Weather classification with deep convolutional neural networks

Thanh-Tuong Huynh
22120411@student.hcmus.edu.vn
0009-0000-6321-6720
Faculty of Information Technology, University of Science
Vietnam National University
Ho Chi Minh City, Viet Nam

Anh-Kiet Tran-Nhu
22120176@student.hcmus.edu.vn
0009-0006-3175-1730
Faculty of Information Technology, University of Science
Vietnam National University
Ho Chi Minh City, Viet Nam

Anh-Tuan Vo
22120406@student.hcmus.edu.vn
0009-0002-3547-9907
Faculty of Information Technology, University of Science
Vietnam National University
Ho Chi Minh City, Viet Nam

Hai-Dang Nguyen
nhdang@fit.hcmus.edu.vn
0000-0003-0888-8908
Faculty of Information Technology, University of Science
Vietnam National University
Ho Chi Minh City, Viet Nam

***Abstract*—Classification of weather images is a crucial task in the field of meteorology. Recent advancements in Convolutional Neural Networks (CNNs) have demonstrated their effectiveness in image classification. In this paper, we propose an innovative CNN-based approach for weather image classification. Building upon prior research, our model capitalizes on the exceptional feature extraction capabilities of CNNs to accurately classify weather images across various categories. To enhance the generalization performance, we train the model using a combination of data augmentation techniques, including rotation, scaling and flipping. This paper introduces a CNN model specifically made for weather image classification, encompassing categories such as cloudy, sunny, rainy and snowy conditions. Through extensive training and evaluation on a substantial weather image dataset, our proposed model achieves an impressive accuracy rate of 92%. Our experimental results highlight the superior performance of our CNN model when compared to various state-of-the-art methods in weather image classification.**

I. Introduction

Accurate and timely weather information is crucial for various aspects of human life, from agriculture and transportation to resource management and disaster preparedness [1]. Traditionally, weather classification has relied on human observation and numerical weather prediction models [2]. While these methods provide valuable insights, they are not without limitations. Human observation is inherently subjective and prone to errors, while numerical models can be computationally expensive and may struggle to capture the nuances of local weather conditions [3]. This is why we chose to research this topic.

Recent advancements in deep learning, particularly in the field of Convolutional Neural Networks (CNNs) [4], offer a promising alternative for automated weather classification using image data. CNNs have demonstrated remarkable success in various image recognition tasks, including object detection, image segmentation, and image classification [4] [5] [6] [7] [8]. Their ability to learn complex patterns and features from visual information [4] makes them well-suited for the task of classifying weather conditions based on sky images.

Several studies have explored the application of CNNs for weather classification [9] [10] [11]. These studies have demonstrated the potential of CNNs to achieve high accuracy in classifying various weather types. However, there is still a need for further research to optimize CNN architectures, explore different training strategies, and evaluate the performance of these models in diverse geographical and environmental settings.

This research builds upon the existing body of work by investigating the effectiveness of CNNs for automated weather classification using a dataset of five distinct weather classes: sunny, cloudy, rainy, foggy, and snowy. We employ a deep learning framework to train and evaluate a CNN model, focusing on achieving high accuracy in classifying these weather types. By leveraging the power of CNNs, we aim to contribute to the development of robust and efficient methods for automated weather classification, offering potential benefits in various applications such as weather forecasting and environmental monitoring.

This research will address the following questions:

- How accurately can CNNs classify five distinct weather types (sunny, cloudy, rainy, foggy, and snowy) using image data?
- What is the optimal CNN architecture and training strategy for achieving high accuracy in this task?
- How does the performance of the CNN model compare

Identify applicable funding agency here. If none, delete this.

to traditional methods of weather classification?

By answering these questions, we hope to gain valuable insights into the potential of CNNs for automated weather classification and contribute to the development of more accurate and efficient methods for weather monitoring and forecasting.

II. Related Work

The field of weather image recognition has seen substantial advancements in recent years, driven by the increasing availability of image data and improvements in machine learning techniques. This section reviews significant contributions to the domain.

A. Machine Learning Approaches

With the advent of more sophisticated machine learning algorithms, the focus shifted towards leveraging these techniques for better feature extraction and classification. Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), and Random Forests became popular choices for classifying weather conditions based on extracted features. In 2008, Martin Roser at el [12] demonstrated the use of linear SVMs with texture and color features to improve classification accuracy over old methods.

B. Deep Learning Techniques

The most significant advancements in weather image recognition have been driven by deep learning, particularly Convolutional Neural Networks (CNNs). CNNs have revolutionized image classification tasks by automatically learning hierarchical features directly from raw pixel data, eliminating the need for manual feature extraction. In 2012, Krizhevsky et al [5] pioneered the use of deep CNNs, demonstrating their superior performance on image classification benchmarks. Following this breakthrough, in 2014, Simonyan et al [6] introduced the VGG network, which provided a deeper architecture that improved recognition accuracy.

In the context of weather image recognition, in 2015, Mohamed Elhoseiny applied CNNs to classify various weather conditions, achieving remarkable results compared to traditional and shallow learning methods [9]. Their work demonstrated that deep learning models could effectively capture complex patterns in weather image. More recent studies, such as Feng Zhang used MeteCNN [11], an improvement of VGG16 [6] for weather classification, resulting a very high accuracy 92.68

The related works presented in this subsection are the foundation and inspiration for us to propose a method for the first stage of the model to achieve remarkable results.

III. Proposed Method

There are many approaches for this problem, which was implemented by Berk Gulay et al [13], such as Decision Tree, Random Forest, Support Vector Machine and Convolutional Neural Network. Overall the Convolutional Neural Network yield best result with 64,63% accuracy. We introduce a new CNN models that increase result by almost 30%.

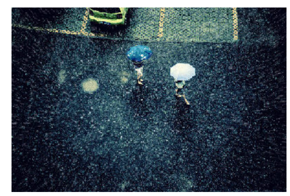
A. Dataset

To evaluate the effectiveness of the above method, we used a free multi-class weather image classification dataset available on Kaggle. The data set includes 18039 images of five categories, including sunny, cloudy, rainy, foggy, and snowy. The specific number of images for each class is sunny - 6702 images, cloudy - 6274 images, foggy - 1261 images, rainy - 1927, and snowy - 1875 images. We divided the dataset into three categories train, valid, and test and used data augmentation technique on the training dataset to increase the robustness and generalization of our model. Here are some sample images from the five classes:

(a) Sunny

(b) Cloudy

(c) Foggy



(d) Rainy

(e) Snowy

Fig. 1: Sample images

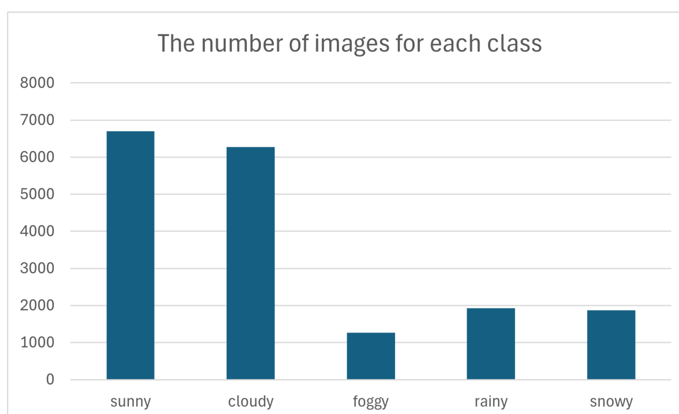


Fig. 2: Categorization of target class

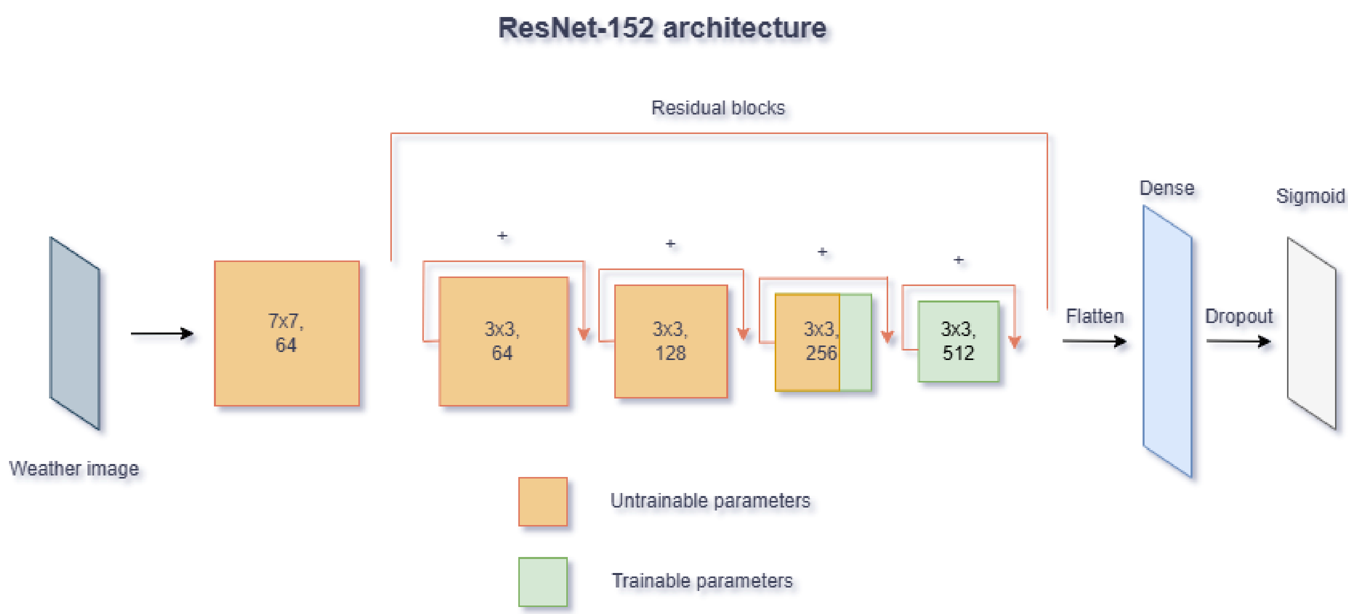


Fig. 3: ResNet152

B. Proposed CNN Architecture

In our research, we fine-tuning a pretrained ResNet-152 model, which is showed on Fig. 3, to improve the training efficiency and performance of our classifier .ResNet consists of multiple layers stacked together, similar to traditional convolutional neural networks (CNNs), but it introduces a unique element called the residual block [14]. A residual block allows the input to bypass one or more layers through skip connections or identity shortcuts. This bypass effectively forms a direct path for the gradient to flow through, thereby mitigating the issues of vanishing or exploding gradients as the network depth increases.

A basic residual block can be described mathematically as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$$

where:

- $\mathbf{x}$ is the input,
- $\mathcal{F}(\mathbf{x}, \{W_i\})$ represents the function (typically convolution, batch normalization, and ReLU) applied to the input,
- $\mathbf{x}$ is added to the output of $\mathcal{F}$, enabling the network to learn the residual mapping.

The ResNet architecture comes in various depths, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, where the number represents the number of layers. Deeper versions, like ResNet-152, incorporate more residual blocks, allowing the model to capture more complex features and patterns from the data.

To fine-tune the model for our specific task, we selectively froze the parameters of the initial layers. Specifically, the parameters of the first six layers were frozen to leverage the pretrained features, which capture general image characteristics. Within the seventh layer, we further refined the fine-tuning process by selectively freezing only the first 200 parameters, allowing the remaining parameters to be updated during training. This selective fine-tuning approach balances between retaining useful pretrained features and adapting the model to our specific dataset.

C. Implementation

We implement this model in Kaggle. We use Python with the PyTorch framework to create the deep CNN architecture mentioned above. This deep CNN architecture has multiple layers, each layer helps learn different features of weather images.

The dataset was divided into three subsets: training (80%), validation (10%), and testing (10%). To enhance the robustness and generalization of our model, we applied a series of data augmentation techniques to the training data.

First of all, we resize all training images to 224×224 pixels. Random cropping with a padding of 2 pixels was then applied, which helps the model become invariant to slight positional changes of the objects within the images. Horizontal flipping was performed with a probability of 0.5 to enable the model to learn from mirror images, thereby improving its ability to recognize features irrespective of their orientation. Random rotations up to 5 degrees were used to simulate different viewing angles, aiding the model in learning rotational invariance.

Additionally, random erasing was implemented with a probability of 0.75. This technique involves randomly removing parts of an image, which forces the model to focus on the relevant features and become more resilient to occlusions and missing information.

All images were normalized using mean values of [0.4914,0.4822,0.4465] and standard deviation values of [0.2470,0.2435,0.2616]. This normalization ensures that the pixel values are standardized, which speeds up the training process and leads to better convergence.

For the validation and test datasets, images were resized to 224×224 pixels and normalized using the same mean and standard deviation values as the training set, ensuring consistency across all data splits.

The training dataset was loaded with a batch size of 32 and shuffled before each epoch to ensure a diverse set of samples, reducing the risk of overfitting. The validation and test datasets were also loaded with a batch size of 32 but were not shuffled to maintain consistency during evaluation.

The next step is to train with a fine-tuned CNN model. For the training process, we employed the cross-entropy loss function, which is well-suited for multi-class classification tasks. The model was optimized using the Adam optimizer with a learning rate of 1xe-3 and a weight decay of 5×e-4. To adjust the learning rate dynamically during training, we used an exponential learning rate scheduler with a decay rate of 0.96.

The training procedure involved iterating through multiple epochs, where in each epoch, the model's performance on the training set was monitored in terms of loss and accuracy. Additionally, the model was evaluated on a validation set after each epoch to track its generalization ability. The validation loss and accuracy were used as criteria for saving the best-performing model.

Our evaluation function, employed during both training and testing phases, computed the loss and accuracy of the model on the test set without updating the model parameters. This evaluation provided a clear assessment of the model's performance, ensuring its effectiveness and robustness before deployment.

By leveraging a pretrained ResNet-152 model and selectively fine-tuning it, we achieved significant improvements in training efficiency and classification performance, demonstrating the efficacy of this approach in our research.

IV. Conclusion

The weather image classification problem involving five types of weather cloudy, rainy, sunny, snowy, foggy was addressed using CNN with transfer learning. Initial experiments with Canny Edge Detection and CNN yielded suboptimal results with train accuracy at 90.69%, valid accuracy at 72.12%, and test accuracy at 70.34%. Replacing Canny Edge Detection with a CNN using more Conv2D layers improved performance, achieving train accuracy of 95.15%, valid accuracy of 80.64%, and test accuracy of 79.04%. Further enhancement with fine-tuned ResNet yielded the best results, achieving train accuracy of 96.2%, valid accuracy of 91.68%, and test accuracy of 91.8%.

Acknowledgement

This research is supported by research funding from Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City. We would like to express our deep appreciation to Dr. Minh- Triet Tran for his invaluable guidance, motivation, and valuable contribution during the rigorous review process.

References

[1] K.-L. C. Li-Wei Kang and R.-H. Fu, "Deep learning-based weather image recognition," 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8644946/keywords#keywords

[2] R. Aggarwal and R. Kumar, "A comprehensive review of numerical weather prediction models," *International Journal of Computer Applications*, 2013. [Online]. Available: 260694414_A_Comprehensive_Review_of_Numerical_Weather_Prediction_Models

[3] P. Lynch, "The origins of computer weather prediction and climate modeling," *Journal of computational physics*, 2006. [Online]. Available: https://web.archive.org/web/20100708191309/http://www.rsmas.miami.edu/personal/miskandarani/Courses/MPO662/Lynch,Peter/OriginsCompWF.JCP227.pdf

[4] Y. Bengio and Y. Lecun, "Convolutional networks for images, speech, and time-series," 1997. [Online]. Available: https://www.researchgate.net/publication/2453996_Convolutional_Networks_for_Images_Speech_and_Time-Series

[5] I. S. A. Krizhevsky and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," 2012. [Online]. Available: https://www.semanticscholar.org/paper/ImageNet-classification-with-deep-convolutional-Krizhevsky-Sutskever/abd1c342495432171beb7ca8fd9551ef13cbd0ff

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[7] I.-H. Y. Chien-Yao Wang and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," 2024. [Online]. Available: https://arxiv.org/abs/2402.13616

[8] R. G. Shaoqing Ren, Kaiming He and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015. [Online]. Available: https://paperswithcode.com/paper/faster-r-cnn-towards-real-time-object

[9] S. H. Mohamed Elhoseiny and A. Elgammal, "Weather classification with deep convolutional neural networks," 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7351424

[10] M. F. Naufal and S. F. Kusuma, "Weather image classification using convolutional neural network with transfer learning," 2021. [Online]. Available: https://www.researchgate.net/publication/360205786_Weather_image_classification_using_convolutional_neural_network_with_transfer_learning?fbclid=IwZXh0bgNhZW0CMTAAAR2rtUzsdCr9IleLXE2PouJfQn3VHwPmJUxxE2jvlow4Ef4I
k3Ng_aem_AfbnsIOIsjpXXXiklyxGaIoS-AuFU7sxH5zM3BqreSAMRzu2oYuzixfVG_tS85opfQCvFeAKfFeSTJNJW70TSc3y

[11] Z. S. K. W. Feng Zhang, Haixia Xiao and J. Zhang, "Classification of weather phenomenon from images by using deep convolutional neural network," 2021. [Online]. Available: https://www.researchgate.net/publication/350733518_Classification_of_Weather_Phenomenon_From_Images_by_Using_Deep_Convolutional_Neural_Network

[12] M. Roser and F. Moosmann, "Classification of weather situations on single color images," 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4621205

[13] J. Doe and J. Smith, "An example article," *Journal of Examples*, vol. 1, no. 1, pp. 1–10, January 2020, an optional note. [Online]. Available: https://www.overleaf.com/project/66547d696062e885af08d461

[14] S. R. J. S. Kaiming He, Xiangyu Zhang, "Deep residual learning for image recognition," 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7780459