Case Study Rubric

Why am I doing this?

You are contributing to active research in satellite-based weather classification. By using modern machine learning architectures and testing their reproducibility on a real dataset, you are engaging in a practical, open-ended scientific challenge that mirrors the work of real data scientists.

What am I going to do?

Use the repository to guide your workflow. Begin by reading the following documents inside the repository:

- Hook.pdf (scenario and motivation)
- Rubric.pdf (this document)
- Resources.pdf (guides for scripts)
- README.md (step-by-step instructions for reproduction)

How will I know I have Succeeded?

- Your MobileNetV2 model achieves $\geq$ 92% accuracy on the test set
- Your EfficientNetV2 model achieves $\geq$ 92% accuracy on the test set
- You produce the full set of deliverables described below

Spec Category	Spec Details
Formatting	The provided GitHub repository must maintain the provided top-level structure: • README.md • Hook.pdf • Rubric.pdf • Resources.pdf • DATA/ • OUTPUT/

	● SCRIPTS/

DATA	This folder must contain:
	
	● raw_data.pdf
	    ○ This contains the link to the page that hosts the original satellite images
	    ○ You must download the raw data yourself
	● cleaned_data/
	    ○ Created after running `preprocess.py`
	    ○ Must contain the final cleaned dataset organized into the following folders:
	        ● extreme/
	        ● normal/

OUTPUT	This folder must contain:
	
	● EDA/
	    ○ All EDA plots generated by running `eda.py`
	● Model outputs generated by running `train_efficientnetv2.py` and `train_mobilenetv2.py`:
	    ○ gradcam_efficientnetv2/
	    ○ gradcam_mobilenetv2/
	    ○ efficientnet_v2_classification_report.txt
	    ○ efficientnet_v2_confusion_matrix.png
	    ○ efficientnet_v2_metrics.csv
	    ○ mobilenetv2_classification_report.txt
	    ○ mobilenetv2_confusion_matrix.png
	    ○ mobilenetv2_metrics.csv

SCRIPTS	This folder must contain: ● eda.py ○ Generates all EDA plots and routes them to OUTPUT/EDA

	- preprocess.py - Reads raw data and produces cleaned_data/ in the DATA folder - train_efficientnetv2.py - Trains EfficientNetV2 - Produces Grad-CAM images, confusion matrix, metrics, and classification report - train_mobilenetv2.py - Trains MobileNetV2 - Produces Grad-CAM images, confusion matrix, metrics, and classification report