NeuralSky: Deep Learning-Based Weather Prediction Using Cloud Images

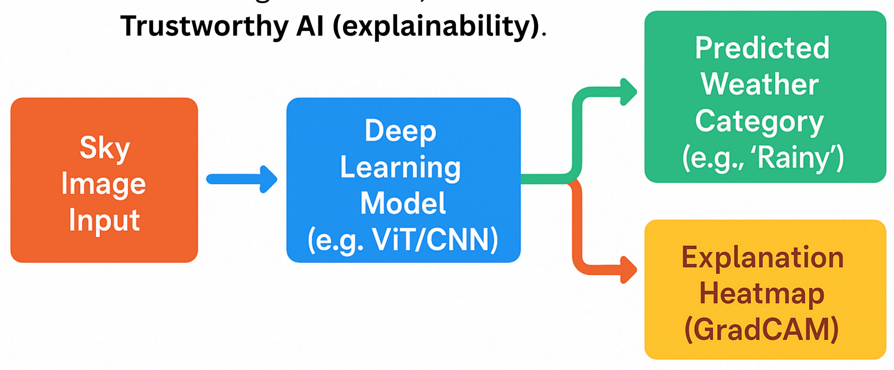
Pranav Dwivedi | Prabhjot Kaur | Charity Gwese | AJ De Vera

1 . Executive Summary / Abstract

Accurate, localized weather prediction is crucial for disaster mitigation and precision agriculture. This project pioneers a **deep learning approach** to classify weather conditions (clear, cloudy, rainy, stormy) directly from ground-based sky images, bypassing expensive sensor networks. We merged heterogeneous cloud image datasets (CCSNv2, Howard-Cloud-X) and trained multiple architectures, with **Vision Transformer (ViT)** achieving **70% test accuracy** – a 10% improvement over baseline CNNs. Crucially, **Grad-CAM** visualizations revealed model reliance on physically meaningful cloud features (e.g., stratocumulus patterns for 'cloudy' predictions), enhancing interpretability. The system was deployed as a **containerized Flask app on Google Cloud Run**, demonstrating real-world viability for accessible, explainable weather forecasting.

2 . Introduction / Motivation

- **Problem:** Need for **automated**, accessible weather **prediction using readily available visual data (cloud images).** Limitations of manual observation and predictions in remote areas.
- **Goal:** Develop an **ML model** to **classify weather** conditions from **cloud images** and **explain why** it makes a prediction.
- **Importance:** Potential for **localised forecasting**, climate monitoring assistance, research tools. Need for **Trustworthy AI (explainability)**.



3 . Datasets & Preprocessing

- **Data Sources:** Merged **CCSN_v2** and **Howard-Cloud-X** datasets.
- **Preprocessing: Resizing** (224x224), **Normalisation**.
- **Splitting:** Data divided into **Training**, **Validation**, and **Test** sets.

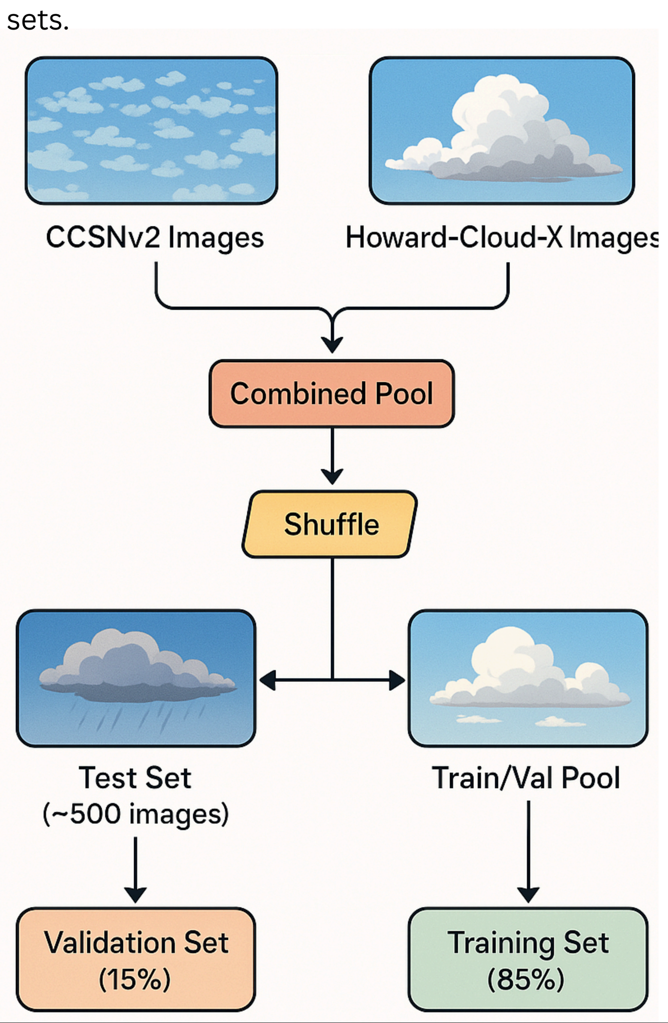
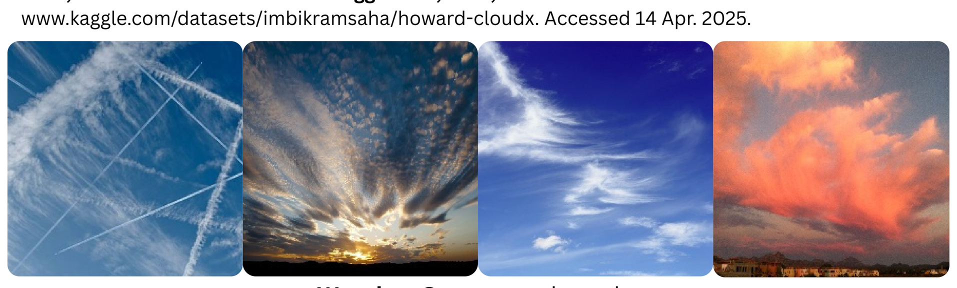


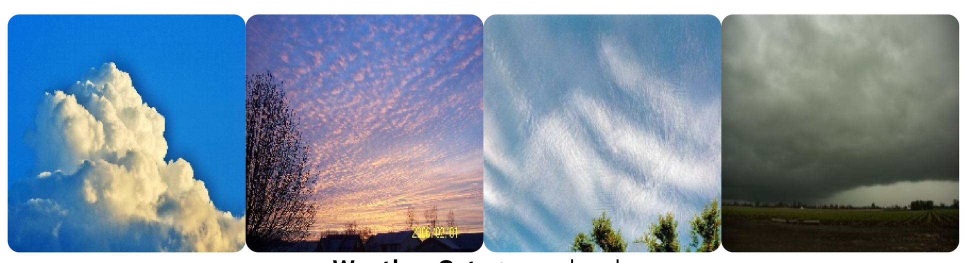
Image Count Per Category	Train	Validation	Test
Clear Sky	662	95	199
Cloudy	1415	277	244
Rainy	583	93	104
Stormy	280	54	33
Total images	3959		

Dataset References:

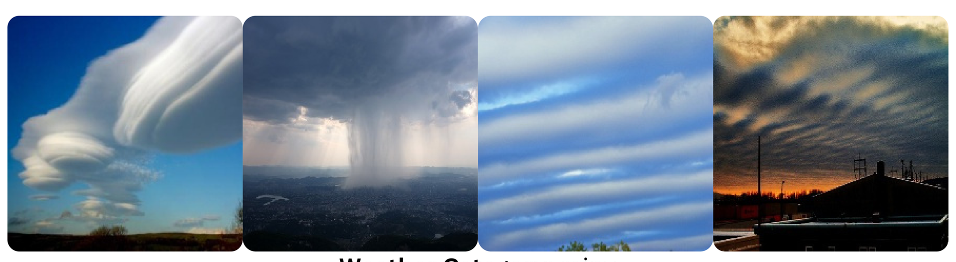
- **Mario. "Cirrus Cumulus Stratus Nimbus (CCSN) Database." Kaggle.com, 2021,** www.kaggle.com/datasets/mmichelli/cirrus-cumulus-stratus-nimbus-ccsn-database.
- **Saha, Bikram. "Howard-Cloud-X." Kaggle.com, 2022,** www.kaggle.com/datasets/imbikramsaha/howard-cloudx. Accessed 14 Apr. 2025.



Weather Category: clear_sky



Weather Category: cloudy



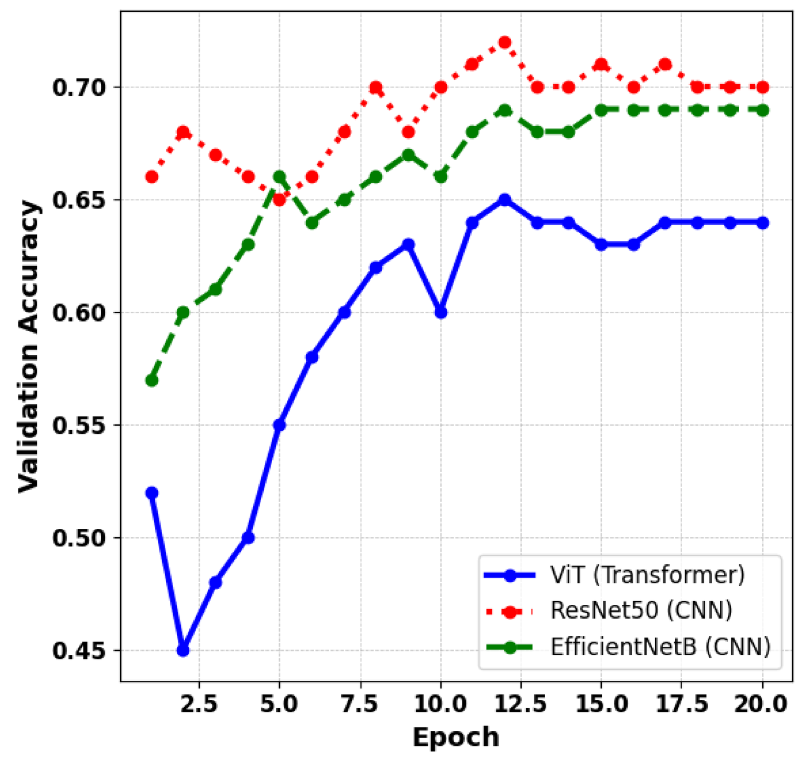
Weather Category: rainy

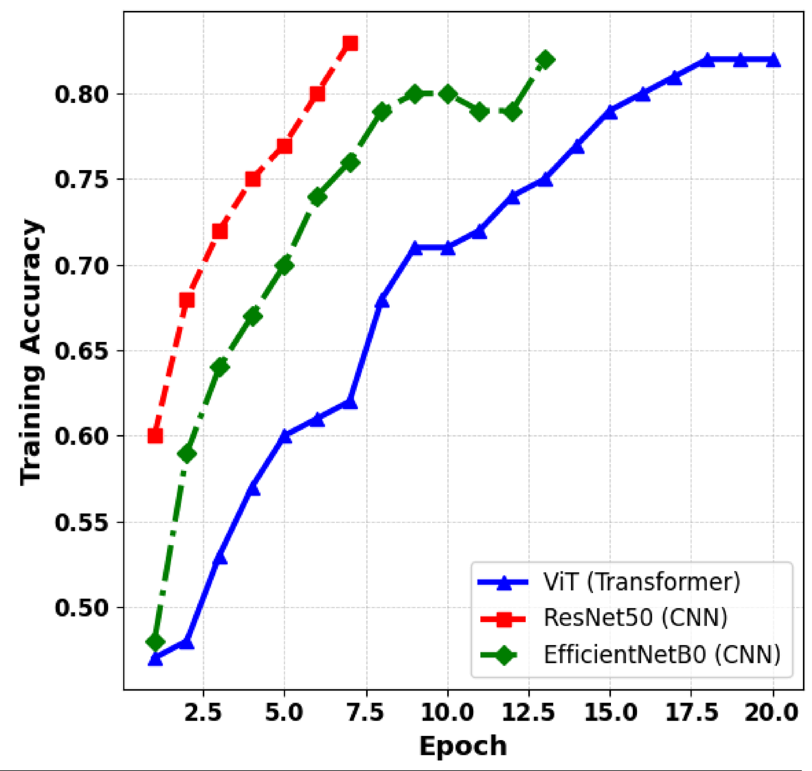
Weather Category: stormy

4 . Methodology

4.1 Model Training & Selection

- **Architectures Tested: Vision Transformer (ViT-Base), ResNet50, EfficientNet-B0.** Utilised **transfer learning** with pre-trained weights.
- **Task:** 4-Class Weather Classification.
- **Training: AdamW** optimizer, **CrossEntropyLoss**, **Early Stopping** based on validation accuracy.

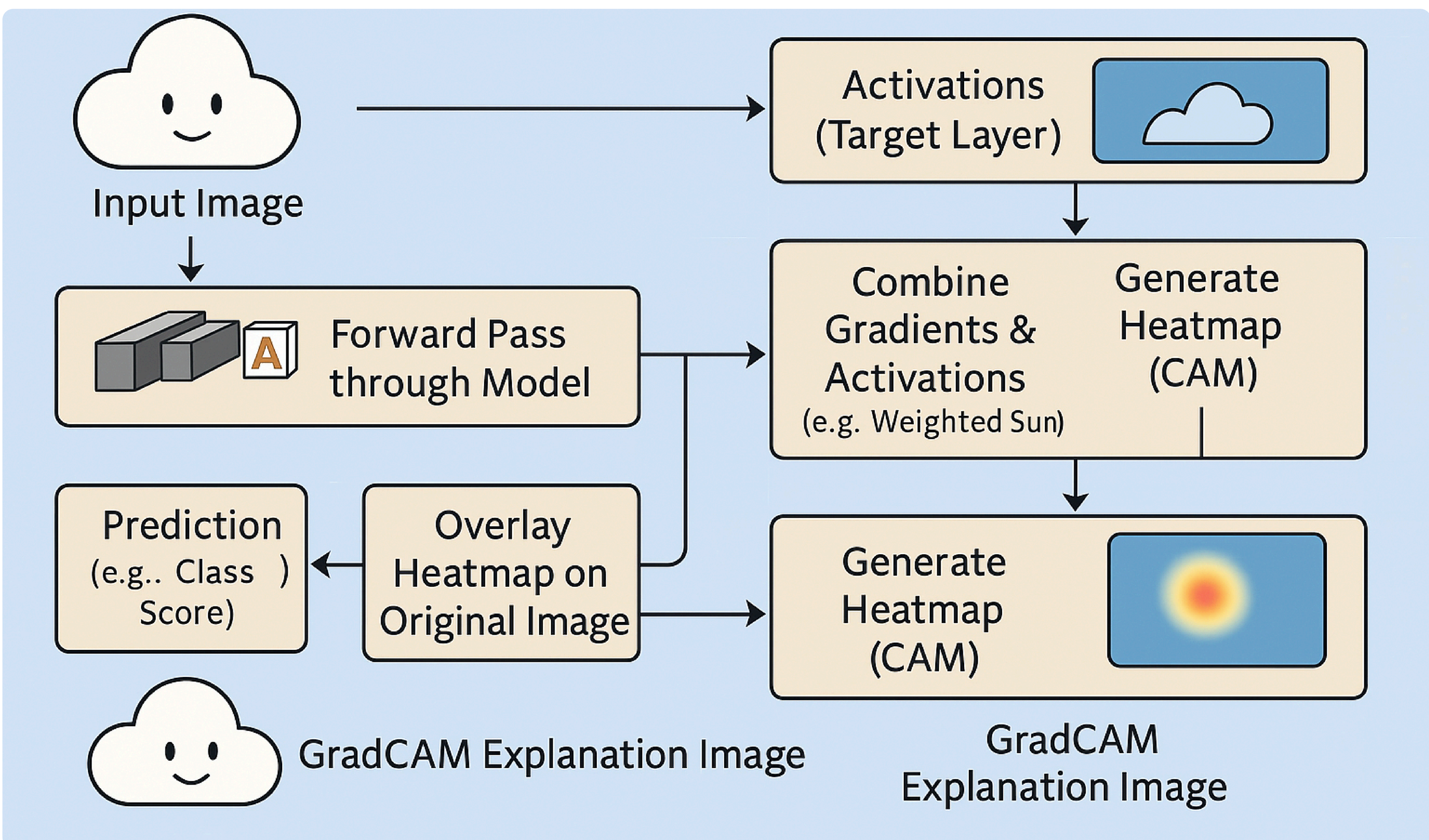




Model	Architecture	Epochs Trained	Final Train Accuracy	Final Validation Accuracy	Test Accuracy
ViT	Transformer	20	0.8156	0.6705	0.6980
ResNet50	Convolutional (CNN)	7	0.8241	0.6455	0.6880
Effecient NetB0	Convolutional (CNN)	13	0.8153	0.6455	0.6660

4.2 Explainability: GradCAM

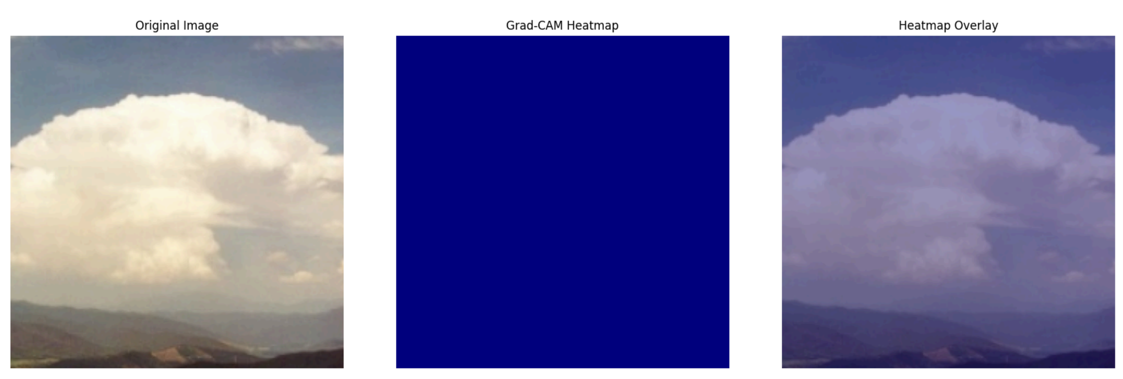
- **Technique:** Gradient-weighted Class Activation Mapping (GradCAM) used to visualize model focus.
- **Implementation:** Adapted for both **CNN** and **ViT** architectures, **targeting relevant layers** (e.g., last conv layer for CNNs, last norm layer for ViT). **Hooks gradients** and activations during backward pass.



GradCAM Paper:
Selvaraju, Ramprasaath R., et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." International Journal of Computer Vision, vol. 128, no. 2, 1 Feb. 2020, pp. 336–359, https://doi.org/10.1007/s11263-019-01228-7.

4.3 GradCAM Explanations

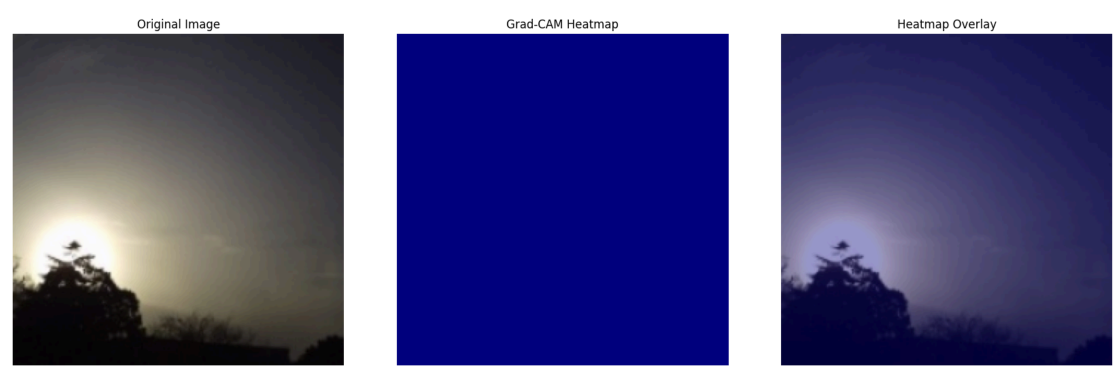
Case 1: True: stormy | Prediction: stormy (0.85) [Correct]



Explanation:

- **Correct 'stormy' prediction** (high confidence).
- **Uniform blue GradCAM** heatmap lacks local region focus.
- **Common** in Vision Transformers
- Suggests **global context reliance** or final-layer gradient averaging.

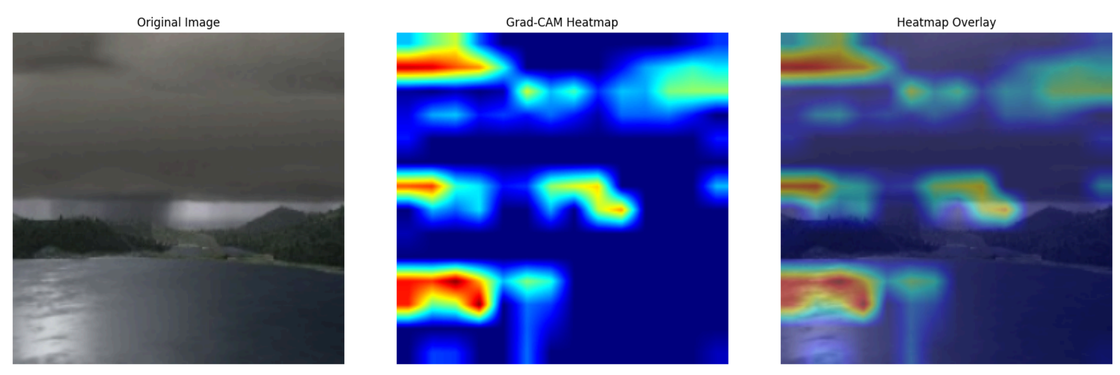
Case 2: True: clear_sky | Pred: stormy (0.47) [Incorrect]



Explanation:

- **Incorrect 'stormy' prediction** (low confidence).
- **Uniform blue GradCAM:** no localized features for prediction.
- **Common in ambiguous inputs** (e.g., glare) or diffuse gradient signals.
- Suggests **weak activation patterns** at the target layer.

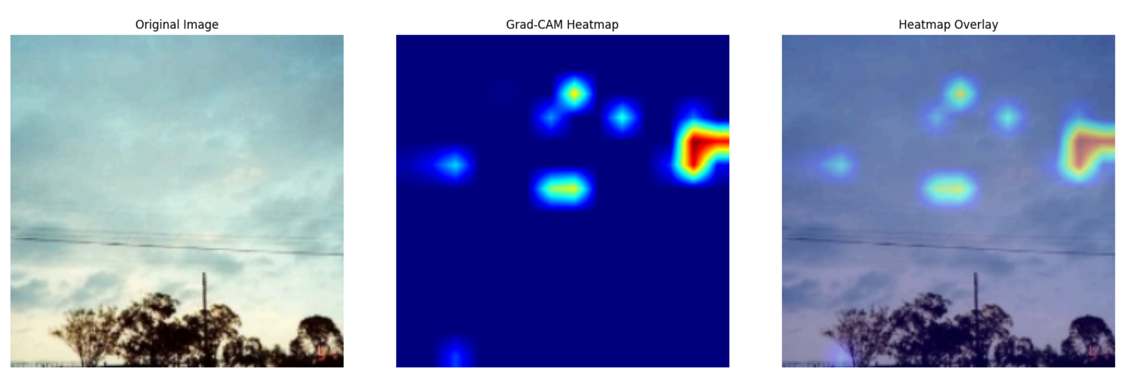
Case 3: True: rainy | Pred: rainy (0.48) [Correct]



Explanation:

- **Correct 'rainy' prediction** (moderate confidence).
- GradCAM **highlights dark horizon clouds & water texture.**
- Model **prioritizes typical rainy-weather features.**
- Demonstrates **focused activation** on relevant visual cues.

Case 4: True: cloudy | Pred: cloudy (0.73) [Correct]



Explanation:

- **Correct 'cloudy' prediction** (high confidence).
- GradCAM **highlights scattered sky clouds**, strong activation on upper-right formation.
- **Multiple cloud regions** identified as evidence.
- Reflects **robust feature recognition** for class-specific traits.

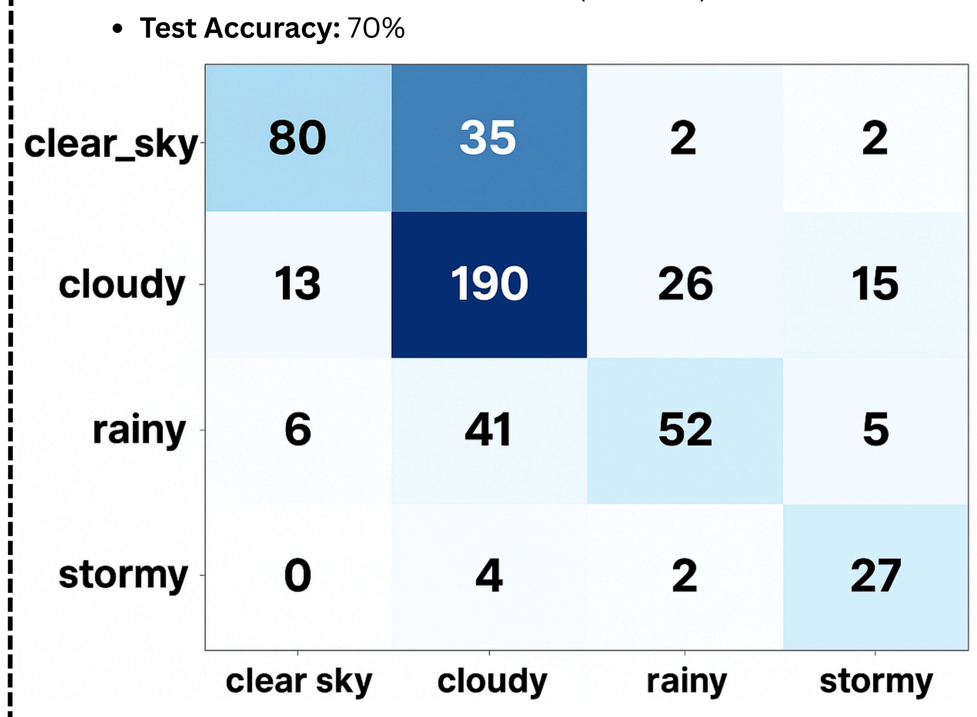
5 . Key Findings

- The **ViT-Base** architecture demonstrated the **highest** classification **accuracy 70%** on the unseen test set.
- **GradCAM** visualizations successfully **highlight discriminative cloud features** relevant to the predicted weather category (e.g., dark, turbulent areas for 'Stormy', thin wisps for 'Clear Sky').
- The model and explanation system were successfully **containerised using Docker** and **deployed scalably** on **Google Cloud Run**.
- Validation split and **early stopping** were effective in **preventing overfitting** during training.

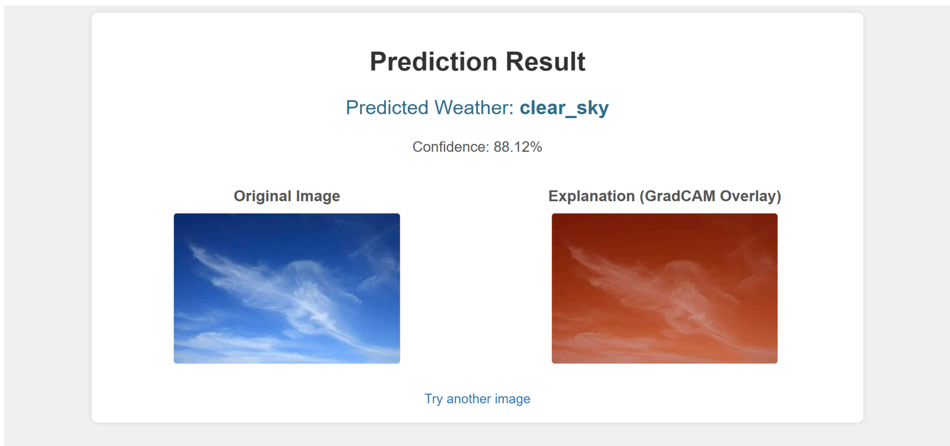
6 . Results

6.1 Classification Performance

- **Best Model:** Vision Transformer (Vit-Base)
- **Test Accuracy:** 70%



6.3 Deployed Web Application



1. **User Uploads Image:** The user navigates to the site (via an initial upload page, not shown here) and uploads an image containing clouds or sky.
2. **Backend Processing:** The uploaded image is sent to the backend (the deployed Flask application running in Cloud Run/Docker).
   a. The application **preprocesses the image** to the format expected by the trained machine learning model.
   b. It **feeds** the image to the loaded **ML model**.
   c. The model **predicts** the **weather category** ("clear_sky" in this example) and **calculates** a **confidence score** (88.12%).
   d. The application then runs the **GradCAM** algorithm to **generate** a **visual explanation** (heatmap) showing which **parts** of the image **most influenced** the model's prediction.
3. **Result Display:** The backend renders this "Prediction Result" page, sending back the results:
   a. The **predicted weather category** and **confidence score** are displayed as text.
   b. The **original image** uploaded by the user is shown.
   c. The **GradCAM explanation** is visualized as a **heatmap overlay** on the original image, providing insight into the model's decision process.
   d. A "Try another image" link allows the user to return to the upload page to analyse a different picture.

7 . Conclusion & Future Work

- **Successfully developed** and **deployed** an end-to-end, **explainable deep learning system** for **weather prediction** from **cloud images**.
- **Demonstrated** the **feasibility** of using **ViT/CNN** models and **GradCAM** for this task.

Future Work:

- Incorporate **larger**, more **diverse datasets**.
- Explore **time-series analysis** using **sequences of images**.
- **Investigate** other **explainability techniques** (e.g., LIME, SHAP).
- **Optimize** model size for **faster inference** (quantization, pruning).
- **Integrate** with **real-time camera feeds**.

8 . Acknowledgements

- **Unit Conveners:** We appreciate the guidance, framework, and support provided by our Unit Conveners.
- **Sponsor:** Our sincere thanks go to our project sponsor, **Mrs. Jansi Sankar**, for providing us with this challenging and engaging project opportunity.
- **Mentor:** We would like to express our heartfelt thanks to our mentor, **Mr. James Ireland**, for his dedicated guidance and invaluable support throughout this project.
- **Team Members:** Finally, a special thanks to the entire project team.

Try out the website for yourself!!

Contact: u3240814@uni.canberra.edu.au | **GitHub:** github.com/pronzzz/weather-prediction

Project No: 2025-S1-09