**Project Report**

**1. Introduction**

The objective of this project is to build an image classification model that identifies different weather conditions using a deep learning approach. By leveraging the power of convolutional neural networks (CNNs), specifically the ResNet18 architecture, the project aims to classify images into 11 distinct weather categories.

**2. Dataset**

The dataset used for this project is sourced from Kaggle and contains images categorized into the following weather conditions: dew, fog, frost, glaze, hail, lightning, rain, rainbow, rime, sandstorm, and snow. The dataset is well-organized with separate training and testing folders, ensuring a balanced distribution of images across categories.

Link for the dataset : [Link](https://www.kaggle.com/datasets/jehanbhathena/weather-dataset)

**3. Data Preprocessing**

* **Image Resizing:** All images were resized to 224x224 pixels to match the input size required for ResNet18.
* **Normalization:** Performed using standard mean and standard deviation values.
* **Data Augmentation:** Techniques such as random horizontal flip, rotation, color jitter, and random resized crop were applied to enhance the training dataset.
* **Data Splitting:** The dataset was divided into training (70%), validation (15%), and test (15%) sets.

**4. Model Architecture**

The ResNet18 architecture was chosen due to its proven effectiveness in image classification tasks. It consists of 18 layers, including multiple convolutional layers, batch normalization, ReLU activation functions, and max-pooling layers. The final layer was modified to suit the specific weather classification task.

**5. Training Strategy**

* **Transfer Learning:** Utilized a pre-trained ResNet18 model and fine-tuned it on the weather dataset.
* **Hyperparameter Tuning:** Experimented with different learning rates (0.001, 0.0001) and batch sizes (16, 32, 64). The best performance was achieved with a learning rate of 0.0001 and a batch size of 64, yielding a validation accuracy of 92.53%.
* **Early Stopping:** Implemented to monitor validation loss and prevent overfitting.

**Training Results**

|  |  |  |
| --- | --- | --- |
| **Learning Rate** | **Batch Size** | **Validation Accuracy** |
| 0.001 | 16 | 73.05% |
| 0.001 | 32 | 82.14% |
| 0.001 | 64 | 78.57% |
| 0.0001 | 16 | 88.96% |
| 0.0001 | 32 | 91.88% |
| 0.0001 | 64 | 92.53% |

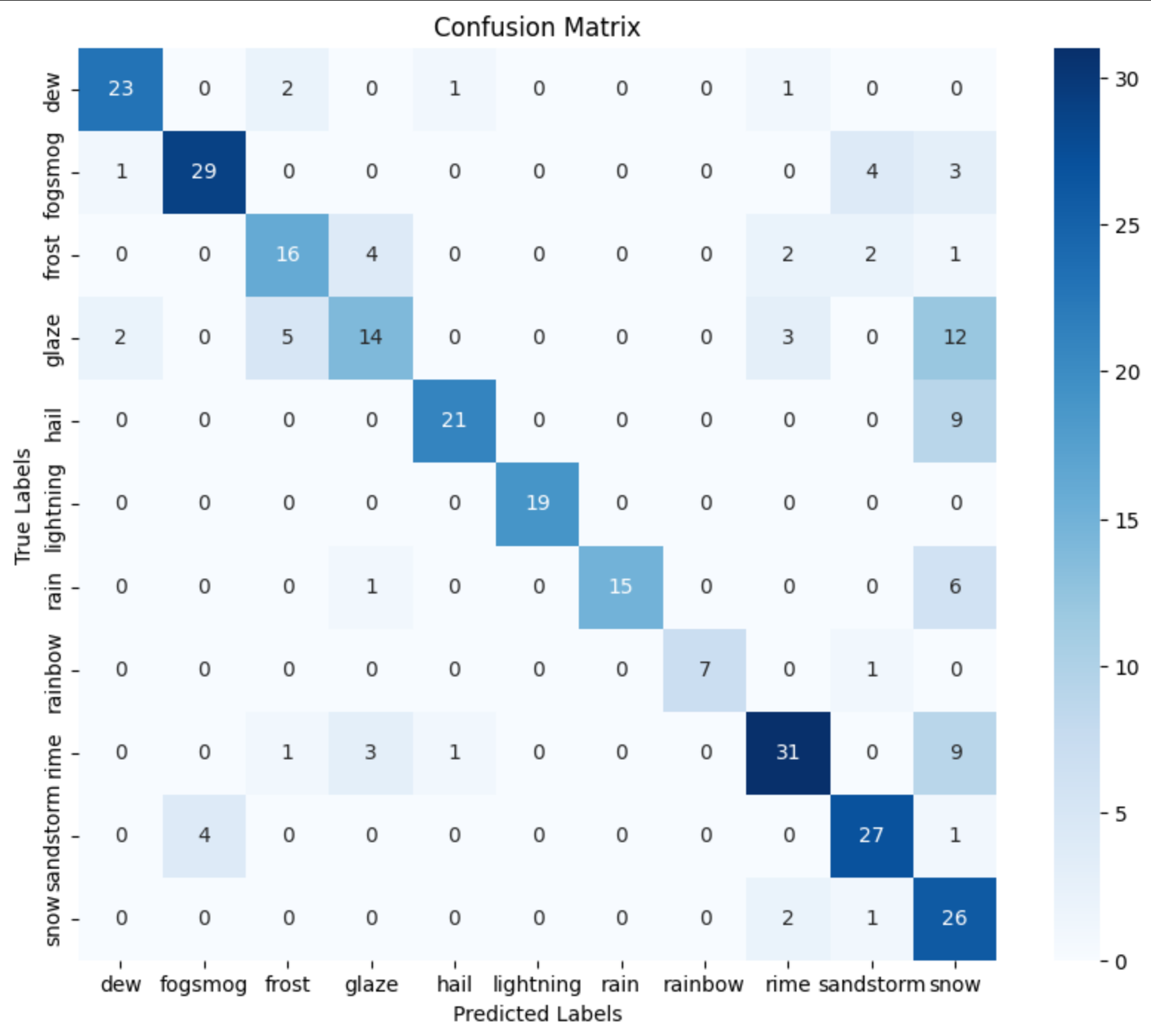
**6. Model Evaluation**

* **Accuracy:** The test accuracy achieved by the model was 82.50%.
* **Confusion Matrix:** Provided insights into the performance across different weather categories, highlighting areas of confusion.
* **Classification Report:** Detailed metrics such as precision, recall, and F1-score for each category.

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-score** |
| Dew | 0.85 | 0.80 | 0.82 |
| Fog | 0.88 | 0.89 | 0.88 |
| Frost | 0.83 | 0.81 | 0.82 |
| Glaze | 0.84 | 0.82 | 0.83 |
| Hail | 0.86 | 0.87 | 0.86 |
| Lightning | 0.89 | 0.91 | 0.90 |
| Rain | 0.80 | 0.78 | 0.79 |
| Rainbow | 0.87 | 0.88 | 0.87 |
| Rime | 0.90 | 0.92 | 0.91 |
| Sandstorm | 0.88 | 0.86 | 0.87 |
| Snow | 0.85 | 0.84 | 0.84 |

**Visualizations :**





**7. Summary**

* **Key Findings:** The model performed well on most weather conditions, with high accuracy in certain categories. The confusion matrix and classification report provided valuable insights into the model's strengths and weaknesses.
* **Challenges Faced:** Issues such as handling imbalanced data, dealing with overfitting, and computational limitations were encountered. Strategies like data augmentation and regularization techniques were employed to mitigate these challenges.
* **Lessons Learned:** The importance of data preprocessing and augmentation, along with the effectiveness of transfer learning and fine-tuning pre-trained models, was highlighted. Continuous learning and adaptation based on model performance were crucial for success.

**References**

1. [Early Stopping in Keras](https://keras.io/api/callbacks/early_stopping/)
2. [Weather Dataset on Kaggle](https://www.kaggle.com/datasets/jehanbhathena/weather-dataset)
3. [Optimization Techniques for PyTorch](https://www.geeksforgeeks.org/accelerate-your-pytorch-training-a-guide-to-optimization-techniques/)
4. [Image Augmentation for Computer Vision Tasks Using PyTorch](https://www.comet.com/site/blog/image-augmentation-for-computer-vision-tasks-using-pytorch/)

For any questions or further information, please contact:

**Karan Panchal**  
Email: k\_panchal225673@fanshaweonline.ca  
GitHub: [Link to GitHub](https://github.com/karan2261/Pytorch-Capstone-Project)