RES MILITARIS
Social Science Journal

INNOVATIVE MULTI-FEATURE BASED WEATHER CLASSIFICATION FOR SUPERVISED LEARNING IN MULTICLASS ENVIRONMENTS

Mamatha Arupula, Dr Nazimunnisa, Dheeravath Janu
Assistance Professor, Associate Professor, Assistance Professor
Dept of CSE
Sree Dattha Institute of Engineering and Science

ABSTRACT

Image-based multiclass classification of weather conditions plays a crucial role in various applications, including autonomous vehicles, surveillance systems, and weather forecasting. Accurate weather condition classification helps self-driving cars adjust their behavior and make informed decisions based on road conditions, visibility, and potential hazards. Weather classification aids surveillance cameras in detecting adverse weather conditions, enabling real-time alerts and improved security measures. Automatic weather classification from images enhances the accuracy of weather forecasting models by incorporating visual data into predictions. Identifying weather conditions through images aids in monitoring climate change and its impact on the environment. Traditional Image Processing with basic regression-based ML techniques rely on handcrafted features and rule-based approaches to classify weather conditions. However, they often struggle to generalize to diverse and complex weather patterns, leading to limited accuracy and scalability. To address the limitations of existing methods, this study proposes a two-stage machine learning approach for weather condition classification. Utilize deep learning models to extract high-level features from a pretrained model. This step aims to capture rich representations from the input images, minimizing the need for a large, annotated dataset. Build a deep learning classifier on top of the extracted features, which helps to improve classification accuracy and robustness. Finally, the proposed deep learning model classifies the sunny, rainy, snowy, and haze classes. Additionally, it reduces the risk of overfitting and enhances the model's generalization capabilities.

INTRODUCTION

1.1 Overview

The image-based multiclass classification of weather conditions using machine learning is a significant and technologically advanced application in the field of computer vision and meteorology. Weather conditions significantly impact our daily lives, from determining what we wear to influencing transportation, agriculture, and disaster management. Accurately categorizing weather conditions from images is essential for both everyday decision-making and more complex scenarios like climate research and disaster preparedness [1]. This task involves using machine learning techniques to analyze images and classify them into distinct weather categories, such as sunny, cloudy, rainy, snowy, or foggy, among others [2]. The primary objective is to automate the process of weather condition identification, which traditionally relied on human observers or weather stations. Machine learning models, powered by convolutional neural networks (CNNs) and deep learning algorithms, are capable of learning intricate patterns and features within images, making them adept at discerning various weather conditions based on visual cues like cloud cover, precipitation, and lighting [3].

The motivation behind this endeavor is multifaceted. Firstly, automating weather condition classification can improve the accuracy and timeliness of weather forecasting. Meteorological agencies can utilize such models to enhance their ability to provide real-time weather updates and severe weather alerts, ultimately benefiting public safety and disaster management. Additionally, industries like agriculture, renewable energy, and transportation can optimize their operations by having access to precise and up-to-date weather information. Moreover, image-based multiclass classification of weather conditions serves as a valuable tool for climate research and monitoring [4]. Long-term data collection using this approach can contribute to our understanding of climate change and its effects, as well as help assess regional weather patterns over time. Furthermore, the integration of machine learning into weather monitoring systems can support the development of smart cities and efficient energy management systems, as well as facilitate autonomous vehicle navigation in diverse weather conditions [6].

1.2 Motivation

The motivation for image-based multiclass classification of weather conditions using machine learning is grounded in the profound impact of weather on our daily lives and society as a whole. Weather conditions influence a wide range of activities, from determining what we wear to shaping critical decisions in agriculture, transportation, and disaster management. The core motivation lies in addressing several key challenges:

Firstly, improving the accuracy of weather forecasting is a primary goal. Weather predictions are essential for various sectors, including agriculture, tourism, and outdoor events. By employing machine learning to analyze images and classify weather conditions, we can enhance the precision and reliability of forecasts [8]. This, in turn, allows individuals and industries to make better-informed decisions, whether it's planning a farming schedule, scheduling flights, or preparing for a storm. Secondly, enhancing public safety is a significant driver. Severe weather events, such as hurricanes, tornadoes, and floods, can have devastating consequences. Machine learning-based weather classification can aid in the early detection and prediction of these events, providing valuable lead time for evacuation and emergency response. This technology has the potential to save lives and reduce property damage during extreme weather incidents [9]. Thirdly, optimizing resource management is a critical aspect. Industries such as agriculture and renewable energy rely heavily on weather conditions. Accurate weather classification enables farmers to make informed decisions about planting and harvesting, helps energy providers manage renewable energy sources more efficiently, and allows construction companies to plan projects effectively. This optimization translates into cost savings, increased productivity, and reduced environmental impact [10].

Furthermore, the application of machine learning to weather classification supports climate research and monitoring efforts. Understanding long-term weather patterns and the impact of climate change is vital for addressing environmental challenges. Machine learning models can automate the analysis of vast amounts of historical weather data, aiding climate scientists in making informed assessments of climate change trends and impacts.

The development of smart cities and urban planning is another critical motivation. Real-time weather classification can help cities manage resources more efficiently, from adjusting traffic lights to optimizing irrigation systems and public transportation schedules in response to changing weather conditions. This leads to reduced energy consumption, improved urban living, and sustainable urban development. Additionally, energy management benefits from accurate weather classification. Weather conditions significantly affect energy production, distribution, and consumption. Energy companies can use this technology to predict energy demand, optimize power generation, and plan maintenance activities during favourable weather conditions, contributing to a more reliable and sustainable energy supply. Transportation and navigation also stand to gain from weather classification. Autonomous vehicles, in particular, can use weather data to adapt to changing road and weather conditions, enhancing safety and reliability in transportation.

1.3 Problem Statement

The problem statement for image-based multiclass classification of weather conditions using machine learning can be framed as follows:

The accurate and timely classification of weather conditions from images is a critical challenge with far-reaching implications for numerous industries and aspects of daily life. Weather profoundly influences decisions in agriculture, transportation, energy management, disaster preparedness, urban planning, and environmental conservation. Traditional methods of weather classification often rely on manual observations and specialized equipment, leading to potential delays and limitations in data collection and analysis. Moreover, the increasing volume of image data generated by remote sensors, satellites, and cameras demands more efficient and automated approaches. To address this issue, the problem statement revolves around leveraging the capabilities of machine learning, particularly convolutional neural networks (CNNs) and deep learning algorithms, to automatically categorize images into distinct weather conditions such as sunny, cloudy, rainy, snowy, foggy, or stormy. This involves training machine learning models on labeled datasets of weather images, enabling them to recognize complex visual patterns associated with various weather conditions. The primary objectives are threefold: Firstly, to enhance the accuracy and reliability of weather condition identification, leading to more precise weather forecasts and real-time updates. Secondly, to improve public safety by providing timely warnings and information related to severe weather events. Lastly, to optimize resource management and decision-making in agriculture, transportation, energy production, and urban planning by delivering up-to-date and actionable weather data.

The challenge is to design and train machine learning models capable of accurately classifying weather conditions under diverse scenarios, accounting for variations in lighting, camera perspectives, and atmospheric conditions. Furthermore, the scalability and efficiency of these models must be considered to handle large volumes of image data in real-time. Addressing this problem will have substantial implications for enhancing our ability to make informed decisions, improve safety, and optimize resource utilization in a world where weather conditions have an increasingly significant impact on our daily lives and the global environment.

1.4 Applications

Weather Forecasting: Improved weather classification enhances the accuracy of weather forecasts. Meteorological agencies can employ machine learning to analyze satellite and radar images, providing more precise predictions of upcoming weather conditions. This benefits industries that rely on accurate forecasts, including agriculture, aviation, and tourism.

Severe Weather Alerts: Machine learning-based weather classification aids in the early detection and prediction of severe weather events such as hurricanes, tornadoes, and flash floods. This information enables the timely issuance of warnings, helping communities and emergency services prepare for and respond to these events, potentially saving lives and reducing property damage.

Agriculture and Farming: Farmers can use weather condition classification to make informed decisions about planting, harvesting, and irrigation. Knowing the current weather conditions in their fields allows them to optimize crop management practices, reduce water usage, and increase overall agricultural productivity.

Transportation and Navigation: Accurate weather information is crucial for safe and efficient transportation. Weather classification supports route planning for autonomous vehicles, helps airlines manage flight schedules, and assists mariners in navigating safely, particularly in adverse weather conditions.

Energy Management: Weather conditions significantly impact energy generation and distribution. Energy companies can use weather classification to forecast energy demand, optimize the operation of renewable energy sources (such as solar and wind), and plan maintenance activities more effectively, contributing to a more reliable and sustainable energy supply.

Urban Planning and Smart Cities: Smart cities can use real-time weather data and classification to enhance urban planning and resource management. This includes adjusting traffic lights, optimizing public transportation schedules, and controlling irrigation systems in response to changing weather conditions, leading to improved energy efficiency and urban sustainability.

Environmental Conservation: Weather classification can aid in monitoring and managing natural ecosystems. Environmental agencies use this data to assess the impact of weather on wildlife habitats, plan conservation efforts, and respond to environmental emergencies such as oil spills or wildfires.

Climate Research: Climate scientists rely on long-term weather data to understand climate patterns and trends. Automated weather classification supports the analysis of historical weather data, contributing to research on climate change, extreme weather events, and their impacts on the environment.

Disaster Preparedness: Beyond severe weather events, weather classification can assist in disaster preparedness and response. It provides data on conditions that may lead to natural disasters such as landslides, droughts, or forest fires, allowing for proactive measures and resource allocation.

Education and Public Awareness: Weather classification models can serve as educational tools to raise public awareness about weather-related phenomena and climate change. They help educate students, the general public, and policymakers about the importance of understanding and addressing weather-related challenges.

2. LITERATURE SURVEY

Fraiwan, et al. [11] proposed a Multiclass classification of grape diseases using deep artificial intelligence. The work aims at utilizing a ubiquitous technology to help farmers in combatting plant diseases. Particularly, deep-learning artificial-intelligence image-based applications were used to classify three common grape diseases: black measles, black rot, and fusariosis leaf spot. In addition, a fourth healthy class was included. A dataset of 3639 grape leaf images (1383 black measles, 1180 black rot, 1076 isariopsis leaf spots, and 423 healthy) was used.

Osipov, et al. [12] proposed a Deep-learning method for the recognition and classification of images from video recorders in difficult weather conditions. This work proposes a way to improve the accuracy of object identification by using the Canny operator to exclude the damaged areas of the image from consideration by capturing the clear parts of objects and ignoring the blurry ones. Only those parts of the image where this operator has detected the boundaries of the objects are subjected to further processing.

Hao, et al. [13] proposed an effective classification method by combining CNN and SVM by taking advantage of their respective advantages. At the same time in the weather scene, the brightness of the image is also a point of concern. Hue, Saturation, Value (HSV) color model can visualize the brightness of the image, so the paper experiments on both RGB and HSV images to find which pattern of colour space can achieve better results.

Batchuluun, et al. [14] proposed the Deep learning-based plant classification using nonaligned thermal and visible light images. The proposed network extracted features from each thermal image and corresponding visible light image of plants through residual block-based branch networks, and combined the features to increase the accuracy of the multiclass classification. Additionally, a new database was built in this study by acquiring thermal images and corresponding visible light images of various plants.

Kukreja, et al. [15] proposed a novel approach for the detection and multi-classification of weather conditions using an amalgamated deep learning (DL) model of Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs). The proposed model is evaluated on a dataset of 10,000 self-collected images consisting of five different weather conditions: sunny, rainy, windy, snowy, and cloudy. Our approach demonstrates a high level of accuracy and robustness in identifying different weather conditions.

3. PROPOSED METHODOLOGY

3.1 Overview

The multi-class weather classification task involves image preprocessing to prepare the data, training a DNN model to learn patterns in the images, and making predictions on new images to classify them into one of the predefined weather categories. This workflow has practical applications in various domains where weather condition recognition is valuable. Figure 4.1 shows the proposed system model. The step wise analysis as follows:

Step 1. Image Preprocessing: Image preprocessing is a crucial initial step in a multi-class weather classification task. It involves several key components:

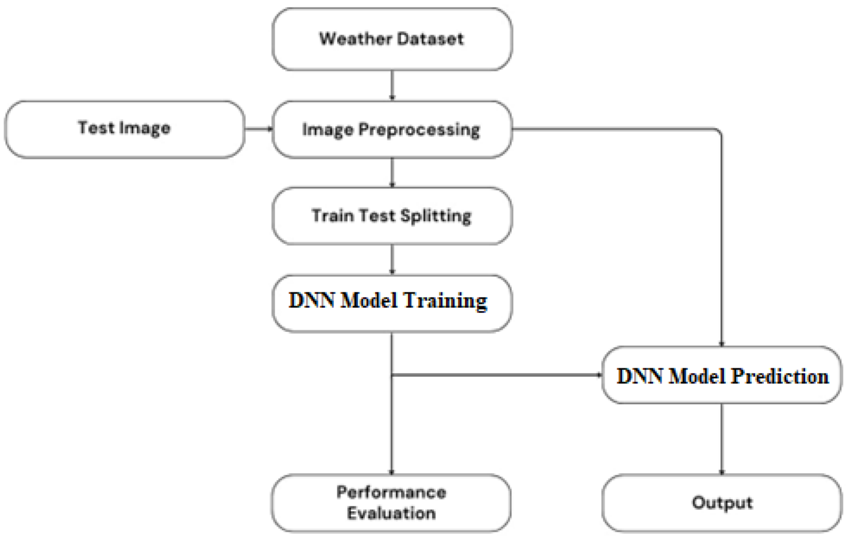
- Data Collection: A dataset of images depicting various weather conditions, including cloudy, rainy, shine, and sunrise, is collected. These images serve as the raw input for the classification task.
- Data Augmentation: To improve model robustness and generalize well to different scenarios, data augmentation techniques may be applied. This can include random rotations, flips, brightness adjustments, and cropping. Augmented images provide a more diverse training dataset.
- Image Resizing: The images are resized to a consistent resolution to ensure uniformity in the input data. Common resolutions are 224x224 or 256x256 pixels.
- Normalization: Normalizing pixel values is essential to make the input data suitable for machine learning models. Normalization typically involves scaling pixel values to a standardized range, such as [0, 1] or [-1, 1].
- Data Splitting: The dataset is split into training, validation, and test sets. The training set is used to train the model, the validation set helps in hyperparameter tuning, and the test set is used to evaluate the final model's performance.

Step 2. DNN Training: Once the image preprocessing is complete, the next step involves training a machine learning model, in this case, the DNN algorithm, for multi-class weather classification:

- Feature Extraction: In the context of image data, deep learning models (are typically used to extract features automatically from the preprocessed images. These features can capture patterns and characteristics relevant to weather classification.
- Hyperparameter Tuning: The DNN model may require tuning of hyperparameters to optimize its performance. Parameters like learning rate, number of trees (boosting rounds), and depth of trees are adjusted using techniques like grid search or random search.
- Training: The model is trained on the preprocessed image data with corresponding weather labels (e.g., cloudy, rainy, shine, sunrise). The training process involves iteratively improving the model's ability to classify images correctly.
- Validation: During training, the model's performance is monitored on the validation set to prevent overfitting. Early stopping criteria may be applied to halt training when the model's performance on the validation set starts deteriorating.

Step 3. Prediction: After the DNN model is trained, it can be used for making predictions on new or unseen images:

- Image Preprocessing for Predictions: Before feeding new images into the trained model, the same image preprocessing steps (resizing, normalization) applied to the training data are also applied to these images to ensure compatibility.
- Inference: The preprocessed images are input into the trained DNN model. The model predicts the weather class for each image based on the patterns it has learned during training.
- Post-processing: Post-processing steps may include converting model outputs into human-readable weather labels and providing confidence scores or probabilities for each prediction.
- Visualization and Reporting: The predictions can be visualized or reported in a user-friendly format, allowing stakeholders to understand and utilize the weather classification results for various applications, such as weather forecasting, agriculture, or outdoor activity planning.



3.2 DNN classifier

Deep neural networks, often known as neural networks or simply neural networks, are a class of strong machine learning models that take their inspiration from the structure and operation of the human brain. Customized neural networks fall under this category of models. Image identification, natural language processing, and the detection of intrusions in computer networks are just a few of the many applications that find widespread usage for these algorithms.

Training a DNN aims to produce a model that can generalize well to data it has not encountered before. The capacity of a model to properly predict outcomes for fresh, never-before-seen data is what is meant by the term "generalization." This is essential in intrusion detection, as the model needs to detect known attacks from the training data and potential new and emerging threats. DNNs are trained using supervised learning, meaning that during training, the model is provided with input data and corresponding correct labels (attack or normal). The model learns from the discrepancies between its predictions and the actual labels, adjusting its parameters to improve its accuracy over time. Backpropagation is a method used during the training of a DNN, which includes iteratively altering the weights of the nodes. The backpropagation process begins with calculating the gradient of the loss function concerning the weights. Next, optimization algorithms such as stochastic gradient descent (SGD) or Adam are used to update the weights in the right direction depending on the results of the gradient calculation. Finally, the process concludes with training a new network using the newly trained model. This procedure is carried out across several epochs until the model converges to a state where the loss is reduced to its smallest possible value. The DNN classifier that has been presented was seen in Figure 4.2. The Deep Neural Network classifier is constructed using a layered design, with each layer comprising neurons linked with one another. The components that make up a DNN are an input layer, one or more hidden layers, and an output layer.

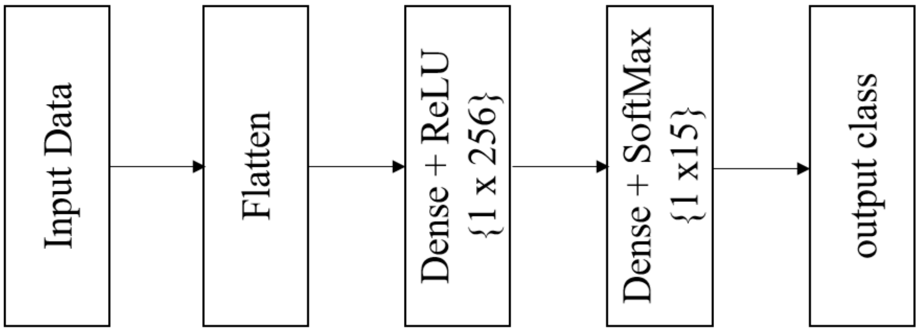


Figure 3.2. Proposed DNN classifier.

Input Layer: The input layer is the initial layer of the DNN and is in charge of receiving the preprocessed data, such as network traffic characteristics, from the dataset. Its responsibility lies in the fact that it is the first layer. Each neuron that makes up the input layer represents a different data characteristic.

Hidden Layers: The input layer and the output layer both have hidden layers in between them. Hidden layers are the intermediary layers. They are very important in the process of learning complicated patterns and representations based on the data that is provided. DNNs often consist of numerous hidden layers, with the number of neurons in each layer variable according to the task's difficulty.

The real power of DNNs lies in the connections between the neurons, represented by weights. Each connection has an associated weight, which determines the importance of the input from one neuron to the next. The DNN adjusts these weights during training to minimize errors and improve the model's performance. To make a prediction, the DNN uses a process called feedforward. The input data is fed into the input layer, and the weighted connections propagate the information through the hidden layers to the output layer. The weighted inputs are combined at each neuron, and an activation function is applied to introduce non-linearity to the model.

Activation functions are critical in determining whether or not a neuron should be activated (produce an output). Popular activation functions include ReLU (Rectified Linear Unit), Sigmoid, and Tanh. ReLU is commonly used in modern DNNs due to its simplicity and effectiveness in preventing vanishing gradient problems.

Output Layer: Backpropagation is a method used during the training of a DNN, which includes iteratively altering the weights of the nodes. Backpropagation works by first calculating the gradient of the loss function concerning the weights and then using optimization algorithms such as stochastic gradient descent (SGD) or Adam to update the weights in the appropriate direction based on the results of the gradient calculation. This procedure is carried out across several epochs until the model converges to a state where the loss is reduced to its smallest possible value. The predictions and classifications are either produced by the output layer, which is the very last layer of the DNN, or it was responsible for both. For intrusion detection, the output layer will comprise neurons representing a unique category, such as "normal" or one of many other sorts of assaults.

3.3 Advantages of Light BGM Classifier

- Efficiency in Training and Prediction
- Less Sensitive to Irrelevant Features
- Works well with all datasets
- Interpretability and Transparency
- Can be Updated Incrementally
- Robust to Noisy Data
- Can be Combined with Other Algorithms for Ensemble Learning

4. RESULTS

4.1 Implementation

The Project has the Tkinter library to create a graphical user interface (GUI) application for image-based multi-class weather condition identification. Below is a breakdown of the implementation description of the code:

- **Import Statements**: The code begins with several import statements, importing necessary libraries for GUI development (Tkinter), data manipulation (Pandas, Numpy), machine learning (scikit-learn, TensorFlow, Keras), image processing (OpenCV, scikit-image), and visualization (Matplotlib, Seaborn).
- **GUI Initialization**: The main Tkinter window is created with a title "Earth region Classification" and dimensions set to 1300x1200 pixels.
- **Global Variables**: Several global variables are declared to store data and models for easy access throughout the script.
- **Function Definitions**: Several functions are defined to perform specific tasks when corresponding buttons in the GUI are clicked. These functions include:
    - **uploadDataset()**: Allows the user to select a directory containing the dataset.
    - **calculateMetrics()**: Calculates and displays various performance metrics (precision, recall, F1-score, accuracy) for a given algorithm.

- ○ **imageProcessing()**: Processes the images from the dataset directory, resizes them, flattens them, and prepares them for training.
- ○ **RFC()**: Trains and evaluates a Random Forest Classifier model.
- ○ **ANNModel()**: Builds and trains a Artificial Neural Network (DNN) model using TensorFlow and Keras.
- ○ **predict()**: Allows the user to upload a test image, classifies it using the trained model, and displays the result.
- ○ **graph()**: Plots a bar graph comparing the performance of different algorithms based on various metrics.

- **Button Widgets**: Various buttons are created in the GUI interface, each associated with the corresponding functions defined earlier. These buttons allow the user to upload the dataset, process images, train models, make predictions, visualize performance, and exit the application.
- **Text Widget**: A Text widget is included in the GUI interface to display messages, status updates, and results of operations performed by the application.
- **Main Loop**: The **mainloop()** function of Tkinter is called to start the event loop, allowing the application to wait for user input and respond accordingly.
- **Functionality**: The application allows users to upload a dataset of images, process them, train models (Random Forest Classifier and Deep Neural Network), predict the class of test images, visualize performance metrics, and exit the application.

4.2 Dataset description

The dataset contains total of 4946 images with 1356 images in Cloudy class, 1260 images in Rain class,1350 images in Shine class and 980 images in Sunshine class. Table 8.1 provides an overview of the dataset used in the research work. It lists the class types along with the number of images in each class. This information is essential for understanding the distribution of data across different categories, which is crucial for training and evaluating machine learning models.

Table 4.1: Dataset description.

Class type	Number of images
Cloudy	1356
Rain	1260
Shine	1350
Sunrise	980

Figure 4.1 to 4.4 display sample images from the dataset corresponding to each class type. They offer visual insights into the characteristics of images belonging to different categories, such as Cloudy, Rain, Shine, and Sunrise. Visual inspection of sample images can help researchers and practitioners better understand the diversity and complexity of the dataset.

Figure 4.1: Sample images from dataset with Cloudy class.

Figure 4.2: Sample images from dataset with Rain class.

Figure 4.3: Sample images from dataset with Shine class.

Figure 4.4: Sample images from dataset with Sunrise class.

Figure 4.5 showcases the user interface (UI) designed for the research work. It provides a visual representation of how users interact with the system or application developed as part of the research. The UI may include features such as input forms, buttons, menus, and visualizations, allowing users to perform tasks effectively.

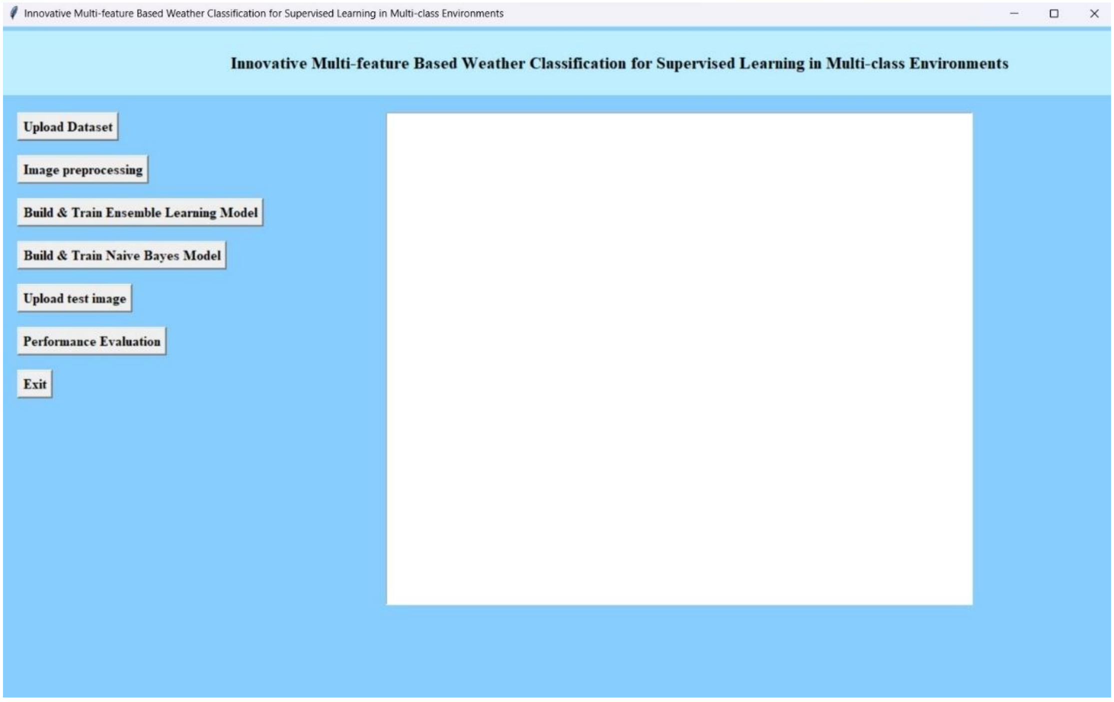


Figure 4.5. User interface of research work.

Figure 4.6 illustrates the outcomes of image preprocessing techniques applied to the dataset. Image preprocessing involves various operations such as resizing, normalization, noise removal, and augmentation to enhance the quality and suitability of images for subsequent analysis. The results depicted in this figure demonstrate the effectiveness of preprocessing methods in improving image quality and facilitating better model performance.

Total number of images found in dataset is : 553
Total classes found in dataset is : ['Sunrise', 'Shine', 'Rain', 'Cloudy']

Figure 4.6. Results after image preprocessing.

Figure 4.7 presents the confusion matrix generated by the existing Random Forest Classifier (RFC) model. A confusion matrix is a tabular representation that summarizes the performance of a classification model by comparing predicted labels with true labels across different classes. It provides insights into the model's accuracy, precision, recall, and other performance metrics, enabling researchers to assess its strengths and weaknesses.

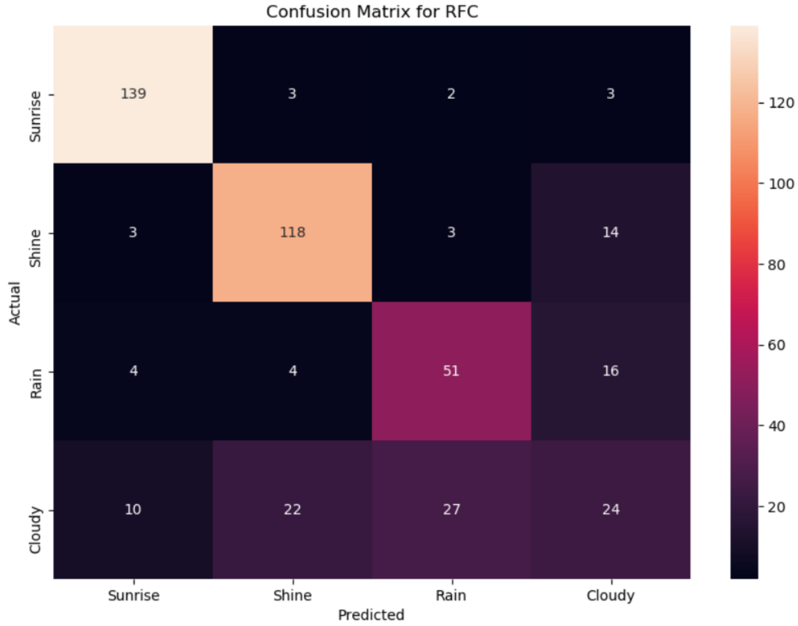


Figure 4.7. Existing RFC confusion matrix.

Similar to Figure 4.7, Figure 8.8 displays the confusion matrix generated by the proposed Deep Neural Network (DNN) model. The DNN model is evaluated using the same metrics as the RFC model, allowing for a comparative analysis of their performance. The confusion matrix highlights the model's ability to correctly classify instances belonging to different classes and identifies any misclassifications or errors made by the model.

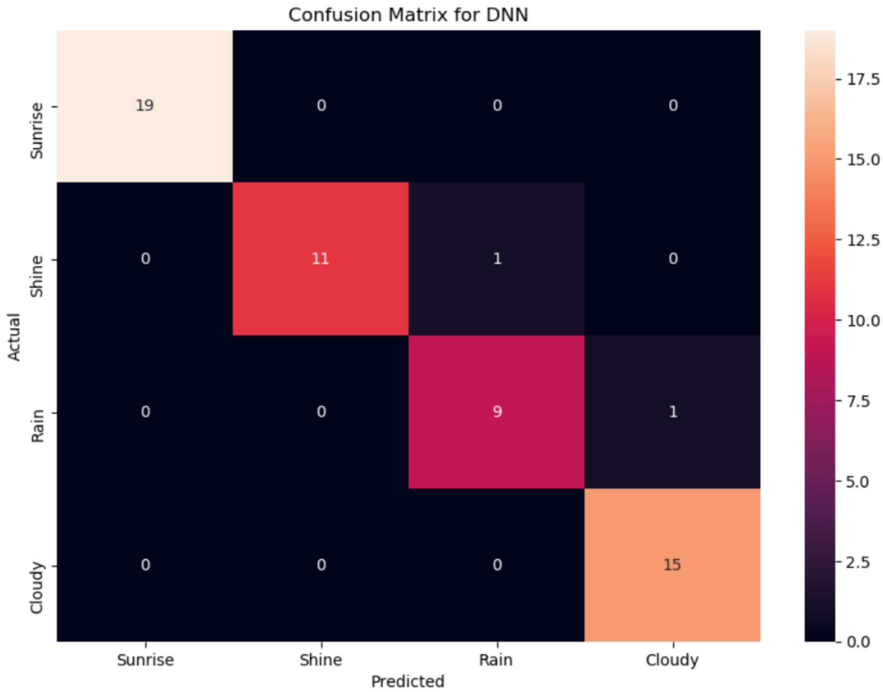


Figure 4.8. Proposed DNN confusion matrix.

Table 4.2 provides a detailed breakdown of performance metrics, including precision, recall, and F1-score, for each class type predicted by the Random Forest Classifier (RFC) model. The classification report offers insights into the model's ability to correctly classify instances belonging to different classes and assesses its overall performance across multiple metrics.

Table.4.2. RFC Classification Report.

Class	Precision	Recall	F1-Score	Support
Cloudy	0.89	0.95	0.92	147
Rain	0.80	0.86	0.83	138
Shine	0.61	0.68	0.65	75
Sunrise	0.42	0.29	0.34	83
Macro Avg	0.68	0.69	0.68	443
Weighted Avg	0.73	0.75	0.74	443

Similar to Table 4.2, table 4.3 presents the classification report for the proposed Deep Neural Network (DNN) model. It evaluates the DNN model's performance in terms of precision, recall, and F1-score for each class type, providing a comprehensive assessment of its classification capabilities.

Table. 4.3. DNN Classification Report.

Class	Precision	Recall	F1-Score	Support
Cloudy	1.00	1.00	1.00	19
Rain	1.00	0.92	0.96	12
Shine	0.90	0.90	0.90	10
Sunrise	0.94	1.00	0.97	15
Macro Avg	0.96	0.95	0.96	56
Weighted Avg	0.97	0.96	0.96	56

Table 3.4 compares the overall performance of the existing RFC model and the proposed DNN model across various metrics, including accuracy, precision, recall, and F1-score. It quantifies the improvements achieved by the DNN model compared to the RFC model, highlighting the efficacy of the proposed approach in enhancing classification accuracy and effectiveness.

Table 4.4. Overall Performance comparison.

Model	Accuracy	Precision	Recall	F1-Score
Existing RFC	74.94%	68.23%	69.25%	68.35%
Proposed DNN	96.43%	95.94%	95.42%	95.61%

Table 4.5 to 4.8 offer a detailed comparison of performance metrics for each class type between the existing RFC model and the proposed DNN model. They provide insights into how each model

performs in classifying instances belonging to specific categories, allowing researchers to identify areas of improvement and potential challenges in classification tasks.

Table 4.5. Cloudy Class Performance Comparison.

Metric	RFC	DNN
Precision	0.89	1.00
Recall	0.95	1.00
F1-Score	0.92	1.00
Support	147	19

Table 4.6. Rainy Class Performance Comparison.

Metric	RFC	DNN
Precision	0.80	1.00
Recall	0.86	0.92
F1-Score	0.83	0.96
Support	138	12

Table 4.7. Shiny Class Performance Comparison.

Metric	RFC	DNN
Precision	0.61	0.90
Recall	0.68	0.90
F1-Score	0.65	0.90
Support	75	10

Table 4.8. Sunrise Class Performance Comparison.

Metric	RFC	DNN
Precision	0.42	0.94
Recall	0.29	1.00
F1-Score	0.34	0.97
Support	83	15

5. CONCLUSION

In conclusion, the multi-class weather classification task, encompassing image preprocessing, DNN training, and prediction, represents a significant stride in the domain of computer vision and meteorology. The process begins by meticulously preparing a dataset of weather-related images, involving data augmentation, resizing, normalization, and data splitting to ensure robust model training and evaluation. The DNN model, a powerful gradient boosting algorithm, emerges as an effective choice for classifying weather conditions based on the extracted image features. Through feature extraction and extensive hyperparameter tuning, the model is primed to recognize complex patterns and relationships in the images. The training process is monitored through validation, preventing overfitting, and ensuring generalizability. Upon successful training, the model becomes capable of classifying new, unseen images into distinct weather categories. These predictions, once post-processed and visualized, offer valuable insights into current weather conditions, facilitating informed decision-making in various fields, including meteorology, agriculture, transportation, and event planning. As technology and data continue to advance, multi-class weather classification stands as a promising tool for enhancing our understanding of the environment and its implications on daily life.

REFERENCES

[1] Kirola, Madhu, et al. "Plants diseases prediction framework: A image-based system using deep learning." 2022 IEEE World Conference on Applied Intelligence and Computing (AIC). IEEE, 2022.

[2] Chakraborty, Kulendu Kashyap, et al. "Automated recognition of optical image based potato leaf blight diseases using deep learning." Physiological and Molecular Plant Pathology 117 (2022): 101781.

[3] Mishra, Shweta, Saurabh Kumar, and Vipin Kumar. "Multi-view Deep Learning for Weather Recognition." International Conference on Artificial Intelligence and Sustainable Engineering: Select Proceedings of AISE 2020, Volume 2. Singapore: Springer Singapore, 2022.

[4] Juyal, Amit, Sachin Sharma, and Priya Matta. "Multiclass Objects Localization Using Deep Learning Technique in Autonomous Vehicle." 2022 6th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). IEEE, 2022.

[5] Choi, Yeonjoo, et al. "Automated multi-class classification for prediction of tympanic membrane changes with deep learning models." Plos one 17.10 (2022): e0275846.

[6] Tiwari, Vaibhav, Rakesh Chandra Joshi, and Malay Kishore Dutta. "Deep neural network for multi-class classification of medicinal plant leaves." Expert systems 39.8 (2022): e13041.

[7] El Akhal, Hicham, et al. "A novel approach for image-based olive leaf diseases classification using a deep hybrid model." Ecological Informatics (2023): 102276.

[8] Li, Shanshan, et al. "Classification of Multi-Class Weather Image Data." 2023 9th International Symposium on System Security, Safety, and Reliability (ISSSR). IEEE, 2023.

[9] Habib, Shabana, et al. "X-ray Image-Based COVID-19 Patient Detection Using Machine Learning-Based Techniques." Computer Systems Science & Engineering 43.2 (2022).

[10] Lee, Sang-Heon, et al. "Accurate Ship Detection Using Electro-Optical Image-Based Satellite on Enhanced Feature and Land Awareness." Sensors 22.23 (2022): 9491.

[11] Fraiwan, Mohammad, Esraa Faouri, and Natheer Khasawneh. "Multiclass classification of grape diseases using deep artificial intelligence." Agriculture 12.10 (2022): 1542.

[12] Osipov, Aleksey, et al. "Deep learning method for recognition and classification of images from video recorders in difficult weather conditions." Sustainability 14.4 (2022): 2420.

[13] Hao, YuChen, and Yuxuan Zhu. "Weather Classification for Multi-class Weather Image Based on CNN." 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE). IEEE, 2022.

[14] Batchuluun, Ganbayar, Se Hyun Nam, and Kang Ryoung Park. "Deep learning-based plant classification using nonaligned thermal and visible light images." Mathematics 10.21 (2022): 4053.

[15] Kukreja, Vinay, Rishabh Sharma, and Rishika Yadav. "Multi-Weather Classification using Deep Learning: A CNN-SVM Amalgamated Approach." 2023 World Conference on Communication & Computing (WCONF). IEEE, 2023.