International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 2
ISSN: 2088-5334

Improving Accuracy of Cloud Images Using DenseNet-VGG19

Gita Fadila Fitriana [a,*], Amalia Beladinna Arifa [b], Agi Prasetiadi [b], Faisal Dharma Adhinata [a], Nur Ghaniaviyanto Ramadhan [a]

[a] *Department of Software Engineering, Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Purwokerto, 53147, Indonesia*
[b] *Department of Informatics, Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Purwokerto, 53147, Indonesia*
Corresponding author: [*]gita@ittelkom-pwt.ac.id

***Abstract*— Weather classification has become a significant challenge due to the unpredictable nature of climate conditions. For farmers, predicting the start of the rainy season is very important. This is because it is related to the cost factor that must be incurred, and also, the waiting time for the harvest has an effect if the weather is not supportive. Farmers also have to prepare seeds for the start of their farming. Therefore, farmers who start nurseries early in the rainy season will miss significant planting time. Based on these problems, this study uses a convolutional neural network (CNN) for weather classification using cloud imagery. CNN is shown to classify different spectro-temporal features of sound and is thus suitable for cloud image classification. We collect cloud image data using secondary data. Our model used a layer based on the convolution CNN architecture with a pooling layer and a solid layer as the output layer. The cloud dataset used is 1230 data divided into five classes, namely cloudy, foggy, rain, shine, and sunrise, which we use to train our model in research for the feature extraction process using DenseNet and VGG19. We use two types of classification, namely fully connected and Global Average Pooling (GAP). Our model can achieve a classification accuracy of 90.8% DenseNet-Fully Connected from our training process. From our testing process, our model can reach 95.7% using DenseNet-Fully Connected classification accuracy. Thus, the CNN model proved very accurate in classifying cloud images.**

***Keywords*— Cloud images; CNN; DenseNet; VGG19; weather classification.**

Manuscript received 31 Aug. 2022; revised 31 Oct. 2022; accepted 6 Nov. 2022. Date of publication 30 Apr. 2023.

I. INTRODUCTION

Weather classification has become a significant challenge due to the unpredictable nature of climatic conditions [1]. Weather conditions do not only affect daily life. Several areas are involved, including the agricultural sector [2]. For farmers, estimating the beginning of the rainy season is essential, and farmers must prepare the seeds for the beginning of their agriculture. If farmers start seeding early in the rainy season, they will lose significant planting time [2]. In recent years, weather classification techniques have evolved. However, weather classification differs significantly from image-to-image applications and raises some unanswered concerns [3]. A neural network approach is used to skillfully predict the “weather” from a simplified climate model and mimic its climate [4]. The success of weather classification in analyzing data fed into shallow resolution determined that it was “fundamental” to produce deep learning-based weather classification [5], [6].

Weather classifications are usually completed with human eyesight. Recently, academics proposed that computer vision algorithms could be developed to reliably categorize weather conditions using images, saving human resources and expensive instruments (e.g., sensors). It is because inexpensive security cameras are widely available and would suffice to provide accurate weather classification [7]. In previous research [8], as far as we know, the accuracy of weather classification using Convolutional Neural Network (CNN) Architecture using an image dataset consisting of 10,000 images obtained an accuracy of 82.2% with sunny and cloudy classes. This study utilized CNN to complete the task of weather classification. CNN architecture is a neural network model that encapsulates nonlinear mapping between various spaces, such as future and label space [9]. Weather categorization can be simplified without engineering features due to CNN's straightforward and explicit architecture [10], [11]. Most CNN architectures are built to address tasks involving object detection and recognition [12].

Regarding this, there are a variety of weather classifications. Compared to information about objects like shape and texture, it is more sensitive to variables like illumination, the state of the sky, and shadows [13]. Before, the CNN method worked to get optimal results. An image must go through a pre-processing stage. Pre-processing techniques include image splitting, sky detection, and cloud edge detection. Pre-processing is done to improve accuracy in determining the weather [14].

Research by Ferdiana et al. [15] tried to use a convolutional neural network (CNN) to identify different cat sounds. CNN architecture is shown to classify different patterns of Spectro-temporal features of sound and thus is suitable for sound classification. Research by Gunawan et al. [16] presents a better approach for training models that can accurately predict the presence of animals based on their sounds with a limited data set. Currently, the in-depth study model dominates the advanced methods for audio classification tasks due to its predictive ability with the classification technique used, namely CNN.

Based on several previous studies, this study discusses weather classification from five types of cloud imagery, namely cloudy, foggy, rain, shine, and sunrise, which assist farmers in determining the planting calendar. In this study, the classification technique of the convolutional neural network was used.

II. MATERIALS AND METHODS

This research uses cloud datasets as weather classification with five classes. The original dataset is pre-processed before entering the feature extraction stage. The pre-processing process starts with resizing the dataset to uniform the size of all datasets. Then divide the data into training data and testing data. In this study, the training process uses two transfer learning models, namely VGG19 and DenseNet201. We evaluated the training results of these two models. We also compared the Global Average Polling (GAP) and Fully Connected layers in the classification process. The best results were used using the new cloud dataset for the testing phase. Fig. 1 shows the architecture of our proposal.

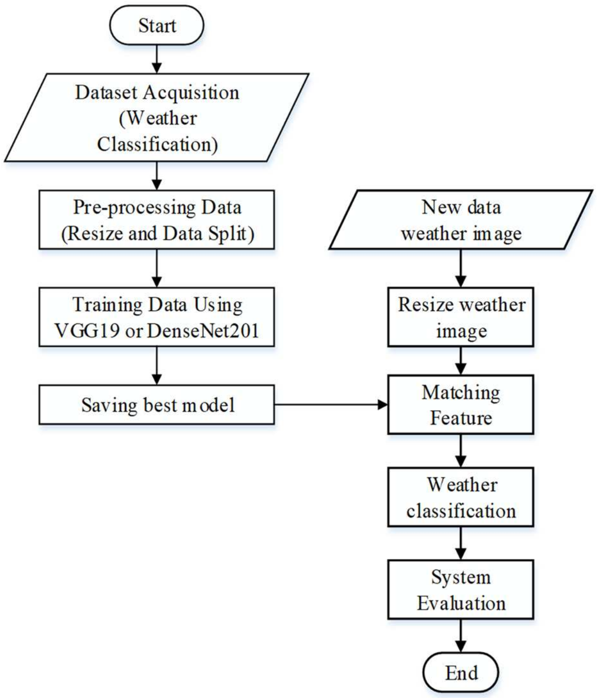

Fig. 1 The Proposed System Architecture

A. Data Acquisition

This research uses cloud datasets as weather classification with five classes: foggy, sunrise, shiny, rainy, and cloudy [17]. Weather is essential for farmers; farmers always look at the weather before farming [18]. This data was gathered from the internet, and images of interest with a Creative Commons license were taken from Flickr, Unsplashed, and Pixels. Many effective technologies have been developed to solve various problems involved in data acquisition and processing [19]. Different images are licensed under a variety of different licenses. There are around 1230 labeled photos in the dataset, including the validation images.

cloudy foggy rainy shine sunrise

Fig. 2 A sample dataset from this research

The images are not fixed in size, and the images vary in size. Each image is associated with a weather category and is saved in a separate folder corresponding to the identified class. This dataset has various sizes with RGB (Red, Green, and Blue) colors [20]. Fig. 2 shows an example of the dataset used in this study.

B. Pre-processing Data

In image processing technology, whether the data is binary, colored, or grayscale is irrelevant. By extracting features for identification, classification, diagnosis, classification, clustering, recognition, and detection, image processing can be carried out. Feature extraction techniques are employed to extract as much information as possible from an image. Selecting and extracting useful features is a big issue now [21]. The extraction of features can be done in a variety of ways depending on geometric, statistical, textural, and aesthetic factors. Numerous subtypes exist for each major form of feature; for instance, color features can be separated into three categories: average RGB, color histogram, and color moment [2]. The original data in this study varied in size. Therefore, before entering the feature extraction stage, the original dataset is resized to equalize its size. In this study, the dataset was resized to a size of 150 x 150. Then, the resizing results were divided into training data and testing data. In this study, the training data used is 80% of the total dataset in each class. The rest is used as testing data.

C. Feature Extraction

In this research, the convolution neural network (CNN) has attained state-of-the-art performance. However, there are still issues with CNN training's overall optimization [14], [22], [23]. The CNN used in this study uses cloud images that have been pre-processed. Cloud photos, part of extracted feature map vectorization, part of merging feature vectors, and fully connected layers [6]. The CNNs of VGG19 were employed in this investigation. This model deepens the CNNs, which raises the recognition rate. It moves through 19 layers of weight. Due to the exclusion of the fully connected layer, 16 weight layers are used in this investigation. It includes 5 pooling layers and 16 convolutional layers. The final feature map's dimensions (width, height, and depth) are shaped by the configuration of the pooling layer and the composite product neural network. 3 by 3 product and 2 by 2 max pooling were employed. A final feature map with a depth of 512 is extracted after passing through the feature map extraction layer, which results in a 32-fold reduction in width and height [24]. Fast the feature extraction process used transfer learning techniques. Unlike the case with deep learning, transfer learning techniques do not require much data during the training process [25][26]. Transfer learning was carried out in the previous training process with the ImageNet dataset and tested for accuracy [27]. The VGG19 and DenseNet201 transfer learning models were tested in this research.

1) VGG19 Model:

VGG uses six main structures, where each structure is composed of multiple connected convolutional and fully connected layers. The convolution kernel size is 3 x 3, and the input image used in this model is 224 x 244 x 3. As the model's name suggests, VGG-19 uses 19 layers. It was built and trained at the University of Oxford in 2015 by K. Simonyan and A. Zisserman [17]. The VGG-19 network is trained using photos from the ImageNet collection totaling more than 1 million. It was trained on colorful images with a resolution of 224 x 224 pixels. This network has been pre-trained to classify up to 1000 items. Fig 3 shows the architecture of the VGG-19 model.

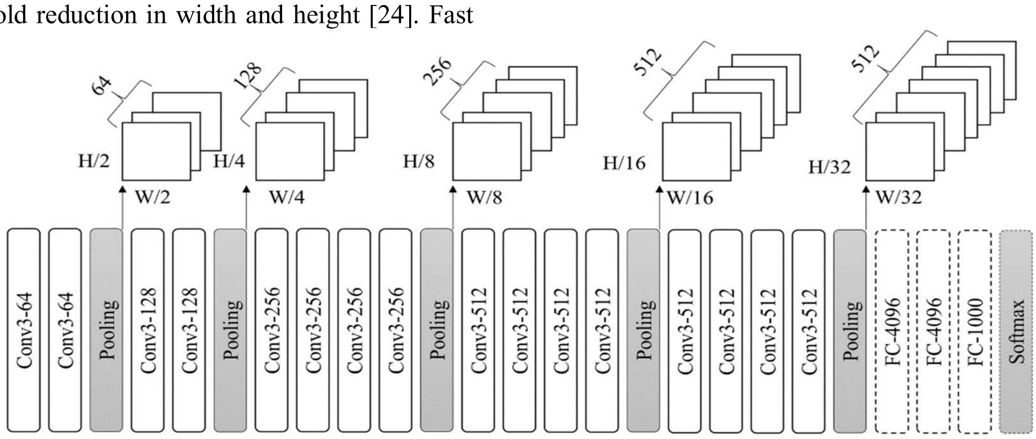

Fig. 3 The architecture of the VGG19 model [19]

2) DenseNet201 Model:

The DenseNet201 model is also a member of the DenseNet family of image classification models [28][29]. The primary difference between this model and the DenseNet201 model is the model's size and accuracy. The DenseNet201 is more extensive, at over 77MB than the DenseNet121, approximately 31MB in size. The inventor changed them from Torch to Caffe* format after they were initially trained on Torch. All DenseNet models have been trained in images from the ImageNet database. The model's input is a blob composed of a single image containing the numbers 1, 3, 224, 224 in the BGR sequence. Before feeding the image blob into the network, the BGR mean values must be removed: [103.94, 116.78, 123.68]. Furthermore, values must be multiplied by 0.017. DenseNets have the enhanced ability to transfer data and graphics throughout the network, which makes them simple to train. There is an implicit deep supervision because each layer has direct access to the gradients from the loss function and the original input signal [30]. Deeper network architecture training is made easier as a result. Additionally, we see that thick connections have a regularizing impact that lessens over-fitting [29]. DenseNets fully utilize the network's capabilities, producing condensed models that are simple to train and extremely parameter efficient. The input of succeeding layers is more varied and more effective when feature maps from various layers are combined. This is a significant distinction between DenseNets and ResNets. In contrast to Inception networks [31], [32], which include many data [29]. Fig 4 shows the architecture of the DenseNet201 model.

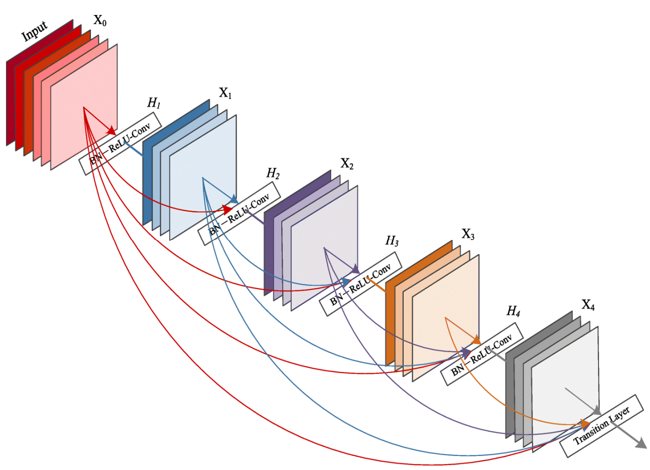

Fig. 4 The architecture of the DenseNet201 model [21]

D. Classification

Weather classification based on cloud datasets in this study uses transfer learning as feature extraction, and CNN is used as classification [33]. We freeze the VGG19 and DenseNet201 models for feature extraction and replace the classification layer with a Flatten or GAP layer. Layers that are fully connected are prone to overfitting. Dropout can be used to regularize by randomly setting half of the activations to fully connected layers to zero during training. It has enhanced generalization capacity and substantially reduces overfitting. In recent years, researchers have employed Global Average Pooling (GAP) layers to reduce overfitting by lowering the overall number of model parameters. Like max-pooling layers, GAP layers are used to compress a three-dimensional tensor's spatial dimensions. A tensor with dimensions $h\,x\,w\,x\,d$ is reduced in size by GAP layers to dimensions $1\,x\,1\,x\,d$, which is a more severe form of dimensionality reduction. GAP layers boil down each $h\ x\ w$ feature map to a single integer by averaging all h x w values.

E. System Evaluation

We tuned parameters on both models in this research, utilizing Flatten and GAP layers. The batch size and the number of epochs is the parameters investigated. For the epoch number experiment, the optimal batch size value was chosen. We make observations about the value of loss and the accuracy of measurements. Finally, we evaluate the findings using a confusion matrix to determine the accuracy of [34], [35].

III. Results and Discussion

A. Data Collection

In this study, we used a cloud image dataset. We selected data based on specific labels in the image set. The labels we chose were cloudy, foggy, rain, shine, and sunrise. Each of these images is used to interpret the weather. Cloud dataset data consists of 1230 data. We chose to collect data that only had specific labels (cloudy, foggy, rain, shine, and sunrise) and nothing else. In the end, we divided it into five labels as shown in Table I.

TABLE I
DISTRIBUTION OF DATA

Label	Quantity
Cloudy	240
Foggy	240
Rain	240
Shine	200
Sunrise	280

In the data collection process, we resized and divided it into test data and training data. We used the size 150 x 150 and .jpg format.

B. Experiment

For this study, we chose to use the VGG19 and DenseNet models. For our model, each convolution layer has an increasing number of nodes. We used 16 nodes in the input layer up to 128 nodes in the last layer. Convolutional layers are called convolutional because they work in the same concept as convolutional filters used in image processing. In the training process, the data was broken down into training and testing data. We used 80% of the dataset for the training data, and for the test data, we used 20% of the dataset. For the validation process, we used accuracy to measure the performance of our machine-learning model.

C. Results

The results of this research experiment can be seen in Table II, Table III, and Fig. 5. The results were separated into training and testing to facilitate the analysis of each model used.

TABLE II
RESULT OF TESTING

	Accuracy %				
	DenseNet		VGG19		Traditional CNN
	Fully Connected	Global Average	Fully Connected	Global Average	
Cloud	92.1	90	84	59	83
Foggy	97.8	93	80.3	74.6	64.6
Rain	90.6	87.5	90.3	93.6	93.1
Shine	99	93.4	81.9	71.5	85.9
Sunrise	99	96.6	97.5	83.3	72.4

Based on Table II, it can be seen the results of testing the data. The best results in the shine and sunrise classes are 99% with the DenseNet model of the fully connected type. Because the two classes have similar images, the DenseNet model can detect the cloud image well. Meanwhile, the lowest result of 59% was obtained in the cloud class using VGG19-GAP average pooling model. That happens because the GAP is only connected on average, which is fully connected. The results for the VGG19-fully connected model in the cloud class are 84%. The difference in outcomes between types can reach 26% in the VGG19 model for the cloud class.

Based on Table III, it can be seen the results for the training data. The highest results were obtained in the sunrise class with the DenseNet model fully connected and the VGG19-Fully Connected at 97%. Meanwhile, the lowest result is seen in the foggy class with the DenseNet-GAP type of 80.3%. Compared with the testing results, the results of this training show that the results in each class and model are not below 80%.

TABLE III
RESULT OF TRAINING

	Accuracy %				
	DenseNet		VGG19		Traditional CNN
	Fully Connected	Global Average Pooling	Fully Connected	Global Average Pooling	
Cloud	93	83.7	89	86	87
Foggy	85	80.3	89	81	82
Rain	88	92	90	94	87
Shine	91	91	86	83	88
Sunrise	97	94	97	90	93

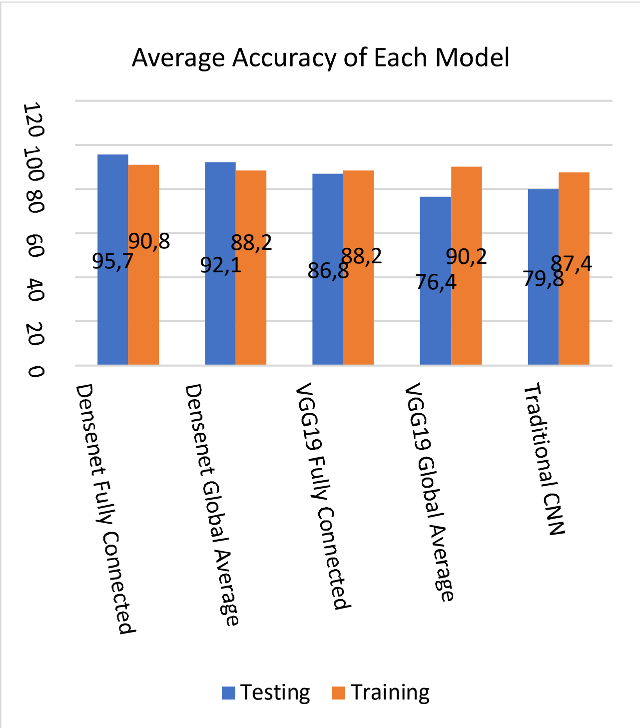

Fig. 5 Average Accuracy of Each Model

Fig. 5 is a graph of the results of the averages of Table II and Table III. Overall, the results show that the best in the training process is because it excels in three processing, namely VGG19-fully connected, VGG19-GAP, and traditional CNN. Meanwhile, in the testing process, the best results are DenseNet-fully connected and DenseNet-global average processing. Overall, both the training process and the testing model are suitable for use in this study, namely DenseNet-fully connected, with an accuracy of 95.7%. Meanwhile, overall, the model that is not suitable for use in this research problem is the VGG19-global average because it can only produce an accuracy of 76.4%. The traditional CNN model in this study makes an accuracy that is not too low and not too high.

D. Discussion

The performance comparison between Fully Connected Layer and Global Average Pooling on 2 different CNN architectures namely VGG19 and DenseNet201 has been conducted with the result that DenseNet201 with Fully Connected Layer scenario achieved the highest classification accuracy, 95.7%, for differentiating 5 weather classes with relatively small dataset. This result is interesting because several experiments in the past did not really pass this level of accuracy.

There were cloud classification experiments with 19 extracted features from the camera image that is passed into Random Forest classifier. However, this form of input only resulting in 78% accuracy for 7 cloud classes case [36]. Another experiment used their own 3 weather images dataset and applied with 10 different CNN models. However, they only achieve a maximum accuracy of 80.70%, which is comparable to our traditional CNN accuracy of 79.8% [37]. There is also another model called DeepCTC, where the model consists of 4 fully connected hidden layers that are fed by 16 neurons from GOES-16 ABI as input and 9 SoftMax nodes as clouds categories. Based on their experiment, their model achieves approximately 85% accuracy [38].

It shows us that partial features from the weather or cloud images do not result in good classification performance on the model. Weather prediction should be based on a full picture of the sky or environment. This experiment on weather prediction based on full picture of the environment reinforces this idea.

It also can be seen that based on our experiment; Fully Connected Layer performed better than Global Average Pooling in decision making for classifying weather pictures. The weakness in our experiment is that a significantly bigger dataset is required for model training; at the very least, the dataset should be in the 100,000 level for better performance. In the future, we hope that more weather datasets can be collected and trained to produce a better model.

IV. Conclusion

This study's objectives and experimental results estimate cloud weather based on cloud images using a convolutional neural network type model. This study evaluated cloud images for five types of cloud images: cloudy, foggy, rainy, shine, and sunrise. The model that has the best average accuracy result is DenseNet-fully connected. The highest accuracy is obtained in the sunrise and shine classes of 99% using the DenseNet-fully connected model. Meanwhile, the lowest accuracy is received in the cloud class using the VGG19-global average model of 59%.

Acknowledgment

The authors thank Directorate of Research and Community Service (DRPM) of the Directorate General of Strengthening Research and Development from the Ministry of Research, Technology and Higher Education of the Republic of Indonesia for funding this research through the 2020 Beginner Lecturer Research scheme.

REFERENCES

[1] G. Chavan and B. Momin, "An Integrated approach for Weather Forecasting over Internet of Things: A Brief Review," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017, pp. 83–88, doi: 10.1109/I-SMAC.2017.8058291.

[2] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting," *J. Adv. Model. Earth Syst.*, vol. 12, no. 11, pp. 1–17, 2020, doi: 10.1029/2020MS002203.

[3] S. Kaji and S. Kida, "Overview of image-to-image translation by use of deep neural networks: denoising, super-resolution, modality conversion, and reconstruction in medical imaging," *Radiol. Phys. Technol.*, vol. 12, no. 3, pp. 235–248, 2019, doi: 10.1007/s12194-019-00520-y.

[4] S. Scher, "Toward Data-Driven Weather and Climate Forecasting: Approximating a Simple General Circulation Model With Deep Learning," *Geophys. Res. Lett.*, vol. 45, no. 22, pp. 12,616-12,622, 2018, doi: 10.1029/2018GL080704.

[5] P. Dueben and P. Bauer, "Challenges and design choices for global weather and climate models based on machine learning," *Geosci. Model Dev.*, vol. 11, no. 10, pp. 1–17, 2018, doi: 10.5194/gmd-2018-148.

[6] S. H. Jeong, J. P. Yun, H. G. Yeom, H. J. Lim, J. Lee, and B. C. Kim, "Deep learning based discrimination of soft tissue profiles requiring orthognathic surgery by facial photographs," *Sci. Rep.*, vol. 10, no. 1, pp. 1–5, 2020, doi: 10.1038/s41598-020-73287-7.

[7] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 343–366, 2021, doi: 10.1007/s10044-020-00898-1.

[8] M. Elhoseiny, S. Huang, and A. Elgammal, "Weather classification with deep convolutional neural networks," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2015-Decem, pp. 3349–3353, 2015, doi: 10.1109/ICIP.2015.7351424.

[9] S. ur Rehman, S. Tu, O. ur Rehman, Y. Huang, C. M. S. Magurawalage, and C. C. Chang, "Optimization of CNN through novel training strategy for visual classification problems," *Entropy*, vol. 20, no. 4, pp. 1–10, 2018, doi: 10.3390/e20040290.

[10] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning," *J. Biomol. Struct. Dyn.*, vol. 39, no. 15, pp. 5682–5689, 2021, doi: 10.1080/07391102.2020.1788642.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[12] S. ur Rehman, S. Tu, O. ur Rehman, Y. Huang, C. M. S. Magurawalage, and C. C. Chang, "Optimization of CNN through novel training strategy for visual classification problems," *Entropy*, vol. 20, no. 4, pp. 1–11, 2018, doi: 10.3390/e20040290.

[13] I.-J. Kim and X. Xie, "Handwritten Hangul recognition using deep convolutional neural networks," *Int. J. Doc. Anal. Recognit.*, vol. 18, no. 1, pp. 1–13, 2015, doi: 10.1007/s10032-014-0229-4.

[14] K. Xu, W. Long, Y. Sun, and Y. Lin, "A novel image feature extraction algorithm based on the fusion AutoEncoder and CNN," *J. Phys. Conf. Ser.*, vol. 1646, no. 1, 2020, doi: 10.1088/1742-6596/1646/1/012039.

[15] R. Ferdiana, W. F. Dicka, and A. Boediman, "Cat Sounds Classification with Convolutional Neural Network," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 755–765, 2021, doi: 10.15676/ijeei.2021.13.3.15.

[16] K. W. Gunawan, A. A. Hidayat, T. W. Cenggoro, and B. Pardamean, "A Transfer Learning Strategy for Owl Sound Classification by Using Image Classification Model with Audio Spectrogram," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 546–553, 2021, doi: 10.15676/ijeei.2021.13.3.3.

[17] V. Gupta, "Weather dataset | Kaggle," Kaggle.com, 2020. https://www.kaggle.com/vijaygiitk/multiclass-weather-dataset (accessed Nov. 07, 2021).," 2020.

[18] L. W. Morton, J. M. McGuire, and A. D. Cast, "A good farmer pays attention to the weather," *Clim. Risk Manag.*, vol. 15, pp. 18–31, 2017, doi: 10.1016/j.crm.2016.09.002.

[19] R. F. Rachmadi and I. K. E. Purnama, "Paralel Spatial Pyramid Convolutional Neural Network untuk Verifikasi Kekerabatan berbasis Citra Wajah," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 4, pp. 152–157, 2018, doi: 10.14710/jtsiskom.6.4.2018.152-157.

[20] M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, "Fundus image classification using VGG-19 architecture with PCA and SVD," *Symmetry (Basel).*, vol. 11, no. 1, 2019, doi: 10.3390/sym11010001.

[21] E. Alpaydin, *Introduction to machine learning*, vol. 1107. 2014.

[22] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, "Feature Extraction Methods: A Review," *J. Phys. Conf. Ser.*, vol. 1591, no. 1, 2020, doi: 10.1088/1742-6596/1591/1/012028.

[23] M. A. Farooq, H. Javidnia, and P. Corcoran, "Performance estimation of the state-of-the-art convolution neural networks for thermal images-based gender classification system," *J. Electron. Imaging*, vol. 29, no. 06, pp. 1–20, 2020, doi: 10.1117/1.jei.29.6.063004.

[24] M. Bojarski *et al.*, "VisualBackProp: Efficient visualization of CNNs for autonomous driving," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4701–4708, 2018, doi: 10.1109/ICRA.2018.8461053.

[25] E. Lima, X. Sun, J. Dong, H. Wang, Y. Yang, and L. Liu, "Learning and Transferring Convolutional Neural Network Knowledge to Ocean Front Recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 354–358, 2017, doi: 10.1109/LGRS.2016.2643000.

[26] F. D. Adhinata, N. A. F. Tanjung, W. Widayat, G. R. Pasfica, and F. R. Satura, "Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 2, p. 230, 2021, doi: 10.26555/jiteki.v7i2.20758.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[28] M. A. Farooq, H. Javidnia, and P. Corcoran, "Performance estimation of the state-of-the-art convolution neural networks for thermal images-based gender classification system," *J. Electron. Imaging*, vol. 29, no. 06, pp. 1–20, 2020, doi: 10.1117/1.jei.29.6.063004.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[30] C. Y. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *J. Mach. Learn. Res.*, vol. 38, pp. 562–570, 2015.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[32] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[33] K. W. Gunawan, A. A. Hidayat, T. W. Cenggoro, and B. Pardamean, "A transfer learning strategy for owl sound classification by using image classification model with audio spectrogram," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 546–553, 2021, doi: 10.15676/IJEEI.2021.13.3.3.

[34] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.

[36] Huertas-Tato, J., Rodríguez-Benítez, F. J., Arbizu-Barrena, C., Aler-Mur, R., Galvan-Leon, I., & Pozo-Vázquez, D. (2017). Automatic cloud-type classification based on the combined use of a sky camera and a ceilometer. Journal of Geophysical Research: Atmospheres, 122, 11,045– 11,061. doi:10.1002/2017JD027131.

[37] J. C. Villarreal Guerra, Z. Khanam, S. Ehsan, R. Stolkin and K. McDonald-Maier, "Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks," 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2018, pp. 305-310, doi: 10.1109/AHS.2018.8541482.

[38] Afzali Gorooh, Vesta, Subodh Kalia, Phu Nguyen, Kuo-lin Hsu, Soroosh Sorooshian, Sangram Ganguly, and Ramakrishna R. Nemani. 2020. "Deep Neural Network Cloud-Type Classification (DeepCTC) Model and Its Application in Evaluating PERSIANN-CCS" Remote Sensing 12, no. 2: 316. doi:10.3390/rs12020316.