Weather Image Classification Report

Carlos Velasco, ITC A01708634, Tecnológico de Monterrey Campus Querétaro

Abstract.- This project explores the development of a machine learning model capable of classifying weather phenomena based on images. Two model iterations were conducted: the first using a Recurrent Neural Network (RNN) based on academic references, and the second employing a Convolutional Neural Network (CNN) inspired by advancements in image classification literature. The CNN ultimately outperformed the RNN significantly in both accuracy and reliability.

I. Introduction

The ability to classify weather conditions from images has important applications in areas such as environmental monitoring, autonomous vehicles, and safety forecasting. This project aims to build a machine learning model that can predict the type of weather depicted in an image, returning the top three probable classes along with their corresponding probabilities.

II. Dataset

The dataset, sourced from Kaggle, contains 6,862 images categorized into 11 different weather classes:

- Dew
- Fog/Smog
- Frost
- Glaze
- Hail
- Lightning
- Rain
- Rainbow
- Rime
- Sandstorm
- Snow

Each image is stored in a folder corresponding to its class label, simplifying automatic data loading and labeling.

III. Analysis Process

The process started with dataset preparation, which included loading the images, visual inspection for quality control, and manual cleaning to ensure dataset reliability. At the end, all images from the Kaggle dataset were used because no corrupted or no readable images were found. The images were then normalized and resized to 64x64 pixels for computational efficiency. Custom PyTorch datasets and dataloaders were created for efficient batch processing.

IV. 1st Model Proposal

The first model used a simple Recurrent Neural Network (RNN) architecture. A RNN is a deep neural network trained on sequential or time series data to create a machine learning (ML) model that can make sequential predictions or conclusions based on sequential inputs.

This approach was based on academic literature that applied RNNs for visual tasks:

- Long-term Recurrent Convolutional Networks for Visual Recognition and Description (Donahue et al., 2015)
- Recurrent Models of Visual Attention (Mnih et al., 2014).

These studies demonstrated the ability of RNNs to capture spatial-temporal dependencies in visual data.

My hypothesis was that RNNs could recognize spatial patterns across the sequence of pixels.

The RNN was designed to process images flattened into sequences and trained over 10 epochs.

V. Train and Test for 1st Proposal

The RNN model was trained using Cross Entropy Loss and evaluated with accuracy as the main performance metric. The dataset was split into training (85%) and validation (15%).

VI. 1st Model Results

- **Training Accuracy**: 72.68%
  Training Loss: 0.7536
- **Validation Accuracy**: 58.15%
- **Validation Loss**: 1.111

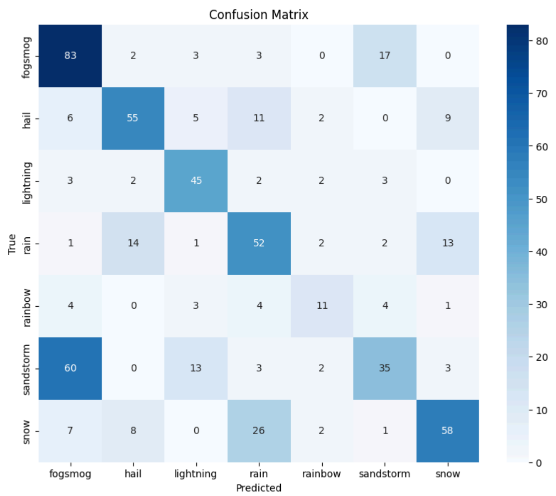

Figure 1. Confusion Matrix for model first iteration

The confusion matrix shows that while the model frequently predicts the correct class, it struggles with distinguishing visually similar classes, such as "sandstorm" and "fog/smog".

VII. 1st Model Result Interpretation

The model exhibited **overfitting**. Training accuracy was significantly higher than validation accuracy, and training loss was lower compared to validation loss, suggesting that the model learned the training data too well and failed to generalize effectively.

VIII. 1st Model Conclusions

The RNN-based approach achieved moderate success but was ultimately insufficient for the task of weather image classification. Although inspired by previous studies, RNNs are inherently more suited for sequential data rather than spatial pattern recognition in images.

IX. How to Improve the 1st Model

Given the nature of the problem (identifying spatial patterns in images) a Convolutional Neural Network (CNN) would be more appropriate. CNNs are designed to detect spatial hierarchies and features in visual data, unlike RNNs which are better suited for sequential dependencies.

This decision aligns with the work of Elhoseiny, Huang, and Elgammal in the paper "Weather Classification with Deep Convolutional Neural Networks", which demonstrated that CNNs achieve superior performance in weather image classification by efficiently capturing local and global spatial features.

X. Proposal of 2nd Model: CNN

The second model was based on a Convolutional Neural Network (CNN) architecture.
A CNN is a type of deep learning algorithm that is most often applied to analyze and learn visual features from large amounts of data. While primarily used for image-related AI applications, CNNs can be used for other AI tasks, including natural language processing and in recommendation engines.
The CNN model included convolutional layers, ReLU activations, max pooling, and fully connected layers. Regularization techniques such as dropout were incorporated to mitigate overfitting.

XI. 2nd Model Results

- **Training Accuracy**: 90.68%
  Training Loss: 0.2792
- **Validation Accuracy**: 83.05%
- **Validation Loss**: 0.63

Below is the graph showing training and validation accuracy across epochs:

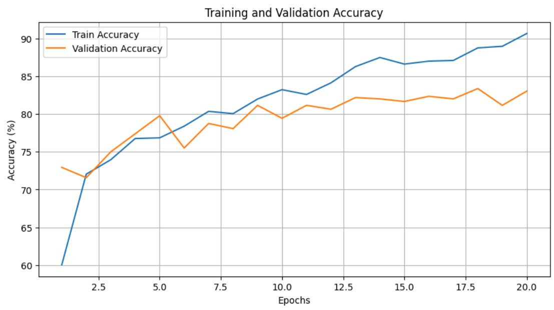

Figure 2. Training and Validation accuracy across epochs in model second iteration.

The CNN showed significant improvement over the RNN, achieving higher accuracies and lower losses. While there is still a slight overfitting (visible as a small gap between training and validation accuracies), it is much less pronounced than in the first model.

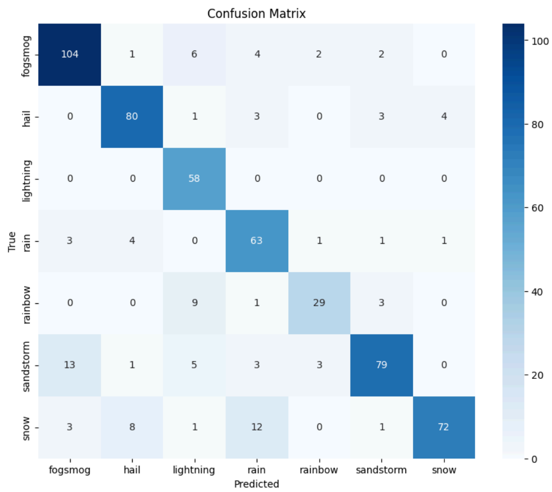

Figure 3. Confusion Matrix for model second iteration

The confusion matrix indicates that the model can now much more reliably distinguish between classes, with fewer misclassifications even among similar weather conditions.

XII. 2nd Model Results Interpretation

The CNN model demonstrated a slight overfitting problem. As we can see in Figure 3, the validation and training accuracies are not aligned, meaning that the data is somewhat memorizing the training data. Nevertheless, the difference between these sets is not that high, so I can say that the model performs well on unseen data.

XIII. 2nd Model Conclusions

The CNN-based model significantly outperformed the RNN model. This confirms that CNNs are a superior choice for image classification tasks, especially those involving complex visual patterns like weather phenomena. The second model not only achieved higher accuracy but also exhibited greater robustness and generalization ability compared to the RNN.

XIV. References

1. Convolutional neural networks (CNN) and deep learning. (n.d.). Intel. https://www.intel.la/content/www/xl/es/internet-of-things/computer-vision/convolutional-neural-networks.html

2. Donahue, J., Lisa, A. H., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. https://openaccess.thecvf.com/content_cvpr_2015/html/Donahue_Long-Term_Recurrent_Convolutional_2015_CVPR_paper.html

3. IBM. (2025, April 17). Recurrent Neural Network (RNN). IBM. https://www.ibm.com/think/topics/recurrent-neural-networks

4. Mnih, V., Heess, N., Graves, A., & Kavukcuoglu, K. (2014). Recurrent Models of Visual Attention. https://proceedings.neurips.cc/paper_files/paper/2014/hash/3e456b31302cf8210edd4029292a40ad-Abstract.html

5. Mohamed E., Sheng H., & Ahmed E. (2015). Weather classification with deep convolutional neural networks. https://ieeexplore.ieee.org/abstract/document/7351424/

6. Weather image recognition. (2021, November 26). Kaggle.

https://www.kaggle.com/datasets/jehanbhathena/weather-dataset