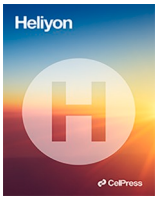
Heliyon 10 (2024) e39569

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Maize disease classification using transfer learning and convolutional neural network with weighted loss

Krisnanda Ahadian [a], Novanto Yudistira [a,*], Bayu Rahayudi [a], Ahmad Hoirul Basori [b], Sharaf J. Malebary [b], Sami Alesawi [b], Andi Besse Firdausiah Mansur [b], Almuhannad S. Alorfi [b], Omar M. Barukab [b]

[a] *Informatics Department, Faculty of Computer Science, Brawijaya University, 65145, Malang, Indonesia*
[b] *Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh, 21911, Makkah, Saudi Arabia*

ARTICLE INFO

Keywords:
Maize disease
Classification
Convolutional neural network
Deep learning
Machine learning

ABSTRACT

Maize stands out as a versatile commodity, finding applications in food and animal feed industries. Notably, half of the total demand for maize is met through its utilization as animal feed. Despite its importance, maize cultivation often grapples with crop failures resulting from delayed disease management or insufficient knowledge about these diseases, impeding timely intervention. The advent of technological advancements, particularly in Machine Learning, presents solutions to address these challenges. This research focuses on employing a Convolutional Neural Network (CNN) to classify maize plant diseases. Two datasets form the foundation of this study. The first dataset encompasses 4144 images distributed across 4 classes, while the second dataset comprises 5155 images distributed among 7 to 8 classes. The second dataset encounters issues related to imbalanced class distribution, where certain classes possess substantially more data than others. To mitigate this imbalance, the weighted cross-entropy loss method is employed. During experimentation, three distinct architectural models—ResNet-18, VGG16, and EfficientNet—are rigorously tested. Additionally, various optimizers are explored, with noteworthy results indicating that both datasets achieve peak accuracy through the use of the SGD (Stochastic Gradient Descent) optimization. For the first dataset, optimal results are obtained with the VGG16 architecture, leveraging a frozen layer in the classification stage and achieving an impressive accuracy of 97.146 %. Shifting the focus to the second dataset, the most favorable outcome is realized by employing the EfficientNet architecture without a frozen layer, coupled with the implementation of weighted loss to address the class imbalance, resulting in an accuracy of 94.798 %.

1. Introduction

Maize became the most vital cereal crop globally, boasting the highest worldwide production and adaptability to diverse climates. It can be a staple food for human diets and high-quality animal feed. Moreover, maize serves as the principal source material for numerous industrial products. Despite its capacity for generating high grain yields, the vulnerability of maize crops to various diseases presents a significant obstacle in increasing yields, resulting in an annual production decrease of 6–10 % [1]. In several regions of

* Corresponding author.
E-mail address: yudistira@ub.ac.id (N. Yudistira).

https://doi.org/10.1016/j.heliyon.2024.e39569
Received 7 June 2024; Received in revised form 8 October 2024; Accepted 17 October 2024
Available online 19 October 2024

Fig. 1. Common rust.

Fig. 2. Blight.

Fig. 3. Gray leaf spot.

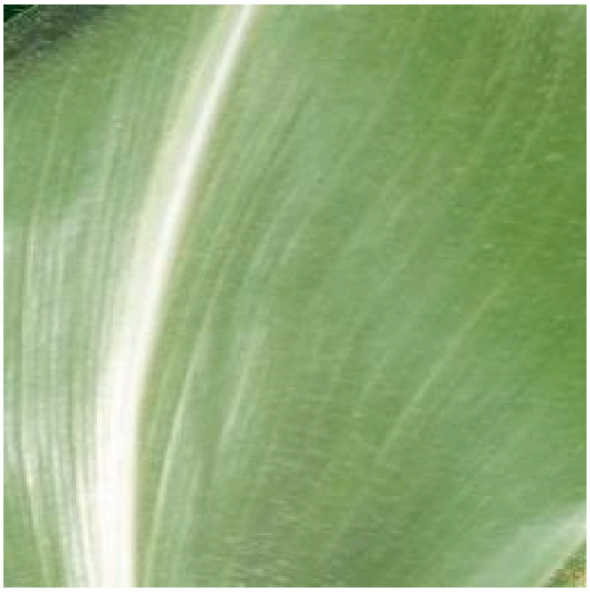


Fig. 4. Healthy.

Fig. 5. Fallarmyworm.

Indonesia, maize is also a staple food. Besides being a source of carbohydrates, maize is used for multiple purposes including livestock feed, oil extraction, flour production, and as a raw material for industries [2]. However, multiple leaf diseases arise in maize crops, affection to the quality of the crops [3] and its yield. Moreover, these plant leaf diseases significantly compromise the production rate of maize crops [4].

Plant diseases pose a significant challenge in agricultural production [5]. Artificial Intelligence technology is widely contributed in this field. Deep learning can be employed for multiple tasks including event prediction, object recognition, and disease diagnosis. Image processing aims to facilitate the recognition and classification of objects while handling large amounts of data simultaneously [6]. Deep Learning can automatically reveal the essential features required for classification tasks. Among these methods, CNN (Convolutional Neural Network), a specific type of Deep Learning architecture, has demonstrated great performance across multiple domains, particularly within agriculture [7].

The process of identifying leaf diseases through visual observation requires experts and continuous monitoring of the crops [8]. This method becomes expensive, time-consuming, and less reliable. Implementing deep learning methods allows for automated, rapid,

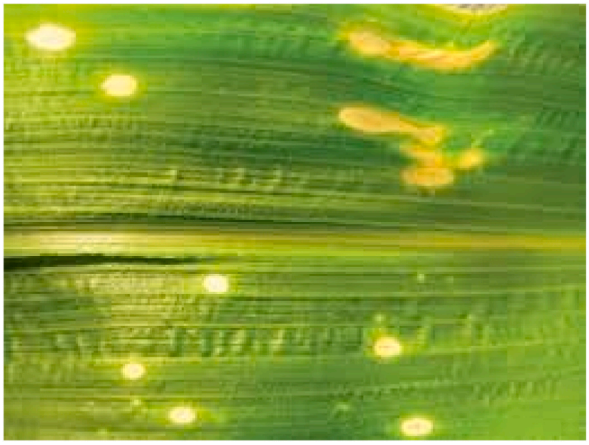


Fig. 6. Hericideburn

Fig. 7. Zincdeficiency.

and highly accurate detection of leaf diseases [9]. Currently, the application of computer vision (CV) and machine learning (ML) is increasing in the field of plant disease detection due to their adept performance, even in challenging conditions [10].

Indonesia faces a deficiency in human resources within the field of managing maize plant diseases. When dealing with these concerns, numerous mistakes could be made, resulting in increased failure rates in Indonesia's agricultural production [11]. According to Huda et al. (2021) [12], The traditional approach done through manual observation by farmers can be improved with the rapid advancements in Information Technology. The identification of disease in maize leaves presents a challenge due to the presence of multiple diseases with overlapping symptoms that complicate their differentiation. Conventional methods heavily rely on manual recognition, which is susceptible to inaccurate results [13]. Consequently, this can lead to crop loss, hindering the established maize production goals.

This research is supported by previous studies. Rasywir et al. (2020) [14] discovered that CNN achieved excellent accuracy in image recognition. The test result from testing was stored in a configuration matrix, including 2.490 labeled images of oil palms in 11 disease categories. The highest accuracy reached was 0.89, and the lowest was 0.83, with an average accuracy of 0.87. This showed the effectiveness of CNN in classifying oil palm images.

On the other hand, research by Akhyari et al. (2021) [15] found that employing the CNN method for diagnosing maize leaf diseases resulted in an overall accuracy of over 90 %. The results accuracy is based on the leaf characteristics.

According to Huda et al. (2021) [12], deep learning delivers better results compared to other methods like Multi-Layer Perceptron (MLP). This is because CNN has a large depth of neurons and is commonly used for image classification. The MLP method lacks the retention of spatial data from image classifications, assuming that each pixel are an independent feature that results in the worst results. CNN can classify images more effectively than other methods due to its higher accuracy.

After the background explanations provided, it has been discovered that previous researchers have made significant findings in

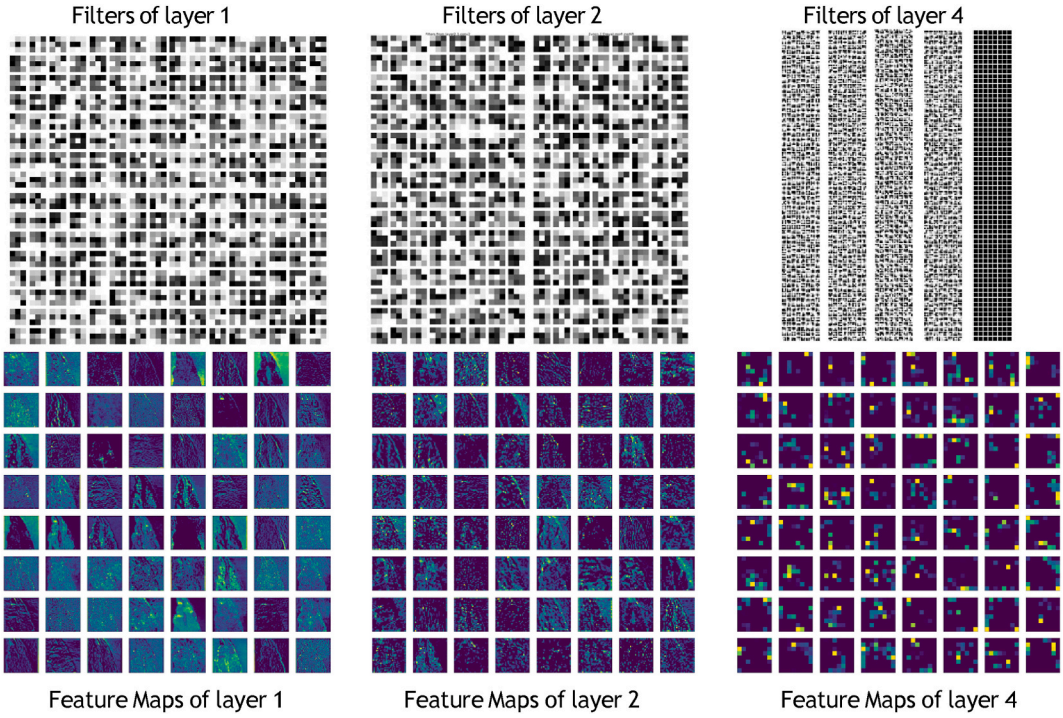


Fig. 8. Visualization of the filters and feature maps across different layers of the ResNet18 architecture. Filters from layers 1, 2, and 4 are shown alongside their corresponding feature maps. These visualizations highlight the progressive transformation of input data, where initial layers (layer 1) capture basic features, while deeper layers (layer 4) extract more complex and abstract patterns.

disease detection in plants, making it a compelling subject for further research. The goal of this research is to reevaluate and test the previous findings. Thus, this research will use the CNN to classify maize disease. The CNN utilizes the convolution process by employing a convolutional kernel or filter of a specific size, which is systematically moved across an input image. In this study, three CNN architectures will be evaluated, there will be ResNet18, VGG16, and EfficientNet-b0. Besides using these architectures, this study will implement the Weighted Cross Entropy Loss technique due to the presence of imbalanced classes within the dataset.

2. Related works

Ubaidillah et al. (2022) [16], in their research utilized the Random Forest and Näıve Bayes methods. The study employed a dataset consisting of 3500 images of maize plant leaves categorized into 4 classes. Their testing results indicated that the Neural Network method yielded the best outcomes, achieving an AUC of 90.09 %, classification accuracy of 74.44 %, an f1-score of 72.01 %, precision of 74.14 %, and a recall of 74.43 %.

In a study by Sandotra et al. (2023) [17], a CNN architecture was employed, testing several models. The resulting model successfully classified 4 classes: Healthy, Blight, Gray Leaf Spot, and Common Rust, using a dataset of 4188 images. Evaluation metrics, including precision, recall, and F1-Score, revealed mean average precision values of 92.91 % for EfficientNet-b0, 89.95 % for InceptionNetV3, 88.53 % for VGG19, 91.08 % for VGG16, and 78.19 % for ResNet50 in model testing on the test dataset.

Yuliany and Nur Rachman (2022) [6] identified overfitting issues with the CNN method in their research. To address this, the study proposed three types of data division between training and testing data, alongside various parameters. The evaluation indicated that the 90 %:10 % data split was most suitable for the dataset, with the architectures achieving training accuracies of 83.02 %, 78.30 %, and 81.13 %. Testing accuracy values for these three models were 69.33 %, 77.33 %, and 76 %.

Huda et al. (2021) [12] showcased in their research the success of a web system utilizing Python and CNN, achieving good classification results. The best accuracy value reached 94.44 %.

In another study by Irawan et al. (2021) [18], validation results were presented for an application built using CNN and SqueezeNet architecture. The application demonstrated the ability to recognize Anthracnose, Ringspot Virus, and healthy papaya through leaves with 97 % accuracy, while accuracy through fruits reached 70 %.

The research conducted shares a common aspect, emphasizing the use of models related to testing methods utilizing CNN.

This approach capitalizes on CNN's effectiveness in image learning functions, designed with specialized layers called convolutional layers to extract patterns from various parts of the image, facilitating efficient classification.

Despite significant advancements, there remains a need to enhance the diagnosis and classification of diseases affecting maize leaves in real-world field conditions. Specific models that perform exceptionally well in controlled laboratory environments often yield

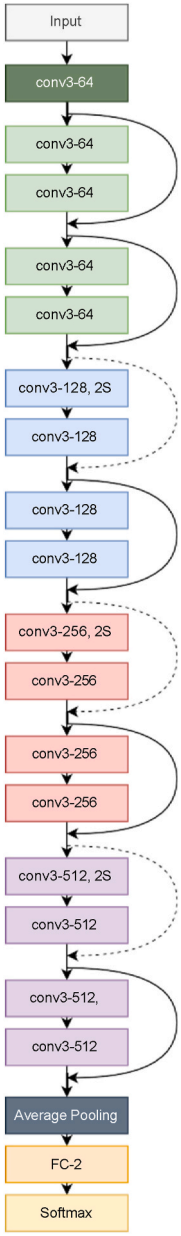


Fig. 9. ResNet-18 architecture [36].

unsatisfactory outcomes in real-world scenarios [19]. Challenges in precisely categorizing maize diseases include strong visual resemblances between disease types, substantial background noise in field environments, and inconsistent occurrences of various crop diseases [20].

The distinctive feature of this research, compared to previous studies, lies in its focus on maize plants as the second staple crop after rice. The increasing demand for maize aligns with the growing population and industrial needs. Additionally, the exploration of alternative energy sources from plant-based fuels, including maize for bioethanol production, addresses to the scarcity of fossil fuels and the need for sustainable alternatives.

3. Methodology

3.1. Dataset

To classify diseases in maize plants, an appropriate dataset is required, which is a dataset that contains images of both diseased and

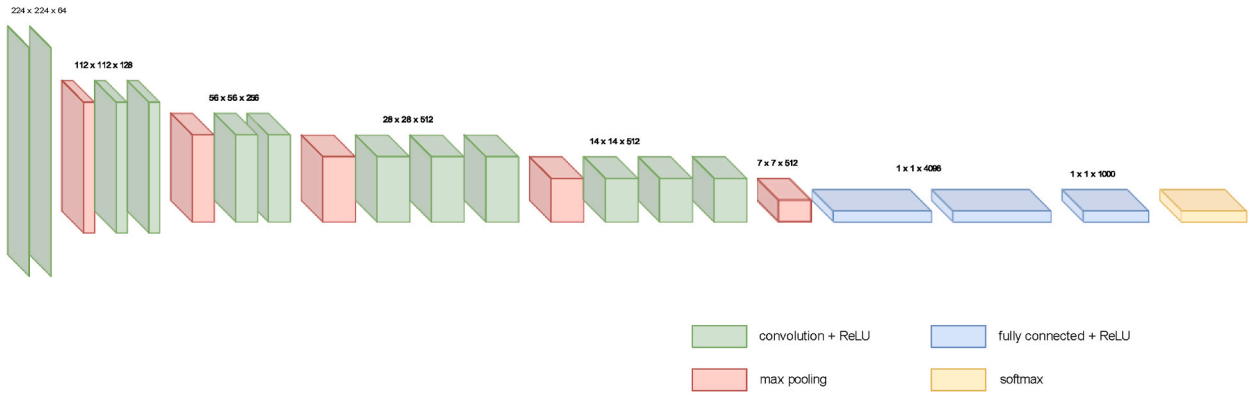


Fig. 10. VGG16 architecture [43].

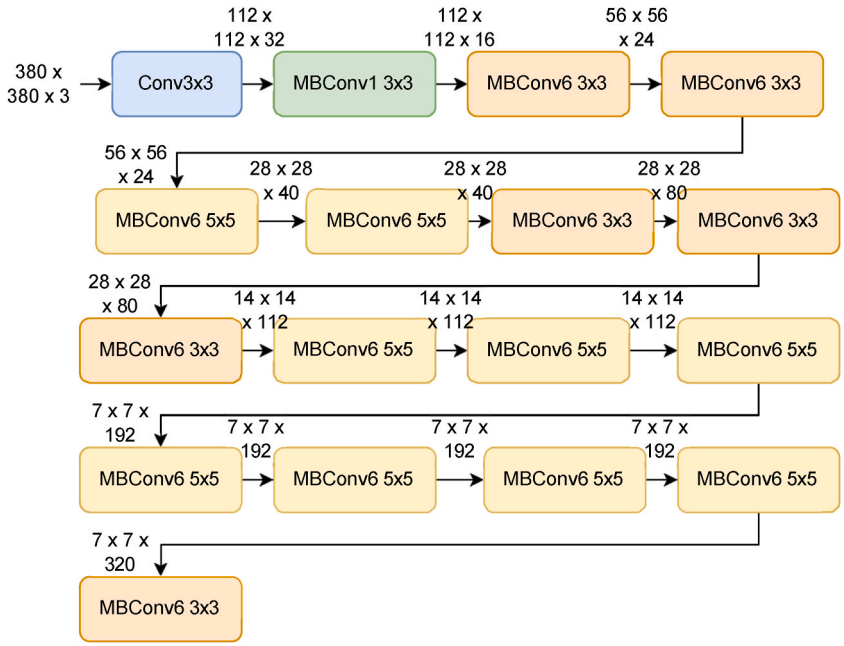


Fig. 11. EfficientNet-B0 architecture [46].

healthy maize plants. In this research, two datasets will be used.

3.1.1. First dataset

The first dataset is obtained from Kaggle, and the link to it is as follows: https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset. This dataset is a combination of PlantVillage and PlantDoc. The first dataset consists of 4188 images of Maize plants, which are divided into four classes. There are Healthy, Blight, Common Rust, and Gray Leaf Spot.

1 Common Rust

Common rust typically occurs at moist and cold environments, yet most of the time does not cause yield loss. Young leaves are more prone to this disease. Symptoms include the appearance of rust or dark brown-colored pustules on the bottom of leaf surfaces. These pustules carry brown spores inside of them. The pustules can darken over time, and under severe conditions, may cause leaf chlorosis and death [21]. Sheaths and leaves can also be infected. An example image of the Common Rust is shown in Fig. 1.

2 Blight

There are two types of blight included in this class, which are northern blight and southern blight. Northern blight is caused by wet

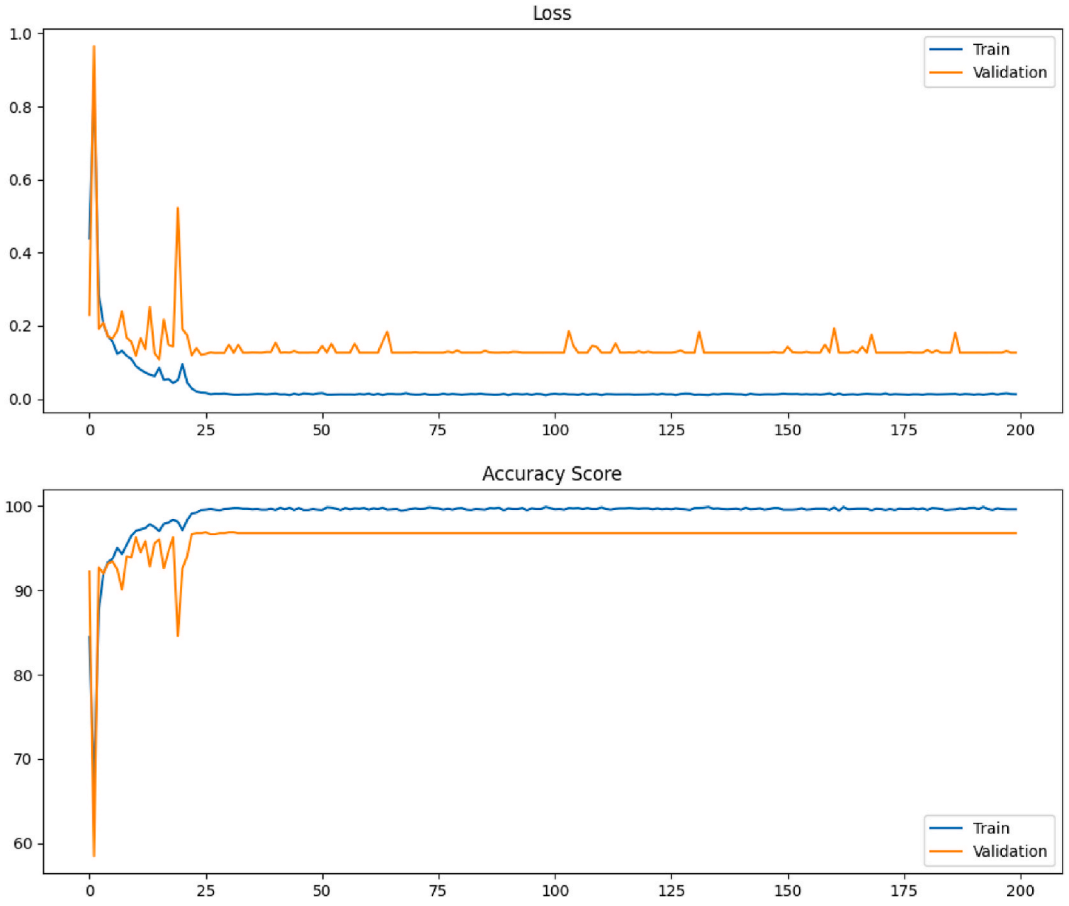


Fig. 12. ResNet-18 with first dataset.

humid cool weather, while southern blight is caused by warm and humid weather [22]. Common symptoms include:

- Northern Maize Leaf Blight

Tan lesions appear to be slender and oblong tapering at the ends with sizes around 1–6 inches. Lesions start from lower leaves, and run parallel to the leaf margins to the upper parts of the plant. In some cases may cover up the entire leaf. Spores may be produced under the lesions which give the appearance of dusty green fuzz [23].

- Southern Maize Leaf Blight

Early symptoms appear to be small diamond-shaped lesions, which will elongate over time when mature [24]. Lesions are usually seen on lower leaves and will move to the upper part of the plants, with lengths ranging from 1/4 to 3/4 inches. Necrotic lesions will appear as brown, and sometimes purplish tinge or brownish red edge [25].

An example of the Blight disease is shown in Fig. 2.

3 Gray Leaf Spot

Gray leaf spot is a fungal disease that usually occurs during warm temperatures and high-humidity environments. Initially, it will first appear in the lower leaves as a small lesion with a yellow halo. Over time, the lesions will expand to a rectangular shape with blunt ends and pale brown or gray colors. Mature lesions may appear opaque when brought to light and have distinct edges [26]. These lesions may eventually kill the leaves. An example of the Gray Leaf Spot disease is shown in Fig. 3.

4 Healthy

This class contains images of maize plant leaves that are not affected by any diseases. Therefore, the characteristics of the leaves in

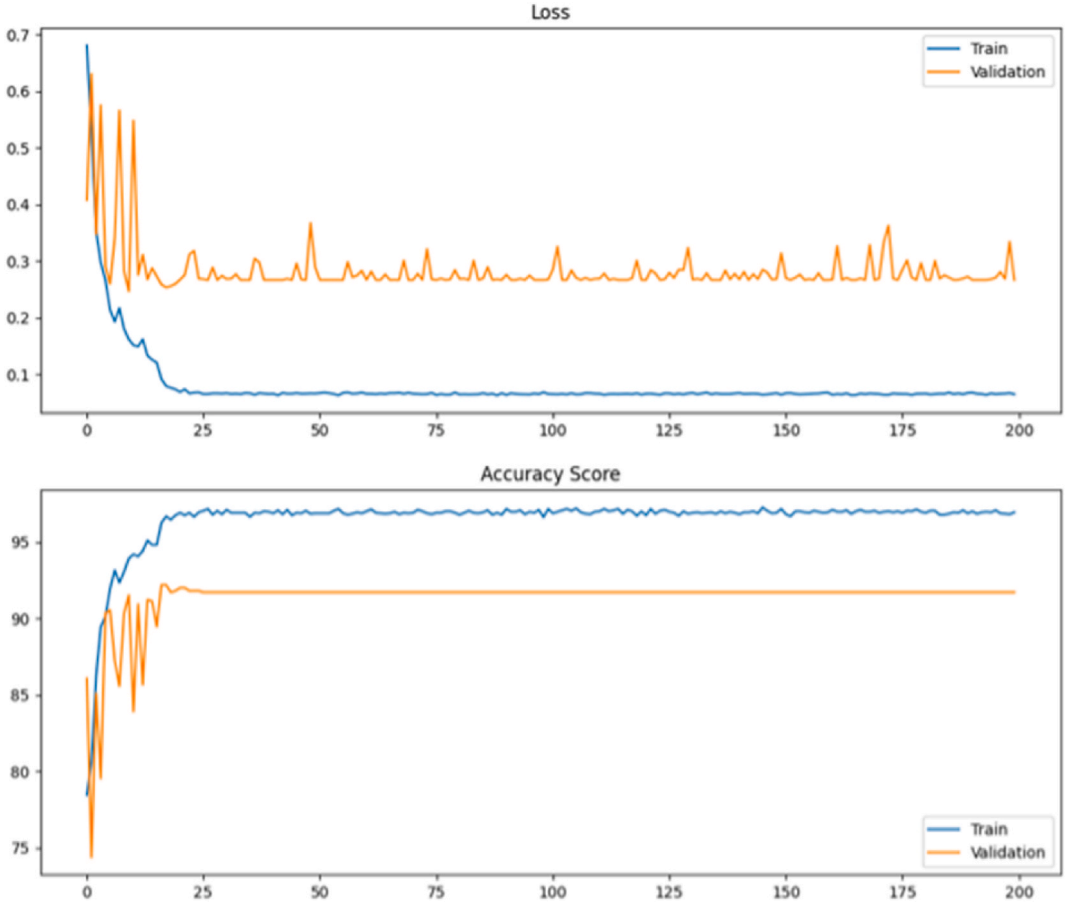


Fig. 13. ResNet-18 with second dataset.

this class resemble those of a generally healthy maize plant. An example of the Healthy is shown in Fig. 4.

3.1.2. Second dataset

The second dataset is a combination of the first dataset and another dataset obtained from previous research titled Maize leaf disease identification based on WG-MARNet. The total number of maize plant images in the second dataset is 5155. The dataset initially consists of 8 classes, which are blight, common rust, gray leaf spot, healthy, cerospora, fallarmyworm, herbicideburn, and zincdeficiency. However after further research of the classes, cerosopora and gray leaf spot are found to be the same disease, hence another dataset is created in which the data in the two classes are combined. This results in the new dataset having 7 classes; blight, common rust, gray leaf spot, healthy, fallarmyworm, herbicideburn, and zincdeficiency. Below is the explanation of the classes except those already mentioned above.

1 Fallarmyworm

Fallarmyworm larvae feed on grasses and are one of the pests to maize [27]. The leaves of the maize may appear to have small holes and "window pane" due to its feeding. Large larvae may also consume a large amount of leaf tissue causing the appearance to be ragged and cause extensive defoliation [28]. An example of the Fallarmyworm is shown in Fig. 5.

2 Herbicideburn

Improper applications of herbicide or environmental conditions after applying herbicide may cause injury to maize [29]. Different types of herbicides may cause different types of symptoms in plants. The clas-sification by the Weed Science Society of America (WSSA) outlines various symptoms across different groups of plant diseases. WSSA Groups 1 to 22 present specific symptoms such as chlorosis in new leaves, stunted growth, tissue death, yellowing, necrotic spotting, mal-formed leaves, and water-soaked appearance, among others. These diverse groups highlight distinct symptoms observed in various plant diseases as categorized by the WSSA. An image will of Herbicideburn is shown in Fig. 6.

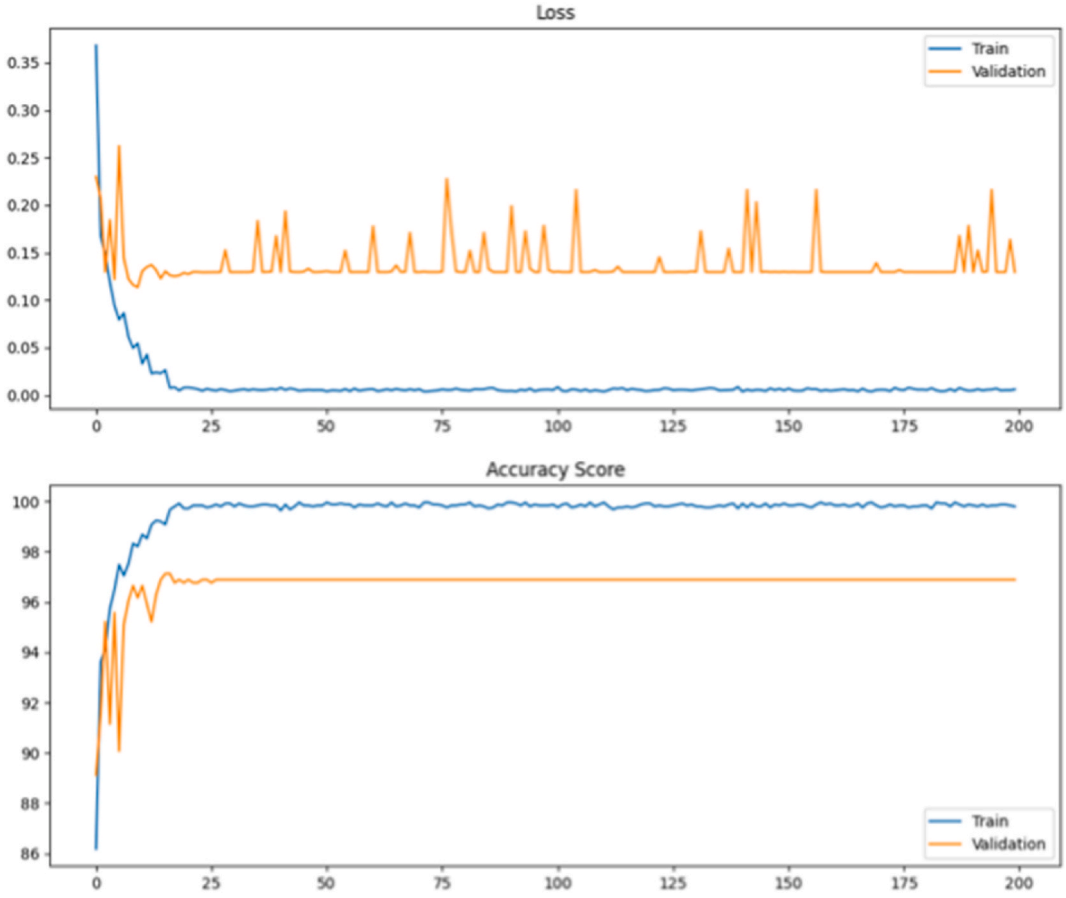


Fig. 14. VGG16 with first dataset.

3 Zincdeficiency

Zinc plays an important role in plant growth, including maize [30]. A deficiency (Fig. 7) can cause broad bands of tissue with white or yellowish-white color to appear on each side of the leaf midrib, but not until the leaf tip. These bands may turn brown or bronze, yet the midrib and outer edges maintain their green color [31]. Zinc-deficient maize may also stunt growth.

3.2. Preprocessing data

Before performing training and testing on the available dataset, data preprocessing will be conducted. To craft an intelligent system, it is crucial to effectively preprocess the images. This involves removing the back-ground, and pectoral muscle, and adding noise, as well as applying image enhancements [32]. However, there has been limited research conducted on techniques for enhancing images during the pre-processing phase [33]. In this research, the data will be read and then divided into three parts: training, testing, and validation. The percentage allocation for each division is 60 % for training data, 20 % for validation data, and 20 % for testing data. The next step is data transformation, which includes operations such as rotating, resizing, applying Gaussian blur, converting to tensors, and normalization. All images are resized to 256x256 pixels.

3.3. Structure model

In this study, the transfer learning method will be used. Utilizing transfer learning offers advantages such as reduced training time, decreased generalization error, and lower computational costs when constructing a Deep Learning (DL) model [34]. Transfer learning involves utilizing a pre-trained model that has been employed on a larger dataset and applying it to a smaller dataset. This approach of transfer learning has recently demonstrated notable success across various sectors, such as medical image classification, manufacturing, and luggage screening [35]. The transfer learning models used in this study are the pre-trained ResNet18, VGG16, and EfficientNet-B0 models.

The first step is defining a CNN model with two convolutional layers, followed by three fully connected layers. The model takes an image as input and passes it through the layers to extract features and make predictions. During training, the model's parameters are

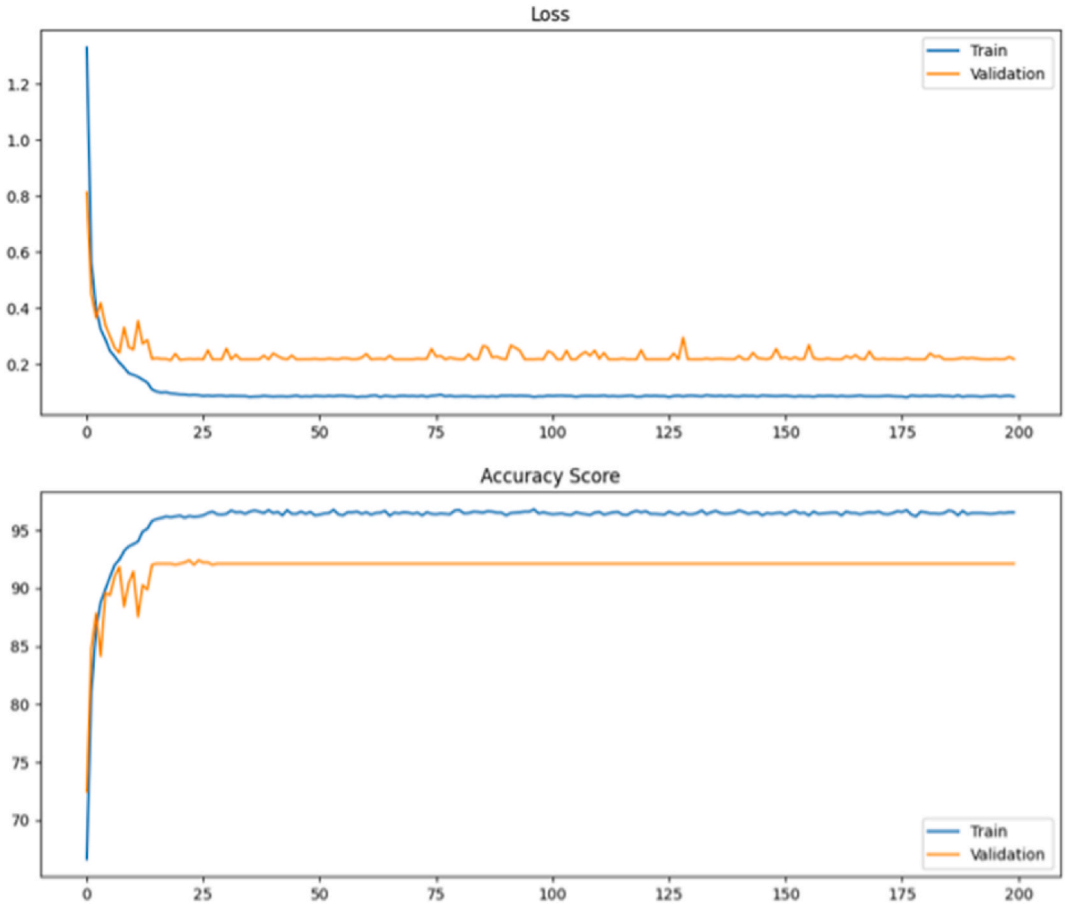


Fig. 15. VGG16 with second dataset.

adjusted to minimize the classification error, allowing it to accurately classify images into one of four classes.

ResNet-18 is a CNN that addresses the challenges associated with image classification, particularly the issue of vanishing gradients prevalent in deep networks. This architecture innovatively employs residual blocks, which facilitate the bypassing of layers during training, thereby ensuring a robust flow of information throughout the network. These residual connections represent the fundamental advancement of ResNet-18, enabling the training of deeper networks while enhancing their performance on complex tasks.

The ResNet-18 architecture (Fig. 9) comprises multiple convolutional layers designed to extract hierarchical features from input images, supplemented by residual blocks that preserve these features across the network's layers. The architecture culminates in fully connected layers, which classify the images into predefined categories.

In Fig. 8, the filters and feature maps are displayed for three specific layers of ResNet18, a deep convolutional neural network architecture. Filters in convolutional networks are responsible for detecting various features in the input images, such as edges, textures, and patterns. As the data passes through successive layers, these filters become more complex, capturing higher-level abstract features.

In the earlier layers (e.g., layer 1), the network tends to capture simple features like edges or corners. These are the building blocks for more sophisticated representations. As the network deepens, such as in layer 4, the filters focus on more abstract concepts like shapes or textures, which are critical for recognizing complex objects. The feature maps represent how the image is transformed at each stage, with each map showing the output of a particular filter.

These visualizations provide an important insight into how neural networks like ResNet18 progressively extract and interpret features from input data, contributing to their ability to perform tasks such as image classification with high accuracy.

A notable advantage of ResNet-18 lies in its capacity to sustain high accuracy even as the network depth increases, owing to the incorporation of residual connections. This capability is particularly advantageous in image classification tasks, where the ability to capture intricate details and patterns across numerous layers is paramount. The efficiency and effectiveness of ResNet-18 have rendered it a preferred choice in diverse image processing applications, including facial expression recognition, where it has demonstrated remarkable accuracy in classifying emotions from images.

VGG16 is a CNN model trained on roughly one million images sourced from the ImageNet database [37]. VGG16, in particular, is known for its depth, as it consists of 16 layers, including 13 convolutional layers and 3 fully connected layers [38]. VGG16 consists of

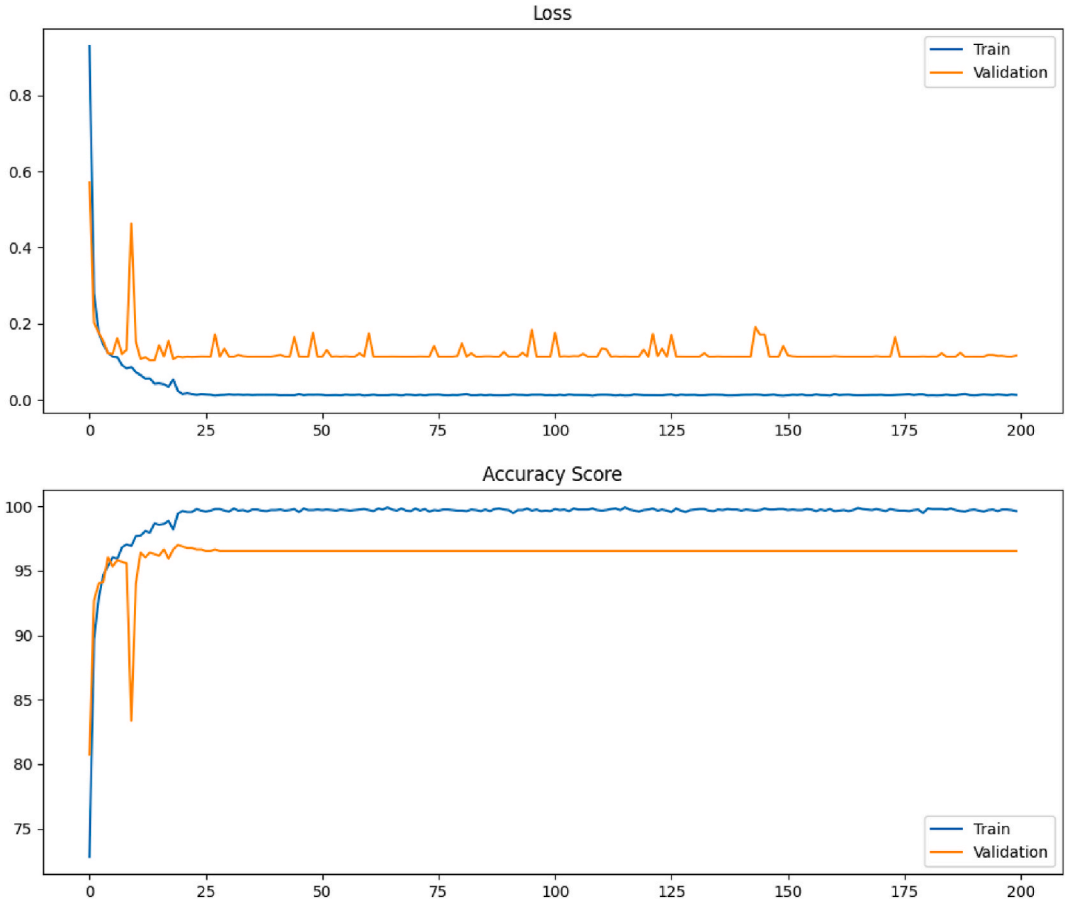


Fig. 16. EfficientNet with first dataset.

13 convolutional layers [39]. These layers perform convolution operations on the input image to extract features. The convolutional layers have small 3x3 filters and are often followed by rectified linear unit (ReLU) activation functions [40]. VGG16 has three fully connected layers towards the end of the network [41]. These layers make final predictions based on the extracted features [42]. The architecture of VGG16 will be shown in Fig. 10.

EfficientNet-B0 is the base model in the EfficientNet family, known for its efficiency and effectiveness in computer vision tasks. It represents the simplest and smallest variant of the EfficientNet architecture, with fewer layers and computational requirements compared to the larger models in the family [44]. The convolutional layers use small 3x3 filters and are followed by rectified linear unit (ReLU) activation functions to introduce non-linearity [45]. There are also inverted residual blocks, a global average pooling layer, and also fully connected layers. The architecture of EfficientNet-b0 will be shown in Fig. 11.

3.4. Freezing layer

Freezing a layer is a technique related to controlling the way weights are updated. When a layer is frozen, the weights in that layer will not be further modified or updated. The purpose of this technique is to reduce computa-tional time during training without compromising its accuracy results [47]. When aiming to alter the weights within a layer, completely avoiding the backward pass process can expedite the execution time. For instance, if half of the model is frozen and trained, it would take approximately half the time needed compared to training a fully unfrozen model. However, the model still requires training. Freezing layers too early may result in less accurate predictions [48].

3.5. Weighted Cross Entropy Loss

To address the class imbalance in the dataset, the approach utilized involves assigning weights to the loss function formula based on the number of samples in each class.

According to research conducted by Ben Naceur et al. (2020) [49], the formula applied to compute the weighted loss using weighted cross-entropy loss can be formulated in the following equation:

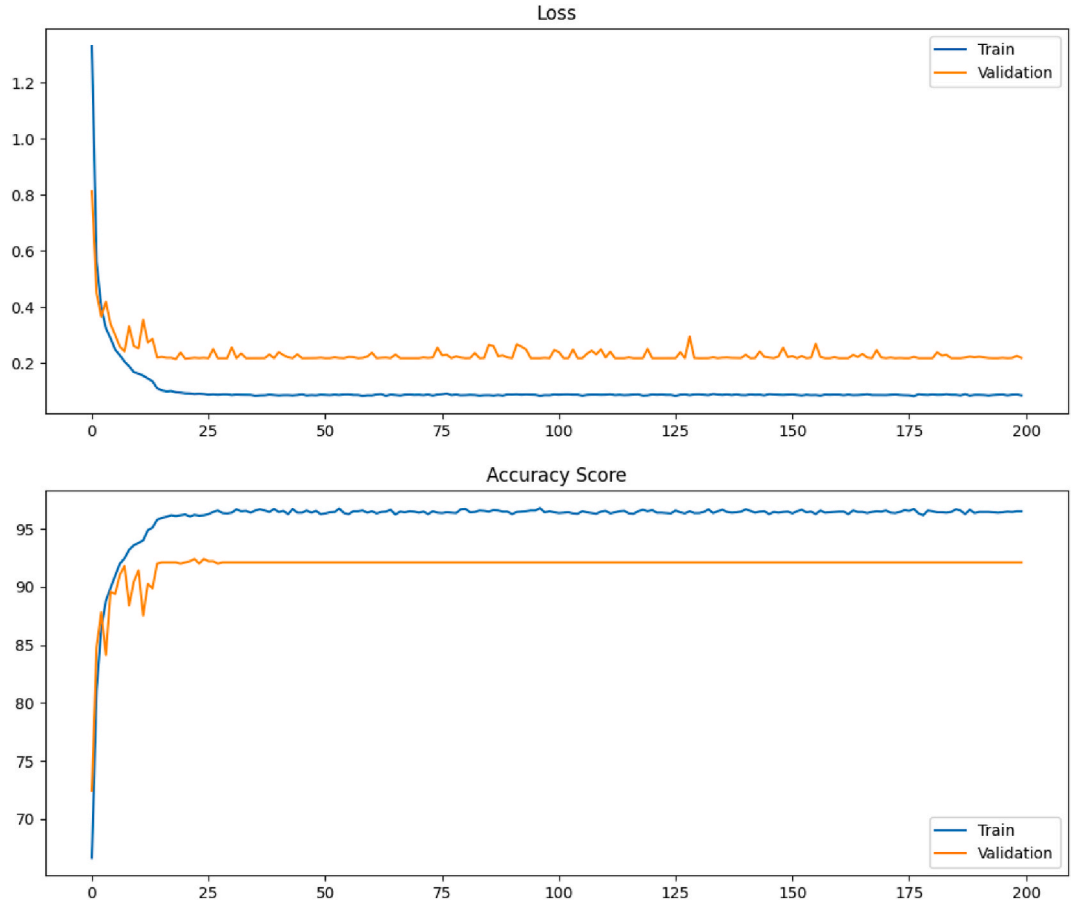


Fig. 17. EfficientNet with second dataset.

Table 1
Optimizer testing on the first dataset.

Optimizer Name	Result
SGD	96.908
Adam	83.234
RMSProp	91.914

$$p_{(i)} = \frac{e^{x(i)}}{\sum_{k=1}^{N} e^{x(k)}} \tag{1}$$

$$\sum_{i=1}^{N} W_{p(i)} y_{p(i)} \, log(p_{(i)}) \tag{2}$$

Equation (1) introduces equations for the Softmax function and equation (2) introduces Weighted Cross-Entropy Loss. Here, k is the number of classes, $y_{p(i)}$ is the i_{th} element of normalized ground truth vector, and $p(i)$ is the i_{th} element of estimated vector for class i. $W_{p(i)}$ (equation (3)) represents the specific weight assigned to the class i.

$$W_{p(i)} = 1 - \frac{n_i}{N} \tag{3}$$

In Equation (3), n_i represents the number of samples in class i and N is the total number of samples across all classes.

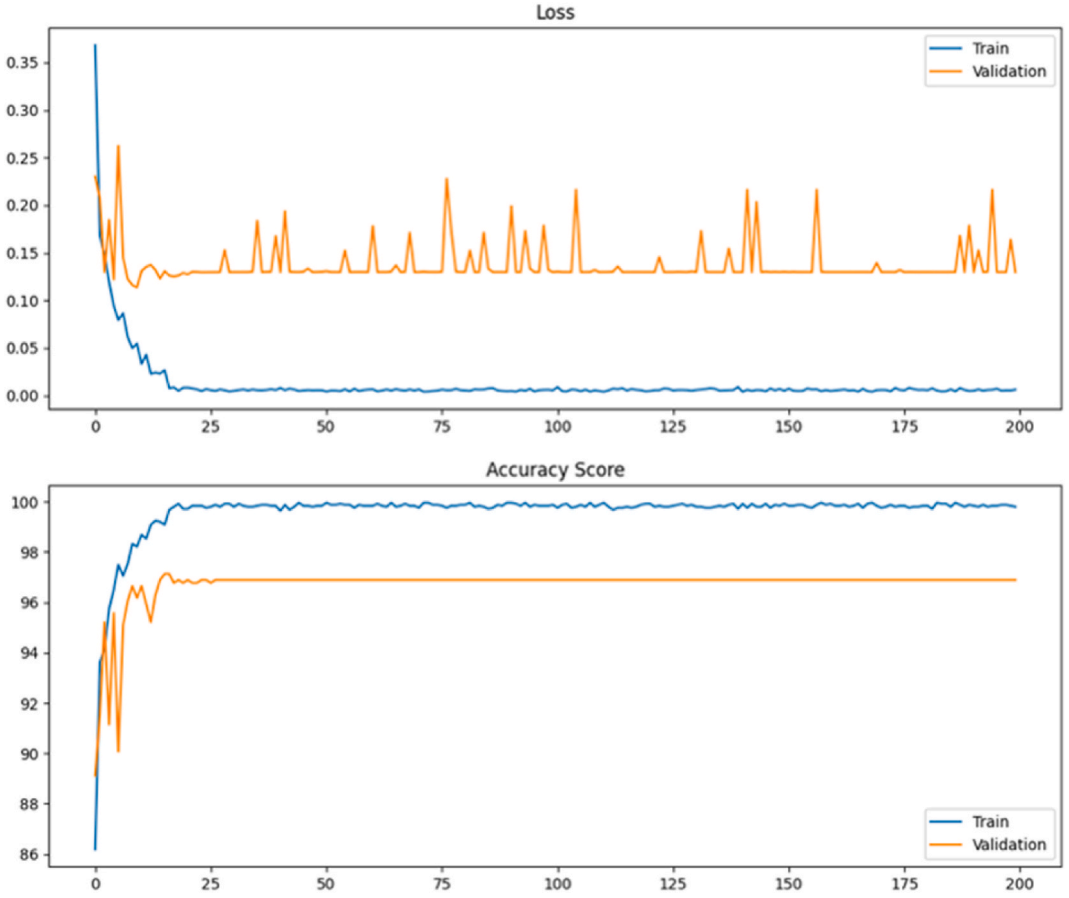


Fig. 18. SGD Optimizer on the first dataset.

3.6. Testing model

Testing will be conducted to assess the performance of the CNN-designed model, encompassing both training and testing phases. During the training phase, the CNN model will be evaluated using pre-existing training data, comprising 2511 and 3089 samples from two distinct datasets. These datasets will be partitioned into 60 % for training, 20 % for validation, and 20 % for testing. Upon completion of the training phase, the process will transition to the testing stage, involving 811 and 1038 images as the testing data. In this testing phase, different images from those employed in training will be utilized to gauge the accuracy of the model under assessment.

3.7. Materials and methods

The CNNs were trained using the following parameters: a learning rate of 0.001 which adaptively decreases, epochs of 200, and a batch size of 16. These hyper-parameters were carefully optimized through experiments to ensure the model's optimal performance on the dataset.

The training was conducted on a computer with configuration of RTX 3060 12 GB of VRAM and CPU of AMD Ryzen 5 5600 6-Core system. The authors utilized Python along with libraries such as PyTorch to implement and train the CNN models.

4. Experiments

4.1. Architecture testing

In this testing phase, testing will be conducted on the architectures to determine the effect of parameters and to find the best architecture. The architectures to be tested include ResNet-18, AlexNet, and VGG16. Each architecture is tested using the same number of iterations, which is 50 iterations.

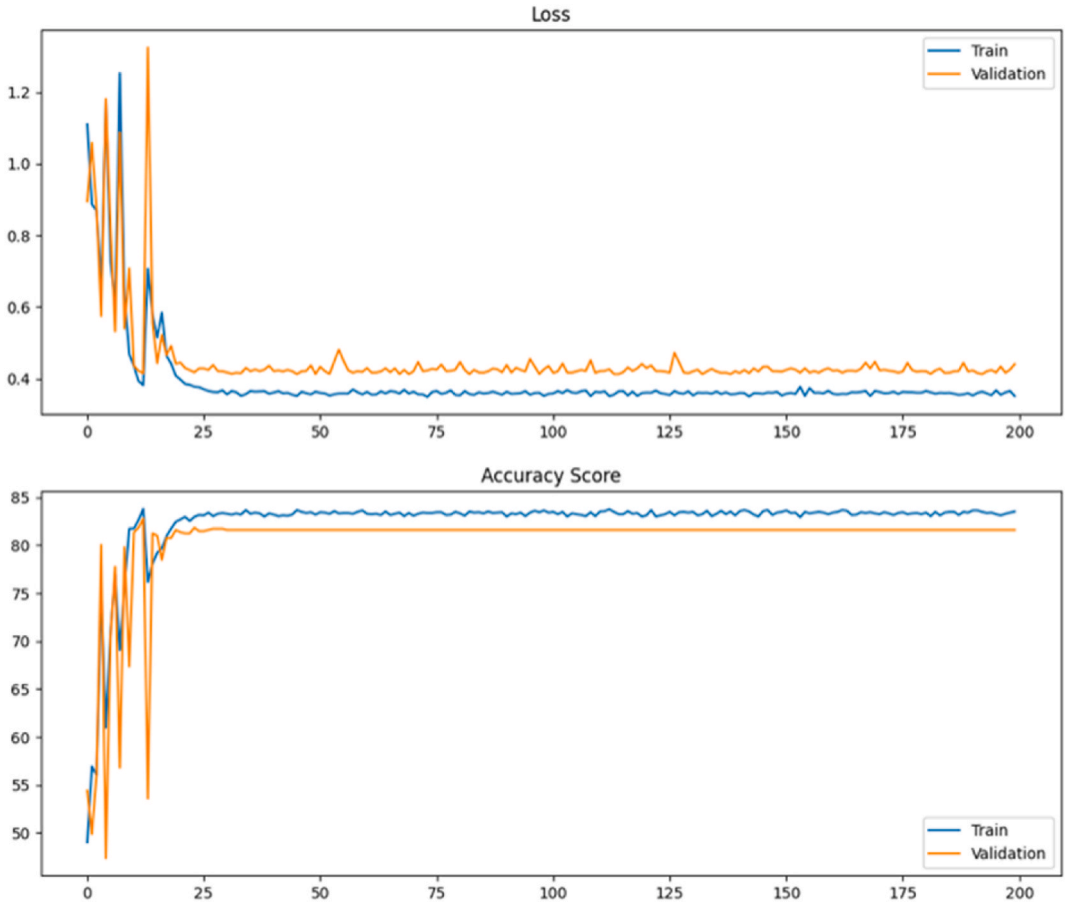


Fig. 19. Adam Optimizer on first dataset.

4.1.1. *ResNet-18*

In this testing phase, testing will be conducted using the ResNet-18 architecture to assess the impact of this architecture on accuracy in this particular test. The results of testing with the ResNet-18 architecture can be seen in Figs. 12 and 13.

The first dataset achieved an accuracy of 95.838 %. Training reached convergence after completing the initial 25 epochs.

The second dataset achieved 92.300 %, which is better than the first dataset. The second dataset reached convergence after also completing the first 25 epochs. It has a greater distance between its training and validation compared to the experiment conducted on the first dataset.

4.1.2. *VGG16*

In this testing phase, testing will be carried out using the VGG16 architecture to understand the influence of this architecture on the accuracy of this particular test. The results of testing with the VGG16 architecture can be seen in Figs. 14 and 15.

It can be seen in Fig. 37 that the first dataset achieved an accuracy of 96.908 %. Training reached convergence before completing the initial 25 epochs.

As seen in Fig. 15, the second dataset has a better graph than the first dataset's training because the distance of the validation and training is closer than the first dataset's graph. Even having a better graph, the accuracy result is lower (92.011 %) than the first dataset (96.908 %).

4.1.3. *EfficientNet-b0*

In this testing phase, testing will be conducted using the EfficientNet-b0 architecture to assess the influence of this architecture on accuracy in this particular test. The results of testing with the EfficientNet architecture can be seen in Figs. 15 and 16.

Fig. 16 shows that the first dataset reached convergence after completing the initial 25 epochs. The training on the first dataset achieved an accuracy of 95.841 %.

Fig. 17 shows that the second dataset has a better graph than the first dataset's training because the training on the second dataset does not significantly differ from the first dataset. Even with a better graph, the accuracy result is 92.974 %, which is lower than the first dataset.

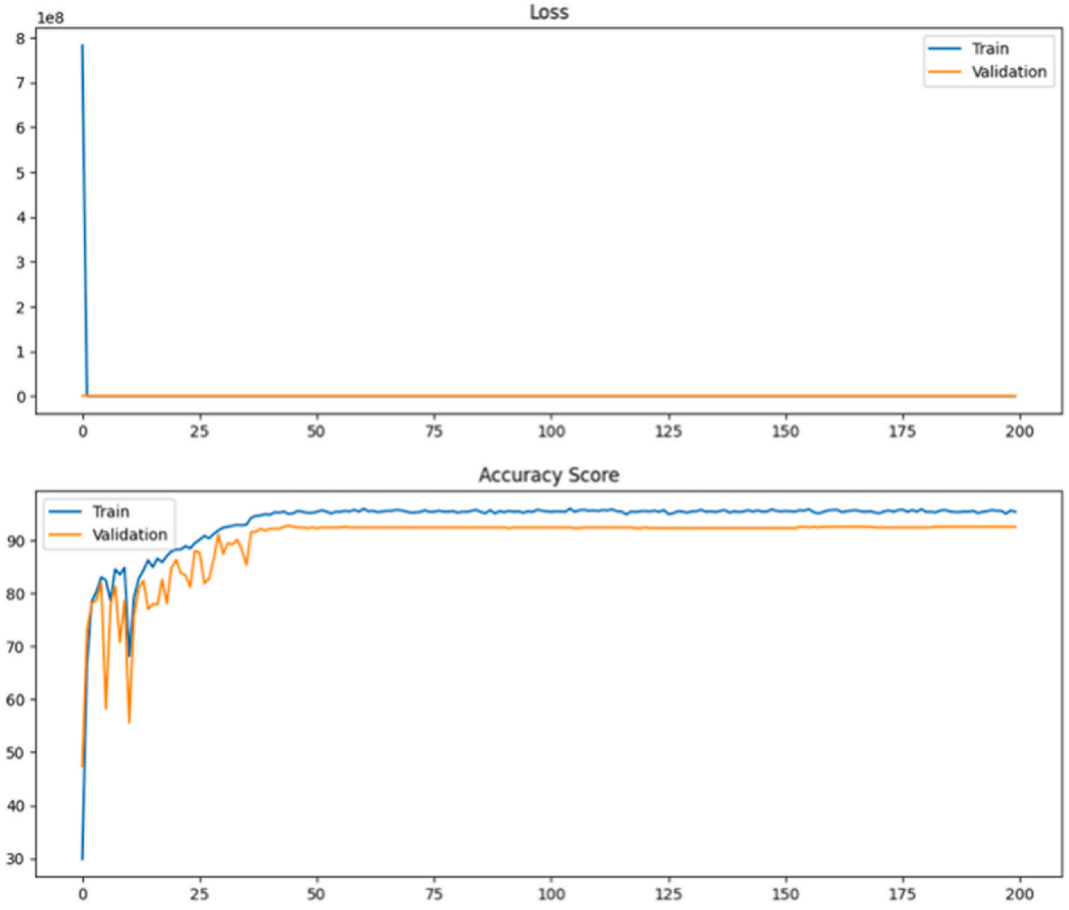


Fig. 20. RMSProp Optimizer on first dataset.

Table 2
Optimizer testing on second dataset.

Optimizer Name	Result
SGD	92.974 %
Adam	26.121 %
RMSProp	46.198 %

4.2. Optimizer testing

In this testing phase, optimization techniques will be evaluated. For this particular test, the optimization methods being employed are SGD, Adam, and RMSProp.

4.2.1. Optimizer testing on the first dataset

Testing will be conducted for a total of 50 iterations, and the evaluation results, as well as tables and graphs of accuracy and loss, will be compared.

The evaluation results will be presented in Table 1. Since the previous experiments yielded the best results with the VGG16 architecture, this testing will focus exclusively on the VGG16 architecture.

Based on the table above, the SGD optimization method yields significantly higher accuracy than the other two optimization methods. To provide a clearer picture, graphs of accuracy and loss for each optimization method will be presented. The graphs can be seen in Figs. 18–20, respectively.

Fig. 18 shows that the first dataset using SGD reached convergence even before completing the initial 25 epochs on the accuracy score. The loss score has a bigger distance on the train and validation compared to the accuracy score. The training on the first dataset using SGD optimizer achieved an accuracy of 96.908 %.

Fig. 19 shows that the first dataset using the Adam optimizer reached convergence before completing the initial 25 epochs. The

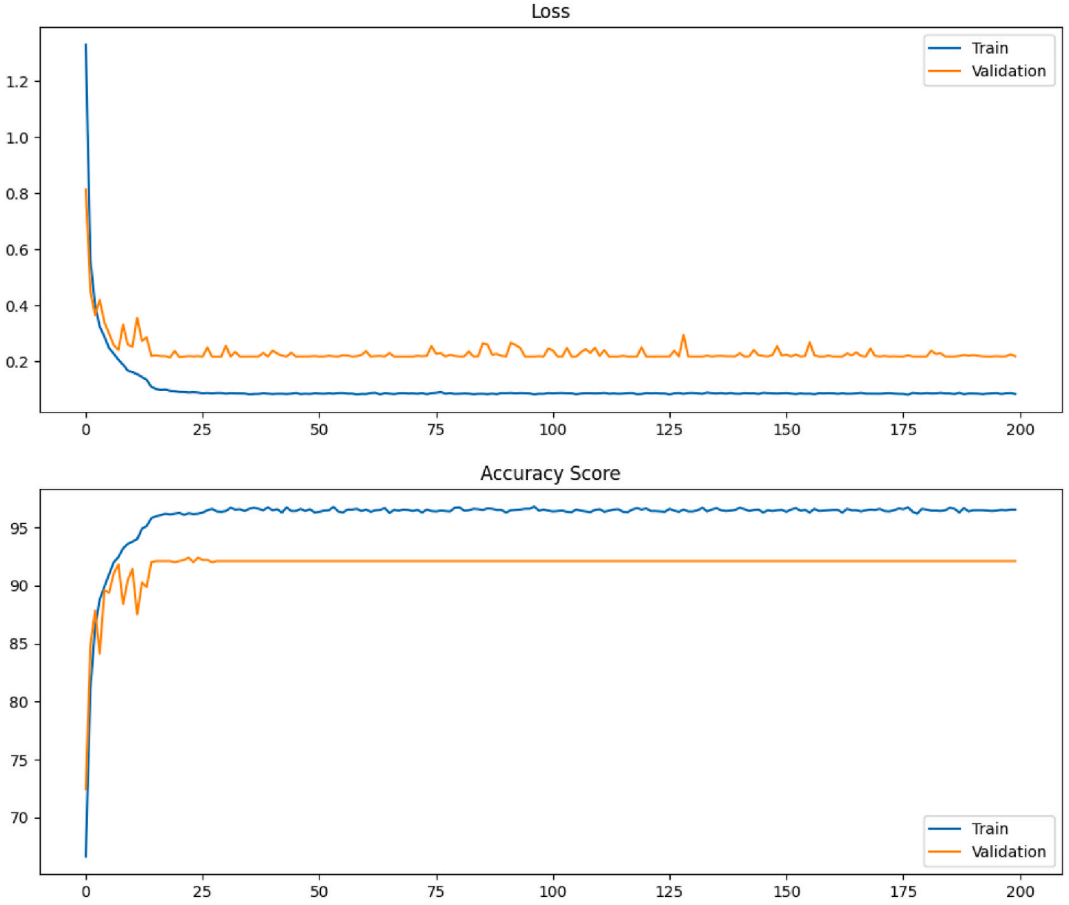


Fig. 21. SGD Optimizer on second dataset.

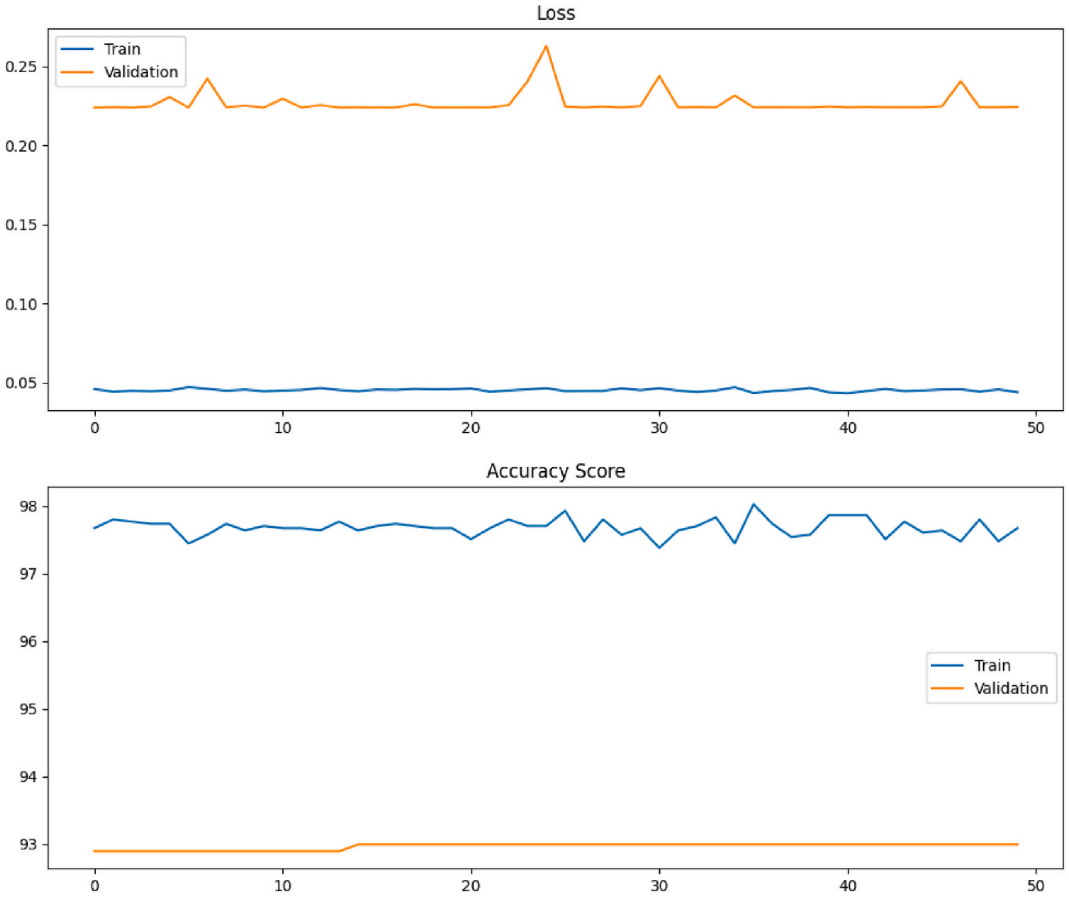


Fig. 22. Adam Optimizer on second dataset.

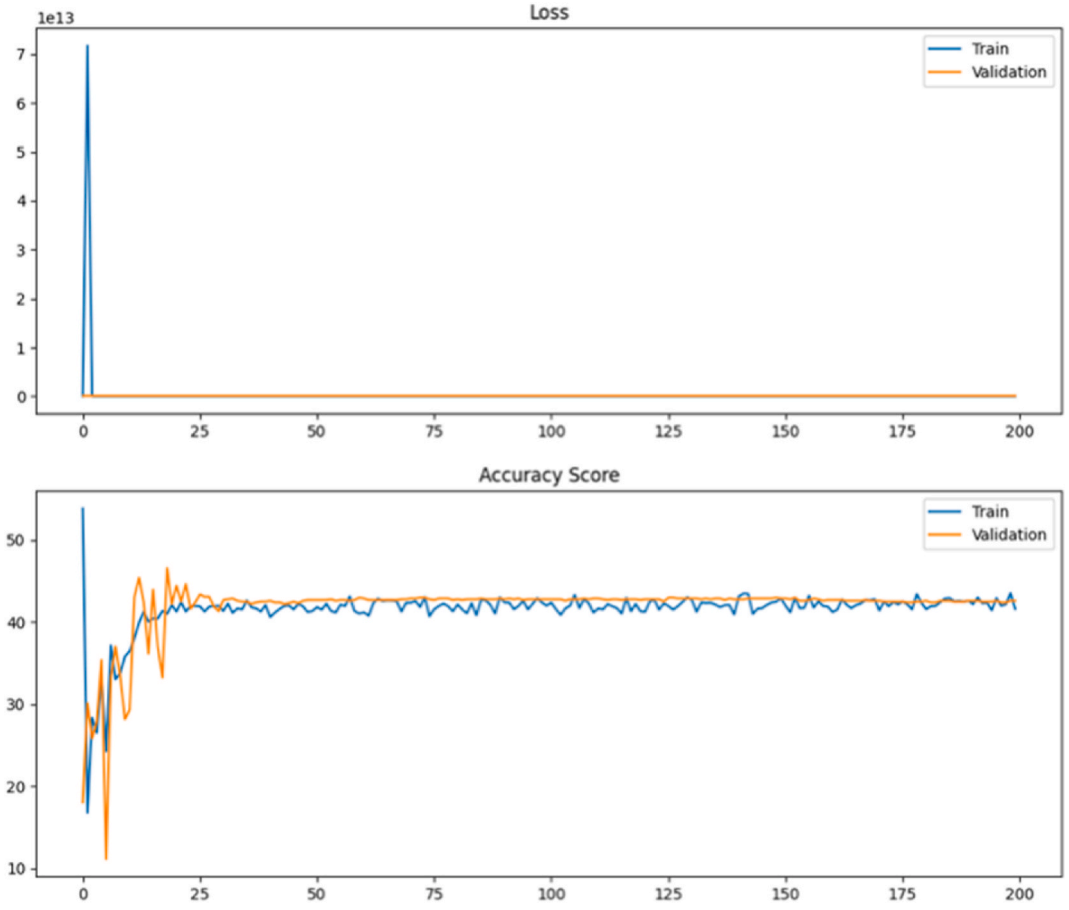


Fig. 23. RMSProp Optimizer on second dataset.

Table 3
Freezing layer on the first dataset.

Frozen Layer	Result
No Frozen Layer	96.617 %
First Layer	96.789 %
Second Layer	97.027 %
Classification Layer	97.146 %

graph looks better than using the SGD Optimizer by looking at the distance of the train and validation on both accuracy and loss score. The training on the first dataset using Adam Optimizer achieved an accuracy of 83.234 %.

Fig. 20 shows that Using RMSProp obtained the best results on the graph, as indicated by the proximity between the training and validation curves. Using RMSProp on the first dataset achieved 91.914 % accuracy.

4.2.2. Optimizer testing on the second dataset

Testing will be conducted for a total of 50 iterations, and the evaluation results, as well as tables and graphs of accuracy and loss, will be examined. The evaluation results will be presented in Table 2. Since the previous experiments yielded the best results with the EfficientNet architecture, this testing will focus exclusively on the EfficientNet architecture.

Based on the table above, it can be observed that the SGD optimization method yields significantly higher accuracy compared to the other two optimization methods. To provide a clearer picture, graphs of accuracy and loss for each optimization method will be presented. The graphs can be seen in Figs. 21–23, respectively.

Fig. 21 shows that the second dataset using SGD reached convergence even before completing the initial 25 epochs on the accuracy score. The graph on the second dataset is better than the first dataset while using SGD optimizer as seen in Fig. 18. The training on the second dataset using SGD optimizer achieved an accuracy of 92.974 %.

Fig. 22 shows that the second dataset using the Adam optimizer reached convergence before completing the initial 25 epochs. The

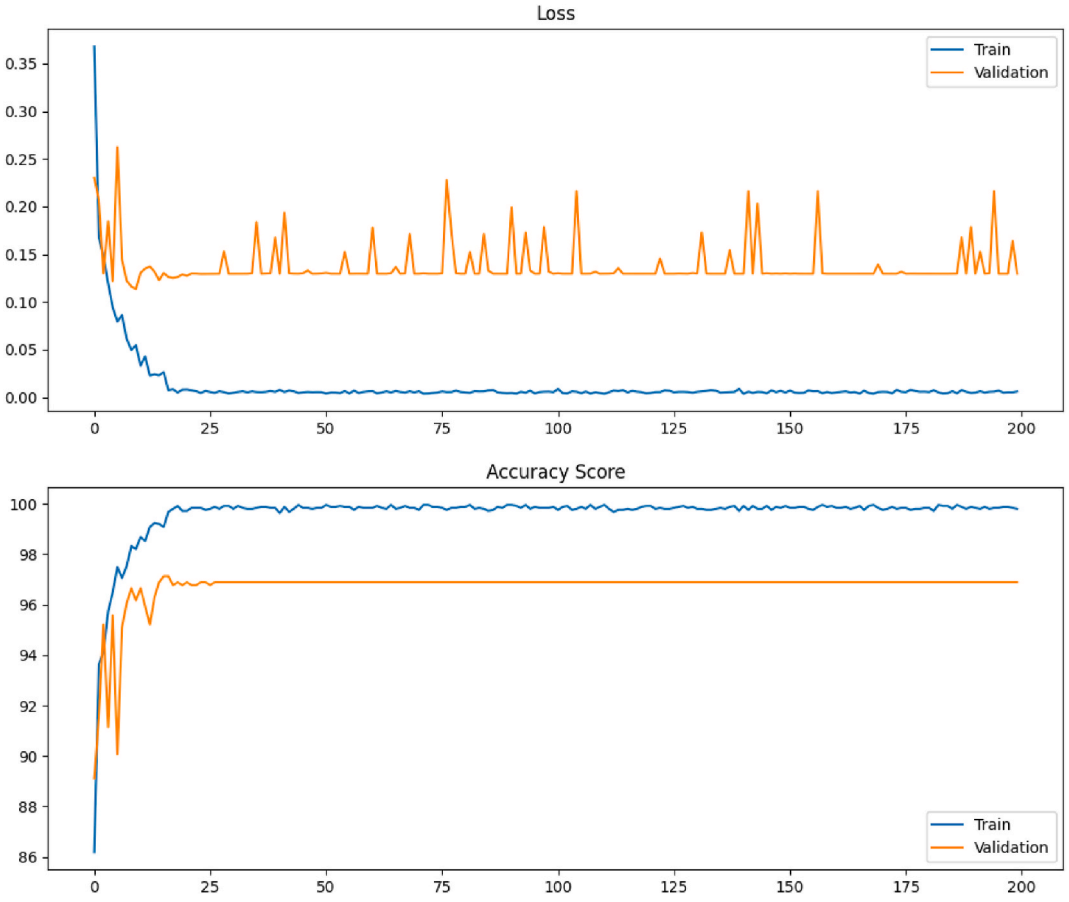


Fig. 24. Without frozen layer on the first dataset.

graph of this training is the worst graph between using SGD and RMSProp. The training on the first dataset using Adam Optimizer achieved an accuracy of 26.371 %.

Fig. 23 shows that Using RMSProp obtained the best results on the graph, as indicated by the proximity between the training and validation curves. Using RMSProp on the second dataset achieved 46.198 % accuracy.

4.3. Testing using frozen layer method

In this testing, we will conduct experiments by applying frozen layers to the first layer, second layer, and classification layer.

4.3.1. Frozen layer testing on the first dataset

This testing is carried out with a total of 200 epochs. In previous experiments, the best results were obtained for the initial dataset using the VGG16 architecture and SGD optimization. The evaluation results will be presented in Table 3 below.

Based on the table above, it can be observed that freezing the layers in the classification layer results in the highest accuracy, which is 97.146 %. To provide a clearer view of the testing results, they will be illustrated through graphs in Figs. 24–26, and Fig. 27, respectively.

It can be seen in Fig. 24 that using no frozen layer on the first dataset reached convergence even before completing the initial 25 epochs on the accuracy score. The graph on the accuracy score shows that the distance between the train and validation is closer and more stable than the graph on the loss score. The training on the first dataset with no frozen layer achieved 96.617 % accuracy.

Fig. 25 shows that freezing the first layer did not reach convergence. The graphic shows that the results are still unstable because the line on the graph continues to fluctuate, going up and down unpredictably. The training when freezing the first layer on the first dataset achieved an accuracy of 96.789 %.

Fig. 26 shows that freezing the second layer did not reach convergence. The graphic looks better than Fig. 25, but it still shows that the results are still unstable because the line on the graph continues to fluctuate. The training when freezing the second layer on the first dataset achieved an accuracy of 97.027 %.

Fig. 27 shows that the freezing classification layer did not reach convergence. The graphic looks the same as Fig. 25, it shows that the results are still unstable because the line on the graph continues to fluctuate, going up and down unpredictably. The training when

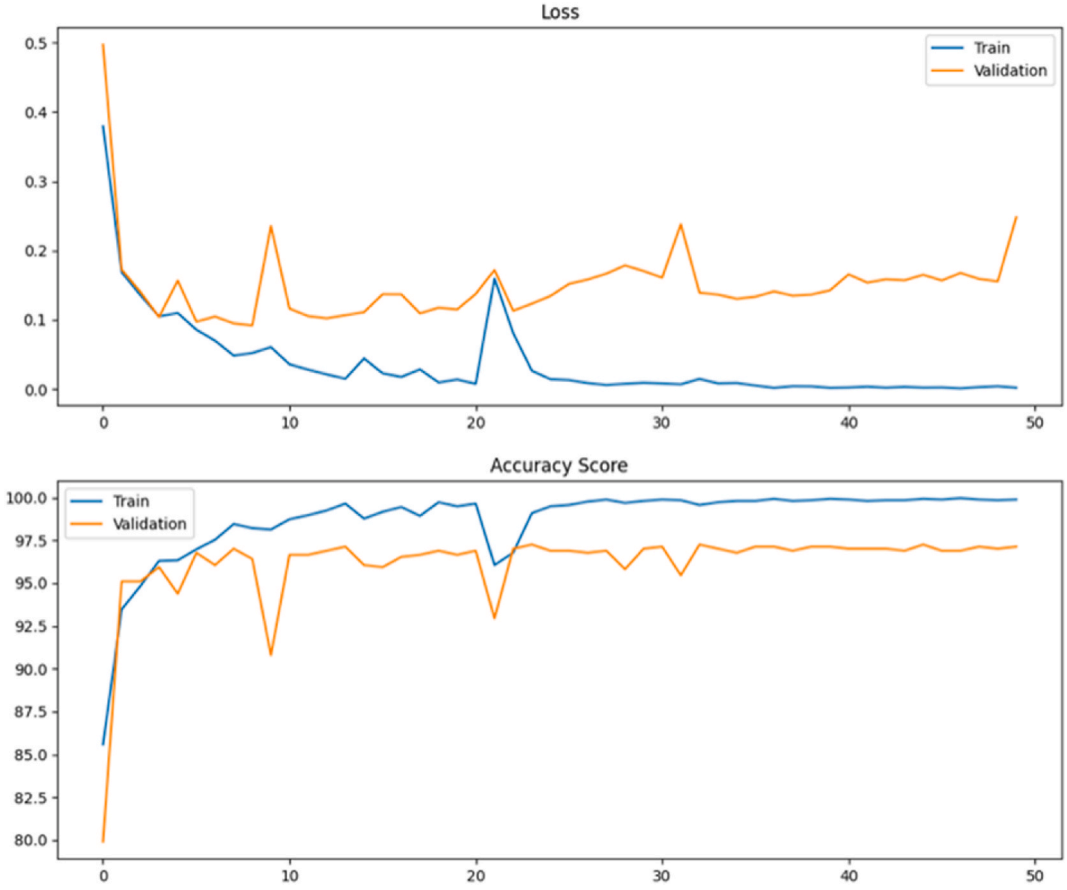


Fig. 25. Freezing first layer on the first dataset.

freezing the classification layer on the first dataset achieved an accuracy of 97.146 %.

It can be seen that training on the first dataset has the best graph while not using a frozen layer (Fig. 24) compared to others. The graph looks more stable if compared to another figure. While not using a frozen layer, the graph converges when the others look uncertain.

4.3.2. Frozen layer testing on the second dataset

In the previous experiments, the best results were obtained for the initial dataset using the EfficientNet architecture and SGD optimization. The evaluation resultsis presented in Table 4 below.

Based on the table above, it is evident that keeping the layers unfrozen results in higher accuracy compared to using frozen layers. To provide a clearer view of the testing results, they will be illustrated through graphs in Figs. 28–30, and Fig. 31, respectively.

It can be seen in Fig. 28 that using no frozen layer on the second dataset reached convergence even before completing the initial 25 epochs on the accuracy score. The graph shows the close distance between the train and validation on both accuracy and loss scores. The training on the second dataset with no frozen layer achieved 92.974 % of the accuracy.

Fig. 29 shows that freezing the first layer on the second dataset reached convergence after 20 epochs. The graphic shows that the results started to be stable after the first 20 epochs. The training when freezing the first layer on the second dataset achieved an accuracy of 91.915 %.

Fig. 30 shows that the freezing second layer on the second dataset reached convergence before 30 epochs. The training when freezing the second layer on the second dataset achieved an accuracy of 91.915 %.

Fig. 31 shows that the freezing classification layer on the second dataset reached convergence after 25 epochs. The graphic looks better than Fig. 28 that it shows a closer distance between train and validation on both accuracy and loss scores. The training when freezing the classification layer on the first dataset achieved an accuracy of 92.107 %.

It can be seen that training on the second dataset has the best graph while using the frozen layer on the classification layer(Fig. 31) because it has the closest distance between the train and validation line compared to another figure. Even though not having the best graphs, the training when not using the frozen layer on the second dataset has the best accuracy score, it achieved 92.974 %.

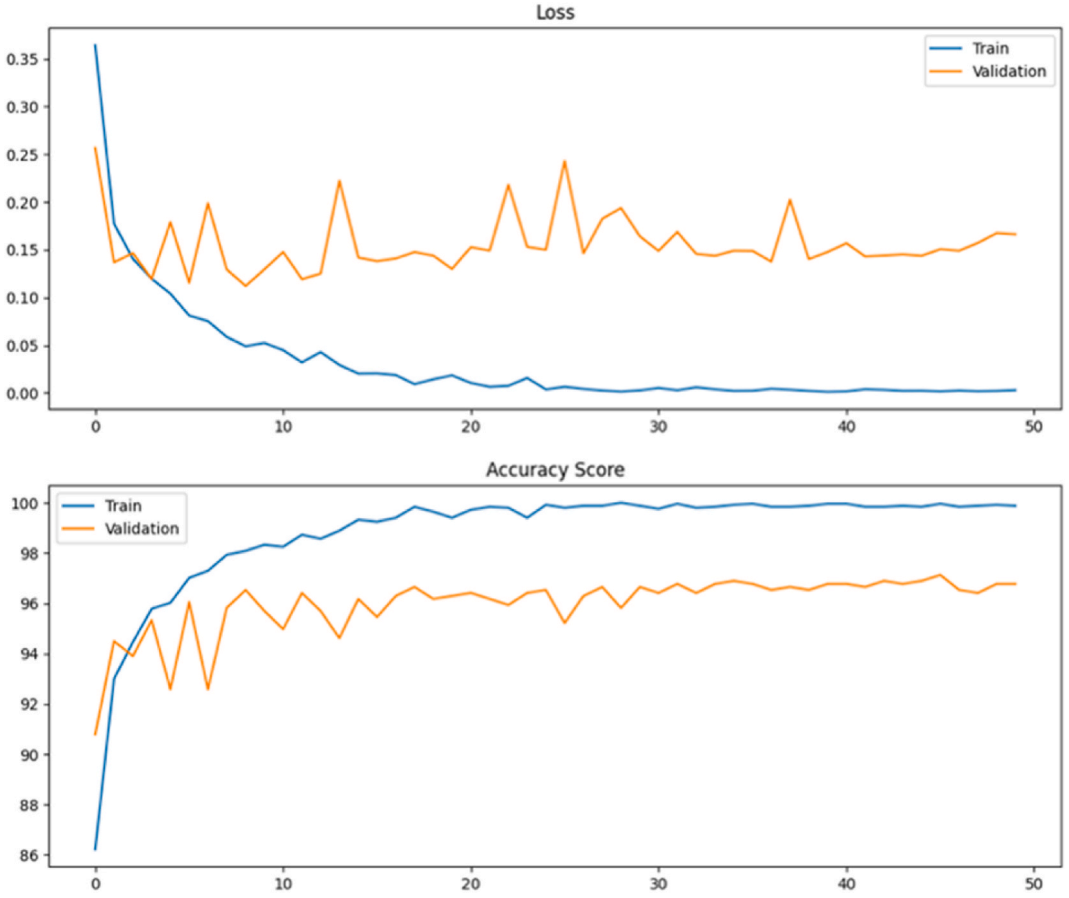


Fig. 26. Freezing second layer on the first dataset.

4.4. Testing using weighted loss method

In this testing, weights are applied to the loss function, or a weighted loss is used. This testing is conducted due to class imbalance, where the number of samples in each class is not balanced in the mixed dataset. In this testing, the data is divided into two sets: one with 7 classes and another with 8 classes. The results of the training is presented in Figs. 32 and 33 below. Meanwhile, the results of training without using weighted loss is shown in Fig. 34.

Fig. 32 shows that using weighted loss on 7-class datasets reached convergence before 20 epochs. The graph shows the stable line on both train and validation. The training when using weighted loss on 7-class datasets achieved an accuracy of 94.798 %

Fig. 33 shows that using weighted loss on 8-class datasets reached convergence after 20 epochs. The graph shows the stable line on both training and validation in terms of accuracy. However, on the loss score, the line still indicates slight changes on both the train and validation lines. The training when using weighted loss on 7-class datasets achieved an accuracy of 92.877 %

Fig. 34 shows that when not using weighted loss on 8-class datasets reached convergence after 25 epochs. The graph shows the stable line on both train and validation. The training when using weighted loss on 7-class datasets achieved an accuracy of 92.204 %

The accuracy results show that the second data with 7-classes achieved the highest accuracy at 94.798 %, followed by the mixed data with 8-classes at 92.877 %, while the training without weighted loss resulted in the lowest accuracy at 92.204 %. The confusion matrix results for all three scenarios are presented to demonstrate the improvement in handling class imbalance, and they can be seen in Figs. 35–37, respectively.

It can be seen in Fig. 34 that all classes have high accuracy on each class. The confusion matrix indicates that the utilization of weighted loss enables the classification of each class with fairly high accuracy. The highest accuracy achieved is 99.62 %.

Fig. 36 shows that using weighted loss on the 8-class dataset could classify better than not using weighted loss which is shown in Fig. 37. Even though still has low accuracy on class number 1, it still classifies better than when not using the weighted loss.

Fig. 37 shows that when not using weighted loss on an imbalanced dataset could produce poor results in some classes. In this case, it can be seen that only 1 picture that correctly classified on class number 1.

Based on the results of the three confusion matrices above, it can be observed that the confusion matrix results for the data with 7 classes are the most favorable compared to the other two scenarios. In this experiment, classes of 6, 7, and 8 had the least data. Before the use of weighted loss, it is evident that the accuracy for these three classes was lower, and it improved after implementing weighted

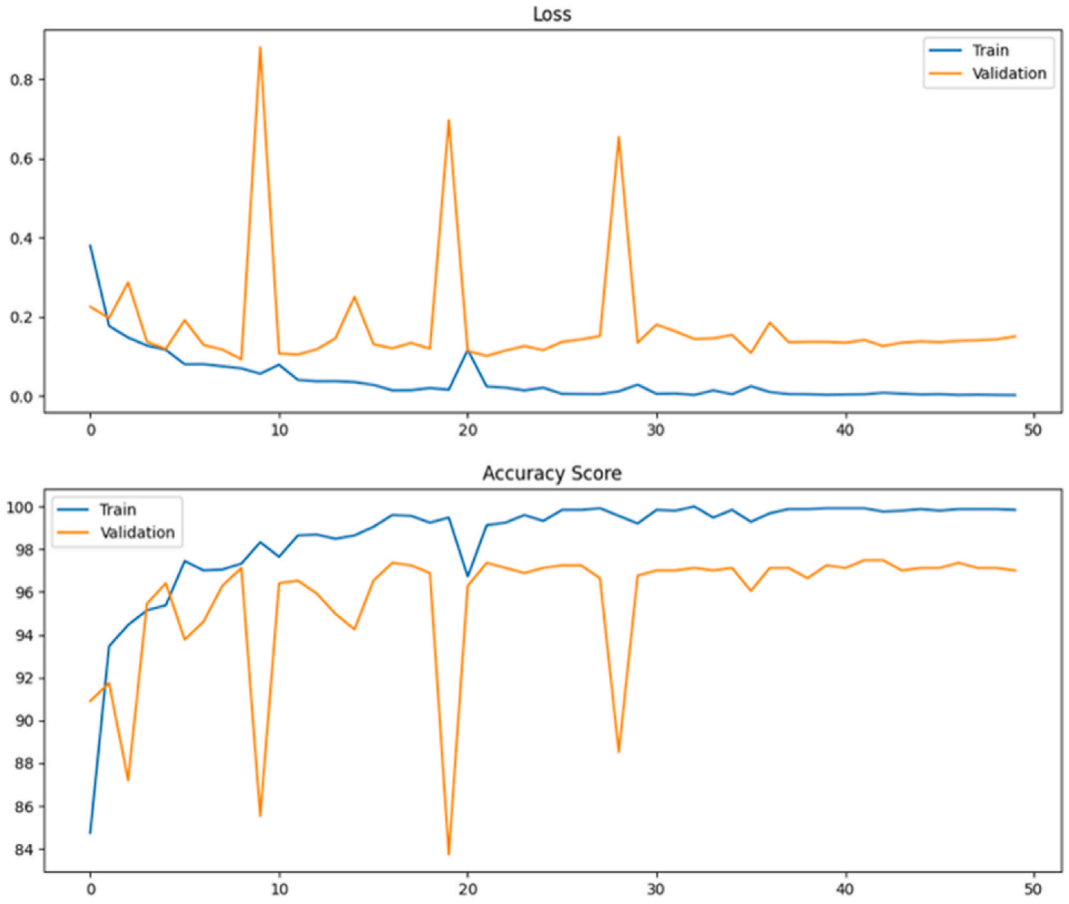


Fig. 27. Freezing classification layer on the first dataset.

Table 4
Freezing layer on second dataset.

Frozen Layer	Result
No Frozen Layer	92.974 %
First Layer	91.915 %
Second Layer	91.915 %
Classification Layer	92.107 %

loss.

4.5. CNN architectures: Structure and Topology consideration

The performance differences among the models can be justified based on the unique features of each CNN architecture as shown in Table 5.

The residual connections in ResNet-18 facilitate training convergence and improve accuracy by addressing vanishing gradient problems. VGG16's depth and simplicity contribute to higher accuracy, but this comes at a computational cost. EfficientNet-b0 stands out due to its compound scaling, which efficiently utilizes parameters for optimal performance across various datasets.

4.6. Optimizers: impact on training stability and convergence

The choice of optimizer has a significant impact on training stability and convergence.

As shown in Table 6, SGD often results in higher accuracy, especially in simpler datasets, as it converges steadily. In contrast, Adam and RMSProp may struggle to converge due to their adaptive learning rates, which can occasionally lead to suboptimal local minima.

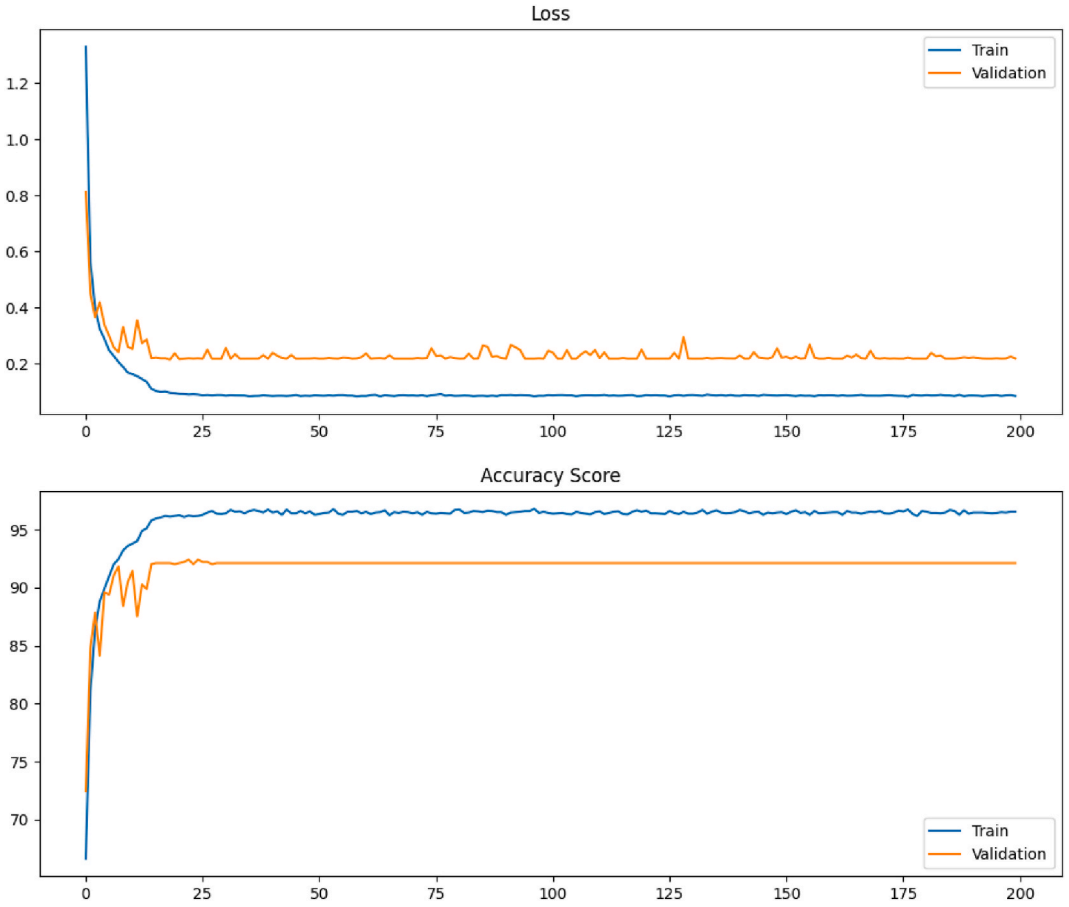


Fig. 28. Without frozen layer on the second dataset.

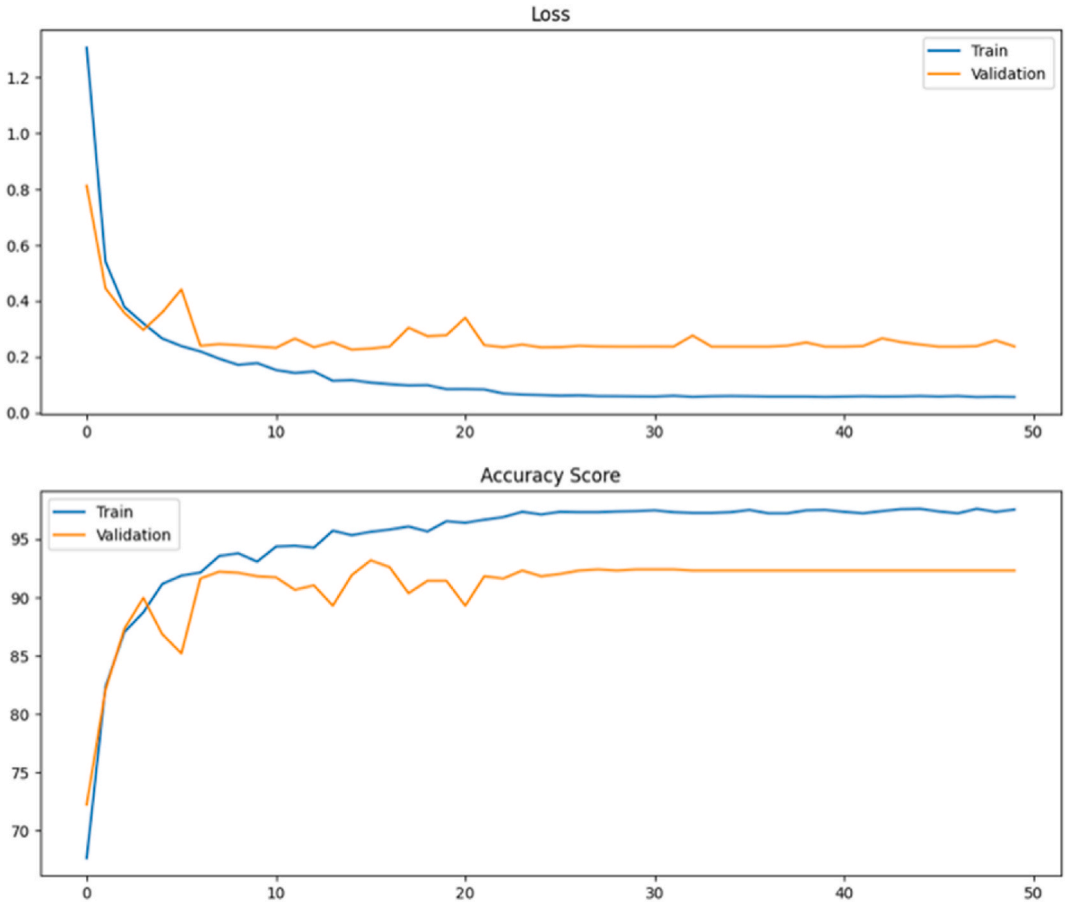


Fig. 29. Freezing first layer on the second dataset.

4.7. Frozen layers: layer-specific freezing effects

The rationale behind freezing different layers is crucial for optimizing performance. Freezing earlier layers typically retains general features, while freezing later layers, such as classification layers, preserves learned task-specific features. This approach can lead to variations in accuracy, depending on the dataset.

4.8. Weighted loss: addressing class imbalance

Applying weighted loss is beneficial in combating class imbalance, particularly in skewed datasets. This method has shown improved performance as evidenced by confusion matrices and accuracy results, highlighting its effectiveness in enhancing model predictions for minority classes.

Holistic analysis of the different methods tested—varying architectures, optimizers, frozen layers, and weighted loss—reveals insights into their effectiveness. Each approach has its advantages, and the choice of method should be guided by specific scenarios. For instance, ResNet-18 may excel in certain tasks due to its architecture, while weighted loss can significantly enhance performance in imbalanced datasets.

4.9. Comparison with previous research

Based on Table 7 above, a comparison can be made between the proposed research with the first dataset and five previous research because using the same amount of class. The proposed research conducted testing with three different types of models, all of which exhibited higher accuracy compared to the 3 previous research except research proposed by Jasrotia et al. (2023) [50] which achieved an accuracy of 96.76 % using its method. However, the highest accuracy was attained when using the VGG16 architecture on the proposed model using the first dataset, reaching 96.908 %.

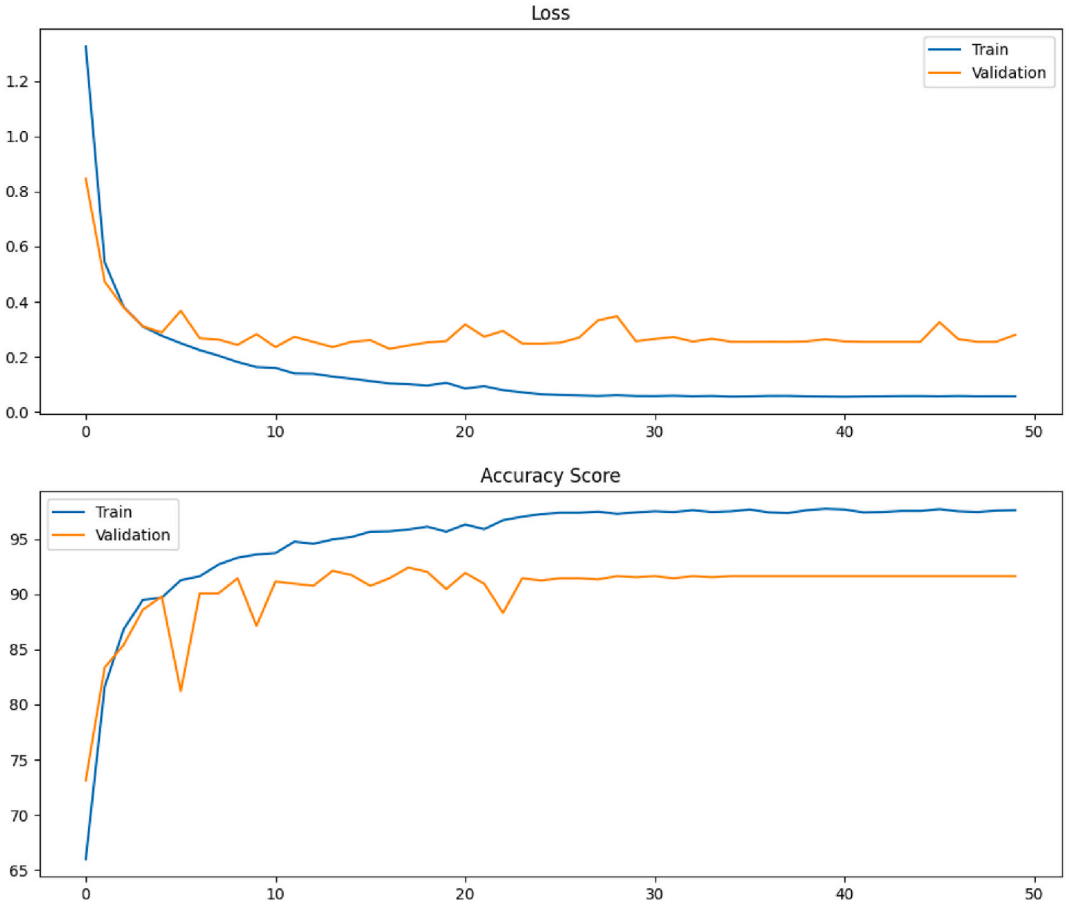


Fig. 30. Freezing second layer on the second dataset.

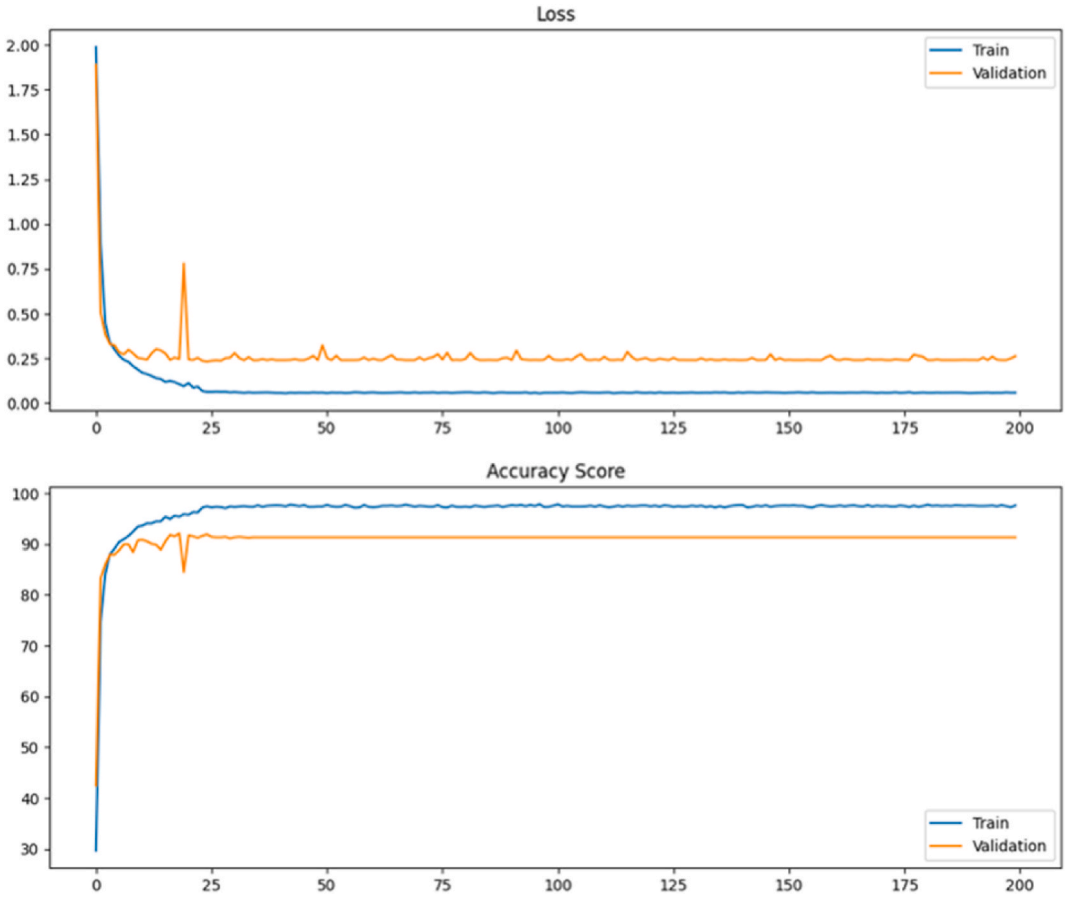


Fig. 31. Freezing Classification layer on the second dataset.

4.10. Analysis

From the evaluations conducted on both the first and second datasets, it is evident that the second dataset contains a larger volume of images and encompasses a greater number of classes. However, this increase in size and diversity also led to an issue of class imbalance within the second dataset.

Initially, in the assessment of the first dataset, the VGG16 pre-trained model exhibited the highest accuracy, achieving an impressive 96.908 %. Conversely, the second dataset encountered an issue where one class was consistently misclassified as another. Further investigation revealed that these two classes shared a common disease cause and displayed similar characteristics. To address this, the second dataset underwent restructuring, condensing these two classes into a singular entity, resulting in a revised dataset of 7 classes.

Despite this refinement, instances of misprediction persisted during testing due to the underlying issue of unbalanced data distribution.

To tackle this concern, the approach involved implementing weighted cross-entropy loss. Following this implementation, significant enhancements were observed in the test results compared to the previous outcomes. The most notable improvement on the revised second dataset was achieved using the EfficientNet-b0 pre-trained model, delivering an accuracy of 92.974 %.

5. Conclusion

Based on the testing results from the research on disease classification in maize plants using the Convolutional Neural Network (CNN) algorithm, several conclusions can be drawn.

Firstly, the CNN method demonstrated exceptional performance in classifying diseases in maize plants, achieving notably high accuracy in both datasets utilized. Specifically, the VGG16 architecture with SGD optimization and a frozen layer in the classification layer yielded the highest accuracy in the first dataset. Conversely, in the second dataset, optimal accuracy was achieved with the EfficientNet-b0 architecture using SGD optimization without a frozen layer.

Secondly, the impact of frozen layers on results was significant in this study. In the first dataset, the use of frozen layers, particularly in the classification layer, notably improved accuracy. However, there were limitations observed, with dataset characteristics

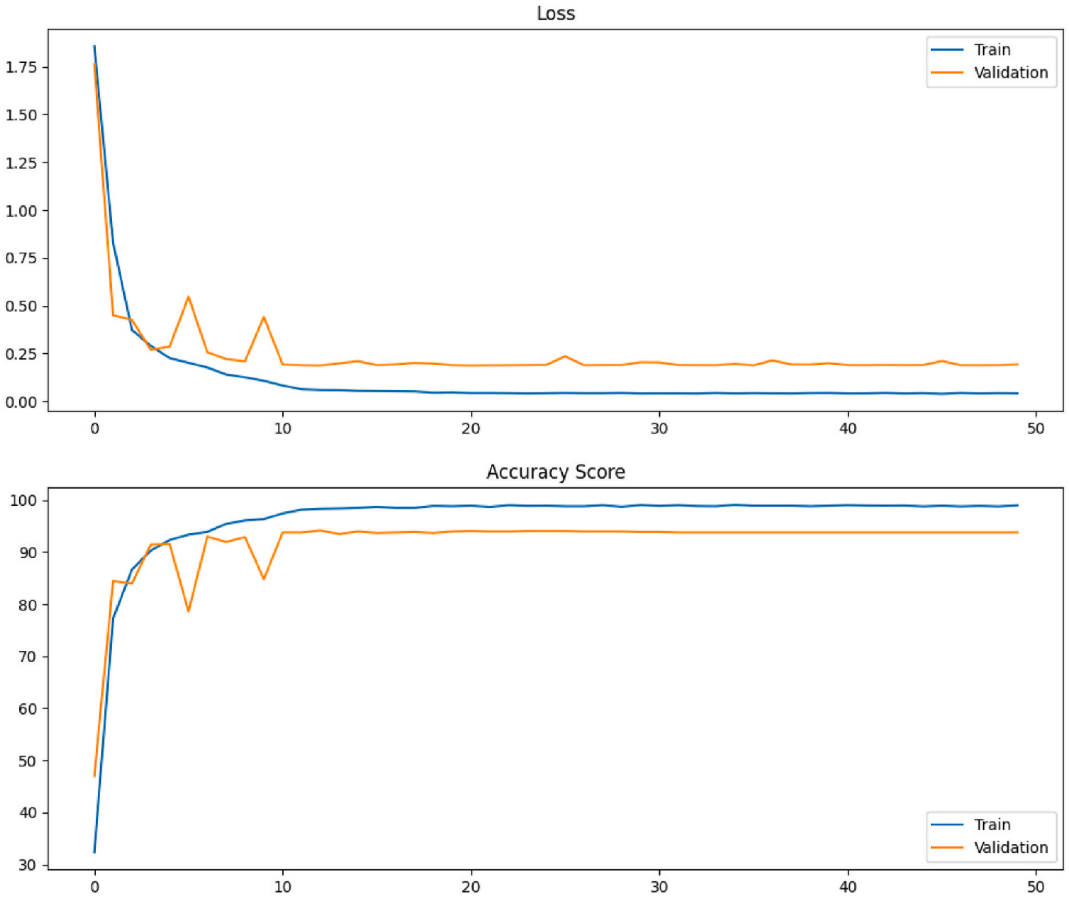


Fig. 32. Weighted loss on the 7 class dataset.

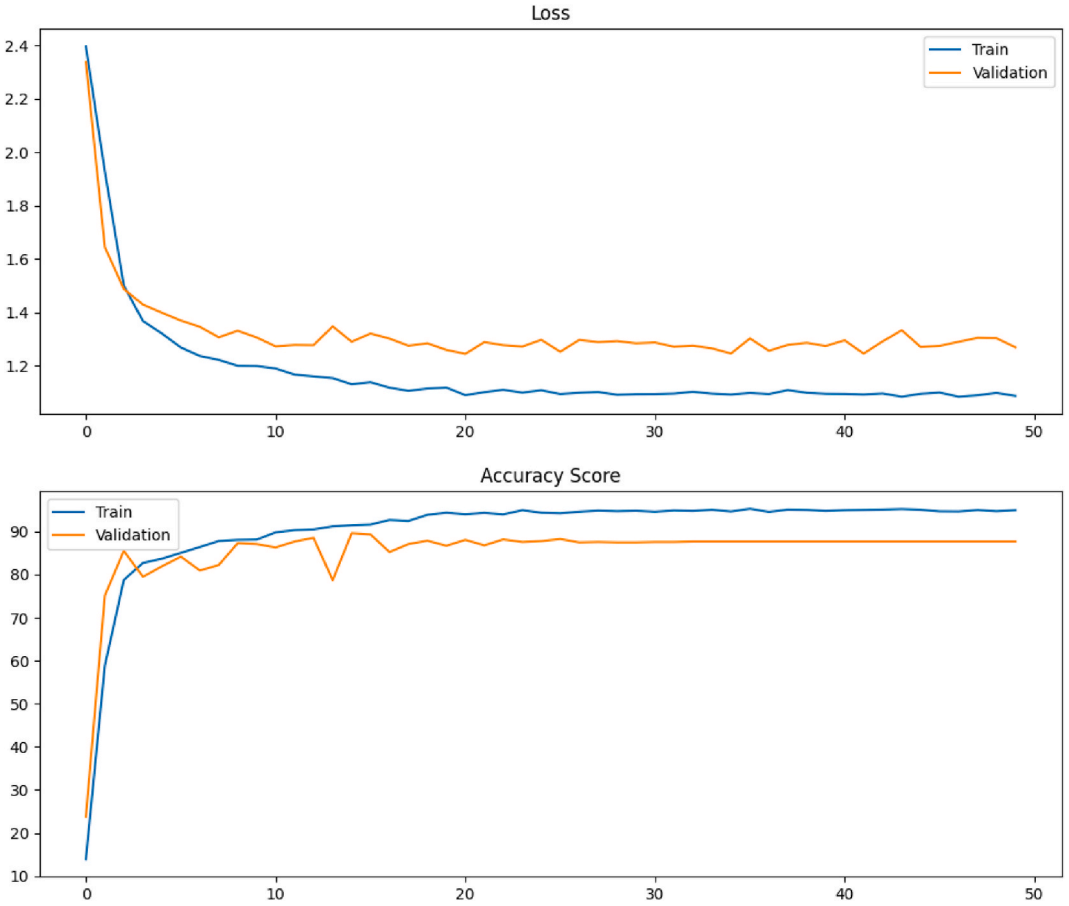


Fig. 33. Weighted loss on the 8 class dataset.

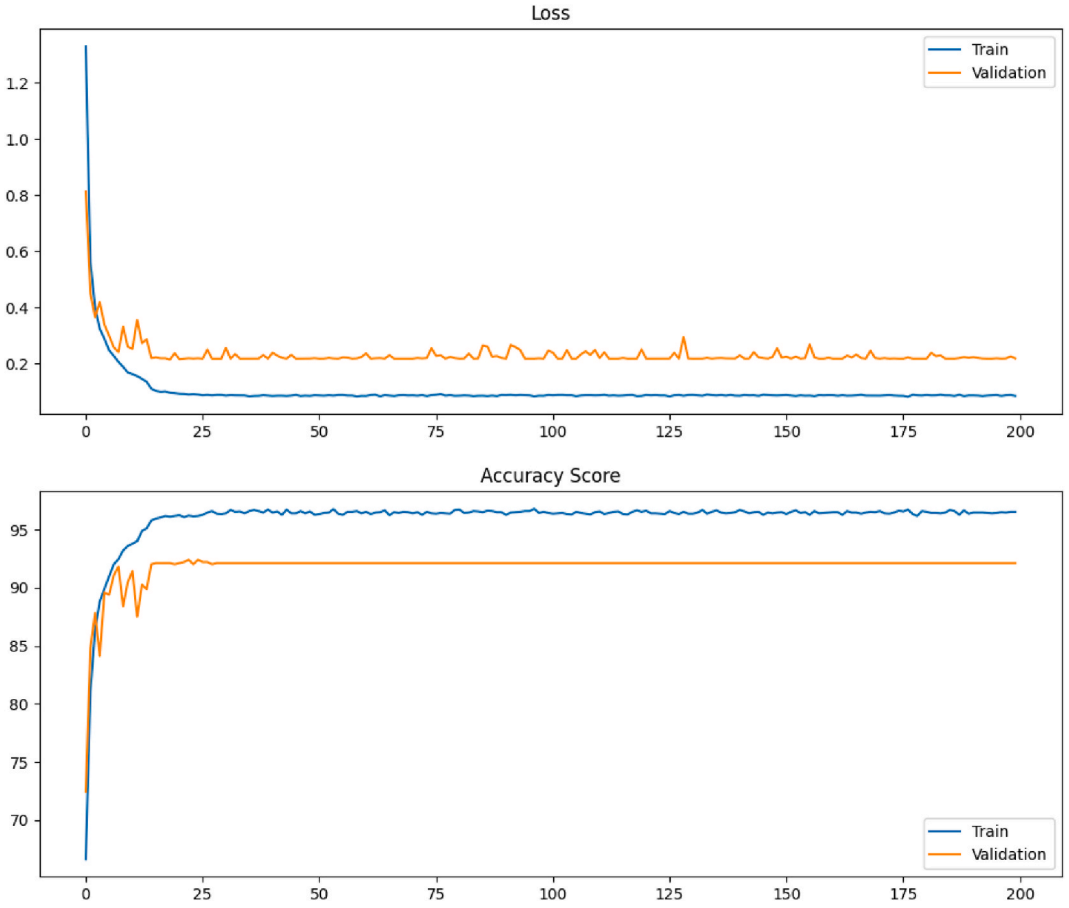


Fig. 34. No weighted loss testing.

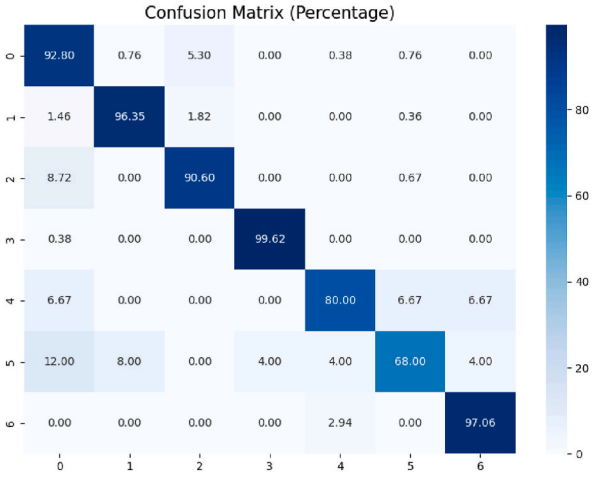


Fig. 35. Confussion Matrix on 7 class dataset.

influencing the effectiveness of frozen layers, as evidenced in the second dataset where the highest accuracy was attained without their use.

Thirdly, the implementation of weighted loss proved effective in addressing class imbalance, notably present in the second dataset. This approach improved the model's ability to classify minority classes more accurately. These three findings serve as key highlights of this research, offering insights into maize disease classification to aid maize plant production.

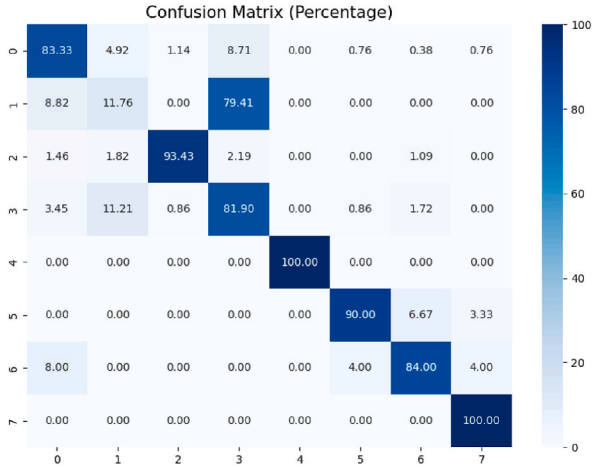


Fig. 36. Confussion Matrix on 8 class dataset.

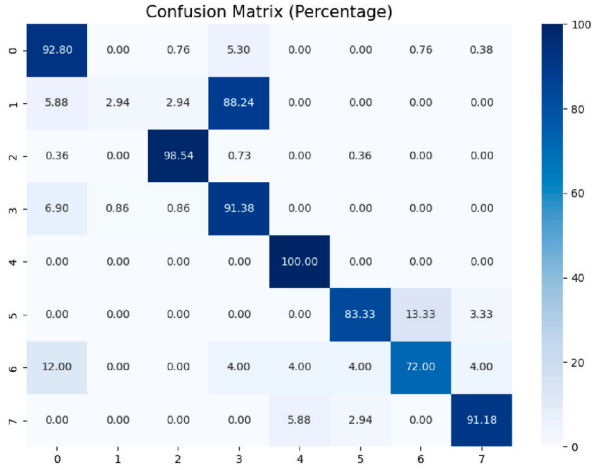


Fig. 37. Confussion matrix without weighted loss.

Table 5
CNN architectures and performance insights.

Model	Key Features	Performance Insights
ResNet-18	Residual connections that mitigate vanishing gradient issues	Effective in simpler tasks; performs well in specific scenarios.
VGG16	Deep architecture with uniform layers contributing to higher accuracy	High accuracy due to depth; potential for overfitting.
EfficientNet-b0	Compound scaling balances depth, width, and resolution	Robust performance; efficient parameter utilization.

Table 6
Impact of optimizers on training.

Optimizer	Characteristics	Performance
SGD	Tends to converge slowly but steadily in simpler datasets.	Higher accuracy in well-behaved datasets
Adam	Adaptive learning rates can lead to suboptimal minima.	Struggles with convergence in certain datasets.
RMSProp	Balances speed and stability, but may show inconsistency.	Variable performance across different scenarios

For future research endeavors, several suggestions are proposed. These include utilizing datasets with a greater diversity of classes to mitigate similarities between classes. Additionally, employing automated feature selection methods could enhance the selection of appropriate training data. Lastly, incorporating state-of-the-art activation functions in future advancements could further improve model performance.

Table 7
Comparison with previous model.

Research	Architecture	Result	Dataset	Total Class
Ubaidillah et al., 2022	Random Forest, Neural Network, Naive Bayes	69.76 % (Random Forest), 74.44 % (Neural Network), 60.25 %(Naive Bayes)	Kaggle	4 (Healthy, Common Rust, Gray Leaf Spot, Blight)
Sandotra et al. (2023)	ResNet50, VGG19, InceptionV3, EfficientNet-b0	78.19 %(ResNet50), 88.54 % (VGG19), 89.95 %(InceptionV3), 92.91 % (EfficientNet-b0)	PlantVillage (kaggle)	4 (Healthy, Common Rust, Gray Leaf Spot, Blight)
Jasrotia et al., (2023)	Convolutional Neural Network (CNN) using Contrast Limiting Adaptive Histogram Equalization (CLAHE) on each RGB, log transformation, and RGB to HSV conversion.	96.76 %	PlantVillage (kaggle)	4 (Healthy, Blight, Common Rust, Gray Leaf Spot)
Pratama et al., (2023)	AlexNet, LeNet, MobileNet	75.87 % (AlexNet), 83.37 % (MobileNet), 80.87 % (LeNet)	Kaggle with additions from maize field in Bantul, Special Region of Yogyakarta	4 (Healthy, Blight, Gray Leaf Spot, Common Rust)
Fadhilla et al. (2023)	Convolutional Neural Network (CNN)	93 %	Mendeley Data	4 (Healthy, Common Rust, Leaf Blight, Leaf Spot)
Mishra et al. (2020)	Deep Convolu-tional Neural Network (CNN)	88.46 %	PlantVillage combined with images from maize plantation in Raebaeli and Sultanpur district.	3 (Rust, North-ern Leaf Blight, Healthy)
Proposed Research with first dataset	ResNet18, VGG16, EfficientNet-b0	95.838 % (ResNet18), 96.908 % (VGG16), 95.841 % (EfficientNet-b0)	PlantVillage (kaggle)	4 (Healthy, Common Rust, Gray Leaf Spot, Blight)
Proposed Research with second dataset on 7 class	EfficientNet-b0 with weighted loss	94.798 %	PlantVillage (kaggle) with previous research	7 (Healthy, Common Rust, Gray Leaf Spot, Blight, Fallarmy-worm, Herbi-cideburn, Zincdefi-ciency)
Proposed Research with second dataset on 8 class (weighted loss)	EfficientNet-b0	92.877 % (using weighted loss)	PlantVillage (kaggle) with previous research	8 (Healthy, Common Rust, Gray Leaf Spot, Blight, Fallarmy-worm, Herbi-cideburn, Zincde-ficiency, Cerospora)

CRediT authorship contribution statement

Krisnanda Ahadian: Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation. **Novanto Yudistira:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Conceptualization. **Bayu Rahayudi:** Writing – review & editing, Supervision, Project administration. **Ahmad Hoirul Basori:** Writing – review & editing, Project administration, Funding acquisition. **Sharaf J. Malebary:** Writing – review & editing, Project administration, Funding acquisition, Conceptualization. **Sami Alesawi:** Writing – review & editing, Supervision, Funding acquisition, Formal analysis. **Andi Besse**

Firdausiah Mansur: Writing – review & editing, Resources, Project administration, Funding acquisition. **Almuhannad S. Alorfi:** Writing – review & editing, Resources, Funding acquisition. **Omar M. Barukab:** Writing – review & editing, Supervision, Project administration.

Data and code availability statement

The datasets generated and analyzed during the current study are avail-able in the https://www.kaggle.com/datasets/smaranjitghose/corn-or- maize-leaf-disease-dataset or are available from the corresponding au-thor upon reasonable request.

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.

Acknowledgments

The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education, in Saudi Arabia for funding this research work through project number "IFPRC-216-830-2020" and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

References

[1] Y. Zhang, S. Wa, Y. Liu, X. Zhou, P. Sun, Q. Ma, High-accuracy detec- tion of maize leaf diseases cnn based on multi-pathway activation func- tion module, Rem. Sens. 13 (21) (2021), https://doi.org/10.3390/rs13214218.
[2] Minister of Trade, Market Brief: HS 1005 Jagung, 2013.
[3] Z. Liu, Z. Du, Y. Peng, M. Tong, X. Liu, W. Chen, Study on corn disease identification based on PCA and SVM, in: Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automa- Tion Control Conference, ITNEC, 2020, pp. 661–664, https://doi.org/10.1109/ITNEC48623.2020.9084689. Itnec, 2020.
[4] K.P. Panigrahi, A.K. Sahoo, H. Das, A CNN approach for corn leaves disease detection to support d igital agricultural system, in: Proceedings of the 4th International Conference on Trends in Elec- Tronics and Informatics, ICOEI 2020, Icoei, 2020, pp. 678–683, https://doi.org/10.1109/ICOEI48184.2020.9142871.
[5] C.O. Zden, Corn disease detection using transfer learning, Black Sea, J. Eng. Sci. 6 (2023) 387–393, https://doi.org/10.34248/bsengineering.1322907.
[6] S. Yuliany, Aradea, A.N. Rachman, Implementasi deep learning pada sistem Klasifikasi Hama tanaman padi menggunakan metode convolu- tional neural network (CNN), Jurnal Buana Informatika 13 (1) (2022) 54–65, https://doi.org/10.24002/jbi.v13i1.5022.
[7] W. Albattah, A. Javed, M. Nawaz, M. Masood, S. Albahli, Artificial intelligence-based drone system for multiclass plant disease detection using an improved efficient convolutional neural network, Front. Plant Sci. 13 (June) (2022), https://doi.org/10.3389/fpls.2022.808380.
[8] V. Singh, A.K. Misra, Detection of plant leaf diseases using image segmentation and soft computing techniques, Information Processing in Agriculture 4 (1) (2017) 41–49, https://doi.org/10.1016/j.inpa.2016.10.005, 10.1016/j.inpa.2016.10.005.
[9] A. Waheed, M. Goyal, D. Gupta, A. Khanna, A.E. Hassanien, H.M. Pandey, An optimized dense convolutional neural network model for disease recognition and classification in corn leaf, Comput. Electron. Agric. (2020) 175, https://doi.org/10.1016/j.compag.2020.105456.
[10] V.K. Vishnoi, K. Kumar, B. Kumar, Plant disease detection using com- putational intelligence and image processing, J. Plant Dis. Prot. 128 (1) (2021) 19–53, https://doi.org/10.1007/s41348-020-00368-0.
[11] H.T. Sihotang, Sistem pakar untuk mendiagnosa penyakit pada tana- man jagung dengan metode bayes, Journal Of Informatic Pelita Nusan- tara 3 (2018) 17–22.
[12] P.A.P. Huda, A.A. Riadi, Evanita, KLASIFIKASI PENYAKIT TANAMAN PADA DAUN APEL DAN ANGGUR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORKS, JURNAL MANAJE- MEN INFORMATIKA KLASIFIKASI 8 (1) (2021) 10–17.
[13] D. Dai, P. Xia, Z. Zhu, H. Che, MTDL-EPDCLD: a multi- task deep-learning-based system for enhanced precision detec- tion and diagnosis of corn leaf diseases, Plants 12 (13) (2023), https://doi.org/10.3390/plants12132433.
[14] E. Rasywir, R. Sinaga, Y. Pratama, Analisis dan implementasi diag- nosis penyakit sawit dengan metode convolutional neural network (CNN), Paradigma - Jurnal Komputer dan Informatika 22 (2) (2020) 117–123, https://doi.org/10.31294/p.v22i2.8907.
[15] M.W. Akhyari, A. Suyoto, F.W. Wibowo, Klasifikasi penyakit pada daun jagung menggunakan convolutional neural network, in: Jurnal in- Forma : Jurnal Penelitian Dan Pengabdian Masyarakat, vol. 7, 2021. URL, https://github.com.
[16] A. Ubaidillah, E.M. Rochman, D.A. Fatah, A. Rachmad, Classifica- Tion of Corn Diseases Using Random Forest, Neural Network, and Naive Bayes Methods, vol. 2406, Institute of Physics, 2022, https://doi.org/10.1088/1742-6596/2406/1/012023.
[17] N. Sandotra, P. Mahajan, P. Abrol, P.K. Lehana, Analyzing perfor- mance of deep learning models under the presence of distortions in iden- tifying plant leaf disease, International International Journal of Infor- matics and Communication Technology (IJ-ICT) 12 (2) (2023) 115–126, https://doi.org/10.11591/ijict.v12i2.pp115-126.
[18] F.A. Irawan, M. Sudarma, D.C. Khrisne, Rancang bangun aplikasi identifikasi penyakit tanaman pepaya californiaberbasis android meng- gunakan metode cnnmodel arsitektur squeezenet, Jurnal SPEKTRUM 8 (2021).
[19] M. Subramanian, K. Shanmugavadivel, P. Nandhini, On fine-tuning deep learning models using transfer learning and hyper-parameters op- timization for disease identification in maize leaves, Neural Comput. Appl. 34 (08 2022). doi:10.1007/s00521-022-07246-w.
[20] X. Qian, C. Zhang, L. Chen, K. Li, Deep learning-based identifi- cation of maize leaf diseases is improved by an attention mecha- nism: self-attention, Front. Plant Sci. 13 (April) (2022) 1–15, https://doi.org/10.3389/fpls.2022.864486.
[21] D. Malvick, Common rust on corn. https://extension.umn.edu/corn-pest-management/common-rust-corn. (Accessed 25 November 2023).
[22] S. Chhetri, S. Biswas, J. Kuilya, S. Debnath, Studies on southern corn leaf blight disease in West Bengal, Maize Journal 7 (2018) 42–47. URL, https://www.researchgate.net/publication/347653065.
[23] C. U. C. of Agriculture, L. Sciences, Northern corn leaf blight. https://cals.cornell.edu/field-crops/corn/diseases-corn/northern-corn-leaf-blight. (Accessed 25 November 2023).
[24] A.B. Kutawa, K. Ahmad, A. Ali, M.Z. Hussein, M.A.A. Wahab, K. Sijam, State of the art on southern corn leaf blight disease incited by cochliobolus heterostrophus: detection, pathogenic variability and novel control measures, Bulgarian Journal of Agricultural Science 27 (2021) 147–155.
[25] C.S.B.U. States, If southern corn leaf blight strikes. https://www.cropscience.bayer.us/articles/cp/if-southern-corn-leaf-blight-strikes, 2018. (Accessed 27 November 2023).
[26] C.U.C. of Agriculture, L. Sciences, Gray leaf spot. https://cals.cornell.edu/field-crops/corn/diseases-corn/gray-leaf-spot. (Accessed 25 November 2023).
[27] B. Potter, Armyworm (2021). https://extension.umn.edu/corn-pest-management/armyworm. (Accessed 25 November 2023).

[28] R. Bessin, Fall Armyworm in Corn, 2019.
[29] S.A. Clay, Herbicide Injury to Corn, 2016.
[30] J. Camberato, S. Maloney, Zinc Deficiency in Corn, 2012.
[31] A.K. Sutradhar, Zinc for crop production. https://extension.umn.edu/micro-and-secondary-macronutrients/zinc-crop-production, 2016. (Accessed 25 November 2023).
[32] J. Dabass, S. Arora, R. Vig, M. Hanmandlu, Segmentation tech- niques for breast cancer imaging modalities- A review. Proceed- Ings of the 9th International Conference on Cloud Comput- Ing, Data Science and Engineering, Confluence 2019, 2019, pp. 658–663doi, https://doi.org/10.1109/CONFLUENCE.2019.8776937.
[33] A.R. Beeravolu, S. Azam, M. Jonkman, B. Shanmugam, K. Kan- noorpatti, A. Anwar, Preprocessing of breast cancer images to cre- ate datasets for deep-CNN, IEEE Access 9 (2021) 33438–33463, https://doi.org/10.1109/ACCESS.2021.3058773.
[34] M. Hussain, J.J. Bird, D.R. Faria, A study on CNN transfer learning for image classification, Adv. Intell. Syst. Comput. 840 (October) (2019) 191–202, https://doi.org/10.1007/978-3-319-97982-316.
[35] M. Al-Amidie, A. Al-Asadi, A.J. Humaidi, A. Al-Dujaili, L. Alzubaidi, L. Farhan, M.A. Fadhel, R.G. McGarvey, N.E. Islam, Robust spectrum sensing detector based on mimo cognitive radios with non- perfect channel gain, Electronics (Switzerland) 10 (5) (2021) 1–19, https://doi.org/10.3390/electronics10050529.
[36] F. Ramzan, M.U. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, Z. Mehmood, A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks, J. Med. Syst. 44 (12) (2019), https://doi.org/10.1007/s10916-019-1475-2.
[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei- Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y, arXiv:1409.0575.
[38] D. Theckedath, R. Sedamkar, Detecting affect states using vgg16, resnet50 and se-resnet50 networks, SN Computer Science 1 (2020) 1–7.
[39] A.K. Rangarajan, R. Purushothaman, Disease classification in egg- plant using pre-trained vgg16 and msvm, Tech. rep. 12 (2020), https://doi.org/10.1038/s41598-020-59108-x.
[40] D.I. Swasono, H. Tjandrasa, C. Fathicah, Classification of Tobacco Leaf Pests Using Vgg16 Transfer Learning, 2019, pp. 176–181.
[41] Q. Guan, Y. Wang, B. Ping, D. Li, J. Du, Y. Qin, H. Lu, X. Wan, J. Xi, Deep convolutional neural network vgg-16 model for differential di- agnosing of papillary thyroid carcinomas in cytological images: a pilot study, J. Cancer 10 (2019) 4876–4882, https://doi.org/10.7150/jca.28769.
[42] N.L. Tun, A. Gavrilov, N.M. Tun, D.M. Trieu, H. Aung, Remote Sensing Data Classification Using a Hybrid Pre-trained Vgg16 Cnn-Svm Classifier, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 2171–2175, https://doi.org/10.1109/ElConRus51938.2021.9396706.
[43] M. Loukadakis, J. Cano, M. O 'Boyle, Accelerating deep neural net- works on low power heterogeneous architectures. 11th International Workshop on Programmability and Architectures for Heterogeneous Multicores (MULTIPROG-2018) (August), 2018. URL, http://homepages.inf.ed.ac.uk/jcanore/pub/2018_multiprog.pdf.
[44] M. Tan, Q.V. Le, Efficientnet: rethinking model scaling for convolu- tional neural networks, URL, http://arxiv.org/abs/1905.11946, 2019.
[45] V.K. Dubey, K. Murari, T. Nath, K. Poddar, Advanced mri segmen- tation algorithm for the detection of brain tumor using u-net architec- ture with transfer learning efficientnet-b7, in: R.K. Tiwari, G. Sahoo (Eds.), Recent Trends in Artificial Intelligence and IoT, Springer Nature Switzerland, Cham, 2023, pp. 183–199.
[46] T.A. Putra, S.I. Rufaida, J.S. Leu, Enhanced skin condition prediction through machine learning using dynamic training and testing augmen- tation, IEEE Access 8 (2020) 1–13, https://doi.org/10.1109/ACCESS.2020.2976045.
[47] R. Sagar, What does freezing A layer mean and how does it help in fine tuning neural networks, URL, https://analyticsindiamag.com/what-does-freezing-a-layer-mean-and-how-does-it-help-in-fine-tuning-neural-networks/, may 2019.
[48] H. Gupta, Train your deep learning faster: FreezeOut, URL, https://www.kdnuggets.com/2017/08/train-deep-learning-faster-freezeout.html, aug 2017.
[49] M. Ben Naceur, M. Akil, R. Saouli, R. Kachouri, Fully automatic brain tumor segmentation with deep learning-based selective attention using overlapping patches and multi-class weighted cross-entropy, Journal of Medical Image Analysis 63 (jul 2020).
[50] S. Jasrotia, J. Yadav, N. Rajpal, M. Arora, J. Chaudhary, Convolu- Tional Neural Network Based Maize Plant Disease Identification, vol. 218, Elsevier B.V., 2023, pp. 1712–1721, https://doi.org/10.1016/j.procs.2023.01.149.