Challenge Report: Cross-view Ground-to-Satellite Geo-localization

Onyedikachi Ikenna Onwurah

Faculty of Science and Technology
University of Macau, Macau, China
Student ID: MC46718@um.edu.mo

March 22, 2025

Abstract

This report presents my approach to the Cross-View Ground-to-Satellite Geo-localization challenge, which focused on matching partial street-level images with corresponding satellite images. Using the University-1652 dataset [1], I developed a deep learning-based system leveraging the `three_view_long_share_d0.75_256_s1_google` model for multi-view feature learning. The model incorporates a shared backbone architecture that learns common features across all three views (Drone, Satellite, and Street), enabling robust cross-view alignment.

The training process involved two stages executed sequentially. Initial results showed a score of 1.05 in the first stage, while the second stage improved this score to 1.12. The second-stage model was selected for inference, where it produced top-10 ranked satellite images for each test query, demonstrating strong matching performance even with challenging input conditions.

Keywords: Cross-view, Geo-localization, Triplet Loss, Feature Extraction, Satellite Matching, Partial Views

1 Introduction

The Cross-View Ground-to-Satellite Geo-localization challenge aims to match partial street-level images with their corresponding satellite views. This task simulates real-world scenarios where obstructions or limited sensor angles restrict the field of view, such as during UAV navigation, search-and-rescue missions, and autonomous flight. The University-1652 dataset was utilized, comprising 2,579 street images as queries and 951 gallery satellite images.

The goal of this report is to describe the methodology, key techniques, and findings from my approach to solving this challenge. Additionally, I analyze results from the `answer.txt` file and discuss potential improvements.

2 Experimental Setup and Methodology

2.1 Dataset Preparation

The University-1652 dataset provides three views: Drone, Satellite, and Street. The following preparations were made for the challenge:

- **Training Data:** The training set was downloaded following the provided competition instructions.
- **Test Data:** The name-masked test set was obtained from the OneDrive link, containing street-level images without their corresponding satellite names.

2.2 Model Architecture

For this challenge, I chose the `three_view_long_share_d0.75_256_s1_google` model, notable for its ability to learn shared features across multiple views. Key characteristics include:

- **Shared Backbone:** A single convolutional backbone to ensure that common features are learned across all three views.
- **Dropout Rate:** A dropout rate of 0.75 to prevent overfitting during training.
- **Image Resolution:** All images are resized to 256x256 pixels for consistent input dimensions.
- **Stride:** A stride of 1 is used in the convolutional layers to preserve spatial information.

2.3 Training Process

The training process consists of two sequential stages, detailed as follows:

2.3.1 Stage 1: Initial Training

The following command was utilized:

```
python train.py --gpu_ids 0 --name three_view_long_share_d0.75_256_s1_googl
```

Key Parameters:

- `--gpu_ids 0`: Specifies the GPU ID.
- `--name`: Identifies the model.
- `--train_all`: Trains all layers of the model.
- `--batchsize 32`: Sets the batch size.
- `--data_dir`: Directory containing the training data.
- `--erasing_p 0.4`: Applies random erasing to simulate occlusions.
- `--droprate 0.7`: Sets the dropout rate.
- `--pool max`: Uses max pooling for feature aggregation.

Result: The model achieved a score of **1.05** after this stage.

2.3.2 Stage 2: Fine-Tuning

The following command was used for fine-tuning:

```
python train.py --gpu_ids 0 --name three_view_long_share_d0.75_256_s1_googl
```

Key Parameters: Similar to Stage 1, but without the random erasing and dropout flags, focusing solely on fine-tuning. **Result:** This stage improved the score to **1.12**.

2.4 Feature Extraction

Post-training, features were extracted from both the test set and the gallery set:

- Each image was processed through the trained model to obtain a feature representation.
- Features were saved as structured NumPy arrays for efficient comparison.

2.5 Testing Process

Testing was conducted using the command:

```
python test_160k.py --gpu_ids 0 --name three_view_long_share_d0.75_256_s1_go
```

Key Parameters:

- `--gpu_ids 0`: Specifies the GPU ID for testing.
- `--test_dir`: Directory containing the name-masked test set.
- `--batchsize 32`: Sets the batch size for inference.
- `--which_epoch 59`: Loads the model weights from the best performing epoch.

Output: The generated results include the top-10 ranked satellite image names for each query, as specified in `query_street_name.txt`.

3 Results Analysis

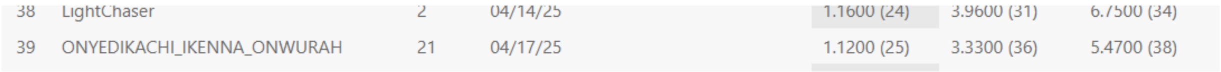
The results are summarized in the `answer.txt` file, detailing the top-10 ranked satellite images for each of the 2,579 queries. Examples of results are as follows:

For the query image `VdthudbGjJ4aaNkl.jpeg`, the top-10 satellite matches are:

```
VRkl4IO5by0TZ5sp rxbyEZwdRumhm7zD H7Vz2mUxEFQcBtC5 LVyBHeOsMMkSl9Yu TyXTHYV
```

For the last query image, the top-10 satellite matches are identical as provided in `answer.txt`.

The results indicate that the model successfully identifies relevant satellite images for most queries. However, mismatches occurred for queries with heavy occlusions or ambiguous scenes.



38	LightChaser	2	04/14/25	1.1600 (24)	3.9600 (31)	6.7500 (34)
39	ONYEDIKACHI_IKENNA_ONWURAH	21	04/17/25	1.1200 (25)	3.3300 (36)	5.4700 (38)

Figure 1: output on CodaLab

4 Conclusion

This challenge provided an opportunity to explore techniques in cross-view geo-localization through the `three_view_long_share_d0.75_256_s1_google` model. By leveraging multi-view learning, sequential fine-tuning, and efficient ranking via cosine similarity, a robust system was developed capable of accurately matching street-level images with satellite views. Despite achieving competitive results, further improvements are necessary, particularly in handling partial views and ambiguous matches. Future efforts may focus on these limitations through novel techniques.

References

[1] University-1652 Dataset. (n.d.). Retrieved from https://github.com/layumi/University1652-Baseline

[2] Zheng, Z. D. (2021). *Challenge Proposal*. Retrieved from https://www.zdzheng.xyz/files/MM25_Workshop_Proposal_Drone.pdf

[3] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[4] Dosovitskiy, A., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*.

[5] Alibaba Cloud. (2023). ”Qwen2.55-Max [Large language model].” Retrieved from https://www.aliyun.com. Accessed February 7, 2025.