Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 2, June 2024, pp. 373-384
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i2.28817

Detection of Eight Skin Diseases Using Convolutional Neural Network with MobileNetV2 Architecture for Identification and Treatment Recommendation on Android Application

Ainul Furqon, Kamil Malik, Fathorazi Nur Fajri
Department of Informatics Engineering Nurul Jadid University, Paiton, Probolinggo and 67291, Indonesia

ARTICLE INFO

Article history:

Received May 12, 2024
Revised June 29, 2024
Published July 12, 2024

Keywords:

Android;
CNN MobileNetV2;
Skin Diseases;
Treatment

ABSTRACT

Skin diseases are common in Indonesia due to the tropical climate, high population density, and low public awareness about skin health. These diseases are often caused by infections, chemical contamination, or other external factors and typically develop internally before becoming visible, with contact dermatitis being the most frequently reported condition. To address this issue, this research proposes the use of Artificial Intelligence (AI), specifically Convolutional Neural Network (CNN) with the MobileNetV2 architecture, to detect eight types of skin diseases, namely cellulitis, impetigo, athlete's foot, nail fungus, ringworm, cutaneous larva migrans, chickenpox, and shingles. MobileNetV2 was chosen for its efficiency and high accuracy in mobile applications. The methodology involves developing a detection system using CNN MobileNetV2, integrated into an Android application to identify skin diseases and provide treatment recommendations. The dataset was collected, labeled, resized, and normalized to meet the model requirements. After training, the model was tested using a separate dataset to ensure its generalization ability and was finally integrated into the Android application. This application allows users to detect skin diseases and receive treatment advice directly. The research results show that the CNN MobileNetV2 model achieves high accuracy in classifying the eight types of skin diseases, with stable performance over several training epochs. Evaluation of the test dataset revealed an overall accuracy of 97%, with high precision, recall, and F1-score for all disease classes. The application achieved an accuracy of 84% on general data, demonstrating its practical utility. However, the need for real-time updates of treatment information was identified as a limitation. This research advances skin disease detection technology and improves public access to accurate healthcare services. Future studies should focus on real-time treatment information updates and expanding the range of detectable diseases to enhance skin disease application.

Corresponding Author:

Kamil Malik, Departement of Informatics Engineering,
Faculty of Engineering, Nurul Jadid University, Paiton, Probolinggo and 67291, Indonesia
Email: Kamil@unuja.ac.id

1. INTRODUCTION

The skin acts as an intermediary between internal organs and the external environment, playing a role in detecting stimuli such as pressure and pain and protecting against harmful environmental influences [1], [2]. Skin disorders can be triggered by various causes, including diseases, chemical contamination, infections, or external stimulation, and these conditions usually develop internally before becoming visible on the skin's surface [3], [4]. This study examines eight types of skin diseases: cellulitis, impetigo, athlete's foot, nail fungus, ringworm, cutaneous larva migrans, chickenpox, and shingles, one of which is commonly found among the

Indonesian population. According to epidemiological data from March 2022, the high prevalence of skin diseases in Indonesia reached 97% of 389 reported cases, with contact dermatitis being the most frequently occurring case, indicating that the challenges of skin health faced in this country are significant.

In Indonesia, skin diseases are a common health problem due to the tropical climate. The high population density in major cities and the lack of public awareness about the importance of skin health are significant factors [5]. Maintaining skin health is crucial because unhealthy skin can trigger various skin diseases. Personal and environmental hygiene are the first steps in preventing skin problems. However, when skin issues arise, consulting a doctor is usually necessary for an accurate diagnosis [6]. The challenge is that some skin diseases have similar symptoms but require different treatments. Economic factors often hinder people from seeing a doctor due to the high costs of examinations and treatments. Therefore, using artificial intelligence systems to detect skin diseases is expected to help improve accessibility to quick and accurate diagnoses for the public [7].

Artificial Intelligence (AI) refers to the ability of computer systems to mimic human behavior and solve various specific tasks [8]. One application of AI is detecting objects in digital images, including recognizing colors and patterns of skin diseases. For object detection, the method used is Convolutional Neural Network (CNN) [9]. CNN is a deep learning technique that utilizes convolutional layers to process inputs with filters [10]. Various CNN architectures, such as MobileNetV2, ResNet, and VGG, are used in image data processing [11]. This study chose the MobileNetV2 architecture because it focuses on increasing speed and efficiency while maintaining high accuracy in image recognition. The Convolutional Neural Network (CNN) architecture MobileNetV2 can also be called CNN MobileNetV2. MobileNetV2 is specifically designed for mobile applications such as Android devices, making it very suitable for implementation in skin disease detection that requires high performance and quick response [12].

Previous research by [13] successfully detected six skin diseases, including acne, athlete's foot, chickenpox, eczema, skin cancer, and vitiligo. They used a skin disease dataset from the internet and processed the model using CNN methods, achieving an accuracy rate of 81.75%. Although the results of this study were applied to a mobile application, the available features only facilitated the display of skin disease detection results. Other studies, such as the one conducted in [14], focused on detecting monkeypox using the CNN SE-ResNet architecture, achieving a model accuracy of 95%, which is higher than pure CNN. Unfortunately, this detection has not been implemented in a mobile application.

The study [15] also focused on detecting skin cancer to simplify risk assessment. They used CNN MobileNetV2 and achieved a 95% accuracy in data processing, although detection results on their Android application showed an accuracy of 70%. While CNN MobileNetV2 and CNN SE-ResNet have comparable accuracy, the use of SE-ResNet in Android applications was not pursued, as MobileNetV2 is more suitable for mobile applications with high efficiency in image recognition. In contrast, SE-ResNet is usually used in applications requiring more complex and accurate feature representations.

This research improves skin disease detection using Convolutional Neural Network (CNN) and the MobileNetV2 architecture. Unlike previous studies that only focused on a few types of skin diseases [16], this study covers eight types of skin diseases. MobileNetV2 was chosen for its efficiency and exemplary performance in mobile applications, which can provide high detection accuracy, as evidenced in previous similar studies [17]. Furthermore, this research develops a detection model and integrates it into an Android application. This lets users quickly detect skin diseases and receive treatment recommendations directly through the application. Thus, this research advances skin disease detection technology and expands public access to accurate and integrated health services.

2. METHODS

The research methodology aims to develop a skin disease detection system using a Convolutional Neural Network (CNN) [18] with a MobileNetV2 architecture, which will be implemented in an Android application for skin disease identification and treatment recommendations. The research methodology steps are explained in Fig. 1. First, the collected dataset is processed to prepare labeled images according to the type of disease, divided into training and testing data, resized, and normalized to be suitable for machine learning with the CNN MobileNetV2 model [19]. Next, the testing data is used to evaluate the model's performance. Once the model is trained and tested, it is integrated into the Android application [20]. This application will feature treatment recommendations for eight types of skin diseases, allowing users to receive treatment advice based on the disease identification results from the model.

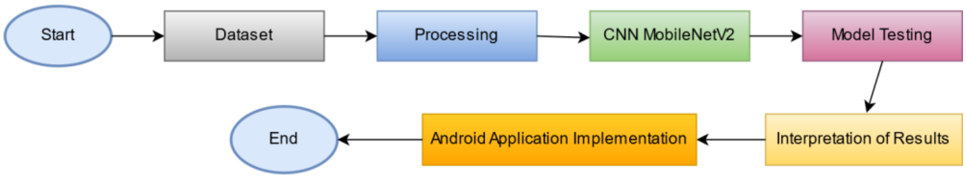


Fig. 1. Research System Design

2.1. Dataset

In the era of Society 5.0, data can be collected from various sources that are accessible to anyone. For several fundamental reasons, this study uses eight types of skin disease data: cellulitis, impetigo, athlete's foot, nail fungus, ringworm, cutaneous larva migrans, chickenpox, and shingles. First, these diseases are common and frequently encountered in daily medical practice [21], ensuring relevance and wide application in the clinical world. Second, each disease has distinct visual and clinical characteristics, allowing the model to be tested on various patterns and features [22]. Third, some of these diseases have symptoms similar to other skin conditions, requiring accurate detection to help evaluate the model's ability to address diagnostic complexity. Fourth, the dataset used was obtained from Kaggle and provided high-quality images with explicit annotations for each disease, which is a crucial factor in deep-learning research. Fifth, quickly and accurately identifying these skin diseases can significantly impact healthcare, as these diseases can cause severe complications if not recognized and treated promptly. Finally, selecting these diseases provides sufficient challenges to test the MobileNetV2 model's capabilities [23], allowing this study to evaluate the model's performance and generalization comprehensively. The dataset from Kaggle was chosen to ensure quality, validity, and ease in this study, as the platform provides high-quality datasets with explicit annotations and is well-regarded and widely recognized. Additionally, the dataset used in this study has been accessed by approximately 14,600 users and downloaded thousands of times, ensuring its excellent quality and reputation. Besides the dataset from Kaggle, additional data was also collected from the general public in the research environment to test the model's capability and observe the accuracy of the results from this skin disease detection Android application.

2.2. Processing

The processing procedure involves a series of actions or operations performed on raw data to transform it into a more valuable and meaningful form. First, the data is divided into training and testing data to evaluate the trained model accurately [24]. Training data trains the model to learn patterns and features from the dataset [25]. In contrast, testing data is used to test the model's performance on unseen data, allowing the evaluation of the model's generalization ability and ensuring the model does not simply memorize the training data. Next, data labelling is performed, and the names of skin diseases are converted into numerical labels. This aims to facilitate the modelling process, ensure consistent data formatting, improve processing efficiency, and avoid errors that can occur with text data. The machine learning model can learn and predict more effectively and accurately with numerical labels. Data scaling is conducted, which involves resizing images to ensure all images have consistent dimensions before being used in the machine learning model training [26]. This helps maintain data consistency, optimize the training process, avoid image distortion, and meet the input size requirements of the deep learning model, thus enabling the model to process images efficiently and accurately. Finally, the data is normalized by dividing each pixel value to fall within the range of 0-1, which aims to accelerate model convergence during training and enhance numerical stability.

2.3. CNN MobileNetV2

Convolutional Neural Network (CNN) MobileNetV2 is a deep learning model architecture designed for mobile and embedded applications with limited computing resources, combining efficiency and high accuracy in recognizing images [27]. In this research, MobileNetV2 is used as a feature extractor through TensorFlow Hub. A feature extractor via TensorFlow Hub is a pre-trained model component that extracts important features from raw data such as images or text, allowing users to utilize models already trained on large datasets. This saves time and computational resources and improves the accuracy and efficiency of the machine-learning process without the need to train models from scratch [28]. Therefore, the feature extractor through TensorFlow Hub offers high efficiency and speed with a small model size. It is pre-trained on large datasets such as ImageNet and has good generalization capabilities for various computer vision tasks. The model is implemented in the Keras layer with pixel-sized input images to meet the requirements of the MobileNetV2 architecture. Keras is one of the popular deep-learning libraries due to its ease of use. It provides various types of layers that are used to build neural network models. This layer is disabled for retraining (trainable=False) to maintain the optimized weights and speed up the training process [29]. The Keras layer used in this study involves a dense layer with eight neurons and uses softmax activation. Softmax activation is a function that is often used in artificial neural networks, especially in the output layer in multiclass classification problems. The softmax function converts the network output into probabilities that can be summed up to one so that each class has a probability between 0 and 1.

$$\sigma(Z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{1}$$

The explanation of the formula is as follows: z_i is the score produced by the neural network for class i, K is the total number of classes, and $\sigma\,(z_i)$ is the probability of class i. Softmax activation is added to classify the images into eight types of skin diseases because softmax is suitable for multiclass classification tasks. The model is compiled using the 'Adam' optimizer, known for its efficiency in optimization, and the 'sparse_categorical_crossentropy' loss, which is appropriate for multiclass classification with unordered labels. The accuracy metric is used to evaluate the model's performance. The training process is conducted over several epochs, with normalized data, to accelerate convergence and improve numerical stability [30]. The final results are summarized using 'the model. Summary ()' provides essential information about the model's structure and parameters.

2.4. Model Testing

Model testing is a crucial stage in the machine learning model development cycle, aimed at evaluating the model's performance that has been trained using training data [31]. This evaluation uses test data, different from the training data, to assess how well the model can generalize to previously unseen data. The test data has been normalized, meaning feature values are scaled to a smaller range to ensure balanced contributions from all features. The evaluation process involves invoking specific methods designed to calculate performance metrics, such as accuracy, which measures how correct the model's predictions are compared to the actual values [32]. This evaluation aims to ensure that the model performs well not only on the training data but also on new data, referred to as generalization capability. Evaluation with test data helps identify overfitting issues, which occur when the model is overly trained on the training data, capturing noise or irrelevant details [33]. If the model performs well on the training data but poorly on the test data, this indicates overfitting. Using test data not used during training suggests the model's reliability and accuracy in real-world situations, giving us more confidence that the model will perform well when applied to new or unknown data.

2.5. Interpretation of Results

The interpretation of results in this study includes evaluating model performance by predicting labels on normalized test data using the trained model. These prediction results are then converted into predicted labels by applying the np. Arg max function on each output selects the class with the highest probability as the predicted label for each sample [34]. Next, the predicted labels are compared with the actual labels using classification_report and confusion_matrix from sklearn to calculate evaluation metrics such as precision, recall, f1-score, and accuracy. This confusion matrix is then visualized using a heatmap from Seaborn, facilitating the analysis of the model's performance in identifying eight types of skin diseases. The calculation formulas for precision, recall, f1-score, and accuracy based on the confusion matrix are presented to enable manual calculation of these metrics, providing a deeper understanding of how the metrics are derived from the confusion matrix. Precision calculation formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall calculation formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1-score calculation formula:

$$\text{F1} - \text{score} = 2\text{x}\frac{\text{Precision} + \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Accuracy calculation formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5}$$

2.6. Android Application Implementation

The implementation of the Android application involves converting the trained model to TensorFlow Lite format using tf. Lite.TFLiteConverter.from_keras_model(model), where the converted model is then saved as a .tflite file for use in the Android application [35]. This process enables the integration of the machine learning model into mobile devices [36] to efficiently detect and identify eight types of skin diseases. The design of the Android application can be seen in Fig. 2.

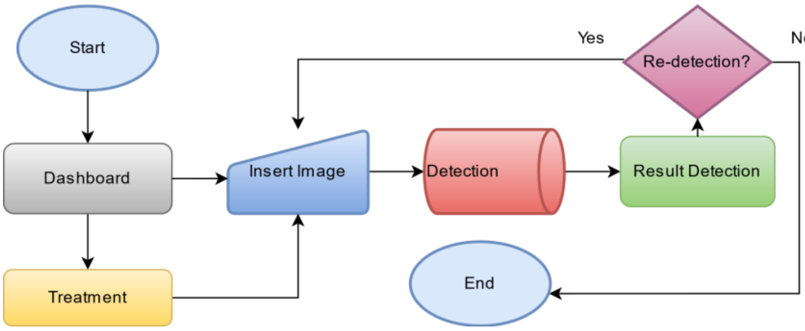


Fig. 2. System Design Of Android Application

This flowchart illustrates the operational process of an application that begins with the 'Start' stage and then guides the user to the 'Dashboard' to upload an image through the 'Insert Image' option. Once the image is uploaded, the detection system with the 'Detection' function will display the results in 'Result Detection' and evaluate whether 'Re-detection' is needed. If so, the process returns to the 'Insert Image' stage; if not, the process ends at 'End.' Additionally, users can access 'Treatment' from the 'Dashboard' for further information about their disease treatment.

3. RESULTS AND DISCUSSION

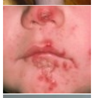
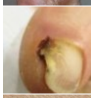
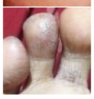
This section will discuss the evaluation results of the Convolutional Neural Network (CNN) [37] model's performance using the MobileNetV2 architecture trained to detect eight types of skin diseases. The model's prediction results on normalized test data will be analyzed in depth. The evaluation begins with predicting labels on the test dataset [38], converting the prediction results into predicted labels, and then comparing those labels with the actual labels. Evaluation metrics such as precision, recall, and F1-score are calculated using the classification_report from sklearn, while the overall model performance is visualized through a confusion matrix using a heatmap from Seaborn. This analysis aims to clearly understand the model's ability to identify each type of skin disease [39] and identify areas needing further improvement. This approach ensures the developed model is accurate and reliable in real-world applications. The results and discussions presented in this section will offer comprehensive insights into the model's effectiveness, outlining its strengths and weaknesses and the potential application of the model in everyday medical practice.

3.1. Dataset

This study's dataset from Kaggle consists of 1159 images, divided into 925 training data and 234 testing data. Additionally, there are ten extra images taken from the general public. The dataset includes eight types of skin diseases: cellulitis, impetigo, athlete's foot, nail fungus, ringworm, cutaneous larva migrans, chickenpox, and shingles. The types of diseases in the data from the general public have not been classified. They will be used as test data for the Android application to assess the performance and accuracy of the CNN MobileNetV2 model. Detailed information about data distribution is presented in Table 1.

This study's discussion of the dataset highlights eight types of skin diseases with high-quality images and well-defined disease patterns from the training data [40], in contrast to the testing data, which has less clear resolution and patterns. Images from the general public have been adjusted in size and resolution to improve clarity and visibility. This dataset, sourced from Kaggle and scientifically recognized, provides more consistent and verified data than previous studies that often collected random images from the internet. In addition to Kaggle, additional public data was also used to validate the CNN model MobileNetV2 further [41].

Table 1. Dataset

Type of Disease	Amount of Data			Image
	Training Data	Testing Data	General Data	
Cellulitis	136	34	-	
Impetigo	80	20	-	
Athlete's Foot	124	32	-	
Nail Fungus	129	33	-	
Ringworm	90	23	-	
Cutaneous Larva Migrans	100	25	-	
Chickenpox	136	34	-	
Shingles	130	33	-	
General Data	-	-	10	
Total Data	**925**	**234**	**10**	-

3.2. Processing

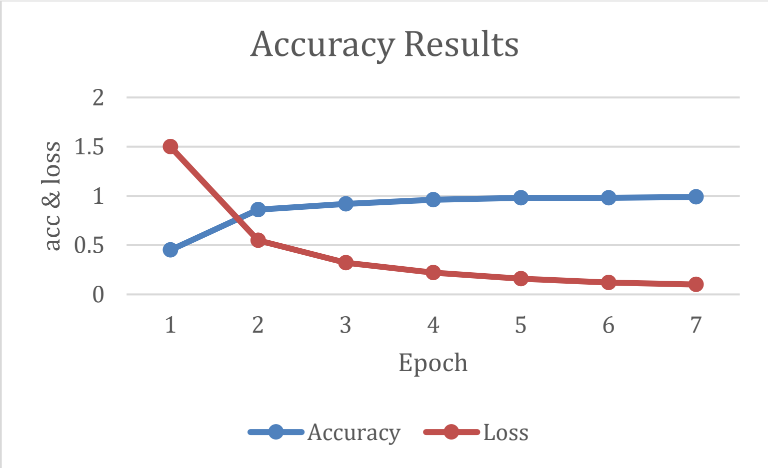
At the result processing stage, this study split the dataset into training data (80%) and testing data (20%) using sklearn's train_test_split method. This split ensures that the model can generalize well to new data, as it allows the model to be trained on most of the data and evaluated for performance on previously unseen data. After splitting the dataset, the data was relabelled by changing the skin disease names to numerical labels, as shown in Table 2. This change was made to ease the modelling process, ensure consistency of data format, improve processing efficiency, and avoid errors that may occur with text data. Next, each image in the dataset was resized to 224×224 pixels using the cv2.resize function of OpenCV. This was done to ensure input uniformity by the requirements of the MobileNetV2 model [42], as consistent input size is a prerequisite for incorporating images into deep learning models with a particular architecture. After that, data normalization was performed by dividing each pixel value by 255 so that it falls within the range of 0-1. This step aims to accelerate the convergence of the model during training and improve numerical stability so that the model can learn more effectively and efficiently. Normalization helps reduce large-scale issues in pixel values that may affect the model's learning process [43].

3.3. CNN MobileNetV2

In this study, MobileNetV2 was used as the feature extractor through TensorFlow Hub with the URL https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4; due to its high efficiency and speed, small model size, and pre-training on large datasets such as ImageNet, which grants it good generalization ability for various computer vision tasks [44], [45]. The model implementation was performed in the Keras layer with input images of 224×224 pixels to meet the requirements of the MobileNetV2 architecture. This layer was set to non-trainable (trainable=False) to maintain the optimized weights and accelerate the training process. Subsequently, a dense layer with eight neurons and softmax activation was added to classify the images into eight skin disease types, as softmax is suitable for multiclass classification tasks. The model was compiled using the 'Adam' optimizer, known for its efficiency in optimization, and the 'sparse_categorical_crossentropy' loss, which is suitable for multiclass classification with unordered labels. An accuracy metric was used to evaluate the model's performance [46], [48]. The training process was conducted for seven epochs using normalized data to speed up convergence and improve numerical stability. The accuracy results of the 7-epoch training are shown in Graph 1. The final results were summarized using the model. Summary () to provide an overview of the model architecture and parameters.

Table 2. Labeling Data

Table Head	Labeling Data	
	Skin Disease Name	Labeling
1	Cellulitis	0
2	Impetigo	1
3	Athlete's Foot	2
4	Nail Fungus	3
5	Ringworm	4
6	Cutaneous Larva Migrans	5
7	Chickenpox	6
8	Shingles	7



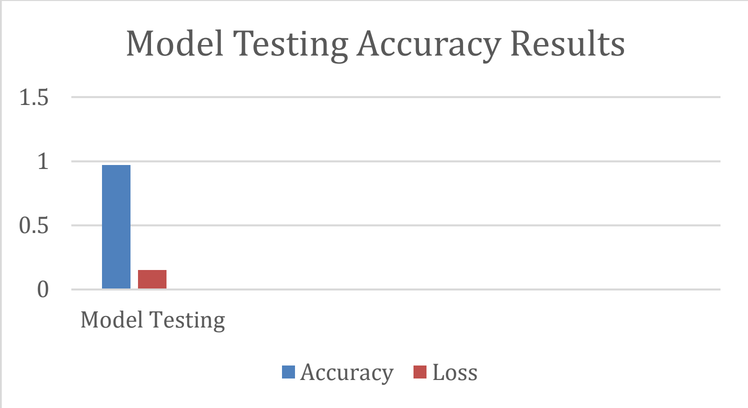
Graphs 1. CNN MobileNetV2 Accuracy Results

Based on the provided graph, the MobileNetV2 model shows promising results in image classification after seven epochs of training. In the first epoch, the model starts with a low accuracy of around 0.4 and a high loss value of about 1.5. However, in the second epoch, there is a significant increase in accuracy, approaching 0.8, and a sharp decrease in the loss value to below 0.8. From the third to the seventh epoch, the model's accuracy tends to stabilize around 0.8, indicating that it achieves stability in its learning process. The loss value decreases consistently from around 0.8 in the third epoch to nearly 0.2 in the seventh, indicating ongoing improvement in minimizing prediction errors. These results suggest that MobileNetV2, as a feature extractor, can perform well in image classification tasks with only a few epochs of training, leveraging weights pre-trained on large datasets like ImageNet. This model demonstrates good generalization ability, effectiveness in learning the training data, and consistent stability and loss reduction during the training process.

3.4. Model Testing

The model testing was conducted to evaluate the performance of the trained model using normalized test data, aiming to measure performance metrics such as accuracy on unseen data through the method model. Evaluate (x_test_scaled, y_test), ensuring the model can generalize well [49], as indicated by the accuracy results in Graph 2.

Based on the provided graph, two main metrics of the model testing results are shown: accuracy and loss. Accuracy is displayed with blue bars and approaches a value of 1, indicating that the model has a high accuracy level in predicting test data. The loss value, shown with red bars, is relatively low and close to 0, indicating that the errors made by the model in predicting test data are minimal. This combination of high accuracy and low loss suggests that the model has excellent performance, can make correct predictions in most cases, and is stable in making predictions with minimal error. Therefore, these results demonstrate that the model can generalize well to new data, does not experience significant overfitting issues, and provides confidence that the model will perform well in real-world situations.



Graphs 2. Testing Model Accuracy Results

3.5. Interpretation of Results

The interpretation of the results in this study includes evaluating the model's performance by predicting labels on the normalized test data (x_test_scaled) using the trained model. These predictions are converted into predicted labels by applying the np—argmax function to each output. Subsequently, using the classification_report (the results of which can be seen in Fig. 3 and confusion_matrix from sklearn, the predicted labels are compared with the actual labels (y_test) to calculate evaluation metrics. The confusion matrix is then visualized using a seaborn heatmap, which facilitates the analysis of the model's performance in identifying eight types of skin diseases.

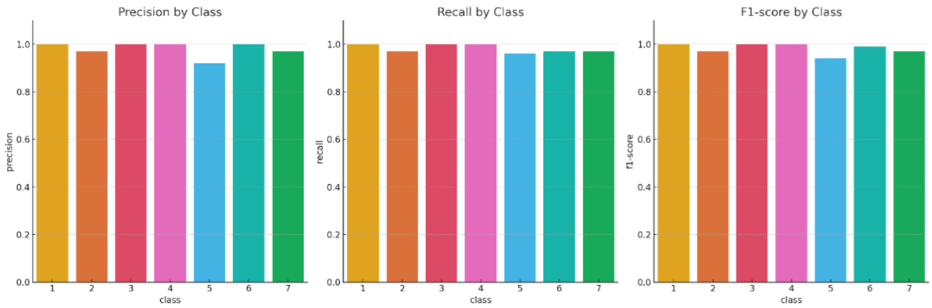


Fig. 3. Classification Report

The image displays each class's precision, recall, and F1-score graphs, showing the model's performance in identifying eight types of skin diseases. The precision graph shows that classes 1, 3, 4, and 6 have a perfect precision of 1.00, meaning all predictions for these classes are correct. Meanwhile, classes 2, 5, and 7 have slightly lower but still very high precision, at 0.97, 0.92, and 0.97, respectively. The recall graph shows similar results, with classes 1, 3, 4, and 6 having a perfect recall of 1.00, indicating that the model correctly identified all actual occurrences of these classes. Classes 2, 5, and 7 have recalls of 0.97, 0.96, and 0.97, which is also very high. The F1-score graph, which is the harmonic mean of precision and recall, shows that classes 1, 3, 4, and 6 have perfect F1-scores of 1.00, while classes 2, 5, and 7 have F1-scores of 0.97, 0.94, and 0.97, respectively. In conclusion, this model demonstrates excellent performance in identifying types of skin diseases, with very high precision, recall, and F1 scores for all classes. Classes 1, 3, 4, and 6 show perfect performance, while classes 2, 5, and 7 show excellent and acceptable performance. These graphs visualize how well the model works for each class and help identify areas needing further improvement. The formulas for calculating precision, recall, F1-score, and accuracy based on the confusion matrix are provided to allow for manual computation of these metrics, with the results presented in Table 3.

Based on the calculation table, the model performs exceptionally well in classifying eight skin disease types. For classes 1, 3, and 4, the precision, recall, and F1-score are all perfect at 1.000, indicating that the model makes no errors in classifying samples from these classes. Class 2 also shows near-perfect performance with a precision of 0.967, recall of 1.000, and F1-score of 0.983. Class 5 demonstrates precision and recall of 0.960, with the same F1-score, indicating that although there are a few errors, the model's performance remains excellent. Classes 6 and 7 have precision values of 0.971 and 0.970, recall values of 1.000 and 0.970, and F1-

scores of 0.985 and 0.970, respectively, showing that despite some errors, the model is still highly efficient in classifying the data. Overall, the model's accuracy reaches 0.979, indicating outstanding performance in this classification task.

Table 3. Confusion Matrix Result

Class	TP	FP	FN	Precision	Recall	F1-Score
1	13	0	0	1.000	1.000	1.000
2	29	1	0	0.967	1.000	0.983
3	33	0	0	1.000	1.000	1.000
4	23	0	0	1.000	1.000	1.000
5	24	1	1	0.960	0.960	0.960
6	33	1	0	0.971	1.000	0.985
7	32	1	1	0.970	0.970	0.970

3.6. Android Apllication Implementation

In the Android application implementation results, this research presents Fig. 4, which shows the flow of the application's screenshots running on the researcher's mobile phone. This image demonstrates that this is not an application design but an already-running application. Users will see a "Splashscreen" that provides an initial app overview when accessing the application. After that, users are directed to the dashboard, which has three main features: Treatment, Detection, and About the Application. The Treatment feature provides information on handling various types of skin diseases in terms of medication and treatment methods. The Detection feature includes information about the skin disease experienced by the user. In this feature, two detection results are displayed: from the dataset and general data, to compare and see the accuracy of the application in detecting skin diseases.

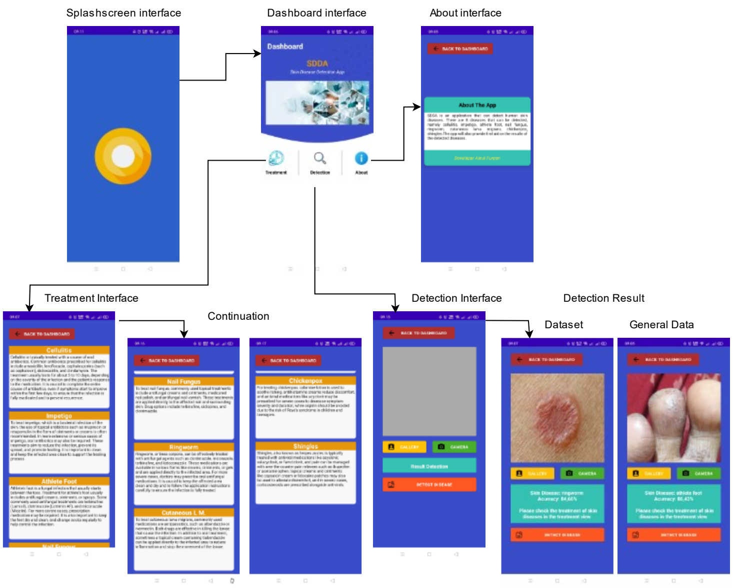


Fig. 4. Skin Disease Detection App

The accuracy from the dataset shows 84.66% for ringworm, while the general data shows an accuracy of 86.43% for athlete's foot. The About the Application feature provides information about the app itself. In discussing the Detection feature, when users want to re-detect, they must re-upload the image of the skin disease. They can return to the dashboard if they do not wish to detect it again. The detection accuracy of skin diseases on Android is almost close to the accuracy of manual interpretation, showing excellent accuracy values, and the skin disease detection results are in line with the doctor's examination results. With these results, the skin disease detection application provides clear and accurate results in detecting skin diseases, offering significant benefits for both the public and doctors in examining skin diseases. However, this research has a limitation: the treatment information is only sometimes real-time, and treatment conditions change yearly. The treatment information was obtained from ChatGPT-4 sources in 2024 and may not be the same. Therefore, the researchers propose that future research develops real-time annual updates on skin disease treatment information to ensure that treatment remains accurate and effective in addressing skin diseases.

4. CONCLUSION

This research successfully developed an Android application [50] to detect eight types of skin diseases using a Convolutional Neural Network (CNN) model with the MobileNetV2 architecture. The implementation results show that this application provides not only accurate detection results, with a testing accuracy of 97% for the CNN MobileNetV2 model and 84% detection accuracy for skin diseases in the Android application using general data, but also offers helpful treatment information for users. The evaluation indicates that the model has a high accuracy rate, with excellent precision, recall, and F1-score for all tested skin disease classes. This application lets users quickly detect skin diseases and receive treatment recommendations through mobile devices. Thus, this research enhances skin disease detection technology and expands public access to accurate and integrated healthcare services. However, the study also identifies several areas for improvement, including updating treatment information in real time to remain relevant to the latest medical developments. Therefore, further research is recommended to develop a system that can automatically update treatment information, expand the types of skin diseases detected, and improve model accuracy by using real-world data and developing more advanced model architectures. With development prospects that include integration with wearable devices and other platforms, this research has the potential to provide broader and more significant benefits to society and medical professionals, particularly in the context of skin health in Indonesia.

REFERENCES

[1] S. Malliga *et al*., "Skin Disease Detection and Classification Using Deep Learning Algorithms," *IJSREM*, vol. 8, no. 5, pp. 1-5, 2024, https://doi.org/10.55041/IJSREM34738.
[2] S. Aher and A. K. Shahi, "Skin Disease Detection Using Machine Learning," I*nternational Journal of Food and Nutritional Science,* vol. 8, no. 4, 2023, https://openurl.ebsco.com/EPDB%3Agcd%3A15%3A11699807/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A174054464&crl=c.
[3] A. S. Jaradat *et al*., "Automated Monkeypox Skin Lesion Detection Using Deep Learning and Transfer Learning Techniques*," IJERPH*, vol. 20, no. 5, p. 4422, Mar. 2023, https://doi.org/10.3390/ijerph20054422.
[4] A. Kumari and Dr. P. Rattan, "Skin Cancer Detection and Classification using Deep learning methods*," IJEER*, vol. 11, no. 4, pp. 1072-1086, Nov. 2023, https://doi.org/10.37391/ijeer.110427.
[5] S. A. AlDera and M. T. B. Othman, "A Model for Classification and Diagnosis of Skin Disease using Machine Learning and Image Processing Techniques*," IJACSA*, vol. 13, no. 5, 2022, https://doi.org/10.14569/IJACSA.2022.0130531.
[6] J. H. Yoon, M.-Y. Kim, and J. Y. Cho, "Apigenin: A Therapeutic Agent for Treatment of Skin Inflammatory Diseases and Cancer*," IJMS*, vol. 24, no. 2, p. 1498, Jan. 2023,https://doi.org/10.3390/ijms24021498.
[7] P. Kaler, S. Kodli, and S. Anakal, "Diagnosis of Skin Cancer Using Machine Learning and Image Processing Techniques*," IJEME,* vol. 12, no. 5, pp. 38-45, Oct. 2022, https://doi.org/10.5815/ijeme.2022.05.05.
[8] E. N F *et al*., "A Hybrid Model using MobileNetv2 and SVM for Enhanced Classification and Prediction of Tomato Leaf Diseases*," SSRG-IJEEE*, vol. 10, no. 8, pp. 37-50, Sep. 2023, https://doi.org/10.14445/23488379/IJEEE-V10I8P104.
[9] S. Dissaneevate *et al*., "A Mobile Computer-Aided Diagnosis of Neonatal Hyperbilirubinemia using Digital Image Processing and Machine Learning Techniques*," IJIRSS*, vol. 5, no. 1, pp. 10-17, Jan. 2022, https://doi.org/10.53894/ijirss.v5i1.334.
[10] A. Jadhav, A. Gadekar, "Brain Tumor Detection by using Fine-tuned MobileNetV2 Deep Learning Model*," IJRITCC*, vol. 11, no. 5, pp. 134-140, May 2023, https://doi.org/10.17762/ijritcc.v11i5.6587.
[11] P. Kaur, "Performance and Accuracy Enhancement During Skin Disease Detection in Deep Learning*," IJERR*, vol. 35, pp. 96-108, Nov. 2023, https://doi.org/10.52756/ijerr.2023.v35spl.009.
[12] I. Hestiningsih, A. N. A. Thohari, - Kurnianingsih, and N. D. Kamarudin, "Mobile Skin Disease Classification using MobileNetV2 and NASNetMobile*," Int. J. Adv. Sci. Eng. Inf. Technol*., vol. 13, no. 4, pp. 1472-1479, Jul. 2023,https://doi.org/10.18517/ijaseit.13.4.18290.
[13] R. Saifan and F. Jubair, "Six skin diseases classification using deep convolutional neural network*," IJECE*, vol. 12, no. 3, p. 3072, Jun. 2022, https://doi.org/10.11591/ijece.v12i3.pp3072-3082.
[14] K. Thiruppathi, S. K, and V. Shenbagavel, "SE-RESNET: Monkeypox Detection Model*," IJACSA*, vol. 14, no. 9, 2023, https://doi.org/10.14569/IJACSA.2023.0140959.
[15] A. Wibowo, C. Adhi Hartanto, and P. Wisnu Wirawan, "Android skin cancer detection and classification based on MobileNet v2 model*," Int. J. Adv. Intell. Informatics,* vol. 6, no. 2, p. 135, Jul. 2020, https://doi.org/10.26555/ijain.v6i2.492.
[16] P. N. Srinivasu *et al,*. "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM." *Sensors,* vol. 21, no. 8, 2021, https://doi.org/10.3390/s21082852.
[17] Naresh *et al*., "Detection and Classification of Dog Skin Disease using Deep Learning*," IJSREM*, vol. 07, no. 03, Mar. 2023, https://doi.org/10.55041/IJSREM18053.
[18] R. Agarwal and D. Godavarthi, "Skin Disease Classification Using CNN Algorithms*," EAI Endorsed Trans Perv Health Tech,* vol. 9, Oct. 2023, https://doi.org/10.4108/eetpht.9.4039.

[19] P. N. Huu, V. T. Quang, C. N. Le Bao, and Q. T. Minh, "Proposed Detection Face Model by MobileNetV2 Using Asian Data Set," *Journal of Electrical and Computer Engineering*, vol. 2022, pp. 1-19, Oct. 2022, https://doi.org/10.1155/2022/9984275.
[20] M. Masparudin, I. Fitri, and S. Sumijan, "Development of Apple Fruit Classification System using Convolutional Neural Network (CNN) MobileNet Architecture on Android Platform," *SISTEMASI*, vol. 13, no. 1, p. 230, Jan. 2024,https://doi.org/10.32520/stmsi.v13i1.3533.
[21] H. Yildirim *et al*., "An automated diabetic retinopathy disorders detection model based on pretrained MobileNetv2 and nested patch division using fundus images," *Journal of Health Sciences and Medicine*, vol. 5, no. 6, pp. 1741-1746, Oct. 2022,https://doi.org/10.32322/jhsm.1184981.
[22] T.-H. Nguyen and B.-V. Ngo, "ROI-based features for classification of skin diseases using a multi-layer neural network," *IJEECS*, vol. 23, no. 1, p. 216, Jul. 2021, https://doi.org/10.11591/ijeecs.v23.i1.pp216-228.
[23] T. D. Nigat, T. M. Sitote, and B. M. Gedefaw, "Fungal Skin Disease Classification Using the Convolutional Neural Network," *Journal of Healthcare Engineering,* pp. 1-9, May 2023, https://doi.org/10.1155/2023/6370416.
[24] S. Kitsiranuwat, T. Kawichai, and P. Khanarsa, "Identification and Classification of Diseases Based on Object Detection and Majority Voting of Bounding Boxes," *JAIT*, vol. 14, no. 6, pp. 1301-1311, 2023, https://doi.org/10.12720/jait.14.6.1301-1311.
[25] B. Kleuser and W. Bäumer, "Sphingosine 1-Phosphate as Essential Signaling Molecule in Inflammatory Skin Diseases," *IJMS*, vol. 24, no. 2, p. 1456, Jan. 2023, https://doi.org/10.3390/ijms24021456.
[26] S. S, S. Saranya, and M. Devaraju, "Skin Cancer Detection and Segmentation Using Convolutional Neural Network Models," *IJEER*, vol. 10, no. 4, pp. 984-987, Dec. 2022, https://doi.org/10.37391/ijeer.100438.
[27] M. Akay *et al*., "Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model," *IEEE Open J. Eng. Med. Biol*., vol. 2, pp. 104-110, 2021, https://doi.org/10.1109/OJEMB.2021.3066097.
[28] H. M. Ahmed and M. Y. Kashmola, "Performance Improvement of Convolutional Neural Network Architectures for Skin Disease Detection," *IJCDS,* vol. 13, no. 1, pp. 657-669, Apr. 2023, https://doi.org/10.12785/ijcds/130152.
[29] P. R. Togatorop, Y. Pratama, A. Monica Sianturi, M. Sari Pasaribu, and P. Sangmajadi Sinaga, "Image preprocessing and hyperparameter optimization on pretrained model MobileNetV2 in white blood cell image classification," *IJ-AI,* vol. 12, no. 3, p. 1210, Sep. 2023, https://doi.org/10.11591/ijai.v12.i3.pp1210-1223.
[30] G. Bizel, A. Einstein, A. G. Jaunjare, and S. K. Jagannathan, "Machine Learning Study: Identification of Skin Diseases for Various Skin Types Using Image Classification.," *JBDAI,* vol. 2, no. 1, Jan. 2024, https://doi.org/10.54116/jbdai.v2i1.32.
[31] S. Panda, A. Sunil Tiwari, and M. Ranjan Prusty, "Comparative study on different Deep Learning models for Skin Lesion Classification using transfer learning approach," *IJSRP,* vol. 11, no. 1, pp. 219-229, Dec. 2020, https://doi.org/10.29322/IJSRP.11.01.2021.p10923.
[32] A. H. Khan, "Information Technology Usage in Skin Disease Detection," *IJCSRR*, vol. 06, no. 07, Jul. 2023, https://doi.org/10.47191/ijcsrr/V6-i7-38.
[33] J. Xiong *et al*., "Implementation Strategy of a CNN Model Affects the Performance of CT Assessment of EGFR Mutation Status in Lung Cancer Patients," *IEEE Access,* vol. 7, pp. 64583-64591, 2019,. https://doi.org/10.1109/ACCESS.2019.2916557.
[34] M. F. Aslan, "Comparison Of Vision Transformers And Convolutional Neural Networks For Skin Disease Classification," *In Proceedings of the International Conference on New Trends in Applied Sciences,* vol. 1, pp. 31-39, 2023, https://doi.org/10.58190/icontas.2023.51.
[35] W. M. Pradnya Dhuhita, M. Y. Ubaid, and A. Baita, "MobileNet V2 Implementation in Skin Cancer Detection," *Ilk. J. Ilm*., vol. 15, no. 3, pp. 498-506, Dec. 2023, https://doi.org/10.33096/ilkom.v15i3.1702.498-506.
[36] A. Yilmaz, G. Gencoglan, R. Varol, A. A. Demircali, M. Keshavarz, and H. Uvet, "MobileSkin: Classification of Skin Lesion Images Acquired Using Mobile Phone-Attached Hand-Held Dermoscopes," *JCM*, vol. 11, no. 17, p. 5102, Aug. 2022, https://doi.org/10.3390/jcm11175102.
[37] C. Uyulan, "Development of LSTM&CNN based hybrid deep learning model to classify motor imagery tasks," *Commun. Math. Biol. Neurosci.*, 2021, https://doi.org/10.28919/cmbn/5265.
[38] M. A. Araaf, K. Nugroho, and D. R. I. M. Setiadi, "Comprehensive Analysis and Classification of Skin Diseases based on Image Texture Features using K-Nearest Neighbors Algorithm," *J. Comput. Theor. Appl*., vol. 1, no. 1, pp. 31-40, Sep. 2023, https://doi.org/10.33633/jcta.v1i1.9185.
[39] F. A. Alsalman, S. Khorshid, and A. Sallow, "Disease Diagnosis Systems Using Machine Learning and Deep learning Techniques Based on TensorFlow Toolkit: A review," *AL-Rafidain Journal of Computer Sciences and Mathematics*, vol. 16, no. 1, pp. 111-120, Jun. 2022, https://doi.org/10.33899/csmj.2022.174415.
[40] R. A. Pratiwi, S. Nurmaini, D. P. Rini, M. N. Rachmatullah, and A. Darmawahyuni, "Deep ensemble learning for skin lesions classification with convolutional neural network," *IJ-AI*, vol. 10, no. 3, p. 563, Sep. 2021, https://doi.org/10.11591/ijai.v10.i3.pp563-570.
[41] D. Moturi, R. K. Surapaneni, and V. S. G. Avanigadda, "Developing an efficient method for melanoma detection using CNN techniques," *J Egypt Natl Canc Inst*, vol. 36, no. 1, p. 6, Feb. 2024, https://doi.org/10.1186/s43046-024-00210-w.
[42] M. Akay *et al*., "Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model," *IEEE Open J. Eng. Med. Biol.*, vol. 2, pp. 104-110, 2021, https://doi.org/10.1109/OJEMB.2021.3066097.

[43] A. M. Dhayea, N. K. El Abbadi, and Z. G. Abdul Hasan, "Human Skin Detection and Segmentation Based on Convolutional Neural Networks," *Iraqi Journal of Science*, pp. 1102-1116, Feb. 2024, https://doi.org/10.24996/ijs.2024.65.2.40.
[44] Aher *et al.*, "Skin Disease Detection Using Machine Learning And Convolutional Neural Network," *IRJMETS*, May 2023, https://doi.org/10.56726/IRJMETS40689.
[45] A. K. Mandal, P. K. D. Sarma, and S. Dehuri, "Image-based Skin Disease Detection and Classification through Bioinspired Machine Learning Approaches," *IJRITCC,* vol. 12, no. 1, pp. 85-94, Sep. 2023, https://doi.org/10.17762/ijritcc.v12i1.7914.
[46] F. D. Wibowo, I. Palupi, and B. A. Wahyudi, "Image Detection for Common Human Skin Diseases in Indonesia Using CNN and Ensemble Learning Method," *JoSYC,* vol. 3, no. 4, pp. 527-535, Sep. 2022, https://doi.org/10.47065/josyc.v3i4.2151.
[47] M. Abbas, M. Imran, A. Majid, and N. Ahmad, "Skin Diseases Diagnosis System Based on Machine Learning," *JCBI*, vol. 4, no. 1, Dec. 2022, https://doi.org/10.56979/401/2022/53.
[48] B. Shetty, R. Fernandes, A. P. Rodrigues, R. Chengoden, S. Bhattacharya, and K. Lakshmanna, "Skin lesion classification of dermoscopic images using machine learning and convolutional neural network," *Sci Rep*, vol. 12, no. 1, p. 18134, Oct. 2022, https://doi.org/10.1038/s41598-022-22644-9.
[49] X. Bai, J. Li, C. Zhang, H. Hu, and D. Gu, "Distracted driving behavior recognition based on improved MobileNetV2," *J. Electron. Imag.*, vol. 32, no. 05, Sep. 2023,https://doi.org/10.1117/1.JEI.32.5.053021.
[50] N. Ramya, "Application For Skin Disease Diagnosis System Using Convnet," *IJSREM*, vol. 7, no. 4, Apr. 2023, https://doi.org/10.55041/IJSREM19280.

BIOGRAPHY OF AUTHORS

Ainul Furqon obtained his bachelor's degree in Informatics Engineering from the Faculty of Engineering, Nurul Jadid University, Paiton, Probolinggo in 2024. He is a developer of computer vision technology, sentiment analysis, data mining, android developer and can be contacted via email: elfrand3@gmail.com.

Kamil Malik is a lecturer in the Informatics Engineering Study Program at Nurul Jadid University. He obtained his bachelor's degree in Informatics Engineering from Nurul Jadid College of Technology and his master's degree in Informatics Engineering from Surabaya Integrated Institute of Science and Technology (ISTTS). His research interests include computer vision, data analysis, and software development. He can be reached via email at Kamil@unuja.ac.id.

Fathorazi Nur Fajri is a lecturer at the Department of Informatics Engineering at the University of Nurul Jadid. He received a bachelor's degree in Informatics from Nurul Jadid College of Technology and a master's degree in Informatics from the University of Dian Nuswantoro. His research interests include computer vision, software development, and database design and analysis. He can be contacted at fathorazi@unuja.ac.id.