International Journal of Information Technology & Computer Engineering

ISSN 2347-3657

Volume 13, Issue 3, 2025

Detecting And Predicting Weather Using Cloud

T. Sudha Rani, K .Sai Srija, R. Shirisha

[1]Assistant Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women

[2.3]B,tech students, Department of Information Technology, Bhoj Reddy Engineering College for Women

ABSTRACT

Detecting and predicting weather ***using clouds begins with the collection of satellite or ground-based cloud images. These images, often obtained from meteorological satellites like INSAT, provide detailed visual data on cloud formations, density, color, and movement. The collected images undergo preprocessing steps such as normalization, resizing, and noise removal to ensure consistent input quality for analysis. The visual features of clouds—such as their shape, thickness, and color gradient—are critical indicators of weather conditions like rain, storms, or clear skies.***

Once the cloud images are preprocessed, machine learning techniques are applied to identify patterns and classify different cloud types associated with specific weather outcomes. Algorithms such as K-Means Clustering, Decision Trees, and Convolutional Neural Networks (CNNs) are used to analyze cloud structures. CNNs, in particular, are highly effective in image-based tasks due to their ability to capture spatial hierarchies. These models learn from historical weather data linked to cloud images, enabling them to predict future weather based on current cloud observations. The prediction output may include forecasts like rainfall probability, storm warning, or sunshine likelihood.

To make the system practical, real-time cloud data is continuously fed into the trained model through an integrated user interface or weather dashboard. This allows for continuous monitoring and rapid prediction updates, making the system valuable for agriculture, aviation, and disaster preparedness. The final output can be visualized or sent as alerts, helping users and authorities make informed decisions. By combining visual cloud analysis with predictive algorithms, this system enhances traditional meteorological forecasting methods and supports more accurate and timely weather predictions.

Keywords : Weather prediction ,Cloud detection , Machine learning , Satellite images , Convolutional Neural Networks (CNNs)

1. INTRODUCTION

Detecting and predicting weather using cloud analysis is a modern approach that leverages image processing and machine learning techniques. Weather patterns are closely associated with cloud formations, making them valuable indicators of upcoming conditions. Satellite images of clouds are collected and preprocessed to enhance clarity and consistency [1], [4]. These images are analyzed to detect cloud types, density, and movement. A Convolutional Neural Network (CNN) model is trained on cloud image datasets to recognize visual patterns linked to specific weather outcomes [2], [3]. The model can identify indicators of rain, storms, or clear skies based on cloud characteristics. Real-time cloud images are fed into the system during daily observations. The model provides predictive insights such as rainfall likelihood or storm alerts. This tool supports meteorologists and farmers with early warnings and accurate weather updates. It enhances traditional forecasting methods by adding AI-driven precision. The system acts as a supportive technology to assist in reliable and timely weather prediction.

Existing System:

The existing weather prediction systems mainly rely on numerical data like temperature, pressure, and humidity. They often overlook visual cloud patterns that signal upcoming weather changes. Traditional models may miss real-time visual cues captured through satellite imagery. There is limited use of image-based AI analysis in routine forecasts. As a result, short-term weather shifts due to cloud dynamics can be underpredicted or detected too late.

Proposed System:

TheThe proposed system uses cloud image analysis to assist in detecting and predicting weather conditions accurately. It employs a Convolutional Neural Network (CNN) to analyze satellite images of clouds in real time. Features such as cloud shape, density, and movement are identified as indicators of specific weather patterns. The system flags potential conditions like rain, storms, or clear skies based on visual cloud cues. It acts as a supportive forecasting tool alongside traditional meteorological methods . This approach enables early prediction and timely alerts for better weather preparedness.

2-RELATED WORK

Weather forecasting has traditionally depended on numerical weather prediction (NWP) models that analyze data such as temperature, pressure, humidity, and wind speed. While effective to an extent, these models often lack responsiveness to rapid changes in atmospheric conditions that are visually evident through cloud formations. As a result, researchers have explored the use of cloud image analysis to enhance short-term weather prediction accuracy. Recent studies have demonstrated that cloud types and structures provide significant cues about upcoming weather events. For instance, cumulonimbus clouds often signal thunderstorms, while cirrus clouds may indicate fair weather or incoming systems. With the advent of artificial intelligence (AI), particularly Convolutional Neural Networks (CNNs), it has become feasible to automate the detection and classification of cloud types using satellite imagery. CNNs have been successfully applied in several projects to classify cloud patterns and forecast related weather outcomes. Miao et al. used deep learning to distinguish cloud types and achieved higher accuracy than traditional image classification methods. The "Nowcasting" approach, supported by organizations like the UK Met Office, integrates high-resolution satellite images with AI models to predict rainfall within the next 1–3 hours. Similarly, Google's "SkyNet" project applies AI to real-time satellite data for short- term weather forecasts. Some research also focuses on integrating multimodal data, such as combining cloud imagery with ground sensor data (e.g., humidity and temperature), which improves forecasting reliability. However, these hybrid models also introduce computational complexity. To make AI-based weather forecasting more accessible, lightweight CNN-based systems are being proposed, capable of running on local servers or even edge devices.

3-REQUIREMENT ANALYSIS

Functional Requirements:

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract.

Functional requirements for Detecting and predicting weather using cloud using technology and machine learning include:

- *Image Acquisition*
  The system shall capture cloud images from satellites, drones, or ground-based cameras.
- *Image Processing*
  The system shall preprocess images (enhancement, noise removal, and segmentation).
- *Weather Prediction*
  System will analyse cloud patterns to predict short-term weather conditions.
- *Data Integration*

  The system shall integrate with external meteorological data sources.

Visualization & Reporting

The system shall display real-time cloud images and movement maps

3.2 Non-Functional Requirements:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

- *Security*
  Weather image data and prediction results must be securely transmitted and stored using encryption.
- *Reliability*
  The system must operate consistently without crashing or producing inconsistent results. Should handle invalid inputs gracefully with meaningful error messages.
- *Scalability*
  Scalability ensures that the weather prediction system can handle increasing workloads, data volume, and user demand without performance degradation.
- *Performance*
  Image processing and weather prediction must be done within a reasonable response time (e.g., <5 seconds).
- *Portability*
  Portability in the context of weather detection and prediction using cloud technologies refers to how easily the system can be transferred, deployed,

3.2.1 Software Requirements:

Blockchain Platform
: Ethereum , Hyperledger Fabric

Libraries
: OpenCV , NumPy , Pandas and Matplotlib

Front-End
: HTML, CSS , JavaScript

Backend
: Python 3.6.0

Development Tools
: Jupyter Notebook /Google Colab ,Postman

3.2.2 Hardware Requirements:

Processor
: I5 Core
Ram : 8GB
Hard Disk
: 1TB
Operating System
: Windows 10, Linux

4-DESIGN

System Architecture:

The system architecture is designed as a modular and layered structure that efficiently processes satellite or cloud images to detect cloud patterns and predict weather conditions. It integrates a user- friendly front-end, robust backend, and intelligent machine learning components into a seamless workflow.

The front-end is developed using HTML, CSS, and JavaScript, enabling users to upload satellite images or interact with real-time visualizations. A GUI built with PyQt5 enhances usability, allowing even non-technical users to operate the system. Features include uploading cloud images, selecting regions, and displaying prediction results. The backend, written in Python, manages the core operations such as image preprocessing, cloud segmentation, and classification of weather types.

Machine learning algorithms like K-Means, Decision Trees, and CNN are used to analyze cloud formations and generate predictions. Libraries such as OpenCV, NumPy, Pandas, and Matplotlib are used for image processing, data handling, and visualization. Supabase provides real-time user authentication, data storage, and session handling on the cloud. Optionally, MySQL is used for managing structured data such as user logs and prediction records.

The system ensures secure data communication between the front-end and backend, while maintaining fast processing speeds. It supports real-time or batch-mode prediction depending on the input and model. The architecture is cloud-deployable and can be hosted on platforms like AWS or Google Cloud. The use of containerization (e.g., Docker) makes it portable and easy to deploy across various environments.

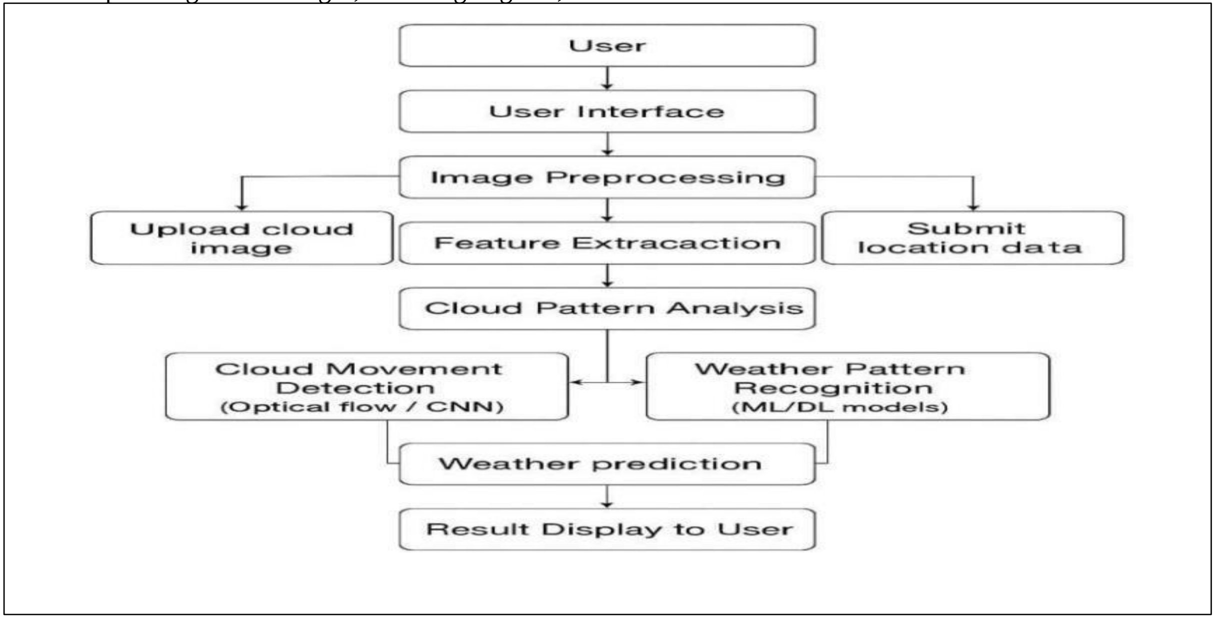


Fig. 4.1.1.1 System Architecture

1.1.1 Technical Architecture:

The technical architecture for detecting and predicting weather using cloud technology is built using a combination of machine learning models, satellite image processing tools, and scalable cloud-based services. The system is designed to ingest real-time satellite imagery—such as from INSAT or similar sources—and analyze atmospheric conditions to detect cloud patterns and predict weather phenomena. The core backend logic is developed in Python 3.6 and uses libraries such as TensorFlow, Keras, NumPy, OpenCV, and scikit-learn for image processing, clustering, and prediction tasks.

Cloud detection is primarily done using techniques like K-means clustering and edge-based segmentation on grayscale satellite images. CNNs are used to classify cloud types, while models like Decision Trees, K-NN, and Random Forests are employed to predict weather outcomes such as rainfall, thunderstorms, or clear skies. Preprocessing involves normalization, denoising, and feature extraction of cloud textures and shapes. Extracted features are correlated with historical meteorological data to improve accuracy. The system is trained on labeled datasets including satellite time series and local weather station data.

Detecting and predicting weather using cloud

On the frontend, the application is built using HTML, CSS, and JavaScript and offers an intuitive user dashboard to upload satellite images or view real-time feeds from APIs. The interface includes visual outputs like temperature, humidity, and cloud movement overlays. Webcam functionality is optional but can be integrated for local sky-view input. For storage, user management, and backend operations, the system leverages Supabase as a serverless cloud platform. Supabase manages data storage using PostgreSQL, handles user authentication, and logs user activities and predictions in real time.

RESTful APIs enable communication between the frontend and backend, ensuring smooth operation and real-time data updates. Satellite images and model results are processed and stored on the cloud, making the system accessible across devices. WebSocket or real-time API support allows live updates of prediction results, including alerts for severe weather. The system is scalable horizontally through the cloud, allowing it to handle large volumes of satellite data. It supports modular updates for models and front-end components independently. Monitoring tools are integrated to check prediction reliability and server performance.

The cloud infrastructure ensures fast processing, even for high-resolution satellite imagery. Data security, version control, and user privacy are handled by Supabase's built-in access policies and role-based control. The system is optimized for performance, maintainability, and extensibility, making it suitable for researchers, meteorological departments, and academic institutions. It provides consistent, automated weather insights by intelligently combining cloud computing with image-based weather analytics.

Machine learning models, such as K-Means clustering or CNNs, are trained on historical weather-labeled data to classify cloud types and forecast weather outcomes. These models are deployed on cloud platforms like AWS or Google Cloud for scalable and real-time processing. Additionally, APIs and web interfaces enable users to upload images, view predictions, and monitor weather trends interactively.

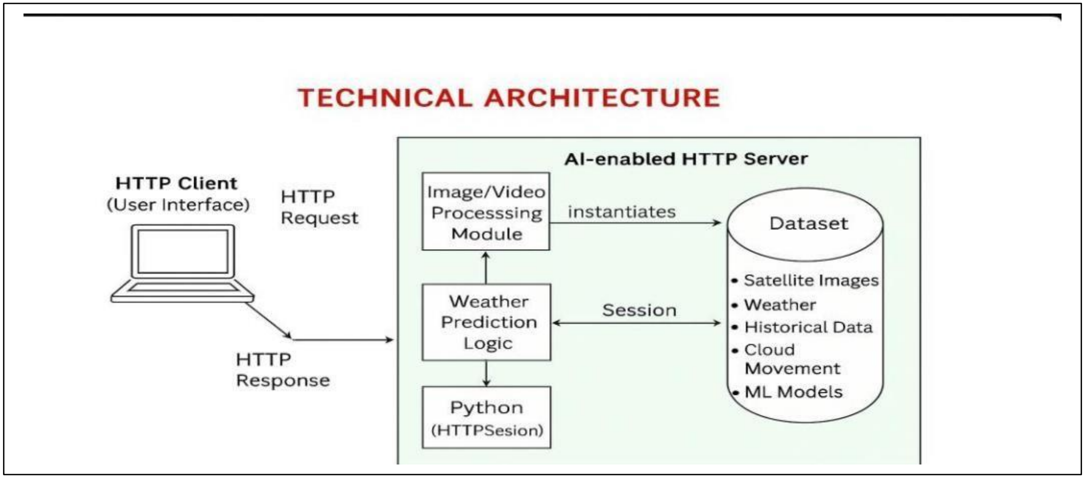


Fig. 4.1.2.1 Technical Architecture

5-IMPLEMENTATION

1.1 Libraries

- *Pandas:*
  Pandas is a powerful library used for managing and analyzing structured data in weather prediction using cloud imagery. It helps in loading satellite metadata and historical weather records efficiently. The library supports cleaning and preprocessing tasks like handling missing values and organizing image-based features. Additionally, Pandas enables time series analysis, which is essential for tracking weather trends and forecasting.
- *Numpy:*
  Numpy is essential for performing numerical computations in detecting and predicting weather using cloud images. It efficiently handles large arrays of pixel data from satellite images, enabling fast mathematical operations. The library supports operations like normalization, filtering, and statistical analysis of image data. NumPy also plays a key role in feature extraction, helping identify cloud patterns crucial for accurate weather forecasting.
- *Matplotlib:*
  Matplotlib is used for visualizing cloud image data and weather prediction results in a clear and interpretable format. It helps plot cloud cover, temperature trends, and prediction graphs for analysis. The library allows overlaying cloud masks on satellite images to verify detection accuracy. These visualizations support better understanding and communication of weather patterns and model performance.

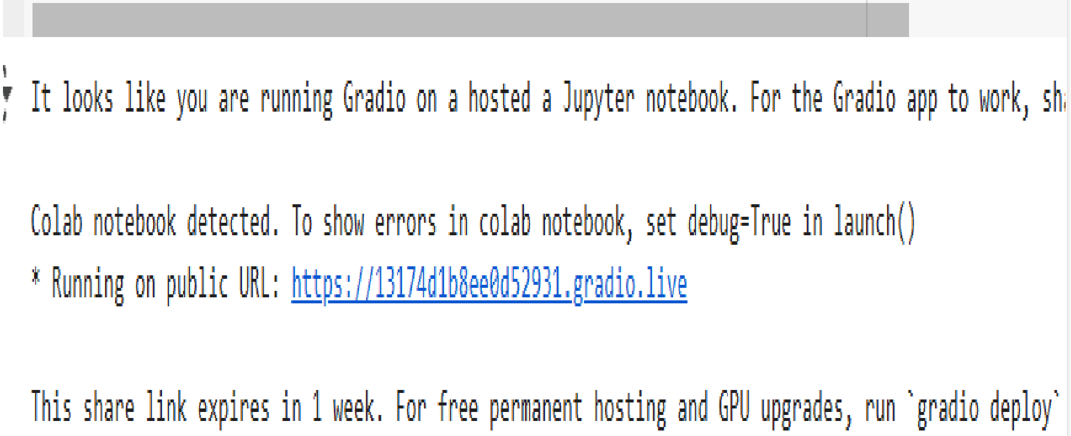
- *Seaborn:*
  Seaborn is a visualization library used to create informative and attractive statistical graphics in weather prediction using cloud data. It helps display correlations between variables like humidity, temperature, and cloud density through heatmaps and scatter plots. Seaborn enhances data interpretation by providing easy-to-read plots with built-in themes. These visual insights assist in understanding complex weather patterns and refining prediction models.

- *Pickle module:*
  The Pickle module is used to save and load trained machine learning models for weather prediction using cloud images. It allows efficient serialization of models, ensuring they can be reused without retraining. This helps in quickly deploying models for real-time prediction tasks. Pickle simplifies the workflow by preserving model states and parameters for consistent results across sessions.

- *Jupyter Notebook:*

  Jupyter Notebook is a powerful interactive environment widely used for developing and testing models for detecting and predicting weather using cloud data. It enables users to write and run code in small, manageable cells, making experimentation more efficient. The platform supports real-time visualization of cloud images, weather graphs, and prediction output.

6. SCREENSHOTS

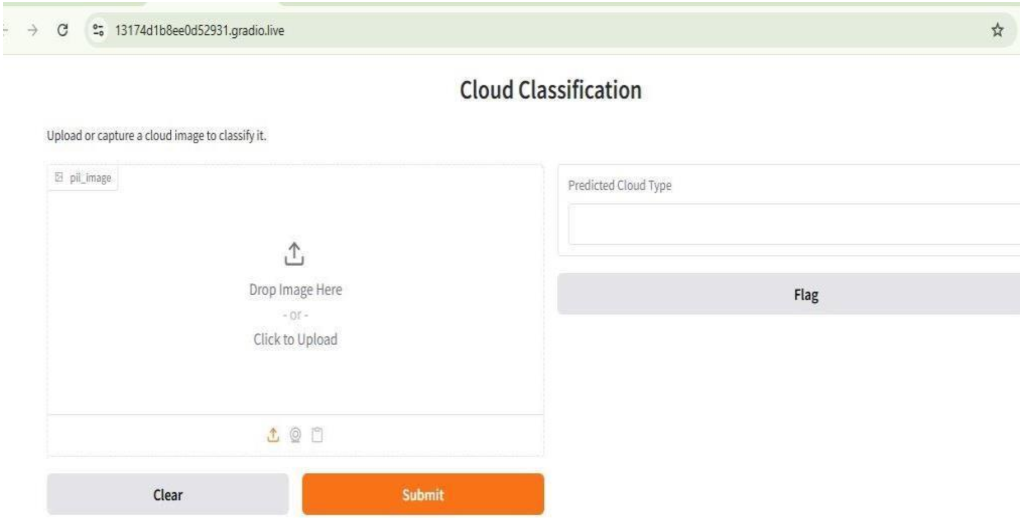


```
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to work, sh

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://13174d1b8ee0d52931.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy`
```

Screenshot 6.1 Link

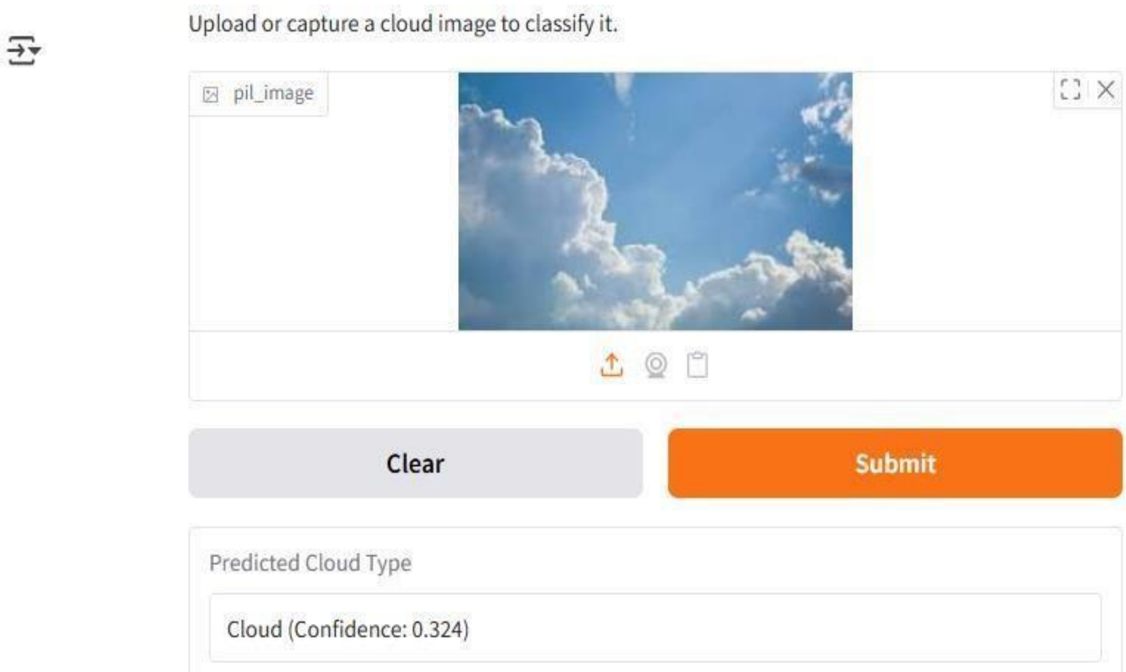


13174d1b8ee0d52931.gradio.live

Cloud Classification

Upload or capture a cloud image to classify it.

Screenshot 6.2 open in browser

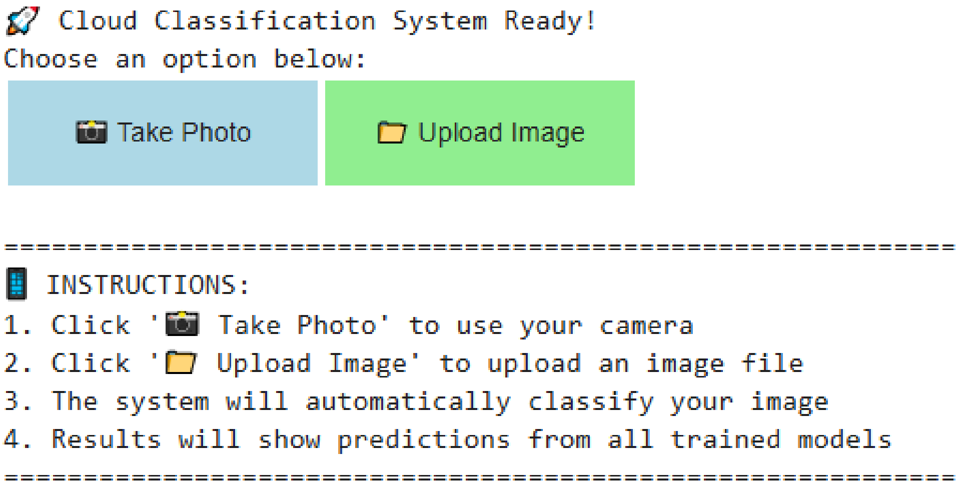


Upload or capture a cloud image to classify it.

Predicted Cloud Type

Cloud (Confidence: 0.324)

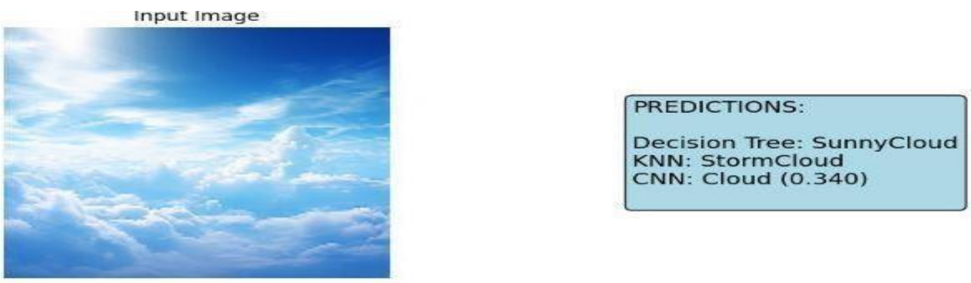
Screenshot 6.3 Upload Image



```
🚀 Cloud Classification System Ready!
Choose an option below:
```

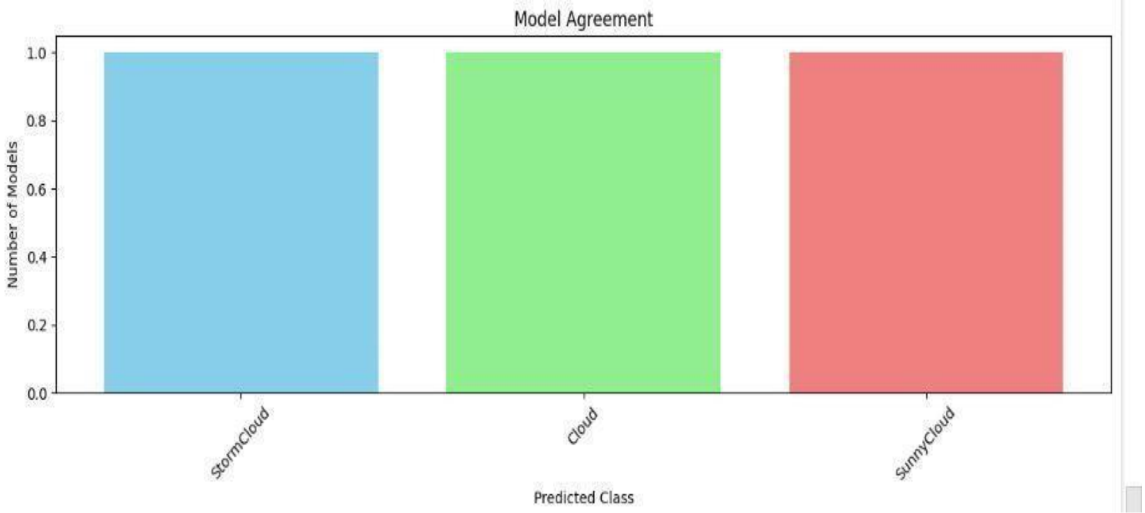
```
============================================================
📱 INSTRUCTIONS:
1. Click '📷 Take Photo' to use your camera
2. Click '📁 Upload Image' to upload an image file
3. The system will automatically classify your image
4. Results will show predictions from all trained models
============================================================
```

Screenshot 6.4 Take Photo Your Camera

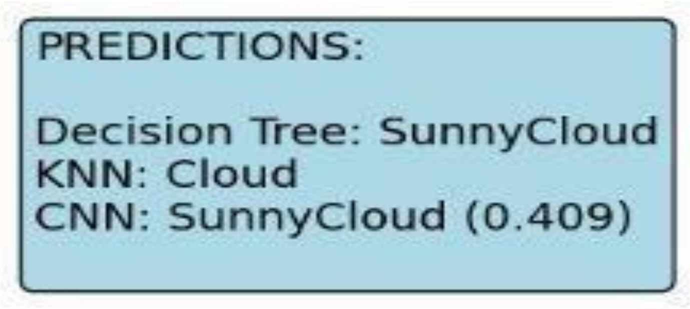


```
PREDICTIONS:

Decision Tree: SunnyCloud
KNN: StormCloud
CNN: Cloud (0.340)
```

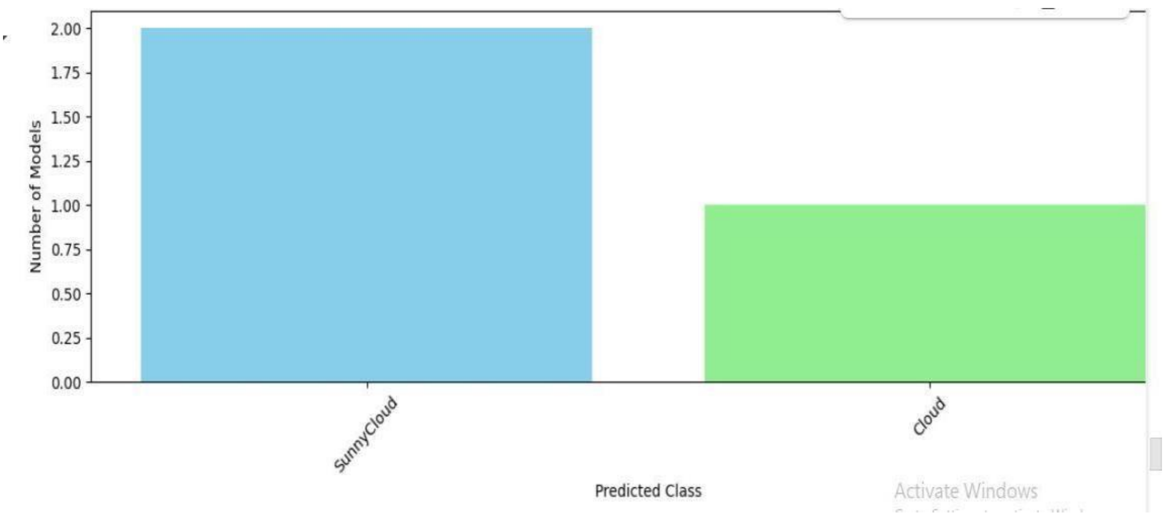


Screenshot 6.5



PREDICTIONS:

Decision Tree: SunnyCloud
KNN: Cloud
CNN: SunnyCloud (0.409)



Screenshot 6.6

7. CONCLUSION

This project demonstrates the ability to detect and classify weather conditions—such as sunny, cloudy, or rainy—using satellite cloud images and machine learning techniques. By combining K-means clustering for image segmentation and classification algorithms like Decision Tree, KNN, and CNN, the system accurately identifies cloud patterns and predicts weather types. The inclusion of a user-friendly interface through Gradio enhances usability, making the system accessible for real-time image analysis. All components of the pipeline— from dataset handling to prediction—performed as expected, showing that the solution is both effective and practical for real-world weather monitoring and forecasting applications.

REFERENCES

[1] M. S. Amini and A. M. Ali,"Cloud Motion Estimation Using Satellite Image Sequences and Optical Flow for Weather Forecasting",*Atmospheric Research*vol.264,2022. Available: [https://doi.org/10.1016/j.atmosres.2021.105887](https://doi.org/10.1016/j.atmosres.2021.105887

[2] T. J. Santoso, I. Ardiyanto,"Cloud Detection and Tracking from Satellite Imagery Using Deep Learning for Rainfall Estimation",*Procedia Computer Science* ,vol 216, pp.511–519 , 2023. Available: https://doi.org/10.1016/j.procs.2022.11.058

[3] S. M. Basha and A. S. Basha,"Cloud Classification and Movement Prediction Using Convolutional Neural Networks",*IEEE International Conference on Computer Vision and PatternRecognition(CVPR)*,2020. Available: [https://doi.org/10.1109/CVPR.2020.00457](**https://doi.org/10.1109/CVPR.2020.00457**)

[4] L. Liu, Y. Zhang, and W. Zheng,"Automatic Cloud Motion Tracking for Nowcasting Using Geostationary Satellite Imagery",*Remote Sensing*, vol. 13, no. 6, 2021. Available: https://doi.org/10.3390/rs13061178