Date : Mar 20, 2026

GSoC 2026 Proposal : GFOSS - TIRAuxCloud

A Modular AI Pipeline for Thermal Satellite Data with Uncertainty, Explainability, and Unimodal Bias Mitigation

Project : Open-Source AI Framework for Thermal Satellite Payload Data Analysis

Basic Details :

- Full Name : **Samrat R M**
- Email **samratrm2@gmail.com**
- GitHub Username : **samrat-rm**
- IRC nickname on #sugar : **Not on IRC but open to joining**
- Your first language : **Hindi, English**
- Location and Timezone : **Tamil Nadu, India (IST) GMT+5:30**
- Share links, if any, of your previous work on open source projects ?
  I am new to open source projects. **I have contributed to the TIRAuxCloud repo through multiple PRs and Issues.** In addition, I do have 1.5 years of experience as a web dev at Swiggy.

Commitment and Expectations :

I can commit 3 hours daily to this project (except Sundays). I have no planned absences and will inform in advance if any arise, ensuring continuity of work.

AI Usage :

AI tools are used only to improve clarity, grammar, and assist in refining or restructuring ideas. All code, implementations, and submissions are manually reviewed and fully understood before use. I do not commit or submit any code without verifying its correctness and relevance. I also explicitly disclose any use of AI and ensure it is used responsibly within these guidelines.

Relevant Contributions & Prototypes in TIRAuxCloud :

I have actively contributed to the TIRAuxCloud repository through multiple prototype pull requests, focusing on modularization, explainability, and evaluation. This includes

1. Refactoring the codebase into a modular pipeline
2. Implementing DeepSHAP-based explainability with configurable visual outputs
3. Integrating GradCAM++ for SegFormer on Landsat data

4. Adding mean single-pass uncertainty metrics during testing

I also addressed challenges in adapting GradCAM++ to transformer-based architectures (Issue #6), particularly around feature map extraction and layer selection, proposing practical solutions to enable reliable visual explanations beyond CNN-based models.

My Motivation

- What is your motivation to take part in Google Summer of Code ?

I am transitioning from web development to data science and machine learning. I want hands-on learning through real-world projects. GSoC provides an excellent platform for me to gain this practical experience.

- Why do you want to work on this particular project ?

Once I understood what TIRAuxCloud represents, it immediately stood out among the GSoC projects. The use of satellite imagery combined with auxiliary data to distinguish between snow and cloud is both compelling and technically rich. **This aligns closely with the kind of real-world, impactful problems I set out to work on when transitioning my career into machine learning.**

- What are your expectations from us during and after successful completion of the program ?

I'm really interested in working on areas like self-supervised learning and adapting pretrained models for this project. Right now, I don't have deep experience, but I'm actively working on improving it in the coming months.

During the program, I am looking forward to guidance and learning from mentors and I aim to be a consistent contributor to this project beyond GSOC.

Project Details

- What are you making ?

My work on the TIRAuxCloud framework includes making modular AI pipelines, implementing uncertainty quantification and explainability tools with visual explainers. Then I will work on a multis-cale functional entropy regularization approach to balance thermal and auxiliary inputs. This will improve the training loss so that each modality contributes more equally. Finally I will work on clear documentation for the project.

The end result will be a more modular and extensible framework, with focus on improved performance and more robust evaluation and metrics capabilities.

- How will it impact Open Technologies Alliance(GFOSS) ?

This project will enrich GFOSS's open-source ecosystem by providing reusable, modular components for model explainability, uncertainty quantification, and result visualization. Depending on the final implementation scope, these tools can be integrated into broader Earth observation workflows, supporting reproducible research and improving transparency in climate and remote sensing models. It aligns with GFOSS values of open data and reproducible research.

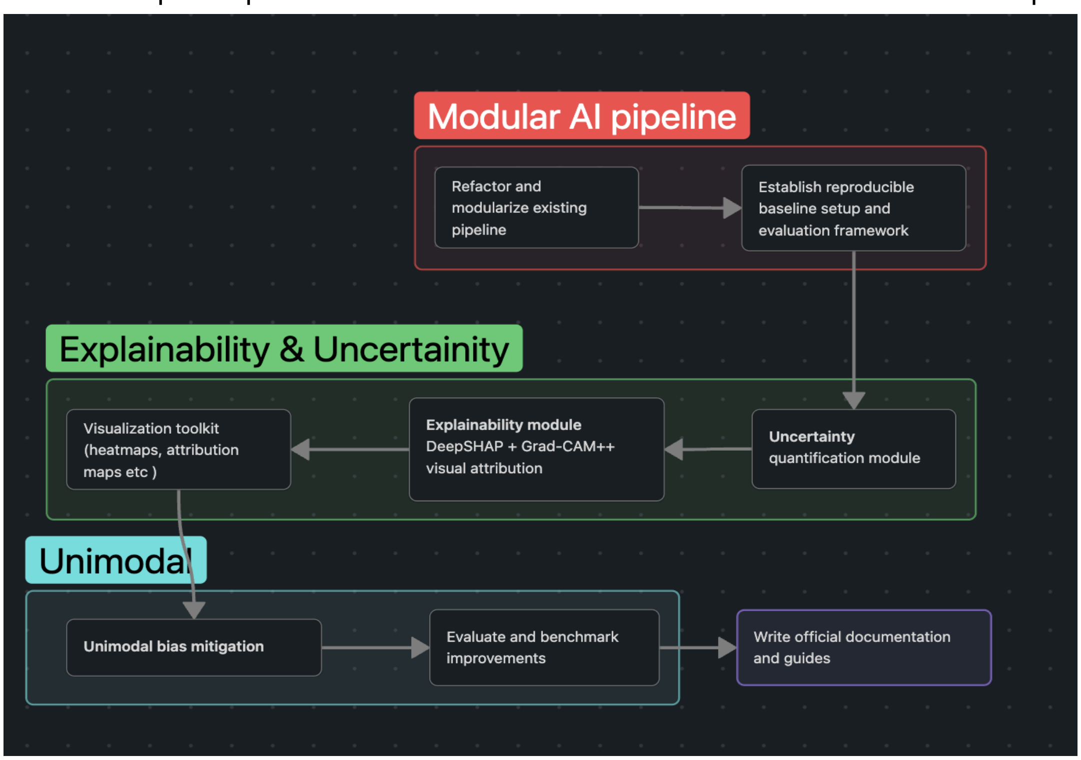
- What technologies (programming languages, etc.) will you be using ?

I will primarily be working with Python, and libraries such as PyTorch for model development and training, along with NumPy and Pandas for data processing. For geospatial and satellite data handling, I will use tools like Rasterio, GeoPandas, and Xarray. Additionally, I plan to use libraries such as pytorch_grad_cam, SHAP and Matplotlib for the new feature implementations like explainability, uncertainty quantification, etc

Timeline:

Expected deliverables :

- A modular and reproducible AI pipeline for thermal satellite data with clear training and evaluation workflows.
- Integrated uncertainty quantification and explainability tools (DeepSHAP, Grad-CAM++, single-pass metrics).
- Implementation of unimodal bias mitigation using entropy-based regularization with evaluated improvements.
- Complete open-source release with documentation and visualization outputs.



Week 1 - 2 : Modular Pipeline & Evaluation Framework

- **Set up and validate baseline pipeline :** Run TIRAuxCloud end-to-end, reproduce baseline results on a controlled Landsat subset, and define a fixed evaluation setup (data split + metrics). A working evaluation/testing setup is already established; this step extends the same reproducible setup to the training pipeline.
- **Refactor into modular structure :** Organize code into clear components for extensibility:

```
.
├── configs/
├── data/
├── evaluation/
├── module_builder/
├── models/
├── training/
├── utils/
├── results/
└── sats_v3/
```

Separate responsibilities and simplify experimentation. Refactor all the model related components into

```
models/
  ├── architecture/
  ├── components/
  |       ├── encoders/
  |       ├── decoders/
  ├── multimodal/
  ├── factory/
  └── utils/
```

- **Extend unified evaluation framework (Optional)**
  Improve existing evaluation by creating a single script that:
  - Runs inference
  - Computes metrics
  - Saves outputs (CSV/JSON + visualizations)

- **Add extension hooks + documentation :** Introduce clean integration points for explainability, uncertainty, and loss functions, and document the full workflow for reproducibility.

Goal: A clean, modular pipeline with a consistent and extensible evaluation system ready for further feature integration.

Prior work : Established an initial modular pipeline structure and dependency setup as a foundation for this work. [PR : Refactor project structure into modular pipeline]

Week 3: Uncertainty Quantification Integration

- Implement single-pass uncertainty estimation using softmax entropy, integrate it into the evaluation pipeline, and generate per-image uncertainty scores for analysis and comparison.
- Review and assess alternative uncertainty estimation methods for possible future integration.
- Extend the implementation to generate spatial uncertainty maps (pixel-wise entropy) for visual analysis (optional, based on mentor preference; otherwise reported as aggregate uncertainty metrics only).

Goal : Functional uncertainty quantification module integrated into the pipeline with consistent evaluation outputs.

Prior work : Implemented a prototype of single-pass uncertainty in the evaluation pipeline, reporting mean uncertainty alongside mIoU and Pixel Accuracy.
[PR : Compute and print single-pass uncertainty metrics during model test]

Prior work output :

```
Pixel Accuracy: 0.7946, mIoU: 0.6550, Mean Uncertainty: 0.0275
Class     IoU  Precision   Recall     F1
0       0.617      0.768    0.758  0.763
1       0.693      0.815    0.823  0.819
(base) ➜  TIRAuxCloud git:(feat/single_pass_mean_uncertainity) ✗ 
```

Week 4: Explainability Integration (Grad-CAM++)

Implement Grad-CAM++ for model interpretability and integrate it into the evaluation pipeline for segmentation models (e.g., SegFormer). Generate class-specific activation maps to highlight spatial regions influencing predictions and enable saving of heatmaps for analysis and comparison.

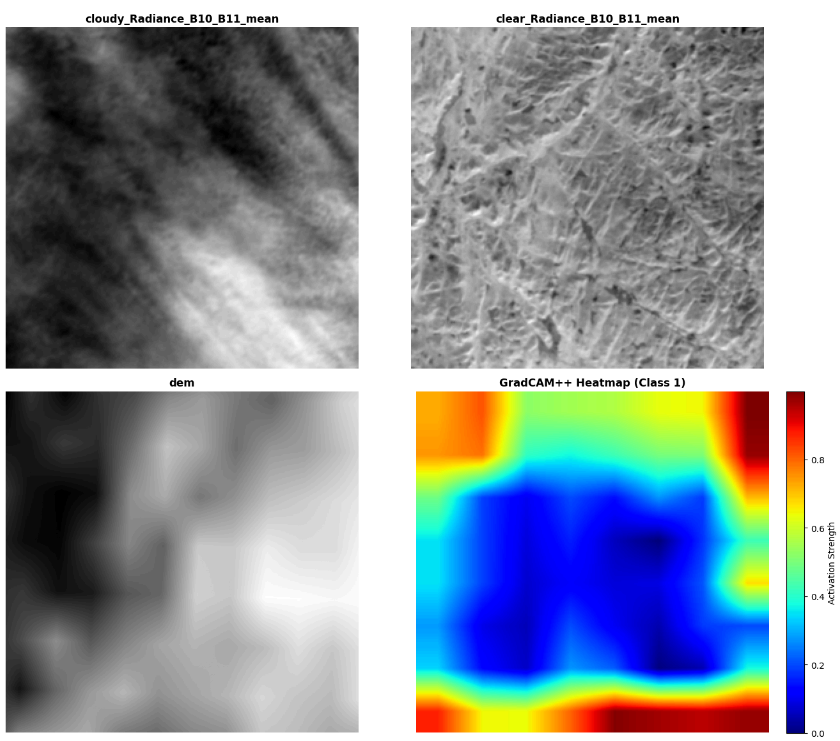
- Adapt Grad-CAM++ for segmentation outputs (pixel-wise logits instead of classification scores).
- using the **pytorch-grad-cam** framework (GradCAMPlusPlus method) and Matplotlib for visualization.

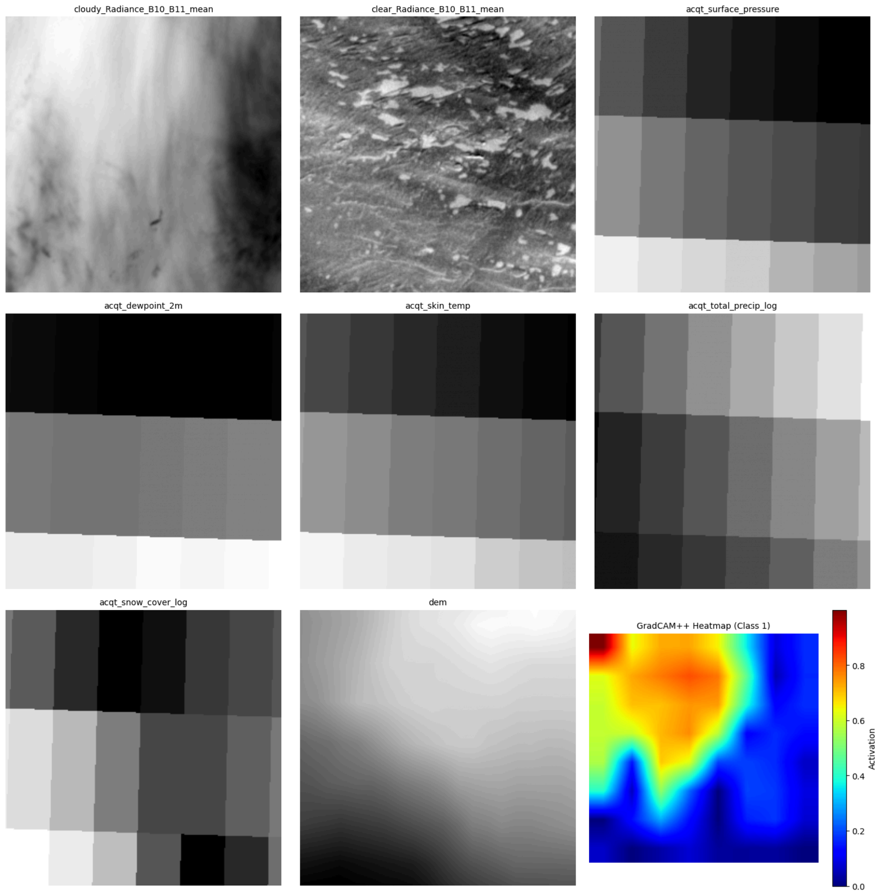
- Identify and hook appropriate intermediate feature layers for meaningful spatial attribution for different modals.
- Implement upsampling and normalization of activation maps to align with input resolution *(For overlay output format).*
- Support batch-wise processing and saving of outputs for large-scale evaluation.
- Generate class-specific activation maps to highlight spatial regions influencing model predictions.
- Extend Grad-CAM++ to support multi-class visualization (3 classes) for comparative analysis *(optional, based on mentor preference)*
- Validate stability and consistency of heatmaps across different samples and classes.

Goal: Functional Grad-CAM++ explainability module producing consistent and interpretable activation maps integrated within the evaluation framework.

Prior Work : Developed an initial Grad-CAM++ implementation for SegFormer-based models as a prototype and raised an issue highlighting all the problems with Grad-CAM++
[PR : GradCAM++ implementation for SegFormer (Landsat dataset)]
[Discussion: Scope and Design Decisions for GradCAM++ Explainability Feature]

Prior work output :





Week 5 : (Midterm Buffer / 30-Day Evaluation Checkpoint)

This week is reserved to consolidate progress so far and ensure all components are stable before moving forward.

- Fix bugs and stabilize the modular pipeline and evaluation framework.
- Validate outputs from the uncertainty module and ensure consistency.
- Clean up code, improve structure, and update documentation.
- Prepare intermediate results for mentor review and feedback.

Goal: Enter Weeks 6–7 with a stable setup and no pending issues, ready for DeepSHAP integration.

Weeks 6 - 7 : Explainability Integration (DeepSHAP)

Implement DeepSHAP-based feature attribution for model interpretability and integrate it into the evaluation pipeline. Generate pixel-level and modality-wise attribution maps using representative background samples to approximate Shapley values.

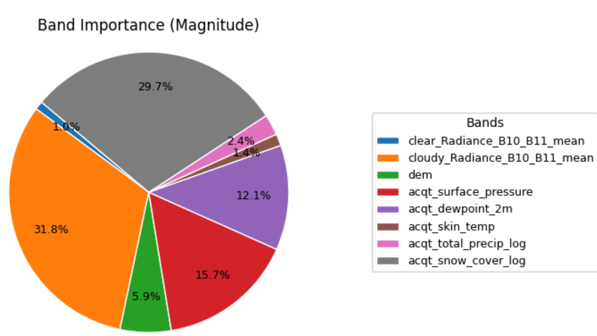
- Integrate DeepSHAP with the segmentation model (SegFormer) using appropriate input wrappers and hooks.
- Use SHAP library for attribution computation and Matplotlib for visualization.
- Support both DeepExplainer (fast, approximate) and GradientExplainer (slower, more accurate) for SHAP value estimation.
- Define and optimize background dataset selection for stable attribution. The dataset should be representative , well-distributed and balanced.
- Generate and store attribution maps for each modality (TIR, auxiliary inputs).
- Analyze attribution distributions to quantify modality contribution (foundation for unimodal bias analysis).
- Optimize computation using batching/subsampling for efficient evaluation.
- Generate SHAP outputs as bar plots, pie charts, waterfall plots, and summary tables.

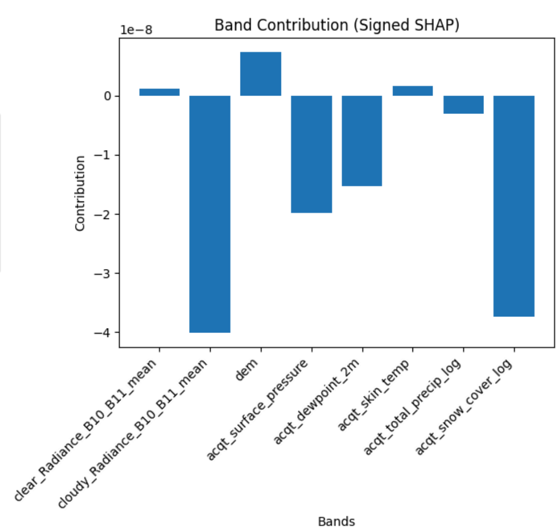
Goal: Functional DeepSHAP explainability module producing consistent feature attribution maps and modality-level insights integrated into the pipeline.

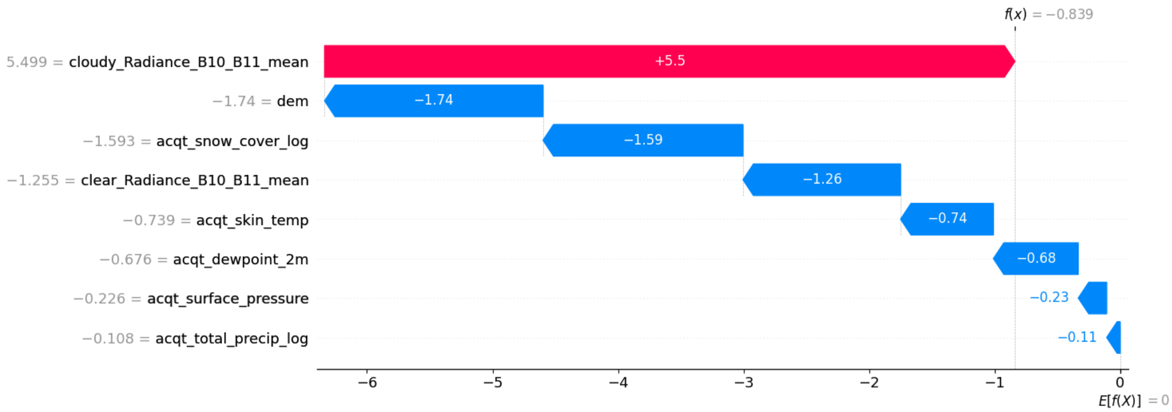
Prior Work:

Developed an initial DeepSHAP-based explainer with configurable visualization outputs.
[PR: DeepSHAP-based model explainer with configurable visualisation outputs.]

Prior Work output :







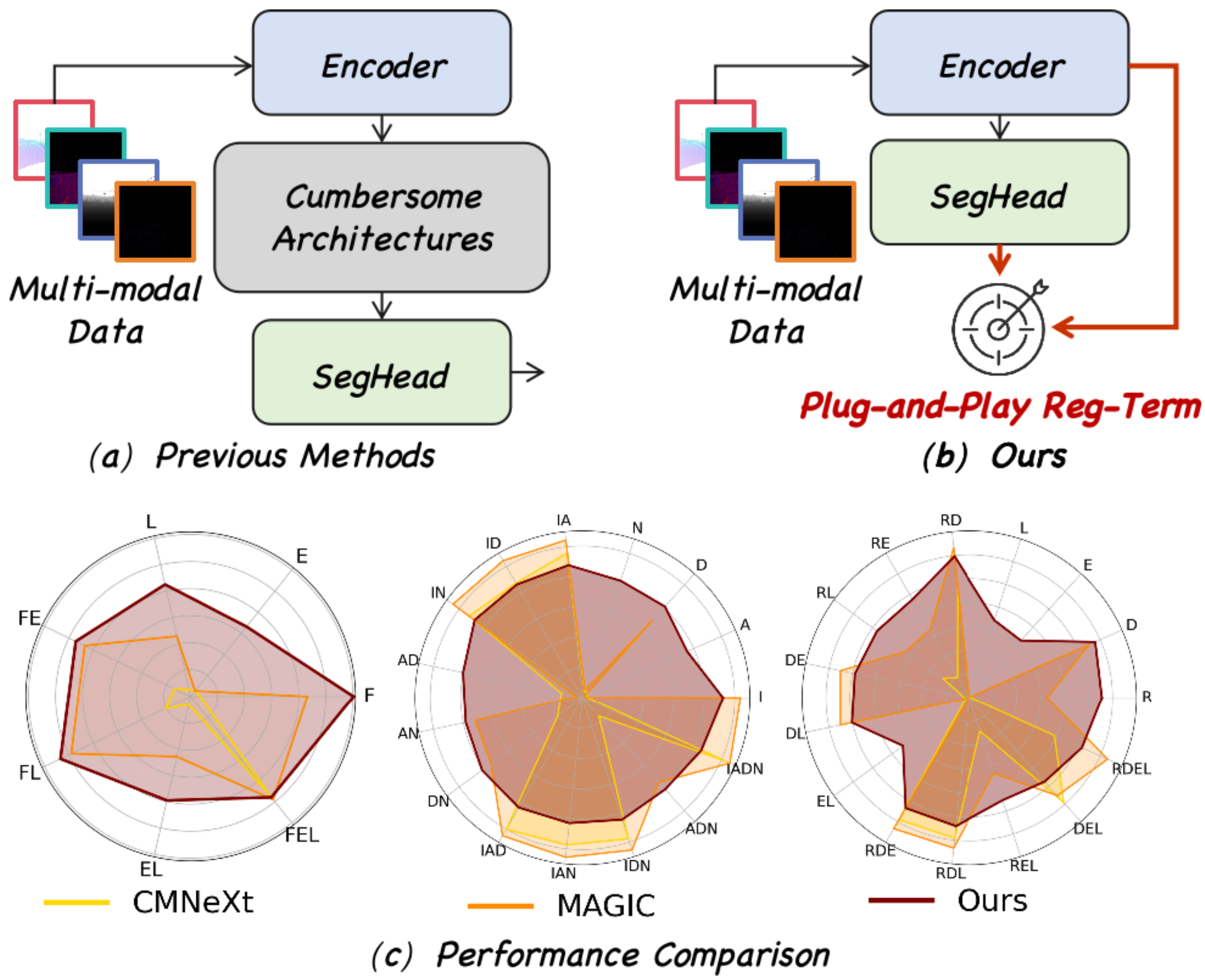

```
Band Contribution (aligned):
Feature                        |     Signed |  Magnitude
-------------------------------------------------------
clear_Radiance_B10_B11_mean    |        -ve |     0.3150
cloudy_Radiance_B10_B11_mean   |        -ve |     0.3291
dem                            |        -ve |     0.0497
acqt_surface_pressure          |        +ve |     0.0876
acqt_dewpoint_2m               |        -ve |     0.0163
acqt_skin_temp                 |        +ve |     0.0161
acqt_total_precip_log          |        +ve |     0.1649
acqt_snow_cover_log            |        -ve |     0.0212
(base) ➜  TIRAuxCloud git:(feat/explainability_DeepSHAP) ✗
```

Weeks 8 - 9 : Unimodal Bias Mitigation (Entropy-Based Regularization)

Implement and integrate multi-scale functional entropy regularization into the training pipeline to reduce unimodal dominance and balance contributions from thermal and auxiliary inputs. This approach is based on *“Reducing Unimodal Bias in Multi-Modal Semantic Segmentation with Multi-Scale Functional Entropy Regularization”* (arXiv: 2505.06635).

Reasoning for Method Selection :

Existing approaches (e.g., CMNeXt, MAGIC) rely on complex architectures for multi-modal fusion. In contrast, the entropy-based regularization method introduces a lightweight, plug-and-play loss term that enforces balanced modality contribution without modifying the model architecture.



This makes it easier to integrate into the existing TIRAuxCloud pipeline while maintaining scalability and reproducibility. The method has also demonstrated good performance compared to architecture-heavy approaches such as CMNeXt and MAGIC.

- Integrate the regularization term into the existing training loop with configurable weighting.
- Train models on controlled subsets to evaluate stability and feasibility.
- Use DeepSHAP attribution to quantify modality contribution before and after regularization.
- Compare performance against baseline models (mIoU, F1) and analyze robustness under modality imbalance.

Goal: A validated training modification that reduces unimodal bias and improves balanced modality utilization, supported by both quantitative metrics and attribution-based analysis.

Week 10 : (2nd Midterm Buffer / 30-Day Evaluation Checkpoint)

Reserve one week to consolidate progress, resolve pending issues, and stabilize implementations before evaluation.

- Fix bugs in explainability (DeepSHAP, Grad-CAM++) and uncertainty modules.
- Validate consistency of uncertainty and attribution outputs.
- Clean and refactor code, finalize modular structure, and update documentation.
- Prepare intermediate results, benchmarks, and reports for mentor review.

Goal: No pending issues; stable and validated modules ready for final development phase.

Weeks 11 – 12 : Documentation and Developer Guides

- Develop comprehensive documentation for pipeline usage, modules (uncertainty, explainability, training), and configuration.
- Create step-by-step guides with examples for running evaluation, generating visualizations, and extending modules.
- Document code structure and integration points for future contributors.
- Use standard documentation tools (e.g., Markdown + GitHub Docs / MkDocs) for structured and maintainable docs.

Goal: Clear, user-friendly, and developer-ready documentation enabling reproducibility and easy extension of the framework.

Motivation **:**

- **Convince us that you will be a good fit** for the GFOSS project you have selected.

I've already spent some time working on the TIRAuxCloud repository through a few PRs around modularization, explainability, and uncertainty. That experience helped me understand how the project is structured and where things can be improved, so I feel comfortable continuing from there instead of starting fresh.

What I like about this project is that it's not just about training a model, but actually understanding it better and making it more reliable. Adding explainability and uncertainty makes the system more useful in real-world scenarios, especially in something like Earth observation where mistakes can matter.

I also want to build things that others can use and extend. If the modules I work on (like explainability or uncertainty tools) end up being useful beyond this project, that would be a big win for me and aligns well with the open-source spirit of GFOSS.

Additional Interest :

If this project is not assigned to me, I am still very interested in contributing to TIRAuxCloud. I would be happy to work on areas such as data preprocessing, data augmentation, or any other tasks where I can add value to the project.

AI Use Disclosure:

AI tools were used only for correcting grammar, minor refactoring, and improving sentence clarity. All ideas and technical content in this proposal are my own. I have reviewed and understood everything before submission.