Pre-Trained Convolutional Neural Network Benchmark For Multi-Class Weather Modeling

Sinta Dhea Ramadhany[1], Ervin Yohannes[2]

[1,2] Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya
[1]sinta.21043@mhs.unesa.ac.id
[2]univervinyohannes@unesa.ac.id

***Abstract*— Weather forecasting plays a crucial role in reducing the risks of extreme events that threaten human safety, economic stability, and the environment. Traditional forecasting methods relying on manual observation have developed into modern approaches using satellite, radar, and computational models; however, prediction accuracy remains limited due to the complexity of atmospheric systems and data constraints. In this context, deep learning, particularly Convolutional Neural Networks (CNNs), provides significant potential for automatic weather classification through digital imagery. This study evaluates and compares the performance of four pre-trained CNN architectures VGG16, ResNet50, AlexNet, and InceptionV3 on the Kaggle "Multi-class Weather Dataset," which contains 860 images categorized into four classes: Cloudy, Shine, Rain, and Sunrise. The methodology involves data augmentation, fine-tuning, and systematic experimentation with various hyperparameters and data split ratios to enhance model generalization.**

The evaluation metrics applied include accuracy, precision, recall, and F1-score. Experimental results reveal that InceptionV3 outperforms other models, achieving up to 98% training accuracy and 96% validation accuracy due to its effective multi-scale feature extraction and regularization. ResNet50 delivers balanced results with validation accuracy up to 94%, while AlexNet records relatively high detection counts but lower overall performance. In contrast, VGG16 yields the lowest accuracy among the tested models. These findings highlight InceptionV3 as the most robust architecture for weather image classification and emphasize the importance of model selection in balancing prediction accuracy and computational efficiency. The study contributes as a foundation for the development of deep learning-based weather recognition systems that can support early warning applications and disaster risk reduction.

***Keywords*— Convolutional Neural Network, Weather Classification, ResNet50, VGG16, AlexNet, InceptionV3.**

I. INTRODUCTION

Weather refers to the atmospheric condition within a specific region and short time frame, influenced by factors such as air pressure, temperature, and humidity. Variations in temperature due to differences in the angle of solar incidence across latitudes drive atmospheric circulation, jet streams, and large-scale weather systems, including cyclones and tropical storms [1][2]. According to the Indonesian Meteorological, Climatological, and Geophysical Agency (BMKG), weather is defined as the short-term state of the atmosphere at a specific time and place [3]. Historically, weather has significantly impacted human activities, particularly in agriculture, which heavily relies on meteorological conditions such as rainfall and daily temperatures. Extreme weather events can lead to severe consequences, including crop failures, disruptions in logistics and transportation, accidents, and communication network failures [4].

Weather forecasting is critical for predicting future atmospheric conditions. While traditional forecasting relied on manual observations, modern methods leverage supercomputers to process data from satellites, radars, and ground-based sensors [5]. BMKG has noted an annual temperature increase in Jakarta, signaling the growing impact of climate change and extreme weather events. Consequently, robust weather and climate information systems are essential for monitoring, evaluating, classifying patterns, and predicting weather conditions continuously [6]. Despite technological advancements, forecasting accuracy remains challenged by the atmosphere's complexity, assumptions in mathematical models, and limitations in observational data [7].

Addressing weather-related challenges is vital for safeguarding lives, economies, and the environment. Technologies such as Convolutional Neural Networks (CNNs) enable accurate weather predictions and early warning systems [8], while media outlets like CNN enhance public awareness and preparedness [9]. Effective weather forecasting supports agriculture and energy sectors by mitigating economic losses and helps prevent health crises caused by disasters such as floods and heatwaves. Conversely, inadequate handling of weather-related issues can escalate risks of disasters, economic losses, health crises, and environmental degradation, leading to costly infrastructure damage, overwhelmed healthcare systems, and irreversible ecosystem degradation. Thus, a proactive approach is crucial to address the increasing frequency of extreme weather events.

Multi-class weather image classification poses a significant challenge in computer vision, particularly for automated weather forecasting and early warning systems. Convolutional Neural Networks (CNNs) have become the cornerstone for developing image classification models due to their ability to automatically extract hierarchical features. Pre-trained architectures such as VGG16, ResNet50, AlexNet, and InceptionV3 are widely utilized, as transfer learning enables rapid adaptation to specific domains like weather imagery. However, selecting the optimal architecture remains a topic of debate, as each model has distinct strengths and limitations in terms of accuracy, computational complexity, and resource requirements [10].

Research by Prof. Dr. Afroza Nahar et al. (2025) evaluated the performance of ResNet101, VGG16, and InceptionResNetV2 on a weather dataset with an 80:10:10 training split, finding that ResNet101 achieved the highest accuracy of 83.39%. Similarly, Mürüvvet Kalkan et al. (2022) reported a 91.4% accuracy using VGG16 for classifying clear and cloudy weather, though limited to a dataset of 5,000 images. Meanwhile, Kartika Purwandi et al. (2023) demonstrated that ResNet-18 (86.6% accuracy) outperformed ResNet-50 and ResNet-34 on a small-scale Jakarta weather dataset, highlighting the risk of overfitting in overly complex models. Comparable complexity was observed in Tran Quy Nam's (2024) study, where an ensemble of VGG19 and InceptionV3 achieved 91.47% accuracy but required extensive training time and computational resources.

Other studies emphasize the importance of architecture optimization. Md Nasim Khan et al. (2022) reported a 97.3% accuracy in weather detection using ResNet18 on a CCTV dataset, while Chunzhu Meng (2024) showed that a custom CNN outperformed VGG19 (97% accuracy) in four-class weather classification, albeit requiring intensive data augmentation.

Previous studies have not comprehensively compared the performance of VGG16, ResNet50, AlexNet, and InceptionV3 in the context of multi-class weather classification. Many studies rely on limited datasets or complex ensemble approaches, leaving the need for a balanced model that optimizes both accuracy and computational efficiency unaddressed. This study aims to fill this gap by conducting a comparative evaluation of these four CNN architectures on an expanded multi-class weather dataset, incorporating data augmentation and fine-tuning to enhance model generalization.

This research investigates the effectiveness of deep learning models in classifying weather conditions based on digital imagery. The models are developed using CNN architectures, with the dataset sourced from Kaggle's "Multi-class Weather Dataset," encompassing various weather conditions such as clear, cloudy, rainy, and sunrise. Four distinct weather classification models VGG16, ResNet50, AlexNet, and InceptionV3 are proposed to evaluate the performance of CNN architectures in recognizing visual weather patterns. This study aims to identify the most optimal and accurate deep learning model for weather condition recognition based on digital imagery.

II. METHODOLOGY

This section describes the methodology for developing a multi-class weather classification system using pre-trained CNN architectures (ResNet50, AlexNet, InceptionV3, VGG16), covering the research approach, dataset, and model design.

A. Research Methodology

This study employs Deep Learning with pre-trained Convolutional Neural Network (CNN) architecturesResNet50, AlexNet, InceptionV3, and VGG16 for multi-class weather image classification. The methodology includes the following key stages:

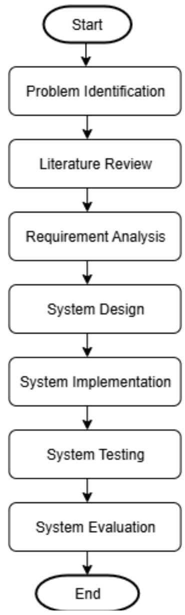

Figure I. Research Methodology

a) Problem Identification

The study addresses challenges in accurately classifying weather conditions from digital images, including variations in lighting, angles, backgrounds, and resolution, as well as issues with limited or imbalanced datasets. The goal is to develop a robust CNN model that avoids overfitting and generalizes well to new data.

b) System Design and Implementation

The system is designed to classify weather images into four categories (Cloudy, Shine, Rain, Sunrise) using the selected CNN architectures. The process involves dataset loading, hyperparameter tuning (e.g., learning rate, epochs, batch size), model training, and testing with unseen data to evaluate generalization. Each model (AlexNet, ResNet50, VGG16, InceptionV3) follows a similar workflow: loading the dataset, training to optimize weather pattern recognition, and evaluating performance using standard metrics.

c) System Evaluation

Model performance is assessed using metrics such as accuracy, precision, recall, and F1-score to determine the effectiveness of each architecture in classifying weather conditions

B. Dataset

The dataset, sourced from Kaggle's "Multi-class Weather Dataset," comprises 860 images across four classes: Cloudy (214), Shine (214), Rain (207), and Sunrise (225). The dataset is split according to the needs of training and validation to ensure robust model training and objective evaluation.

Table I
Dataset

Class Name	Number of Image
Cloudy	215
Shine	215
Rain	215
Sunrise	215
Total	860

C. Model Design

Four pre-trained CNN architectures are utilized: AlexNet: Processes images through convolutional layers with tuned hyperparameters to classify weather conditions.

ResNet50: Leverages residual connections for deeper learning, trained to optimize weather pattern recognition.

VGG16: Uses stacked convolutional layers for feature extraction, fine-tuned for weather classification.

InceptionV3: Employs inception modules for efficient multi-scale feature processing, trained for accurate classification.

Each model is trained on the dataset, with hyperparameters adjusted to maximize performance. Testing is conducted with random external images to evaluate generalization, and results are analyzed to identify the most effective architecture for multi-class weather classification.

III. EXPERIMENT

This section presents the experimental setup, implementation, and evaluation of the multi-class weather classification system using pre-trained Convolutional Neural Network (CNN) architectures: ResNet50, VGG16, AlexNet, and InceptionV3. The experiments aim to analyze the performance of these models in recognizing weather conditions from digital images, focusing on accuracy, generalization, and the impact of hyperparameter configurations and data splitting ratios.

A. Experimental Setup

The experiments were conducted using Python on Google Colab, leveraging several libraries to support model development and evaluation

Table II.
Libraries Used

No	Library	Function
1	Tensorflow	Building and training machine learning and deep learning models
2	Keras	Python API for constructing and training deep learning models
3	Numpy	Managing arrays and performing numerical operations
4	Matplotlib	Creating data visualizations and graphs
5	Scikit-learn	Providing machine learning algorithms and model evaluation tools

B. Dataset Loading

The dataset, stored at "/content/drive/MyDrive/dataset_baru" on Google Drive, was loaded for training and testing. It consists of 860 images across four weather classes: Cloudy (215), Shine (215), Rain (215), and Sunrise (215). The dataset was split into training and validation sets based on varying ratios (50:50, 60:40, 70:30, 80:20, 90:10) to evaluate the impact of data distribution on model performance.

C. Model Training

Training was performed using the model.fit() function, with the training dataset (train_ds) as the primary input and the validation dataset (val_ds) for monitoring performance. Key hyperparameters included the number of epochs, batch size, and learning rate, which were adjusted to optimize model learning. The training process was tracked using the history object, recording metrics such as accuracy and loss for both training and validation sets at each epoch.

D. Performance Analysis

The performance of each model was evaluated through accuracy and loss curves, as well as classification metrics (precision, recall, F1-score). Training accuracy reached up to 98%, with validation accuracy stabilizing at 96%, indicating effective learning without significant overfitting. Loss curves showed a consistent decrease, reflecting the models' ability to minimize prediction errors and generalize to unseen data.

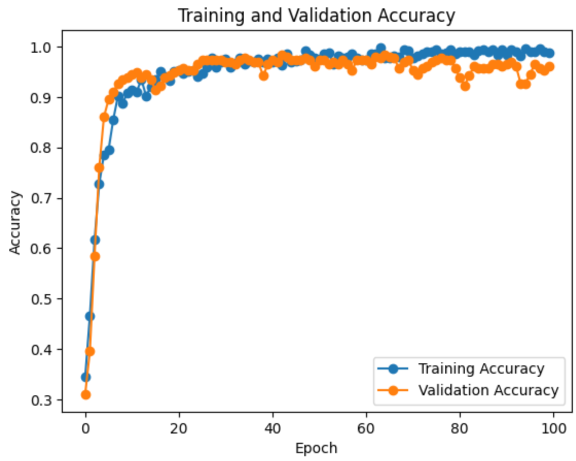

Figure II. Training and Validation Accuracy Curves

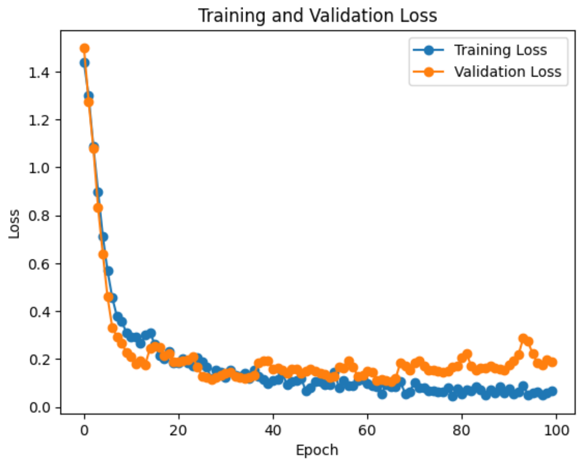

Figure III. Training and Validation Loss Curves

E. Evaluation Metrics

The classification report revealed balanced performance across the four classes, with precision, recall, and F1-scores ranging from 0.24 to 0.30, and an overall test accuracy of 26%. Despite high training (98%) and validation (96%) accuracies, the lower test accuracy suggests challenges in distinguishing classes with similar visual features. Visualizations of classification results are shown below.

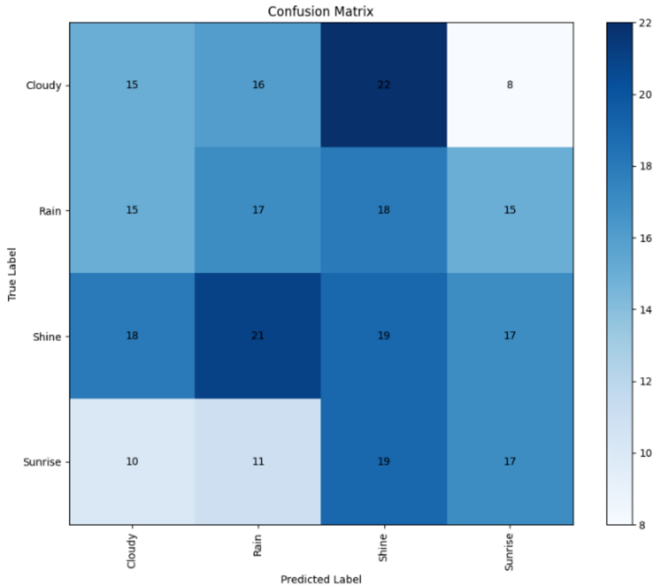

Figure IV. Classification Performance Visualization

F. Model Evaluation

Each model was evaluated across different data split ratios (50:50, 60:40, 70:30, 80:20, 90:10) to assess the impact of data distribution on performance. The top five configurations for each model, based on training and validation accuracy, are presented below

1. ResNet50 Evalution

The ResNet50 model was tested across various data splits, with results summarized in the following tables

Table III.
ResNet50 with Data Split 50:50

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,01	24	100	90	91
0,001	24	100	91	93
0,001	8	50	89	93
0,001	24	50	90	91
0,0001	32	50	74	90

Table IV.
ResNet50 with Data Split 60:40

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	16	10	80	91
0,001	8	100	88	93
0,001	32	100	88	93
0,001	16	50	87	94
0,01	24	10	81	90

Table V.
ResNet50 with Data Split 70:30

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	8	100	88	93
0,001	32	100	88	93
0,0001	8	100	88	94
0,01	32	25	88	90
0,001	16	50	87	94

Table VI.
ResNet50 with Data Split 80:20

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	16	100	90	88
0,001	24	100	88	91
0,0001	8	100	87	91
0,0001	16	100	87	91
0,001	8	50	87	91

Table VII.
ResNet50 with Data Split 90:10

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	16	25	83	90
0,001	24	50	87	89
0,0001	8	50	82	93
0,001	32	50	85	91
0,001	16	10	84	90

2. VGG16 Evaluation

The VGG16 model was tested across various data splits, with results summarized in the following tables.

Table VIII.
VGG16 with Data Split 50:50

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	8	50	89	90
0,0001	24	100	93	86
0,001	16	100	85	87
0,001	16	50	84	81
0,0001	32	10	79	80

Table IX.
VGG16 with Data Split 60:40

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	8	25	85	86
0,0001	32	50	89	85
0,0001	32	50	89	85
0,0001	16	10	81	83
0,001	8	25	76	83

Table X.
VGG16 with Data Split 70:30

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	8	25	85	86
0,0001	32	50	89	85
0,001	32	50	84	81
0,0001	16	10	81	83
0,0001	24	100	91	90

Table XI.
VGG16 with Data Split 80:20

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	32	100	88	86
0,01	16	50	84	91
0,001	24	100	88	84
0,0001	24	25	87	87
0,001	24	25	75	86

Table XII.
VGG16 with Data Split 90:10

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	8	100	84	83
0,001	16	50	84	91
0,001	32	10	72	81
0,0001	32	25	86	81
0,001	8	50	76	80

3. AlexNet Evaluation

The AlexNet model was tested across various data splits, with results summarized in the following tables

Table XIII.
AlexNet with Data Split 50:50

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	24	50	84	85
0,0001	16	100	85	89
0,0001	24	25	79	76
0,0001	32	25	78	86
0,001	24	25	69	73

Table XIV.
AlexNet with Data Split 60:40

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	24	100	81	84
0,001	24	100	86	83
0,001	16	100	75	83
0,001	32	25	67	69
0,01	16	100	67	65

Table XV.
AlexNet with Data Split 70:30

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	32	25	78	88
0,001	24	100	79	84
0,001	32	25	73	76
0,001	32	100	85	88
0,0001	16	50	75	86

Table XVI.
AlexNet with Data Split 80:20

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,001	16	50	79	87
0,001	32	100	87	89
0,001	24	50	81	69
0,01	32	100	84	80
0,001	32	25	74	68

Table XVII.
AlexNet with Data Split 90:10

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	16	100	80	90
0,0001	24	100	82	87
0,001	16	50	79	74
0,001	32	10	71	75
0,001	8	50	67	79

4. InceptionV3 Evaluation

The InceptionV3 model was tested across various data splits, with results summarized in the following tables

Table XVIII.
InceptionV3 with Data Split 50:50

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	16	25	92	93
0,0001	16	50	93	95
0,0001	24	25	96	93
0,0001	32	50	97	93
0,0001	8	10	73	91

Table XIX.
InceptionV3 with Data Split 60:40

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	24	10	86	93
0,0001	16	25	90	93
0,0001	32	25	90	91
0,0001	32	100	97	91
0,0001	32	10	85	90

Table XX.
InceptionV3 with Data Split 70:30

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	32	25	94	94
0,0001	16	10	90	94
0,0001	16	25	91	95
0,0001	32	10	91	93
0,000	32	100	98	96

Table XXI.
InceptionV3 with Data Split 80:20

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	32	10	90	91
0,0001	32	50	97	95
0,0001	24	10	89	91
0,001	32	50	88	84
0,0001	32	50	97	95

Table XXII.
InceptionV3 with Data Split 90:10

LEARNING RATE	BATCH SIZE	EPOCH	ACCURACY	VALIDATION ACCURACY
0,0001	16	25	92	95
0,0001	16	50	97	90
0,0001	32	50	95	93
0,0001	32	10	89	89
0,0001	32	25	94	89

5. Hyperparameter Configuration

Hyperparameters were tuned to optimize model performance. The final configuration included:
Image Size: (100, 100)
Batch Size: 32
Learning Rate: 0.0001
Epochs: 100
Data Split: 70:30 (30% validation)
These values were selected based on iterative experiments to balance training stability and accuracy.

6. InceptionV3 Performance Analysis

InceptionV3 exhibited competitive performance, particularly with the 70:30 split, achieving 98% training accuracy and 95% validation accuracy using a learning rate of 0.0001, batch size of 32, and 50 epochs. Classification metrics showed precision, recall, and F1-score at 23%, with a test accuracy of 39% (16/21 images correctly classified). The model's ability to extract multi-scale features via parallel kernels (1x1, 3x3, 5x5), combined with dropout and batch normalization, contributed to its stability and generalization. Visual results for 21 test images are shown below, with correct classifications often linked to distinct visual features (e.g., contrast, texture), while errors occurred in images with ambiguous features.

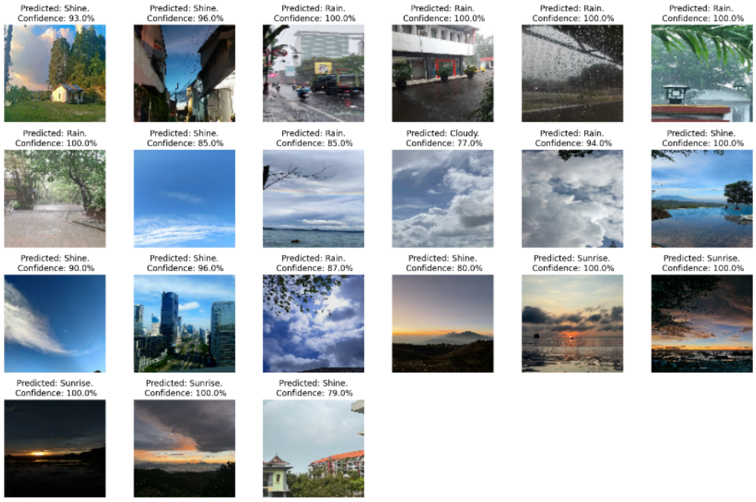

Figure V. InceptionV3 Test Results Visualization

IV. CONCLUSION

This study investigated the application of Deep Learning for multi-class weather image classification using four pre-trained Convolutional Neural Network (CNN) architectures: ResNet50, VGG16, AlexNet, and InceptionV3. Each model was trained and tested on a dataset of 860 weather images, divided into various training and validation splits (50:50, 60:40, 70:30, 80:20, 90:10) and optimized with different hyperparameter configurations, including learning rate, batch size, and number of epochs. The training process enabled the models to learn distinct visual patterns associated with four weather classes—Cloudy, Shine, Rain, and Sunrise—while testing evaluated their ability to generalize to unseen images. Comparative analysis revealed that InceptionV3 consistently outperformed the other architectures, achieving a training accuracy of 98% and a validation accuracy of 96%, demonstrating robust generalization and stability. Although AlexNet recorded the highest number of detections, its training and validation accuracies were lower at 89%. ResNet50 exhibited balanced performance with a validation accuracy of up to 94%, while VGG16 showed the lowest performance in both accuracy and detection metrics. The superior performance of InceptionV3, attributed to its ability to capture multi-scale features through parallel kernel operations and effective regularization, makes it the recommended model for weather image classification tasks, particularly when prioritizing high accuracy and reliable generalization across diverse weather conditions. These findings provide a foundation for further refinement of CNN-based weather classification systems and their application in real-world scenarios.

REFERENCES

[1] Susilo B. Mengenal Iklim dan Cuaca di Indonesia n.d. https://books.google.co.id/books?id=Cl5zEAAAQBAJ (accessed January 14, 2025).
[2] Fahmy Fadholy. Cuaca. Yogyakarta: RELASI INTI MEDIA; 2023.
[3] BUKU SAKU KLIMATOLOGI: IKLIM DAN CUACA KITA n.d.

https://iklim.bmkg.go.id/bmkgadmin/storage/brosur/Buku Saku_KLIMATOLOGI_bnew .pdf.
[4] Lakshmi G. Weather Classification Using Deep Convolutional Neural Network 2024;6:1–5.
[5] Ghaleb M, Moushier H, Shedeed H, Tolba M. Weather Classification using Fusion Of Convolutional Neural Networks and Traditional Classification Methods. Int J Intell Comput Inf Sci 2022;0:1–13. https://doi.org/10.21608/ijicis.2022.117060.1156.
[6] Purwandari K, Sigalingging JWC, Hidayat AA, Cenggoro TW, Pardamean B. Implementation of Computer Vision of Jakarta Weather Image Categorization Using ResNet. Procedia Comput Sci 2023;227:813–22. https://doi.org/10.1016/j.procs.2023.10.587.
[7] Tilasefana RA, Putra RE. Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca. J Informatics Comput Sci 2023;05:48–57.
[8] Kukreja V, Sharma R, Yadav R. Multi-Weather Classification using Deep Learning: A CNN-SVM Amalgamated Approach. 2023 World Conf Commun \& Comput 2023:1–5.
[9] Gyamfi PA. Comparative analysis of CNN coverage of weather-related disasters in USA, Japan, and India. Media Asia 2021;48:21–33.
[10] Mittal S, Sangwan OP. Classifying Weather Images using Deep Neural Networks for Large Scale Datasets. Int J Adv Comput Sci Appl 2023;14:337–43. https://doi.org/10.14569/IJACSA.2023.0140136.
[11] Eesd E, Ahmed Z. Weather identification using models based on deep learning. Sustain Dev 2011;92:2771642. https://doi.org/10.22581/muet1982.2905.
[12] Kalkan M, Bostancı GE, Güzel MS, Kalkan B, Özsarı Ş, Soysal Ö, et al. Cloudy/clear weather classification using deep learning techniques with cloud images. Comput Electr Eng 2022;102:108271. https://doi.org/10.1016/j.compeleceng.2022.108271.
[13] Nam TQ. Evaluation of Stacked Ensemble Model on Weather Image Recognition. Procedia Comput Sci 2024;234:1664–71. https://doi.org/10.1016/j.procs.2024.03.171.
[14] Khan MN, Ahmed MM. Weather and surface condition detection based on road-side webcams: Application of pre-trained Convolutional Neural Network. Int J Transp Sci Technol 2022;11:468–83. https://doi.org/10.1016/j.ijtst.2021.06.003.