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: 81-92
ISSN(P): 3024-921X | ISSN(E): 3024-8043

Comparative Analysis of Weather Image Classification Using CNN Algorithm with InceptionV3, DenseNet169 and NASNetMobile Architecture Models

Vina Wulandari[1*], Windy Junita Sari[2],
Zaid Husham Al-Sawaff[3] , Selvakumar Manickam[4]

[12]Department of Information System, Faculty of Science and Technology,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
[3]Department of Medical Instrumentation Technology,
Center of Technical Research Northern Technical University Mosul, Iraq
[4]National Advanced IPv6 Centre (NAv6),
University Sains Malaysia, Penang, Malaysia

E-Mail: [1]12150321941@students.uin-suska.ac.id,
[2]windyjunitasari843@gmail.com, [3]zaidalsawaff@ntu.edu.iq,[4]selva@usm.my

Received Aug 8th 2024; Revised Nov 25th 2024; Accepted Dec 29th 2024; Available Online Jan 12th 2025, Published Jan 18th 2025
Corresponding Author: Vina Wulandari

Abstract

Rapid weather changes have a significant impact on various aspects of human life, including social and economic development. Weather analysis traditionally relies on data from Doppler radar, weather satellites, and weather balloons. However, advancements in computer vision technology provide new opportunities to enhance weather prediction systems through image recognition and classification. Studies evaluating and comparing deep learning architectures for weather image classification remain limited.This research utilizes Convolutional Neural Networks (CNN) to classify weather images using three architectures: InceptionV3, DenseNet169, and NASNetMobile. The results show that InceptionV3 achieved 97.94% accuracy on training data, 92.34% on validation data, and 93.81% on test data. DenseNet169 achieved 98.09% accuracy on training data, 88.46% on validation data, and 92.33% on test data. NASNetMobile achieved 96.51% accuracy on training data, 87.82% on validation data, and 89.97% on test data. Based on these results, InceptionV3 is the optimal choice for weather classification due to its consistent performance.This research addresses the gap in evaluating CNN architectures for weather data and contributes to improving weather monitoring systems, early disaster warnings, and applications reliant on accurate predictions. These findings also provide a foundation for the development of advanced technologies in image analysis and weather forecasting in the future.

Keyword: Convolutional Neural Networks, DenseNet169, Inception V3, NASNetMobile, Weather

1. INTRODUCTION

Rapid weather change has a big impact on people's lives and the advancement of society. Understanding weather patterns accurately is essential to reducing their consequences. Data from Doppler radar, weather satellites, and weather balloons, as well as conventional techniques like temperature and air measurements, are crucial for weather analysis [1]. However, new prospects for improving weather prediction systems arise with the growth of computer vision, which focuses on using computers for image recognition and categorization. Specifically, Weather Image Classification makes use of satellite imagery to improve weather forecasting. The integration of computer vision techniques to automate real-time weather forecasting without requiring internet connectivity makes this research more pertinent. The research contribution is its potential to increase weather prediction models precision and effectiveness, especially when using sophisticated modeling techniques that make use of satellite imagery data. The development of reliable models that can manage massive amounts of weather data is the main goal of this work, as the technical difficulty of putting these techniques into practice [2].

To generate weather images, techniques capable of handling diverse and complex image data are required. This research is important because it can improve the accuracy of weather prediction, which has a great impact on various industries such as agriculture, transportation, and disaster management. It is proven that deep learning is excellent for image recognition [3]. Convolutional Neural Networks (CNN) are becoming standard in image analysis due to their ability to recognize patterns and important elements [4]. CNNs consist of various layers of artificial neurons that allow them to extract features from images by using great computational power to detect small patterns that may not be visible to the human eye [5].

This research introduces a comprehensive classification model in predicting weather in various climate zones and classifying weather into four categories namely cloudy, rain, shine, sunrise using CNN algorithm with Keras framework and TensorFlow library. The final result of this research displays the performance of the model that has been designed and developed. This model shows accuracy, validation accuracy, losses, and approximately 94%, 92%, 18%, and 22% [1]. Research conducted by Kukreja et al. in 2023 proposed a CNN and SVM combined model for weather condition detection and multi-classification using 10,000 images with five weather conditions. The model achieved an overall accuracy of 97.24%, showing its superiority compared to other weather detection models [6], Research conducted by mittal and sangwan in 2023 proposed a new way to classify weather conditions from outdoor images using faster and more efficient machine learning techniques. By utilizing InceptionV3's pre-trained CNN model and Logistic Regression classifier, the experimental results show an accuracy of 97.77% [7].

Furthermore, research in 2024 by Rinanda and his colleagues on the classification of leaf disease images of mango plants. They compared the accuracy of CNN, VGG16, and InceptionV3 models. The results showed that VGG16 was the most optimal, with an accuracy of 96.87% in all three modeling test scenarios. InceptionV3 took second place with an accuracy of 96.50%, while CNN obtained an accuracy of 81% [8]. Then the research conducted by Pratama et al. to compare VGG16 architecture with DenseNet169 in tumor image classification. The results show that DenseNet169 has better performance compared to VGG16. For the accuracy metric, DenseNet169 reached a value of 98%, while VGG16 only reached 75%. The other performance metrics, recall and f1-score, also show similar results, with DenseNet169 outperforming VGG16. In the precision metric, DenseNet169 has a much higher result, which is 97%, compared to VGG16 which only reaches 76% [9].

Fuadi and Suharso's subsequent study from 2022 contrasted the NASNetMobile and MobileNet architectures for the classification of diseases in photos of potato leaves. On mobile devices, picture categorization issues are resolved using both architectures. Several training and test data separation strategies, including 90:10, 80:20, 70:30, 60:40, and 50:50, are used in this study. The photos of potato leaves that are in good health, those that are diseased with early blight, and those that are infected with late blight comprise the data that was employed. Using NASNet Mobile architecture and a 90:10 training to test data ratio, the testing scheme produced results at the end of the study that included 90.96% accuracy, 90.86% precision, 91.11% recall, and 92.93% f1 score [10].

From the research results that have been presented previously, the InceptionV3, DenseNet169, and NASNetMobile architecture models have superior accuracy compared to other architectures in classifying image data, therefore this study will use the three architectures to be applied to the data we have provided, namely weather image data. Each of these architectures has its own advantages that can provide high accuracy. This research will test whether the three architectures will also provide high accuracy results on the weather image data we have.

Based on the results of testing the InceptionV3, DenseNet169, and NASNetMobile architecture models on weather image data, the accuracy results obtained will be compared. Where in previous studies no comparison has been made between the three architectural models, the results of the comparison of the three architectural models can be used as recommendations for other researchers in choosing architectural models that have high accuracy for classifying weather image data. With increased accuracy in recognizing weather image data, this research is expected to improve weather monitoring systems, early warning of natural disasters, and other applications that rely on accurate weather information. In addition, the findings of this research can be the basis for the development of advanced technologies in image analysis and weather prediction in the future.

2. MATERIAL AND METHOD

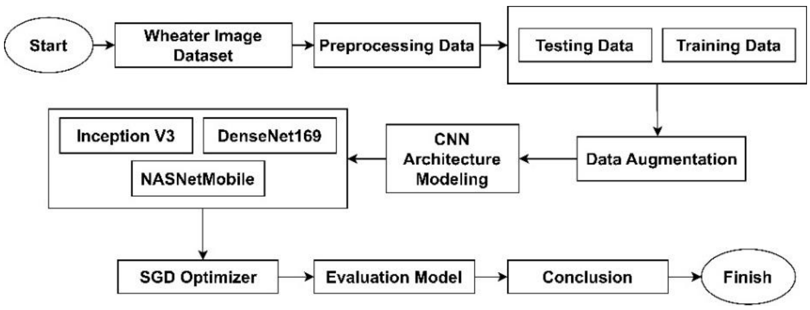

Figure 1. Research Metdology

The following is an explanation of the research methodology carried out in this study:

2.1. Weather Image Dataset

This dataset contains 1125 labeled photos, each representing a specific weather type. The photos are categorized into Four classes: Cloudy (300 images), Rain (215 images), Shine (253 images) and Sunrise (357 images). These images are sourced from a Kaggle dataset, accessible at https://www.kaggle.com/datasets/kotabarufighter/data-cuaca. The dataset provides diverse representations of weather phenomena, making it highly valuable for training weather classification models.However, it is important to note that although the dataset covers a wide range of meteorological conditions, some classes are not equally represented. This imbalance could impact the model's ability to generalize across all weather types. To address this, several preprocessing steps are applied to ensure data quality. These steps include cleaning (removing incorrect or irrelevant images), normalization (scaling pixel values to a standard range), and augmentation (applying transformations such as rotation, flipping, and zooming to increase the dataset's size and diversity). These measures enhance the dataset's utility, making it more suitable for effective machine learning applications [11].

2.2. Preprocessing Data

After obtaining the data, the next step is preprocessing [12]. Data preprocessing is an important step in deep learning modeling and serves as the cornerstone of reliable data analysis [13]. These deep learning models require a lot of training data, and small datasets often lead to overfitting and poor performance on large datasets [14]. Therefore, to improve accuracy, various data preprocessing approaches are used [15][16].

In this stage, the data is divided into two parts [17], namely 70% training data and 30% testing data. 70% of the test data is used to train the model, so that the model can learn the patterns and features contained in the data, while 30% of the test data is used to evaluate the performance of the model on data that has never been seen before to ensure that the model can generalize well and does not experience overfitting.

2.3. Data Augmentation

Data augmentation is an approach that allows practitioners to dramatically expand the diversity of data available for training models without having to collect additional data. In machine learning, data augmentation is an important technique to improve model performance and generalization capabilities [18]. Data augmentation methods are commonly used in deep learning to increase the amount of data required to train an accurate model [19].

Cropping, padding, and horizontal flipping are popular data augmentation strategies for training large neural networks. Although these techniques are simple, they are very effective in improving model performance as the architecture of neural networks has been widely researched [20]. The augmentation used in this study aims to enrich the variation of weather images, making the model more responsive to different conditions. Rotation (45 degrees) helps the model recognize weather from various orientations, Rescale adjusts pixel values to the range of 0 to 1 to facilitate image processing. Shear creates perspective variations by shifting parts of the image, while zoom enlarges the image up to 10% to simulate different object proximity levels, Horizontal and vertical flips allow the model to recognize weather from different viewpoints, Brightness adjusts the image's brightness to handle natural lighting variations. Finally, fill mode fills empty pixels after transformation with the nearest pixel value to maintain image quality.

2.4. Convolutional Neural Network (CNN)

One kind of deep learning algorithm used for picture recognition is the Convolutional Neural Network (CNN) algorithm. CNN is specifically designed to extract features from image data gradually through convolution and pooling processes [20][21]. CNN is the most common type of artificial neural network used to analyze visual images [22].

CNN analyzes input data, especially images, through multiple arrays. This allows it to process spatiotemporal features with increased resolution, then transform these features into more complicated ones at lower resolutions [23]. The basic formula in CNN used for the convolution process is as Equation 1.

$$\boldsymbol{a_{i,j}} = \sum_{\boldsymbol{m=0}}^{\boldsymbol{s}} \sum_{\boldsymbol{n=0}}^{\boldsymbol{s}} \boldsymbol{w_{m,n} x_{i+mj+n}} \quad (1)$$

Description: y is the output of the neuron; f is the activation function; w_i is the weight of the input to the neuron; x_i is the input to the neuron; b is the bias; and n is the number of inputs to the neuron.

2.5. InceptionV3

InceptionV3 is a deep learning architecture developed in 2015 by Google researchers Zbigniew Wojna, Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Jonathon Shlens. This modified version of the inception architecture has many improvements such as 7x7 convolution, label smoothing, and the use of additional classification to send label data to the network at the bottom. This architecture has a Module called "Inception block" which is used to extract features [8] [24]. The Inception V3 model uses an Artificial Neural Network (ANN) with fully connected layers and softmax is used for incremental classification [25].

2.6. DenseNet169

DenseNet-169 is a dense convolutional neural network that connects each layer with all other layers in subsequent blocks. With this, DenseNet-169 allows for a drastic reduction in the number of parameters and increases the flow of information in the network. The architecture consists of four dense blocks with a transition layer after each block, followed by a classification layer with softmax activation at the end. Each convolutional layer inside utilizes Batch Normalization, Rectified Linear Unit (ReLU), and convolution, with each dense block consisting of 1x1 and 3x3 convolutions. DenseNet-169 has a total of 169 layers, including 82 sets of 1x1 and 3x3 convolutional layers, as well as transition and classification layers [26].

2.7. NasNetMobile

The NASNet Mobile architecture, designed by Zoph and Le uses an innovative approach in neural network architecture design by utilizing a combination of reinforcement learning and Recurrent Neural Networks (RNN). This method enables efficient search for the optimal configuration of neural networks by reducing the computational complexity typically required to design CNN architectures, especially on large datasets such as ImageNet. NASNet Mobile allows adaptive scalability according to the size of the data used, making it flexible in various application contexts and computational constraints. Figure 3 is the architecture of NASNetMobile [27].

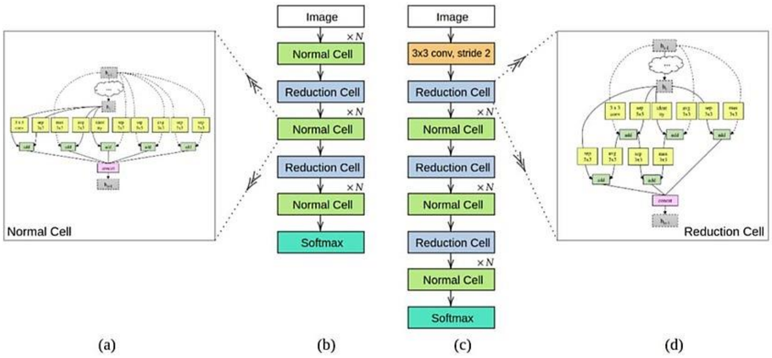

Figure 2. NASNetMobile Architecture

2.8. Stochastic Gradient Descent (SGD)

One of the best options for deep neural networks is SGD. Different types of variants are categorized based on the amount of data used to identify the gradient of the objective function [28]. SGD is an optimization technique that can find model parameters and make accurate predictions by iterating over a wide range of data samples [29]. It is essential to develop a private SGD algorithm to reduce the privacy leakage posted by the gradient as SGD is widely used in machine learning models [30].

2.9. Model Evaluation

Model evaluation is a process used to assess the performance accuracy and effectiveness of a machine learning or artificial neural network model. The model evaluation process is an important part of model development, and the metrics used for model evaluation are critical for model calibration and validation [29]. The following equation is used in calculating the accuracy value, Equation (2) [31].

$$Accuracy \frac{TN + TP}{TN + TP + FP + FN} \tag{2}$$

The purpose of this accuracy assessment is to assess how well the model performs the desired task, such as prediction or classification, and to identify areas that require improvement.

2.10. Classification

Classification is the process of finding a set of functions (models) that can explain and distinguish classes of data or concepts. The goal of classification is to use this set of models to predict the class of an unknown object [32].

3. RESULTS AND DISCUSSION

3.1. Dataset

After the weather dataset is obtained from Kaggle.com, in the Python programming process, the data is displayed randomly by displaying 5 images from each class. This dataset consists of 4 classes, namely Cloudy, Rain, Shine, and Sunrise, as seen in figure 3.

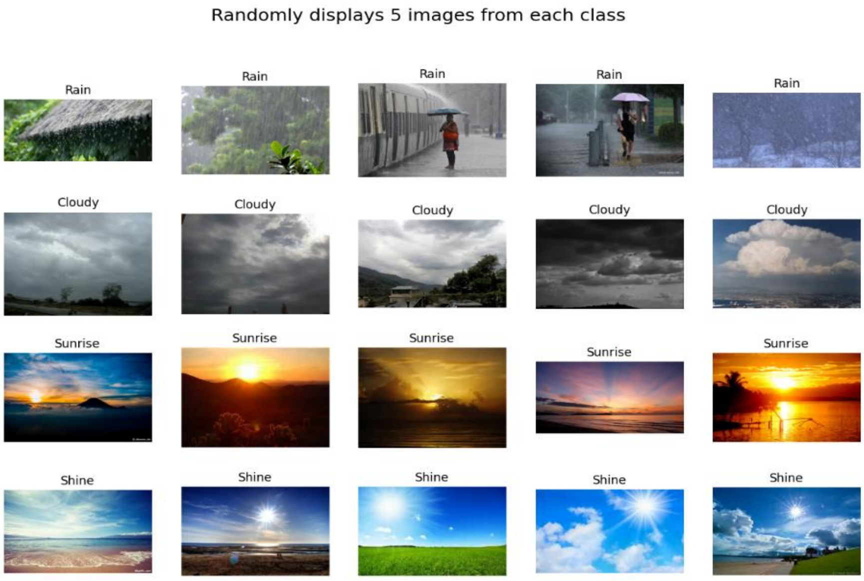

Figure 3. Weather Dataset

3.2. Pre-processing Testing

After the data collection stage is complete, data preprocessing such as augmentation of weather image data is carried out. The augmentation process performed in this study is that the image data is randomly rotated by 45 degrees to introduce variations in object orientation. In addition, the pixel values in the image are rescaled to a range of 0 to 1 by rescaling. Shear distortion with a random angle of 15% is applied to introduce realistic geometric deformation in the image. Random zoom up to 10% is applied to randomly enlarge or reduce the image. Horizontal and vertical flip was used to create additional geometric variations. Also, the brightness of the image is randomly changed between 80% to 120% of the original value. Furthermore, when there are empty pixels, they are filled with the nearest pixel value to maintain the consistency of the image structure. The following results of weather image data augmentation are shown in Figure 4 [33].

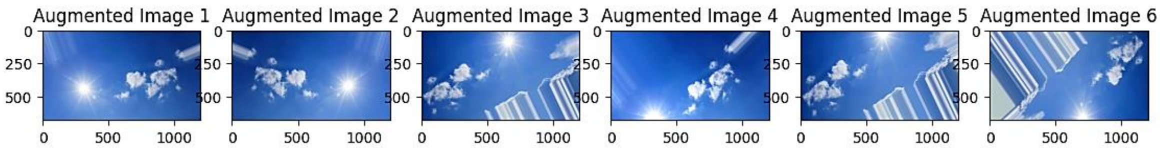

Figure 4. Augmentation Result

3.3. CNN Architecture Modeling

CNN modeling uses a model architecture consisting of several different layers, including convolution (Conv2D), pooling (MaxPooling2D), dropout, and fully connected (Dense) layers. Each layer has a different output shape according to the transformation applied to the input data. The figure 5 diagram shows each layer in the model, along with the form of output produced at each stage.

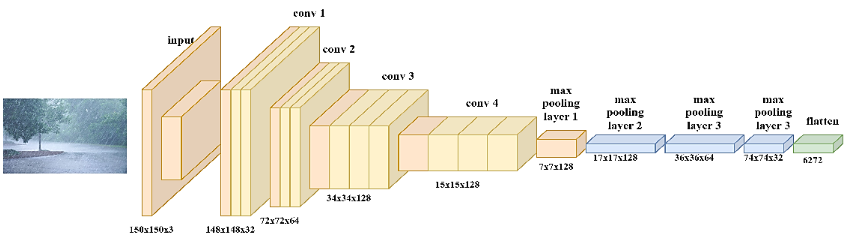

Figure 5. CNN Layer Architecture

The parameters used for model training using SGD, Early Stopping, Model Checkpoint, and ReduceLROnPlateau are shown in Table 1. SGD parameters were used for model optimization with a learning rate of 0.001 and a momentum value of 0.9. Next Early stopping is used to prevent overfitting by stopping the training model after 5 epochs without any improvement. The Model Checkpoint saves the best model based on the val_loss value which only saves the model weights and reduces the size of the model file to be saved. If the val_loss value does not increase after 5 epochs, ReduceLROnPlateau will reduce the learning rate. This is done with a reduction factor of 0.1 to help the model get out of the local minima condition.

Table 1. Hyperparameter

Parameter Name	Setting Parameter	Value Used
SGD	Learning Rate	0.01
	Momentum	0.9
Early Stopping	Patience	5
Model Checkpoint	Save_best_only	True
	Save_weights_only	True
	Monitor	Val_loss
	Mode	Min
	Verbose	1
	Filepath	Best_model.h5
ReduceLROnPlateau	Monitor	Val_loss
	factor	0.1
	Patience	5
	Verbose	1
	Mode	min
	Min_delta	0.001
	Epoch	35

As can be observed, Table 2 displays the CNN model's best preprocessing test results. The accuracy scores were 82.86% for training, 82.98% for validation, and 80.53% for testing. Furthermore, there is a 0.448 training loss, a 0.469 validation loss, and a 0.465 testing loss. When preprocessing the images before to the classification procedure, this CNN model performs admirably. A thorough understanding of the model's performance is offered by the use of accuracy and loss as evaluation metrics. Loss calculates the difference between the expected and actual values, which helps determine how well the model has learned, whereas accuracy shows the percentage of accurate predictions. These findings demonstrate the importance of preprocessing in enhancing the model's ability to categorize weather conditions by lowering the loss and increasing the quality of the weather data used.

Table 2. Effect of Preprocessing on Performance

Result	Training	Validation	Testing
Accuracy (%)	82.86 %	82.98%	80.53%
Loss	0.448	0.469	0.465

Table 2 shows the results of testing preprocessing on CNN models on accuracy and loss performance. The almost equal accuracy between training and validation shows that the model does not suffer from significant overfitting, and the small difference in testing accuracy indicates that the model can generalize well to data that was not seen before. Training and Loss CNN can view figure 6.

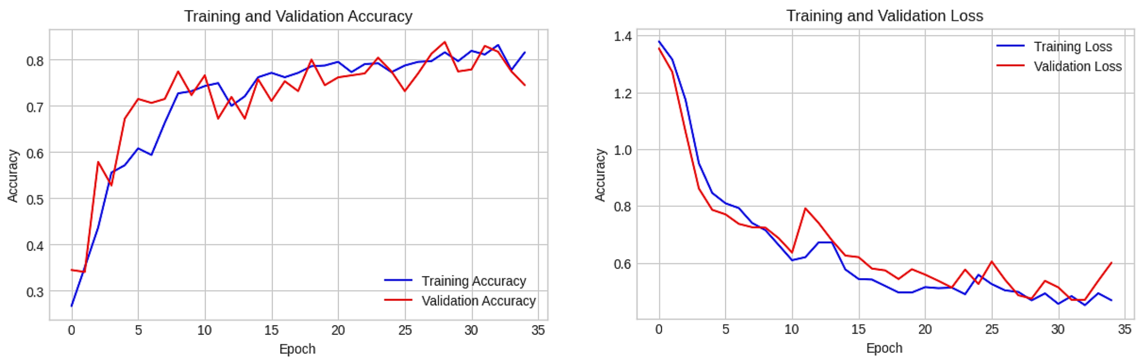

Figure 6. Training and Loss CNN

The first graph showing the training and validation accuracies shows the improvement in model accuracy over 35 epochs. At the beginning of training (epoch 0-5), there is a significant increase in the accuracy of the training and validation data, indicating that the model is learning well about the data features. In the middle of training (epochs 6-20), the accuracy continued to increase with some fluctuations. The validation accuracy is sometimes slightly higher than the training accuracy, indicating that the model is not overfitting at this stage. At the end of training (epoch 21-35), the accuracy starts to stabilize, with both lines approaching consistent high values, indicating that training and validation are doing a balanced job.

The second graph shows the training and validation loss. The loss values for training (blue line) and validation (red line) data decrease with time. At the beginning of training (epoch 0-5), there is a sharp decrease in the loss values for both training and validation data, indicating that the model learns quickly in the early stages. In the middle of the training (epochs 6-20), the loss continues to decrease but slower than in the early phase, with little change in the validation loss indicating that the model is having difficulty with some validation examples.

3.4. CNN modeling using InceptionV3 Architecture

Below are two graphs showing the accuracy and loss of the machine learning model training and validation process over 35 epochs. The first graph shows the accuracy and loss of training and validation, and the second graph shows the loss of training and validation. Training and Loss InceptionV3 can view figure 7.

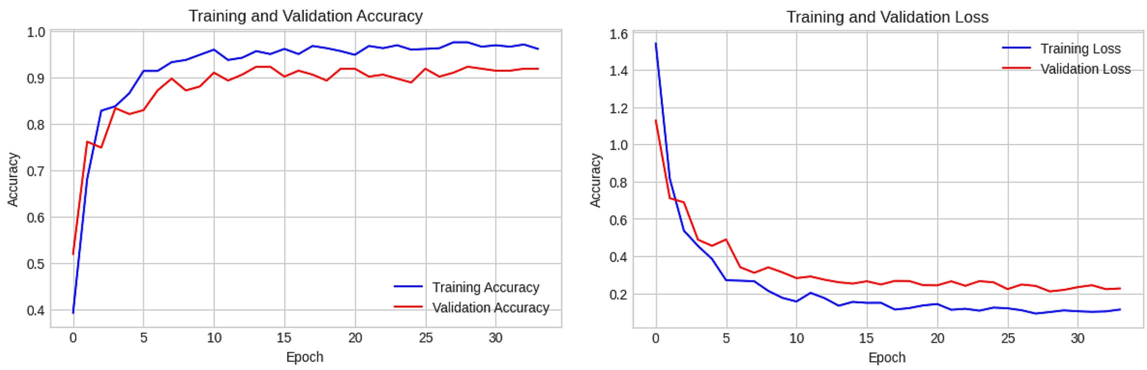

Figure 7. Training and Loss InceptionV3

Overall, these graphs show that the model trained well, with a significant increase in accuracy and a consistent decrease in loss on both training and validation data. Figure 8 shows the loss and accuracy values of the model on training, validation, and testing data.

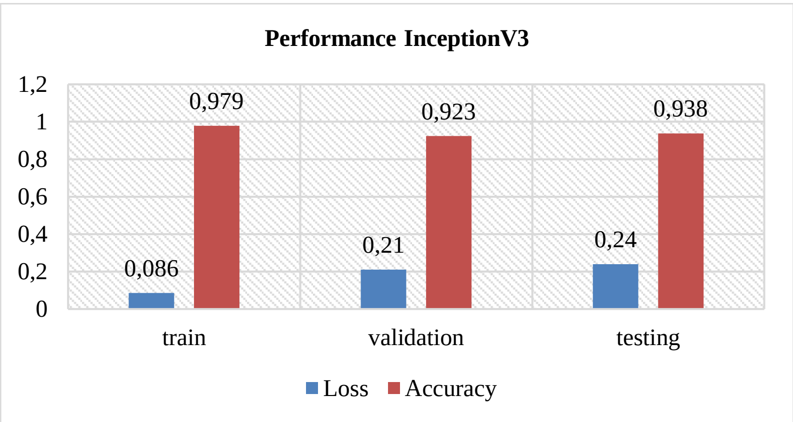

Figure 8. InceptionV3 Comparison Diagram

Performance of the model on training, validation, and testing data is displayed in the table results. With a low loss value of 0.086 and a high accuracy of 97.9% on the training data, the model demonstrated good learning. In the validation data, the accuracy decreased somewhat to 92.3% and the loss value increased to 0.21, suggesting little overfitting but still good performance. The model demonstrated strong generalization ability and reliability on fresh data, as seen by the accuracy of 93.8% and the loss value of 0.24 on the test data. With minimal loss values and good accuracy across all data sets, the model performed admirably overall. Nevertheless, additional error analysis is required to investigate how the model behaves in more difficult weather scenarios, such severe storms, fog, or uncommon weather occurrences. For example, these conditions can provide special problems for image recognition models, so it would be helpful to examine the classification mistakes in these situations to find places where the model might fail or misclassify. This in-depth investigation may shed light on the model's handling of challenging situations and point up possible areas for added development.

3.5. CNN modeling using DenseNet169 Architecture

Figure 9 show the results of Convolutional Neural Network (CNN) modeling using the DenseNet169 architecture. This architecture, a variant of DenseNet (Densely Connected Convolutional Networks), has 169 layers and allows for more efficient modeling with fewer parameters and reduces the vanishing gradient problem. Training and Loss DensNet169 can view figure 9.

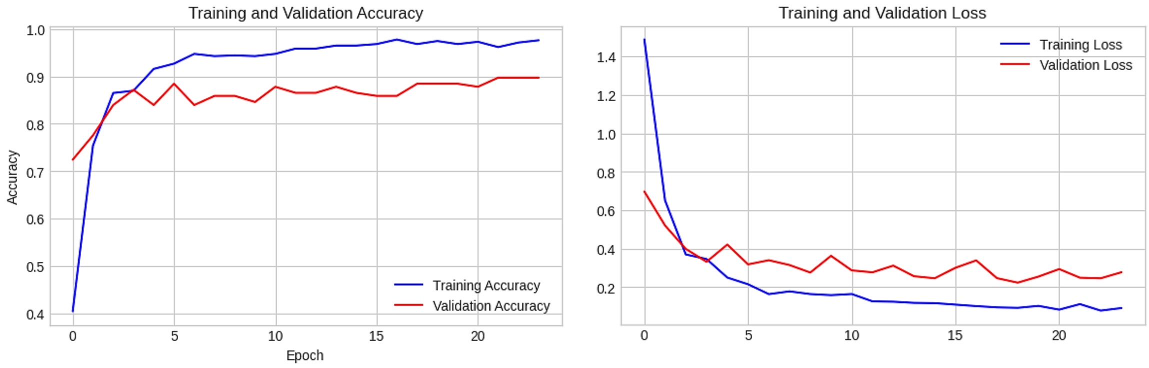

Figure 9. Training and Loss DensNet169

The results in figure 10 show the performance of the DenseNet169 CNN model. On the training data, the model has a loss of 0.069 and an accuracy of 98%, indicating effective learning. On the validation data, the loss rose to 0.223 and the accuracy dropped to 88.4%, indicating slight overfitting. On the test data, the loss was 0.179 and the accuracy was 92.3%, indicating good generalization. Overall, the model performed very well.

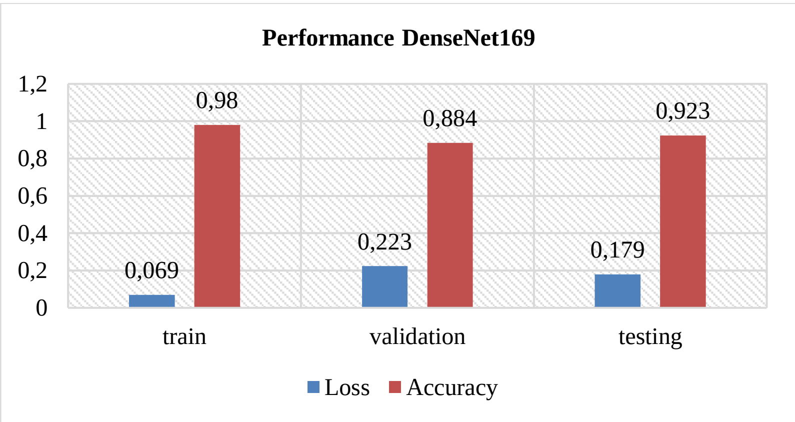

Figure 10. DenseNet169 comparison diagram

The DenseNet model showed good performance with an accuracy of 0.98 on the training data and a loss of 0.069, but there was a slight decrease on the validation data with an accuracy of 0.884 and a loss of 0.223. On the test data, the model achieved an accuracy of 0.923 with a loss of 0.179. A smaller loss value indicates a more accurate model, as the loss measures how much the model's prediction error is.

3.6. CNN modeling using NasNet Mobile Architecture

The following figure shows the results of Convolutional Neural Network (CNN) modeling using the NasNetMobile architecture. NasNetMobile is a lightweight version of the Neural Architecture Search Network (NASNet), which is meant for devices that have limited resources. This architecture is intended to achieve high performance while remaining efficient in terms of the number of parameters and computing power used. Training and Loss NasNet Mobile Architecture can view figure 11.

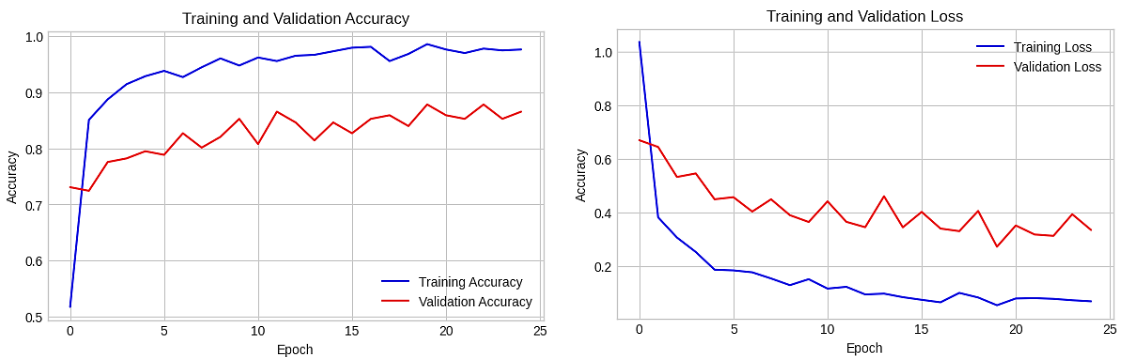

Figure 11. Training and Loss NasNet Mobile Architecture

Figure 12 shows excellent performance in all evaluation phases. During training, the model achieved low loss (0.086) and high accuracy (0.965), showing a strong ability to learn patterns from the training data. Despite a slight increase in loss and a decrease in accuracy in the validation data (loss: 0.272, accuracy: 0.878), the model remained consistent in its performance. In the testing stage, although the loss increased slightly (0.284), the accuracy increased to 0.899, demonstrating the model's ability to deal well with new data. Overall, the model showed good generalization without showing significant overfitting.

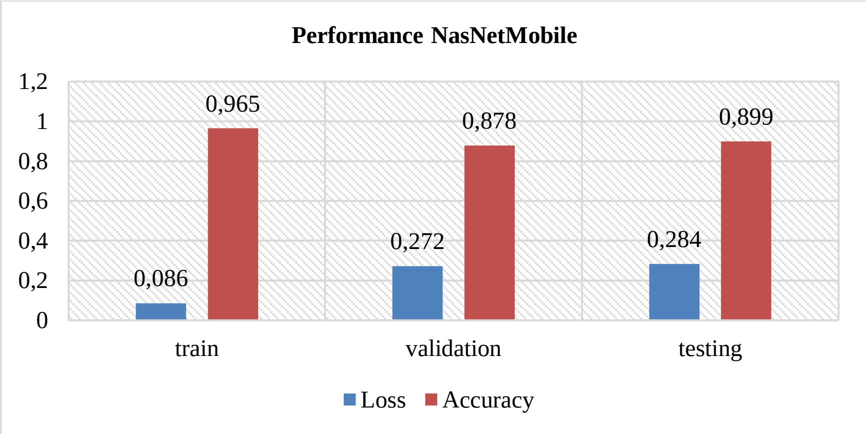

Figure 12. NasNet Mobile comparison diagram

The NASNet Mobile model showed good results despite a slight drop in performance between the training data and the validation data. On the training data, the accuracy reached 0.965 with a loss of 0.086, indicating the model could learn well from the training data. However, on the validation data, the accuracy dropped to 0.878 and the loss increased to 0.272. On the test data, the model showed an accuracy of 0.899 with a loss of 0.284, which is still relatively good despite a slight decrease compared to the training data. A smaller loss value indicates that the model has a more accurate prediction.

3.7 Comparison result of architecture accuracy value

After the testing process is complete, an accuracy comparison is performed on the test, validation, and training data to determine the best model performance. Figure 13 shows the results of the accuracy comparison. InceptionV3 has a far higher validation accuracy than DenseNet169 and NasNet Mobile, even though Figure 13 demonstrates that the DenseNet169 architecture has the best training accuracy for classifying weather images. For training data, the InceptionV3 architecture offers the greatest accuracy value. Based on the comparison findings, the InceptionV3 architecture is the finest and most reliable design for weather classification. It's important to take into account how well these models work with photographs of different resolutions, though. The algorithms' capacity to categorize meteorological data may differ significantly depending on whether they are tested using photos with lower or higher resolution. Whether these models' robustness could be increased by tweaking them to handle various resolutions and accuracy could be investigated further.

A review of different methods for weather image classification reveals a variety of techniques and outcomes. Prior studies that integrated CNN with SVM achieved an accuracy of 77.38% on RGB images [34], whereas a method employing CNN with Keras and TensorFlow reached approximately 94% accuracy on the Kaggle dataset [35]. In contrast, our research, which assessed InceptionV3, DenseNet169, and NASNetMobile, yielded even better results, with InceptionV3 achieving the highest accuracy of 97.94% on training data, 92.34% on validation data, and 93.81% on test data. These findings underscore the advantages of advanced CNN architectures compared to hybrid or simpler models. Future investigations could focus on incorporating a wider range of data augmentation and regularization techniques to enhance model generalization, as well as leveraging larger datasets that encompass extreme weather events to further advance understanding in weather image classification.

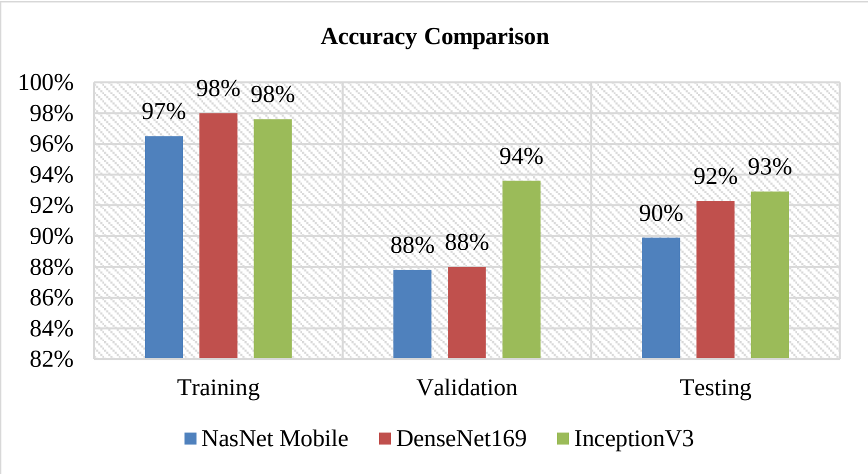

Figure 13. Accuracy Comparison

4. CONCLUSION

In this research, CNN algorithm is used with three different architectures, namely Inception V3, DenseNet169, and NASNetMobile, to recognize weather image data. Inception V3 recorded the highest accuracy with 97.94% on training data, 92.34% on validation data, and 93.81% on test data. DenseNet169 achieved 98.09% accuracy on training data, 88.46% on validation data, and 92.33% on test data. On the other hand, NASNetMobile recorded 96.51% accuracy on training data, 87.82% on validation data, and 89.97% on test data. Based on the final results of the three CNN architectures evaluated, Inception V3 proved to be a more optimal choice for weather data classification applications. Inception V3 showed consistency in achieving high accuracy in all stages of testing, making it a more reliable and effective choice in handling different weather data conditions.

REFERENCES

[1] T. Q. Nam, "Evaluation of Stacked Ensemble Model on Weather Image Recognition," *Procedia Comput Sci*, vol. 234, no. 2023, pp. 1664–1671, 2024, doi: 10.1016/j.procs.2024.03.171.

[2] K. Purwandari, J. W. C. Sigalingging, A. A. Hidayat, T. W. Cenggoro, and B. Pardamean, "Implementation of Computer Vision of Jakarta Weather Image Categorization Using ResNet," *Procedia Comput Sci*, vol. 227, pp. 813–822, 2023, doi: 10.1016/j.procs.2023.10.587.

[3] H. Mo and G. Zhao, "RIC-CNN: Rotation-Invariant Coordinate Convolutional Neural Network," *Pattern Recognit*, vol. 146, no. June 2023, p. 109994, 2024, doi: 10.1016/j.patcog.2023.109994.

[4] N. Zaeri and R. Qasim, "Thermal image identification against pose and expression variations using deep learning," *Journal of Engineering Research (Kuwait)*, no. October, 2024, doi: 10.1016/j.jer.2023.10.043.

[5] M. M. Mashamba, A. Telukdarie, I. Munien, U. Onkonkwo, and A. Vermeulen, "ScienceDirect ScienceDirect Detection of bacterial spot disease on tomato leaves using a Detection of bacterial spot disease on tomato leaves using a Convolutional Neural Network (CNN) Convolutional Neural Network (CNN)," *Procedia Comput Sci*, vol. 237, pp. 602–609, 2024, doi: 10.1016/j.procs.2024.05.145.

[6] V. Kukreja, R. Sharma, and R. Yadav, "Multi-Weather Classification using Deep Learning: A CNN-SVM Amalgamated Approach," *2023 World Conference on Communication & Computing (WCONF)*, pp. 1–5, Sep. 2023.

[7] S. Mittal and O. P. Sangwan, "Classifying Weather Images using Deep Neural Networks for Large Scale Datasets." [Online]. Available: www.ijacsa.thesai.org

[8] 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 Research Journal of Engineering, Data Technology and Computer Science*, vol. 1, no. 2, pp. 56–61, Feb. 2024, doi: 10.57152/predatecs.v1i2.872.

[9] N. Pratama *et al.*, “Perbandingan Model Klasifikasi Transfer Learning Convolutional Neural Network Tumor Otak Menggunakan Citra Magnetic Resonance Imaging,” *Jurnal Sehat Indonesia*, vol. 6, no. 1, 2024.

[10] A. Fuadi *et al.*, “Perbandingan Arsitektur MobileNet dan NASNetMobile untuk Klasifikasi Penyakit Pada Citra Daun Kentang,” Sep. 2022. doi: https://doi.org/10.29100/jipi.v7i3.3026.

[11] L. E. Boucheron, T. Vincent, J. A. Grajeda, and E. Wuest, “Solar active region magnetogram image dataset for studies of space weather,” pp. 1–13, 2023, doi: 10.1038/s41597-023-02628-8.

[12] V. Wulandari, W. J. Sari, Z. Alfian, L. Legito, and T. Arifianto, “Implementasi Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor untuk Klasifikasi Penyakit Ginjal Kronik,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 710–718, 2024, doi: 10.57152/malcom.v4i2.1229.

[13] W. J. Sari *et al.*, “Performance Comparison of Random Forest, Support Vector Machine and Neural Network in Health Classification of Stroke Patients,” *Public Research Journal of Engineering, Data Technology and Computer Science*, vol. 2, no. 1, pp. 34–43, 2024, doi: 10.57152/predatecs.v2i1.1119.

[14] E. Oluwasakin *et al.*, “Minimization of high computational cost in data preprocessing and modeling using MPI4Py,” *Machine Learning with Applications*, vol. 13, no. May, p. 100483, 2023, doi: 10.1016/j.mlwa.2023.100483.

[15] A. Ahmad, X. Xiao, H. Mo, and D. Dong, “Tuning data preprocessing techniques for improved wind speed prediction,” *Energy Reports*, vol. 11, no. September 2023, pp. 287–303, 2024, doi: 10.1016/j.egyr.2023.11.056.

[16] A. Zompola, A. Korfiati, K. Theofilatos, and S. Mavroudi, “Heliyon Omics-CNN : A comprehensive pipeline for predictive analytics in quantitative omics using one-dimensional convolutional neural networks,” *Heliyon*, vol. 9, no. 11, p. e21165, 2023, doi: 10.1016/j.heliyon.2023.e21165.

[17] D. Bhagat, A. Vakil, R. Kumar, and A. Kumar, “ScienceDirect ScienceDirect Facial Emotion Recognition (FER) using Convolutional Neural Network (CNN),” *Procedia Comput Sci*, vol. 235, no. 2023, pp. 2079–2089, 2024, doi: 10.1016/j.procs.2024.04.197.

[18] T. Islam, S. Hafiz, J. Rahman, M. Kabir, and M. F. Mridha, “Healthcare Analytics A systematic review of deep learning data augmentation in medical imaging : Recent advances and future research directions,” *Healthcare Analytics*, vol. 5, no. December 2023, p. 100340, 2024, doi: 10.1016/j.health.2024.100340.

[19] X. Liu, K. Ono, and R. Bise, “A data augmentation approach that ensures the reliability of foregrounds in medical image segmentation,” *Image Vis Comput*, vol. 147, no. February, p. 105056, 2024, doi: 10.1016/j.imavis.2024.105056.

[20] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[21] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999–7019, 2022, doi: 10.1109/TNNLS.2021.3084827.

[22] A. Karthikeyan, S. Jothilakshmi, and S. Suthir, “Measurement : Sensors Colorectal cancer detection based on convolutional neural networks (CNN) and ranking algorithm,” *Measurement: Sensors*, vol. 31, no. November 2023, p. 100976, 2024, doi: 10.1016/j.measen.2023.100976.

[23] N. Islam, S. Mekhilef, H. Pota, and M. A. Abido, “Fault classification and location of a PMU-equipped active distribution network using deep convolution neural network (CNN),” *Electric Power Systems Research*, vol. 229, no. January, p. 110178, 2024, doi: 10.1016/j.epsr.2024.110178.

[24] M. F. A. Ilhami and S. Wibisono, “Klasifikasi Rimpang Menggunakan Metode Jaringan Saraf Konvolusi Dengan Arsitektur Alexnet,” *INTECOMS: Journal of Information Technology and Computer Science*, vol. 6, no. 2, pp. 666–670, 2023, doi: 10.31539/intecoms.v6i2.6634.

[25] K. Joshi, V. Tripathi, C. Bose, and C. Bhardwaj, “Robust Sports Image Classification Using InceptionV3 and Neural Networks,” *Procedia Comput Sci*, vol. 167, no. Iccids 2019, pp. 2374–2381, 2020, doi: 10.1016/j.procs.2020.03.290.

[26] P. Pradeep, D. Damodar, and R. Edla, “Diagnosis of Coronavirus Disease From Chest X - Ray Images Using DenseNet - 169 Architecture,” *SN Comput Sci*, vol. 4, no. 3, pp. 1–6, 2023, doi: 10.1007/s42979-022-01627-7.

[27] P. Enkvetchakul and O. Surinta, “Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition,” no. March 2021, 2022, doi: 10.14416/j.asep.2021.01.003.

[28] M. Rajesh, B. Senapati, R. Das, and S. Martha, “Identifying Colorectal Tumor for Single Cell RNA Sequence Using Rectified Linear Unit with Stochastic Gradient Descent,” *Procedia Comput Sci*, vol. 218, pp. 189–198, 2022, doi: 10.1016/j.procs.2023.01.001.

[29] D. Armiady and I. Muslem R, “Klasifikasi Kualitas Buah Pisang Berdasarkan Citra Buah Menggunakan Stochastic Gradient Descent,” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 2, pp. 1207–1215, 2023, doi: 10.30865/klik.v4i2.1243.

[30] P. Wang, Y. Lei, Y. Ying, and D. X. Zhou, “Differentially private stochastic gradient descent with low-noise,” *Neurocomputing*, vol. 585, no. March 2023, p. 127557, 2024, doi: 10.1016/j.neucom.2024.127557.

[31] 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 Research Journal of Engineering, Data Technology and Computer Science*, vol. 1, no. 2, Feb. 2024, doi: 10.57152/predatecs.v1i2.865.

[32] A. H. Nasrullah, “Implementasi Algoritma Decision Tree Untuk Klasifikasi Data Peserta Didik,” *Jurnal Pilar Nusa Mandiri*, vol. 7, no. 2, p. 217, 2021.

[33] N. D. Miranda, L. Novamizanti, and S. Rizal, “Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50,” *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 61–68, 2020, doi: 10.20884/1.jutif.2020.1.2.18.

[34] Y. Hao and Y. Zhu, “Weather Classification for Multi-class Weather Image Based on CNN,” *2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, pp. 363–366, 2022.

[35] A. Sharma and Z. Saad Ismail, “Weather Classification Model Performance: Using CNN, Keras-Tensor Flow,” *ITM Web of Conferences*, vol. 42, p. 01006, 2022, doi: 10.1051/itmconf/20224201006.