CS3: Extreme Weather Image Classification - Student Rubric

Goal: Successfully construct and rigorously evaluate a Deep Learning model for binary classification of satellite imagery (Extreme vs. Normal Weather).

Task Summary

Your work begins with the original 9,081 image dataset (Harvard Dataverse) and must proceed through the complete pipeline: data normalization, class consolidation, model development (baseline and advanced), full testing, and interpretable analysis.

Assessment Criteria (What Constitutes Successful Completion)

Category	Criterion	Exceeds Expectations (100%)	Meets Expectations (80%)	Needs Work (50%)
A. Data Preparation	**Replication of Pipeline**	Correctly implements stratified 80/10/10 split AND applies all necessary pre-processing (224x224 resizing, deduplication).	Successfully executes the stratified 80/10/10 split on the consolidated 2-class dataset.	Class consolidation is incorrect or the splitting process shows significant error.
B. Model Selection	**Advanced Architecture Use**	Implements and trains **two** distinct models: a foundational CNN (baseline) AND a sophisticated Transfer Learning architecture.	Implements and trains **one** appropriate Transfer Learning model.	Only a basic, non-transfer learning model is attempted.

C. Performance Metrics	**Target F1-Score Attainment**	Achieves a Macro-Averaged F1-Score of **0.90 or greater** on the Test Set.	Achieves a Macro-Averaged F1-Score between **0.85 and 0.89** on the Test Set.	The final F1-Score is below the 0.85 threshold.
D. Interpretation	**Grad-CAM Visualization**	Generates clear Grad-CAM heatmaps AND includes thoughtful analysis of which meteorological features the model attends to.	Generates the required Grad-CAM visualizations and provides a sufficient description of the model's focus.	Model interpretability visualization is omitted entirely.
E. Documentation	**Quality & Reproducibility**	All scripts are thoroughly commented, easily reproducible, and the final report offers a substantive discussion of potential bias.	Code executes reliably, and the report adequately documents the key steps and final metrics.	The submitted code is non-functional, difficult to follow, or lacks adequate documentation.

Required Deliverables (Submitted via GitHub Repository)

1. **Code (`Scripts/` folder):** One Python script/Jupyter Notebook (`classification_pipeline.py/ipynb`) documenting the entire workflow.
2. **Results (`Reports/` folder):** Final Confusion Matrices and Classification Reports.
3. **Visualizations (`Visuals/` folder):** Key Grad-CAM heatmaps.
4. **Final Reflection (README, 300-500 words):** A summary addressing:
    - Justification for the chosen transfer learning model and its effectiveness.
    - Assessment of the most difficult analytical choice (Class Collapsing, Resizing, or Deduplication).
    - Comparison of final metrics (Accuracy, F1-Score) against original hypothesis targets.