|  |  |
| --- | --- |
| Lebanese University |  |
| Lebanese University  Faculty of Sciences – Hadath |

Submitted in partial fulfillment of the requirements for the graduate course GISD514 – Artificial Intelligence for Earth Observations at the Lebanese University.

Submitted to Dr Ali GHANDOUR

Group III

Ghinwa Bdeir

Hawraa Sweidan – 8989

Mohammad Ali Chouaib

June 6, 2025

# Multi-View Geo-Localization using Sample4Geo on the University-1652 Dataset

Progress Report

## Project Overview

This project addresses the problem of multi-view geo-localization, which is the task of predicting the location of a drone-view or street-view image by matching it with a geo-tagged satellite image. The challenge involves designing a robust image retrieval model that performs well even under domain shifts (e.g., weather variations).  
  
To align with the course project requirements, we chose to work on the University-1652 challenge (hosted on Codalab) using an existing open-source solution: Sample4Geo.

## Team Roles and Contributions

The work was divided evenly among team members:

* Repository Review: Each member reviewed at least two public GitHub codebases from the challenge suggestions and contributed to the comparative analysis.
* Codebase Selection: After evaluation, we collectively chose Sample4Geo for its simplicity, interpretability, and clean implementation structure.
* Cloning and Setup: We cloned and prepared the repository for local execution.
* Debugging and Code Analysis: Multiple iterations were done to investigate errors in data loading and model training.

## Dataset Used

We used the University-1652 dataset, properly restructured into the expected format with per-class folders and subfolders (e.g., train/0000/satellite/, train/0000/drone/). The query and gallery test sets were also separated and placed under query\_drone/ and gallery\_drone/ respectively.

## Codebase Chosen: Sample4Geo

Sample4Geo was selected because:

* It has modular design and clean integration of image encoders.
* It supports fine-tuning of advanced vision backbones like ConvNeXt.
* It offers flexibility for extensions (e.g., adding attention modules or multi-scale features).

## Current Modifications and Setup

Code Integration:

* Integrate the Sample4Geo code (including training scripts and helper modules) into our own GitHub repository.

Dataset Preparation:

* Organize the University-1652 dataset into the desired folder structure or update the configuration paths to correctly load:
  + Training images (satellite, drone, street, Google images)
  + Testing images (query and gallery splits for drone, street, and satellite views)

Initial Model Training:

* Fine-tune the pretrained ConvNeXt model on the adapted dataset and verify the geo-localization performance.
* Documentation and Troubleshooting:
* Document all modifications and tackle encountered issues (e.g., folder mismatch, authentication, and transformation warnings).

So far, the following adjustments were made:

* Dataset restructuring to match the expected folder layout.
* Minor fixes in university.py to resolve compatibility issues with the albumentations library (e.g., replacing deprecated transform arguments).
* Initial attempts to adapt training on a CPU environment (since GPU is not available locally).

## Training Status and Observed Issues

The model configuration selected was:

* Model: convnext\_base.fb\_in22k\_ft\_in1k\_384
* Input Size: (384, 384)

Tools and Libraries:

Python 3.13, PyTorch, Albumentations for data augmentation, and Hugging Face Hub for loading pretrained model checkpoints.

However, training failed due to the following errors:

* Albumentations Warnings: Invalid transform arguments (quality\_lower, always\_apply, max\_holes, etc.)
* No CUDA Device: Torch attempted to use GradScaler on CPU, resulting in AMP deactivation.
* Data Loader Empty: Despite restructuring, the dataloader returned 0 samples; investigation revealed a mismatch between expected and actual image indexing in the dataset.

## Planned Improvements (Next Steps)

* Fix albumentations transform arguments for version compatibility
* Re-index dataset samples to match expected naming conventions (0000\_000\_satellite.jpg, etc.)
* Enable CPU-compatible training mode by disabling AMP logic conditionally
* Run training for at least one epoch to generate intermediate results (e.g., loss, recall@1)
* Add evaluation logic for drone-to-satellite retrieval accuracy

## Repository

GitHub Repository: https://github.com/Haouurra/GeoAIM2\_AI4E2025 (currently under setup)