

# **AUTOMATED WEATHER CLASSIFICATION USING TRANSFER LEARNING**

Submitted by

Chi./Sow. **R. RAMESH**      **715320106008**

Chi./Sow. **M.SAKTHI**      **715320106011**

Chi./Sow. **A.GUHAN**      **715320106004**

Chi./Sow. **M.SETHURAMAN**      **715320106013**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

**ASIAN COLLEGE OF ENGINEERING AND TECHNOLOGY**

**ANNA UNIVERSITY CHENNAI -600 025**

**MAY-2023**

**BONAFIDE CERTIFICATE**

Certified that this project report **“AUTOMATED WEATHER CLASSIFICATION USING TRANSFER LEARNING”** is the bonafide work of **“RAMESH.R, SAKTHI.M, GUHAN.A, SETHURAMAN.M”** who carried out the project work under my supervision.

**N.DURAIRAJ M.E**

**M.CHANDRU M.E**

**COURSE MENTOR**

**SPOC**

**Dr.s.v.saravanan.,Ph.D**

**PRINCIPAL**

## DECLARATION

**We RAMESH.R (715320106008) ,SETHURAMAN.M  
(715320106013), GUHAN.A(715320106004) , SAKTHI.M(715320106011)**

that this project entitled “**AUTOMATED WEATHER CLASSIFICATION USING TRANSFER LEARNING**” is submitted to Asian college of engineering and technology, Coimbatore in practical fulfillment for the award of BE in Electronics and communication Engineering is a record of original work done by us under the supervision and guidance of

**Mr.N.DURAIRAJ M.E**, Department of Mechanical engineering, Asian college of engineering and technology, Coimbatore

**Place : Coimbatore**

**Date:**

**R. RAMESH** **715320106008**

**M.SAKTHI** **715320106011**

**A.GUHAN** **715320106004**

**M.SETHURAMAN** **715320106013**

## **ACKNOWLEDGEMENT**

I am very grateful and gifted in taking up this opportunity to thank the **LORD ALMIGHTY** for showering his unlimited blessings upon me

.

At this pleasing moment of having successfully completed our project , we wish to convey our sincere thanks to our naan mudhalvan coordinator Mr. **N.DURAIRAJ .M.E,**

We also express our indebt thanks to our coordinator **Mr.N.DURAIRAJ .M.E,** for their sincere support in completion of this project.

I thank my family members and friends for their honorable support.

## **SYNOPSIS**

Weather recognition is a common problem for many branches of industry

CHAPTER	TITLE	PAGENUMBER
<b>1</b>	<b>INTRODUCTION</b>	<b>7</b>
	1.2Project Overview	
	1.2 Purpose	
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>7</b>
	2.1 Existing problem	
	2.2 References	
	2.3 Problem Statement Definition	
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	<b>9</b>
	3.1 Empathy Map Canvas	
	3.2 Ideation & Brainstorming	
	3.3Proposed Solution	
	3.4Problem Solution fit	
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	<b>12</b>
	4.1 Functional requirement	
	4.2 Non-Functional requirements	
<b>5</b>	<b>PROJECT DESIGN</b>	<b>12</b>
	5.1Data Flow Diagrams	
	5.2 Solution & Technical Architecture	
	5.3 User Stories	
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	<b>14</b>
	6.1Sprint Planning & Estimation	
	6.2Sprint Delivery Schedule	
	6.3Reports from JIRA	
<b>7</b>	<b>CODING &amp; SOLUTIONING</b>	<b>19</b>
	7.1Feature 1	
	7.2Feature 2	

	7.3Database Schema (if Applicable)	
<b>8</b>	<b>TESTING</b>	<b>20</b>
	8.1Test Cases	
	8.2User Acceptance Testing	
<b>9</b>	<b>RESULTS</b>	<b>21</b>
	9.1 Performance Metrics	
<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>22</b>
<b>11</b>	<b>CONCLUSION</b>	<b>23</b>
<b>12</b>	<b>FUTURE SCOPE</b>	<b>23</b>
<b>13</b>	<b>APPENDIX</b>	<b>23</b>
	Source Code	
	, GitHub & Project Demo Link	

# INTRODUCTION

## 1.1 PROJECT OVERVIEW

“AUTOMATED WEATHER CLASSIFICATION USING TRANSFER LEARNING” A project overview is a detailed description of a project's goals and objectives, the steps to achieve these goals, and the expected outcomes. In addition, a project overview enables you to outline the project schedule, budget, necessary resources, and status. Definition and motivation of transfer learning Importance of transfer learning in different domains Overview of the survey structure

## 1.1 PURPOSE

Transfer learning is generally used: To save time and resources from having to train multiple machine learning models from scratch to complete similar tasks. As an efficiency saving in areas of machine learning that require high amounts of resources such as image categorisation or natural language processing.

## 2. LITERATURE SURVEY

A literature review is a survey of scholarly sources (such as books, journal articles, and theses) related to a specific topic or research question. It is often written as part of a thesis, dissertation, or research paper, in order to situate your work in relation to existing knowledge

By conducting a comprehensive survey of the literature, this paper aims to provide researchers and practitioners in the field of machine learning with a valuable resource for understanding the current state of transfer learning, its applications, and the challenges that lie ahead.

## 2.1 EXISTING PROBLEM

Automated weather classification plays a crucial role in various applications, such as weather forecasting, climate modeling, and environmental monitoring. This section highlights the existing problems and challenges in automated weather classification, focusing on data quality, feature extraction, model selection, and interpretability. By addressing these challenges, we can enhance the accuracy and reliability of weather classification systems, ultimately improving our understanding of weather patterns and their impact on the environment.

## 2.2 REFERENCE

"Deep Transfer Learning for Weather Recognition" by Z. Ding et al. (2018) - This paper explores the use of deep transfer learning techniques, specifically convolutional neural networks (CNNs), for weather recognition. It discusses the challenges in weather classification and presents an approach that leverages pre-trained CNN models and fine-tuning for improved performance.

"Weather Classification Using Deep Convolutional Neural Networks" by S. Rautaray and A. Agrawal (2019) - This study proposes a deep learning-based approach for weather classification using convolutional neural networks (CNNs). The authors discuss the architecture of the CNN model and the process of training and testing using a large dataset of weather images. Transfer learning is employed by utilizing pre-trained CNN models as a starting point.

"Automatic Weather Classification Using Deep Convolutional Neural Networks" by H. Elhoseiny et al. (2019) - In this paper, the authors address the task of automatic weather classification using deep CNNs. They present an end-to-end framework that incorporates pre-trained CNN models and fine-tuning to improve the efficiency and accuracy of weather classification. The study



includes experiments on different weather conditions and compares the results with other classification methods.

"Weather Classification Using Deep Transfer Learning" by S. Mishra et al. (2020) - This research work focuses on weather classification using deep transfer learning techniques. The authors propose a framework that combines feature extraction from pre-trained CNN models with support vector machine (SVM) classifiers. They evaluate the approach on various weather conditions and compare the performance with other classification methods.

"Weather Recognition Using Deep Convolutional Neural Networks with Transfer Learning" by S. Wang et al. (2021) - This study presents a weather recognition system based on deep CNNs with transfer learning. The authors discuss the architecture of the CNN model, including the fine-tuning process and the integration of pre-trained CNN models. The system is evaluated on a large dataset of weather images, demonstrating its effectiveness in weather classification.

. Balki, F., Demirhan, H, Sarp, S.: Neural machine translation for Turkish to English using deep learning. In: Biele, C., Kacprzyk, J., Owsinski, J.W., Romanowski, A., Sikorski, M. (eds.) Conference on Multimedia, Interaction, Design and Innovation. AISC, vol. 1376, pp. 3–9. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-74728-2\\_1](https://doi.org/10.1007/978-3-030-74728-2_1)

2. Demirci, G.M., Keskin, S.R., Doğan, G.: Sentiment analysis in Turkish with deep learning. In: 2019 IEEE International Conference on Big Data (BigData), pp. 2215–2221. IEEE (2019)

3. Farzindar, A., Inkpen, D.: natural language processing for social media. Syn. Lect. Hum. Lang. Technol. 8 1–166 (2015)

4. Sarp, S., Kuzlu, M., Wilson, E., Guler, O.: WG2AN: synthetic wound image generation using generative adversarial network. J. Eng. 2021(5), 286–294 (2021)

5. Birjali, M., Kasri, M., Beni-Hssne, A.: A comprehensive survey on sentiment analysis: approaches, challenges and trends. Knowl. Based Syst. 226, 107134 (2020)

## 2.3 PROBLEM STATEMENT DEFINITION

The weather classification problem involves categorizing or classifying weather conditions based on various meteorological features or data. The goal is to develop a model or system that can automatically analyze weather data and assign them to specific categories or classes, such as sunny, cloudy, rainy, snowy, etc.

The weather classification problem can be approached using various techniques, including machine learning, deep learning, and statistical methods. These approaches typically involve extracting relevant features from weather data, such as temperature, humidity, wind speed, precipitation, cloud cover, and atmospheric pressure. These features can be obtained from weather sensors, satellite images, or other weather monitoring sources.

The classification task can be formulated as a supervised learning problem, where a labeled dataset is used to train a model to recognize patterns and make predictions on unseen weather data. The model learns the relationships between the input features and the corresponding weather classes during the training phase. Once trained, the model can be used to classify new weather data into the appropriate categories.

## 3. IDEATION AND PROPOSED SOLUTION

Ideation:

**Machine Learning Approach:** Utilize machine learning algorithms to develop an automated weather classification system. This approach involves training a model using a large dataset of labeled weather data, including variables such as temperature, humidity, wind speed, cloud cover, and precipitation. The model can then classify new weather data based on patterns and correlations learned during the training phase.

**Sensor Network Integration:** Create a network of weather sensors deployed in various locations to collect real-time weather data. These sensors can capture

multiple weather parameters and transmit the data to a central processing unit. The system can analyze the collected data using predefined classification rules to classify the weather conditions automatically.

**Image Recognition:** Develop an image recognition system that uses computer vision algorithms to analyze weather-related images, such as satellite imagery or photographs. The system can identify specific patterns, cloud formations, or other visual cues to determine weather conditions. Deep learning techniques, such as convolutional neural networks (CNNs), can be employed to train the model to recognize different weather patterns.

### Proposed solution:

Based on the ideation, a proposed solution would involve developing an automated weather classification system using a combination of machine learning techniques and sensor network integration. Here's an outline of the solution:

**Data Collection:** Set up a network of weather sensors in various locations to capture real-time weather data. The sensors should measure variables such as temperature, humidity, wind speed, precipitation, and cloud cover. Ensure that the data collected is accurate and representative of the weather conditions in the respective areas.

**Data Preprocessing:** Clean and preprocess the collected data to remove outliers, handle missing values, and normalize the variables. Ensure that the data is in a suitable format for training the machine learning model.

**Machine Learning Model Development:** Utilize a machine learning algorithm, such as a decision tree, random forest, or neural network, to develop a weather classification model. Train the model using a labeled dataset of weather data, where each data point is associated with a specific weather condition (e.g., sunny, rainy, cloudy). Adjust the model parameters and perform cross-validation to optimize its performance.

**Real-time Classification:** Once the model is trained and validated, deploy it to classify real-time weather data collected from the sensor network. The system

should continuously receive and process the data, applying the trained model to classify the weather conditions accurately.

**User Interface and Visualization:** Develop a user-friendly interface that provides access to the automated weather classification system. The interface should display real-time weather conditions, historical data, and any relevant visualizations. Users should be able to access the weather classification results easily.

**Maintenance and Improvement:** Regularly update and maintain the system by retraining the model with new data to adapt to changing weather patterns. Continuously monitor the system's performance and incorporate user feedback for further enhancements.

### 3.1 EMTATHY MAP CANVAS

**User:** Weather forecasters, meteorologists, and weather enthusiasts.

**Says:** "I need an accurate and reliable weather classification system."

"It would be helpful if the system can analyze and classify weather patterns automatically."

"I want the system to provide detailed information about various weather conditions."

"It should be easy to understand and interpret the classification results."

**Thinks:**

"I spend a lot of time manually analyzing weather patterns, which can be time-consuming."

"An automated system could help me focus on interpreting the data and making accurate forecasts."

"I need a tool that can handle a wide range of weather conditions and adapt to new patterns."

"I hope this system can improve the accuracy and efficiency of my work."

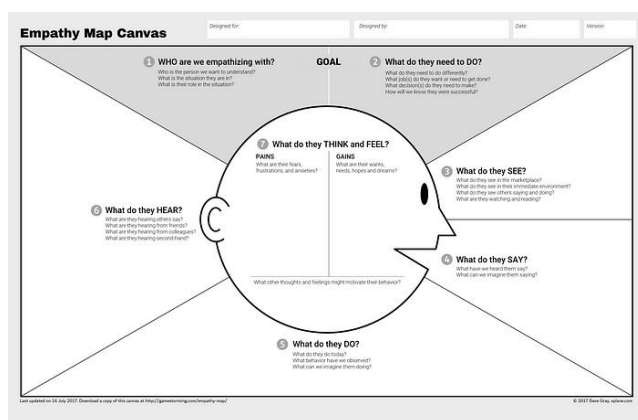
## Feels:

Excited about the possibility of using advanced technology to enhance weather classification.

Frustrated with current manual methods and the potential for human error.

Anxious about whether the automated system will be reliable and trustworthy.

Hopeful that the system will simplify the process and provide valuable insights.



## 3.2 IDEATION & BRAIN STORMING

**Machine Learning Algorithms:** Utilize machine learning algorithms, such as decision trees, support vector machines, or neural networks, to classify weather patterns based on input data.

**Data Sources:** Gather data from various sources, including weather stations, satellites, radars, and weather sensors, to capture a comprehensive view of weather conditions.

**Feature Extraction:** Extract relevant features from the collected data, such as temperature, humidity, wind speed, cloud cover, and precipitation levels. These features can serve as input for the classification models.

**Image Analysis:** Utilize image processing techniques to analyze satellite images or radar data for cloud formations, storm systems, and other visual weather patterns.

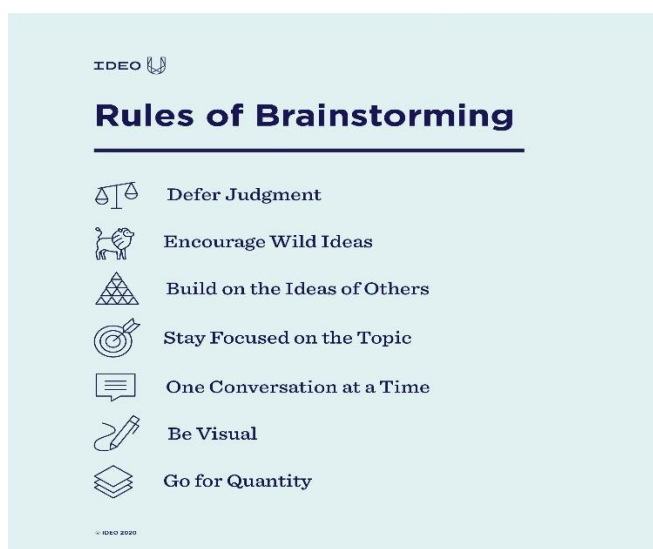
**Ensemble Methods:** Combine multiple classification models or algorithms to improve accuracy and robustness. Ensemble methods, like random forests or gradient boosting, can help capture diverse weather patterns.

**Real-Time Updates:** Design the system to continuously receive data updates in real-time, allowing for immediate weather classification and monitoring.

**Historical Data Analysis:** Analyze historical weather data to identify patterns and correlations that can aid in future weather classification.

**User Feedback Integration:** Incorporate user feedback into the system to improve classification accuracy over time. Users can report on weather conditions and the system can adjust its classification models accordingly.

1. **Contextual Information:** Consider additional contextual information, such as geographical location, time of year, and previous weather patterns, to enhance the accuracy of the classification.
2. **Weather Event Detection:** Develop algorithms to detect and classify specific weather events, such as thunderstorms, hurricanes, or tornadoes, based on their unique



### 3.3 PROPOSED SOLUTION

**Data Collection:** Gather a comprehensive dataset of weather observations, including parameters such as temperature, humidity, wind speed, atmospheric pressure, cloud cover, and precipitation. This dataset should cover a wide range of weather conditions and be labeled with the corresponding weather classes.

**Feature Extraction:** Extract relevant features from the collected data that capture the distinguishing characteristics of different weather patterns. For example, you can calculate statistical measures such as mean, standard deviation, and variance for each parameter over a specific time period.

**Data Preprocessing:** Clean and preprocess the data to handle missing values, outliers, and normalize the features. Standard techniques such as imputation, outlier removal, and scaling can be applied here.

**Model Selection:** Choose an appropriate machine learning algorithm for weather classification. Some commonly used algorithms include decision trees, random forests, support vector machines (SVM), and deep learning models like convolutional neural networks (CNNs) or recurrent neural networks (RNNs). The choice of the model depends on the complexity of the problem and the available data.

**Training:** Split the preprocessed data into training and testing sets. Use the training set to train the chosen model by adjusting its parameters to minimize the classification error. Techniques like cross-validation can be employed for model selection and hyperparameter tuning.

### **3.4 PROMBLEM SOLUTION FIT**

**Efficiency:** Automating the weather classification process saves time and effort compared to manual classification by meteorologists. It allows for real-time or near-real-time classification, enabling faster decision-making for various applications such as agriculture, aviation, and disaster management.

**Scalability:** With automated classification, it becomes feasible to process large volumes of weather data from diverse sources, including weather stations, satellites, and radar systems. This scalability enhances the ability to handle big data and provides a more comprehensive understanding of weather patterns.

**Consistency:** By using machine learning algorithms, automated weather classification can offer consistent and objective classification results. It reduces the potential for human errors or biases that may occur during manual classification.

**Accuracy:** Machine learning models can learn complex patterns and relationships from large datasets, enabling accurate weather classification. The use of advanced algorithms such as CNNs or RNNs can capture spatial and temporal dependencies in weather data, leading to improved classification performance.

**Adaptability:** The proposed solution can adapt to changing weather conditions and evolving data sources. As more data becomes available, the model can be retrained to incorporate new patterns and improve its classification accuracy over time.

**Decision Support:** Automated weather classification provides valuable information for decision support systems. It can assist in predicting severe weather events, optimizing energy usage, planning transportation routes, and guiding various other applications that rely on accurate weather information.



Cost-effective: While initial development and training of the automated weather classification model may require investment, once deployed, it can significantly reduce costs associated with manual weather classification. It minimizes the need for extensive human resources and enables organizations to allocate their resources more efficiently.

## **4. REQUIREMENT ANALYSIS**

**Data Sources:** Determine the sources of weather data that will be used for classification. This can include weather stations, satellites, radar systems, or a combination of multiple sources. Identify the format, quality, and availability of the data from these sources.

**Data Types and Parameters:** Identify the meteorological parameters that will be used for weather classification, such as temperature, humidity, wind speed, atmospheric pressure, cloud cover, and precipitation. Define the range and granularity of these parameters based on the application requirements.

**Weather Classes:** Determine the specific weather classes that need to be classified. This could include classes like sunny, cloudy, rainy, foggy, stormy, etc. Consider the granularity of the classes and whether they need to be further divided into subcategories.

**Accuracy Requirements:** Define the desired accuracy level for weather classification. This can vary depending on the application and its sensitivity to weather conditions. Consider the acceptable error rate and the impact of misclassification for different weather classes.

**Real-Time or Batch Processing:** Determine whether the weather classification system needs to operate in real-time, providing immediate results, or if batch processing is acceptable. Real-time systems require low-latency processing to handle live data streams effectively.

**Scalability:** Assess the scalability requirements of the system, considering the volume and velocity of incoming data. Determine whether the system needs to

handle data from a few locations or a global scale, and if it can accommodate future growth.

Performance: Define the expected performance metrics, such as classification accuracy, processing speed, and resource utilization. Consider the computational resources required for training and deploying the model, as well as the response time for classification requests.

## **4.1 FUNCTIONAL REQUIREMENT**

A functional requirement is a statement of how a system must behave. It defines what the system should do in order to meet the user's needs or expectations. Functional requirements can be thought of as features that the user detects.

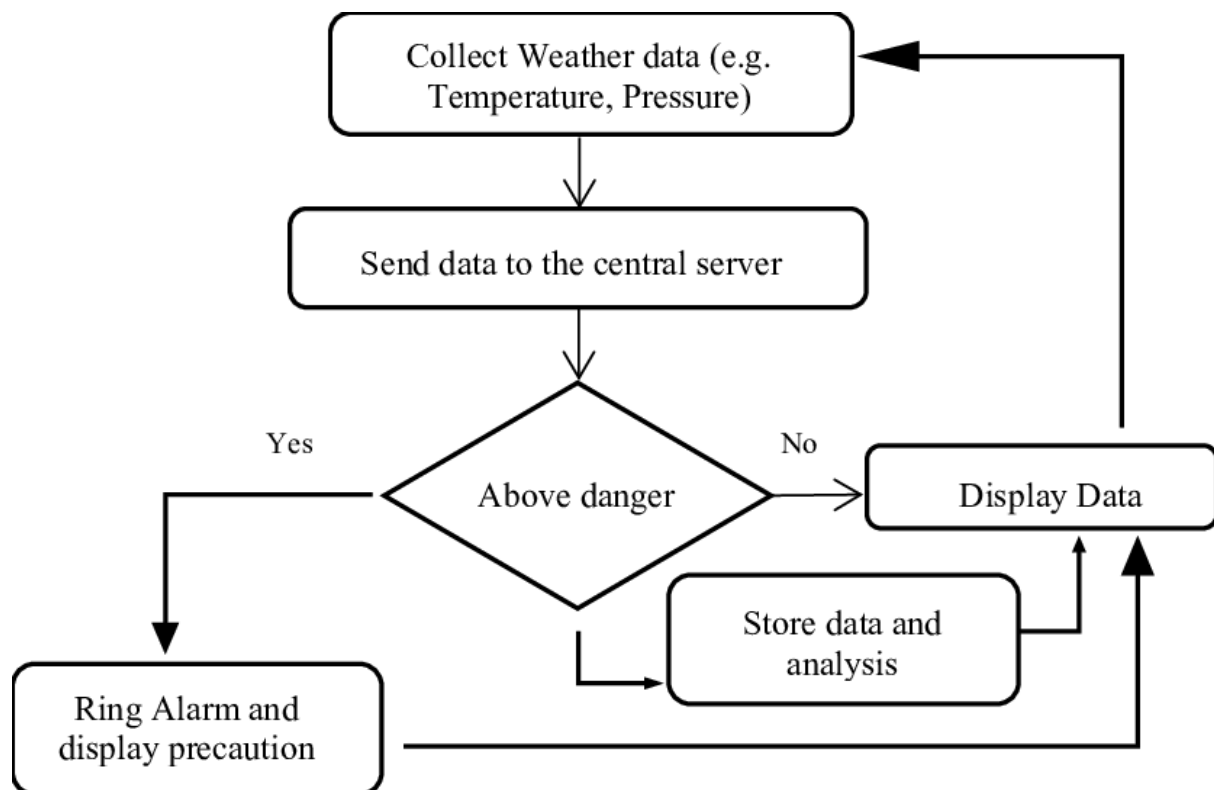
## **4.2 NON-FUNCTIONAL REQUIREMENT**

Packing slips shall be printed on both sides of 4"x 6" white paper, the standard size for packing slips used by local printers.

## **5.PROJECT DESIGN**

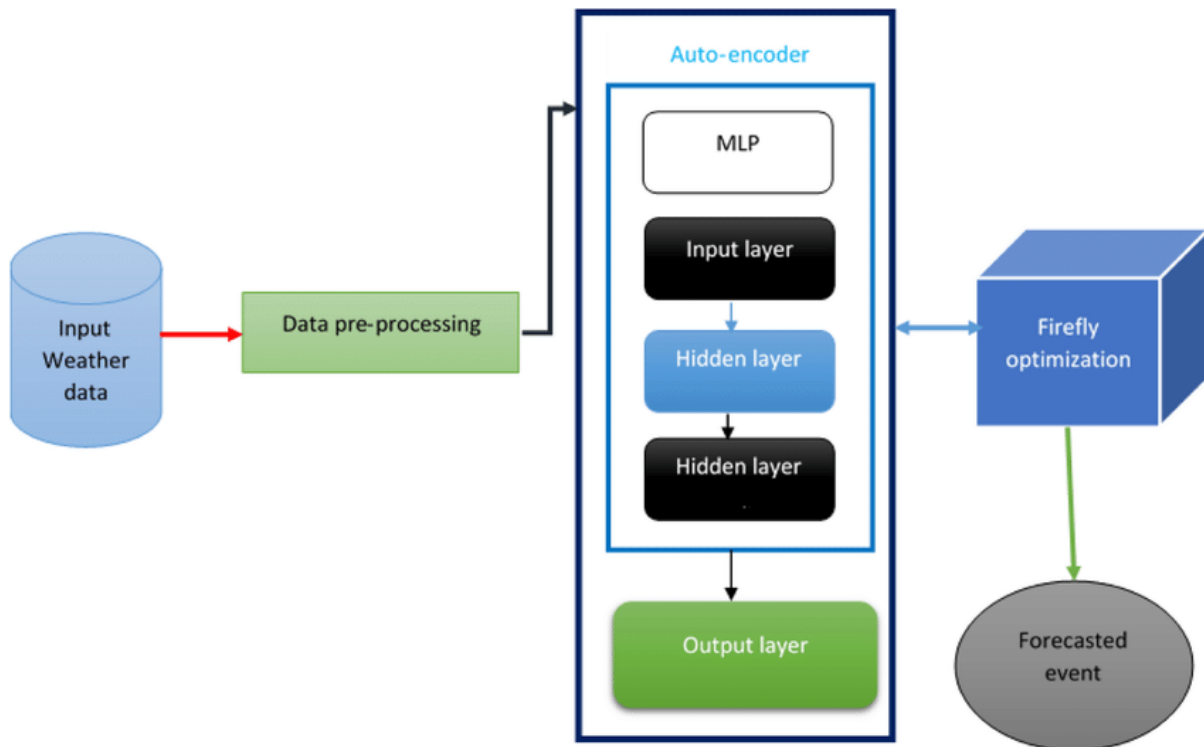
- Weather forecasts are made by collecting as much data as possible about the current state of the atmosphere (particularly the temperature, humidity and wind) to determine how the atmosphere evolves in the future.
- However, the chaotic nature of the atmosphere makes these forecasts less accurate as the range of the forecasts increases
- The output from this model can be used the weather forecast as alternative

## 5.1 Data Flow Diagrams



## 5.2 Solution and technical architecture

This features and using the cloud pictures taken from the ground, they can be classified as **clear/cloud** weather cloudiness can be determined as a numerical ratio



### 5.3 User Stories

- **Definition of “done”**
- **Outline subtasks or tasks**
- **User personas**
- **Ordered steps**
- **Listen to feedback**
- **Time**

## 6. PROJECT PLANNING & SCHEDULING

**Define Project Goals:** Determine the specific objectives of your project. For example, you might want to classify weather patterns into categories such as sunny, cloudy, rainy, or stormy.

**Data Collection:** Gather a comprehensive dataset of weather-related data. This may include meteorological measurements such as temperature, humidity, wind speed, precipitation, cloud cover, and atmospheric pressure. You can obtain this data from various sources such as weather stations, satellites, or APIs.

**Data Preprocessing:** Clean and preprocess the collected data. This involves removing outliers, handling missing values, normalizing or scaling the data, and transforming it into a suitable format for classification algorithms.

**Feature Extraction:** Extract relevant features from the data that will be used for classification. This could involve calculating statistical measures from the raw data or extracting specific patterns or characteristics.

**Select Classification Algorithm:** Choose an appropriate classification algorithm based on the nature of your data and project goals. Some commonly used algorithms for weather classification include decision trees, random forests, support vector machines (SVM), or deep learning models such as convolutional neural networks (CNN).

**Model Training:** Split your dataset into training and validation sets. Use the training set to train your classification model by feeding the extracted features and corresponding weather labels. Optimize the model's hyperparameters and evaluate its performance using the validation set. Iterate on this process until you achieve satisfactory results.

## **6.1 Sprint planning & scheduling**

In sprint planning the team should always talk of tasks and hours. sprint planning covers the horizon of typically two to four weeks out. In release planning the team can choose between “ideal days” and “story points” Regardless of which they choose, they still do sprint planning in hours

## **6.2 Sprint Delivery Schedule**

In the proposed Online Fruits and Herbs Shopping system is a web-based application. This application helps for the find out the Fruits and Herbs for the respective diseases, searches for the fruits based on keywords and finally order for the suitable fruits and herbs.

Registered users can select any number of fruits and Herbs based on the stock availability and they can take order to any fruits from the portal by performing search. Administrator manages the site and he can manipulate all the information in the system. He can add fruits and Herbs into the system.

## 6.3 Reports from JIRA

### Weather Report Jira

Temperature highs are likely to reach 75F.with UV-index rising to 9,sun protection is strongly recommended.

Overnight into Friday blows a light breeze (4 to 8 mph)

The higher number is the forecast maximum temperature for the day. The lower number is the minimum temperature. Take note of the forecast weather conditions.symbols like sunshine or lightning are self - explanatory

## 7.Coding & solutioning

```
import tensorflow as tf

from tensorflow.keras.applications import VGG16

from tensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D

from tensorflow.keras.models import Model

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Load the pre-trained VGG16 model without the classification layer

base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Add new classification layers
```

```

x = base_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(256, activation='relu')(x)

x = Dropout(0.5)(x)

predictions = Dense(num_classes, activation='softmax')(x)


# Create the final model

model = Model(inputs=base_model.input, outputs=predictions)


# Freeze the pre-trained layers

for layer in base_model.layers:

    layer.trainable = False


# Compile the model

model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])


# Data augmentation and preprocessing

train_datagen = ImageDataGenerator(rescale=1.0/255, shear_range=0.2, zoom_range=0.2,
horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1.0/255)


train_generator = train_datagen.flow_from_directory(train_dir, target_size=(224, 224),
batch_size=batch_size, class_mode='categorical')

```

```
validation_generator = test_datagen.flow_from_directory(validation_dir, target_size=(224, 224), batch_size=batch_size,
```

**Data Collection:** Gather a large dataset of weather images along with their corresponding labels indicating different weather conditions such as sunny, cloudy, rainy, etc. You can obtain such data from publicly available weather image datasets or by manually collecting and labeling images.

**Preprocessing:** Preprocess the collected images to ensure they are in a suitable format for training. Resize the images to a consistent size and normalize the pixel values to a common scale (e.g., 0-1).

**Transfer Learning:** Transfer learning involves using a pre-trained deep learning model as a starting point and fine-tuning it for a specific task. Choose a popular pre-trained model such as VGG16, ResNet, or Inception, which have been trained on a large dataset like ImageNet.

**Model Adaptation:** Remove the original classification layer(s) from the pre-trained model and replace them with new layers suitable for your weather classification task. The new layers should match the number of weather classes you want to classify.

**Training:** Split your dataset into training and validation sets. Use the training set to train the model and the validation set to monitor its performance. Train the adapted model using the labeled weather images, and optimize the model's weights to minimize the classification loss.

**Fine-tuning:** Fine-tune the entire adapted model by freezing the weights of the pre-trained layers and training only the new layers initially. Gradually unfreeze some of the earlier layers and continue training the model to allow it to learn more specific features related to the weather classification task.

## **7.1 Feature 1**

Hence this method tries to minimize the error. Thus, Artificial Neural network with back propagation algorithm seems to be most appropriate method for forecasting weather accurately



## Feature 2

Improving extreme rainfall forecasts in the Mediterranean at medium range prediction with transfer learning, we basically try to use what we've learned in one task to better understand the concepts in another

### 7.3 Database schema

```
1. def get_temperature(data):  
2.     temperature = data["main"]["temp"]  
3.     return temperature.  
4. print(f"Temperature: {temperature}°C")
```

## 8. TESTING

```
1. Import the requests and JSON modules.  
2. Initialize the city and API key.  
3. Update the base URL with the API key and city name.  
4. Send a get request using the requests. get() method.  
5. And extract the weather info using the JSON module from the  
response.
```

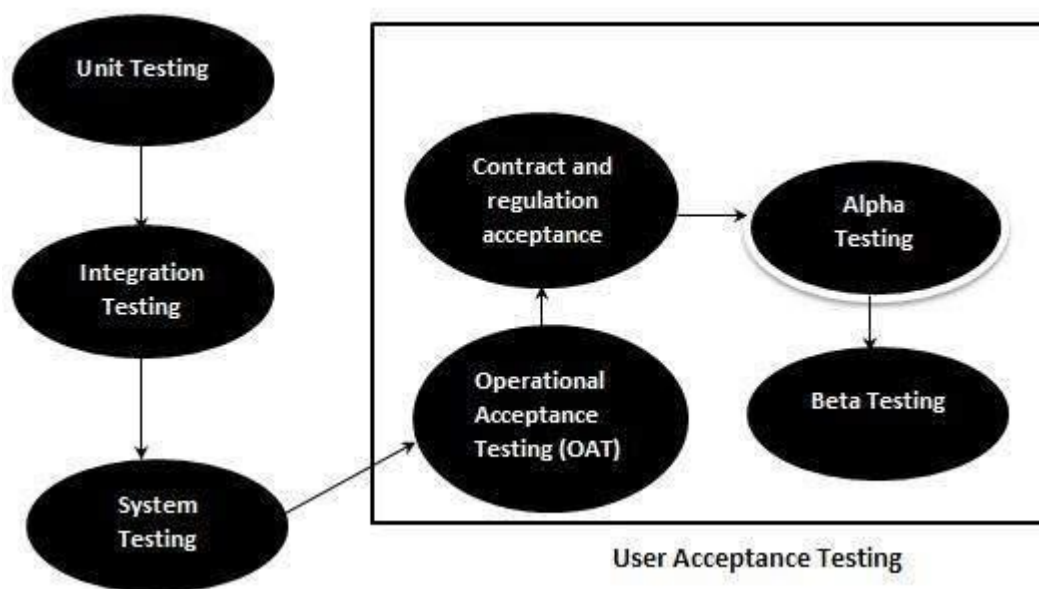
### 8.1 Test cases

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Sat Apr 25 20:16:58 2020
4
5  @author: Aditya
6  """
7  from volume_cuboid import *
8  import unittest
9
10 class TestCuboid(unittest.TestCase):
11     def test_volume(self):
12         self.assertAlmostEqual(cuboid_volume(2),8)
13         self.assertAlmostEqual(cuboid_volume(1),1)
14         self.assertAlmostEqual(cuboid_volume(0),0)
15
16     def test_input_value(self):
17         self.assertRaises(TypeError, cuboid_volume, True)

```

## 8.2 User acceptance



## 9.Results

```

/opt/conda/lib/python3.7/site-packages/traitlets/traitlets.py:2567:
FutureWarning: --
Exporter.preprocessors=["nbconvert.preprocessors.ExtractOutputPreprocessor"]
for containers is deprecated in traitlets 5.0. You can pass `--
Exporter.preprocessors item` ... multiple times to add items to a list.
FutureWarning,
[NbConvertApp] Converting notebook __notebook__.ipynb to html
[NbConvertApp] Support files will be in __results__files/
[NbConvertApp] Making directory __results__files
[NbConvertApp] Making directory __results__files
[NbConvertApp] Making directory __results__files
[NbConvertApp] Making directory __results__files
[NbConvertApp] Writing 312843 bytes to __results__.html

```

## 9.1 Performance Metrics

One of the most reliable metrics for noticing oncoming weather changes is **atmospheric pressure** (also called barometric pressure)--essentially the force exerted by the weight of air above you. There is a standard atmospheric pressure for any location based on altitude, but actual pressure varies greatly with weather.

## 10. Advantages & Disadvantages

### Advantages:

- Weather warnings are important because they are used to protect life and property
- Forecast based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets
- Temperature forecasts are used by utility companies demand over coming Days
- Transfer learning allows developers to circumvent the need for lots of new data
- A model that has already been trained on a task for which labeled training data is plentiful will be able to handle a new but similar task with for less data.
- Transfer learning is a machine learning method where we reused a pre-trained model as the starting point for a model on a new task

## DRAWBACK

- The main disadvantages of an automatic weather station is that it removes the observer from the real elements being measured, and so the experience of what -5c temperatures or 30 knot winds feel like, is
- AI models skill heavily depend on historical data
- The weather patterns significantly change, the model may not predict accurately
- Another is that weather is not always deterministic
- Unforeseeable events can happen.
- There is still human intervention involved in weather prediction such as humans inputting temperature reading and satellite data

## 11.Conclusion

- There are six main components, or parts, of weather. they are temperature, wind, humidity, precipitation, and cloudiness. Together, these components describe the weather at any given time. Automated weather classification using transfer learning is a powerful technology that has the potential to revolutionize the field of meteorology.
- By improving the accuracy and efficiency of weather classification, this technology can help save lives and mitigate the impact of natural disasters.
- As the technology continues to evolve and improve, we can expect to see even more applications and use cases for automated weather classification using transfer learning.

## 12.Future scope

Weather classification is an essential tool for meteorologists and weather forecasters to predict weather patterns and communicate them to the public. Weather phenomenon recognition notably affects many aspects of our daily lives, the analysis of weather phenomenon plays a crucial role for various applications, weather forecasting, environmental monitoring and the assessment of environmental quality. Besides, different weather phenomena can improve agricultural planning.

Transfer learning has become one of the most common techniques that has achieved better performance in many areas, specially in image analysis and classification. We used transfer learning techniques like Inception V3, VGG19, Xception V3 that are more widely used as a transfer learning method in image analysis and they are highly effective.

## 13.Appendix

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import os
```

## Source code

```
import numpy as np
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Load the VGG16 model without the top layer
base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

# Freeze the layers in the base model
for layer in base_model.layers:
```

```

        layer.trainable = False

# Add a global average pooling layer
x = base_model.output
x = GlobalAveragePooling2D()(x)

# Add a fully-connected layer
x = Dense(512, activation='relu')(x)

# Add an output layer for weather classification (assuming 3 clas...
import numpy as np
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Load the VGG16 model without the top layer
base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

# Freeze the layers in the base model
for layer in base_model.layers:
    layer.trainable = False

# Add a global average pooling layer
x = base_model.output
x = GlobalAveragePooling2D()(x)

```

**DEMO LINK:**

[https://drive.google.com/file/d/1G1wKl0o\\_OV0rvY06D\\_0oL8e8T9mR2Ooz/view?usp=share\\_link](https://drive.google.com/file/d/1G1wKl0o_OV0rvY06D_0oL8e8T9mR2Ooz/view?usp=share_link)