Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2905

2025, 44(1) 44-51

Weather identification using models based on deep learning

Afroza Nahar [a, *], Rifat Al Mamun Rudro [a], Bakhtiar Atiq Faisal [a], Md. Faruk Abdullah Al Sohan [a], Laveet Kumar [b]

[a] *Department of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh*

[b] *Department of Mechanical Engineering, Mehran University of Engineering and Technology, Jamshoro, 76090 Sindh, Pakistan*

[*] Corresponding author: Afroza Nahar, Email: afroza@aiub.edu

Received: 11 July 2023, Accepted: 26 December 2024, Published: 01 January 2025

KEYWORDS

Weather Forecasting

Deep Learning

TensorFlow

Keras Applications

Image Classification

Weather Phenomena Database

ABSTRACT

Accurate weather forecasting is increasingly crucial as climate change intensifies the unpredictability of weather patterns, posing challenges to traditional forecasting models reliant on human observation or numerical methods. Researchers are working on precise weather forecasting to improve our preparedness, enabling fast response to any disaster. Among other techniques, deep learning is a prudent method to predict weather forecasts since it can automatically learn and train from a vast amount of data to generate and portray accurate features of an incident. This study evaluates deep learning techniques for weather forecasting based on different meteorological characteristics. This paper examines a few weather variables to evaluate the prediction performance of several deep learning solutions using TensorFlow and pre-trained Keras applications models. For this purpose, the top ten accuracy-based deep learning model architectures have been investigated and evaluated. The operation of each model is distinct. Models like EfficientNetB7, ResNet, MobileNet, VGG19, Xception Inception, ResNetV2, and VGG16 employ a combination of image classification and deep learning models to predict the weather. The WEAPD dataset of 6877 images representing 11 weather phenomena categories was utilized, and the models were trained and validated using an 80:10:10 split. Predictions, extraction of features, and fine-tuning of models were achieved with an accuracy of up to 83.39%. Most models performed well in image classification, enhancing the proposed framework and achieving significant precision in generating weather photos and reports.

1. Introduction

Nowadays, concerns about the world's climate are growing among the civil society because many scientists have already confirmed much undeniable evidence regarding global warming [1]. On the other hand, agriculture, travelling, sports, and many other events in our life profoundly rely on precise weather forecasts [2]. Varying weather conditions have their own impacts on economic activities as well as our everyday life. So, accurate weather forecasting can improve human and environmental security and generate economic benefits [3]. Therefore, researchers are attempting to employ diverse methods to improve weather forecasting.

In the past few years, several weather forecasting techniques and models have been developed that are based on numerical and historical data. Forecasting is a stochastic and computationally intensive process and wholly dependent on the precision of the data. Since weather is a dynamic system, it is quite difficult to achieve accuracy in numerical models. Apart from that, simulation mistakes hamper the validity of the model which emerge from three primary sources like faulty input data, flawed physics models, imprecise numerical solutions, etc. Traditional techniques of classifying meteorological events mostly rely on human observation. And our capacity to forecast any physical phenomenon is dependent upon the precision of our input data and our modelling strategy [4]. Nevertheless, the typical artificial visual difference between meteorological occurrences is time-consuming and error prone. Therefore, there is an urgent need to create precise, efficient, and automated technology for classifying meteorological events.

Deep learning, a specialized subfield of machine learning, has evolved quickly in recent years and been applied to numerous arenas [3]. Deep learning has found widespread application in disciplines as diverse as computer vision, image processing, natural language processing, and many more due to its capacity to automatically learn and train from enormous amounts of sample data to develop higher feature expression [2]. Deep learning is adept at approximating nonlinear systems and extracting high-dimensional characteristics. As the basis and primary driving force of deep learning, deep neural networks are less concerned with numerical modelling. Deep neural network (DNN) utilizes a data-driven approach, where models are constructed using data-driven learning [4]. The Weather Phenomena Database (WEAPD) from the Harvard Data verse was used as a dataset. There were 6877 photos available in 11 weather phenomenon categories.

In this research, we have used TensorFlow and Keras' top ten accuracy-based deep learning model architectures for evaluating accuracy in weather forecasting. Google's TensorFlow is a sophisticated open-source software library for machine learning and deep learning. Keras is a high-level application programming interface (API) for assembling building blocks to design and train deep learning models, and it can be integrated with TensorFlow. We implemented various deep learning model architectures including Resnet, MobileNet, DenseNet201, EfficientNetB7, and VGG19, among others. When done by the model layer, our proposed models performed exceptionally well in terms of picture categorization. Each model operates in a unique way. The model architecture provides the best level of predictive performance with the aid of these model representations of image resolution and model layer execution procedures. It could improve the proposed framework and deliver up to 80% accuracy in predicting weather from photographs and classification reports. To improve weather forecasting, our models based on deep learning achieved satisfactory results, validating the research.

2. Literature Review

The application of neural networks and deep learning techniques in weather prediction has been extensively studied, demonstrating diverse approaches to tackle the challenges in accuracy and reliability. Salman et al. conducted a foundational study employing recurrent neural networks (RNNs) optimized heuristically for rainfall prediction using a dataset composed of ENSO (El Niño-Southern Oscillation) variables such as wind, SOI, SST, and OLR [1]. In their experiments, rainfall was the dependent variable, and three different leaps—1, 2, and 3—were evaluated for predictive performance. Notably, leap 1 emerged as the most effective, yielding an R^2 value of 84.8% and an RMSE of 125 in the first experiment. Comparatively, the second experiment exhibited a lower R^2 value of 59.9% with an RMSE of 155.29. These findings underscore the potential of RNNs with robust backpropagation for rainfall forecasting, highlighting the sensitivity of the model's performance to data segmentation and parameter optimization.

In another study, Fang et al. proposed and tested the MeteCNN model for weather event classification, achieving significant improvements over traditional models [2]. Using their weather dataset, the MeteCNN architecture attained an impressive classification accuracy of 92.68%, surpassing ResNet18, which achieved an accuracy of 88.73%, by approximately 4%. The precision, recall, and F1-score of MeteCNN hovered around 93%, underscoring its reliability in classification tasks. This model's superiority in handling weather event classification suggests that tailored deep learning architectures can significantly outperform generalized models, particularly when optimized for specific datasets and weather phenomena. Building on advancements in convolutional neural networks (CNNs), Xiao et al. developed a CNN-based deep learning model for weather classification, trained on a Kaggle dataset [3]. The model achieved initial metrics of 94% training accuracy and 92% validation accuracy, with training and validation losses at 18% and 22%, respectively. While promising, Xiao et al. acknowledged the need for further development, including integrating additional attributes such as humidity, precipitation, air pressure, and solar radiation. These enhancements aim to improve the model's robustness and applicability to a broader range of weather scenarios, demonstrating the iterative nature of developing weather prediction systems. Further comparative studies, such as those by Arcucci et al., explored the effectiveness of neural networks against traditional downscaling methods like ensemble means, multiple linear regression, and regional models [4].

Downscaling, critical for refining low-resolution forecasts, was tested using seven-day rainfall predictions. Results showed that neural networks could capture storm-related details, particularly over the Amazon, more effectively than traditional models. These findings align with the performance of models like ResNet, emphasizing the value of neural networks in extracting nuanced weather patterns that conventional methods may overlook. Other researchers have explored innovative neuronal frameworks tailored for various weather conditions, such as clear, cloudy, and overcast days [5, 6]. Using multivariate neural networks (NNE) combined with wavelet-adjusted outputs of PV, solar irradiance, wind speed, temperature, and moisture, they demonstrated significant enhancements in prediction accuracy [9]. This method utilized clarity indices to classify days and employed a trimming aggregation approach to combine upper and lower prediction bounds effectively. Results indicated that the proposed framework consistently outperformed individual and benchmark models, showcasing its utility in complex, multivariate weather scenarios [10].

3. Methodology

Deep learning models are used to anticipate object classification or prediction. In this study, we suggest the use of a picture dataset to forecast the weather images. There are a total of 6877 images in the collection, which includes 11 types of meteorological events. The WEAPD dataset was split into three sets: a training set, a validation set, and a testing set, with no parts of any two pictures crossing across. This research seeks to evaluate deep learning approaches for weather forecasting. To address these challenges, this study investigates the application of deep learning models to automate and enhance the accuracy of weather forecasting [11]. Specifically, it evaluates ten architectures for their performance in predicting weather categories based on image data. Fig. 1 shows the workflow of the methodology.

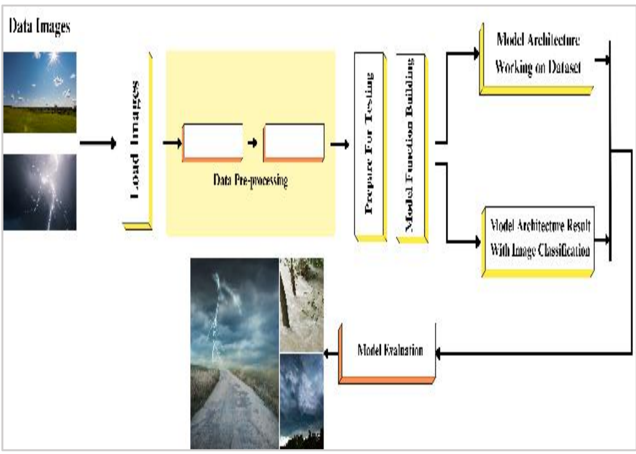

Fig. 1. Methodology Workflow

3.1 Data Classification

It is crucial to create a large dataset of properly labelled photos of meteorological events. Therefore, we collected the dataset called the Weather Phenomena Database (WEAPD) from the Harvard Data verse, where 11 categories of weather phenomena are available comprising a total of 6877 photos. The dataset WEAPD was divided into a training set, a validation set, and a testing set in a ratio of 80:10:10, and no portion of any one picture overlaps with any other.

Within the dataset, the different types of weather are broken down into eleven distinct groups, where the following terms are included in each category: rime, fog, smog, dew, sandstorm, glaze, snow, hail, rain, frost, and lightning [12]. There is an eight-to-one validation ratio in the train dataset. We have integrated the deep learning models library into the Keras program so that we may identify photos in accordance with a variety of different model principles. Fig. 2 shows the data distribution in the dataset.

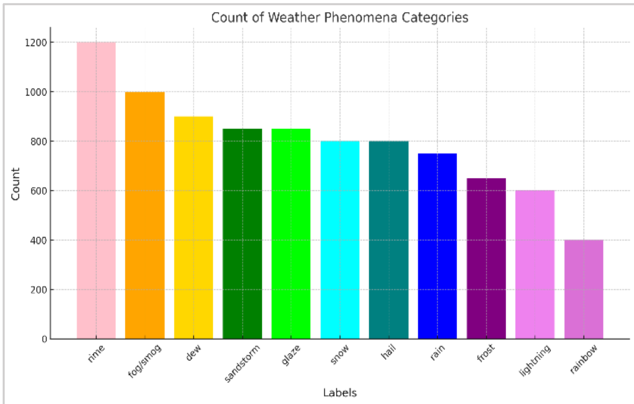

Fig. 2. Data Distribution in Dataset

3.2 Models

3.2.1 EfficientNetB7

In this study, one of the significant advancements in weather identification involves the use of deep learning models to analyze and classify weather patterns accurately. By using the EfficientNet, it uses compound scaling to uniformly adjust depth, breadth, and resolution with a single scaling coefficient. This approach ensures that larger input images are paired with deeper networks for a broader receptive field and wider networks to capture finer details, optimizing model performance [13]. The principle behind this approach is rooted in the observation that larger input images require deeper networks to expand the receptive field and wider networks to capture finer details.

By modifying the weights, each iteration is carefully designed and tested to deliver optimal outcomes, occasionally deviating from the scaling formula for improved performance in specific scenarios [14]. Table 1, illustrates the classification performance of EfficientNet across multiple classes, showcasing its ability to consistently deliver high accuracy and robust results across datasets.

Table 1

Classification report

Data Class	Precision	Recall	F1-Score	Support
Dew	0.85	0.85	0.85	145
Fog/smog	0.68	0.94	0.79	176
Frost	0.66	0.45	0.53	87
Glaze	0.66	0.65	0.64	116
Hail	0.74	0.96	0.84	113
Lightning	0.75	0.93	0.83	74
Rain	0.84	0.88	0.86	230
Rainbow	0.94	0.89	0.91	53
Rime	0.94	0.95	0.94	230
Sandstorm	0.94	0.55	0.70	50
Snow	0.62	0.57	0.59	121

3.2.2 ResNet

In the context of weather identification and prediction, Shaoqing et al.'s Residual Network (ResNet) offers a transformative approach for handling deep neural networks [7]. ResNet's architecture, which incorporates residual blocks with skip connections, is particularly relevant for analyzing complex weather datasets that require deep models to capture intricate spatial and temporal patterns.

The skip connections in ResNet facilitate the efficient training of deep networks by mitigating issues like vanishing gradients, a common challenge when processing high-dimensional data such as weather attributes. In weather identification tasks, these connections allow the network to retain crucial information from earlier layers while also learning fine-grained transformations, leading to improved accuracy and generalization. In the absence of the skip connection in the model architectur, the input x is first multiplied by the weights of the layer, and then a bias term is added to the product. After that, the activation function (f) is called, and the result is H(x).

$$H(x) = f(w * x + b) \quad (1)$$

Due to the implementation of architectural design of an innovative skip connection method, the output H(x) is transformed into in Eq. 2.

$$H(x) = f(x) + x \quad (2)$$

In the design influenced by VGG19, there is a 34-layer plain network to which shortcut connections, or skip connections, have been introduced [15,16]. The design is then transformed into the residual network by use of these skip connections, or residual blocks.

3.2.3 MobileNet

MobileNet is a mobile-optimized computer vision model developed by TensorFlow, designed for use in mobile applications. In the context of weather identification, this lightweight model can be extremely useful for real-time weather classification tasks, especially in mobile devices. [17].

By deploying MobileNet for tasks such as identifying cloud patterns, detecting weather anomalies, or classifying weather conditions based on visual data, mobile applications can provide timely weather updates directly to users. Due to the open-source development, it enables developers to adapt and train the model for specific weather-related use cases, such as forecasting, weather pattern recognition, or even detecting severe weather conditions in real time.

3.2.4 VGG19

The VGG19 model is a subset of the VGG model that adds one extra layer to the base model, the architecture contains a total of 19 layers (16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and final 1 SoftMax layer). In addition to VGG11 and VGG16, there are further VGG variations. Since the models were made publicly accessible by the authors, they may be utilized as-is or with minor adjustments for other comparable tasks, making this mode a useful classification architecture for a wide variety of datasets. Facial recognition is another area where transfer learning may be used. Weights are freely accessible with other frameworks like Keras, allowing for arbitrary customization and application. The VGG-19 network's aesthetic and content loss. Fig. 3 shows the VGG16 Model Architecture.

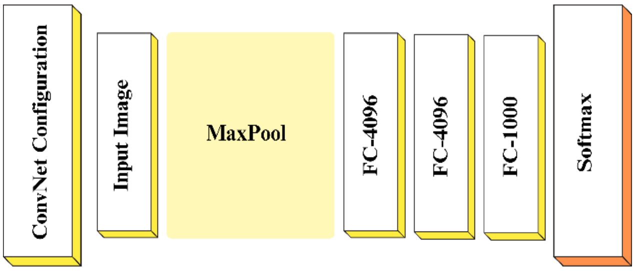

Fig. 3. VGG16 Model Architecture

3.2.5 Xception

The model takes the concepts behind the base Inception model and makes them more extreme. In the original Inception model, the input was compressed using 1×1 convolutions, followed by the application of different types of filters to the depth areas of each input vector. Xception reverses this approach. Instead of compressing the input first, it applies filters to each of the depth maps individually. Afterward, 1×1 convolution is applied across the depth to compress the input space [18]. This design improves the model's efficiency and performance by better using depth wise separable convolutions. Fig. 4 shows the Xception Model Architecture.

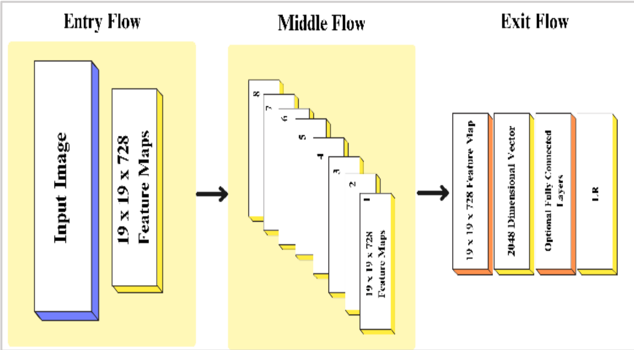

Fig. 4. Xception Model Architecture

The data will first travel through the entry flow, then it will go through the middle flow (during which it will repeat itself eight times), and lastly it will proceed through the exit flow. TensorFlow, a framework developed by Google, was used to create Xception, and each of its 60 NVIDIA K80 GPUs was used to train the model.

3.2.6 InceptionResNetV2

ResNet and Inception offer advancements in image recognition tasks, with exceptional performance with low computational cost. The Inception architecture, known for its efficient design, further improved by integrating residual connections, leading to the development of Inception-ResNet [17].

Over a million pictures from the ImageNet collection are used to train the convolutional neural network known as Inception-ResNet-v2. The network has 164 layers [18,19], and it can categorize pictures into 1000 different item categories, including things like surface, and objects. As a direct consequence of this, the network has acquired the ability to learn rich feature representations for a diverse set of picture types. With input images sized at 299 x 299 pixels, the network outputs a list of predicted class probabilities, enabling it to perform large-scale image classification efficiently [19].

3.2.7 VGG16

To improve accuracy of classification performance, the model's developers are increasing depth through the use of tiny (3x3) convolution filters. Where architecture design allowed them to add more layers, resulting in a network with 16 to 19 weight layers. As a result, VGG16 has approximately 138 million trainable parameters, enabling it to learn complex feature representations for image classification tasks. Fig. 5 shows the VGG16 Model Working Flow.

Fig. 5. VGG16 Model Working Flow

VGG16 is a highly effective model in the field of image classification, achieving an impressive success rate of 92.7% when tasked with categorizing 1,000 images into 1,000 distinct categories in weather identifications. The image recognition task is partly due to its ease of use with transfer learning. Fig. 6 shows the accuracy of the model. Using 16 weighted layers in the network, which are the layers with trainable parameters. In total, VGG16 consists of 21 layers, including 13 convolutional layers, 5 Max Pooling layers, and 3 fully connected (dense) layers. Despite having 21 layers, only 16 are weight layers, focusing on those with trainable parameters.

3.2.8 Resnet101

This model utilizes the concept of residual blocks to solve the problem of removing or exploding gradients. In terms, instead of forcing the network to fit the initial mapping, H(x), it allows the network to adjust and focus on learning the difference (residual) between the input and output, making training and efficient architectural design.

Fig. 7 shows that the error rate for a 56-layer CNN is greater than that of a 20-layer CNN on both the training and testing datasets. The authors conducted more research on the mistake rate and concluded that the vanishing/exploding gradient is to blame.

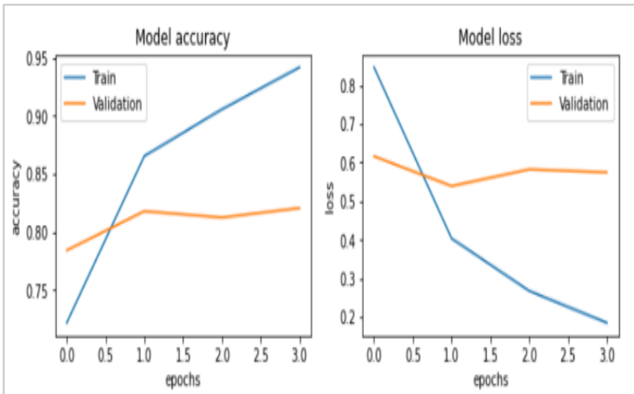

Fig. 6. Model Accuracy Report

The ResNet architecture (including the residual blocks) may be designed from scratch using the TensorFlow and Keras application programming interfaces. Different ResNet designs are shown in Fig. 8.

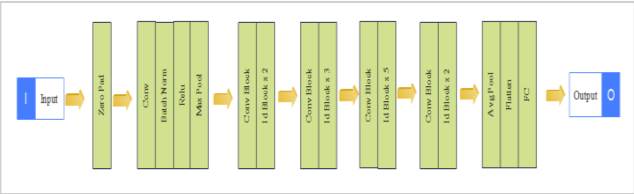

Fig. 7. Model Architecture

Fig. 7, illustrates input images are processed through residual blocks, with multiple layers forming each block. To optimize the model, the study removes the fully connected (FC) layer that originally contained one thousand different object categories. Depending on the number of target classes, a new FC layer to accommodate the specific classification requirements.

3.2.9 DenseNet201 and DenseNet169

As can be seen in the graphic below, the forward pass of a typical convolutional neural network is rather uncomplicated, with the input image being fed into the network and the output being a predicted label.

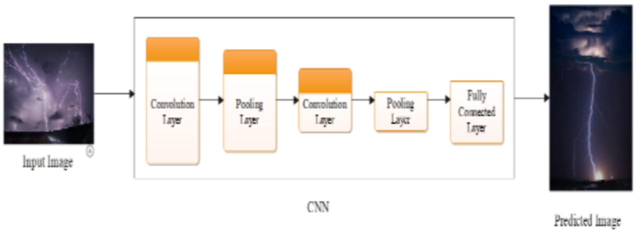

Fig. 8. DenseNet Model Architecture

In DenseNet, each layer processes the feature maps of the preceding layers, except for the first convolutional layer, which directly processes the input image. For an L-layer network, there are L direct connections, one between each consecutive layer. Total of L(L+1)/2 direct connections between layers. In DenseNet, each layer's feature maps serve as inputs for the layer above it, while the feature maps of the upper layer are used as inputs for the one below. Downsampling can occur outside the dense blocks through convolution and pooling operations. Within the dense block, feature maps are concatenated, ensuring that the sizes of the feature maps are consistent across all layers.

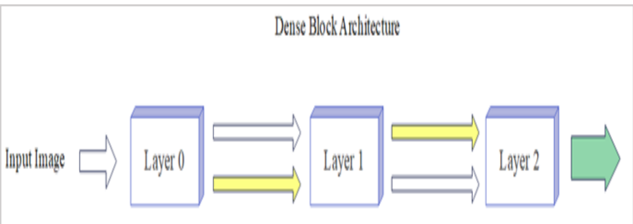

Fig. 9. DenseNet Model Block Architecture

4. Result Analysis

There is a wide variety of accessible frameworks for deep learning today. We have used a batch size of 32 and 100 iterations, where the learning algorithm will work through the entire training dataset. The reason behind choosing Keras instead of any other system is that it has garnered significant support from the scholarly community as well as inside the industry. Keras, when combined with the second version of TensorFlow, has more users than any other deep learning solution. Both Keras and TensorFlow 2 are quite popular among academics, as shown by the fact that they rank first in terms of the number of references they have received in scholarly articles indexed by Google Scholar.

Deep learning models that come pre-trained with Keras applications are known as Keras applications. These models are useful for making predictions, extracting features, and fine-tuning other models. According to the documentation that they have provided, an overview of the prediction table allows us to quickly compare the outcome with our own observations. The top 10 implementations correspond to the ImageNet validation dataset performance of the model.

The results of prior study, as shown in Table 2, are obtained through the Keras application. Their results and our most recent discoveries, which were predicated on their findings, are quite different in a few significant ways. We are getting up to 80% model accuracy from that model architecture, which is impressive considering the wide variety of deep learning models that we can implement (EfficientNetB7, Resnet, MobileNet, VGG19, Xception, Inception ResNetV2, VGG16, ResNet101, DenseNet201, and DenseNet169). We can achieve an efficiency of 80% across all the models using only five of them.

Table 2

Model accuracy by Keras application

Model Names	Top A-10 Accuracy	Parameters
EfficientNetB7	84.3%	66.7M
ResNet	74.9%	25.6M
MobileNet	70.4%	4.3M
VGG19	71.3%	143.7M
Xception	79.0%	22.9M
InceptionResNetV2	80.3%	55.9M
VGG16	71.3%	138.4M
ResNet101	76.4%	44.7M
DenseNet201	77.3%	20.2M
DenseNet169	76.2%	14.3M

The results of our implementation are summarized in Table 3, both in terms of test model accuracy and test model loss. Out of all the models, we have achieved a model accuracy of up to 80% in ResNet, MobileNet, Resnet101, and DenseNet201, DenseNet169. These models have a model test loss rate of greater than 0.5 less than 1, making them more suited for forecasting the weather using image recognition. Fig. 10 demonstrates the summary of the implemented models.

Table 3

Implementation findings

Model Names	Top 10 Accuracy	Parameters
EfficientNetB7	76.47%	0.67648
ResNet	80.63%	0.68640
MobileNet	81.94%	0.72586
VGG19	79.24%	0.78651
Xception	76.40%	0.87771
InceptionResNetV2	71.81%	1.16593
VGG16	79.10%	0.81799
ResNet101	83.39%	0.56006
DenseNet201	82.96%	0.54366
DenseNet169	82.96%	0.60020

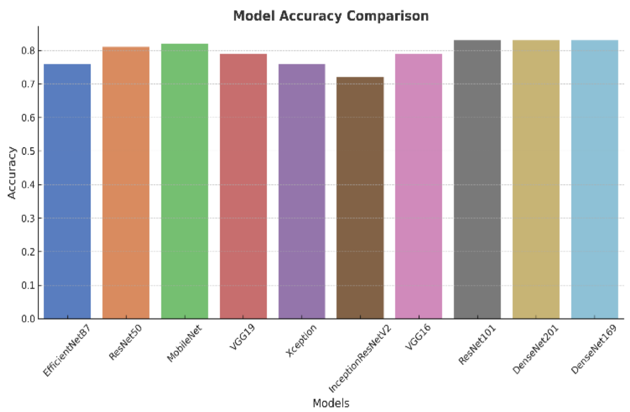

Fig. 10. Implemented Model Summary

4.1 Comparative Analysis

Our work utilizes multiple pre-trained deep learning models for weather forecasting, achieving competitive performance compared to previous studies while introducing novel aspects. This study is novel because it integrates multiple pre-trained models that provide versatility and scalability, allowing for more efficient training and better generalization across diverse climatic conditions. Unlike previous studies that often relied on custom-built or singular models, our approach reduces training time, resource usage, and offers a broader adaptability, making it a more comprehensive solution for weather forecasting.

Salman et al. [1] achieved an R^2 value of 84.8% for rainfall prediction, while our models, such as ResNet101 and DenseNet201, obtained accuracies of up to 83.39%, demonstrating effective generalization. Compared to the lower accuracy models in other studies, our work shows greater versatility and scalability [15]. This combination of efficiency, flexibility, and scalability sets our work apart from previous studies, offering a novel solution to the challenges of weather forecasting [18].

5. Conclusion

This study evaluated ten deep learning models, including EfficientNetB7, ResNet, MobileNet, VGG-19, Xception, InceptionResNetV2, VGG-16, ResNet101, DenseNet201, and DenseNet169, for weather forecasting using image data. Among these, ResNet101, DenseNet201, and DenseNet169 achieved the highest accuracy, demonstrating the effectiveness of deep learning models in automating weather prediction. The findings highlight the potential of these models to improve weather forecasting precision through advanced image recognition techniques.

However, several constraints were encountered in this work. The dataset, while diverse, was limited in size and exhibited imbalances across some weather categories, which may have impacted the generalizability of the models. Additionally, computational demands restricted the exploration of larger architectures and more extensive hyperparameter tuning. The study also focused solely on image classification, without integrating meteorological attributes such as temperature, humidity, and atmospheric pressure, which are critical for real-world applications. Future research should address these limitations by utilizing larger datasets, incorporating image segmentation methods like U-Net, and integrating additional meteorological variables to enhance model performance and practical applicability.

6. References

[1] Salman, A. G., Kanigoro, B., & Heryadi, Y., "Weather forecasting using deep learning techniques", 2015 international conference on advanced computer science and information systems (ICACSIS), 2015, pp. 281-285.

[2] Fang, W., Xue, Q., Shen, L., & Sheng, V. S., "Survey on the application of deep learning in extreme weather prediction", Atmosphere, Vol. 12, No. 6, 2021, pp. 661.

[3] Xiao, H., Zhang, F., Shen, Z., Wu, K., & Zhang, J., "Classification of weather phenomenon from images by using deep convolutional neural network", Earth and Space Science, Vol. 8, No. 5, 2021.

[4] Arcucci, R., Zhu, J., Hu, S., & Guo, Y. K., "Deep data assimilation: integrating deep learning with data assimilation", Applied Sciences, Vol. 11, NO. 3, 2021, pp. 1114.

[5] Rodrigues, E. R., Oliveira, I., Cunha, R., & Netto, M., “DeepDownscale: a deep learning strategy for high-resolution weather forecast”, 2018 IEEE 14th International Conference on e-Science (e-Science), 2018, pp. 415-422.

[6] Raza, M. Q., Mithulananthan, N., Li, J., Lee, K. Y., & Gooi, H. B., “An ensemble framework for day-ahead forecast of PV output power in smart grids”, IEEE Transactions on Industrial Informatics, Vol. 15, No. 8, 2018, pp. 4624-4634.

[7] He, K., Zhang, X., Ren, S., & Sun, J., “Deep residual learning for image recognition”, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[8] R. Abbas and N. Gu, "Improving Deep Learning-Based Image Super-Resolution with Residual Learning and Perceptual Loss Using SRGAN Model", Soft Computing, vol. 27, no. 4, pp. 1201-1215, 2023.

[9] X. Zhang, K. He, J. Sun, and S. Ren, "Deep Residual Learning for Image Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 2, pp. 1123-1135, Feb. 2023.

[10] W. Fang, Q. Xue, and V. S. Sheng, "Residual Attention Network for Image Classification", IEEE Access, vol. 11, pp. 2109-2119, 2023.

[11] S. Mehta, A. Rastegar, and P. Rai, "Residual Dense Networks for Image Restoration and Enhancement", Proceedings of the IEEE International Conference on Image Processing (ICIP), 2023, pp. 1503-1511.

[12] H. Xiao, F. Zhang, and J. Shen, "Enhancing Image Recognition with Deformable Residual Networks", Computer Vision and Image Understanding, vol. 223, pp. 45-58, Jan. 2023.

[13] T. Chen and Z. Liu, "Residual Neural Networks with Multi-scale Aggregations for Visual Recognition", Neural Networks, vol. 162, pp. 77-90, 2023.

[14] Y. Lin, C. Sun, and H. Lee, "Improving Object Detection with Cascaded Residual Learning Networks", Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 2301-2310.

[15] G. Huang, H. Fu, and A. G. Bors, "Masked Image Residual Learning for Scaling Deeper Vision Transformers", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1234-1245

[16] Z. Wang and J. Wang, "Deep Residual Networks for Cross-Domain Image Recognition", IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 3, pp. 456-469, Mar. 2023.

[17] A. Bansal, M. Jain, and D. Mehta, "Deep Residual Learning for Medical Image Analysis", IEEE J. Biomed. Health Inform., vol. 27, no. 5, pp. 1234-1245, May 2023.

[18] Q. Liu, Y. Chen, and H. Zhang, "Optimizing Residual Networks for Low-Latency Inference", IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 4, pp. 889-898, Apr. 2023.

[19] M. Alom and T. Li, "Advanced Residual Neural Architectures for Image-Based Weather Prediction", Appl. Soft Comput., vol. 135, pp. 110-122, Apr. 2023.

[20] L. Chen, K. He, and X. Sun, "Fusion of Residual Learning and Transformer Networks for Image Recognition", Pattern Recognit., vol. 139, pp. 203-214, 2023.