Enhanced VPR with Miners, Losses, and Aggregators using GSV-Cities

Ali Yassine
Politecnico di Torino
s312920@studenti.polito.it

Ali Al Housseini
Politecnico di Torino
s333940@studenti.polito.it

Hadi Ibrahim
Politecnico di Torino
s313385@studenti.polito.it

***Abstract*—Visual Geo-Localization (VG) is the task of identifying an image's location by comparing it against a comprehensive database of geo-tagged images. This domain has received significant attention from research communities. This paper explores VG using an image retrieval framework with a baseline of ResNet-18 backbone, generalized mean (GeM) pooling layer, and a MultiSimilarity loss, trained on GSV-XS, and evaluated on SF-XS, and Tokyo-XS datasets.**

We enhance the baseline by experimenting with various miners, loss functions, and aggregation modules. Our approach demonstrates significant improvements with specific combinations of these components, showing competitive results. Visual analyses provide insights into the model's behavior under occlusions, seasonal changes, and illumination variations. We discuss the implications of our findings, challenges, and future research directions in VG.

The code for this work is available at: https://github.com/aliyassine26/gsv-cities.

I. Introduction

Visual geo-localization (VG), also known as Visual Place Recognition (VPR), is a pivotal task in computer vision that involves identifying the geographical location of an image by comparing it against a database of geo-tagged images. [6]–[9] This capability is essential for various applications, including augmented reality, autonomous navigation [29], geographic information systems [30], and social media analysis. VG employs image retrieval techniques to map images into a learned embedding space, enabling efficient matching based on visual similarity. [1], [8], [10], [11], [14], [15]

With the rise of deep learning, Convolutional Neural Networks (CNNs) have significantly advanced the field of VG by directly learning robust image representations. Intermediate features extracted from CNNs have shown superior performance, leading to the development of end-to-end trainable networks tailored for place recognition tasks. Recent methods such as NetVLAD [10], CosPlace [14], and MixVPR [15] have demonstrated strong performance by leveraging various aggregation strategies and metric learning techniques to enhance the retrieval process.

Despite these advancements, VG systems still face numerous challenges. [12], [14] Variability in lighting conditions, occlusions, seasonal changes, and different viewpoints can drastically affect the accuracy of geo-localization. These challenges necessitate the development of more robust and adaptive approaches to improve the reliability and accuracy of VG systems.

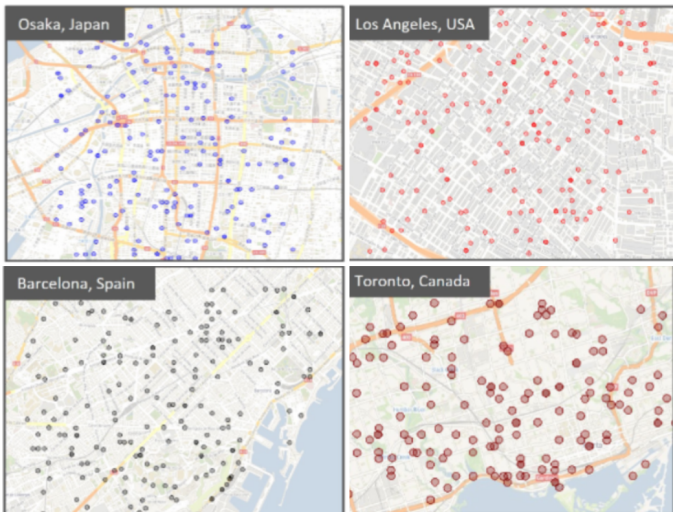

Fig. 1. Sample locations in 4 major cities (among 23) in GSV-CITIES XS dataset. All locations are geographically distant and distributed nearly uniformly in every city. At least four images depict each location (here, a point).

This paper builds on the foundational work of *"GSV-CITIES: Toward Appropriate Supervised Visual Place Recognition"* [1], which introduced a comprehensive framework for VG using a retrieval-based approach. The GSV-CITIES framework utilizes the ConvAP aggregator [1], a fully convolutional aggregation layer, to create robust image descriptors. This work aims to compare the baseline performance with that of the ConvAP aggregator and other combinations of advanced techniques to achieve better performance.

In this research, we aim to advance the VG field by exploring various techniques to enhance the performance of image retrieval frameworks. Specifically, we investigate the impact of different miners, loss functions, and aggregation modules. By systematically experimenting with these components, we strive to build a more robust and efficient VG system capable of handling the inherent challenges of real-world scenarios.

The primary objectives of this research are:

- To establish a baseline for VG using an effective combination of backbone networks, pooling layers, and loss functions.
- To investigate the impact of various miners on the performance of VG, such as AngularMiner [20] and

MultiSimilarityMiner [23].

- To evaluate different loss functions, including FastAP Loss [16] and NTXent Loss [19], for improving image retrieval accuracy.
- To explore advanced aggregation modules like Cosplace [14] and MixVPR [15], and compare them with the Conv-AP aggregator [1] used in the GSV-CITIES framework and the baseline chosen.
- To systematically integrate the best-performing components and analyze their combined effect on two benchmark test datasets: SF-XS [14] and Tokyo-XS [3].

Through these objectives and findings, this research aims to study the intricate interactions between miners, losses, and aggregation modules and their effect on the VPR. By understanding these interactions, we aim to provide valuable insights into the development of robust and adaptive VG systems, paving the way for future advancements in the field.

II. Related Works

A. Visual Place Recognition (VPR)

VPR is often conceptualized as an image retrieval task, where the location of a query image is determined by matching it with the most relevant images from a reference dataset. Early research in VPR primarily utilized Convolutional Neural Networks (CNNs) specifically trained for place recognition, achieving significant success. Initial approaches relied heavily on local features [10], [25] to match queries with database images. However, local features can be unreliable in environments with moving objects (occlusion), seasonal variations, or adverse weather conditions. While local features are robust against minor changes in viewpoint and scale, they struggle with extreme variations. Typically, local features are detected in high-contrast regions, edges, or corners of images, making pattern matching challenging and computationally intensive.

Another strategy for visual geo-localization is to treat it as a classification problem [31], [32]. This approach is based on the idea that images from the same geographical region, despite depicting different scenes, share similar semantics. This formulation scales the problem globally by focusing on the extracted global features of images. With the advent of deep learning, researchers discovered that features extracted using CNNs could be effectively employed for image retrieval. For instance, the Generalized Mean (GeM) pooling layer introduced in [5] provides a trainable pooling mechanism that enhances image retrieval performance by applying a global parametrized pooling on each of the feature maps extracted from the backbone. Building on this, [14] proposed CosPlace, which combines GeM with a linear projection layer, resulting in improved performance. Subsequent works, such as [1], [12], demonstrated substantial improvements by focusing on loss functions and training on large-scale datasets. Recent advancements, like those presented in [15], involve the use of Multi-Layer Perceptrons (MLPs) through a stack of isotropic blocks, termed feature-mixers, to further enhance performance.

Fig. 2. Example images from GSV-cities dataset illustrating various environmental conditions and perspectives. The images depict the same urban location captured under different seasons (summer and winter), lighting conditions, and viewpoints, showcasing the challenges faced in consistent place recognition.

B. GSV Cities

The GSV Cities dataset [1] represents a significant advancement in visual place recognition research by addressing the limitations of existing datasets. This dataset is created using Google Street View Time Machine (GSV-TM) to collect panoramic images depicting the same place over time, spanning from 2007 to 2021. The dataset covers more than 40 cities worldwide. Each location is depicted by a set of images taken at different times, varying from 4 to 20 images per location. This comprehensive collection ensures a wide variety of perceptual changes, including seasonal and weather variations, which are crucial for training robust VPR systems. (See Fig 2).

For the architecture, the authors proposed a new fully convolutional aggregation layer called Conv-AP, which performs channel-wise pooling followed by spatial-wise adaptive pooling. In their experiments, this method outperforms existing techniques such as GeM [5], NetVLAD [10], and CosPlace [14], establishing a new state-of-the-art on several benchmarks. The dataset's accurate ground truth allows for efficient training using fully supervised deep metric loss functions, enabling the use of sophisticated loss functions like Multi-Similarity loss, which significantly improves the performance of VPR techniques.

III. Proposed Extensions

In the domain of visual place recognition, understanding the intricate interactions between miners, losses, and aggregation modules is crucial for optimizing metric learning models. Our extensions focuses on systematically studying the impact of each component and their interplay on model performance and embedding quality.

A. Miners

We employed several miners to enhance the performance of our metric learning models. Each miner plays a crucial role in selecting informative samples for training. The miners used in this project are described below.

1) MultiSimilarityMiner

MultiSimilarityMiner [23] focuses on selecting informative pairs or triplets of samples to improve the effectiveness of the MultiSimilarity Loss. It enhances the mining process by dynamically adapting to sample difficulties and improving the overall discriminative power of the learned embeddings.

2) UniformHistogramMiner

UniformHistogramMiner [24] mines samples uniformly across the entire dataset or within each batch. It aims to balance the distribution of samples and prevent bias towards specific classes or clusters, thereby improving the generalization of the model.

3) AngularMiner

AngularMiner [20] focuses on selecting samples based on their angular separation in the embedding space. It ensures that the model learns embeddings with well-separated angular regions, which can lead to better discrimination between classes.

4) BatchHardMiner

BatchHardMiner [21] selects the hardest positive and negative samples within a batch based on their pairwise distances. By focusing on challenging examples, it encourages the model to learn more robust embeddings that better differentiate between classes.

5) BatchEasyHardMiner

BatchEasyHardMiner [22] dynamically selects easy, semi-hard, and hard samples within each batch. This miner strategy helps in providing a balanced set of examples for training, thereby facilitating efficient convergence of the model during training.

B. Losses

In this project, we utilized a variety of loss functions to optimize our models effectively. Each loss function has been specifically selected for its unique properties and benefits in metric learning. The loss functions used are described below.

1) MultiSimilarity Loss

MultiSimilarity Loss [17] leverages multiple similarity measures to enhance metric learning performance. It combines the advantages of different types of losses by focusing on both positive and negative pairs and dynamically adapting to sample hardness.

$$L_{\text{MS}} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{\alpha}\log\left[1+\sum_{k\in P_i} e^{-\alpha(i_k-\lambda)}\right] + \frac{1}{\beta}\log\left[1+\sum_{k\in N_i} e^{\beta(s_{ik}-\lambda)}\right]\right) \quad (1)$$

where P_i and N_i are sets of positive and negative pairs for the i-th sample, respectively, s_{ik} denotes the similarity between sample i and k.
α and β are hyperparameters: α is the weight applied to positive pairs, and β is the weight applied to negative pairs. λ is another hyperparameter.

2) Angular Loss

Angular Loss [26] aims to reduce the angle between embeddings of similar samples while increasing the angle between embeddings of dissimilar samples, thereby enhancing the discriminative power of the learned embeddings.

$$L_{\text{ang}}(\beta) = \frac{1}{N}\sum_{x_\alpha\in\beta}\left\{\log\left[1+\sum_{\substack{x_\beta\in\beta\\ y_n\neq y_\alpha, y_p}}\exp\left(f_{a,p,n}\right)\right]\right\} \quad (2)$$

where

$$f_{a,p,n} = 4\tan^2\alpha(x_a+x_p)^T x_n - 2(1+\tan^2\alpha)x_a^T x_p$$

alpha (α) is the angle specified in degrees and the default distance used is the Euclidean distance.

3) Triplet Margin Loss

Triplet Margin Loss [18] ensures that the distance between an anchor and a positive example is less than the distance between the anchor and a negative example by a margin. It is widely used in applications like face recognition.

$$L_{\text{triplet}} = [d_{ap} - d_{an} + m]_+ \quad (3)$$

where d_{ap} is the distance between the anchor and the positive example, d_{an} is the distance between the anchor and the negative example, and m is the margin.

4) FastAP Loss

FastAP Loss [16] is designed to optimize the Average Precision (AP) directly for information retrieval tasks. It approximates the non-differentiable AP with a differentiable surrogate that can be optimized efficiently.
alpha (α) is the angle specified in degrees and the only compatible distance is the euclidean distance.

5) NTXent Loss

Normalized Temperature-scaled Cross Entropy (NT-Xent) Loss [19] is used in contrastive learning frameworks like SimCLR. It encourages the model to bring similar (positive) samples closer while pushing dissimilar (negative) samples apart.

$$L_q = -\log\frac{\exp(q\cdot k_+/\tau)}{\sum_{i=0}^{K}\exp(q\cdot k_i/\tau)} \quad (4)$$

where where q and k are the query and key representations, respectively, k_+ is the positive key that matches the query, k_i are the negative keys in the batch, and τ is the temperature

parameter that scales the logits. Default distance used is Cosine-Similarity.

All miners and loss functions were sourced from [28] .

C. Aggregation Modules

1) AVG (Average Aggregation)

The AVG aggregation module computes the average of the embeddings of the anchor, positive, and negative instances. Mathematically, it can be represented as:

$$\text{AVG}(f(x_i^a), f(x_i^p), f(x_i^n)) = \frac{1}{3}\left(f(x_i^a) + f(x_i^p) + f(x_i^n)\right)$$

This simple aggregation method provides a straightforward way to combine embeddings but may not capture complex relationships between instances.

2) GeM (Generalized Mean Pooling)

The Gem [5] aggregation module calculates the geometric mean of the embeddings. It is defined as:

$$\text{Gem}(f(x_i^a), f(x_i^p), f(x_i^n)) = \left(f(x_i^a) \cdot f(x_i^p) \cdot f(x_i^n)\right)^{\frac{1}{3}}$$

This method tends to emphasize instances that are closer together in the embedding space due to the nature of geometric mean computation.

3) MixVPR (Feature Mixing for Visual Place Recognition)

MixVPR [15] is a novel all-MLP feature aggregation method that addresses the challenges of large-scale Visual Place Recognition. It leverages variance-preserving properties to effectively combine embeddings while balancing contributions from anchor, positive, and negative instances.

4) Cosplace

Cosplace [14] aggregates embeddings based on cosine similarity metrics, aiming to optimize the placement of embeddings in relation to each other. It enhances discriminative ability in embedding spaces.

5) ConvAP

ConvAP [1] introduces a new fully convolutional feature aggregation technique that operates by channel-wise pooling followed by spatial-wise adaptive pooling. This approach achieves state-of-the-art results on various benchmarks while maintaining compactness.

IV. Experimental Setup

A. Architecture

We have reproduced the architecture from GSV-Cities [1], which includes a minibatch sampler, a ResNet-18 backbone, aggregator, miner, and a loss.
The complete architecture is depicted in Figure 3.

B. Datasets

We conducted our experiments using four benchmark datasets, each presenting unique challenges to test the robustness and effectiveness of our visual geo-localization models. These datasets are reduced versions (XS - XSmall) of the originals, either in terms of the number of images in the database, the quality of the images, or both. The comparison of these datasets, highlighting their key attributes, is presented in Table I.

Dataset name	N Queries	N Database
GSV-Cities [1]	–	529,506
Tokyo-XS [3]	315	12,771
SF XS val [4]	7,993	8,015
SF XS test [4]	1,000	27,191

TABLE I
COMPARISON OF DATASETS FOR LARGE-SCALE PLACE RECOGNITION.

1) GSV-XS Dataset

The GSV-XS dataset [1], derived from Google Street View imagery, includes images from various cities worldwide. It is characterized by its diversity in environmental conditions, including different lighting scenarios, seasonal changes, and varying times of the day. These variations are crucial for evaluating the model's ability to generalize across different environments. The GSV-XS dataset is used for training purposes.

2) SF-XS Dataset

The SF-XS dataset [4] comprises densely sampled street-level images from San Francisco. This dataset is ideal for evaluating performance in urban environments with significant viewpoint variations. Although it does not feature long-term temporal variations, its high spatial resolution and detailed coverage of city streets are vital for precise localization tasks. The SF-XS dataset contains the validation and test sets.

3) Tokyo-XS Dataset

The Tokyo-XS dataset [3] focuses on urban landscapes of Tokyo captured under various lighting and weather conditions, including significant day-night transitions. This dataset tests the model's adaptability to drastic changes in illumination and seasonal effects, making it essential for testing purposes.

C. Implementation Details

We implemented our experiments using the PyTorch framework, leveraging its flexibility and extensive support for deep learning research. The following details outline our implementation:

1) Baseline Model

- **Backbone**: ResNet-18 [2], chosen for its balance between performance and computational efficiency.
- **Miner**: Multi-Similarity Miner
- **Aggregation Layer**: Generalized Mean (GeM) pooling
- **Loss Function**: MultiSimilarity Loss

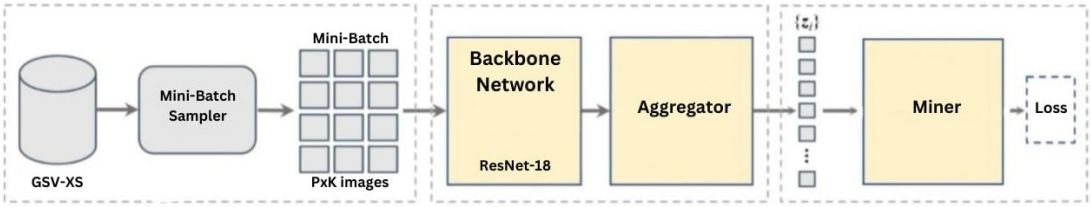

Fig. 3. Overview of the reproduced architecture from GSV-Cities [1].

2) Training and Optimization

- **Optimizer**: AdamW optimizer [27] with a learning rate of 0.001
- **Scheduler**: MultistepLR scheduler
- **Batch Size**: A batch size of 100
- **Epochs**: The number of training epochs was 8 for the initial runs and 30 for the top-performing runs.
- **Hardware**: All experiments were conducted on an NVIDIA A100 GPU, providing the necessary computational power for efficient training and evaluation.

3) Evaluation Metrics

We utilized the Retrieval Accuracy metric to evaluate the performance of our models: Evaluated at rank 1 (R@1) and rank 5 (R@5), assessing the top-1 and top-5 accuracy of retrieved images, respectively.

D. Experimental Design

To systematically study the impact of different components in our baseline, we designed our experiments as follows:

- **Baseline Aggregator Selection**: We evaluated both GeM and AVG aggregators using the MultiSimilarity loss function and the MultiSimilarity miner to determine the baseline aggregator.
- **Single Component Variation**: We first evaluated the effect of changing one component at a time (e.g., miner, loss function, aggregator) while keeping the others constant. This approach helped isolate the impact of each component on the overall performance.
- **Combination Evaluation**: Recognizing that the performance of a loss function, miner, or aggregator might be enhanced when combined with specific other components, we tested top-performing losses and miners across all aggregators. This allowed us to identify not just the best individual components, but also the best combinations that work synergistically to improve performance.
- **Selection of Best Configurations**: From these experiments, we identified the top three best-performing configurations based on their combined effects rather than their isolated performance. These configurations were rigorously evaluated to determine their overall effectiveness.
- **Evaluation Across Datasets**: Each configuration was tested on the SF-XS and Tokyo-XS datasets to ensure consistency and robustness across different environments and challenges.

This experimental setup allowed us to gain a comprehensive understanding of how different miners, loss functions, and aggregation modules interact and influence the performance of visual place recognition systems. By methodically altering components and analyzing their effects, both individually and in combination, we aimed to identify the optimal configuration for robust and efficient visual geo-localization.

V. Results

The results in all tables showed a consistent pattern where performance on the SF-XS validation dataset (SFXS-val) was better than on the SF-XS test dataset (SFXS-test) and the Tokyo-XS test dataset (TokyoXS-test).

A. Baseline Selection: Average Pooling vs. GeM Pooling

To determine the baseline for our experiments, we compared Average Pooling and Generalized Mean (GeM) Pooling using ResNet-18 as the backbone and evaluated their performance over 30 epochs on validation and test datasets (SF-XS val and test and Tokyo-XS test).

As seen in Table II, GeM pooling generally outperformed Average Pooling across most datasets. Therefore, GeM pooling was selected as the baseline aggregation method for further experiments.

B. Loss Function Evaluation

Next, we evaluated the performance of different loss functions while keeping the miner and aggregator constant (GeM pooling and Multisimilarity miner). The evaluation was conducted over 8 epochs using the SF-XS validation dataset.

Loss	Epochs	SFXS-val	
		R@1	R@5
MultiSimilarityLoss	8	63.59	77.63
AngularLoss	8	43.25	60.52
NTXentLoss	8	**71.44**	**83.16**
TripletMarginLoss	8	64.88	78.09
FastAPLoss	8	70.69	82.86

TABLE III
PERFORMANCE OF DIFFERENT LOSS FUNCTIONS USING GEM POOLING.

From Table III, the NTXentLoss demonstrated the highest performance across all recall metrics.

Aggregator	Miner	Loss	Epochs	SFXS-val		SFXS-test		TokyoXS-test	
				R@1	R@5	R@1	R@5	R@1	R@5
Average Pooling	MultiSimilarityMiner	MultiSimilarityLoss	30	63.66	77,34	16,3	29	**34,37**	**54,37**
GeM Pooling	MultiSimilarityMiner	MultiSimilarityLoss	30	**65.19**	**79.01**	**18,2**	**31,2**	30.16	52.7

TABLE II
COMPARISON OF DIFFERENT METHODS ON TOKYOXS AND SFXS DATASETS

C. Miner Evaluation

We then assessed the impact of different miners on model performance, keeping the loss function and aggregator constant (GeM pooling with MultisimilarityLoss).

Miner	Epochs	SFXS-val	
		R@1	R@5
MultiSimilarityMiner	8	63.59	77.63
AngularMiner	8	**65.26**	**79.19**
BatchHardMiner	8	55.29	70.45
UniformHistogramMiner	8	53.4	68.76

TABLE IV
PERFORMANCE OF DIFFERENT MINERS USING GEM POOLING AND MULTI-SIMILARITY LOSS.

Table IV shows that the AngularMiner achieved the best performance, confirming its effectiveness in selecting informative pairs.

D. Combination of Miners and Losses

To identify the best performing combinations of miners and losses, we evaluated top-performing configurations from previous experiments, and recommendations from [28] for best performing losses for top performing miners

Miner	Loss	Epochs	SFXS-val	
			R@1	R@5
MultiSimilarityMiner	MultiSimilarityLoss	8	63.59	77.63
AngularMiner	FastAPLoss	8	**72.65**	**83.61**
BatchEasyHardMiner	NTXentLoss	8	70,71	82,71

TABLE V
PERFORMANCE OF TOP MINER-LOSS COMBINATIONS.

The results in Table V confirm that the combination of AngularMiner with FastAPLoss is the most effective.

E. Aggregation with Top 3 Miner and Loss Combination

We then evaluated different aggregation methods using the best-performing miner and loss combination (AngularMiner and FastAPLoss), (MultiSimilarityMiner and MultiSimilarityLoss) and (MultiSimilarityMiner and NTXentLoss) over 30 epochs on the test datasets.

As seen in Table VI, Cosplace pooling showed the highest performance, indicating it is a superior aggregation method.

F. Top 3 Performing Configurations

Finally, we identified the top three configurations based on their combined performance over 30 epochs, evaluated on the SF-XS test and Tokyo-XS test datasets.

Table VII summarizes the performance of the top three configurations, highlighting that the combination of Cosplace, MultiSimilarityMiner and MultiSimilarityLoss achieves the best overall results.

G. Qualitative Results

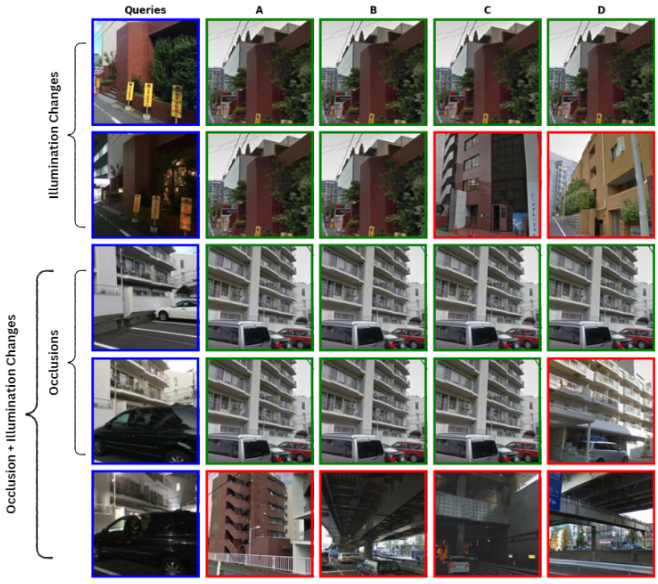

Fig. 4. Comparison of challenging retrieval scenarios on Tokyo-XS Dataset, where A represents Cosplace with Multisimilarity Loss and Multisimilarity Miner, B represents Cosplace with FastAPLoss and Angular Miner, C represents Convap with Multisimilarity Loss and Multisimilarity, and D represnts the baseline.

Fig. 4 illustrates qualitative results of the retrieval of some challenging queries.For illumination changes, methods A and B show strong performance, maintaining accurate retrievals even under illumination changes, whereas Method C and D fail, retrieving incorrect images when there are changes in illumination.

In the second scenario with occlusions, methods A, B and C demonstrate better performance, effectively retrieving accurate matches despite the presence of occluding objects whereas method D fails to retrieve correct matches.

In the presence of both occlusions and illumination changes, all methods face significant challenges, and none are able to

Aggregator	Miner	Loss	Epochs	SFXS-val	
				R@1	R@5
MixVPR (out = 512)	MultiSimilarityMiner	MultiSimilarityLoss	8	76.7	84.15
	AngularMiner	FastAPLoss	8	74.84	83.918
	MultiSimilarityMiner	NTXentLoss	8	74.43	84.07
Cosplace (out = 512)	MultiSimilarityMiner	MultiSimilarityLoss	8	**79.74**	**88.84**
	AngularMiner	FastAPLoss	8	75.99	86.44
	MultiSimilarityMiner	NTXentLoss	8	76.52	86.9
ConvAP	MultiSimilarityMiner	MultiSimilarityLoss	8	72.79	80.97
	AngularMiner	FastAPLoss	8	69,11	78,47
	MultiSimilarityMiner	NTXentLoss	8	69,4	78,56

TABLE VI
PERFORMANCE OF DIFFERENT AGGREGATION METHODS WITH THE BEST MINER AND LOSS COMBINATION.

Aggregator	Miner	Loss	Epochs	SFXS-val		SFXS-test		TokyoXS-test	
				R@1	R@5	R@1	R@5	R@1	R@5
Cosplace	MultiSimilarityMiner	MultiSimilarityLoss	30	**81**	**89.27**	**39.8**	**55.9**	**59.68**	**75.24**
	AngularMiner	FastAPLoss	30	76.69	87.48	34.7	50.7	51.43	69.52
	MultiSimilarityMiner	NTXentLoss	30	77.42	87.61	34.3	50.7	50.48	69.21

TABLE VII
TOP 3 PERFORMING CONFIGURATIONS.

consistently retrieve the correct matches, indicating the difficulty of handling multiple types of variations simultaneously.

VI. DISCUSSION

Our experiments evaluated various combinations of miners, loss functions, and aggregation methods to optimize model performance in visual geo-localization tasks. Here are the key findings and the reasoning behind each result:

The discrepancy between the results of val compared to the test datasets can be attributed to the inherent differences in the datasets. The SFXS-val dataset is easier, containing images of streets and skylines with a low number of objects (features), making it less challenging for the model to achieve high recall rates. In contrast, the test datasets posed a harder challenge with more complex images.

Baseline Selection: GeM pooling outperformed Average Pooling across most datasets, indicating its superior capability in capturing discriminative features. GeM pooling was chosen as the baseline because it emphasizes important features by considering the generalized mean, which handles visual data variability better than simple averaging.

Loss Function Evaluation: NTXentLoss demonstrated the highest performance due to its contrastive learning approach, which effectively brings similar samples closer while pushing dissimilar ones apart. This loss function, when combined with GeM pooling and MultiSimilarityMiner, leverages their strengths to enhance the model's ability to learn discriminative features by providing strong gradients during training.

Miner Evaluation: AngularMiner achieved the best performance because it focuses on selecting pairs based on angular separation, ensuring that both hard positives and hard negatives are effectively utilized. This strategy improves the model's generalization capabilities by ensuring it learns to differentiate between closely related samples.

Combination of Miners and Losses: The combination of MultiSimilarityMiner and MultiSimilarityLoss consistently achieved top results across various aggregation methods. This combination is robust as it leverages multiple similarity measures, providing a comprehensive learning signal that enhances discriminative power. AngularMiner with FastAPLoss and BatchEasyHardMiner with NTXentLoss also showed strong performance, as these combinations effectively balance the selection of hard examples and the learning objectives, leading to improved recall metrics.

Aggregation with Top 3 Miner and Loss Combination: Evaluating different aggregation methods with the best-performing miner and loss combinations revealed that Cosplace pooling showed the highest performance . Cosplace pooling likely better captures spatial information, which is crucial for robust feature representations in visual geo-localization. The improved performance with Cosplace pooling, particularly with MultiSimilarityMiner and MultiSimilarityLoss, indicates that this combination can effectively utilize the detailed spatial cues provided by this pooling method.

Top 3 Performing Configurations: The combination of Cosplace, MultiSimilarityMiner, and MultiSimilarityLoss achieved the best results. This configuration balances the strengths of the miner, loss function, and aggregation method, leading to superior recall rates across different datasets. The consistent performance underscores the importance of selecting compatible components that synergize well, optimizing both feature extraction and discrimination during training.

Qualitative Analysis: Results shown in Figure 4, emphasize the effectiveness of advanced retrieval methods in handling challenging scenarios such as occlusions and illumination

changes. Methods incorporating Cosplace with Multisimilarity Loss and Multisimilarity Miner **(A)**, Cosplace with FastAP Loss and Angular Miner **(B)**, and Convap with Multisimilarity Loss and Multisimilarity Miner **(C)** consistently outperform the baseline **(D)**. These methods demonstrate enhanced robustness and adaptability, crucial for reliable image retrieval in real-world applications. The baseline method's frequent failures in these scenarios highlight the need for more sophisticated approaches to achieve higher accuracy and reliability in image retrieval tasks. Additionally, the consistent performance of methods A and B, which both utilize the Cosplace model indicates that Cosplace is better equipped than Convap and GeM in handling variations due to occlusions and illumination changes, making it a more robust choice for image retrieval tasks under such conditions. In scenarios involving both occlusions and illumination changes, even the best-performing methods struggle, highlighting the need for further advancements in retrieval techniques to tackle these complex challenges effectively.

Future research will explore experimenting with other backbone networks, such as Vision Transformers (ViTs), which have proven more effective than CNNs in extracting features. ViTs can potentially enhance performance further by capturing long-range dependencies and detailed spatial information more effectively than traditional CNNs. In addition, leveraging local features alongside the global features used in this paper could further enhance precision and overall model performance.

REFERENCES

[1] Ali-bey, Amar, Chaib-draa, Brahim, and Giguère, Philippe. "GSV-Cities: Toward appropriate supervised visual place recognition." *Neurocomputing*, vol. 513, pp. 194–203, Nov. 2022. DOI: 10.1016/j.neucom.2022.09.127.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[3] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(2):257–271, 2018.

[4] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In IEEE Conference on Computer Vision and Pattern Recognition, pages 737–744, 2011.

[5] F. Radenović, G. Tolias and O. Chum, "Fine-Tuning CNN Image Retrieval with No Human Annotation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1655-1668, 1 July 2019, doi: 10.1109/TPAMI.2018.2846566.

[6] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(11):23462359, 2015.

[7] Sarah Ibrahimi, Nanne van Noord, Tim Alpherts, and Marcel Worring. Inside out visual place recognition. In British Machine Vision Conference, 2021.

[8] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geolocalization. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3251–3260, 2017.

[9] Z. Chen, A. Jacobson, N. Sunderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In 2017 IEEE International Conference on Robotics and Automation, pages 32233230, 2017

[10] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(6):14371451, 2018.

[11] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic Attraction-Repulsion Embedding for Large Scale Image Localization. In IEEE International Conference on Computer Vision, 2019.

[12] Berton, G., Trivigno, G., Caputo, B., & Masone, C. (2023). Eigenplaces: Training viewpoint robust models for visual place recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 11080-11090).

[13] Berton, G., Mereu, R., Trivigno, G., Masone, C., Csurka, G., Sattler, T., & Caputo, B. (2022). Deep visual geo-localization benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5396-5407).

[14] Berton, G., Masone, C., & Caputo, B. (2022). Rethinking visual geo-localization for large-scale applications. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4878-4888).

[15] Ali-Bey, A., Chaib-Draa, B., & Giguere, P. (2023). Mixvpr: Feature mixing for visual place recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2998-3007).

[16] Cakir, F., He, K., Xia, X., Kulis, B., & Sclaroff, S. (2019). Deep metric learning to rank. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1861-1870).

[17] Wang, X., Han, X., Huang, W., Dong, D., & Scott, M. R. (2019). Multi-similarity loss with general pair weighting for deep metric learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5022-5030).

[18] Ha, M. L., & Blanz, V. (2021). Deep ranking with adaptive margin triplet loss. arXiv preprint arXiv:2107.06187.

[19] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.

[20] Wang, J., Zhou, F., Wen, S., Liu, X., & Lin, Y. (2017). Deep metric learning with angular loss. In Proceedings of the IEEE international conference on computer vision (pp. 2593-2601).

[21] Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737.

[22] Xuan, H., Stylianou, A., & Pless, R. (2020). Improved embeddings with easy positive triplet mining. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2474-2482).

[23] Wang, X., Han, X., Huang, W., Dong, D., & Scott, M. R. (2019). Multi-similarity loss with general pair weighting for deep metric learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5022-5030).

[24] Wu, C. Y., Manmatha, R., Smola, A. J., & Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In Proceedings of the IEEE international conference on computer vision (pp. 2840-2848).

[25] A. Torii, J. Sivic, M. Okutomi and T. Pajdla, Visual Place Recognition with Repetitive Structures, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 11, pp. 2346-2359, 1 Nov. 2015.

[26] Wang, J., Zhou, F., Wen, S., Liu, X., & Lin, Y. (2017). Deep metric learning with angular loss. In Proceedings of the IEEE international conference on computer vision (pp. 2593-2601).

[27] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

[28] Musgrave, K., Belongie, S., & Lim, S. N. (2020). Pytorch metric learning. arXiv preprint arXiv:2008.09164.

[29] A.-D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid. Scalable place recognition under appearance change for autonomous driving. In IEEE International Conference on Computer Vision, pages 9319–9328, October 2019.

[30] R. Cheng, K. Wang, J. Bai, and Z. Xu. Unifying visual localization and scene recognition for people with visual impairment. IEEE Access, 8:64284–64296, 2020.

[31] Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, S. Papadopoulos, and Y. Kompatsiaris. Leveraging efficientnet and contrastive learning for accurate global-scale location estimation. ACM International Conference on Multimedia Retrieval, 2021.

[32] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII, volume 11216 of Lecture Notes in Computer Science, pages 575–592. Springer, 2018.