Institute of Research and Publication Indonesia (IRPI)
Public Research Journal of Engineering, Data Technology and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/predatecs
Vol. 2 Iss. 2 January 2025, pp: 93-106
ISSN(P): 3024-921X | ISSN(E): 3024-8043

A Deep Learning Approach Bassed on Classification to Detect Facial Skin Defect

Alika Rahmarsyarah Rizalde[1*], Haykal Alya Mubarak[2], Batrisia Khairunnisa[3], Mohd. Adzka Fatan[4]

[1,2]Department of Information System, Faculty of Science and Technology,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia
[3]Department of Mimarlık, Faculty Mimarlık, Haliç üniversitesi, Turkey
[4]Department of Dirasat Islamiyah, Faculty of Al Qur'an Al Karim,
University of Qur'an Al-Karim and Islamic Sciences, Yaman

E-Mail: [1]rahmarsyaraha@gmail.com, [2]haykalalya82@gmail.com,
[3]batrisiakh@gmail.com, [4]Akafatan05@gmail.com

Received Jul 17th 2024; Revised Nov 25th 2024; Accepted Dec 28th 2024; Available Online Jan 12th 2025, Published Jan 18th 2025
Corresponding Author: Alika Rahmarsyarah Rizalde

Abstract

As people are more active, facial skin is often neglected, which can lead to acne, eye bags, and redness. In this study, deep learning models such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Networks (GANs) are used to classify facial skin damage. DenseNet201 and MobileNetV2 architectures were also used to evaluate the models in this study. The dataset used consists of facial skin disease photos collected from the Kaggle database. The model was trained and tested to classify the types of skin damage after going through data collection and preprocessing stages. The results showed that the GANs model and the DenseNet201 and MobileNetV2 architectures were the best models, with test accuracy values of 89% for the GANs model, 88% for the DenseNet201 architecture, and 89% for the MobileNetV2 architecture. These results show that deep learning approach techniques can help classify and find facial skin problems well. and it is expected that it will be a great progress in the field of dermatology and skin health.

Keyword: Convolutional Neural Network, Facial Skin, Generative Adversarial Networks, Recurrent Neural Network

1. INTRODUCTION

With the increasingly dense activities of society, some people are less concerned with the health of their facial skin, therefore it can lead to unhealthy facial skin. Problems found on facial skin can be such as acne, eye bags, and redness on facial skin. Acne is called a chronic inflammatory disease with a high prevalence around the world that can reach 80% of people experience acne at some time in their lives which usually begins to appear at the age of 15 and 17 years and this condition can continue into adulthood [1]. The age-adjusted prevalence of acne vulgaris is about 25% higher in females compared to males, which is about 10,911.8 per 100,000 population for females and 8,727.8 per 100,000 population for males based on data from the 2021 Global Burden of Disease Study (GBD) for individuals aged 10-24 years in 204 countries and territories from 1990 to 2021 [2]. Eye bags can appear due to hectic activities which result in less rest time and messy sleeping hours. In addition, facial skin redness is one of the topics often discussed in beauty articles or online consultations on health sites [3]. Facial skin redness can arise due to irritation or allergies to something that has been consumed. Redness of the facial skin can arise due to irritation or allergies to something that has been consumed. Every individual has different skin characteristics, including color, texture and sensitivity. These variations can affected how skin damage, such as acne or redness, appears and is detected. Other than individual characteristics, external factors such as sun exposure, pollution, and the use of cosmetic products also play an important role in skin health.

From some of these facial skin problems, it can classify various types of facial damage, such as acne, eye bags, and redness on facial skin through a deep learning approach. Deep Learning (DL) is a machine learning technique that can teach computers to learn through examples, and its use is growing exponentially as technology and device specifications improve [4]. DL is a subfield of Machine Learning (ML), which has shown particular success in performing image processing, improving the accuracy of classification tasks and reducing variability in regression problems [5]. Deep learning has also shown great potential in various applications, including image analysis and pattern recognition.

There are several types of Deep Learning that will be used in this research process such as the Convolutional Neural Network (CNN) algorithm, CNN is an algorithm inspired by the neural mechanisms underlying the visual system, making it very appropriate for computer vision and recognition tasks. CNNs are known to be specifically designed to handle high-dimensional inputs, such as images, and can automatically learn relevant features from the data [5]. CNNs can play an important role for some deep model architecture tasks [6]. Then, a Recurrent Neural Network (RNN) algorithm that has the ability to collect records of each pattern with each dependent pattern. This RNN also uses a convolution layer to be able to predict the pixel neighborhood. This model analyzes temporal information based on a sequence of data that cannot be achieved by CNN models [7]. In the implementation of CNN, it can use image data augmentation which aims to improve the accuracy of the CNN model [8].

Furthermore, using Generative Adversarial Networks (GANs) algorithms are known to be able to generate a realistic image while formulating the problem using direct objective functions. These deep learning advances collectively demonstrate the diverse and influential applications of GANs in the field of artificial intelligence and beyond [9]. After that, research conducted by Musa and Badmos in 2022 that makes use of the MobileNetV2 architecture in detecting face shields that achieve an accuracy of 96% can be said that the performance of the MobileNetV2 architecture is quite good [10]. The algorithms mentioned above have been widely used for research using image classification data. This research also leveraging the architecture with DenseNet201 and MobileNetV2 models to perform model evaluation. Using this architecture requires a memory that does not take up much space and the accuracy gained by both architectures can reach the highest accuracy and there are no computational constraints.

In previous research conducted by P. D. Rinanda, et al (2024) with research results that show the optimal architecture, namely VGG16, getting 96.87% accuracy, then Inception V3 with 96.50% and the last CNN 81%. In the research to be done this can provide an update by applying a completely different architecture and using 3 algorithms in order to get more optimal results. The purpose of this research is to apply a Deep Learning approach for the classification of health problems or defects in facial skin by implementing CNN, RNN, and Gans. By applying these three algorithms and both architectures to skin disease data, this research is expected to classify and identify images of defects in facial skin and can help us to understand facial skin recognition, or detect objects, and can open up new ideas for research and innovation.

2. MATERIAL AND METHOD

2.1. Stage of Research

Several steps were taken in this study: gathering data, pre-processing the data, modeling the data using three algorithms, each with two architectures, and assessing the accuracy of the test's best findings, as indicated in Figure 1.

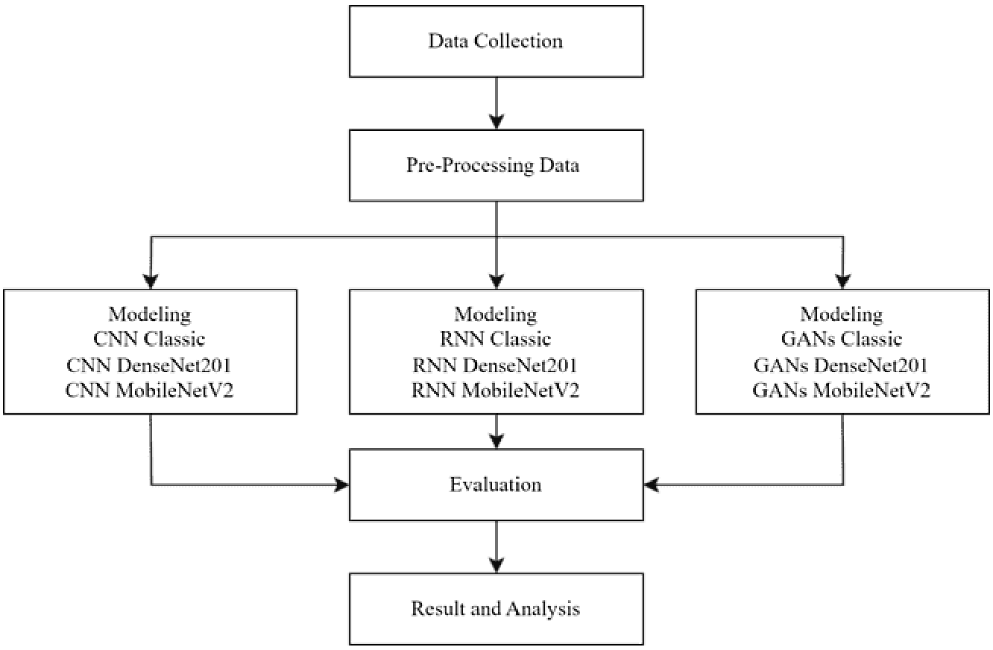

Figure 1. Research Methodology

2.2. Data Collection

Data were collected using the kaggle website, which is data in the form of facial skin defects images that have 3 classes. The resulting image is shown in three sides of the face and from the image classification can be done.

2.3. Pre-Processing Data

Data preprocessing is an important step in this research which is done by augmenting the image data. The benefits of data preprocessing are to improve data quality, model performance, and ease of model implementation.

2.4. Facial Skin Defect

Facial skin defects refer to various skin problems that occur in the facial area, which can affect a person's appearance, health, and psychology. Common skin problems on the face include redness, eye bags, and acne. The complicated pathophysiology of acne vulgaris, also known as acne, involves sebaceous glands, follicular hyperkeratinization, excessive bacterial colonization, immunological response, and inflammation [11]. Genetics, menstrual cycle hormone activity, stress, hyperactive Sebasasic gland activity, cleanliness, diet, and cosmetic use are all known acne triggers [12]. Then there are eye bags, a condition where the area under the eyes is swollen or has a dark discoloration. The main causes of eye bags are aging, fluid retention, lack of sleep, allergies, and genetics. As we age, the tissues and muscles around the eyes weaken, causing fat to migrate to the under-eye area. The impact of these eye bags on average has the ability to reduce self-confidence. Facial redness can be caused by various skin conditions, such as rosacea, dermatitis, allergies, or environmental conditions such as sun and wind exertion. Redness is often accompanied with inflammation and itching.

2.5. Deep Learning

Many facets of contemporary life have benefited from machine learning, including web browsing, social network content screening, e-commerce website recommendations, and the growing usage of machine learning in consumer goods like cameras and cellphones. One of the subfields of machine learning is deep learning [13]. Deep Learning is making significant strides toward resolving issues that have hampered the artificial intelligence community's best efforts for years. mostly employed in natural language processing, object identification [14], and picture classification. Its ability to identify intricate structures in high-dimensional data has significantly increased, making it useful in a wide range of scientific, commercial, and governmental contexts. Also, deep learning has emerged as a highly popular and efficient technique for many uses in image identification, natural language processing, and other areas.

2.6. Convolutional Neural Networks (CNN) Model

In recent years, CNN have taken over the field of machine vision. CNNs are made up of several hidden layers, an output layer, and an input layer. A convolutional layer, a pooling layer, a fully connected layer, and a normalization (ReLU) layer are typically the hidden layers of CNNs [15]. Additionally, the convolutional layer extracts features by using the local correlation of the image's data [16]. These CNNs operate on the foundation of linear algebra. The fundamental method used to express data and weights is matrix vector multiplication [17]. CNNs have also been shown to have excellent performance for image classification [18]. For the collection of photos, each layer has a unique set of properties. CNNs also have fully connected layers that classify the output with one label per node [19]. The following formulas and symbols can be used to describe the CNN mathematical model :

Let $x \in R^{(}N \times M$ represent the input signal, where N and M are the number of channels and samples, respectively (e.g., for a single-channel signal, M = 1). The weight matrix of the lth convolutional layer is denoted by $Wl \in R^{K\times K\times D}l—1 \times Dl$, where K is the filter size, Dl—1) is the input channel, and Dl is the output channel. The lth layer's convolution procedure can be expressed as equation 1 [6].

$$z^l = f \sum D^{l-1} x * W^l) + b^l \tag{1}$$

2.7. Recurrent Neural Network (RNN) Model

Such sequential input is processed using a form of neural network called an RNN [20]. The output of the RNN at one stage will be re-entered as additional input for the following stage in an RNN sequence, in addition to the input. The RNN can execute deduction based on the data sequence that has been processed in the preceding steps thanks to this method. The length of the sequence in RNN is also called sequence length or timesteps [21]. RNN captures high-level label relationships while keeping the computing complexity under control. It was found that RNN significantly improved classification accuracy [22]. An RNN's mathematical model includes the calculation that is done at every time step as well as the update rules for the hidden states. The basic RNN model is expressed mathematically as follows: The input signal at time step t is represented as

$x_t \in R^D$, where D is the number of input dimensions or features. The input signal and the previous hidden state are used to calculate the hidden state h_t R^H at time stept, as view equation 2.

$$h_t = f(W_{H\times H} * h_{t-1} + W_{hx} * x_t + b_{th}), \tag{2}$$

2.8. Generative Adversarial Networks (GAN) Model

One of the deep learning methods and a rapidly expanding area of artificial intelligence, GANs, has advanced quickly and had a big impact on many different domains [23]. It is now common practice to generate fresh image data using GANs, especially when producing sketch images that resemble reference photos. This process involves the combination of samples such as points, lines, planes, and colors to create a sketch or imitation image of a particular object, such as a physical object or a human [24]. A system comprising two neural networks that compete with one another in a zero-sum game framework is used to build GANs. Complex image processing tasks like semantic segmentation, image translation, generative image modeling, image synthesis, data augmentation, and domain adaptation have all been modified for GANs [25]. Aiming to satisfy the objective function (3), GAN model training, also known as the "minmax game," takes into account the following: x denotes the true data; $pz\ (z)$ denotes the previous input noise; $G(z)$ denotes the generated image; $D(G(z))$ denotes the probability that the false image is correct; and $D(x)$ represents the probability that the original image is correct. The goal function of cGAN is quite similar to the original goal function, except that $D(x) \rightarrow D(x \mid y)$, and $D(G(z)) \rightarrow D(G(z \mid y))$, where y is the condition [26].

$$\text{MinGmaxDV}\ (D, G) = E_x{\sim}\text{pdata}(x)\ [\log(D(x))] + E_Z{\sim}\text{pz}\ (z)\ [\log(1 - D(G(z)))) \tag{3}$$

2.9. Dense Convolutional Network 201 (DenseNet201)

One Deep Learning architecture model that links each layer and its feature maps to all ensuing levels is the Dense Convolutional Network (DenseNet). All of the preceding layers' feature maps will be fed into the following layer. DenseNet feed-forward connects each layer/block to every other layer/block. On the other hand, a conventional convolutional network consisting of L layers contains L - 1 connections between each layer, and L (L + 1)/2 direct connections between the subsequent levels [27]. This model is constructed similarly to ResNet, but it feeds information from one layer to the next. A few intriguing benefits of DenseNet are that it decreases the number of parameters significantly, strengthens feature spreading, encourages feature reuse, and resolves the gradient-gradient problem [28]. Relevant features can also be extracted using DenseNet. The architecture of DenseNet 201 is shown in Figure 2.

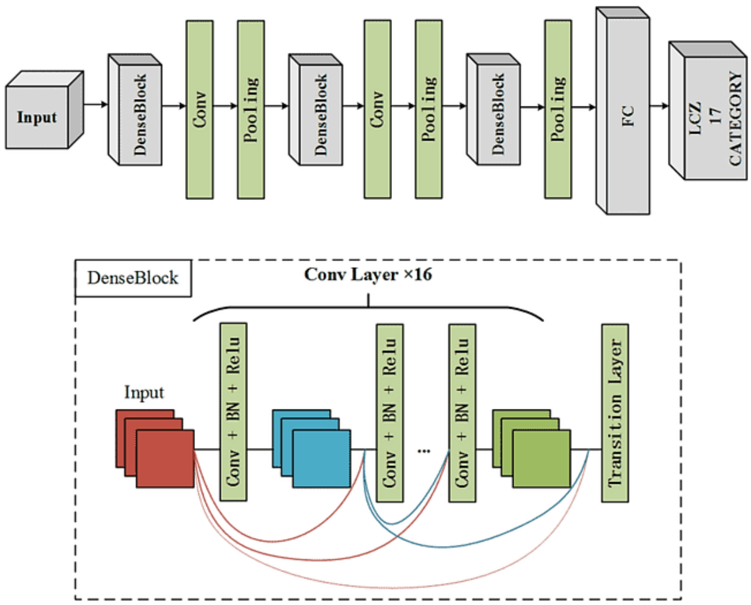

Figure 2. DenseNet201Architecture [29]

2.10. MobileNetV2

The foundational paradigm for the suggested approach is MobileNetV2. Depth-immersive convolution forms the foundation of the MobileNet architecture. Conventional 2D convolution performs convolution on the depth dimension (channel) in addition to processing all input channels directly to create a single output channel. Depthwise convolution divides the input and filter pictures into distinct channels, which are subsequently convoluted with the appropriate filter channel for each input channel. The output channels are restacked once they have been filtered [30].

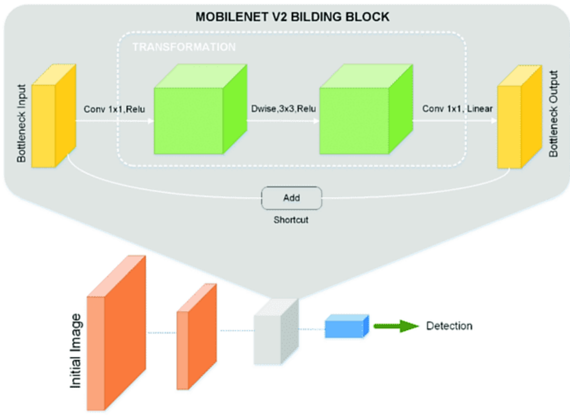

Figure 3. MobileNetV2 Architecture [31]

2.11. Evaluation

From the application of some of the algorithms described, evaluation is used to test the model with the best performance. The evaluation displayed can be in the form of accuracy, precision, recall and other evaluation results [32].

3. RESULTS AND DISCUSSION

3.1. Collecting and Preprocessing Data

The Kaggle database provided the image data used in this study, which depicts facial skin diseases. Three classes—acne, bags under the eyes, and redness of the facial skin—are identified in this image data. The model for identifying and categorizing face skin diseases will be trained and tested using these photos in this study. Since the raw data is still a bit erratic in terms of size and shape at first, leveling is crucial in order to maximize the results of subsequent computations. Figure 4 displays the dataset.

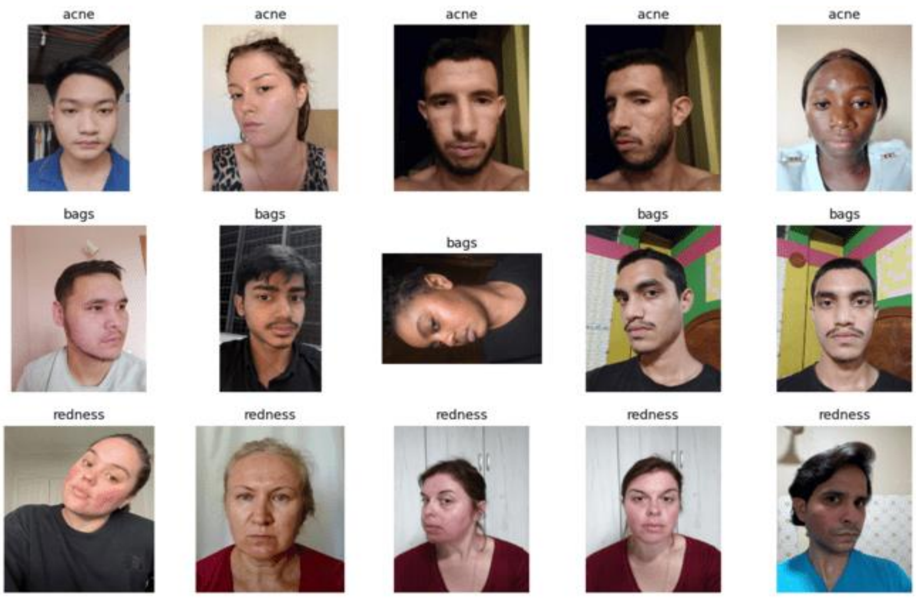

Figure 4. Dataset

90% of the raw data will be used for training and 10% for testing during the first round of data preparation. The separated data will be split up into distinct pathways. Furthermore, this research performs a data augmentation process on the image dataset by using ImageDataGenerator from Keras Tensorflow. The purpose of the augmentation process is to expand the amount of training data by creating variations from existing data. This technique is useful when the amount of data is limited because it helps the model generalize better by incorporating more variations into the training data.

In the training data augmentation performed includes rotating random images up to 45 degrees, rescaling image pixel values to a range of 0.1 by dividing each pixel by 255, deviation of random images with shear angles up to 15%, zooming random images up to 15%, flipping images vertically and horizontally and filling in empty pixels using the closest pixel value. Whereas in the test data, no augmentation process is carried out, only rescaling the pixel values and dividing the validation data. The next stage in data preprocessing is to determine the batch size to set the number of image samples that will be generated by the generator each time it is run.

The next step uses two important callbacks in model training with Keras TensorFlow, namely EarlyStopping to stop training if there is no decrease in val_loss for 50 consecutive epochs, while returning the model to the best weight and ModelCheckpoint stores the best model weight based on val_loss. As well as using Adam optimizer with learning rate 0.0001 to optimize the model using Keras TensorFlow with loss 'categorical_crossentropy' and metric 'accuracy'. The model was trained for 50 epochs. After all the processes carried out, the data is ready to be modeled with a more optimal algorithm.

3.2. CNN Model

In this modeling, CNN model with Keras TensorFlow is used to classify a 224x224 pixel image into three classes. The CNN architecture consists of convolution and pooling layers with filters of 32, 64, and two layers of 128, followed by 512 dense layer and a softmax output. In addition, experiments were conducted using DenseNet201 and MobileNetV2 as base models (pre-trained) without the top classification layer. These two models use GlobalAveragePooling2D, Dense (512, ReLU), Dropout (20%), and Dense (3, softmax) output layers for multi-class classification. And there are some settings used in classification testing which can be seen in table 1.

Table 1. The settings used in classifier training.

Parameter	Detail
Learning rate	0,0001
Epoch	50
Batch size	8
Optimizador	Adam
Loss function	Categorical Cross Entropy

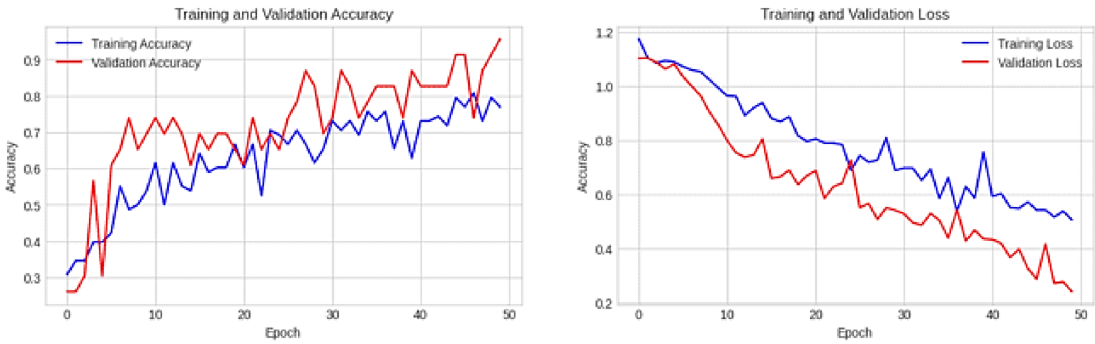

Figure 5. Training and Loss Graph of CNN Model

Table 2. Accuracy Result of CNN Model

	Training	Validation	Testing
Loss	0,53	0,24	0,40
Accuracy	0,77	0,96	0,78

The training performed on this model applies 50 epochs, and stops at the 50th epoch for the experiment graph can be seen in Figure 5. The figure shows the training accuracy value increases significantly, but for the loss value decreases as the number of epochs increases. For the accuracy results obtained on the testing data, which is 0.78 or 78%, the accuracy results obtained are quite optimal, as can be seen in Table 2.

3.2.1. CNN DenseNet201

Previously, the DenseNet201 architecture was trained on the ImageNet dataset for modeling purposes. It then adds two Dense layers with 256 neurons with ReLU activation, a GlobalAveragePooling2D layer, and a Dropout layer with a ratio of 0.3 to lessen overfitting. Lastly, the model features a Dense output layer with three neurons and softmax activation for multi-class classification. Figure 6 and Table 3 present the findings from the 50 epoch experiment.

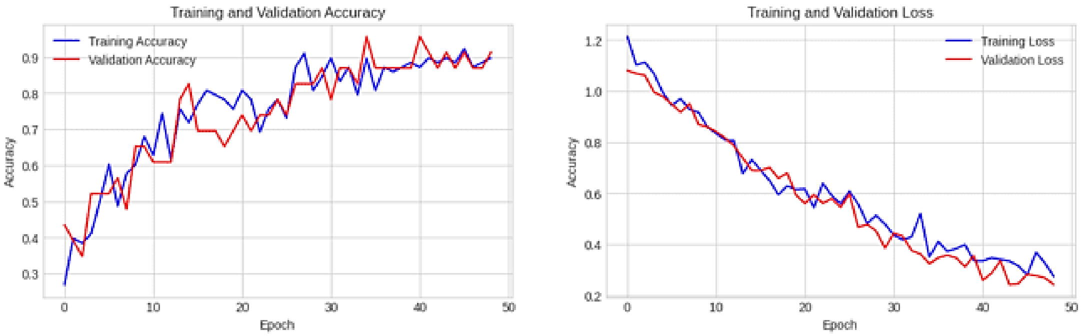

Figure 6. Training and Loss Graph of CNN DenseNet201

Table 3. Accuracy Result of CNN DenseNet201

	Training	Validation	Testing
Loss	0,24	0,24	0,35
Accuracy	0,96	0,91	0,89

The training performed on this model applies 50 epochs, but stops at the 50th epoch for the experiment graph can be seen in Figure 6. in the figure shows the training accuracy value also increases significantly, but for the loss value the graph does not show significant results as the number of epochs increases. For the accuracy results obtained on the testing data, which is 0.89 or 89%, the accuracy results obtained are quite optimal, which can be seen in Table 3.

3.2.2. CNN MobileNetV2

The same was done with the MobileNetV2 architecture, which was pre-trained as the base model. All its layers are set as untrainable, so the weights remain the same during training. Next, the GlobalAveragePooling2D layer is added. To lessen overfitting, there are 256 neurons in each of the two Dense layers that have ReLU activation and one Dropout layer with a ratio of 0.3. Ultimately, the model is classified into three groups using a thick output layer that has three softmax triggered neurons. as displayed in Table 4 and Figure 7.

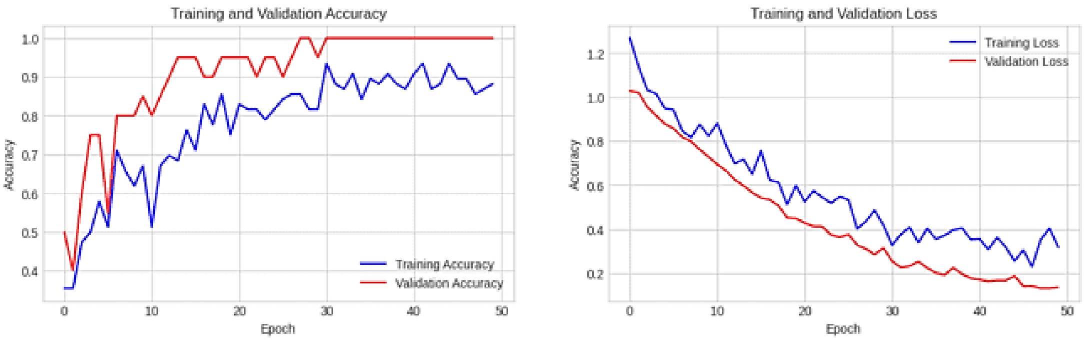

Figure 7. Training and Loss Graph of CNN MobileNetV2

Table 4. Accuracy Result of CNN MobileNetV2

	Training	Validation	Testing
Loss	0,21	0,14	0,47
Accuracy	0,96	1,00	0,78

The training carried out on this model also applies 50 epochs, and stops at the 50th epoch for the experiment graph can be seen in Figure 7. The training accuracy value in the figure is rather good because it is increasing; however, as the number of epochs increases, the graph for the loss value does not show any meaningful results. Table 4 shows the very good accuracy results acquired for conducting modeling trials, which are 0,78 or 78% for the accuracy results obtained on the testing data.

3.3. RNN Model

In the RNN (Recurrent Neural Network) model using TensorFlow. This model uses a SimpleRNN layer with 100 units, with an input shape of (28, 28). Next, a Dense layer with 3 neurons and softmax activation was added for multi-class classification. Finally, a dropout with a rate of 0.2 was performed to reduce overfitting. In addition, the same thing was done in this modeling, namely conducting experiments using DenseNet201 and MobileNetV2. Both models use GlobalAveragePooling2D layer, Dense (512, ReLU), Dropout (20%), and Dense (3, softmax) output for multi-class classification. The results can be seen in Figure 8.

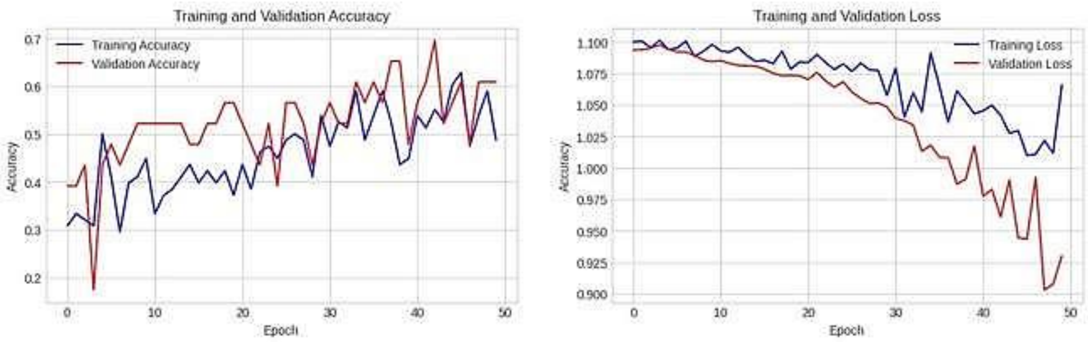

Figure 8. Training and Loss Graph of RNN Model

Table 5. Accuracy Result of RNN Model

	Training	Validation	Testing
Loss	0,69	0,49	1,22
Accuracy	0,80	0,78	0,56

The training performed on this model also applies 50 epochs but stops at the 32nd epoch for the experiment graph can be seen in Figure 8. in the figure shows a fairly significant training accuracy value even though in some epochs it has decreased, but for the loss value the graph shows quite significant results as the number of epochs increases. For the accuracy results obtained on the testing data is 0.56 or 56% which is quite good accuracy results can be seen in Table 5.

3.3.1. RNN DenseNet201

It is set as non-trainable in this architecture, so the weights are not updated during training. Next, it adds a GlobalAveragePooling2D layer, two Dense layers with 256 neurons and ReLU activation, and a Dropout layer with a ratio of 0.3 to reduce overfitting. Finally, for multi-class classification, the model has a Dense output layer with 3 neurons and softmax activation. Table 6 and Figure 9 show the results of the 50 epoch experiment.

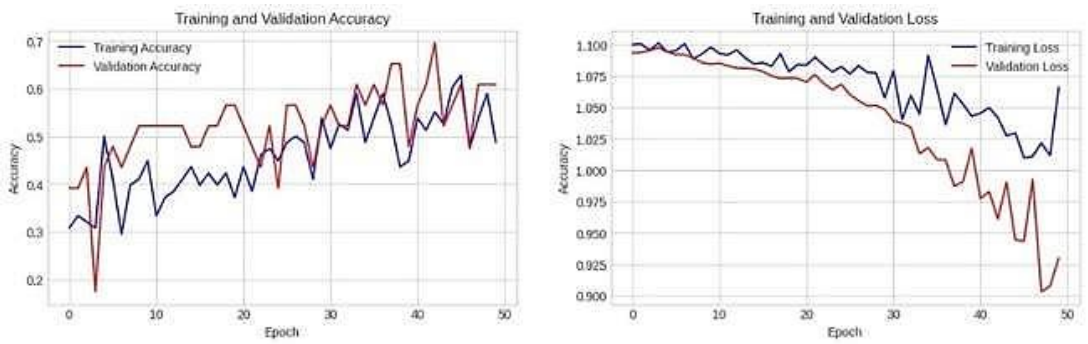

Figure 9. Training and Loss Graph of RNN DenseNet201

Table 6. Accuracy Result of RNN DenseNet201

	Training	Validation	Testing
Loss	0,98	0,93	0,94
Accuracy	0,60	0,61	0,56

The training carried out on this model also applies 50 epochs and stops at the 50th epoch for the experiment graph can be seen in Figure 9. in the figure shows a fairly significant training accuracy value even though in some epochs it has decreased, but for the loss value the graph shows quite significant results because it increases with the increase in the number of epochs. For the accuracy results obtained on the testing data which is 0.56 or 56% with the accuracy results obtained are very good for conducting modeling trials can be seen in Table 6.

3.3.2. RNN MobileNetV2

In this architecture, the same was done with the MobileNetV2 architecture, which was pre-trained as the base model. All its layers are set as untrainable, so the weights remain the same during training. Next, the GlobalAveragePooling2D layer is added. The two Dense layers have 256 neurons with ReLU activation and one Dropout layer with a ratio of 0.3 to reduce overfitting. Finally, there is a dense output layer with three softmax activated neurons to classify the model into three classes. as shown in Figure 10 and Table 7.

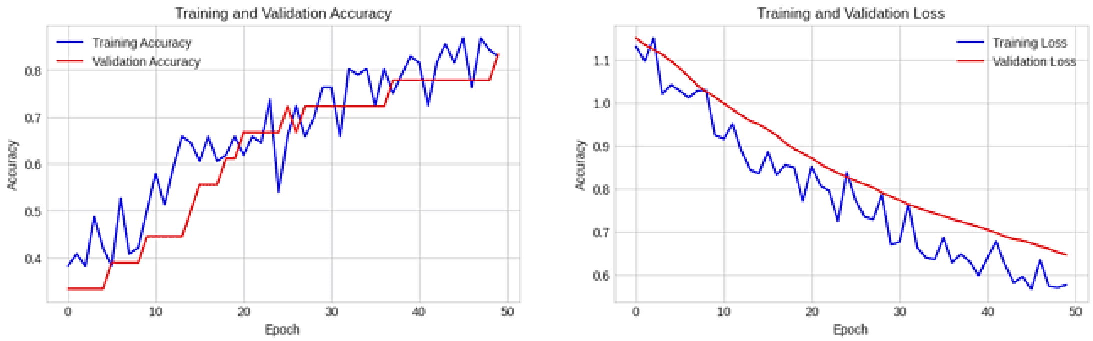

Figure 10. Training and Loss Graph of RNN MobileNetV2

Table 7. Accuracy Result of RNN MobileNetV2

	Training	Validation	Testing
Loss	0,59	0,65	0,84
Accuracy	0,87	0,83	0,67

The training performed on this model also applies 50 epochs and stops at the 50th epoch for the experiment graph can be seen in Figure 10. in the figure shows a fairly significant training accuracy value even though in some epochs it has decreased, but for the loss value the graph shows quite significant results as the number of epochs increases. For the accuracy results obtained on the testing data which is 0.67 or 67% with the accuracy results obtained are very good for conducting modeling trials can be seen in Table 7.

3.4. GANs Model

This Generative Adversarial Networks (GANs) model uses TensorFlow. The model consists of multiple convolution and max pooling layers to reduce the dimensionality of the image. First, a Conv2D layer with 32 filters (3, 3) and ReLU activation, followed by MaxPooling with a factor of (2, 2). This process is repeated with 64 filters and then 128 filters, each time followed by MaxPooling. The result is flattened into a one-dimensional vector, connected with a 512 neuron ReLU Dense layer, and then output with a 3 neuron softmax Dense layer. In addition, experiments were conducted using DenseNet201 and MobileNetV2. Both models use GlobalAveragePooling2D, Dense (512, ReLU), Dropout (20%), and Dense (3, softmax) layers for multi-class classification, see the results in Figure 11.

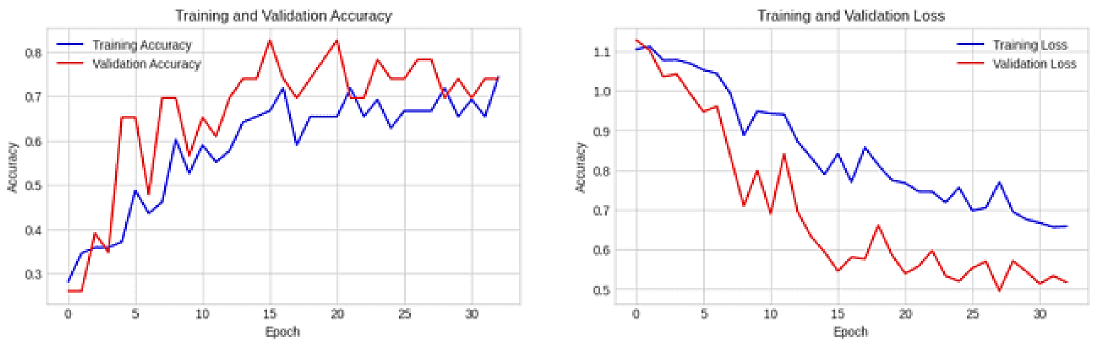

Figure 11. Training and Loss Graph of GANs Model

Table 8. Accuracy Result of GANs Model

	Training	Validation	Testing
Loss	0,68	0,49	0,58
Accuracy	0,73	0,78	0,89

The training carried out on this model also applies 50 epochs and also stops at the 33rd epoch for the experiment graph can be seen in Figure 8. The figure shows the training accuracy value which has increased quite significantly even though in some epochs it has decreased, for the loss value the graph shows quite good results as the number of epochs increases. For the accuracy results obtained on the testing data, 0,89 or 89%, which is a very good accuracy result so that it can be continued for model testing can be seen in Table 8.

3.4.1. GANs DenseNet201

DenseNet201 has been trained on the ImageNet dataset and used without the top classification layer (include_top=False). The input image has a size of 150x150 pixels. followed by two Dense layers (each with 256 neurons and ReLU activation), and Dropout with a ratio of 0.3 to reduce overfitting. The last output layer is Dense with 3 neurons and softmax activation to classify the image into three classes. Figure 12 and Table 9 show the results of the 50 epoch experiment.

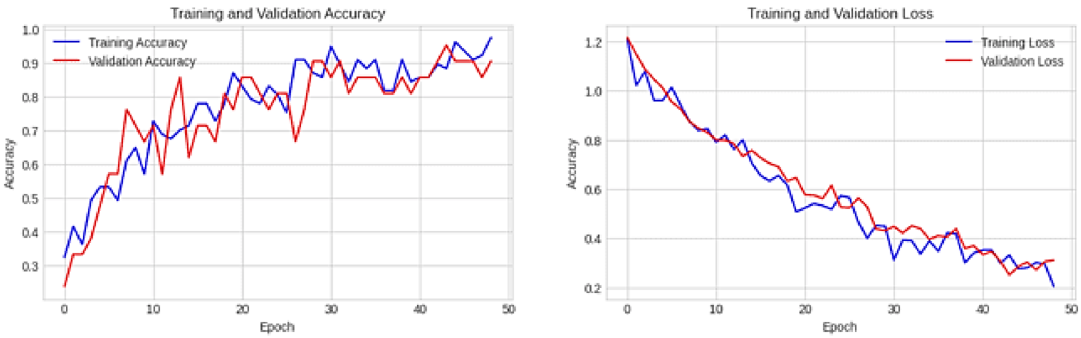

Figure 12. Training and Loss Graph of GANs DenseNet201

Table 9. Accuracy Result of GANs DenseNet201

	Training	Validation	Testing
Loss	0,22	0,25	0,46
Accuracy	0,94	0,95	0,88

The training performed on this model also applies 50 epochs and stops at the 49th epoch for the experiment graph can be seen in Figure 12. The figure shows the training accuracy value has increased significantly, and for the loss value the graph shows quite good results as the number of epochs increases. For the accuracy results obtained on the testing data which is 0.89 or 89% with the accuracy results obtained are very good for conducting modeling trials can be seen in Table 9.

3.4.2. GANs MobileNetV2

In this implementation, the MobileNetV2 model was trained on the ImageNet dataset and used without the top classification layer (include_top=False), with an input image of 150x150 pixels. All layers of MobileNetV2 were set as non-trainable to preserve the trained weights. The classification model uses GlobalAveragePooling2D, followed by two Dense layers with 256 neurons and ReLU activation, and Dropout 0.3 to reduce overfitting. The last output layer is Dense with 3 neurons and softmax activation to classify the image into three classes. Can be seen in Figure 13 and Table 10 for the 50 epoch experiment.

Table 10. Accuracy Result of GANs MobileNetV2

	Training	Validation	Testing
Loss	0,28	0,26	0,37
Accuracy	0,91	1,00	0,89

The training carried out on this model also applies 50 epochs and stops at the 50th epoch for the experiment graph can be seen in Figure 13. in the figure shows a fairly significant training accuracy value even though in some epochs it has decreased, then for the loss value the graph shows pretty good results as the number of epochs increases. For the accuracy results obtained on the testing data which is 0.89 or 89% with the accuracy results obtained are very good for conducting modeling trials can be seen in Table 10.

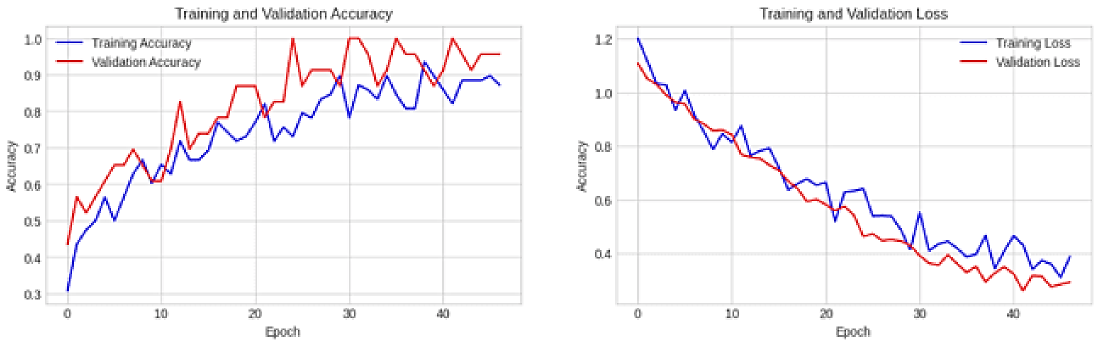

Figure 13. Training and Loss Graph of GANs MobileNetV2

3.5. Evaluation

From the results of the model that has been trained using the three models and architectures, it can be evaluated using the Classification Report and Confusion Matrix presented in the table 11-13.

Table 11. Modeling Performance Comparison

Model	Class Name	Precision	Recall	F1-Score	Support
CNN	Acne	1.000	0.667	0.800	3
	Bags	1.000	0.667	0.800	3
	Redness	0.600	1.000	0.750	3
DenseNet201	Acne	0.750	1.000	0.857	3
	Bags	1.000	1.000	1.000	3
	Redness	1.000	0.667	0.800	3
MobileNetV2	Acne	0.750	1.000	0.857	3
	Bags	1.000	0.333	0.500	3
	Redness	0.750	1.000	0.857	3

In the evaluation of the confusion matrix performance in the Precision section, the highest accuracy was obtained in the DenseNet201 modeling, namely for Acne at 75%, for Bags at 100%, and for Redness at 100%. In the evaluation of the confusion matrix performance in the Recall section, the highest accuracy was obtained in the DenseNet201 modeling, namely for Acne at 100%, for Bags at 100%, and for Redness at 66%. and in the evaluation of the confusion matrix performance in the F1-Score section, the highest accuracy was obtained in the DenseNet201 modeling, namely for Acne at 85.7%, for Bags at 100%, and for Redness at 80%.

Table 12. Modeling Performance Comparison

Model	Class Name	Precision	Recall	F1-Score	Support
CNN	Acne	0.500	0.333	0.400	3
	Bags	0.500	0.667	0.571	3
	Redness	0.667	0.667	0.667	3
DenseNet201	Acne	0.500	0.333	0.400	3
	Bags	0.500	0.667	0.571	3
	Redness	0.667	0.667	0.667	3
MobileNetV2	Acne	0.500	0.667	0.571	3
	Bags	0.667	0.667	0.667	3
	Redness	1.000	0.667	0.800	3

In the evaluation of the confusion matrix performance in the Precision section, the highest accuracy was obtained in the MobileNetv2 modeling, namely for Acne at 50%, for Bags at 66,7%, and for Redness at 100%. In the evaluation of the confusion matrix performance in the Recall section, the highest accuracy was obtained in the MobileNetV2 modeling, namely for Acne at 66,7%, for Bags at 66,7%, and for Redness at 66,7%. and in the evaluation of the confusion matrix performance in the F1-Score section, the highest accuracy was obtained in the MobileNetV2 modeling, namely for Acne at 57,1%, for Bags at 66,7%, and for Redness at 80%.

Table 13. Modeling Performance Comparison

Model	Class Name	Precision	Recall	F1-Score	Support
GANs	Acne	1.000	0.667	0.800	3
	Bags	1.000	1.000	1.000	3
	Redness	0.750	1.000	0.857	3
DenseNet201	Acne	0.750	1.000	0.857	3
	Bags	1.000	0.667	0.800	3
	Redness	1.000	1.000	1.000	3
MobileNetV2	Acne	0.750	1.000	0.857	3
	Bags	1.000	1.000	1.000	3
	Redness	1.000	0.667	0.800	3

In the evaluation of the confusion matrix performance in the Precision section, the modeling accuracy was obtained equally, namely around 75% to 100%. In the evaluation of the confusion matrix performance in the Recall section, the modeling accuracy was obtained equally, namely around 66,7% to 100%. and in the evaluation of the confusion matrix performance in the F1-Score section, the modeling accuracy was obtained equally, namely around 80% to 100%.

After testing the model using CNN, RNN, and GANs and applying 2 architectures, namely DenseNet201 and MobileNetV2, a comparison of the accuracy results of the plementation can be seen in Figure 14 on curacy comparison.

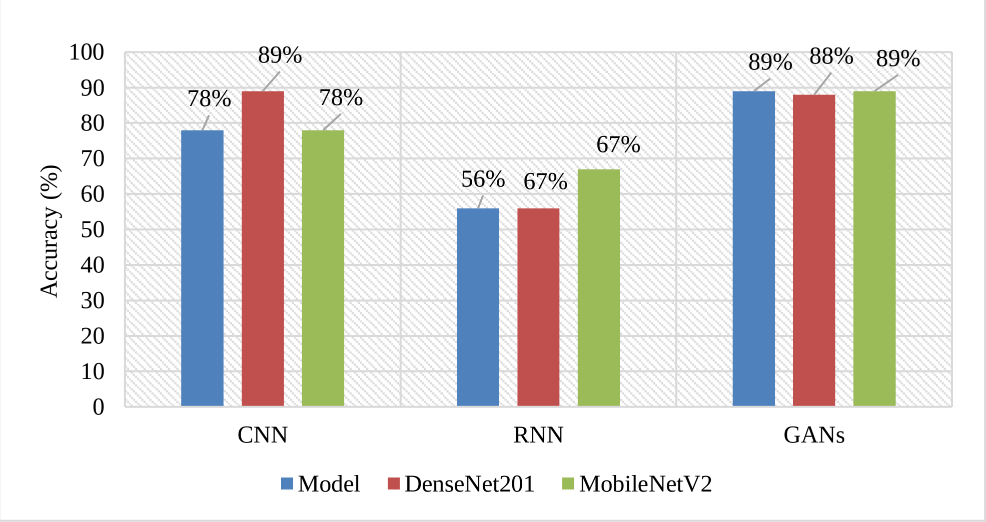

Figure 14. Comparison of Accuracy Results

Based on Figure 23, it can be concluded that the most optimal modeling in the study for classifying damage to facial skin falls on the GANs algorithm along with the DenseNet201 and MobileNetV2 architectures with a testing accuracy value of 89% on the GANs model, 88% on the DenseNet201 architecture, and 89% on the MobileNetV2 architecture.

4. CONCLUSION

This research aims to classify damage to facial skin by utilizing three algorithms, namely Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Networks (GANs), as well as two architectures, named DenseNet201 and MobileNetV2. so that optimal accuracy results are obtained. The training process uses the same data sharing technique, using CNN, RNN, GANs. The dataset used is segmented into 3 classes, including acne, eye bags, and redness. The results showed that GANs with DenseNet201 and MobileNetV2 architectures had the best testing accuracy, namely 89%, 88% and 89%. These results are expected to show that the use of Deep Learning is an effective solution for the classification and identification of facial skin problems, making an important to contribute in the field of dermatology and skin health.

REFERENCES

[1] J. L. López-Estebaranz, P. Herranz-Pinto, and B. Dréno, “Consensus-Based Acne Classification System and Treatment Algorithm for Spain,” Actas Dermosifiliogr., vol. 108, no. 2, pp. 120–131, 2017, doi: 10.1016/j.adengl.2016.10.003.
[2] Z. Zhu et al., “Global, regional, and national burdens of acne vulgaris in adolescents and young adults

aged 10-24 years from 1990 to 2021: a trend analysis.," Br. J. Dermatol., Sep. 2024, doi: 10.1093/bjd/ljae352.
[3] I. Muhimmah, N. F. Muchlis, and A. Kurniawardhani, "Automatic Facial Redness Detection on Face Skin Image," IIUM Eng. J., vol. 22, no. 1, pp. 68–77, 2021, doi: 10.31436/IIUMEJ.V22I1.1495.
[4] S. Saiwaeo, S. Arwatchananukul, L. Mungmai, W. Preedalikit, and N. Aunsri, "Human skin type classification using image processing and deep learning approaches," Heliyon, vol. 9, no. 11, p. e21176, 2023, doi: 10.1016/j.heliyon.2023.e21176.
[5] G. A. Mesías-Ruiz, I. Borra-Serrano, J. M. Peña, A. I. de Castro, C. Fernández-Quintanilla, and J. Dorado, "Weed species classification with UAV imagery and standard CNN models: Assessing the frontiers of training and inference phases," Crop Prot., vol. 182, no. May, 2024, doi: 10.1016/j.cropro.2024.106721.
[6] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, "Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs," Procedia Comput. Sci., vol. 135, pp. 425–432, 2018, doi: 10.1016/j.procs.2018.08.193.
[7] A. Kumar, N. Gaur, S. Chakravarty, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, "Analysis of spectrum sensing using deep learning algorithms: CNNs and RNNs," Ain Shams Eng. J., vol. 15, no. 3, p. 102505, 2024, doi: 10.1016/j.asej.2023.102505.
[8] D. Hastari, S. Winanda, A. R. Pratama, N. Nurhaliza, and E. S. Ginting, "Application of Convolutional Neural Network ResNet-50 V2 on Image Classification of Rice Plant Disease," Public Res. J. Eng. Data Technol. Comput. Sci., vol. 1, no. 2, pp. 71–77, 2024, doi: 10.57152/predatecs.v1i2.865.
[9] K. Murali and K. Murali, "ScienceDirect Conference on Enhancing Fake Image Detection : A Novel Two-Step Approach Combining CNNs Enhancing Fake Image Detection : A Novel Two-Step Approach Combining GANs and CNNs," Procedia Comput. Sci., vol. 235, pp. 810–819, 2024, doi: 10.1016/j.procs.2024.04.077.
[10] M. N. Musa and N. O. Badmos, "Application of MobileNetV2 Transfer Learning for Face-shield Detection Application of MobileNetV2 Transfer Learning for Face-shield Detection," no. July, 2023.
[11] W. Madelina and Sulistiyaningsih, "Review: Resistensi Antibiotik pada Terapi Pengobatan Jerawat," J. Farmaka, vol. 16, no. 2, pp. 105–117, 2018.
[12] R. T. Lestari et al., "Perilaku Mahasiswa Terkait Cara Mengatasi Jerawat," J. Farm. Komunitas, vol. 8, no. 1 SE-Articles, pp. 15–19, Oct. 2020, doi: 10.20473/jfk.v8i1.21922.
[13] L. Deng and D. Yu, "Deep Learning: Methods and Applications," Found. Trends Signal Process., vol. 7, Jan. 2013, doi: 10.1561/2000000039.
[14] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," Biosyst. Eng., vol. 151, pp. 72–80, 2016, doi: https://doi.org/10.1016/j.biosystemseng.2016.08.024.
[15] M. Hussain, J. J. Bird, and D. R. Faria, "A study on CNN transfer learning for image classification," Adv. Intell. Syst. Comput., vol. 840, no. June, pp. 191–202, 2019, doi: 10.1007/978-3-319-97982-3_16.
[16] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," Agric., vol. 11, no. 8, pp. 1–18, 2021, doi: 10.3390/agriculture11080707.
[17] J. Wu, "Introduction to Convolutional Neural Networks," Introd. to Convolutional Neural Networks, pp. 1–31, 2017, [Online]. Available: https://web.archive.org/web/20180928011532/https://cs.nju.edu.cn/wujx/teaching/15_CNN.pdf.
[18] P. D. Rinanda, D. N. Aini, T. A. Pertiwi, S. Suryani, and A. J. Prakash, "Implementation of Convolutional Neural Network (CNN) for Image Classification of Leaf Disease In Mango Plants Using Deep Learning Approach," Public Res. J. Eng. Data Technol. Comput. Sci., vol. 1, no. 2, pp. 56–61, 2024, doi: 10.57152/predatecs.v1i2.872.
[19] M. A. Hossain and M. S. Alam Sajib, "Classification of Image using Convolutional Neural Network (CNN)," Glob. J. Comput. Sci. Technol., vol. 19, no. 2, pp. 13–18, 2019, doi: 10.34257/gjcstdvol19is2pg13.
[20] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A Unified Framework for Multi-label Image Classification," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2285–2294, doi: 10.1109/CVPR.2016.251.
[21] D. Lee et al., "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," China Commun., vol. 14, no. 9, pp. 23–31, 2017, doi: 10.1109/CC.2017.8068761.
[22] Y. Kristian, I. K. E. Purnama, E. H. Sutanto, L. Zaman, E. I. Setiawan, and M. H. Purnomo, "Klasifikasi Nyeri pada Video Ekspresi Wajah Bayi Menggunakan DCNN Autoencoder dan LSTM," J. Nas. Tek. Elektro dan Teknol. Inf., vol. 7, no. 3, pp. 308–316, 2018, doi: 10.22146/jnteti.v7i3.440.
[23] J. Arifianto and I. Muhimmah, "Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow," J. Autom., pp. 21–29, 2021.
[24] A. Satriawan, B. Imran, and S. Erniwati, "Identifikasi Kemiripan Foto Asli Dan Sketsa Menggunakan

Model Generative Adversarial Networks (GANs)," J. Kecerdasan Buatan dan Teknol. Inf., vol. 2, no. 3, pp. 122–127, 2023.

[25] J. J. Jeong, A. Tariq, T. Adejumo, H. Trivedi, J. W. Gichoya, and I. Banerjee, "Systematic Review of Generative Adversarial Networks (GANs) for Medical Image Classification and Segmentation.," J. Digit. Imaging, vol. 35, no. 2, pp. 137–152, Apr. 2022, doi: 10.1007/s10278-021-00556-w.

[26] P. Sitompul, H. Okprana, A. Prasetio, and G. Artikel, "Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201 Identification of Rice Plant Diseases Through Leaf Image Using DenseNet 201," JOMLAI J. Mach. Learn. Artif. Intell., vol. 1, no. 2, pp. 143–150, 2022, doi: 10.55123/jomlai.v1i2.889.

[27] J. Pardede and D. A. L. Putra, "Implementasi DenseNet Untuk Mengidentifikasi Kanker Kulit Melanoma," J. Tek. Inform. dan Sist. Inf., vol. 6, no. 3, 2020, doi: 10.28932/jutisi.v6i3.2814.

[28] A. Dash, P. K. Sethy, and S. K. Behera, "Maize disease identification based on optimized support vector machine using deep feature of DenseNet201," J. Agric. Food Res., vol. 14, no. October, p. 100824, 2023, doi: 10.1016/j.jafr.2023.100824.

[29] H. K. Chhatrapati et al., "Transfer Learning and Supervised Machine Learning Approach for Detection of Skin Cancer: Performance Analysis and Comparison," vol. 10, no. September, p. 1, 2021, [Online]. Available: https://www.researchgate.net/publication/354450739.

[30] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," Procedia Comput. Sci., vol. 197, pp. 198–207, 2021, doi: 10.1016/j.procs.2021.12.132.

[31] J. S. Talahua, J. Buele, P. Calvopina, and J. Varela-Aldas, "Facial recognition system for people with and without face mask in times of the covid-19 pandemic," Sustain., vol. 13, no. 12, pp. 1–19, 2021, doi: 10.3390/su13126900.

[32] P. D. Rinanda and Mustakim, "Implementation of PNN, ANN And K-NN Algorithms on Indonesian Marketplace Reviews on Google Play Store," 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Manama, Bahrain, 2024, pp. 1070-1074, doi: 10.1109/ICETSIS61505.2024.10459477.