Flood Detection System

Nelson Almeida
BSc (Hons) Computing in Software Development
https://github.com/nelsondca/Flood-Detection-System-FYP

Ollscoil Teicneolaíochta an Atlantaigh
Atlantic Technological University

Introduction

Floods rank among the most destructive natural catastrophes, resulting in loss of life, damage to infrastructure, and financial loss.
Recent Cork, Ireland flood (October 2023) caused widespread inconvenience and helped highlight the importance of having an appropriate flood monitoring system.

The flood detection system is an application through which users can select a location and see the flood history using satellite data. NDWI analysis and Sentinel-2 data, on which the system itself gathers data, processes that data to determine changes in the water levels, and return the results in an interactive web application.

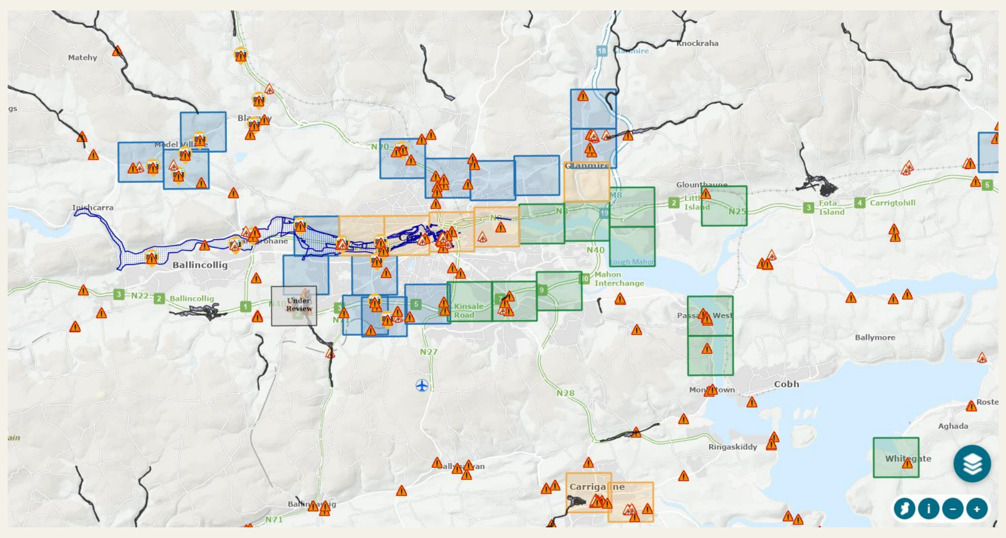

Figure 1 – Region of cork with flood events detected.

Objectives

- Automate flood detection using Sentinel-2 data.
- Calculate NDWI to identify water bodies.
- Detect flooded areas through thresholding.
- Validate results against valid data from trusted sources.
- Visualize results via interactive map using Leaflet.js

Methodology

The project development follows a hybrid agile methodology, combining iterative development with research experimentation.

- Data collection from Google Earth Engine (Sentinel-2, Cork region, September-October 2023).
- Cloud masking and image preprocessing.
- NDWI calculation using the green and near-infrared bands.
- Thresholding NDWI (> 0.3) to create flood masks.
- Exporting results to Google Cloud.
- Visualizing results interactively using Python and geemap.
- Analysis and comparison of data with valid data from trusted sources.

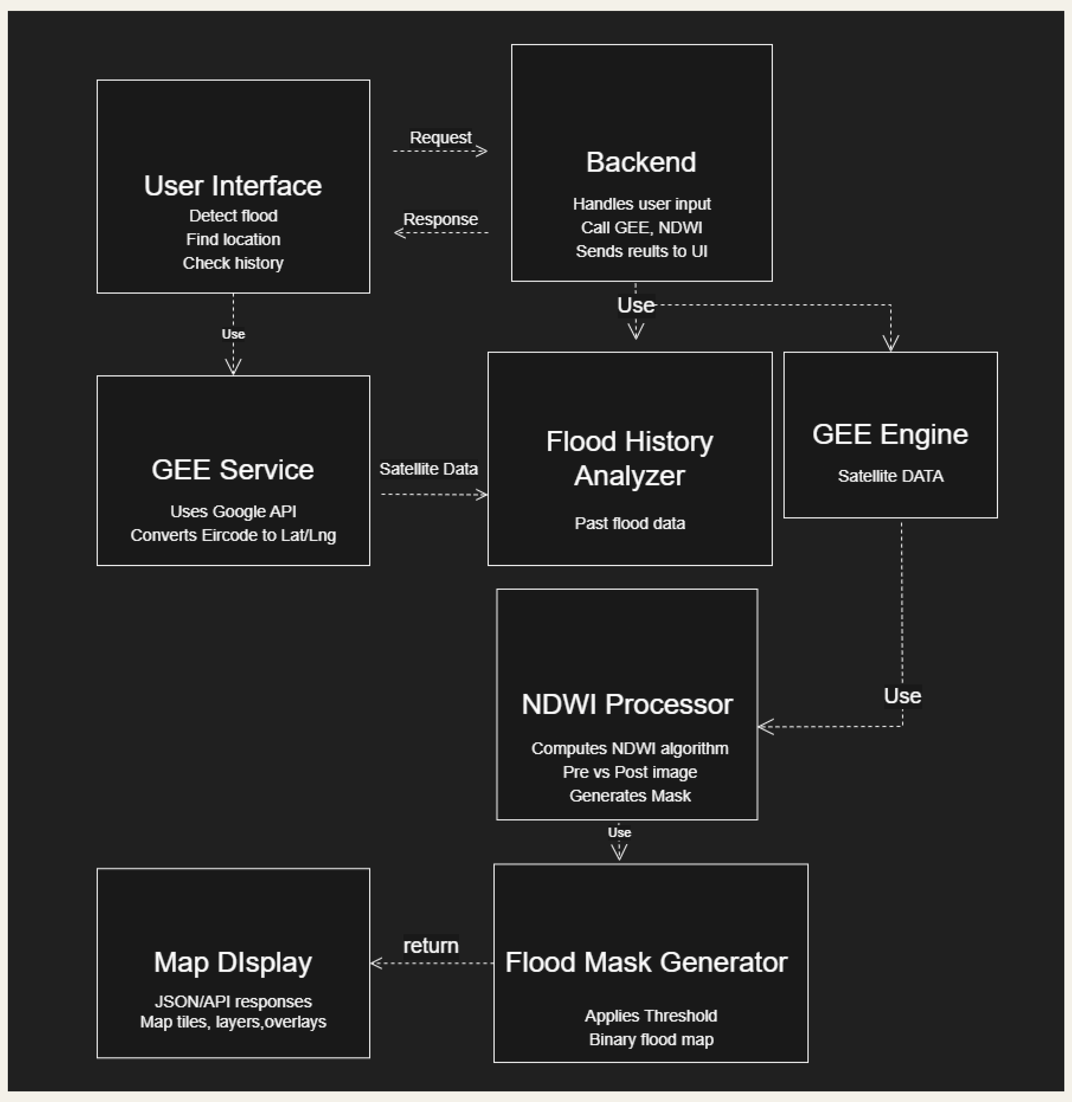

Figure 2 – System Architecture UML Diagram.

Results

The system successfully identified major flooded areas in Cork. NDWI-based masks were similar to official reports. Evaluation criteria showed promising results, confirming the feasibility of flood detection from satellites using open-source software.

Results Outputs

Figure 3 – Cork pre-flood event data for comparison from CEMS (Copernicus Emergency Management Service).

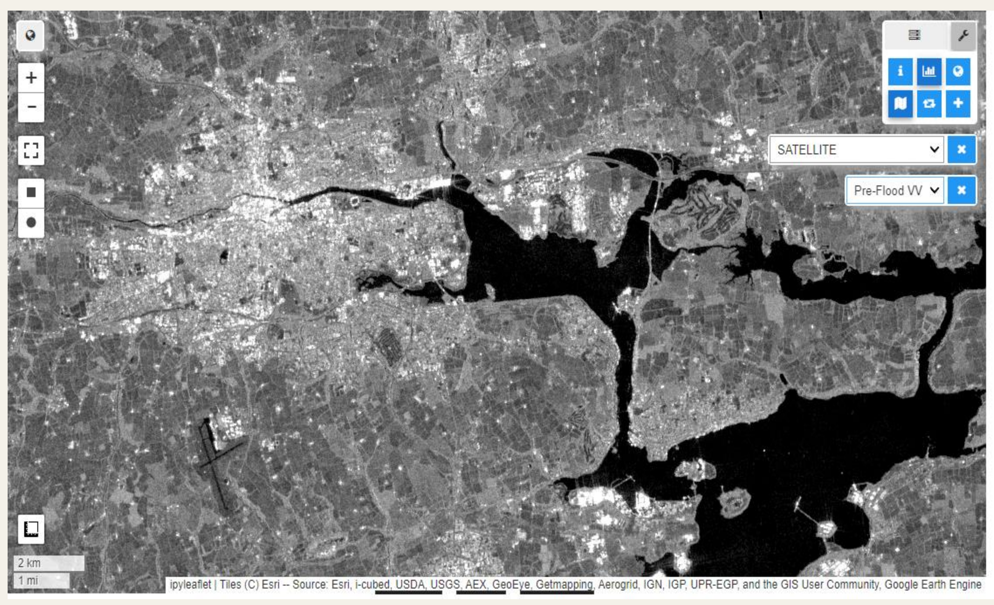

Figure 4 – Pre-flood event using Sentinel-1 Satellite Rada Data

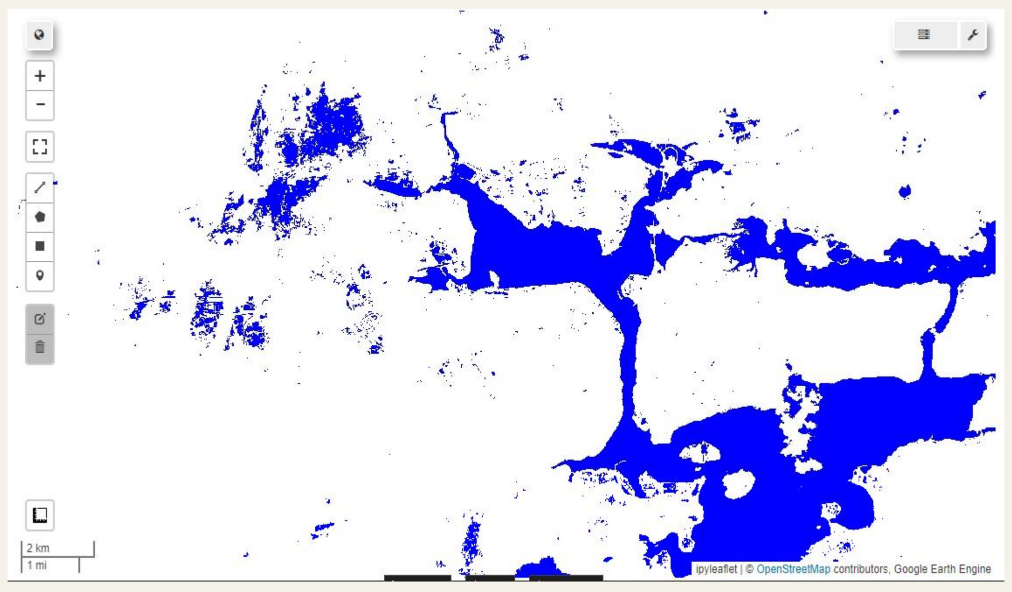

Figure 5 –NDWI layer applied to detect water post-flood event

Technologies Used

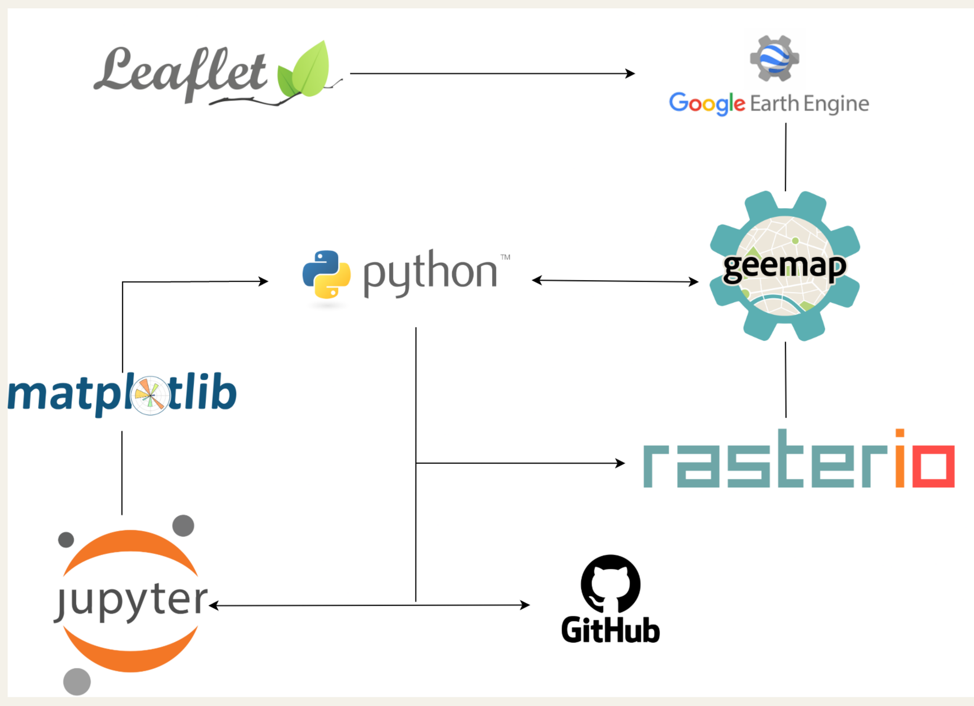

Figure 6 – Technology stack and workflow.

- GEE - Cloud based geospatial platform for processing satellite data.
- Python – Core programming language.
- Geemap – Interface between Python and Google Earth Engine .
- Matplotlib – Data visualization library.
- Rasterio – Raster data manipulation.
- Jupyter Notebook – Notebook interface for research, prototyping and analysis.
- Leaflet.js – Web mapping visualization library.
- Github – Documentation.

Validation

Accuracy was confirmed comparing flood masks derived from Sentinel-2 images with public available records from sources like JRC,CEMS, GSI,Floodinfo.

The performance of flood detection was evaluated by projecting the NDWI-derived water masks into known flood-affected areas. The areas were inspected, and a pixel-based confusion matrix was calculated to derive the metric showed below:

- True positives (TP) – Correct detected water pixels in both NDWI results and public records.
- False positive (FP) – Incorrectly detected water pixels.
- False negatives (FN) – Missed water pixels.

From this, the following were calculated:

- Precision – Ratio of predicted flood pixels that were correct.
- Recall – Ratio of actual flood pixels that were correctly detected.
- F1-score – The average of precision and recall.

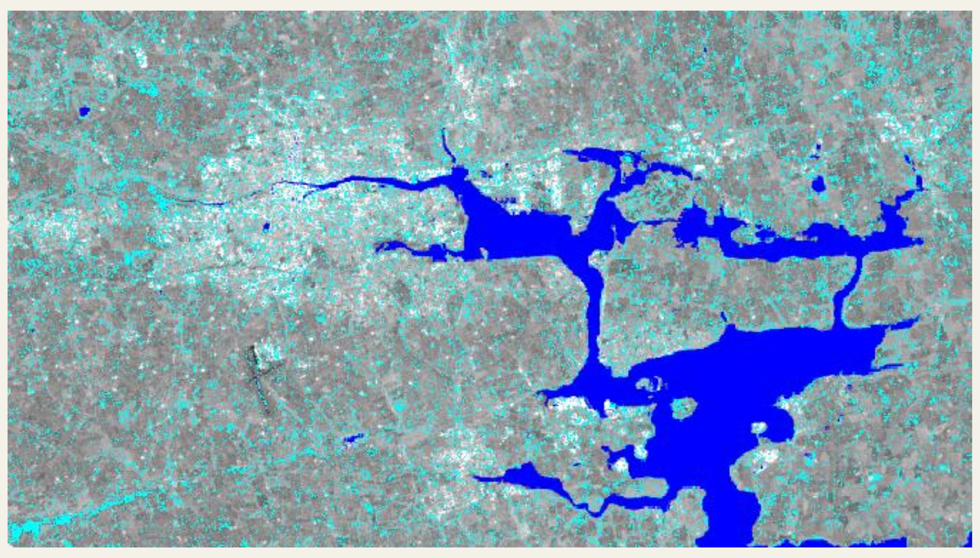

Figure 6 –Confusion Matrix layer Overlapping JRV validation layer

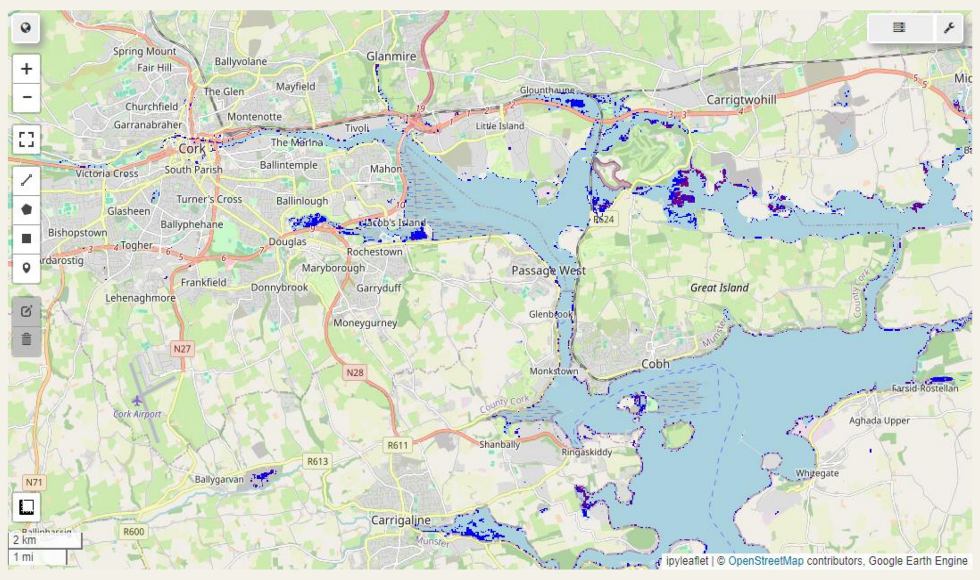

Figure 7 – Results overlaying validation data layers

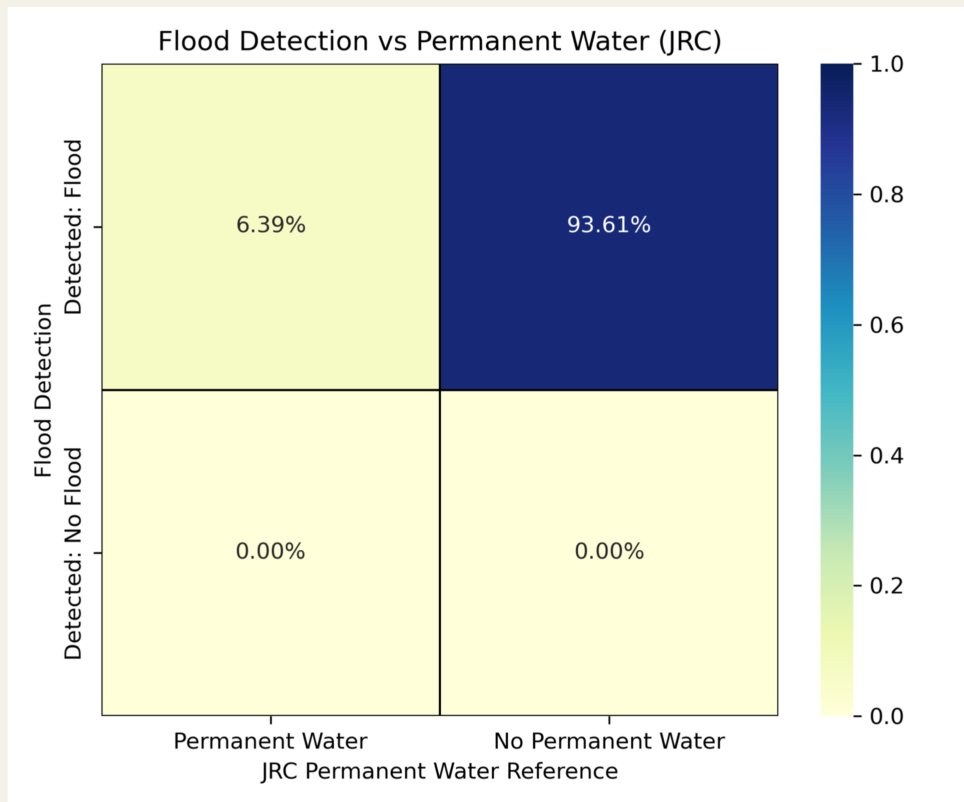

Figure 3 – Percentage Confusion Matrix Heatmap.

Conclusion

The system can

- Fetch satellite Data automate and preprocess it for a given area and time.
- Calculate NDWI values and apply a fixed threshold to produce a binary mask for areas that are likely to get flooded.
- Visualize binary flood masks in the Jupyter Notebook with geemap and Python libraries.
- Export and save results in a web-based interactive map using Leaflet.js
- Validate results and system accuracy with real-world flood data.

Acknowledgements

I would like to thank my supervisor, Andrew Beatty, for the support, guidance, and constant positivity. His presence and availability were a big pilar developing this project

I would also like to thank all the lecturers at Atlantic Technological University for every feedback and time, and my family and friends for their support throughout this whole experience.