Trustworthiness Evaluation

1. Data & Model Bias - Identified Biases: - The dataset is geographically constrained and may not represent global landslide-prone regions. - The dataset is imbalanced, with fewer landslide samples compared to non-landslide samples. - Mitigation: - Used Stratified K-Fold splitting to maintain class balance in all training/validation sets. - Applied image augmentations (flips, rotations, scaling, coarse dropout) to diversify training samples and reduce overfitting. - Limitations: These mitigations improve robustness but cannot fully eliminate geographic or seasonal bias in the dataset.

2. Model Transparency - Transparency Actions: - Developed a multi-model pipeline combining YOLOv11 classification, EfficientNetV2, EVA transformer, and LightGBM models. - Stored out-of-fold (OOF) predictions for every model and fold for traceability and validation. - Used modular scripts to train, validate, and infer with each model separately. - Challenges: - Transformer-based EVA and CNN models remain black-box models; no feature attribution (e.g., SHAP, Grad-CAM) was performed in this iteration.

3. Approach Reusability - Reusability Strengths: - Fold-wise training and saving of predictions makes this pipeline easily extendable to new datasets. - Ensembling strategy (weighted blending of probabilities across different models) is modular and can be reused for other binary or multi-class classification problems. - Limitations: - The EVA model is computationally heavy, requiring significant GPU resources. - Further optimization (quantization, distillation) is needed for real-time or resource-constrained deployments.

4. Sustainability and Efficiency - Efficiency Measures: - Leveraged transfer learning using pretrained EfficientNetV2 and EVA models, avoiding full training from scratch. - Trained folds in chunks to reduce GPU memory load. - Used mixed-precision training to reduce computational overhead. - Trade-offs: - The ensemble improves accuracy but increases computational cost compared to a single-model solution.

5. Ensembling Strategy

To improve prediction reliability, a multi-stage ensembling process was implemented:

Step 1: Individual Model Predictions - YOLOv11, EfficientNetV2, EVA, and LightGBM were trained separately using 5-fold cross-validation. - For each fold, out-of-fold (OOF) predictions and test set predictions were saved as .npy files.

Step 2: Fold-wise Averaging - For each model, fold predictions were averaged to create a single prediction per sample. - Example: probs = mean(fold_preds, axis=0)

Step 3: Model Blending - Model probabilities were blended using weighted averages: - YOLO + EfficientNet: 0.57 * YOLO + 0.43 * EfficientNet - (YOLO + EfficientNet) + LightGBM: 0.6 * blended + 0.4 * LightGBM - Final blend with EVA: Weighted average of EVA and the previous ensemble (w = 0.45). - Thresholding: Final probabilities were converted to binary predictions using a 0.52 decision threshold.

Step 4: Final Output - The final submission contained averaged ensemble probabilities and corresponding binary class predictions for each test sample.