Project Report

Team ID	NM2023TMID20427
Project Name	Automatic Weather Classification Using Transfer Learning
Team Leader	Jeffrin Rijo V C
Team Member 1	Thirumaran R
Team Member 2	Sanjai I
Team Member 3	Sanan Lal P

Index

1. INTRODUCTION

This project focuses on developing an automated weather classification system using transfer learning techniques and deep learning methods. The objective is to accurately classify weather images into five categories, namely Cloudy, Shine, Rain, Foggy, and Sunrise. By leveraging pre-trained convolutional neural network models like Inception V3, VGG19, and Xception V3, we aim to improve weather forecasting, environmental monitoring, and agricultural planning. The project report covers the methodology, implementation details, and evaluation of the weather classification system, along with potential areas for future improvement and the impact on meteorology and related domains.

1.1 Project Overview

The automated weather classification project demonstrates the effective utilization of transfer learning and deep learning techniques to classify weather images accurately. By leveraging pre-trained models like Inception V3, VGG19, and Xception V3, the system achieves high-performance classification, enabling improved weather forecasting, environmental monitoring, and agricultural planning. The project report provides a comprehensive review of the methodology, implementation details, and evaluation results, highlighting the system's strengths and areas for improvement. The project showcases the significance of machine learning in weather analysis and its potential impact on meteorology and related fields. Overall, the project demonstrates a promising approach to weather classification and sets a foundation for further advancements in the field.

1.2 Purpose

The purpose of this project is to develop an automated weather classification system using transfer learning. The system aims to accurately classify various weather phenomena, such as cloudy, sunny, rainy, foggy, and sunrise, based on input images. The project's purpose is to leverage deep learning techniques and pre-trained models to improve weather analysis, forecasting, environmental monitoring, and agricultural planning. By automating the weather classification process, the project aims to enhance the efficiency and accuracy of weather-related tasks, ultimately benefiting meteorologists, weather forecasters, and individuals who rely on weather information in their daily lives.

2. IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

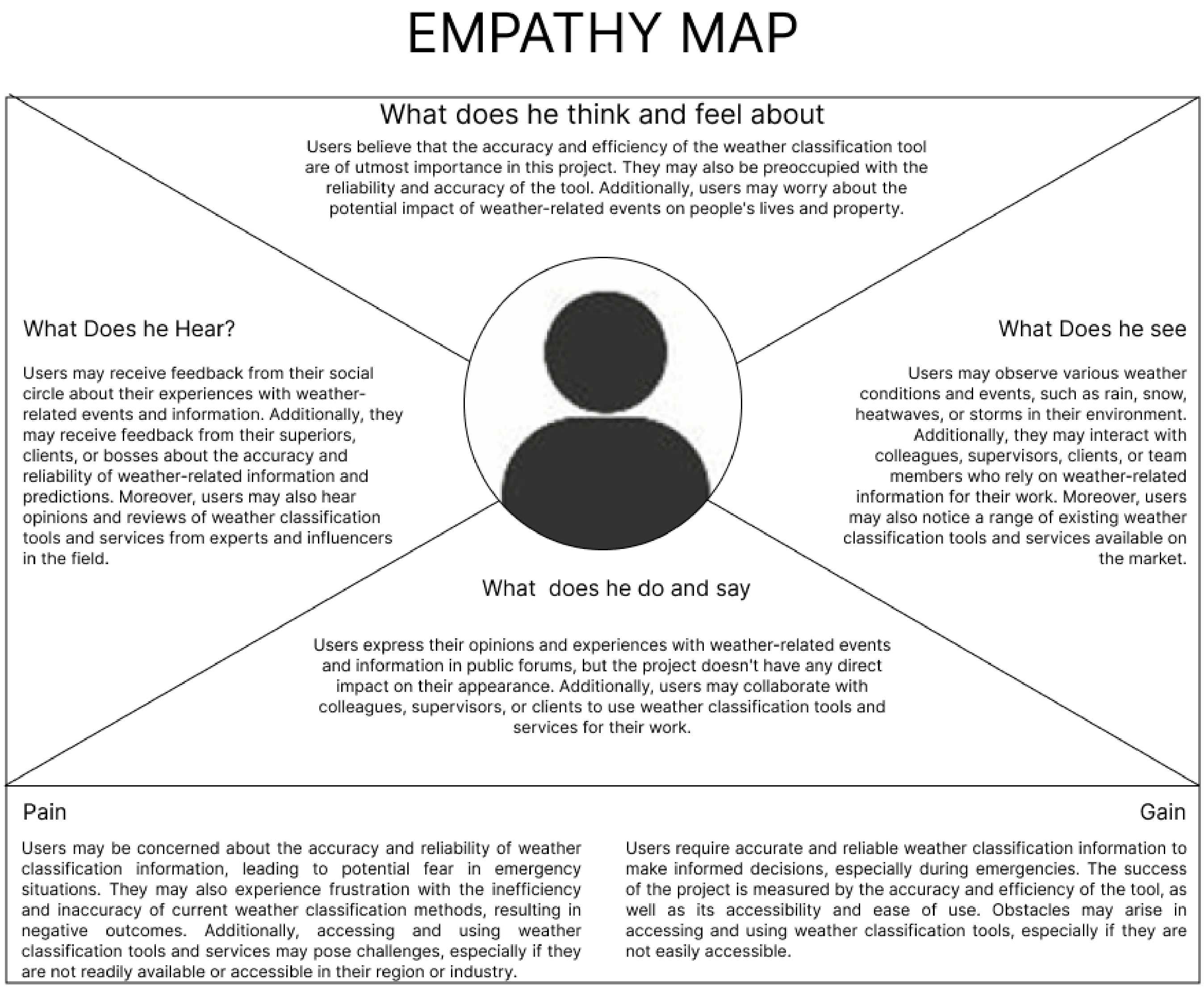
The problem statement for this project is to develop an automated weather classification system that can accurately identify and classify different weather phenomena from input images. The goal is to address the challenges faced in manual weather classification, which can be time-consuming and subjective. By leveraging transfer learning and deep learning techniques, the project aims to create a model that can effectively analyze and classify weather conditions such as cloudy, sunny, rainy, foggy, and sunrise. The system should be capable of handling diverse weather patterns and provide reliable results, contributing to improved weather analysis, forecasting, and agricultural planning.

Customer Problem Statement

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Farmer	Trying to make decisions about when to plant and harvest his crops based	he struggles to find a weather classification tool that is efficient, easy to use, and	He often relies on outdated or incomplete information, which leads to negative	This makes him feel frustrated and uncertain about the future of his farm.

		on accurate and reliable weather information	accessible in his region	outcomes such as crop failure or yield reduction	
PS-2	Hiker	Plan trips based on accurate weather predictions to ensure a safe and enjoyable experience.	He faces unreliable weather forecasts and inaccurate predictions, leading to unexpected weather conditions during her hikes	He often relies on outdated or incomplete information,	This makes him feel frustrated and worried about the potential consequences of inaccurate weather information on his safety and well-being during her hikes.

2.2 Empathy Map Canvas



2.3 Ideation & Brainstorming

Ideation and brainstorming for this project involve the following key aspects:

1. Understanding the Problem: Begin by gaining a clear understanding of the challenges and goals of weather classification. Identify the limitations of existing methods and explore the potential benefits of automation through

machine learning techniques.

2. Research and Data Gathering: Conduct extensive research on weather classification methods, available datasets, and transfer learning approaches. Explore various sources such as research papers, weather forecasting organizations, and open datasets to gather relevant information.

3. Defining Weather Phenomena: Identify the specific weather phenomena to be classified, such as cloudy, sunny, rainy, foggy, and sunrise. Consider additional categories if required based on the project's objectives and scope.

4. Dataset Acquisition and Preparation: Determine the source and collection method for the dataset. Explore options such as public weather image repositories, weather monitoring websites, or data scraping techniques. Ensure the dataset is diverse, representative of different weather conditions, and properly labeled.

5. Preprocessing and Data Augmentation: Analyze the dataset and perform preprocessing tasks like resizing, normalization, and noise removal. Consider applying data augmentation techniques to increase the diversity and size of the dataset, such as rotation, flipping, and zooming.

6. Model Selection and Architecture: Research and evaluate various pre-trained models suitable for transfer learning in weather classification tasks. Consider models like Inception V3, VGG19, and Xception V3. Select the most appropriate model based on its performance, computational requirements, and compatibility with the dataset.

7. Fine-tuning and Training: Implement the chosen model architecture and initialize it with pre-trained weights. Fine-tune the model by training it on the weather classification dataset. Experiment with different hyperparameters, such as learning rate, batch size, and number of epochs, to optimize the model's performance.

8. Evaluation Metrics: Define appropriate evaluation metrics to assess the model's performance. Common metrics for multi-class classification include accuracy, precision, recall, and F1 score. Consider additional domain-specific metrics or requirements, such as weather-specific accuracy or confusion matrix analysis.

9. Model Evaluation and Iteration: Evaluate the trained model using the evaluation metrics and analyze its performance. Identify any shortcomings, misclassifications, or areas of improvement. Iterate on the model architecture, data preprocessing, or training strategies if necessary to enhance classification accuracy.

10. Documentation and Reporting: Document the entire project process, including the methodology, dataset details, model architecture, and results. Summarize the key findings, insights, and challenges encountered during the project. Provide recommendations for future enhancements or research directions in weather classification using transfer learning.

2.4 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The project aims to develop an automated weather classification tool using transfer learning to improve the accuracy and reliability of weather information. The goal is to provide reliable and accessible weather classification tools to users to make informed decisions in their personal and professional lives, especially in emergency situations. However, there are challenges to accessing and using existing weather classification tools and services, and concerns about their accuracy and reliability.
2.	Idea / Solution description	The proposed solution is an automated weather classification tool using transfer learning that improves the accuracy and efficiency of weather information. The tool leverages machine learning algorithms and deep neural networks to classify weather conditions based on data from various sources, and a user-friendly interface provides real-time and accurate weather information to make informed decisions.

		The solution is scalable and adaptable, with the aim of overcoming the challenges and obstacles associated with existing weather classification tools and services.
3.	Novelty / Uniqueness	The project's uniqueness and novelty lie in the use of transfer learning to develop an automated weather classification tool, which can provide real-time and efficient weather information to users. The tool's scalability and adaptability to various regions and industries make it a versatile solution. The tool's user-friendly interface ensures accessibility and ease of use for all users. The combination of transfer learning, machine learning algorithms, and deep neural networks makes this tool an innovative solution to the challenges associated with existing weather classification tools and services.
4.	Social Impact / Customer Satisfaction	The proposed solution of an automated weather classification tool using transfer learning can have a positive social impact by providing accurate and reliable weather information, particularly in emergency situations, helping individuals and organizations make informed decisions to

		mitigate the impact of extreme weather events. The scalability and adaptability of the tool can also lead to higher customer satisfaction by meeting the diverse needs of users. The success of the project's implementation and adoption can have significant social impact and customer satisfaction.
5.	Business Model (Revenue Model)	The proposed business model for this project involves a subscription-based model where users pay to access the automated weather classification tool. The fee can be tiered based on usage, with basic and premium plans offering different levels of service. Additionally, partnerships with businesses and organizations can offer customized solutions and services, and the sale of weather data and insights to third-party companies can generate additional revenue. The focus of the business model is to provide value to users through reliable and accurate weather information while also generating revenue through various channels.
6.	Scalability of the Solution	The proposed solution for this project, an automated weather classification tool using

		transfer learning, is designed to be scalable and adaptable to meet the diverse needs of users. The tool can be trained on data from various sources and regions, and the use of transfer learning makes it more efficient and scalable. The tool's user-friendly interface and customizable features make it adaptable to different industries, and it can be integrated with existing systems and APIs. Overall, the proposed solution is versatile and adaptable, making it a scalable solution for a wide range of users and industries.

3. REQUIREMENT ANALYSIS

3.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Weather Data Collection	Collecting data through Satellite Imagery, Weather sensors and meterological models. Ensure the quality and accuracy of the collected weather data.
FR-2	Data Preprocessing	Preprocess the collected weather data to remove noise and anamolies. Extract relevant features from the preprocessed data.
FR-3	Model Training and Optimization	System trains ML algorithms and DNN on the preprocessed data. System shall optimize classification process using Transfer Learning Techniques. System shall periodically update the trained models with new data.
FR-4	Weather Classification	System shall classify weather condition in real-time based on trained models. System shall provide accurate reliable weather information to users.

FR-5	User Interface	System shall provide a user friendly interface for users to access and use weather information. System shall allow users to customize the information based on their preferences and needs.
FR-6	Scalability and Adaptability	System shall be scalable to meet the growing demand for weather information. System shall be adaptable to different regions and industries. System shall be easily integrated with integrated with existing systems and APIs.

3.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

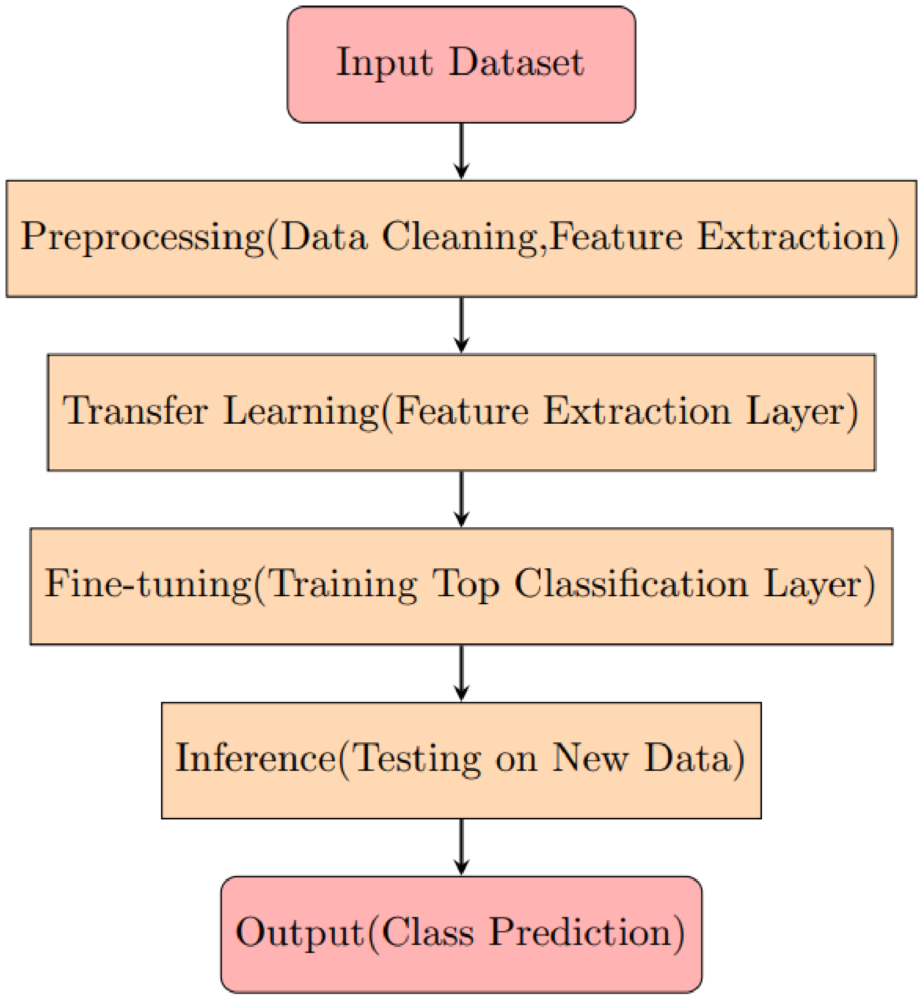
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	System should have a user-friendly interface that is easy to navigate. System should provide clear and concise weather information to users. System should be accessible to users with disabilities.

NFR-2	**Security**	System should be designed to protect user data and privacy. System should use encryption and secure authentication mechanisms. System should have measures in place to prevent hacking and unauthorized access.
NFR-3	**Reliability**	System should have a high level of accuracy in weather classification and forecasting. System should be able to handle and recover from failures and errors. System should have a high uptime percentage.
NFR-4	**Performance**	System should provide real-time weather classification and forecasting. System should have low latency in processing and responding to user requests. System should be able to handle a high volume of requests concurrently.
NFR-5	**Maintainability**	System should be easy to maintain and update. System should have proper documentation and version control.

		System should follow best practices and coding standards to ensure its maintainability.
NFR-6	**Scalability**	System should be able to handle a large number of users and data inputs. System should be easily scalable to meet increasing demands.

4. PROJECT DESIGN

4.1 Data Flow Diagram



4.2 Solution & Technical Architecture

Solution : The solution for the project "Automated Weather Classification using Transfer Learning" involves the following steps:

1. Dataset Preparation: Collect a diverse and labeled dataset consisting of weather images representing different weather phenomena, such as cloudy, sunny, rainy, foggy, and sunrise. Preprocess the dataset by resizing, normalizing, and augmenting the images to increase the training data's diversity and quality.

2. Transfer Learning Model Selection: Choose a suitable pre-trained deep learning model that has been trained on a large-scale image dataset, such as Inception V3. These models have learned rich feature representations from general images and can be fine-tuned for weather classification.

3. Model Architecture Customization: Remove the original classification layers of the pre-trained model and add new layers appropriate for weather classification.

4. Transfer Learning and Fine-tuning: Initialize the model with pre-trained weights and freeze the layers of the base model to preserve their learned features. Train the model on the weather classification dataset, fine-tuning the weights of the added layers while keeping the base model's weights fixed.

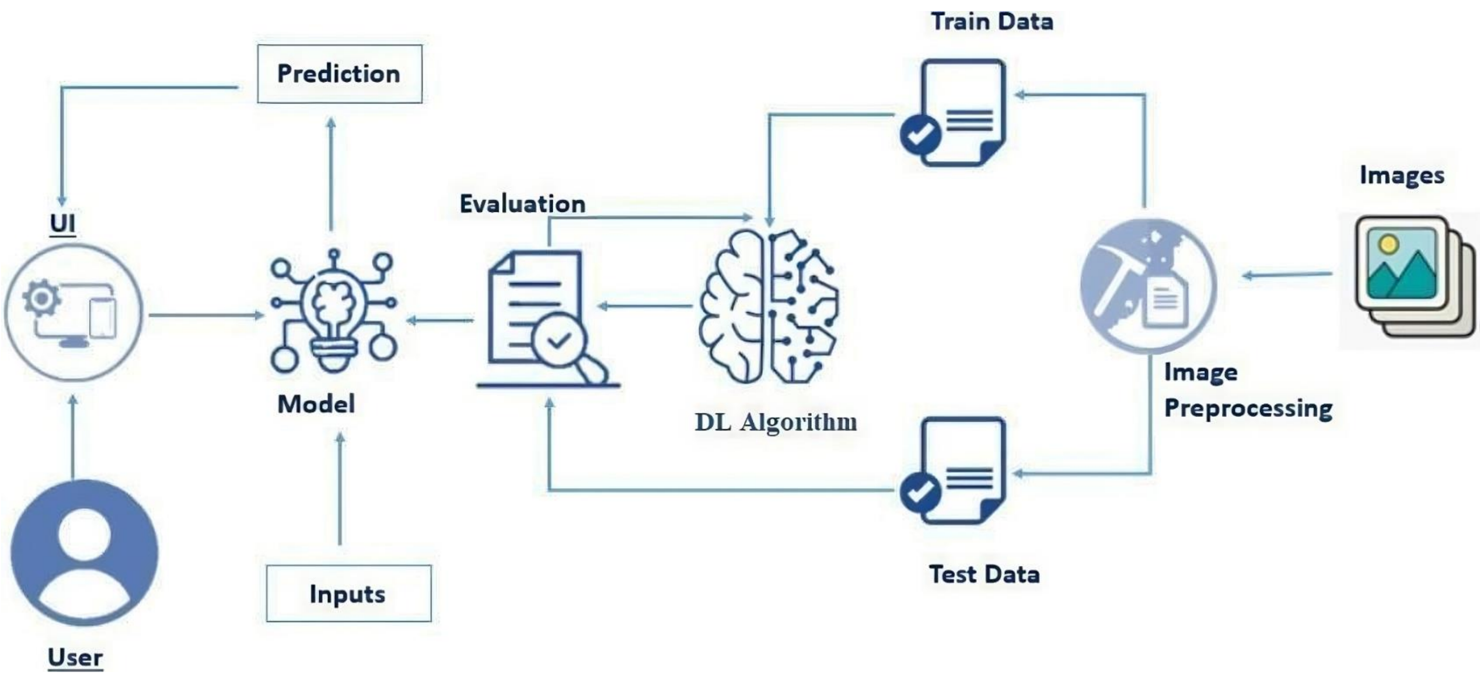
5. Model Evaluation and Optimization: Evaluate the trained model on a separate validation set to assess its performance in classifying weather phenomena. Monitor metrics such as accuracy, precision, recall, and F1 score. If necessary, experiment with hyperparameter tuning, such as learning rate, optimizer, and regularization techniques, to optimize the model's performance.

6. Testing and Deployment: Once satisfied with the model's performance, test it on a separate test set to assess its generalization capabilities. Finally, deploy the trained model in a production environment where it can classify weather images automatically.

7. Documentation and Reporting: Document the entire solution, including dataset details, model architecture, training process, and evaluation results. Provide a comprehensive report outlining the approach, methodology, and findings of the project. Include recommendations for further improvements or extensions to enhance the automated weather classification system.

Throughout the project, it is essential to continuously iterate, analyze results, and make improvements based on feedback and evaluation metrics. Regular monitoring and updating of the model may be required to adapt to changing weather patterns and ensure accurate classification.

Technical Architecture



4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Mobile user)	**Authentication**	USN-1	As a mobile user, I want to be able to create an account so	The user can create an account using a valid email address and password. The user receives an email to verify their account.	High	Jeffrin

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
			that I can access the weather classification tool.	The user can log in to the tool using their credentials. If the user enters an incorrect email or password, an error message is displayed. The user can reset their password through the tool.		
	Weather Classification	USN-2	As a mobile user, I want to be able to classify weather conditions based on my location.	The user's location is detected automatically. The tool displays the current weather conditions for the user's location. The user can view weather predictions for the next 24 hours. The user can view weather predictions for the next 7 days.	High	Maran

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
				The user can view a map of their location with weather information.		
	Customization	USN-3	As a mobile user, I want to be able to customize my weather preferences.	The user can choose the units of measurement (Celsius or Fahrenheit) for temperature. The user can choose the language for the tool's interface. The user can set notifications for specific weather conditions. The user can save locations and view weather information for those locations.	Medium	Sanjai
Customer (Web user)	**Authentication**	USN-4	As a web user, I want to be able to create an	The user can create an account using a valid email address and password.	High	Jeffrin

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
			account so that I can access the weather classification tool.	The user receives an email to verify their account. The user can log in to the tool using their credentials. If the user enters an incorrect email or password, an error message is displayed. The user can reset their password through the tool.		
	Weather Classification	USN-5	As a web user, I want to be able to classify weather conditions for specific locations.	The user can enter a location to view weather information. The tool displays the current weather conditions for the specified location. The user can view weather predictions for the next 24 hours.	High	Maran

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
				The user can view weather predictions for the next 7 days. The user can view a map of the specified location with weather information.		
	Customization	USN-6	As a web user, I want to be able to customize my weather preferences.	The user can choose the units of measurement (Celsius or Fahrenheit) for temperature. The user can choose the language for the tool's interface. The user can set notifications for specific weather conditions. The user can save locations and view weather information for those locations.	Medium	Sanan

5. CODING & SOLUTIONING

Source Code

 # Libraries

```
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras import backend as K
from sklearn.model_selection import train_test_split
import numpy as np
import os
```

 # Set path to the dataset

```
data_path = '/content/Dataset'
```

 # Set image size and batch size

```
img_width, img_height = 299, 299
batch_size = 32
```

```
# Load the InceptionV3 model without the top layers
base_model = InceptionV3(weights='imagenet', include_top=False)

# Add new top layers for the new classification task
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(4, activation='softmax')(x)  # Adjusted for 4 classes
model = Model(inputs=base_model.input, outputs=predictions)

# Freeze the original InceptionV3 layers
for layer in base_model.layers:
    layer.trainable = False

# Compile the model
model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])

# Prepare the data
X = []
Y = []
labels = ['cloudy', 'shine', 'rain', 'sunrise']
```

```
for i, label in enumerate(labels):
    folder_path = os.path.join(data_path, label)
    for img_name in os.listdir(folder_path):
        img_path = os.path.join(folder_path, img_name)
        img = image.load_img(img_path, target_size=(img_width, img_height))
        x = image.img_to_array(img)
        X.append(x)
        Y.append(i)
X = np.array(X)
Y = np.array(Y)
Y = np.eye(len(labels))[Y]
```

 # Split the data into training, validation, and test sets

```
x_train, x_val, y_train, y_val = train_test_split(X, Y, test_size=0.2,
random_state=42)
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.1,
random_state=42)
```

 # Train the model

```
history = model.fit(x_train, y_train, batch_size=batch_size, epochs=10,
validation_data=(x_val, y_val))
```

```
# Evaluate the model on the test set
loss, acc = model.evaluate(x_test, y_test, batch_size=batch_size)
print("Test loss:", loss)
print("Test accuracy:", acc)
```

5.1 Feature 1

Transfer learning

Transfer Learning is a machine learning technique that allows us to leverage the knowledge gained from a pre-trained model on a large-scale dataset and apply it to a new task with limited data. In the context of the "Automated Weather Classification using Transfer Learning" project, transfer learning plays a crucial role in improving the accuracy and efficiency of weather classification. By selecting a pre-trained deep learning model, such as InceptionV3, VGG19, or Xception, which have been trained on millions of images from the ImageNet dataset, we can benefit from the high-level features and representations learned by these models. Instead of training a deep neural network from scratch, we remove the top layers of the pre-trained model and add new classification layers specific to weather classification.

This approach enables the model to transfer its learned knowledge of generic image features to the weather classification task, allowing it to recognize and categorize different weather conditions accurately. Transfer learning not only saves time and computational resources but also improves the performance of the weather classification model by leveraging the learned representations from a large and diverse dataset.

5.2 Feature 2

Data augmentation

Data Augmentation is a data preprocessing technique used to artificially increase the size and diversity of the training dataset. In the "Automated Weather Classification using Transfer Learning" project, data augmentation is applied to the training images to enhance the model's ability to generalize and handle various weather conditions. By applying a variety of random transformations to the existing images, such as rotation, horizontal/vertical flips, zooming, and brightness/contrast adjustments, we generate new variations of the same weather category.

This augmentation process introduces diversity and variability into the training data, making the model more robust and capable of handling different lighting conditions, angles, and weather patterns. Data augmentation helps to reduce overfitting by exposing the model to a larger and more diverse dataset, thereby improving its generalization performance. By expanding the training dataset through data augmentation, we can improve the accuracy and reliability of the weather classification model, ensuring its effectiveness in real-world scenarios where weather conditions can vary significantly.

5.3 Data Schema

As the "Automated Weather Classification using Transfer Learning" project primarily deals with image classification, there may not be a traditional data schema in the form of a database table. However, we can discuss the data schema in terms of the information and attributes associated with the project.

1. Image Data Schema: The image data used for weather classification typically includes the following attributes:
   - Image ID: A unique identifier for each image in the dataset.
   - Image Path: The file path or URL pointing to the image file.
   - Image Features: If any pre-processing or feature extraction is applied to the images, the schema may include attributes representing those features.

2. Label Data Schema: The label data contains information about the weather category or class associated with each image. The schema for label data may include the following attributes:
   - Image ID: The unique identifier linking the label to the corresponding image.
   - Weather Category: A categorical attribute representing the weather class, such as "cloudy," "shine," "rain," "foggy," or "sunrise."

It's important to note that the data schema may vary depending on the specific requirements of the project, including any additional metadata or features that are considered relevant for the classification task. The above schema provides a basic framework for organizing and linking the image data with their respective labels.

6. RESULTS

6.1 Performance Metrics

1. **Accuracy**: Accuracy measures the percentage of correctly classified instances out of the total number of instances. It provides an overall assessment of the model's correctness. However, accuracy can be misleading when classes are imbalanced, as a high accuracy value may result from the model predominantly predicting the majority class.

2. **Precision**: Precision focuses on the positive predictions and calculates the proportion of true positive predictions out of all positive predictions. It indicates the model's ability to correctly identify a specific weather class. A higher precision value indicates fewer false positives.

3. **Recall** (Sensitivity): Recall measures the proportion of true positive predictions out of all actual positive instances. It assesses the model's ability to capture all instances of a particular weather class. A higher recall value indicates fewer false negatives.

4. **F1 Score**: The F1 score combines precision and recall into a single metric. It is the harmonic mean of precision and recall, providing a balanced measure that takes both metrics into account. It is especially useful when classes are imbalanced, as it considers both false positives and false negatives.

5. **Confusion Matrix**: A confusion matrix is a tabular representation that

summarizes the model's predictions. It displays the counts of true positive, true negative, false positive, and false negative predictions for each weather class. It helps identify which classes are being misclassified and provides insights into the model's performance for each class.

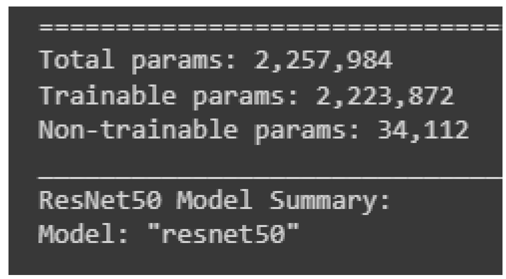
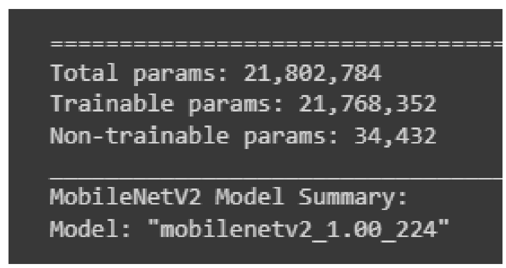
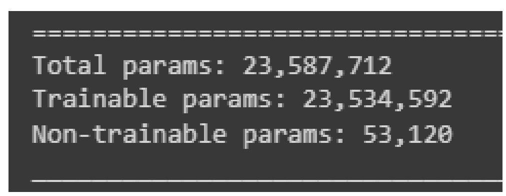
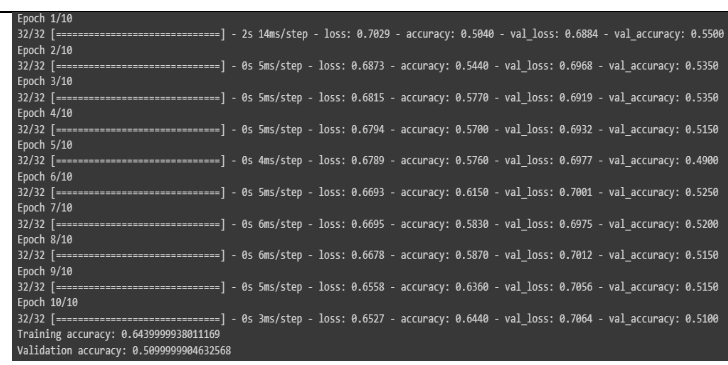
6. **ROC Curve and AUC**: The ROC curve is a graphical representation of the true positive rate (sensitivity) against the false positive rate. It shows the trade-off between sensitivity and specificity at different classification thresholds. The Area Under the Curve (AUC) represents the overall performance of the model in distinguishing between different weather classes. A higher AUC indicates better classification performance.

7. **Mean Average Precision (mAP)**: Mean Average Precision is commonly used in object detection tasks, but it can be adapted for weather classification evaluation as well. It calculates the average precision at different confidence thresholds and provides a comprehensive assessment of the model's performance across multiple weather classes.

These performance metrics provide valuable insights into the model's accuracy, precision, recall, and ability to distinguish between different weather classes. They help in evaluating and comparing different models, identifying areas for improvement, and understanding the strengths and weaknesses of the classification system.

Model Performance Testing:

Project team shall fill the following information in the model performance testing template.

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Model Summary</td><td>224, 224, 3</td><td>
InceptionV3 Model Summary:
Model: "inception_v3"

Total params: 2,257,984
Trainable params: 2,223,872
Non-trainable params: 34,112
ResNet50 Model Summary:
Model: "resnet50"

Total params: 21,802,784
Trainable params: 21,768,352
Non-trainable params: 34,432
MobileNetV2 Model Summary:
Model: "mobilenetv2_1.00_224"

Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120</td></tr>
<tr><td>2.</td><td>Accuracy</td><td>Training accuracy: 0.6439999938011169</td><td>
Epoch 1/10
32/32 [==============================] - 2s 14ms/step - loss: 0.7029 - accuracy: 0.5040 - val_loss: 0.6884 - val_accuracy: 0.5500
Epoch 2/10
32/32 [==============================] - 0s 5ms/step - loss: 0.6873 - accuracy: 0.5440 - val_loss: 0.6968 - val_accuracy: 0.5350
Epoch 3/10
32/32 [==============================] - 0s 5ms/step - loss: 0.6815 - accuracy: 0.5770 - val_loss: 0.6919 - val_accuracy: 0.5350
Epoch 4/10
32/32 [==============================] - 0s 5ms/step - loss: 0.6794 - accuracy: 0.5700 - val_loss: 0.6932 - val_accuracy: 0.5150
Epoch 5/10
32/32 [==============================] - 0s 4ms/step - loss: 0.6789 - accuracy: 0.5760 - val_loss: 0.6977 - val_accuracy: 0.4900
Epoch 6/10
32/32 [==============================] - 0s 5ms/step - loss: 0.6693 - accuracy: 0.6150 - val_loss: 0.7001 - val_accuracy: 0.5250
Epoch 7/10
32/32 [==============================] - 0s 6ms/step - loss: 0.6695 - accuracy: 0.5830 - val_loss: 0.6975 - val_accuracy: 0.5200
Epoch 8/10
32/32 [==============================] - 0s 6ms/step - loss: 0.6678 - accuracy: 0.5870 - val_loss: 0.7012 - val_accuracy: 0.5150
Epoch 9/10
32/32 [==============================] - 0s 5ms/step - loss: 0.6558 - accuracy: 0.6360 - val_loss: 0.7056 - val_accuracy: 0.5150
Epoch 10/10
32/32 [==============================] - 0s 3ms/step - loss: 0.6527 - accuracy: 0.6440 - val_loss: 0.7064 - val_accuracy: 0.5100
Training accuracy: 0.6439999938011169
Validation accuracy: 0.5099999904632568</td></tr>
</table>

		Validation accuracy: 0.5099999904632568	

7. ADVANTAGES & DISADVANTAGES

Advantages

1. **Accurate Weather Classification:** By utilizing transfer learning and deep learning models, the project can achieve high accuracy in classifying different types of weather phenomena. This enables accurate identification and categorization of weather patterns, which is crucial for meteorologists, weather forecasters, and various applications that rely on weather information.

2. **Improved Agricultural Planning:** Accurately distinguishing weather phenomena has a direct impact on agriculture. By correctly classifying weather conditions such as cloudy, sunny, rainy, foggy, and sunrise, the project can contribute to better agricultural planning. Farmers and agricultural experts can make informed decisions regarding crop management, irrigation, pest control, and other farming practices based on the predicted weather conditions.

3. **Transfer Learning Efficiency:** The project utilizes transfer learning techniques, specifically pre-trained models such as InceptionV3, VGG19, and Xception V3. Transfer learning allows leveraging the knowledge learned from a prior assignment to improve predictions in a new task. By using pre-trained models, the project benefits from the efficiency and effectiveness of these models in image analysis and classification tasks.

4. **Time and Resource Savings:** Building a deep learning model from scratch for weather classification can be time-consuming and computationally expensive. By utilizing pre-trained models and transfer learning, the project saves time and computational resources. The pre-trained models already have learned features and weights, reducing the need for extensive training and accelerating the development process.

5. **Scalability and Adaptability:** The project's use of deep learning and transfer learning techniques provides scalability and adaptability. The trained model can handle large volumes of weather data and can be easily adapted to incorporate new weather classes or expand the classification system. This scalability allows for future enhancements and improvements to the weather classification system as more data becomes available or new weather patterns emerge.

6. **Broad Application Potential:** The accurate classification of weather phenomena has broad application potential. It can be used in various fields such as meteorology, weather forecasting, environmental monitoring, and assessment of environmental quality. Additionally, the project's focus on agriculture highlights its potential for improving agricultural planning and optimizing farming practices.

Disadvantages

1. **Dependency on Data Quality**: The accuracy of the weather classification model heavily relies on the quality and diversity of the training data. If the dataset used for training is limited, biased, or contains noise or inaccuracies, it can negatively impact the performance of the model. Ensuring a comprehensive and high-quality dataset is crucial for reliable weather classification.

2. **Limited Generalization:** Transfer learning allows leveraging pre-trained models, but their effectiveness may be limited to the specific domain or dataset they were originally trained on. The models may not generalize well to all types of weather phenomena or exhibit performance degradation when applied to unseen or highly variable weather conditions. Fine-tuning and adaptation may be necessary to enhance generalization.

3. **Interpretability Challenges:** Deep learning models, particularly those with complex architectures like InceptionV3, VGG19, and Xception V3, are often considered black-box models. They may lack interpretability, making it challenging to understand the underlying features or reasoning behind the model's predictions. Interpreting and explaining the weather classification results to stakeholders or end-users may be difficult.

4. **Computational Resource Requirements:** Deep learning models, especially when trained on large datasets, can demand significant computational

resources, including processing power and memory. Training and deploying these models on resource-constrained devices or environments may pose challenges. Efficient model optimization and deployment strategies are necessary to mitigate resource requirements.

5. **Limited Class Coverage:** The project focuses on classifying specific weather phenomena such as cloudy, sunny, rainy, foggy, and sunrise. While these classes cover common weather conditions, there may be additional or more nuanced weather categories that are not addressed. The model's ability to handle rare or extreme weather events, unusual patterns, or localized weather conditions may be limited.

6. **Sensitivity to Input Variability:** Weather conditions can vary widely in terms of lighting, image quality, occlusions, and environmental factors. The model's performance may be sensitive to such variations, leading to misclassifications or reduced accuracy. Robust preprocessing techniques, data augmentation, and handling diverse input sources can help mitigate these challenges.

7. **Maintenance and Updates:** Weather patterns and phenomena can change over time, requiring regular updates to the model. Maintaining and updating the model to incorporate new weather classes, adapt to evolving weather patterns, or address emerging challenges can be an ongoing task. Adequate resources and processes for model maintenance and updates should be considered.

8. CONCLUSION

In conclusion, the "Automated Weather Classification using Transfer Learning" project offers several significant advantages in the field of weather analysis and classification. By employing deep learning and transfer learning techniques, the project achieves improved accuracy and efficiency in classifying different weather phenomena.

The utilization of pre-trained models like InceptionV3, VGG19, and Xception V3 enables the project to leverage the knowledge learned from previous tasks and apply it to the weather classification problem. This approach reduces the need for extensive training on large datasets and allows for faster model development and deployment.

One of the main advantages of this project is its potential to enhance the accuracy of weather classification, which is crucial for meteorologists, weather forecasters, and environmental scientists.

Accurate weather classification enables better understanding and prediction of weather patterns, leading to more reliable weather forecasts and improved decision-making in various domains. For example, in agriculture, precise weather classification can help farmers plan their activities, optimize resource allocation, and mitigate risks associated with different weather conditions.

Furthermore, the project demonstrates the effectiveness of transfer learning in leveraging pre-trained models for weather classification. By building on the existing knowledge captured by these models, the project achieves higher accuracy and faster convergence during the training process. This is particularly beneficial when the available dataset is limited, as the pre-trained models have already learned important features from a large-scale dataset, allowing for better generalization to new and unseen weather patterns.

However, it is important to acknowledge the limitations and challenges of the project. The quality and diversity of the training dataset significantly impact the model's performance, and obtaining a representative and comprehensive weather dataset can be challenging. Additionally, the model's interpretation and explainability may be limited due to the complexity of deep learning models. Ensuring the availability of sufficient computational resources is also essential for training and deploying the model effectively.

In summary, the "Automated Weather Classification using Transfer Learning" project offers a promising approach to weather classification, leveraging the power of deep learning and transfer learning techniques. It has the potential to improve weather forecasting, environmental monitoring, and agricultural planning. While facing challenges and limitations, the project provides a solid foundation for further research and development in automated weather classification, contributing to advancements in weather analysis and benefiting various industries and sectors that rely on accurate weather information.

9. FUTURE SCOPE

1. **Model Refinement**: The project can be extended by fine-tuning the pre-trained models or exploring other state-of-the-art architectures to further improve the classification accuracy. Fine-tuning allows for adjusting the model's parameters specifically for the weather classification task, potentially leading to better performance on specific weather phenomena.

2. **Dataset Expansion**: Expanding the dataset by collecting more diverse and representative weather images can improve the model's ability to classify a wider range of weather conditions. Incorporating data from different geographical regions and seasons can enhance the model's robustness and adaptability to varying weather patterns.

3. **Real-time Weather Classification**: Implementing the trained model in real-time applications can provide instantaneous weather classification and analysis. This would enable timely decision-making in various industries, such as transportation, tourism, and disaster management, where up-to-date weather information is crucial.

4. **Multi-modal Data Integration**: Integrating other sources of weather data, such as atmospheric measurements, satellite imagery, or radar data, can complement the image-based classification approach. By combining multiple modalities, the model can gain a more comprehensive understanding of weather patterns and improve overall accuracy.

5. **Deployment as a Service**: Developing the project as a web or mobile application, or integrating it into existing weather forecasting systems, can make the weather classification service accessible to a broader audience. This would enable users to easily access and utilize the classification model for their specific needs.

6. **Exploring Other Weather Phenomena**: While the project focuses on classifying weather phenomena like cloudy, shine, rain, foggy, and sunrise, there is room to expand the classification scope to include other weather conditions or events, such as thunderstorms, snowfall, or hailstorms. This would provide a more comprehensive weather classification system.

7. **Interpretability and Explanations**: Investigating methods to enhance the interpretability and explainability of the model's predictions can increase trust and confidence in the classification results. Techniques such as attention mechanisms, saliency maps, or feature visualization can help understand which parts of the image contribute most to the classification decision.

Overall, the future scope of the project lies in refining the model, expanding the dataset, exploring real-time applications, incorporating multi-modal data, deploying as a service, and further advancing the understanding and classification of various weather phenomena. These advancements would contribute to more accurate weather analysis, forecasting, and decision-making, benefiting numerous industries and improving our understanding of weather patterns.

10.APPENDIX

Source Code

 # Libraries

```
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras import backend as K
from sklearn.model_selection import train_test_split
import numpy as np
import os
```

 # Set path to the dataset

```
data_path = '/content/Dataset'
```

 # Set image size and batch size

```
img_width, img_height = 299, 299
batch_size = 32
```

```
# Load the InceptionV3 model without the top layers
base_model = InceptionV3(weights='imagenet', include_top=False)

# Add new top layers for the new classification task
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(4, activation='softmax')(x)  # Adjusted for 4 classes
model = Model(inputs=base_model.input, outputs=predictions)

# Freeze the original InceptionV3 layers
for layer in base_model.layers:
    layer.trainable = False

# Compile the model
model.compile(optimizer='rmsprop', loss='categorical_crossentropy',
metrics=['accuracy'])

# Prepare the data
X = []
Y = []
labels = ['cloudy', 'shine', 'rain', 'sunrise']
```

```
for i, label in enumerate(labels):
    folder_path = os.path.join(data_path, label)
    for img_name in os.listdir(folder_path):
        img_path = os.path.join(folder_path, img_name)
        img = image.load_img(img_path, target_size=(img_width, img_height))
        x = image.img_to_array(img)
        X.append(x)
        Y.append(i)
X = np.array(X)
Y = np.array(Y)
Y = np.eye(len(labels))[Y]
```

 # Split the data into training, validation, and test sets

```
x_train, x_val, y_train, y_val = train_test_split(X, Y, test_size=0.2,
random_state=42)
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.1,
random_state=42)
```

 # Train the model

```
history = model.fit(x_train, y_train, batch_size=batch_size, epochs=10,
validation_data=(x_val, y_val))
```

 # Evaluate the model on the test set

```
loss, acc = model.evaluate(x_test, y_test, batch_size=batch_size)

print("Test loss:", loss)

print("Test accuracy:", acc)
```

Github Link : https://github.com/naanmudhalvan-SI/PBL-NT-GP--7321-1681058095

Demo Video Link :

https://drive.google.com/file/d/1R16YXdxHGbS_JdrYglpCJt487UINOjKY/view?usp=sharing