Team Name : Team Visvodaya

Team Leader Name : Guru Brahma Gudimetla

Problem Statement : Developing an AI/ML-based algorithm for identifying tropical cloud clusters using half-hourly satellite data from the INSAT

Team Members

Team Leader:

Name: Guru Brahma Gudimetla
College: PBR VITS

Team Member-1:

Name: Shaik Saleem
College: PBR VITS

Team Member-2:

Name: Booduru Reddy Prasanna
College: VEMU Institute Of Technology

Team Member-3:

Name: Dhanush Babu Janakisetty
College: PBR VITS

Brief About The Idea :

The objective of this project is to **detect Tropical Cloud Clusters (TCCs)** in satellite IR images using deep learning, and to estimate their **potential weather severity**.

TCCs are crucial indicators of atmospheric convection and play a key role in **cyclogenesis**. Detecting them early helps in **forecasting rain, storms, and cyclones**.

Since real INSAT-3D IRBRT data was unavailable, we created a **simulated dataset** using Google-sourced and Python-generated IR-like images.

Our solution involves:

A **Convolutional Neural Network (CNN)** to classify TCC vs No TCC

Image processing algorithms to extract cluster-level parameters:

Size, brightness temperature stats, convective center, cloud-top height

A **rule-based severity estimator** to classify each cluster as:

Normal, Rain, Heavy Rain, Storm, or Cyclone

The entire model is deployed using **Gradio** for real-time use.

Powered by H2S

Opportunity should be able to explain the following:

- How different is it from any of the other existing ideas?
- How will it be able to solve the problem?
- USP of the proposed solution

- **How is it different from existing ideas?**

1. Most cloud detection systems only identify cloud presence or classify broad types. Our solution goes further by: Simulating IR satellite data in the absence of real INSAT-3D access.
2. Using a CNN + post-processing pipeline to detect **Tropical Cloud Clusters.**
3. Extracting cloud-level meteorological parameters, not just classification.

- **How will it solve the problem?**

1. The model automates TCC detection and also predicts potential severity such as **rain, storm, or cyclone**, helping in early warnings.
2. It bridges the data gap by simulating conditions and enables real-time analysis through a deployed interface.

- **USP (Unique Selling Proposition)**

1. End-to-end automated pipeline with **real-time Gradio interface.**
2. **Severity classification** based on extracted physical cloud features.
3. Adaptable to **real INSAT-3D data** and usable by forecasters & researchers alike.

Features of the Solution:

- Detects TCC in real-time using CNN
- Extracts meteorological properties: size, Tb, radius, height
- Predicts severity: Rain, Storm, Cyclone
- Outputs structured data for analysis
- Deployed via Gradio interface — accessible, fast, user-friendly

Solution from the model deployed :

Process Flow Diagram :

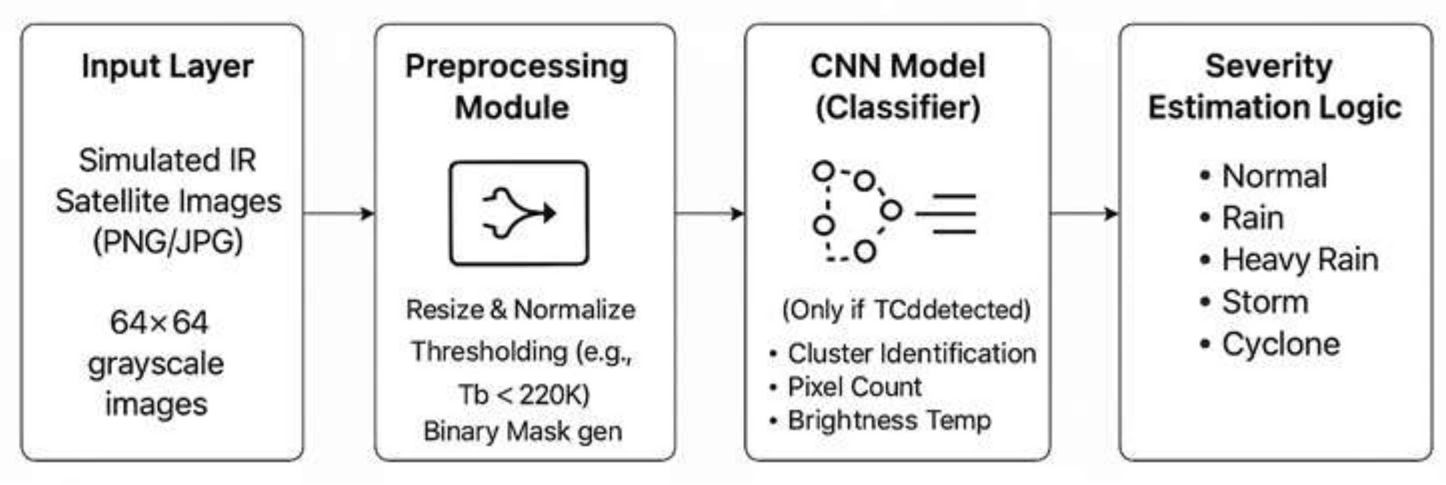
Input Acquisition

Image Preprocessing

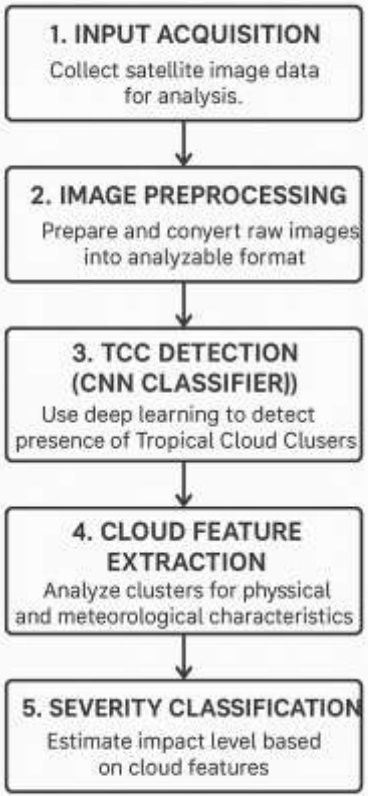
TCC Detection

Cloud Feature Detection

Severity Classification



Wireframes/Mock diagrams of the proposed solution :



- Simulated and real IR images help mimic tropical atmospheric conditions.
- Ensures diverse scenarios for model training and testing.

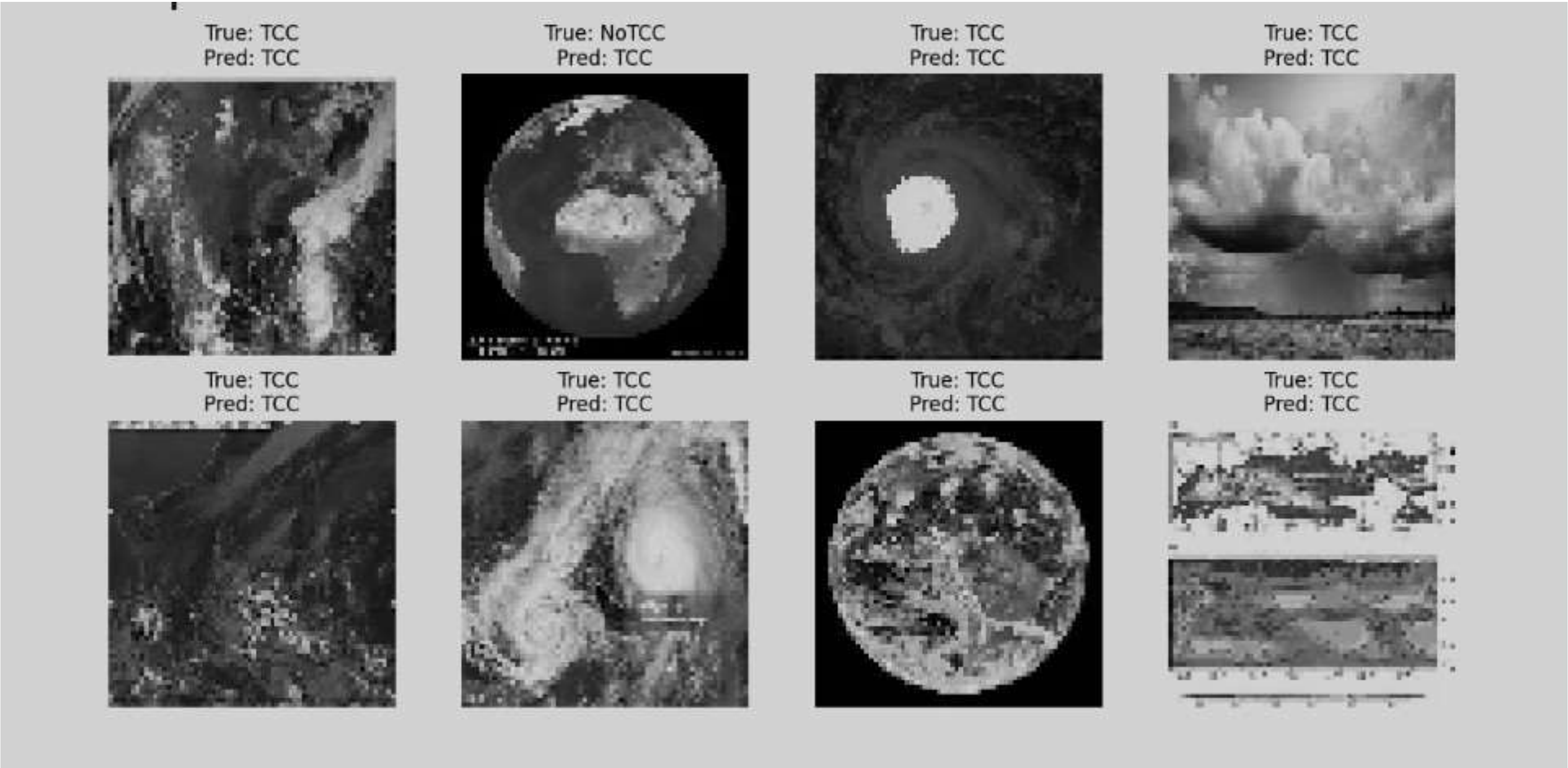
- Images are normalized (e.g., pixel values scaled to 0–1), and thresholding (like Tb < 220K) is applied to highlight colder cloud regions.

- A Convolutional Neural Network is trained to classify whether a given satellite image contains a Tropical Cloud Cluster or not.
- It outputs a binary label (TCC or NoTCC) based on learned spatial features from training data.

- Extract features like size, temperature stats, center, radius, and cloud-top height from detected TCCs.

- Rule-based logic (or a small ML model) uses extracted features such as size and minimum Tb to assign a severity category.
- Categories include: Normal, Rain, Heavy Rain, Storm, and Cyclone, helping in meteorological impact prediction.

Architecture diagram of the proposed solution :



Technologies to be used in the solution:

A. Programming & Environment :
- Python – Core language for all development
- Jupyter Notebook / Google Colab – Interactive development and testing

B. Image Processing

OpenCV / PIL – Reading, resizing, grayscale conversion, thresholding
NumPy / Pandas – Numerical operations and data handling

C. Machine Learning & Deep Learning

TensorFlow / Keras – CNN model design and training
scikit-learn – Evaluation metrics (accuracy, confusion matrix)

D. Cluster & Feature Extraction

SciPy (ndimage) – Cluster detection and center of mass
Custom Python Logic – Feature extraction: size, Tb stats, radius, severity rules

E. Visualization

Matplotlib / Seaborn – Display sample images, masks, results

F. Deployment

Gradio – User-friendly UI for uploading images and viewing TCC predictions

G. Version Control

GitHub – Code hosting and collaboration

Estimated implementation cost :

- The solution is highly cost-effective, using open-source tools (Python, TensorFlow, OpenCV) and free datasets. Development can be done on a personal laptop or Google Colab, keeping the total cost under ₹2,000 for most setups. Optional upgrades like Colab Pro or domain hosting may add ₹1,000–₹3,000, but are not mandatory.

- Estimated Total: ₹0 – ₹5,000 (Max ₹10,000 for advanced needs)

Access the Full Project on GitHub :

https://github.com/Gurubrahma1982/TROPICAL-CLOUD-CLUSTER-TCC-DETECTION/blob/main/TCC.ipynb

Conclusion :

- Developed an end-to-end deep learning pipeline for Tropical Cloud Cluster (TCC) Detection using simulated IR satellite images.
- Successfully built a CNN classifier that identifies TCCs with reliable accuracy.
- Extracted cloud-specific features such as size, brightness temperature stats, radius, and cloud-top height.
- Classified the detected clusters into severity levels: Normal, Rain, Heavy Rain, Storm, Cyclone using logical thresholds.
- Project deployed with organized modules, allowing future integration with real-time satellite feeds and web-based interfaces.

Special Thanks To:

- My team members for their consistent effort and collaboration.
- ISRO and the Bharatiya Antariksh Hackathon 2025 team for the opportunity.
- Mentors and faculty for their support and encouragement throughout this journey.

isro

Powered by H2S

BHARATIYA ANTARIKSH HACKATHON 2025

THANK YOU