NN Assignment Report

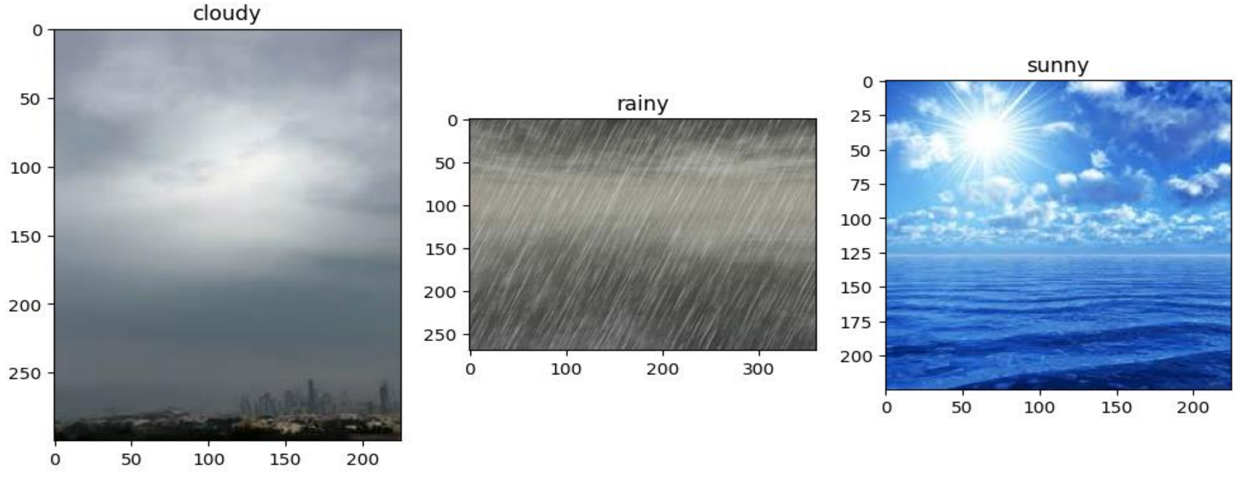
1 - Executive Summary

The purpose of this assignment was to create a convolutional neural network (CNN) model for weather condition classification of landscape images. The primary steps include environment setup, data readiness and exploration, basic CNN development, data augmentation and tuning the model then testing the optimum model. The dataset contained landscape images divided into three categories: cloudy, rainy and sunny. There were class imbalance and overfitting problems which were managed through data simulation as well as tuning the model. Though producing encouraging training results, performance on testing set by the last model still suggested some more room for improvement.

2 - Model Building

2.1 - Data Description

The dataset comprised 550 landscape images, divided into three weather categories: cloudy (285 images), rainy (70 images), and sunny (150 images). The images varied in dimensions, with the smallest being 111x111 pixels. The data was split into a training set (505 images) and a testing set (45 images).



2.2 - Data Preparation

The data preparation involved the following steps:

- Resizing all images to a uniform dimension of 64x64 pixels.
- Normalizing pixel values to range from 0 to 1.
- Encoding the categorical labels into one-hot vectors.
- Splitting the data into training and testing sets2.3 - Basic Model

2.3 - Basic Model

This initial CNN model was designed with the following architecture:

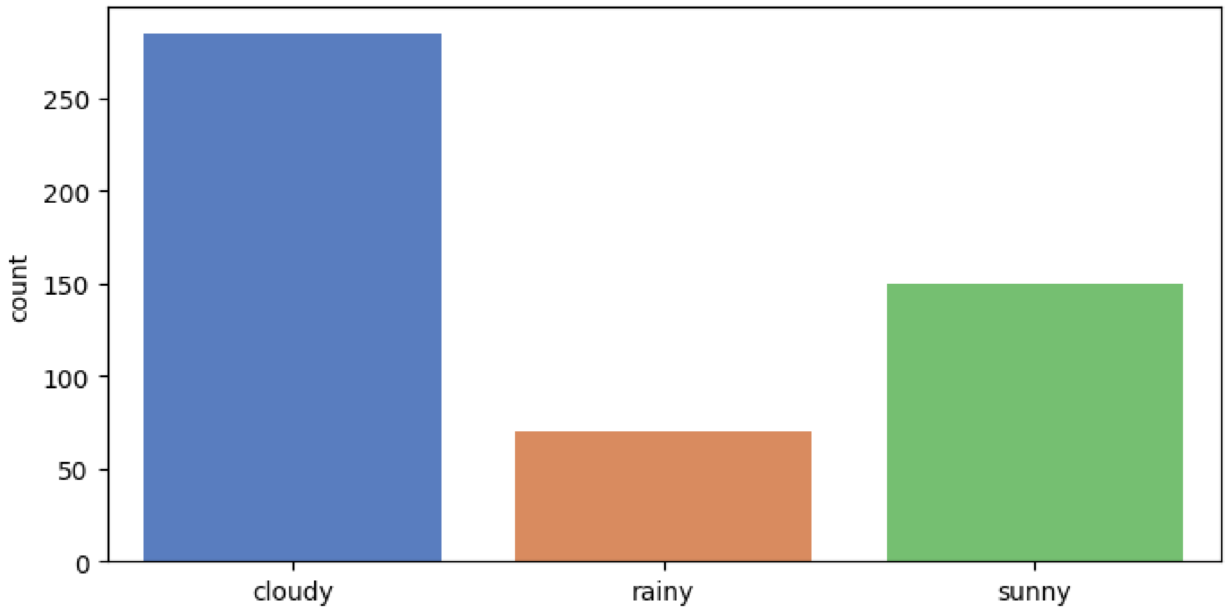
- Input layer of size (64, 64, 3) .
- Two convolutional layers that use ReLU activation function and MaxPooling.
- Flatten layer.
- Dense layer with 128 units using ReLU activation function.
- Output layer with softmax activation for classifying into three categories.

Trial Number	Training Accuracy	Validation Accuracy
1	0.76	0.00
2	0.71	0.00
3	0.84	0.00
4	0.71	0.00
5	0.92	0.56
6	0.86	0.47
7	0.72	0.00
8	0.71	0.00
9	0.92	0.58
10	0.71	0.00
Mean	0.79	0.16
Median	0.76	0.00
Max	0.92	0.58

The model was trained multiple times, and the performance metrics were recorded. The training accuracy was generally high, but the validation accuracy was low, indicating potential overfitting.

2.4 - Challenges

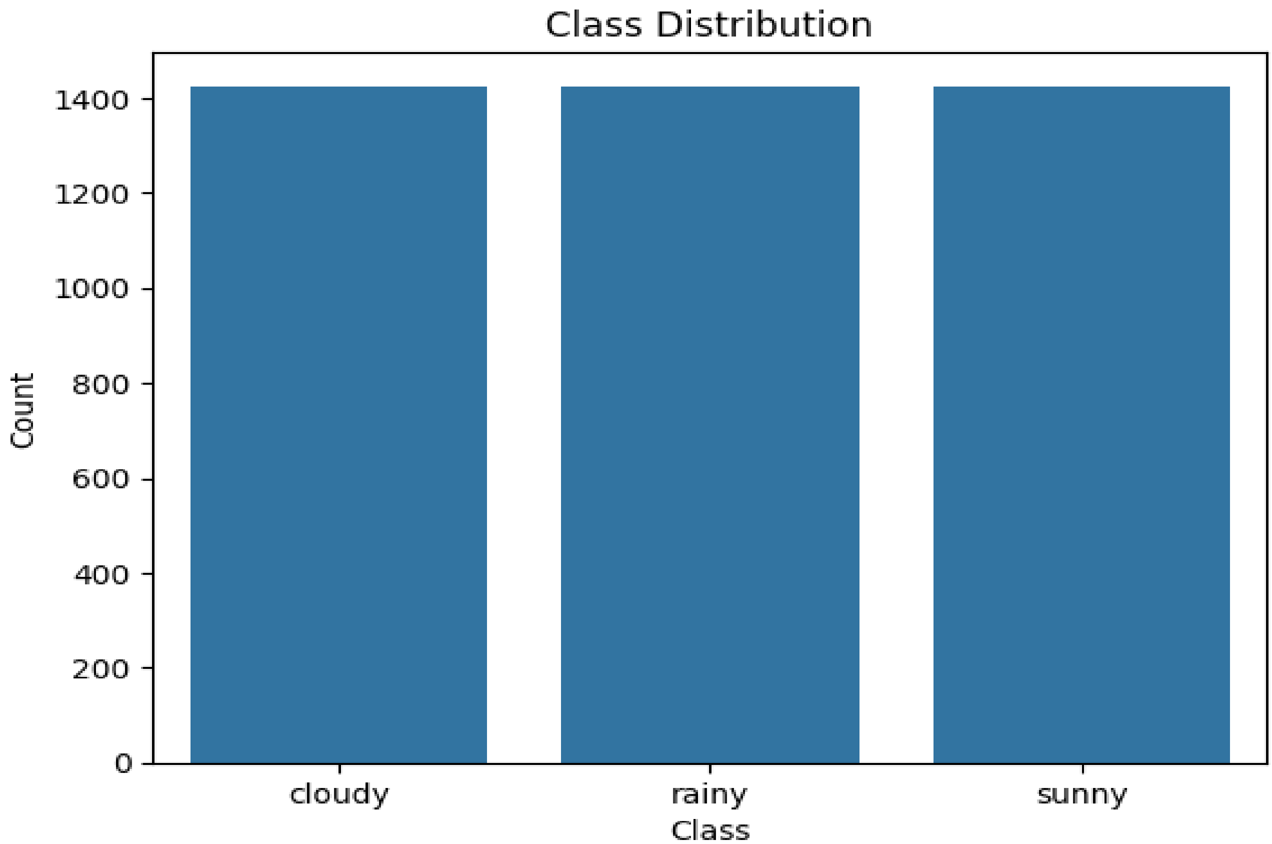
The primary challenge was the class imbalance, with a disproportionate number of images in each category. This imbalance likely contributed to the overfitting observed in the model, where the model performed well on the training data but poorly on the validation data.



3 - Data Augmentation and Model Optimization

3.1 - Data Augmentation

To address the class imbalance, we applied data augmentation techniques, including random rotation, zoom, and horizontal flips, to increase the number of training images in the minority classes. The augmented dataset was balanced, with each class having an equal number of images.



3.2 - Basic Model

The same CNN architecture was used to train on the augmented data. The performance metrics showed improved validation accuracy compared to the basic model, indicating that data augmentation helped in mitigating overfitting.

3.3 - Hyperparameter Tuning

We tuned the model by experimenting with different values of hyperparameters such as learning rate, batch size, and number of epochs. The following table summarizes the performance of models with various hyperparameter combinations:

Model Architecture	**Number of Trainable Parameters**	**Median Training Accuracy**	**Median Validation Accuracy**
Basic CNN	12,40,579	0.82	0.23
Tuned CNN	4,90,451	0.85	0.54

4 - Final Model

The optimal model had the following structure:

- Two convolutional layers with 32 and 64 filters respectively
- MaxPooling layers after each convolutional layer
- Dropout layers to prevent overfitting
- Dense layer with 128 units
- Output layer with softmax activation

The model was trained with a learning rate of **0.001** and achieved a testing accuracy of approximately **33.33%**.

5 - Way Forward

The final model showed improvement but still had limitations in accuracy on the testing data. Future work could involve:

- Further hyperparameter tuning
- Exploring deeper and more complex CNN architectures
- Implementing advanced data augmentation techniques
- Collecting more data to address class imbalance more effectively

Overall, the assignment provided valuable insights into the process of building and optimizing CNN models for image classification tasks.