Infact: International Journal of Computers
Vol.9, No.01, Maret 2025
p-ISSN : 2527-8363, e-ISSN: 2829-0259
DOI:

The Enhanced Dermatological Diagnosis: Autoimmune and Non-Autoimmune Skin Disease Classification Using MobileNet and ResNet

Tyara Regina Nadya Putri[1]*, Agung Mulyo Widodo [2]
[1,2] Department of Computer Science, Universitas Esa Unggul
[1,2]Jl. Arjuna Utara No.9, Duri Kepa, Kec. Kb. Jeruk, Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta 11510

E-mail: tyararegina11@student.esaunggul.ac.id[1], agung.mulyo@esaunggul.ac.id[2]

Submitted: 05/03/2025 , Revision:24/03/2025, Accepted : 27/03/2025

Abstract

Autoimmune diseases arise when the immune system mistakenly attacks the body's healthy cells, causing a range of symptoms that can greatly affect a patient's quality of life. In Indonesia, these conditions present a significant public health concern. According to research by Ministry of Health Republic Indonesia in 2024, autoimmune lupus affects approximately 0.5% of the population, impacting over 1.3 million individuals. This study proposes a classification and detection model utilizing Convolutional Neural Networks (CNN) with transfer learning, incorporating MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152 architectures. The model's performance is assessed using a confusion matrix, evaluating precision, recall, and F1-score, while computational efficiency is analyzed using a GPU T4. Experimental results demonstrate that ResNet152 achieved the highest accuracy at 92%. These findings emphasize the crucial role of selecting an optimal CNN architecture to enhance the accuracy of autoimmune and non-autoimmune skin disease classification and detection.

Keywords:
Autoimmune
Deep Learning
Convolutional Neural Network (CNN)
MobileNet
ResNet

Corresponding Author: Tyara Regina Nadya Putri
E-mail: tyararegina11@student.esaunggul.ac.id

1. Introduction

Autoimmune diseases occur when the immune system attacks healthy cells, leading to various symptoms that can significantly impact quality of life. These conditions often cause chronic pain, fatigue, and skin rashes, leading to physical and emotional distress for patients. In Indonesia, autoimmune diseases pose a major public health challenge. According to research conducted by the Indonesian Ministry of Health, thse prevalence of lupus is estimated at 0.5%, affecting over 1.3 million people, predominantly women aged 15-45 years [1]. However, patient care remains limited due to factors such as inadequate healthcare facilities, lack of physician awareness, restricted medication availability, and insufficient pharmacies providing necessary drugs [2]. Technological advancements have improved access to health information, increasing public awareness of autoimmune diseases. Neurologist Dr. Rocksy Fransisca V Situmeang Sp.S noted a significant rise in awareness over the past decade, particularly in urban areas, as people actively seek information about their symptoms [3]. Despite this progress, many patients continue to experience prolonged discomfort due to late diagnosis and inadequate treatment options. Deep learning presents an innovative approach to improving early detection of autoimmune diseases, particularly those manifesting as skin rashes. Convolutional Neural Networks (CNN) have shown superior performance in analyzing large-scale medical images, distinguishing between autoimmune and non-autoimmune conditions. Convolutional Neural Networks (CNN) surpass traditional algorithms like K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) by automatically and efficiently extracting image features, making them highly effective for image classification tasks [4]. However, limited awareness of early detection results in insufficient image data for analysis. To address this, data augmentation techniques such as rotation, cropping, and color adjustments enhance dataset diversity, making models more robust. Data augmentation serves as a powerful technique to expand the dataset while simultaneously preventing overfitting, ensuring the model generalizes well to new data [5]. Additionally, transfer learning can expedite training and improve model accuracy by leveraging pre-trained datasets. Transfer learning in Convolutional Neural Networks (CNN) allows the use of knowledge from previously trained models to improve performance on new tasks.

CNN extracts features from images through convolution and pooling layers to detect local patterns such as edges or corners. After that, the fully connected and output layers classify the image based on the extracted features. CNNs are widely favored in deep learning for their capability to extract essential features and reduce dimensionality without compromising the core characteristics of an image, making them ideal for complex visual recognition tasks [6]. Convolutional Neural Networks (CNNs) have been widely utilized in medical imaging for disease detection, demonstrating remarkable success through transfer learning. From identifying skin lesions and mpox to diagnosing breast cancer and brain tumors [7], [8], [9], [10], CNN-based models have significantly enhanced accuracy and efficiency in medical diagnostics. By leveraging pre-trained architectures, these models can efficiently analyze complex medical images, enabling faster and more reliable disease classification. Several previous studies have explored the development of classification models for autoimmune diseases; however, their scope has been largely limited. Most of these studies focus on a single autoimmune disease, particularly psoriasis [11], without extending to a broader range of autoimmune conditions. Additionally, many existing models rely on fluoroscopy specimens [12], which, while effective, may not fully capture the diversity of skin manifestations across different autoimmune disorders. This limitation highlights the need for a more comprehensive approach that can classify multiple autoimmune skin diseases as well as non-autoimmune skin conditions that exhibit similar rash-like symptoms, using diverse medical imaging techniques.

Therefore, in this paper, we propose a deep learning-based classification model capable of distinguishing between multiple autoimmune and non-autoimmune skin diseases, addressing the limitations of prior studies that primarily focused on a single condition. Unlike previous works that rely on fluoroscopy specimens, our approach leverages diverse medical imaging, enhancing its applicability across various dermatological conditions. By employing convolutional neural networks (CNNs) such as MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, ResNet152 with additional layers, our model aims to improve diagnostic accuracy and broaden the scope of automated skin disease classification. The research involves data collection, annotation, preprocessing, and augmentation to enhance model robustness. Furthermore, hyperparameter optimization, including learning rate, epoch, and optimizer tuning, is applied to achieve the best model performance in classification and detection tasks.

This research is expected to contribute as follows:

- Enhancing early detection and diagnosis of skin diseases, enabling faster and more accurate treatment, which can significantly improve patient outcomes and quality of life.
- Supporting dermatologists and healthcare professionals by providing an AI-driven tool that streamlines the identification of autoimmune and non-autoimmune skin conditions, reducing diagnostic errors and improving efficiency.
- Driving advancements in medical image processing through the development of an optimized deep-learning approach, paving the way for more innovative and accessible skin disease detection technologies in the future.

2. Related Works

Several studies have explored the application of deep learning models for classifying skin diseases, particularly autoimmune conditions. One study utilized skin images to classify six categories, including normal skin and five types of psoriasis, with an initial dataset of 301 images, which expanded to 1,838 after augmentation. The study employed

the VGG-19 model which is known for having Multiple Convolutional Layers with small kernels, achieving an accuracy of 84%; however, the model exhibited overfitting, with a training accuracy of 97% and validation accuracy of 84% [13]. Another study focused on classifying psoriasis and lichen planus using ResNet-50 which is known for having Residual Blocks to overcome vanishing gradient, obtaining an accuracy of 89% with a balanced dataset of 1,836 images [14]. Further research classified eczema and psoriasis using an augmented dataset of 6,286 images. It tested pretrained models, achieving 60% accuracy with AlexNet, 58% with ResNet, and 82% with VGG-16. Additionally, a custom model named "Derma Care" was developed, outperforming the pretrained models with an accuracy of 88% [15]. Another approach analyzed autoimmune diseases using body cell images, testing multiple pretrained models. The results showed 88% accuracy with MobileNet which is known for having Depthwise Separable Convolution reduces parameters and computation and Pointwise Convolution improves feature efficiency, 92% with InceptionV3 which is known for having Inception Modules with various kernel sizes in one block, Factorized Convolutions to break up large convolutions, and Auxiliary Classifiers to speed up convergence, 95% with DenseNet-121 which is known for having Dense Blocks and Bottleneck Layers to reduce the number of parameters, and 78% with VGG-16 [16]. Different research classified six different autoimmune disease specimens employed AlexNet which is known for having Convolutional Layers with large kernels to extract important features without the need for many layers and achieved an accuracy of 96% [12]. Another research classified six categories, including normal skin and five psoriasis types, utilizing MobileNetV2 and VGGNet-19, achieving 92% and 93% accuracy, respectively. Furthermore, a custom model named "DWSCNN" was developed, surpassing pretrained models with an accuracy of 96% [11]. Although previous studies have shown promising results by utilizing various pretrained models that have significantly improved accuracy and mAP, most remain limited to specific classifications and have not addressed the challenge of distinguishing between autoimmune and non-autoimmune diseases with similar manifestations.

In comparison, our research not only leverages pretrained models but also incorporates additional layers and explores various experimental techniques to optimize model performance. By systematically adjusting the model architecture and implementing different training strategies, we aim to develop a model that achieves good fitting, attaining high accuracy while minimizing overfitting. This approach enables a more robust and reliable classification system, allowing for better differentiation of various skin conditions within a more complex medical context.

3. Methods

3.1 Dataset

The dataset used in this study was obtained from publicly accessible sources, including the ISIC 2020 Archive, DermNet, and Te Whatu Ora Health New Zealand. These platforms provide a vast collection of high-quality dermatological images that have been carefully curated for research and medical purposes. All images utilized in this study have been verified by expert dermatologists, reinforcing their reliability for deep learning-based classification models. The dataset used consists of 2764 images divided into 8 classes, including autoimmune diseases such as dermatomyositis with 128 images, lichen planus with 251 images, psoriasis with 667 images, and vitiligo with 368 images, as well as non-autoimmune diseases such as eczema with 512 images, herpes with 222 images, seborrheic keratosis with 208 images, and tinea with 408 images indicates that there is an imbalance in the number of datasets for each class. An imbalance in the number of dataset samples across classes can lead the model toward overfitting, a challenge that will be explored further in **Section 3**. **Figure 2** illustrates the different skin disease classes included in this study.

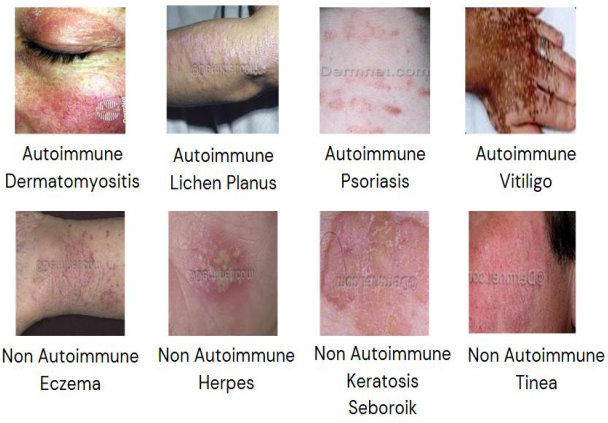

Figure 1 Various skin disease in proposed work

3.2 Processing Data

To overcome problems found in the dataset, such as variations in image size and imbalance in the number of images between classes, a preprocessing process is required which aims to prepare the data to be more consistent and optimal before being used in model training to improve model performance.

3.3 Data Augmentation

In machine learning and deep learning, both the quality and quantity of data play a crucial role in shaping the effectiveness of the model training process. Augmentation will increase the amount and diversity of training data by applying certain transformations to existing data. This augmentation technique is applied through certain transformations to existing data, the author performs rotate, flip left right, flip top bottom, zoom, crop, resize, brightness, and contrast transformations which aim to improve the model's generalization ability. In this study, the author uses the python library Augmentor by increasing the number of samples by 1500 for each class which will help develop a stronger model and be able to generalize better.

3.4 Data Distribution

The dataset is systematically divided into training, validation, and testing sets using a structured data partitioning method in model development. This approach ensures that each subset maintains a balanced class distribution, preserving diversity across the training and

testing phases. The data is allocated into three categories: 40% for training, 40% for validation, and 20% for testing, allowing the model to learn effectively while ensuring reliable performance evaluation.

3.5 Convolutional Neural Network

CNN as a subcategory of Artificial Neural Networks (ANN) is widely used in computer vision because of its ability to extract features and filter important information from images. CNN works by extracting features and filtering important information from image data through a series of convolution and pooling layers that aim to detect local patterns or features, such as edges or corners. Once the features are extracted, the network will proceed to the classification section which usually involves several fully connected layers and an output layer to classify images based on the features to produce class or category predictions. **Figure 3** represents a common CNN architecture that has been widely developed.

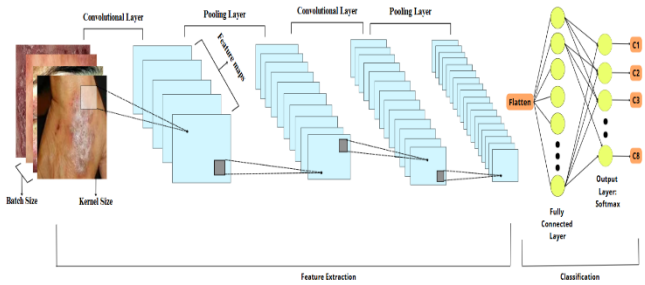

Figure 2 CNN Architecture

The CNN architecture consists of two key components:

The feature extraction layer, which captures essential patterns from input data, and the classification layer, which interprets these features to make predictions.

Feature Extraction

- **Input layer**

  The input layer stores the pixel values of the given image along with its color channels (RGB), serving as the foundation for further processing in the CNN architecture.

- **Convolutional layer**

  The convolutional layer extracts features from input data using filters (kernels) that slide across the image, performing convolution operations. This process generates feature maps that highlight essential information such as edges, textures, and shapes. **Figure 4** below shows the process of the convolution layer.

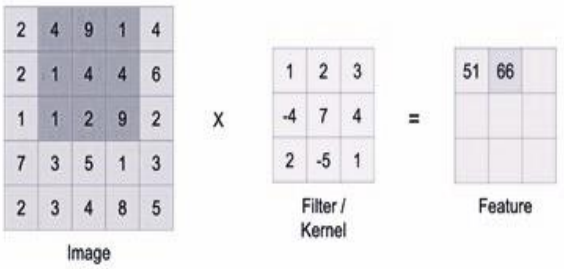

Figure 3 Overview of the convolutional layer process

- **Pooling Layer**

  The pooling layer minimizes the dimensions of the feature map produced by the convolutional layer by applying a filtering technique. In max pooling, this process extracts the highest value from each localized region of the feature map, retaining the most critical details, such as edges and key patterns within the image. **Figure 5** illustrates the differences between various types of pooling.

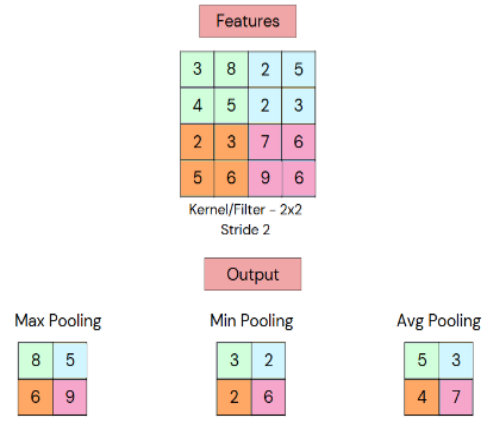

Figure 4 Overview of the pooling layer process

Classification

- **Flatten Layer**

  After passing through a series of convolutional and pooling layers, the feature maps typically retain high-dimensional structures. The Flatten layer serves to convert these high-dimensional representations into a one-dimensional vector, making them suitable for processing in subsequent fully connected layers. This layer acts as a bridge between the feature extraction phase and the decision-making process within the neural network.

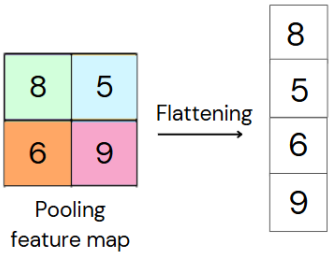

Figure 5 Overview of the flatten layer process

- **Fully Connected Layer**

  The Fully Connected layer establishes connections between every neuron in the preceding layer and each neuron in the subsequent layer, enabling the processing and classification of extracted features. It comprises multiple hidden layers integrated with activation functions such as ReLU, which activates only neurons with positive values to introduce non-linearity, while the output layer delivers the final prediction based on the refined data.

- **Output Layer**

  The Output Layer plays a crucial role in producing the final prediction, typically utilizing a softmax classifier

for multi-class classification. Each neuron in this layer corresponds to a specific class, with the output values representing the probability of the input being classified into each respective category.

3.6 Transfer Learning

Transfer learning is a machine learning technique that leverages a model pre-trained on a vast dataset to tackle a different yet related task. This technique accelerates training and enhances model performance by leveraging previously learned features, which can then be passed to additional layers for further training. One of the most widely used datasets for transfer learning is ImageNet, a widely utilized dataset in transfer learning, consists of millions of labeled images spanning thousands of categories, making it a valuable resource for model training. Pre-trained models on ImageNet can be utilized as feature extractors, where only the final layers are adjusted for the new task, or through fine-tuning, which allows for the modification of weights in some or all layers to improve accuracy on a smaller, domain-specific dataset. As shown in **Figure 7**.

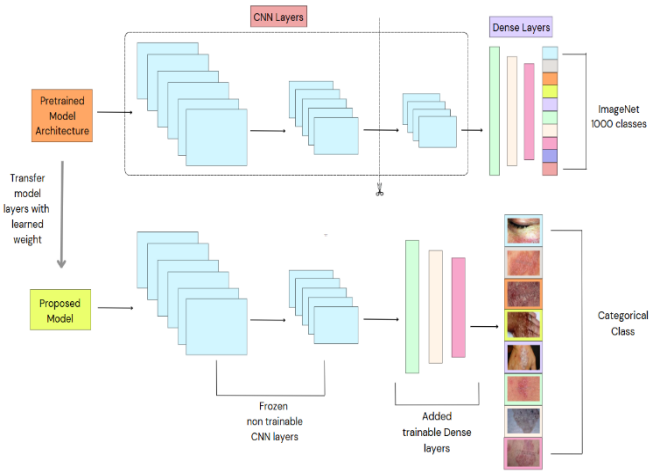

Figure 6 Transfer Learning

In this research there are 7 pretrained model architectures used:

1) MobileNetV2

MobileNetV2 introducing inverted residual blocks with linear bottlenecks, which helps maintain low computational costs while preserving representational power. Unlike traditional CNNs, MobileNetV2 employs depthwise separable convolutions to minimize parameter count and computational demands, enhancing efficiency without compromising performance. The network consists of 17 layers of inverted residual blocks, where each block features depthwise convolutions followed by 1x1 pointwise convolutions. This structure enhances feature reuse and gradient flow.

2) MobileNetV3Small

MobileNetV3 Small is an optimized version of MobileNetV3 that integrates squeeze-and-excitation (SE) modules for better channel-wise attention and uses hard-swish (h-swish) activation functions to improving non-linearity while maintaining efficiency. The architecture includes bottleneck residual blocks, but compared to MobileNetV2, it reduces the number of layers and parameters, making it more suitable for real-time applications with limited processing power.

3) MobileNetV3Large

MobileNetV3 Large is a more powerful version of MobileNetV3 that retains the squeeze-and-excitation (SE) modules, hard-swish activations, and depthwise separable convolutions from MobileNetV3 Small but incorporates more bottleneck layers for improved feature extraction. The architecture consists of larger kernel sizes and more activation layers, allowing it to process complex patterns while still being optimized for mobile applications.

4) ResNet50

ResNet50 is a deep convolutional neural network with 50 layers, comprising convolutional layers, batch normalization, ReLU activations, and fully connected layers. What makes ResNet unique is its residual connections, or skip connections, which enable the model to bypass specific layers and transfer information directly. This mechanism helps mitigate the vanishing gradient problem in deep networks, allowing for more efficient training. ResNet50 is structured into four stages of residual blocks, each containing three bottleneck layers that optimize parameter efficiency while preserving powerful feature extraction capabilities.

5) ResNet101

ResNet101 extends the architecture of ResNet50 by increasing its depth to 101 layers, consisting of 33 bottleneck residual blocks. The deeper architecture allows the model to learn more complex patterns and hierarchical features from images, improving its performance on high-resolution and large-scale datasets.

6) ResNet152

ResNet152 is an even deeper variant of the ResNet family, featuring 152 layers with 50 bottleneck residual blocks, making it one of the deepest networks available for feature extraction. Its extreme depth enhances the model's ability to learn highly abstract representations, making it ideal for highly complex tasks such as satellite image analysis, autonomous driving, and industrial defect detection.

3.7 System Design

Figure 8 shows the framework used to implement this research.

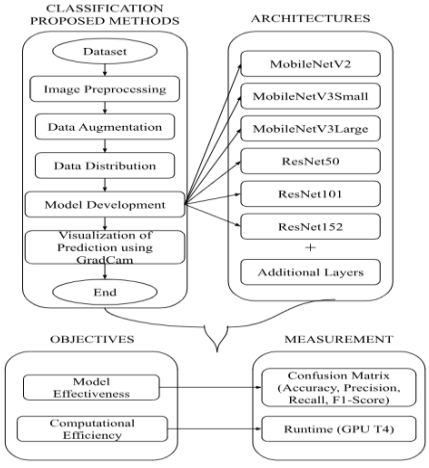

Figure 7 Framework

Based on the framework in the **figure 7**, this study employs two primary methods: classification and object detection, aiming to evaluate both model effectiveness and computational efficiency. The classification method begins with dataset collection, followed by image preprocessing to enhance data quality and data augmentation to improve model generalization. The processed dataset is then split into training, validation, and test sets. The model development utilizes various CNN architectures, including MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152, integrated with additional layers such as Flatten, Fully Connected, Dropout, and Output layers. Grad-CAM is employed to provide interpretability by visualizing model decisions.

The effectiveness of the models is evaluated using metrics such as accuracy, precision, recall, and F1-score, while computational efficiency is measured based on runtime performance on GPU (T4). This approach ensures optimal performance in both prediction accuracy and resource utilization.

1. Proposed Classification Model Architecture

This study focuses on developing a classification model for distinguishing between autoimmune and non-autoimmune skin diseases using image-based analysis. The model utilizes a pre-trained CNN architecture as the input layer to leverage transfer learning, enhancing feature extraction and accelerating training. To refine the classification process, eight additional layers are incorporated, including Flatten, Dropout, BatchNormalization, Fully Connected layers with Dense + ReLU activation, and an Output layer with a Softmax classifier. These layers improve the model's robustness, reduce overfitting, and optimize classification accuracy. **Figure 8** provides an overview of the model architecture used in this study.

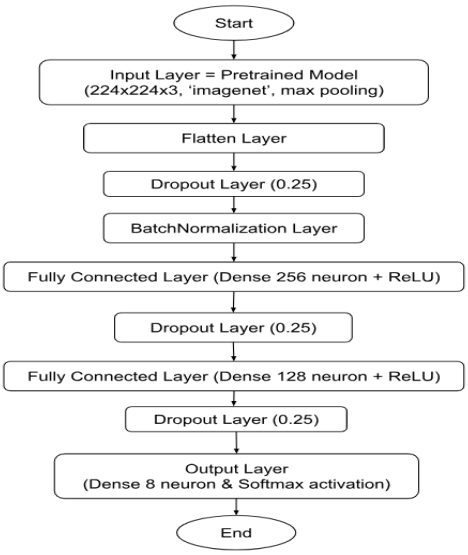

Figure 8 Classification Model Development

- **Input layer:** The model starts with a pre-trained CNN architecture such as MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152 trained on ImageNet, designed to process input images with dimensions 224×224 pixels, where 224×224 represents the spatial resolution and 3 channels correspond to the RGB color space. This layer leverages transfer learning to extract meaningful features from skin disease images efficiently, utilizing max pooling for spatial downsampling.
- **Flatten:** Transforms the multi-dimensional feature maps from the pre-trained model into a one-dimensional vector, ensuring compatibility with the fully connected layers for further processing.
- **Dropout Layer (0.25):** Applies a 25% dropout rate to minimize overfitting by randomly disabling neurons during training, enhancing the model's ability to generalize effectively.
- **BatchNormalization Layer:** Normalizes activations across the batch, stabilizing training by reducing internal covariate shifts and improving convergence speed.
- **Fully Connected Layer (Dense 256 + ReLU):** A dense layer with 256 neurons and ReLU activation, responsible for learning high-level representations from the extracted features.
- **Dropout Layer (0.25):** Another dropout layer with a 25% rate, maintaining model regularization to enhance robustness.
- **Fully Connected Layer (Dense 128 + ReLU):** Incorporates a dense layer with 128 neurons and a ReLU activation function, optimizing feature representation and improving classification performance.
- **Dropout Layer (0.25):** An additional dropout layer with 25% rate, reducing dependency on specific neurons and improving model generalization.
- **Output Layer (Dense 8 + Softmax):** The final fully connected layer with 8 neurons corresponds to the number of disease classes, utilizing a Softmax activation function to output probability distributions for classification.

3.8 Training Model

To avoid overfitting in improving models, techniques such as **early stopping** are used in training deep learning models. This technique prevents the model from being overtrained by stopping the training when the data performance no longer improves. In addition, **model checkpoints** automatically save the best model during training, so that the best performing model can be used without retraining. The implementation of **callback accuracy targets** is also used to manually stop training when the model has reached the set accuracy target.

3.9 Evaluation of Model Effectiveness

The confusion matrix is a crucial evaluation tool for assessing the performance of a classification model. It presents a matrix comparison between predicted and actual values from the test data, offering insights into how effectively the model classifies each class. By utilizing a confusion matrix, the number of correct and incorrect predictions for each category is systematically displayed, allowing for the calculation of key evaluation metrics such as accuracy, precision, recall, and F1-score [17]. his matrix comprises four primary components that illustrate the relationship between model predictions and actual

classifications: True Positive (TP), representing correctly predicted positive instances; True Negative (TN), indicating correctly identified negative instances; False Positive (FP), where negative instances are misclassified as positive; and False Negative (FN), where positive instances are mistakenly labeled as negative. A visual representation of the confusion matrix is provided in **Figure 9**.

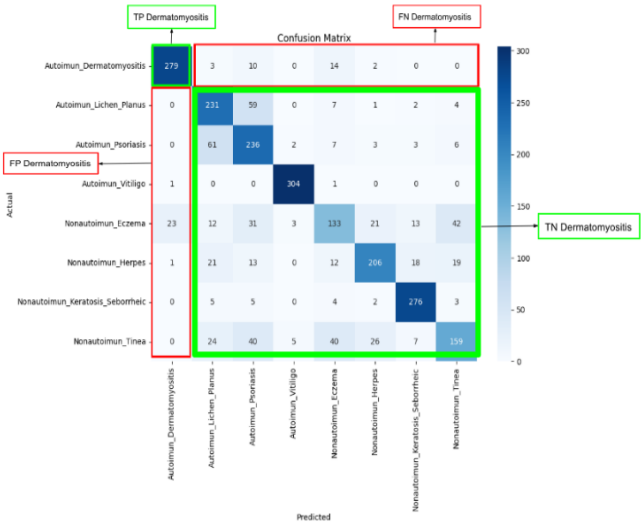

Figure 9 Confusion Matrix

This research assesses system performance based on F1 score, accuracy, recall, and precision. The calculation results of these performance measurements are shown in (1), (2), (3), and (4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + +FN} \quad (3)$$

$$F1 - Score = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

3.10Evaluation of Model Efficiency

Each model development in this study uses the T4 GPU, which is known to be efficient in execution time for deep learning training and inference. With its power-efficient and high-performance architecture, the T4 GPU enables fast and optimal data processing. An overview use of T4 GPU as runtime type is shown in **Figure 10**.

```
1 !nvidia-smi
```

```
Tue Jan 21 17:17:41 2025
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   37C    P8              12W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Figure 10 GPU T4

This study utilizes the pretrained architectures MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152, which have different layer depths. These differences affect the complexity, number of output parameters, and execution time efficiency during training and inference.

4. Results and Discussion

The difference in data quantity between **Table 1** and **Table 2** highlights the impact of data augmentation in this study. **Table 1** presents the original dataset before augmentation, which consists of a limited number of images, potentially restricting the model's ability to generalize across diverse cases. In contrast, **Table 2** showcases the dataset after augmentation, where techniques such as rotation, flipping, scaling, and color adjustments have been applied to artificially expand the dataset. This augmentation process enhances model robustness by increasing variation within the training data, reducing overfitting, and improving overall classification and detection performance.

No.	Directory	Percent	Amount Data
1	Train	40%	1337
2	Validation	40%	891
3	Test	20%	558
	Total		2786

Table 1 Unaugmented Data Distribution

No.	Directory	Percent	Amount Data
1	Train	40%	5760
2	Validation	40%	3840
3	Test	20%	2400
	Total		12000

Table 2 Augmented Data Distribution

4.1 Experiment A Result (Augmented Data)

Experiment A utilized augmented data during both training and testing phases, ensuring the model learns from a more diverse dataset, thereby enhancing its generalization capability and robustness. In this study, hyperparameter tuning was conducted to optimize model performance during training. Key parameters such as batch size, learning rate, number of epochs, network layers, and optimizer type, as detailed in **Table 3**.

No	Hyperparameter	
1	Batch Size	32
2	Layers	9
3	Epoch	35 and 50
4	Optimizer	Adamax
5	Learning Rate	0.0001

Table 3 Experiment A Result

The model is trained using various CNN architectures, including MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152, with modifications applied to the structured layers on **Figure 8**. The training process utilizes the Adamax optimizer and is conducted with different epoch settings, specifically 35 and 50. A summary of the model's performance metrics, including accuracy and loss across these epoch variations, is

presented in **Table 4**.

Epoch	Architecture	Val_ Acc	Acc	Loss	Val_ Loss
35	MobileNetV2	0.7904	0.7909	0.5618	0.5520
	MobileNetV3 Small	0.7305	0.6728	0.8556	0.7297
	MobileNetV3 Large	0.8039	0.7610	0.6417	0.5300
	ResNet50	0.9057	0.9501	0.1482	0.3045
	ResNet101	0.8922	0.9259	0.2091	0.3207
	ResNet152	0.8982	0.9223	0.2146	0.3159
50	MobileNetV2	0.8203	0.8368	0.4418	0.4864
	MobileNetV3 Small	0.7766	0.7324	0.6980	0.6124
	MobileNetV3 Large	0.8299	0.8136	0.4865	0.4559
	ResNet50	0.9198	0.9699	0.0935	0.3010
	ResNet101	0.9167	0.9593	0.1189	0.2898
	ResNet152	0.9203	0.9637	0.1049	0.2828

Table 4 Training Result of Experiment A

The experimental results indicate that the ResNet152 model with 50 epochs delivers the best performance, as evidenced by the minimal gap between training and validation loss. This outcome occurs due to a significant reduction in both training and validation loss values, as illustrated in **Figure 11**.

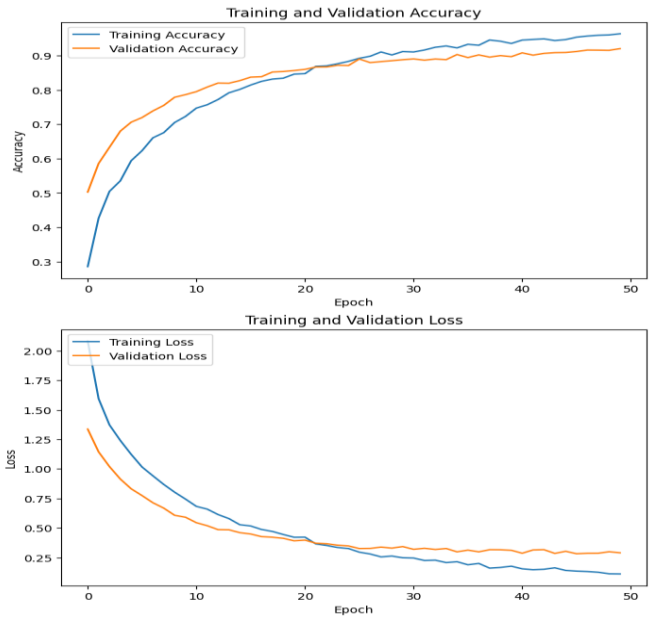

Figure 11 Accuracy and Loss Graphics of ResNet152 Model (50 Epoch)

At epoch 35, the MobileNetV2, MobileNetV3Large, ResNet50, ResNet101, and ResNet152 models exhibit good fitting conditions, maintaining a balance between training and validation performance. However, the MobileNetV3Small model shows signs of underfitting, indicating that it struggles to capture essential patterns from the data. By epoch 50, MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50 ResNet101, and ResNet152 continue to demonstrate good fitting, suggesting stable and well-generalized learning.

Using confusion matrices like accuracy, precision, recall, or F1-score, the model's performance is assessed at the evaluation stage following training. The results of the model evaluation experiments are shown in **Table 5.**

Epoch	Architecture	Accuracy	Precision	Recall	F1-Score
35	MobileNetV2	0.7904	0.80	0.78	0.78
	MobileNetV3 Small	0.7305	0.72	0.72	0.71
	MobileNetV3 Large	0.8039	0.78	0.76	0.75
	ResNet50	0.9057	0.90	0.90	0.89
	ResNet101	0.8922	0.90	0.89	0.89
	ResNet152	0.8982	0.89	0.88	0.88
50	MobileNetV2	0.8203	0.83	0.82	0.82
	MobileNetV3 Small	0.7766	0.75	0.75	0.75
	MobileNetV3 Large	0.8299	0.81	0.79	0.79
	ResNet50	0.9198	0.91	0.90	0.91
	ResNet101	0.9167	0.91	0.90	0.91
	ResNet152	0.9203	0.91	0.90	0.90

Table 5 Evaluation Result of Experiment A

In the first experiment with 35 epochs, the ResNet50 architecture showed 90% accuracy, with precision, recall, and F1-score values consistent at 0.90 for macro average and weighted average. This shows that ResNet50 is able to recognise all classes in the early stages of training. ResNet152 has an accuracy of 89%, slightly lower than ResNet50. In addition, ResNet50 shows stable performance with accuracy, recall, and F1 scores of 0.89 on weighted and macro averages. Then the ResNet101 architecture showed 89% accuracy, with precision, recall, and F1-score values consistent at 0.89 for macro average and weighted average. Then the MobileNetV3Large architecture showed 80% accuracy, with precision, recall, and F1-score values consistent at 0.78 for macro average and weighted average. Then the MobileNetV2 architecture showed 79% accuracy, with precision, recall, and F1-score values consistent at 0.80 for macro average and weighted average. Lastly, MobileNetV3Small has very poor performance showed 73% accuracy, with precision, recall, and F1-score values consistent at 0.72 for macro average and weighted average. This performance indicates that MobileNetV3Small has difficulty identifying data patterns in the early stages of training.

In the second experiment with 50 epochs, the ResNet152 architecture showed 92% accuracy, with precision, recall, and F1-score values consistent at 0.91 for macro average and weighted average. This shows that ResNet152 is able to recognise all classes in the early stages of training. ResNet50 has an accuracy of 91%, slightly lower than ResNet152. In addition, ResNet50 shows stable performance with accuracy, recall, and F1 scores of 0.91 on weighted and macro averages. The ResNet101 architecture achieved an accuracy of 91%, with precision, recall, and F1-score values consistently recorded at 0.91 for both the macro and weighted averages. Meanwhile, the MobileNetV3Large model demonstrated an accuracy of 82%, with its precision, recall, and F1-score maintaining a steady value of 0.81 for

both macro and weighted averages. Similarly, MobileNetV2

attained an accuracy of 82%, with precision, recall, and F1-score values consistently at 0.82 for both evaluation metrics. In contrast, MobileNetV3Small exhibited the weakest performance, reaching only 77% accuracy, with precision, recall, and F1-score values remaining at 0.75 for macro and weighted averages. This performance indicates that MobileNetV3Small has difficulty identifying data patterns in the early stages of training.

4.2 Experiment B Result (Non-Augmented Data)

Experiment B was conducted using non-augmented data for both training and testing, allowing an evaluation of the model's performance on raw, unaltered datasets. This approach helps assess how well the model generalizes without the benefits of increased data diversity. To ensure optimal performance during training, hyperparameter tuning was applied, adjusting key parameters such as batch size, learning rate, number of epochs, network layers, and optimizer type, as presented in **Table 6**.

No	Hyperparameter	
1	Batch Size	32
2	Layers	9
3	Epoch	50
4	Optimizer	Adamax
5	Learning Rate	0.0001

Table 6 Hyperparameter Experiment B

The model is trained using various CNN architectures, including MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152, with modifications applied to the structured layers on **Figure 9**. The training process utilizes the Adamax optimizer and is conducted with different epoch 50. A summary of the model's performance metrics, including accuracy and loss across these epoch variations, is presented in **Table 7**.

Epoch	Architecture	Val_ Acc	Acc	Loss	Val_ Loss
50	MobileNetV2	0.7531	0.8064	0.4978	0.6106
	MobileNetV3 Small	0.6889	0.6167	1.0377	0.8906
	MobileNetV3 Large	0.7568	0.7055	0.8081	0.7079
	ResNet50	0.8111	0.9379	0.2183	0.5335
	ResNet101	0.8077	0.8987	0.3142	0.5524
	ResNet152	0.7924	0.8912	0.2975	0.6080

Table 7 Training Result of Experiment B

The experimental results indicate that the ResNet50 model with 50 epochs delivers the best accuracy but not the best performance, as evidenced by the substansial gap between training and validation loss. This significant discrepancy suggests that while the model performs exceptionally well on the training data, its generalization to unseen validation data is limited, as shown in **Figure 12**.

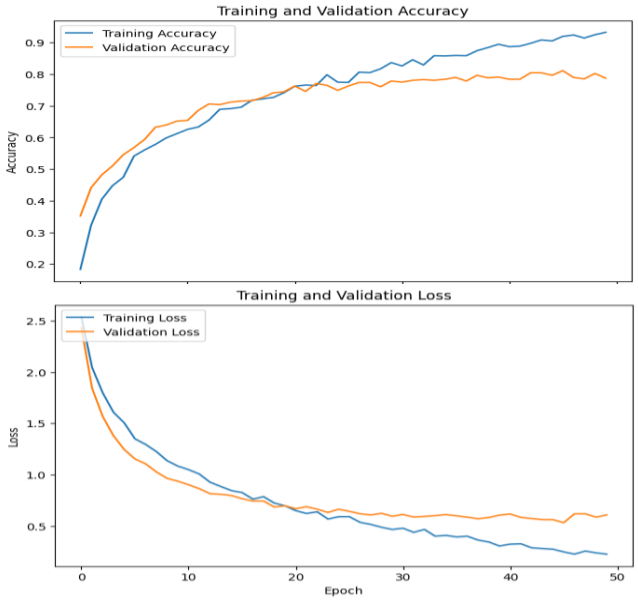

Figure 12 Accuracy and Loss Graphics of ResNet50 Model (Raw Data)

At epoch 50, the MobileNetV3Small model demonstrates signs of underfitting, struggling to capture complex patterns from the data, leading to suboptimal performance on both training and validation sets. In contrast, the MobileNetV2 and MobileNetV3Large models achieve a well-balanced state of good fitting, maintaining stable performance across training and validation. However, MobileNetV3Small continues to exhibit underfitting, indicating its limited learning capacity. Meanwhile, the ResNet50, ResNet101, and ResNet152 models show evident signs of overfitting, marked by a significant disparity between training and validation accuracy and loss values. This suggests that although these models excel on training data, their ability to generalize to new, unseen data is significantly reduced.

Using confusion matrices like accuracy, precision, recall, or F1-score, the model's performance is assessed at the evaluation stage following training. The results of the model evaluation experiments are shown in **Table 8**.

Epoch	Architecture	Accuracy	Precision	Recall	F1-Score
50	MobileNetV2	0.8203	0.73	0.71	0.72
	MobileNetV3 Small	0.7766	0.67	0.55	0.54
	MobileNetV3 Large	0.8299	0.71	0.64	0.66
	ResNet50	0.9198	0.80	0.80	0.80
	ResNet101	0.9167	0.80	0.79	0.79
	ResNet152	0.9203	0.80	0.77	0.78

Table 8 Evaluation Result of Experiment B

In the first experiment with 50 epochs, the ResNet50 architecture achieved an accuracy of 81%, with precision, recall, and F1-score values consistently at 0.80 for both macro and weighted averages. This indicates that ResNet50 can effectively recognize all classes in the early stages of training. ResNet101 followed closely with an accuracy of 80%, slightly lower than ResNet50, yet maintaining stable performance across all evaluation metrics. Meanwhile,

ResNet152 recorded an accuracy of 79%, with precision, recall, and F1-score values at 0.78 for both macro and weighted averages. Among the MobileNet models,

MobileNetV3Large reached an accuracy of 75%, with precision, recall, and F1-score values at 0.71, while MobileNetV2 demonstrated a similar accuracy of 75%, but with slightly better precision, recall, and F1-score values at 0.74. On the other hand, MobileNetV3Small exhibited the weakest performance, achieving only 68% accuracy, with all evaluation metrics consistently at 0.67. This result suggests that MobileNetV3Small struggles to identify meaningful data patterns in the early stages of training, leading to suboptimal performance.

4.3 Prediction Result

The prediction results of this classification model are visualized in **Figure 13**, which is represented by the help of the GradCam Heatmap tool for visualizing disease areas, how the model marks areas detected as skin diseases with a heatmap.

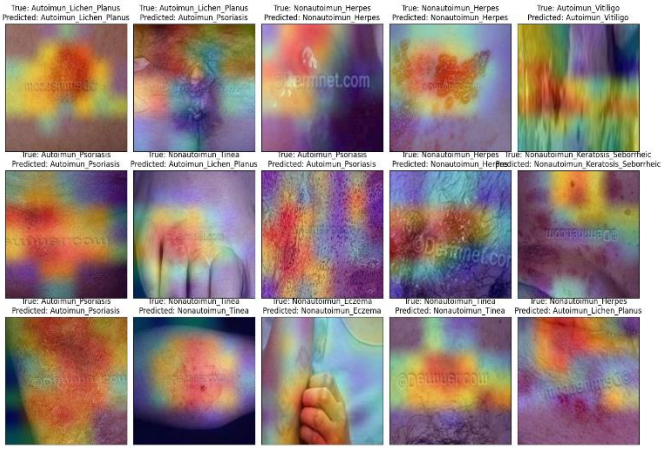

Figure 13 Predisction Result of Classification Model using GradCam Heatmap

4.4 Model Execution Results (Runtime)

In this study, MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152 are used as pretrained model architectures, where each architecture has a different layer depth that directly affects the complexity and number of output parameters produced. These differences contribute to variations in performance and efficiency of model execution time during the training and inference process. Runtime information during model development can be seen in the following **Table 10**.

No	Arsitektur	Trainable Parameters	Epoch	Runtime
1	MobileNetV2	1,416,904 (5.41 MB)	35	01:18:07
			50	02:34:51
			50	02:32:26
2	MobileNetV3Small	533,336 (2.03 MB)	35	01:09:34
			50	02:20:17
			50	02:15:35
3	MobileNetV3Large	1,234,744 (4.71 MB)	35	01:13:57
			50	02:40:29
			50	02:36:48
4	ResNet50	8,444,808 (32.21 MB)	35	01:22:51
			50	02:50:01
			50	02:44:17
5	ResNet101	8,444,808 (32.21 MB)	35	01:43:47
			50	03:10:11
			50	03:06:32
6	ResNet152	8,444,808 (32.21 MB)	35	01:26:25
			50	03:02:43
			50	02:57:27

Table 9 Runtime Information

Based on **Table 10** which displays runtime information from various deep learning model architectures, the use of GPU as a runtime type helps speed up the model training process so that it is more efficient compared to CPU. This can be seen from the variation in training time which is influenced by the number of trainable parameters and the number of epochs used. The smaller the number of trainable parameters, the faster the model execution will be due to the lighter computational load. Conversely, the larger the number of trainable parameters, the longer the model execution time will be because it requires higher computational resources to process complex parameters.

4.5 Discussion

The experimental results demonstrate promising accuracy across models trained for 35 and 50 epochs using augmenteddata, with all models achieving a good fit. This indicates that the models effectively learned from both training and testing data, ensuring balanced generalization. However, an exception was observed in MobileNetV3-Small tested at 35 epochs, which exhibited underfitting, suggesting the model focused more on the testing data than the training data.

In contrast, when models were trained without data augmentation with 50 epochs, only MobileNetV2 and MobileNetV3-Large maintained a good fit. MobileNetV3-Small suffered from underfitting, while ResNet, ResNet101, and ResNet152 experienced overfitting, indicating that these models memorized the training data but struggled to generalize well to unseen data. These findings highlight the importance of data augmentation in achieving optimal model performance and preventing issues related to underfitting and overfitting. These discrepancies can be attributed to the limited data variation and the absence of data balancing, which contributed to the tendencies toward underfitting and overfitting.

One of the key limitations of this study is the imbalance in the number of samples collected for each class, leading to an uneven distribution of data. This imbalance poses significant challenges, as it can cause models to suffer from underfitting, where they fail to capture essential patterns, or overfitting, where they become overly reliant on the training data and struggle to generalize. Additionally, certain classes may be disproportionately represented, increasing the risk of model bias.

As a result, some skin conditions might be classified more accurately than others, potentially affecting the model's reliability in real-world applications. This bias is particularly evident in the autoimmune psoriasis and non-autoimmune eczema classes, as these two categories had the highest

number of samples before augmentation. Consequently, after augmentation, the generated images lacked sufficient diversity compared to other classes. This lack of variation may cause the model to become more proficient at distinguishing these two conditions while struggling to
accurately classify less-represented skin diseases. As a result, the model's decision-making process could be skewed toward these dominant classes, potentially reducing its overall effectiveness in identifying rarer conditions with equal precision. Based on the precision and recall information in **Table 11**, it is evident that there is bias in the autoimmune psoriasis and non-autoimmune eczema classes. The autoimmune psoriasis class shows a lower precision than recall, indicating that the model frequently classifies samples as psoriasis, even at the risk of misclassification. Conversely, the non-autoimmune eczema class exhibits bias in the opposite direction, with a higher precision than recall, meaning the model is more selective in classifying samples as eczema but may fail to detect some actual cases. This imbalance reflects the dominance of certain data during training, which can impact the model's ability to generalize classification across other categories.

No	Class	Precision	Recall	F1-Score
1	Autoimun Dermatomyositis	0.96	1.00	0.97
2	Autoimun Lichen Planus	0.95	0.87	0.91
3	Autoimun Psoriasis	0.73	0.97	0.83
4	Autoimun Vitiligo	0.99	0.99	0.99
5	Non Autoimun Eczema	0.91	0.75	0.82
6	Non Autoimun Herepes	0.94	0.90	0.92
7	Non Autoimun Keratosis Seborrheic	0.96	0.97	0.96
8	Non Autoimun Tinea	0.87	0.77	0.82

Table 10 Precision, Recall, and F1-Score of Each Class

5. Conclusions and Future Works

This study presents the development of a classification and detection model for autoimmune and non-autoimmune skin diseases using deep learning. The classification model utilizes Convolutional Neural Networks (CNN) with transfer learning, including six pretrained architectures: MobileNetV2, MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet101, and ResNet152. Grad-CAM Heatmap is applied to visualize affected areas. Data augmentation significantly improved model performance, increasing the dataset from 2,764 to 12,000 images, reducing overfitting, and enhancing generalization. This is proven by the results of the development of this classification model producing the highest accuracy by ResNet152 of 92% in goodfit conditions. Next by ResNet50 with an accuracy of 91% in overfit conditions. Followed by ResNet101 with an accuracy of 91% in goodfit conditions. Next is MobileNetV3Large with an accuracy of 82% in goodfit conditions. Continued by MobileNetV2 with an accuracy of 82% in goodfit conditions. Lastly, MobileNetV3Small with an accuracy of 77% in underfitting conditions. From the training results, the ResNet152 model with 50 epoch iterations and using data that has been augmented gave the best performance with an accuracy of 92%, making it the

model with the highest accuracy among the six models tested with validation results using the Confusion Matrix showing a model evaluation value with a precision of 91%, a recall of 90%, and an F1-Score of 90%.

Future research can focus on increasing dataset size, balance the amount of data for each class to prevent bias, exploring additional pretrained models and optimizers, incorporating image segmentation techniques like U-Net and Mask R-CNN, implement YOLO for real-time skin disease detection, leveraging its speed and accuracy to enhance model practicality in medical applications, and deploying the model in a web-based application for practical use.

Acknowledgments

I would like to express my deepest gratitude to my supervisors for their invaluable guidance, patience, and unwavering support throughout this research journey. The expertise and thoughtful insights have been instrumental in shaping the direction of this study and ensuring its successful completion. My heartfelt appreciation also goes to my family and friends for their constant encouragement, as well as to my classmates and colleagues for their collaboration and support. Their belief in my work has been a continuous source of motivation. Finally, I extend my sincere thanks to everyone who has contributed to this research, whether directly or indirectly. Your support and encouragement have been truly meaningful, and I am immensely grateful.

References:

[1] A. Muhawarman, "Kemenkes Tingkatkan Upaya Deteksi Dini Lupus Melalui Program SALURI," Kemenkes. [Online]. Available: https://www.kemkes.go.id/id/kemenkes-tingkatkan-upaya-deteksi-dini-lupus-melalui-program-saluri

[2] Hayati, "Tren Lupus di Indonesia Meningkat, Pasien Rujuk Balik Baru 2.000-an Orang," 2023.

[3] A. Faisal, "Kesadaran masyarakat perkotaan terhadap penyakit autoimun semakin baik," ANTARA. Accessed: Jan. 14, 2025. [Online]. Available: https://www.antaranews.com/berita/4125207/kesadaran-masyarakat-perkotaan-terhadap-penyakit-autoimun-semakin-baik

[4] M. F. Naufal, "Analisis Perbandingan Algoritma SVM , KNN , dan CNN untuk Klasifikasi Citra," no. March 2021, 2021, doi: 10.25126/jtiik.202184553.

[5] L. Alzubaidi *et al.*, *Review of deep learning : concepts , CNN architectures , challenges , applications , future directions*. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.

[6] Y. Omori and Y. Shima, "Image Augmentation for Eye Contact Detection Based on Combination of Pre-trained Alex-Net CNN and SVM," vol. 15, no. 3, pp. 85–97, 2020, doi: 10.17706/jcp.15.3.

[7] S. M. and H.-S. P. Thwin, “Skin Lesion Classification Using a Deep Ensemble Model,” 2024.
[8] M. Pal *et al.*, “Deep and Transfer Learning Approaches for Automated Early Detection of Monkeypox (Mpox) Alongside Other Similar Skin Lesions and Their Classification,” 2023, doi: 10.1021/acsomega.3c02784.
[9] M. Nawaz, A. A. Sewissy, and T. H. A. Soliman, “Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network,” vol. 9, no. 6, pp. 316–322, 2018.
[10] M. Rasool *et al.*, “A Hybrid Deep Learning Model for Brain Tumour Classification,” 2022.
[11] N. Bhaswanth, “Psoriasis Classification of Different Types Based on Deep Learning Technique,” vol. 3, pp. 3445–3457, 2024.
[12] D. Cascio, V. Taormina, and G. Raso, “Deep CNN for IIF Images Classification in Autoimmune Diagnostics,” 2019.
[13] S. F. Aijaz, S. J. Khan, F. Azim, C. S. Shakeel, and U. Hassan, “Deep Learning Application for Effective Classification of Different Types of Psoriasis,” vol. 2022, 2022.
[14] A. Eskandari and M. Sharbatdar, “Efficient diagnosis of psoriasis and lichen planus cutaneous diseases using deep learning approach,” *Sci. Rep.*, pp. 1–18, 2024, doi: 10.1038/s41598-024-60526-4.
[15] M. Hammad, P. Pławiak, M. Elaffendi, A. A. A. El-latif, and A. A. A. Latif, “Enhanced Deep Learning Approach for Accurate Eczema and Psoriasis Skin Detection,” 2023.
[16] F. Muhammad *et al.*, “Application of the Deep Convolutional Neural Network for the Classification of Auto Immune Diseases,” 2023, doi: 10.32604/cmc.2023.038748.
[17] M. Heydarian and T. E. Doyle, “MLCM : Multi-Label Confusion Matrix,” *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.