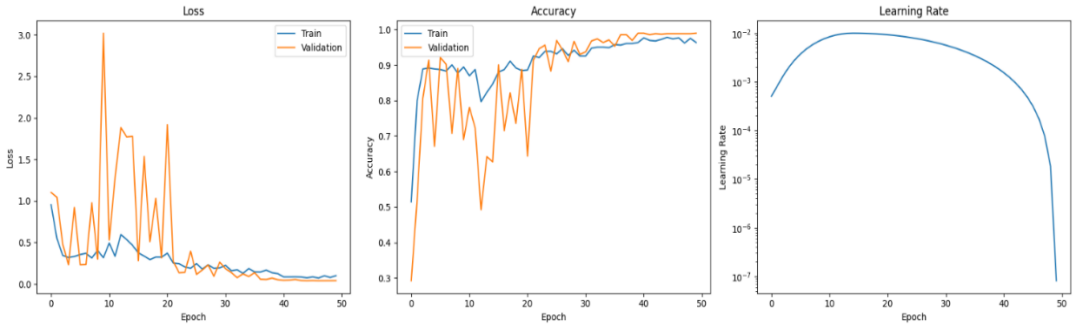

Figure 8. Proposed CNN model for fruit disease classification.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

The findings offer a thorough comparison of a number of deep learning models in terms of important performance indicators like accuracy, precision, recall, F1-score, FLOPs, and total parameters. With an F1-score of 0.94 and a precision of 0.93, ResNet50 demonstrated excellent performance. The computational demand was significantly higher, though, requiring over 25-point 5 million parameters and 7-point 6 billion FLOPs. Similarly, ResNet34 reduced the computational requirements to 6.8 billion FLOPs and 21.8 million parameters while achieving slightly better performance with a precision of 0.96 and an F1-score of 0.96. ResNet18 demonstrated a notable decline in performance with a precision of 0.84 and an F1-score of 0.83, despite being lighter at 3.4 billion FLOPs and 11.7 million parameters, suggesting its limited capacity to manage challenging classification tasks.

VGG16 had a moderate F1 score of 0.90 and accuracy of 0.92, despite having a high recall of 0.99 and a variance in precision of 0.82. With over 138 million parameters and 27-point 2 billion FLOPs, it also displayed the highest computational demand of all the models, which might restrict its use in settings with limited resources. Because of its antiquated architecture, AlexNet had a lower-than-expected precision of 0.83 and an F1 score of 0.83, despite being lightweight with 1.2 billion FLOPs. With only 746 million FLOPs and 5.3 million parameters needed to achieve a precision of 0.92 and an F1 score of 0.92, EfficientNet_b0 provided a fair compromise between computational efficiency and performance.

Our proposed model achieved outstanding performance in precision, recall, F1 score, and accuracy of 0.99, indicating near-perfect classification ability. Moreover, it is highly efficient in terms of computational complexity, with only 335 million FLOPs and 172.3 thousand parameters, making it lightweight and efficient as in Table 2.

Table 2: Classification models results summary.

Models	Precision	Recall	F1-Score	Accuracy	FLOPs	Total params
ResNET50	0.95	0.94	0.94	0.93	7,647,974,656	25,557,032
Resnet34	0.96	0.96	0.96	0.96	6,843,157,120	21,797,672
ResNET18	0.85	0.83	0.83	0.84	3,399,159,296	11,689,512
VGG16	0.82	0.99	0.90	0.92	27,235,905,024	138,357,544
AlexNet	0.87	0.83	0.83	0.83	1,201,895,168	61,100,840
EfficientNet_b0	0.93	0.93	0.92	0.92	746,469,072	5,288,548
Our	*0.99*	*0.99*	*0.99*	*0.99*	*335,157,314*	*172.314*

6. Conclusion

This study used a new Iraqi dataset to create a CNN model with a novel architecture for the classification of diseased fruits. With 172,314 parameters and FLOPs of 335,157,314 and a 99 percent accuracy performance, the optimized CNN used less memory. The CNN model performed better than other well-known classification models, which makes it appropriate for the typical computational cost's limited resources. As a future path, the research focuses on creating an object detection model. For instance, it integrates the CNN model that was developed as the foundation for one of the object detection models and applies it in real time in extensive agricultural settings.

Contribution Statement

Dalya Anwar was solely responsible for all aspects of this work, including the conceptualization and design of the study, preparation of experimental methods, data collection, and analysis. The author also conducted the writing of the original draft, reviewed and revised the manuscript, and prepared all figures and tables. Furthermore, Dalya Anwar provided supervision and ensured the integrity of the study from its inception to the final submission.

Conflict of Interest: The authors declare that they have no conflict of interest.

Reference

[1] D. Li, F. Ahmed, N. Wu, and A. I. Sethi, "Yolo-JD: A deep learning network for jute diseases and pests detection from images," *Plants*, vol. 11, no. 7, p. 937, 2022.

[2] Ramanjot *et al.*, "Plant disease detection and classification: A systematic literature review," *Sensors*, vol. 23, no. 10, p. 4769, 2023.

[3] M. K. Agbulos, Y. Sarmiento, and J. Villaverde, "Identification of leaf blast and brown spot diseases on rice leaf with YOLO algorithm," in *Proc. IEEE 7th Int. Conf. Control Sci. Syst. Eng. (ICCSSE)*, 2021, pp. 307–312.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

[4] L. Goel and J. Nagpal, "A systematic review of recent machine learning techniques for plant disease identification and classification," *IETE Tech. Rev.*, vol. 40, no. 3, pp. 423–439, 2023.

[5] F. Martinelli *et al.*, "Advanced methods of plant disease detection: A review," *Agron. Sustain. Dev.*, vol. 35, pp. 1–25, 2015.

[6] V. Singh, N. Sharma, and S. Singh, "A review of imaging techniques for plant disease detection," *Artif. Intell. Agric.*, vol. 4, pp. 229–242, 2020.

[7] C. H. Bock, G. H. Poole, P. E. Parker, and T. R. Gottwald, "Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging," *CRC Crit. Rev. Plant Sci.*, vol. 29, no. 2, pp. 59–107, 2010.

[8] H. Slimani, J. El Mhamdi, and A. Jilbab, "Drone-assisted plant disease identification using artificial intelligence: A critical review," *Int. J. Comput. Digit. Syst.*, vol. 14, no. 1, pp. 10433–10446, 2023.

[9] Y. Ampatzidis, L. De Bellis, and A. Luvisi, "iPathology: Robotic applications and management of plants and plant diseases," *Sustainability*, vol. 9, no. 6, p. 1010, 2017.

[10] A. Bhargava *et al.*, "Plant leaf disease detection, classification and diagnosis using computer vision and artificial intelligence: A review," *IEEE Access*, 2024.

[11] S. S. Chouhan, U. P. Singh, and S. Jain, "Applications of computer vision in plant pathology: A survey," *Arch. Comput. Methods Eng.*, vol. 27, no. 2, pp. 611–632, 2020.

[12] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Comput. Electron. Agric.*, vol. 153, pp. 69–81, 2018.

[13] S. Saleki and J. Tahmoresnezhad, "Agry: A comprehensive framework for plant diseases classification via pretrained EfficientNet and convolutional neural networks for precision agriculture," *Multimed. Tools Appl.*, pp. 1–39, 2024.

[14] R. Chen, H. Qi, Y. Liang, and M. Yang, "Identification of plant leaf diseases by deep learning based on channel attention and channel pruning," *Front. Plant Sci.*, vol. 13, p. 1023515, 2022.

[15] D. Zhu, Q. Feng, J. Zhang, and W. Yang, "Cotton disease identification method based on pruning," *Front. Plant Sci.*, vol. 13, p. 1038791, 2022.

[16] R. Wang *et al.*, "Deep neural network compression for plant disease recognition," *Symmetry (Basel).*, vol. 13, no. 10, p. 1769, 2021.

[17] F. Rajeena P. P., A. S. U., M. A. Moustafa, and M. A. S. Ali, "Detecting plant disease in corn leaf using EfficientNet architecture—An analytical approach," *Electronics*, vol. 12, no. 8, p. 1938, 2023.

[18] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Compute and energy consumption trends in deep learning inference," *arXiv Prepr. arXiv2109.05472*, 2021.

[19] O. N. Neamah, T. A. Almohamad, and R. Bayir, "Enhancing road safety: Real-time distracted driver detection using Nvidia Jetson Nano and YOLOv8," in *Proc. Zooming Innov. Consumer Technol. Conf. (ZINC)*, 2024, pp. 194–198.

[20] S. B. Jadhav, "Convolutional neural networks for leaf image-based plant disease classification," *IAES Int. J. Artif. Intell.*, vol. 8, no. 4, p. 328, 2019.

[21] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, "New perspectives on plant disease characterization based on deep learning," *Comput. Electron. Agric.*, vol. 170, p. 105220, 2020.

[22] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, 2017, pp. 1–4.

[23] Y. Zhang, C. Song, and D. Zhang, "Deep learning-based object detection improvement for tomato disease," *IEEE Access*, vol. 8, pp. 56607–56614, 2020.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025

[24] M. S. Maqbool, R. Nazeer, A. Basit, and K. Zahra, "Automated detection and localization of fungal infections on cotton leaves using YOLO-based object detection model," *Mach. Algorithms*, vol. 2, no. 2, pp. 121–136, 2023.

[25] H. Zhou *et al.*, "Automatic detection of rice blast fungus spores by deep learning-based object detection: Models, benchmarks and quantitative analysis," *Agriculture*, vol. 14, no. 2, p. 290, 2024.

[26] V.-T. Hoang and K.-H. Jo, "Practical analysis on architecture of EfficientNet," in *Proc. 14th Int. Conf. Human Syst. Interact. (HSI)*, 2021, pp. 1–4.

[27] T. Shanthi and R. S. Sabeenian, "Modified AlexNet architecture for classification of diabetic retinopathy images," *Comput. Electr. Eng.*, vol. 76, pp. 56–64, 2019.

[28] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification," in *Proc. Int. Conf. Disruptive Technol. Multi-Disciplinary Res. Appl. (CENTCON)*, 2021, pp. 96–99.

[29] S. Targ, D. Almeida, and K. Lyman, "ResNet in ResNet: Generalizing residual architectures," *arXiv Prepr. arXiv1603.08029*, 2016.

DOI: https://doi.org/10.54216/FPA.190220
Received: January 05, 2025 Revised: February 04, 2025 Accepted: March 01, 2025