Classification of bitter gourd leaf disease using deep learning architecture: ResNet50

Artika [a,1,*], Wikky Fawwaz Al Maki [b,2]

[a] Telkom University, JL. Telekomunikasi No. 1, Bandung 40257 , Indonesia
[1] artikaq@student.telkomuniversity.ac.id; [2] wikkyfawwaz@telkomuniversity.ac.id
* corresponding author

ARTICLE INFO

Article history
Received September 13, 2022
Revised January 2, 2024
Accepted December 30, 2024
Available online May 31, 2025

Selected paper from The 2022 5th International Symposium on Advanced Intelligent Informatics (SAIN'22), Yogyakarta (Virtually), September 14, 2022, http://sain.ijain.org/2022/. Peer-reviewed by SAIN'22 Scientific Committee and Editorial Team of IJAIN journal.

Keywords
Bitter gourd
Deep learning
Image processing
ResNet50
ResNet101

ABSTRACT

The primary goal of this research is to develop a feasible and efficient method for identifying the disease and to advocate for an appropriate system that provides an early and cost-effective solution to this problem. Due to their superior computational capabilities and accuracy, computer vision and machine learning methods and techniques have garnered significant attention in recent years for classifying various leaf diseases. As a result, Resnet50 and Resnet101 were proposed in this study for the classification of bitter gourd disease. The 2490 images of bitter gourd leaves are classified into three categories: Healthy leaf, Fusarium Wilt leaf, and Yellow Mosaic leaf. The proposed ResNet50 architecture accomplished 98% accuracy with the Adam optimizer. The ResNet101 architecture achieves an average accuracy of 94% with the Adam optimizer. As a result, the proposed model can differentiate between healthy and diseased bitter gourd leaves. This research contributes to the development of methods for detecting bitter melon leaf disease using computer vision and machine learning, achieving high accuracy and supporting automatic disease diagnosis. The results can help farmers quickly and cost-effectively detect diseases early, thereby increasing agricultural productivity.

1. Introduction

One of the most significant problems in agriculture is the presence of diseased plants. Unfortunately, plant diseases are difficult to detect in modern times, making it infeasible to do so for large quantities of crops or in the field [1]. However, most plant diseases exhibit outward symptoms, and the standard method for diagnosis involves an experienced plant pathologist visually inspecting sick plant leaves [2]. Bitter gourd is one type of sample that is high in phytochemicals and can be used as a dietary supplement to combat various diseases [3]. Bitter gourd is well-known for its medicinal properties in Asian countries [4]. However, leaf disease reduces production quality and/or yield. Bitter gourd leaf diseases exhibit a variety of symptoms, including yellow mosaic disease, which causes the leaves to turn dark green, yellow, or brown. In contrast, fusarium wilt disease is characterized by the yellowing of the leaves starting from the oldest leaves. Plant diseases are frequently identified through traditional methods, such as visual inspection. However, these techniques have several drawbacks. They are costly, as they require ongoing professional supervision and take time, since experts are not always available locally. Therefore, it is crucial to accurately and promptly identify bitter gourd leaf disease.

In recent years, significant advances in object classification have been made using various methods and multiple deep learning architectures. Nowadays, numerous plant diseases are classified and detected using this structure. Recent advances in AI have highlighted and accelerated the use of various types of basic AI technologies in agriculture, independent agents (devices) that perform computer vision, and interactions with the environment [5]. Deep learning is among the most suitable techniques for identification. Machine Learning (ML) models improve accuracy by programming large amounts of data to extract features and utilizing multiple hidden layers [6]. Convolutional neural networks (CNNs) are a fundamental deep learning technique. This technique is a popular and widely used deep learning algorithm for image processing. CNN has proposed a class of models that can be used to better understand the contents of an image, providing new opportunities to improve image identification, segmentation, detection, and retrieval [7]. As evidenced by the advanced techniques mentioned above, recent advances in computer vision and machine learning, particularly [8]. The use of deep learning in agriculture is where research on plant leaf disease diagnosis is primarily focused. Techniques that balance accuracy and efficiency in the identification of plant leaf diseases, on the other hand, are rarely used [9].

Numerous papers have been published that use machine learning algorithms to classify and identify plant leaf diseases. These experiments included mango [10], apple [11], tomato [9], citrus [6], pomelo [12], paddy [13], and other leaf varieties in a previous study [14]. They proposed using R-CNN with the VGG16 architecture to detect the leaves of the bitter gourd. The average accuracy rates of healthy leaves, powdery mildew, gray spot, grape blight, and gray spot were 89%, 83%, 81,9%, and 79,5%. In another study by Hasan et al., They proposed a system for detecting diseased or defective bitter gourd. Deep Learning (DL) was used to achieve the goal [4]. This study proposed a deep learning method for identifying bitter gourd leaves (health, yellow mosaic, and fusarium wilt). For our proposed technique, we used the ResNet architecture (ResNet50 and ResNet101). ResNet has been demonstrated to outperform very deep plain networks and to make training easier. Optimization has received a lot of attention in several research areas because it is such an important part of machine learning tasks [15]. Therefore, to improve the performance of the best-obtained model, we used deep learning optimizers such as Adam, Adamax, SGD, Nadam, Adadelta, Adagrad, RMSprop, and Ftrl. The objective of this model was to improve performance in the classification of bitter gourd leaves while minimizing the impact of the disease as much as possible. This paper's main contributions were as follows: to prove that that certain residual networks (ResNet50 and ResNet101) can be used to classify plant diseases. The methods applied to classify the type of disease. Helping a farmer to identify bitter gourd leaf disease. The novelty in our study lies in using the approach: CNN ResNet50 and ResNet101 to solve multi-class classification for three types of disease. In a previous study [14]. The research only obtained an accuracy of under 90%. We used an imbalanced dataset, when data is imbalanced, the proposed frameworks aim to improve model accuracy.

The rest of this paper was organized as follows: Section 2 describes the method of modelling a classification algorithm, Section 3 provides information about the study's results, including accuracy results, and Section 4 concludes.

2. Method

In this study, the author proposed to classify healthy and diseased leaves of bitter gourd. The stages of building a classification model using machine learning methods are as follows: dataset search, preprocessing the dataset, data split, feature extraction, classification, and performance measurements. The algorithm for the classification of bitter gourd leaves using a deep learning architecture is shown in Fig. 1.

Fig. 1. shows the general algorithm used in the process of training a classification model using ResNet. The input stage starts with importing the library and dataset, followed by processing the dataset and dividing it into training data and test data. Next, the ResNet model is trained with repetitions based on the number of epochs and batch size. After the training is complete, the model is used to make predictions and calculate the confusion matrix as a performance evaluation. The final output of this process is a classification report that shows the results of accuracy, precision, recall, and other evaluation metrics.

```
Input:
    Import library
    Import dataset
    Processing dataset
    Split dataset into train and test sets
    For Epoch and batch_size do
        Train the ResNet
    End for
    Make prediction
    Calculate the confusion matrix
Output:
Classification Report
```

Fig. 1. Algorithm of the classifier using a deep learning architecture

Fig. 2 illustrates the flow of the image-based leaf disease classification process. The process begins with importing the dataset, followed by image processing to enhance image quality. Next, the dataset is divided into two parts, namely training data (80%) and testing data (20%). After that, feature extraction is performed to obtain the important features of the image. The features obtained from the training data are used to train the classification model, while the testing data is used to test the accuracy of the resulting model. The process concludes with a classification stage that determines the leaf category based on the extracted features.

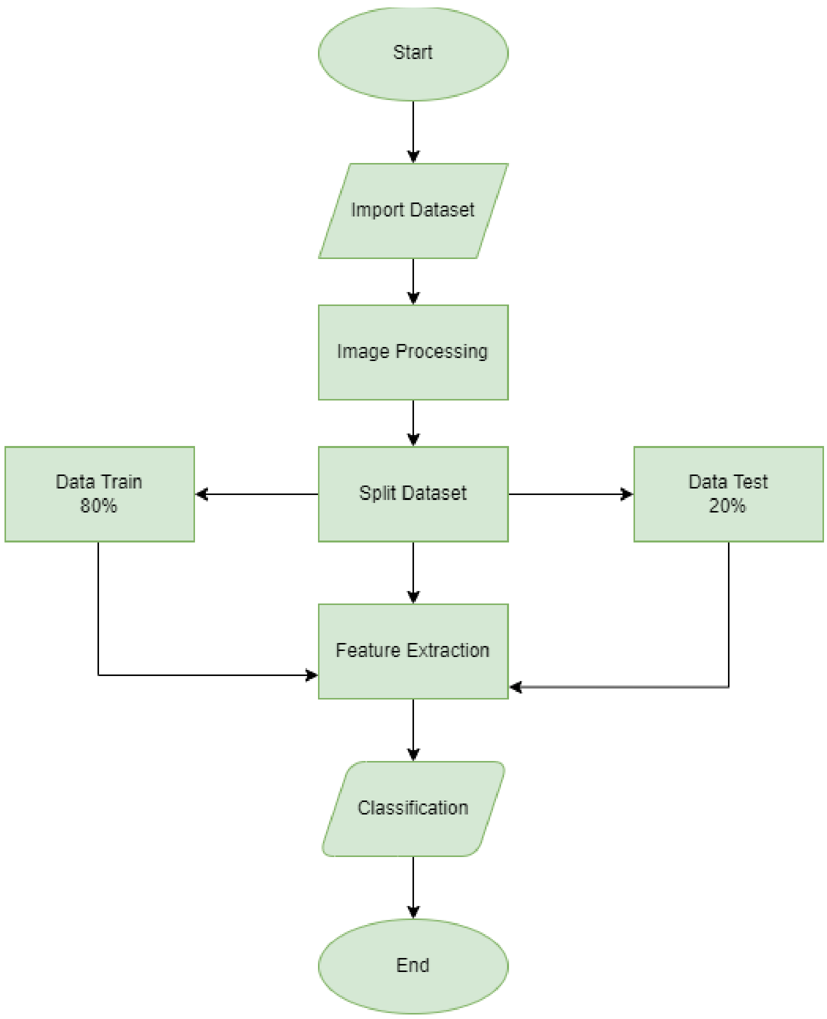


Fig. 2. Flowchart of a classification model

2.1. Image Acquisition

It is the highest level of image processing [1]. Image acquisition is the earliest step in the process of taking a picture or scanning an analog image to obtain a digital image. Camera Resolution, minimum camera resolution 12 MP, lighting technique using a box and installing lights on the ceiling of the box as lighting, zooming, the process of taking pictures should not be too close so that the picture can be seen, the angle of shooting, the image must be adjusted so that it can be seen clearly. The entire dataset was captured using a smartphone camera (Vivo V15 Pro, Oppo Reno2, and Pocco M3) at various angles and resolutions.

2.2. Dataset

The dataset used in this study was a personal dataset captured with three different smartphone cameras. The dataset consisted of three classes: Fusarium Wilt, Yellow Mosaic, and Healthy Leaf. A bipartite Geminivirus causes yellow mosaic disease with a circular topology and a unique single-stranded DNA genome, belonging to the genus Begomovirus and family Geminiviridae. This virus family can easily manipulate their genome, allowing it to infect a wide variety of dicotyledonous plants [16]. Fusarium wilt is a vascular wilt disease caused by Fusarium oxysporum f. sp. Momordica, a soil-borne plant pathogen (FOM) [17]. An agricultural engineer with expertise in this sector assisted in grouping the collected data into three different classes. Table 1 shows the number of each class.

Table 1. Number of datasets

Number Dataset of Bitter Gourd	
Class	*Numbers*
Health	800
Fusarium Wilt	779
Yellow Mosaic	830
Total	2490

All images were colored with RGB to identify the disease. In this step, we split the dataset into two, 80% for training and 20% for testing. The picture can be shown in Fig. 3.

(a) (b) (c)

Fig. 3. (a) Healthy, (b) Fusarium Wilt, (c) Yellow Mosaic

2.3. Preprocessing

The first step in this study was pre-processing after the dataset was collected. Preprocessing is a technique that removes noise from images by enhancing image quality [18]. The images in the dataset for the deep CNNs classification algorithm were also preprocessed before the model was trained to enhance the extraction of features and consistency. Normalization of image size and format is one of the most important operations [19]. In this step, we resized the image. In this research, all images were converted to 128 x 128 pixels [20]. In this issue, the second step involves splitting the dataset into training data and testing data.

2.4. Augmentation

The imbalanced datasets and short sample sizes were the leading causes of poor recognition performance in deep learning. Expanding the amount of data is, therefore, important for a deep learning model to classify leaf diseases [21]. Data augmentation is a necessary step to obtain a sufficiently diverse sample for studying deep tissue from images. Several studies have investigated the role of data augmentation in deep learning [18]. The augmentation process can improve the accuracy of the CNN model. Because the augmentation of the model gets additional image data that can be used to create a model, the data is better generalized. Augmentation performed in this step involved shearing the image by 10%, shifting the image by 10%, and rotating the image by 30 degrees. The picture is shown in Fig. 4. After the augmentation process, the dataset for each class was expanded.

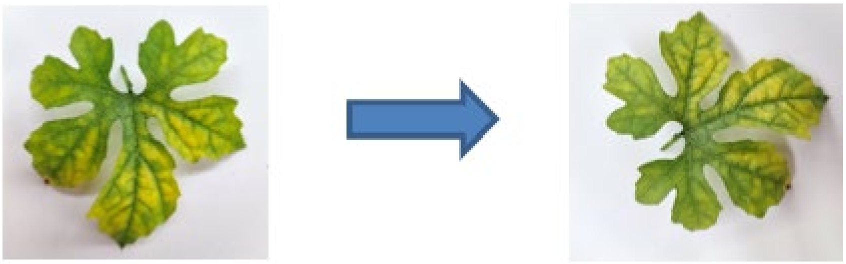


Fig. 4. Result of Augmentation

2.5. Modeling

CNN architectures first showed up in 1998 with LeNET-5 followed by AlexNet in 2012. Later, various other CNN architectures were developed, introducing successful solutions to object problems. The problems are related to the classification and object detection. There are numerous convolutional operations in these architectures. Activation functions of different types and the practice of "operations pooling" were both put to the test. The ImageNet Challenge, also known as the ImageNet Large Scale Competition for Visual Recognition (LLSVRC), was used to categorize millions of training photos. Consequently, it follows that training a new CNN architecture must be a time-consuming and costly process [22]. Convolutional Neural Network (CNN) is the most useful and successful deep learning method for resolving image classification issues. CNNs, or deep neural networks, are recognized to be excellent for computer vision applications, and they may classify diseases more accurately than traditional methods. As a result, it can be applied to images, classification, object detection, and clustering [15]. A Convolutional Neural Network was created to perform the image classification task, and different layers are used. The CNN architecture includes various layers: the input layer, the convolution layer, the ReLU, the pooling layer, the fully connected layer, and the Softmax layer. In the convolution layers, each neuron is represented by a series of deep convolutional kernels. The convolution process transforms into a correlation process, and the kernel is symmetrical. The convolution process has three primary advantages. The weight sharing is modeled in the same function. The method reduces the number of parameters and, as a result, the number of operations Local connectivity enables association analysis between adjacent pixels. Finally, object invariance allows for the location of the object. Target is independent of the object's position in the image [23].

CNN is a feed-forward detection technique that is extremely efficient. The network has a straightforward structure and fewer training parameters. CNN is an acronym that stands for a highly efficient detection method. However, the complexity and weights of the network model are reduced [6]. A type of hierarchical model called a CNN learns the attributes of an item by training on a large number of samples. They are composed of several layers, with the most recent ones developed on top of previously perfected features [24].

This study used ResNet50 and ResNet101 architecture to classify bitter gourd leaf disease. Since deep learning training takes so long and is restricted to a finite number of layers, this architecture was devised to circumvent these issues. ResNet's complexity can be understood as a recommendation to bypass conventional means of connection. The ResNets model has an advantage over other architectural models in that its performance does not degrade as the architecture grows deeper [25]. The only difference between ResNet networks was the layer to be used. BottleNeck blocks make up ResNet. The goal of using a bottleneck was to maximize the utilization of the GPU RAM and avoid wasting it on the costly 3x3 convolutions [26]. ResNet, a powerful deep learning architecture that won the ImageNet classification competition, is capable of solving numerous computer vision problems. ResNet indicates the number of deep layers, so we chose ResNet50, which has 50 layers for processing [27]. ResNet50 [2] refers to a network architecture based on numerous stacked residual units. The ResNet50 architecture's primary benefit is the use of skip connections, which allows for a more efficient solution to the vanishing gradient descent problem. Convolution, max-pooling, and stacking convolutions are the first steps performed by each ResNet. Having this stacked convolution is helpful since it eliminates the vanishing gradient issue [27]. ResNet50's architecture is shown in Fig. 5.

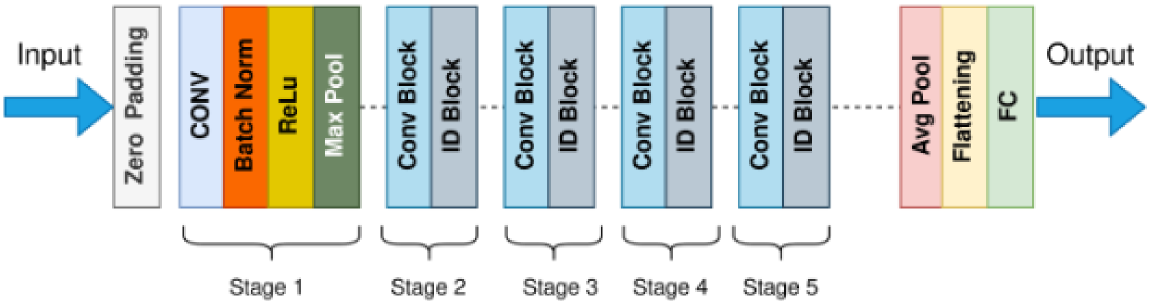


Fig. 5. Architecture of ResNet50

ResNet residual blocks can be obtained if the input and output data dimensions are the same. Furthermore, each ResNet block is made up of two or three layers (for ResNet-18 and ResNet-34 networks) (for ResNet-50 and ResNet-101 networks). ResNet-101 seems to have a total of 104 convolution layers. Furthermore, it has a total of 33-layer blocks, 29 of which use the block's output directly, as described above as a residual connection. At the end of each block, these residual interactions are utilized as the first input of the addition operator, which is used to get input for the next blocks [28]. The output of the first block is utilized in the convolution layer of the next four blocks. This convolution layer has a filter size of 1x1 and a stride of 1, and it is then followed by a batch normalizing layer that carries out a normalization operation. After that, the output that was generated is sent to the summation operator that is already present on the output of that block. The ResNet101 architecture is shown in Fig. 6.

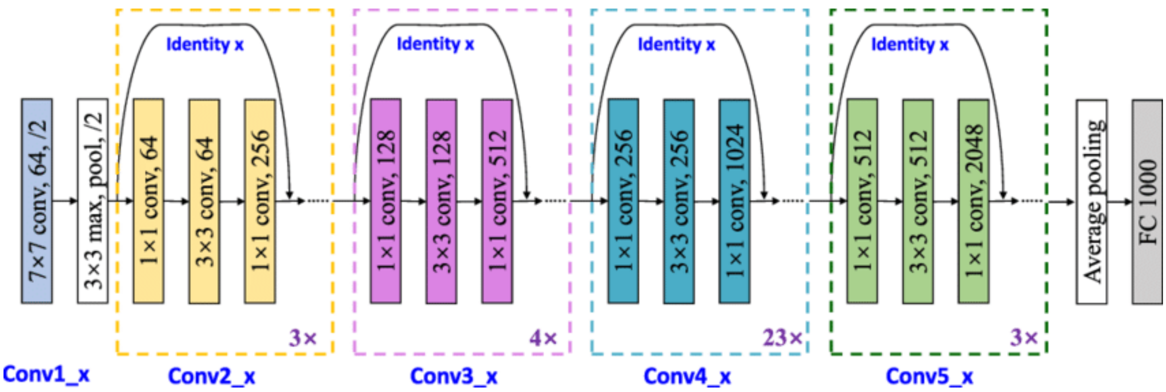


Fig. 6. Architecture of ResNet101

In a convolutional neural network (CNN) and similar neural network-based techniques, gradient descent is used to minimise the error function during training to update the internal parameters [29]. The update rule of an optimization algorithm is usually defined by the hyperparameters that determine its behavior (e.g, learning rate) [30]. The training gradient is computed using the traditional batch gradient descent technique, which slows down the computational process [31]. Keras optimizers were developed to address this issue. Here's the explanation, Adam (Adaptive Momentum) is an SGD optimization method that calculates adaptable learning rates for each parameter. Adam is a popular step-size strategic approach in the field of neural networks [32]. For each parameter, Adam computes adaptive learning rates. Adam, like Adadelta and RMSProp, stores bias-corrected past epoch squared gradients. Furthermore, Adam saves the bias-corrected past average gradient [29]. SGD (Stochastic Gradient Descent) is an iterative method for optimizing an objective function that incorporates proper smoothness features [33]. SGD algorithms have proven to be effective in optimizing large-scale deep learning models. Nadam (Nesterov adaptive momentum). This technique is a hybrid of optimization methods [32]. Adagrad (Adaptive Gradient), an optimization algorithm is proposed by Duchi [34]. The learning rate can be adjusted using this method throughout the convergence process. As a result, this method is effective with sparse datasets. Adagrad, on the other hand, has a new flaw: its learning rate slows during training. AdaDelta (Adaptive Delta) is an AdaGrad development that aims to correct the problem with AdaGrad's learning rate, which is decreasing with each iteration [35]. Adamax (Adaptive max pooling) is an adaptive SGD and the Adam version uses the infinity norm. AdaMax has the significant advantage of being significantly less sensitive to hyper-parameter selection than SGD [32]. RMSProp (Root Mean Square Propagation), to minimize the training time observed in Adagrad, the RMSProp optimizing functions were proposed, and their gradient descent decays [36]. By taking an exponential moving average of gradients rather than the cumulative sum of squared gradients, RMSprop Optimizer outperforms Adagrad Optimizer (Adagrad) [37]. FTRL (Follow The Regularized Leader) is an optimization algorithm developed by Google in the early 2010s for click-through rate prediction. It works best with shallow models with large and sparse feature spaces describing the algorithm. In this case study, we used eight optimizers to find the best optimizer to apply to ResNet50 and ResNet101.

The activation function is important for neural networks. A model for discovering and understanding highly complex and nonlinear systems. The activation function is computed after the convolution operation is completed. ReLU and softmax are common activations in CNN. This study used softmax activation in the last layer of the neural network. The probability distribution of an event is computed using the Softmax function, which takes into account 'n' different events. In a broader sense, this function will compute the probability of each target class across all of the different target classes that are feasible. Classification of several objects can benefit from using softmax. Within the classification process, the probabilistic value is calculated using the values from the layer before it (the Softmax layer). Classification at the Softmax layer results in a value that indicates the closest class. This function calculates the probabilities associated with each class based on the probabilistic value generated by the deep learning network layer [38]. The calculated probabilities will later be used to help determine the target class for the given inputs. Softmax was used more frequently than other functions such as ReLu, sigmoid, and tanh(). Softmax is useful for converting the output of a neural network's final layer into a basic probability distribution. The benefit of using softmax is that the output probability range is 0 to 1, and the sum of all probabilities in a single object is 1. Another benefit of using softmax is that it can be used for a variety of classifications [39].

2.6. Confusion Matrix

The dimensions of a confusion matrix are the real and predicted classes. Each row represents an instance of a real class. Whereas each column represents a predicted case class. A multiclass classifier's accuracy is defined as a percentage of the total number of correct predictions [40]. The confusion matrix, also recognized as the error matrix, is a graph layout that enables the visualization of algorithm performance [41]. After all, models were successfully executed, the results would be evaluated using a confusion matrix. The values of correct and incorrect assumptions are compared. The computations applied to the experiment results consist of accuracy. The theorems toward accuracy are simply as follows.

$$Accuracy = \frac{Number\ of\ correctly\ samples}{Total\ number\ of\ samples} \tag{1}$$

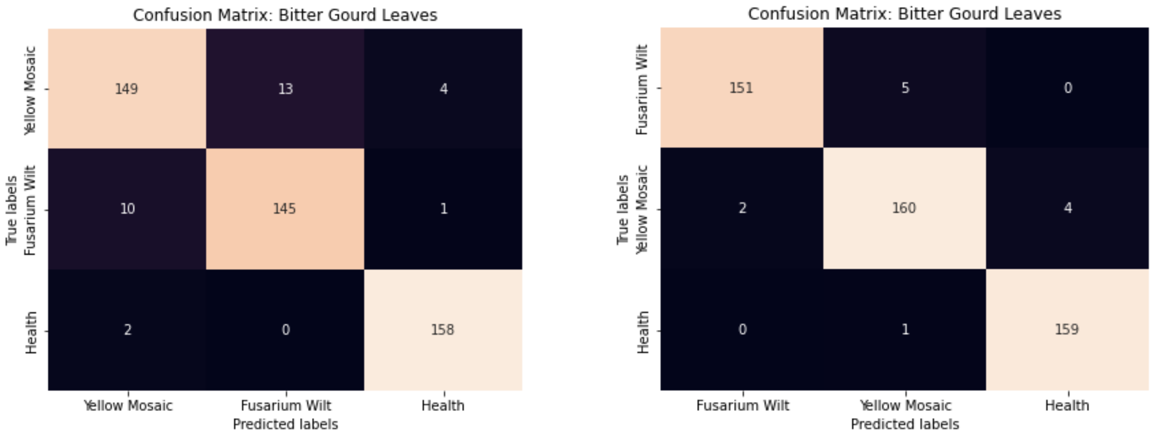
3. Results and Discussion

The results were obtained through various stages. The model has been tested on a device that meets the specifications: Windows 10 Pro 64-bit (10.0, build 19044), Intel(R) Core(TM) i5-6200 CPU @ 2.30GHz, memory 4GB. Python is used to program all of the models. This model employs a total dataset of 2490 images of bitter gourd leaves with three types of classes, namely Fusarium Wilt, Yellow Mosaic, and Healthy Leaf, split into 80% training and 20% testing, with each image measuring 128x128 pixels. We test the bitter gourd image classification performance using the datasets in Table 1. In this part, we discuss the experimental results implemented based on ResNet50 and ResNet101 by using 100 epochs, batch size 32, and eight optimizers (Adam, Adamax, Nadam, SGD, RMSprop, Adadelta, Adagrad, and Ftrl). This study attempted to improve the performance of CNN architectures by training the best models (obtained in the previous stage) through several deep learning optimization functions. The classifier algorithm is presented in Table 2, which outlines a set of rules designed to achieve accuracy.

Table 2. The accuracy report of classification using ResNet101 and ResNet50

Models	Epoch	Optimizer	Accuracy
ResNet101	100	Adam	94%
	100	Adamax	90%
	100	SGD	87%
	100	Nadam	82%
	100	RMSprop	91%
	100	Adadelta	70%
	100	Adagrad	82%
	100	Ftrl	83%
ResNet50	100	Adam	98%
	100	Adamax	96%
	100	SGD	77%
	100	Nadam	97%
	100	RMSprop	95%
	100	Adadelta	73%
	100	Adagrad	92%
	100	Ftrl	90%

Adam was the most effective optimizer across all eight deep learning architectures tested. The ResNet101 model trained with the Adam optimizer achieved the highest accuracy of 94%. The accuracy results are presented in Table 2. From Fig. 7(a), we can evaluate the accuracy performance of ResNet101. The second observation uses the ResNet50 model with the Adam optimizer, and the resulting accuracy is 98%. This method improves accuracy. The details are presented in Table 2. From Fig. 7(b), we can evaluate the accuracy performance of ResNet50. However, when the optimizing functions were switched to Adadelta, the performance was degraded.



(a) (b)

Fig. 7. Confusion Matrix performance (a) ResNet101, (b) ResNet50

Adam's Hyper-parameters have an intuitive interpretation and usually require some fine-tuning, when minimizing the cost function in training neural nets, it appears that the optimizer almost always works better, faster, and more reliably reaching a global minimum, at each iteration, the Adam algorithm estimates moments in each batch using exponentially moving averages [42]. ResNet101 outperformed ResNet50 despite having a more complex and multi-layered architecture than ResNet50, according to model evaluation findings and accuracy tests. This was because the training data was quite limited when trained with such a large architecture. The graph accuracy of ResNet50 with eight optimizers can be shown in Fig. 8. The figure shows a comparison of the accuracy of the ResNet50 model with various optimization algorithms. The highest results were obtained using Adam with 98% accuracy, followed by Nadam (97%), Adamax (96%), and RMSProp (95%). Meanwhile, the lowest accuracy was obtained using Adadelta (73%) and SGD (77%), while Adagard and Ftrl achieved 92% and 90% accuracy, respectively. This shows that Adam is the most optimal optimization algorithm to improve the accuracy of ResNet50 in image classification.

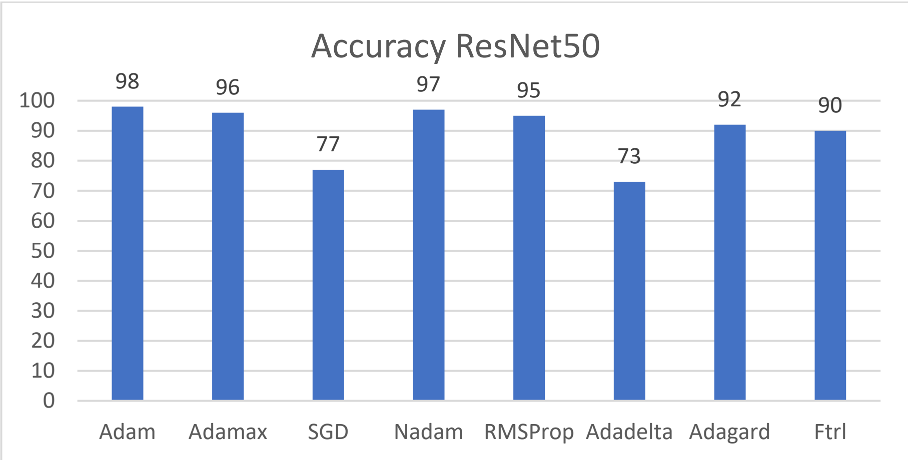


Fig. 8. Accuracy of ResNet50 by 8 optimizers

The graph accuracy of ResNet101 with eight optimizers can be shown in Fig. 9. The figure shows the accuracy comparison of several optimization algorithms used in model training. Based on the graph, the Adam algorithm produces the highest accuracy of 94%, followed by RMSProp with 91% accuracy, and Adamax at 90%. Meanwhile, the algorithm with the lowest accuracy is Adadelta which only reaches 70%, while other optimizations such as SGD, Nadam, Adagard, and Ftrl have accuracies between 82% and 87%. This shows that the selection of optimization algorithms has a significant effect on the performance of the classification model.

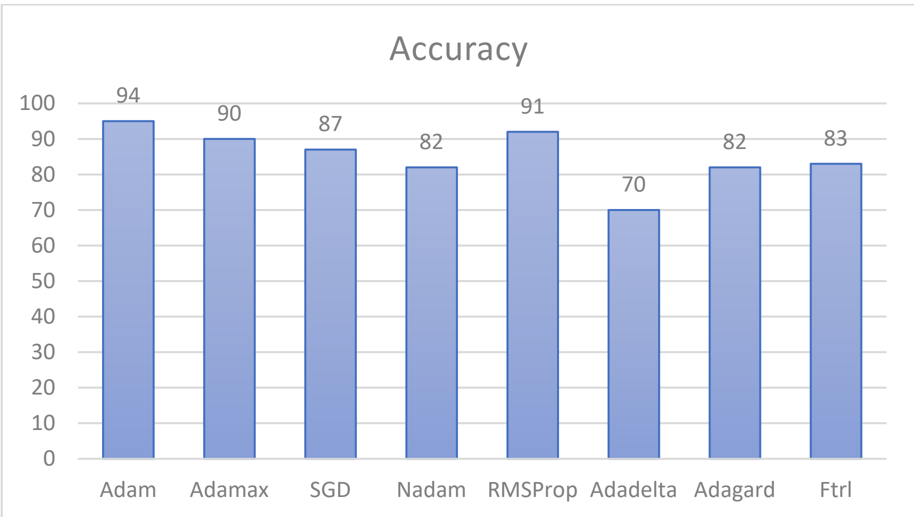


Fig. 9. Accuracy of ResNet101 by eight optimisers

4. Conclusion

This paper introduced a deep learning-based classification method for bitter gourd leaf diseases to classify three classes of bitter gourd leaf diseases: yellow mosaic, fusarium wilt, and health. We tested several deep learning models, including ResNet50 and ResNet101, on 2490 images by eight optimizers: Adam, SGD, Adamax, Nadam, Adadelta, RMSprop, Adagrad, Nadam, and Ftrl. All optimizers had different execution times. Adam was the most successful optimizer. Adam's hyperparameters have an intuitive interpretation and usually require some fine-tuning. Thus, the optimizer almost always works better, faster, and more reliably, reaching a global minimum. By comparing the accuracy of ResNet50 and ResNet101, it was found that ResNet50 had better accuracy than ResNet101. The ResNet101 performance is quite similar to ResNet50 but less efficient. ResNet101 architecture was more complex than ResNet50 architecture, but ResNet101 had worse performance. This is because the training data is relatively small if the data is slightly trained with a large architecture, such as ResNet101. Still, it does not necessarily produce a better accuracy value for classification. There were still some challenges in classifying plant leaf diseases, which were limited by the following: We used an imbalanced dataset. Constant lighting keeps the dataset from diversifying, using only 2 diseased dataset classes. In the future use a balanced and diverse dataset, then add disease classes for research to be carried out.

Acknowledgment

The author is very grateful to have completed this research and is very grateful to the Telkom University campus, which has provided support and funding in carrying out this research.

Declarations

Author contribution. All authors contribution equally to the main contributor to this paper.
Funding statement. This work was supported by Telkom University
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] N. Kaur and V. Devendran, “Research Article Plant leaf disease detection using ensemble classification and feature extraction,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 11, pp. 2339–2352, 2021, [Online]. Available at: https://www.proquest.com/openview/a06521be5cd98081e33c620e2665b3ef/1?pq-origsite.

[2] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant leaf disease classification using EfficientNet deep learning model,” *Ecol. Inform.*, vol. 61, p. 101182, Mar. 2021, doi: 10.1016/j.ecoinf.2020.101182.

[3] F. Saeed *et al.*, “Bitter melon (Momordica charantia): a natural healthy vegetable,” *Int. J. Food Prop.*, vol. 21, no. 1, pp. 1270–1290, Jan. 2018, doi: 10.1080/10942912.2018.1446023.

[4] M. M. Hasan, K. Alam, M. N. Ahmed Diganta, A. U. Nur, M. T. Habib, and F. Ahmed, “Defected Bitter Gourd Detection Using Convolutional Neural Network: A Computer Vision Approach to Reduce Cost and Time,” in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Jul. 2021, pp. 1–6, doi: 10.1109/ICCCNT51525.2021.9579995.

[5] N. E. M. Khalifa, M. H. N. Taha, L. M. Abou El-Maged, and A. E. Hassanien, “Artificial Intelligence in Potato Leaf Disease Classification: A Deep Learning Approach,” in *Studies in Big Data*, vol. 77, Springer, Cham, 2021, pp. 63–79, doi: 10.1007/978-3-030-59338-4_4.

[6] A. R. Luaibi, T. M. Salman, and A. H. Miry, “Detection of citrus leaf diseases using a deep learning technique,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, p. 1719, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1719-1727.

[7] N. Sharma, V. Jain, and A. Mishra, “An Analysis Of Convolutional Neural Networks For Image Classification,” *Procedia Comput. Sci.*, vol. 132, pp. 377–384, Jan. 2018, doi: 10.1016/j.procs.2018.05.198.

[8] M. A. Kadhim and M. H. Abed, “Convolutional Neural Network for Satellite Image Classification,” in *Studies in Computational Intelligence*, vol. 830, Springer, Cham, 2020, pp. 165–178, doi: 10.1007/978-3-030-14132-5_13.

[9] C. Zhou, S. Zhou, J. Xing, and J. Song, "Tomato Leaf Disease Identification by Restructured Deep Residual Dense Network," *IEEE Access*, vol. 9, pp. 28822–28831, 2021, doi: 10.1109/ACCESS.2021.3058947.

[10] S. Arivazhagan and S. V. Ligi, "Mango Leaf Diseases Identification Using Convolutional Neural Network," *Int. J. Pure Appl. Math.*, vol. 120, no. 6, pp. 11067–11079, 2018, [Online]. Available at: https://d1wqtxts1xzle7.cloudfront.net/76332505/731-libre.pdf?1639784653=&response-content-disposition=inline%3B+filename%3DMango_Leaf_Diseases_Identification_Using.pdf&Expires=1751948967&Signature=KKoNfRZstHZZfeyt9F7AqDFlLEmauxIGHBhhomiSNeHQpKnffZWzyiQI-z.

[11] H.-J. Yu and C.-H. Son, "Leaf Spot Attention Network for Apple Leaf Disease Identification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, vol. 2020-June, pp. 229–237, doi: 10.1109/CVPRW50498.2020.00034.

[12] S. Laosim and T. Samanchuen, "Classification of Pomelo Leaf Diseases Using Convolution Neural Network," in *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, May 2021, pp. 577–580, doi: 10.1109/ECTI-CON51831.2021.9454782.

[13] O. K. Pal, "Identification of Paddy Leaf Diseases Using a Supervised Neural Network," in *2021 16th International Conference on Emerging Technologies (ICET)*, Dec. 2021, pp. 1–4, doi: 10.1109/ICET54505.2021.9689788.

[14] Z. Liu, X. Yuan, J. Weng, Y. Liao, and L. Xie, "Application of Bitter Gourd Leaf Disease Detection Based on Faster R-CNN," Springer, Singapore, 2021, pp. 191–198, doi: 10.1007/978-981-16-1843-7_24.

[15] E.-S. M. El-Kenawy *et al.*, "Advanced Meta-Heuristics, Convolutional Neural Networks, and Feature Selectors for Efficient COVID-19 X-Ray Chest Image Classification," *IEEE Access*, vol. 9, pp. 36019–36037, 2021, doi: 10.1109/ACCESS.2021.3061058.

[16] G. Kaur, M. Pathak, D. Singla, A. Sharma, P. Chhuneja, and N. K. Sarao, "High-Density GBS-Based Genetic Linkage Map Construction and QTL Identification Associated With Yellow Mosaic Disease Resistance in Bitter Gourd (Momordica charantia L.)," *Front. Plant Sci.*, vol. 12, p. 671620, Jun. 2021, doi: 10.3389/fpls.2021.671620.

[17] Y. Tian, X. Fu, X. Yan, X. Li, H. Peng, and K. Gao, "The control efficacy and mechanism of Talaromyces purpurogenus on Fusarium wilt of bitter gourd," *Biol. Control*, vol. 165, p. 104804, Feb. 2022, doi: 10.1016/j.biocontrol.2021.104804.

[18] M. Jannesari *et al.*, "Breast Cancer Histopathological Image Classification: A Deep Learning Approach," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2018, pp. 2405–2412, doi: 10.1109/BIBM.2018.8621307.

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

[20] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10.1109/ACCESS.2019.2907383.

[21] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant Disease Detection and Classification by Deep Learning," *Plants*, vol. 8, no. 11, p. 468, Oct. 2019, doi: 10.3390/plants8110468.

[22] S. Uğuz and N. Uysal, "Classification of olive leaf diseases using deep convolutional neural networks," *Neural Comput. Appl.*, vol. 33, no. 9, pp. 4133–4149, May 2021, doi: 10.1007/s00521-020-05235-5.

[23] S. T. Krishna and H. K. Kalluri, "Deep learning and transfer learning approaches for image classification," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 427–432, 2019, [Online]. Available at: https://www.researchgate.net/profile/Hemantha-Kumar-Kalluri/publication/333666150.

[24] T. N. Pham, L. Van Tran, and S. V. T. Dao, "Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection," *IEEE Access*, vol. 8, pp. 189960–189973, 2020, doi: 10.1109/ACCESS.2020.3031914.

[25] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer," *Procedia Comput. Sci.*, vol. 179, pp. 423–431, Jan. 2021, doi: 10.1016/j.procs.2021.01.025.

[26] A. Jibhakate, P. Parnerkar, S. Mondal, V. Bharambe, and S. Mantri, "Skin Lesion Classification using Deep Learning and Image Processing," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, Dec. 2020, pp. 333–340, doi: 10.1109/ICISS49785.2020.9316092.

[27] S. T., R. Khilar, and M. Subaja Christo, "WITHDRAWN: A comparative analysis on plant pathology classification using deep learning architecture – Resnet and VGG19," *Mater. Today Proc.*, Jan. 2021, doi: 10.1016/j.matpr.2020.11.993.

[28] A. Demir, F. Yilmaz, and O. Kose, "Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3," in *2019 Medical Technologies Congress (TIPTEKNO)*, Oct. 2019, vol. 2019-Janua, pp. 1–4, doi: 10.1109/TIPTEKNO47231.2019.8972045.

[29] K. R. Prilianti, T. H. P. Brotosudarmo, S. Anam, and A. Suryanto, "Performance comparison of the convolutional neural network optimizer for photosynthetic pigments prediction on plant digital image," in *AIP Conference Proceedings*, Mar. 2019, vol. 2084, no. 1, p. 020020, doi: 10.1063/1.5094284.

[30] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On Empirical Comparisons of Optimizers for Deep Learning," pp. 1–27, Oct. 2019. [Online]. Available at: https://arxiv.org/pdf/1910.05446.

[31] M. Yaqub *et al.*, "State-of-the-Art CNN Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images," *Brain Sci.*, vol. 10, no. 7, p. 427, Jul. 2020, doi: 10.3390/brainsci10070427.

[32] S. H. Haji and A. M. Abdulazeez, "Comparison Of Optimization Techniques Based On Gradient Descent Algorithm: A Review," *PalArch's J. Archaeol. Egypt / Egyptol.*, vol. 18, no. 4, pp. 2715–2743, Feb. 2021. [Online]. Available at: https://archives.palarch.nl/index.php/jae/article/view/6705.

[33] M. N. Halgamuge, E. Daminda, and A. Nirmalathas, "Best optimizer selection for predicting bushfire occurrences using deep learning," *Nat. Hazards*, vol. 103, no. 1, pp. 845–860, Aug. 2020, doi: 10.1007/s11069-020-04015-7.

[34] N. Zhang, D. Lei, and J. F. Zhao, "An Improved Adagrad Gradient Descent Optimization Algorithm," in *2018 Chinese Automation Congress (CAC)*, Nov. 2018, pp. 2359–2362, doi: 10.1109/CAC.2018.8623271.

[35] A. Wibowo, P. W. Wiryawan, and N. I. Nuqoyati, "Optimization of neural network for cancer microRNA biomarkers classification," *J. Phys. Conf. Ser.*, vol. 1217, no. 1, p. 012124, May 2019, doi: 10.1088/1742-6596/1217/1/012124.

[36] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers," *Plants*, vol. 9, no. 10, p. 1319, Oct. 2020, doi: 10.3390/plants9101319.

[37] S. R. Labhsetwar, S. Haridas, R. Panmand, R. Deshpande, P. A. Kolte, and S. Pati, "Performance Analysis of Optimizers for Plant Disease Classification with Convolutional Neural Networks," in *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, Jan. 2021, pp. 1–6, doi: 10.1109/ICNTE51185.2021.9487698.

[38] A. Çınar, M. Yıldırım, and Y. Eroğlu, "Classification of Pneumonia Cell Images Using Improved ResNet50 Model," *Trait. du Signal*, vol. 38, no. 1, pp. 165–173, Feb. 2021, doi: 10.18280/ts.380117.

[39] A. Kholik, A. Harjoko, and W. Wahyono, "Classification of Traffic Vehicle Density Using Deep Learning," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 1, pp. 69–80, Jan. 2020, doi: 10.22146/IJCCS.50376.

[40] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazan, and H. Avila-George, "Classification of Cape Gooseberry Fruit According to its Level of Ripeness Using Machine Learning Techniques and Different Color Spaces," *IEEE Access*, vol. 7, pp. 27389–27400, 2019, doi: 10.1109/ACCESS.2019.2898223.

[41] G. M. Foody, "Explaining the unsuitability of the kappa coefficient in the assessment and comparison of the accuracy of thematic maps obtained by image classification," *Remote Sens. Environ.*, vol. 239, p. 111630, Mar. 2020, doi: 10.1016/j.rse.2019.111630.

[42] S. Y. SEN and N. OZKURT, "Convolutional Neural Network Hyperparameter Tuning with Adam Optimizer for ECG Classification," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2020, pp. 1–6, doi: 10.1109/ASYU50717.2020.9259896.