Jurnal Informatika: Jurnal pengembangan IT

Vol. 9, No. 2, (2024)
ISSN: 2477-5126, DOI:10.30591/jpit.v9i2.7539

Efficient Weather Classification Using DenseNet and EfficientNet

Mirza Alim Mutasodirin[1], Faiq Miftakhul Falakh[2]
[1] *Politeknik Harapan Bersama, Tegal, Jawa Tengah, Indonesia*
[2] *Independent Researcher, Tegal, Jawa Tengah, Indonesia*

Info Artikel

Riwayat Artikel:

Received 2024-08-25
Revised 2024-08-29
Accepted 2024-08-30

Corresponding Author:

Mirza Alim Mutasodirin

Email:

mirza.alim.m@poltektegal.ac.id

Abstract – *Classifying images of weather conditions using deep learning models is a challenging task due to the computational intensity and resource requirements. To deploy AI models on resource-constrained devices like smartphones and IoT devices, compact and computationally lightweight models are necessary. Efficient deep learning models for weather classification are essential to reduce energy consumption and costs, making AI more accessible and sustainable. To the best of our knowledge, there are limited studies comparing MobileNet, DenseNet, and EfficientNet as efficient models and did not report any hyperparameter optimization. Our study contributes by investigating efficient models with hyperparameter optimization. Firstly, we measured the inference speed of 14 models, namely MobileNet, MobileNetV2, MobileNetV3, EfficientNetB0, EfficientNetV2B0, NASNetMobile, DenseNet121, VGG16, Xception, InceptionV3, ResNet50, ResNet50V2, ConvNeXtTiny, and InceptionResNetV2. Then, the top-7 fast models, which are MobileNet, MobileNetV2, MobileNetV3, EfficientNetB0, EfficientNetV2B0, NASNetMobile, and DenseNet121, were benchmarked for their accuracy. The models were compared by a small dataset having four classes: cloudy, rain, shine, and sunrise. Batch size and learning rate for each model were optimized by grid search method. It turns out that DenseNet121 achieved the best and the most balanced validation and testing accuracy, 0.9821 and 0.9837, followed by EfficientNetB0 with 0.9821 and 0.9740 respectively. This study is important to find efficient models with optimal comparison.*

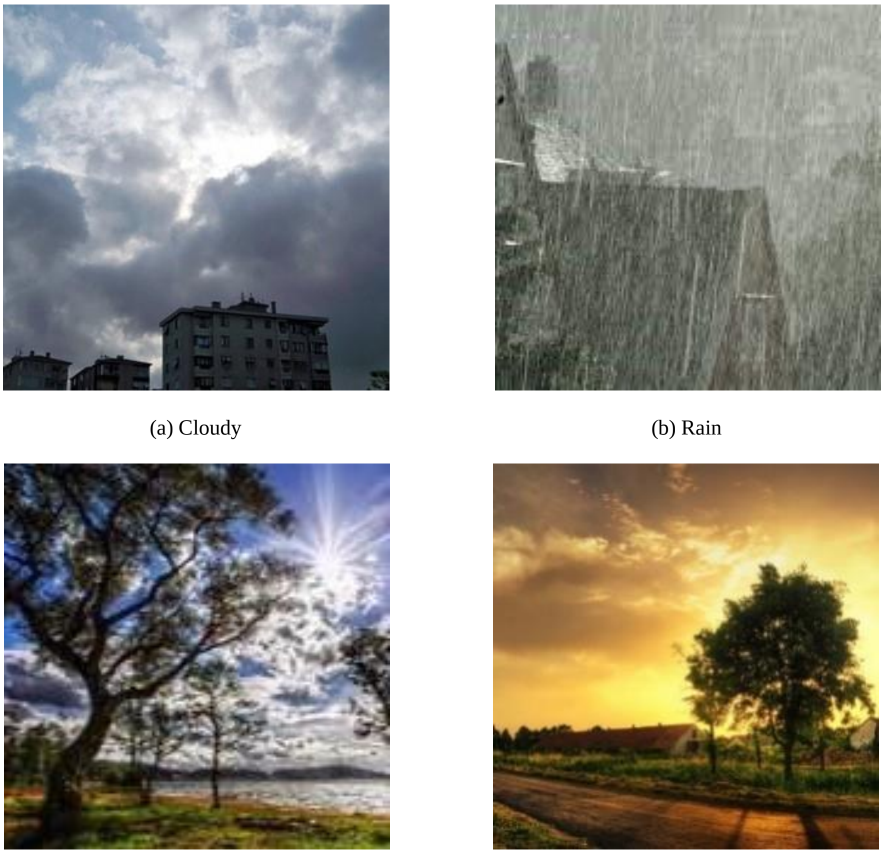
Keywords: *densenet; efficientnet; hyperparameter optimization; weather classification.*

Abstrak – *Mengklasifikasikan gambar kondisi cuaca menggunakan model deep learning merupakan tugas yang menantang karena intensitas komputasi dan kebutuhan sumber dayanya. Untuk menggunakan model AI pada perangkat dengan sumber daya terbatas seperti smartphone dan perangkat IoT, diperlukan model yang ringkas dan ringan secara komputasi. Model deep learning yang efisien untuk klasifikasi cuaca sangat penting untuk mengurangi konsumsi energi dan biaya, membuat AI lebih mudah diakses dan berkelanjutan. Sejauh pengetahuan kami, studi yang membandingkan MobileNet, DenseNet, dan EfficientNet sebagai model yang efisien sangat terbatas dan tidak melaporkan optimasi hiperparameter apa pun. Penelitian kami berkontribusi dengan menyelidiki model yang efisien dengan optimasi hyperparameter. Pertama, kami mengukur kecepatan inferensi dari 14 model, yaitu MobileNet, MobileNetV2, MobileNetV3, EfficientNetB0, EfficientNetV2B0, NASNetMobile, DenseNet121, VGG16, Xception, InceptionV3, ResNet50, ResNet50V2, ConvNeXtTiny, dan InceptionResNetV2. Kemudian, 7 model cepat teratas, yaitu MobileNet, MobileNetV2, MobileNetV3, EfficientNetB0, EfficientNetV2B0, NASNetMobile, dan DenseNet121, dibandingkan keakuratannya. Model-model tersebut dibandingkan dengan kumpulan data kecil yang terdiri dari empat kelas: mendung, hujan, cerah, dan matahari terbit. Ukuran batch dan tingkat pembelajaran untuk setiap model dioptimalkan dengan metode pencarian grid. Ternyata DenseNet121 mencapai akurasi validasi dan pengujian yang terbaik dan paling seimbang, yaitu 0.9821 dan 0.9837, diikuti oleh EfficientNetB0 dengan 0.9821 dan 0.9740. Penelitian ini penting untuk menemukan model yang efisien dengan perbandingan yang optimal.*

Kata Kunci*: densenet; efficientnet; klasifikasi cuaca; optimisasi hyperparameter.*

I. INTRODUCTION

Classifying weather automatically using camera sensors is a challenging task. It categorizes images of weather conditions into predefined categories of weather phenomena. The task is vital for various applications, such as automatic weather monitoring and reporting [1], autonomous vehicles [2], and agricultural monitoring [3]. Key challenges include handling diverse weather phenomena, addressing imbalanced or small datasets, and ensuring robust performance under varying lighting and image quality conditions. Diverse weather phenomena are not always in terms of the number of categories, but also the transition between categories. For example, in Indonesia having only rainy and summer season, the transition between clear and cloudy and the transition between cloudy and rainy could be subtle. The various types of cloud and how heavy the rain also create the need for a greater variety of data. Reducing the image size smaller or the quality lower could be more efficient, but, in the other hand, the image is getting unclear. As one of the advanced solutions, CNN-based deep learning models are often employed to classify the images with high accuracy [4], [5], [6], [7], [8].



(a) Cloudy (b) Rain

(c) Shine (d) Sunrise

Figure 1. Example of Each Class in Dataset

Due to the computational intensity and resource requirements, implementing deep learning models could be costly for many usage contexts. Deploying AI models on resource-constrained devices such as smartphones and IoT devices demands models that are compact and computationally lightweight [9]. The demand for real-time processing necessitates models to operate swiftly with minimal latency in minimal devices. Efficiency is a critical need in the applications. Efficient models can reduce the energy consumption and costs associated with running deep learning tasks, making them more accessible and sustainable. Additionally, the environmental impact of energy consumption is also an issue [10], [11]. Therefore, searching efficient deep learning models for weather classification is essential to unlock the full potential of AI.

Several small models for image classification are available, such as MobileNet [9], EfficientNet [12], and DenseNet [13]. The MobileNet is designed for mobile and embedded applications. The key idea is to reduce the computational complexity and resource requirements of CNN while maintaining the accuracy. MobileNet has three versions, namely MobileNet [9], MobileNetV2 [14], and MobileNetV3 [15]. The DenseNet leverages dense connectivity to enhance feature reuse, improve gradient flow, and reduce parameters. The concept of EfficientNet is compound scaling method, which scales all dimensions of depth, width, and resolution using a simple but effective compound coefficient. The approach allows EfficientNet to achieve an optimal balance between accuracy and resource consumption. EfficientNet has two versions, namely EfficientNet [12] and EfficientNetV2 [16].

There have been many studies on weather classification using MobileNet or DenseNet family. Kukreja et al. [17] compared MobileNetV2 [14] with simple CNN. Młodzianowski [18] and Revathi et al. [19] compared the performance of MobileNetV2, InceptionV3 [20], and ResNet50 [21]. Raj and Pillai [22] compared MobileNetV2 with VGG19 [23], InceptionResNetV2 [24], and InceptionV3. Yu et al. [25] compared MobileNet with ShuffleNet [26], VGG, ResNet, and GSLSNet [25]. Wang and Li [27] and Challa et al. [28] compared DenseNet with ResNet. Albelwi [29] compared DenseNet121 with VGG16, ResNet18, Inception [30], and AlexNet [31]. Sun et al. [1] compared DenseNet161 with VGG19, InceptionV3, ResNet101, and DS-CNN.

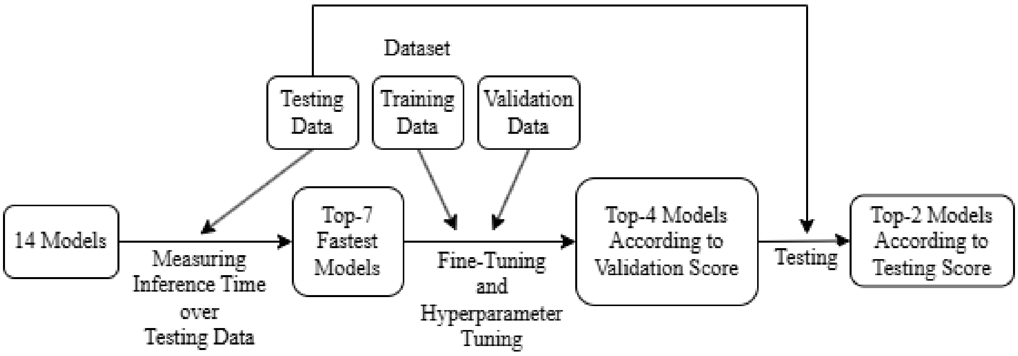


Figure 2. Research Methodology

Some other studies compare MobileNet and DenseNet families. Kalkan et al. [32] compared MobileNetV2 and DenseNet201 with VGG16, ResNet152 V2 [33]. Gupta and Goel [34] compared MobileNet and DenseNet161 with VGG19, ResNet50, and InceptionV3. Nam [4] compared MobileNet and DenseNet201 with VGG19, ResNet50, ResNet152, RegNetX002 [35], Xception [36], InceptionV3, and ConvNeXtBase [37]. Mashudi et al. [3] compared MobileNet and DenseNet201 with Xception and InceptionResNetV2. Nagib et al. [38] compared MobileNetV2, DenseNet121, DenseNet169, DenseNet201 with InceptionV3 and Xception.

EfficientNet family is also tested in some studies. Zhang et al. [39] and Pillai et al. [7] only used EfficientNet. Rajabi et al. [40] compared EfficientNet and ResNet. Sadia et al. [41] compared EfficientNet with ResNet18, ResNet50, and ResNeXt50 [42]. Al Haija et al. [2] compared EfficientNet with SqueezeNet [43] and ResNet50. Samo et al. [8] compared EfficientNetB7 with VGG19, GoogleNet [30], ResNet152, and InceptionV3. Yuan et al. [5] compared EfficientNetV2 with VGG16, ABNet, VisNet [44], and Swin Transformer V2 [45]. Yang et al. [46] and Rani et al. [6] combined EfficientNet and attention mechanism.

As far as our search goes, there are only two studies found comparing MobileNet, DenseNet, and EfficientNet. Afxentiou and Vladimirova [47] compared MobileNetV2, MobileNetV3, DenseNet121, and EfficientNetB0. Phim and Tran [48] compared MobileNetV2, EfficientNet, and DenseNet with Vision Transformer [49]. Phim et al. report that the learning rate was set identically for all models, which is empirically not fair because each model may need different hyperparameter settings to meet the best performance. Both studies do not report any hyperparameter optimization to find optimal settings for each model.

Our study contributes by investigating efficient models with hyperparameter optimization. The purpose is to find the optimal efficient models in a fair comparison that each model meets their optimal performance in appropriate hyperparameter setting. Firstly, we measured the inference speed of fourteen models, namely MobileNet, MobileNetV2, MobileNetV3, EfficientNetB0, EfficientNetV2B0, NASNetMobile [50], DenseNet121, VGG16, Xception, InceptionV3, ResNet50, ResNet50V2, ConvNeXtTiny, InceptionResNetV2. Seven models with the number of parameters under 10M, which are MobileNet, MobileNetV2, MobileNetV3, EfficientNetB0, EfficientNetV2B0, NASNetMobile, and DenseNet121, were then compared for their accuracy performance as they are the fastest of all. The models are compared by a small dataset [51] having four classes: cloudy, rain, shine, and sunrise; as depicted in Figure 1. Batch size and learning rate for each model were optimized by grid search. This study is important to find efficient models with optimal comparison.

II. METHODOLOGY

There are three steps in this work: inference time measurement, fine-tuning and hyperparameter optimization, and testing. The methodology flow is depicted in Figure 2. This section contains the explanation of each step sequentially, starting with an explanation of the dataset used, the preprocessing process, and the deep learning models to compare.

A. Dataset and Preprocessing

The dataset used in this work comes from the work of Oluwafemi and Zenghui [51], and consists of 1125 images. The data were collected from the internet, from various sources, such as Google Images, Yahoo, Getty Images, Flickr, etc. It has 4 classes, namely cloudy, rain, shine, and sunrise. The data were labelled manually. Figure 1 shows examples of each class. Each has 300, 215, 253, and 357 images respectively. The dataset comes with no train-val-test split. Thus, in this work, we split them 60%, 20%, and 20% for training, validation, and testing data respectively. Table 1 depicts the split in more detail.

TABLE 1
DATA SPLIT

Split	Cloudy	Rain	Shine	Sunrise	%
train	180	129	153	215	60%
val	60	43	50	71	20%
test	60	43	50	71	20%
Total	300	215	253	357	100%

TABLE 2
MODELS TO MEASURE THE INFERENCE TIME, SORTED BY THE NUMBER OF PARAMETERS

No.	Model	Param.	No.	Model	Param.
1	MobileNetV2	2.2M	**8**	VGG16	14.7M
2	MobileNetV3	2.9M	**9**	Xception	20.8M
3	MobileNet	3.2M	**10**	InceptionV3	21.7M
4	EfficientNetB0	4.0M	**11**	ResNet50	23.5M
5	NASNetMobile	4.2M	**12**	ResNet50V2	23.5M
6	EfficientNetV2B0	5.8M	**13**	ConvNeXtTiny	27.8M
7	DenseNet121	6.9M	**14**	InceptionResNetV2	54.2M

The dataset is intentionally not augmented to make it more challenging, to test which models are better for small data. All images are cropped in square shape and resized to be 244x244 pixels. Small size makes the computation efficient and faster. The images remain in RGB format since colour feature is one of the most important features beside shape feature. The RGB scale is rescaled to be 0-1, instead of 0-255. No manual feature extraction was done in this preprocessing and we let the models understand the features automatically to test how well they learn.

B. Deep Learning Models

The deep learning models compared in this study comes from an official implementation of TensorFlow Keras library. ImageNet-pretrained weights for all models are also available. We choose 14 models having number of parameters under 60 million. The model names are listed in Table 2. By the number of parameters, there are 5 models under 5 million parameters, 2 models between 5-10 million parameters, 1 model between 10-20 million parameters, 5 models between 20-40 million parameters, and 1 model between 40-60 million parameters. Smaller model is not necessarily faster; depending on the complexity of the model. Thus, empirical experiment is necessary to assure the speed.

Instead of starting from random initialization weights, the models load ImageNet-pretrained weights with input shape 244x244x3, without the classification layer. On top of the base model, we added a layer of 2D global average pooling, a dropout layer with probability of 0.2, and a classification layer, with softmax activation function, having 4 nodes corresponding to the number of classes in this study. No layers were frozen. All layers are trainable.

C. Inference Time Measurement

One of the metrics to measure model efficiency is inference time. It is how long it takes for a model to predict data regardless of the correctness of the answer. Measuring inference time using a single image is insignificant. We used all testing data at once to measure the time. The measurement metric is in seconds. The difference can be clearly seen. We measured 14 models as listed in Table 2. All models were pretrained using ImageNet dataset. In this step we did not train any model. The model weights are available online and downloadable. The inference computation ran on Intel Xeon CPU. By the measurement, top-7 models were taken to compare in the next step.

D. Fine-Tuning and Hyperparameter Optimization

The selected models were fine-tuned in 10 epochs using training data and validated using validation data. All models used ImageNet-pretrained weights. The learning optimizer used in our fine-tuning process was Adam. To evaluate the performance and determine the best model, we used accuracy as the evaluation metric. Accuracy is simply a ratio between all correct predictions and all predictions. The weight checkpoint saved was determined by the best validation accuracy.

In order to achieve optimal performance and fair comparison, we conducted hyperparameter optimization (HPO). We searched for the best batch size (BS) and learning rate (LR) in 10-epoch training. The HPO method used was simple grid search in the range of BS from 1 to 64 and LR from 1e-3 to 1e-5, as detailed in Table 3. The BS is in the power of 2. We lower the learning rate to half, from 1e-3 to 5e-4, from 5e-4 to 2e-4, from 2e-4 to 1e-4, and so on until 1e-5. By finding the optimal setting, the models can be compared in their optimal performance. We then picked 4 models with the highest validation accuracy after hyperparameter tuning.

TABLE 3
HYPERPARAMETER SETTINGS

Learning Optimizer	Adam
Number of Epoch	10
Batch Size	1, 2, 4, 8, 16, 32, 64
Learning Rate	1e-3, 5e-4, 2e-4, 1e-4, 5e-5, 2e-5, 1e-5

E. Testing

The best weight of a model is judged by validation accuracy. The model reloads the weight from its checkpoint. The model is then used to deal with data in the wild or, in this benchmarking case, testing data. Normally, the testing accuracy is lower than the validation accuracy. Nevertheless, as validation data is also unknown data, smaller difference between validation accuracy and testing accuracy means the training result is better and more representative.

III. RESULT DISCUSSION

The result of inference time measurement is depicted in Table 4. It says that the MobileNet family is the fastest, followed by EfficientNet, EfficientNetV2, NASNetMobile, and DenseNet121. VGG16, ConvNeXtTiny, and InceptionResNetV2 seem slow and inefficient. VGG16, with 14.7 million parameters, takes three times longer than ResNet with 23.5 million parameters. While ConvNeXtTiny with 27.8 million parameters takes nine times longer than ResNet. The EfficientNet family is twice as slow as the MobileNet family. While NASNetMobile and DenseNet121 are about three times slower. However, efficiency is not the only parameter to consider. Thus, we take the top-7 fast models to benchmark the accuracy.

Table 5 depicts the best validation accuracy of each model after the HPO process. The best validation accuracy was achieved by DenseNet121 followed by EfficientNetB0 v1, v2, and MobileNet v1. The testing result seems close to the validation result, and the most balanced is DenseNet121. EfficientNetB0 can also be an option if a faster model is needed, at the cost of a bit of accuracy. NASNetMobile, MobileNetV2, and MobileNetV3 do not perform well and the testing accuracy is under 95%.

TABLE 4
INFERENCE TIME RESULT OF IMAGENET PRETRAINED MODELS AGAINST WEATHER TESTING DATA

No.	Model	Inf. Time	No.	Model	Inf. Time
1	MobileNet	12.5 seconds	**8**	ResNet50	40.0 seconds
2	MobileNetV2	12.7 seconds	**9**	ResNet50V2	40.4 seconds
3	MobileNetV3	12.8 seconds	**10**	InceptionV3	41.0 seconds
4	EfficientNetV2B0	20.5 seconds	**11**	Xception	47.6 seconds
5	EfficientNetB0	24.9 seconds	**12**	InceptionResNetV2	78.5 seconds
6	NASNetMobile	33.1 seconds	**13**	VGG16	122.0 seconds
7	DenseNet121	35.2 seconds	**14**	ConvNeXtTiny	381.9 seconds

TABLE 5
THE BEST VALIDATION ACCURACY AFTER HYPERPARAMETER OPTIMIZATION

No.	Model	Max. Epoch	BS	LR	Val. Acc.	Test. Acc.
1	DenseNet121	10	16	2e-4	**0.9821**	**0.9837**
2	EfficientNetB0	10	4	1e-4	0.9821	0.9740
3	EfficientNetV2B0	10	16	2e-4	0.9777	0.9625
4	MobileNet	10	16	2e-4	0.9777	0.9603
5	MobileNetV2	10	8	1e-4	0.9509	0.9047
6	NASNetMobile	10	2	2e-4	0.9196	0.8795
7	MobileNetV3	10	2	1-e5	0.5000	0.2992

IV. CONCLUSION

This study contributes to benchmarking efficient models in an optimal comparison to find efficient yet effective models. We measured the inference time of 14 models and took the top-7 fast models. The selected models were then fine-tuned and hyperparameter-optimized to meet their optimal performance. The top-4 best models by validation accuracy were then tested to find the most balanced accuracy between validation and testing. It turns out that DenseNet121 is the best model followed by EfficientNetB0. Both models can be an efficient solution.

REFERENCES

[1] Z. Sun, P. Wang, Y. Jin, J. Wang, and L. Wang, "A Practical Weather Detection Method Built in the Surveillance System Currently Used to Monitor the Large-Scale Freeway in China," *IEEE Access*, vol. 8, pp. 112357–112367, 2020, doi: 10.1109/ACCESS.2020.3002959.

[2] Q. A. Al-Haija, M. Gharaibeh, and A. Odeh, "Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning," *AI*, vol. 3, no. 2, pp. 303–317, 2022, doi: 10.3390/ai3020019.

[3] N. A. Mashudi, N. Ahmad, S. M. Sam, N. Mohamed, and R. Ahmad, "Deep Learning Approaches for Weather Image Recognition in Agriculture," in *2022 IEEE Symposium on Future Telecommunication Technologies (SOFTT)*, 2022, pp. 72–77. doi: 10.1109/SOFTT56880.2022.10010161.

[4] T. Q. Nam, "Evaluation of Stacked Ensemble Model on Weather Image Recognition," *Procedia Comput Sci*, vol. 234, pp. 1664–1671, 2024, doi: https://doi.org/10.1016/j.procs.2024.03.171.

[5] C. Yuan *et al.*, "Enhancing Road Safety: Real-Time Classification of Low Visibility Foggy Weather Using ABNet Deep-Learning Model," *J Transp Eng A Syst*, vol. 150, no. 10, p. 4024060, 2024, doi: 10.1061/JTEPBS.TEENG-8492.

[6] R. U. Rani, J. Kakarla, and B. Sundar, "Weather image classification using EfficientNet and Dual Attention Block," in *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, 2023, pp. 1–4. doi: 10.1109/ICSTSN57873.2023.10151564.

[7] R. Pillai, N. Sharma, and R. Gupta, "Fine-Tuned EfficientNetB4 Transfer Learning Model for Weather Classification," in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, 2023, pp. 1–6. doi: 10.1109/ASIANCON58793.2023.10270698.

[8] M. Samo, J. M. Mafeni Mase, and G. Figueredo, "Deep Learning with Attention Mechanisms for Road Weather Detection," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020798.

[9] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017.

[10] D. Patterson *et al.*, "Carbon Emissions and Large Neural Network Training," Apr. 2021.

[11] S. Pei, J. Luo, S. Liang, H. Ding, X. Ye, and M. Chen, "Carbon Emissions Reduction of Neural Network by Discrete Rank Pruning," *CCF Transactions on High Performance Computing*, vol. 5, no. 3, pp. 334–346, 2023, doi: 10.1007/s42514-023-00144-0.

[12] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, Aug. 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[15] A. Howard *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324. doi: 10.1109/ICCV.2019.00140.

[16] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, Aug. 2021, pp. 10096–10106. [Online]. Available: https://proceedings.mlr.press/v139/tan21a.html

[17] V. Kukreja, V. Solanki, A. Baliyan, and V. Jain, "WeatherNet: Transfer Learning-based Weather Recognition Model," in *2022 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2022, pp. 1–5. doi: 10.1109/ESCI53509.2022.9758183.

[18] P. Młodzianowski, "Weather Classification with Transfer Learning - InceptionV3, MobileNetV2 and ResNet50," in *Digital Interaction and Machine Intelligence*, J. and K. W. and O. J. W. and R. A. and S. M. Biele Cezary and Kacprzyk, Ed., Cham: Springer International Publishing, 2022, pp. 3–11.

[19] P. Revathi, S. Bhavani, G. Karthika Priya Dharshini, G. Rajaram, S. Siddhan, and R. K. Dwibedi, "Weather Prediction(analysis) using Soft kind of Computing techniques," in *2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2024, pp. 1337–1340. doi: 10.1109/ICACITE60783.2024.10616994.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[22] A. S. Raj and A. S. Pillai, "Performance Comparison of Convolution Neural Networks on Weather Dataset," in *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2022, pp. 1505–1510. doi: 10.1109/ICIRCA54612.2022.9985500.

[23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014.

[24] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.11231.

[25] T. Yu, Q. Kuang, J. Hu, J. Zheng, and X. Li, "Global-Similarity Local-Salience Network for Traffic Weather Recognition," *IEEE Access*, vol. 9, pp. 4607–4615, 2021, doi: 10.1109/ACCESS.2020.3048116.

[26] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856. doi: 10.1109/CVPR.2018.00716.

[27] Y. Wang and Y. Li, "Research on Multi-class Weather Classification Algorithm Based on Multi-model Fusion," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020, pp. 2251–2255. doi: 10.1109/ITNEC48623.2020.9084786.

[28] Sri Venkata Divya Madhuri Challa and Hemendra Kumar, "Weather Categorization Using Foreground Subtraction and Deep Transfer Learning," in *Innovations in Computer Science and Engineering: Proceedings of 7th ICICSE*, R. and B. R. and A. G. Saini Harvinder Singh and Sayal, Ed., Singapore: Springer Singapore, 2020, pp. 595–601. doi: 10.1007/978-981-15-2043-3_64.

[29] Saleh A Albelwi, "Deep Architecture based on DenseNet-121 Model for Weather Image Recognition," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, 2022.

[30] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[32] M. Kalkan *et al.*, "Cloudy/clear weather classification using deep learning techniques with cloud images," *Computers and Electrical Engineering*, vol. 102, p. 108271, 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.108271.

[33] X. and R. S. and S. J. He Kaiming and Zhang, "Identity Mappings in Deep Residual Networks," in *Computer Vision – ECCV 2016*, J. and S. N. and W. M. Leibe Bastian and Matas, Ed., Cham: Springer International Publishing, 2016, pp. 630–645.

[34] A. Gupta and S. Goel, "Deep Learning Empowered Weather Image Classification for Accurate Analysis," in *2023 IEEE 2nd International Conference on Industrial Electronics: Developments & Applications (ICIDeA)*, 2023, pp. 567–572. doi: 10.1109/ICIDeA59866.2023.10295216.

[35] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing Network Design Spaces," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10425–10433. doi: 10.1109/CVPR42600.2020.01044.

[36] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.

[37] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[38] A. E. Nagib, H. Mohamed, and A. Ashraf, "Comparative Analysis of Pre-Trained CNN Architectures for Multi-Class Weather Classification: A Study on Deep Learning Techniques," in *2023 Intelligent Methods, Systems, and Applications (IMSA)*, 2023, pp. 297–302. doi: 10.1109/IMSA58542.2023.10217616.

[39] Y. Zhang, J. Sun, M. Chen, Q. Wang, Y. Yuan, and R. Ma, "Multi-Weather Classification using Evolutionary Algorithm on EfficientNet," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2021, pp. 546–551. doi: 10.1109/PerComWorkshops51409.2021.9430939.

[40] F. Rajabi, N. Faraji, and M. Hashemi, "An intelligent method for detecting weather conditions using deep learning on a combined dataset," in *2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 2022, pp. 1–5. doi: 10.1109/ICSPIS56952.2022.10043995.

[41] R. T. Sadia, Md. A. Ahamed, and Md. A. Hossain, "Multiple Weather Scene Detection Utilizing the EfficientNet Family," in *2021 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, 2021, pp. 1–4. doi: 10.1109/IC4ME253898.2021.9768411.

[42] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5987–5995. doi: 10.1109/CVPR.2017.634.

[43] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," Feb. 2016.

[44] A. Palvanov and Y. I. Cho, "VisNet: Deep Convolutional Neural Networks for Forecasting Atmospheric Visibility," *Sensors*, vol. 19, no. 6, 2019, doi: 10.3390/s19061343.

[45] Z. Liu *et al.*, "Swin Transformer V2: Scaling Up Capacity and Resolution," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11999–12009. doi: 10.1109/CVPR52688.2022.01170.

[46] A. Yang, T. T. Toe, Z. Ran, and S. Xiao, "Multi-class Weather Classification using EfficientNet-B4 with Attention," in *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, 2022, pp. 89–92. doi: 10.1109/ICSESS54813.2022.9930176.

[47] V. Afxentiou and T. Vladimirova, "Multi-Label Weather Image Classification for Embedded Computing Platforms," in *2023 IEEE Smart World Congress (SWC)*, 2023, pp. 1–8. doi: 10.1109/SWC57546.2023.10449086.

[48] Huy Cong Phim and Nam Quy Tran, "EVALUATION OF VISION TRANSFORMER ON WEATHER IMAGE RECOGNITION," *Tra Vinh University Journal of Science*, vol. 13, no. 3, Sep. 2023.

[49] Alexey Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[50] B. Zoph, V. Vasudevan, J. Shlens, and Q. V Le, "Learning Transferable Architectures for Scalable Image Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710. doi: 10.1109/CVPR.2018.00907.

[51] A. Gbeminiyi Oluwafemi and W. Zenghui, "Multi-Class Weather Classification from Still Image Using Said Ensemble Method," in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, 2019, pp. 135–140. doi: 10.1109/RoboMech.2019.8704783.