# A Study on Image Retrieval Techniques for Place Recognition

Ehsan Dashti
Politecnico di Torino
s316511@studenti.polito.it

Shiva Pourfeilieh
Politecnico di Torino
s329207@studenti.polito.it

Alimohammad Rahmatpour
Politecnico di Torino
s326889@studenti.polito.it

## Abstract

*Visual Place Recognition (VPR) is a computer vision task where a system identifies a place based on visual cues, typically from images. It involves matching a query image to a database of images to determine the location. We trained the model on a small version of the GSV-cities dataset. also we used San Francisco eXtra Small (SF-XS) for testing and validation. in addition we used Tokyo eXtra Small (Tokyo-XS) dataset only for testing. We overcome some challenging factors such as the perspective variation,lightening issues and etc in our dataset. We tried to observe how modifying some metric such as loss function and optimizer could affect the recall@ metric.Dataset, code and trained models are available for research purposes at this link.*

## 1. Introduction

Visual Place Recognition (VPR) is essential in computer vision for identifying locations using visual cues, crucial for tasks like geolocalization and robotics. Despite advancements, VPR faces challenges such as perspective changes, lighting variations, and occlusions.This study aims to improve VPR performance through exploration of different network architectures, loss functions, and optimization strategies. We trained our model on a subset of the GSV-Cities dataset and validated it using the San Francisco eXtra Small (SF-XS) dataset, with additional testing on the Tokyo eXtra Small (Tokyo-XS) dataset. The enhancements include integrating the GeM layer and using a truncated ResNet-18 to optimize performance. We also experimented with different loss functions, including contrastive, triplet, and multi-similarity loss, to determine their impact on VPR tasks. Additionally, we compared the effectiveness of various optimizers (Adam, AdamW, and SGD) for VPR tasks. This research contributes to optimizing VPR systems, aiming to develop more accurate models for geolocalization applications.

## 2. Related Works

### 2.1. Visual Geolocalization

Visual geolocalization has garnered significant attention in recent years, driven by advancements in deep learning and computer vision techniques. Arandjelovic et al. introduced NetVLAD, a CNN architecture tailored for weakly supervised place recognition, demonstrating robust performance in challenging environments [9]. Radenovic et al. extended this approach by fine-tuning CNNs for image retrieval without human annotations, highlighting the efficacy of unsupervised methods in real-world scenarios [9]. Masone and Caputo conducted a comprehensive survey in 2021, providing insights into the evolution of deep visual place recognition methods and their practical applications [8].

Liu et al. proposed Stochastic Attraction-Repulsion Embedding, a method focusing on large-scale image localization, enhancing the precision and scalability of geolocalization systems [7]. Kim et al. introduced Learned Contextual Feature Reweighting, which optimizes feature relevance for improved image geolocation accuracy, demonstrating significant performance gains [6]. Recent advancements also include the integration of non-local neural networks by Wang et al., offering enhanced contextual understanding and improving localization robustness [10].

Furthermore, datasets such as Mapillary Street-Level Sequences [5] and benchmarks like Deep Visual Geolocalization [4] and Rethinking Visual Geo-localization [3] have facilitated benchmarking and validation of these methods on large-scale datasets, fostering continuous improvement in visual geolocalization research. Ali-bey et al. contributed with frameworks like GSV-Cities [1] and MixVPR [2], focusing on supervised and feature-mixing approaches to address specific challenges in urban and complex environments.

These works collectively underscore the rapid evolution and diverse approaches in visual geolocalization, reflecting ongoing efforts to enhance accuracy, scalability, and applicability in real-world settings.

### 2.2. Image Retrieval

Image retrieval focuses on finding similar images in big collections. Here's an overview of the latest developments: [10] introduced Non-local Neural Networks, which help understand how different parts of images relate to each other. This method improves image retrieval by looking at the bigger picture. [6] combined local details with global context to make image searches more accurate. Their approach uses both close-up features and overall scene information to find images better. [11] developed DOLG, a way to find images by blending local and global features. This method works well by combining detailed parts of images with broader meanings. [4] created the Deep Visual Geo-localization Benchmark, which not only tests place recognition but also helps improve image retrieval. This dataset has images tagged with location info, making it useful for testing retrieval systems in different situations. Researchers [10] used supervised learning to make image retrieval better in cities. Their methods use more information about each place to find images more accurately. In short, image retrieval benefits from new technologies and big datasets.

## 3. Dataset

Deep neural networks need a lot of data, and we use four different datasets in our research. Each dataset has a full version and a very small version. We use the very small versions for our work. For training, we use the GSV eXtra Small (GSV-XS) dataset. For validation, we use the San Francisco eXtra Small validation (SF-XS val) dataset. For testing, we use either the San Francisco eXtra Small test (SF-XS test) dataset or the Tokyo eXtra Small (Tokyo-XS) dataset. Our model makes several predictions for each query image, and these predictions are ranked by how likely they are to match the database images.Figure 1

### 3.1. GSV

The dataset includes 560,000 images, each with precise geographical coordinates. These images come from 67,000 different locations. Each location has a set of 4 to 20 images taken from various angles and situations. The images are from 23 cities and cover the years 2007 to 2021. They were collected using the Google Street View Time Machine service [3].

### 3.2. SF

The dataset was created using Google Street View images taken from 2009 to 2021. It includes a wide range of pictures from different places, showing various perspectives and angles. Initially, around three million panoramic images were collected. These images were then divided into 12 separate parts, resulting in an average of about 41 images for each location. The dataset features images of different types of locations, such as areas with lots of vegetation, residential neighborhoods, and the outskirts of cities. Importantly, all the images were taken during the daytime, ensuring consistent lighting conditions.

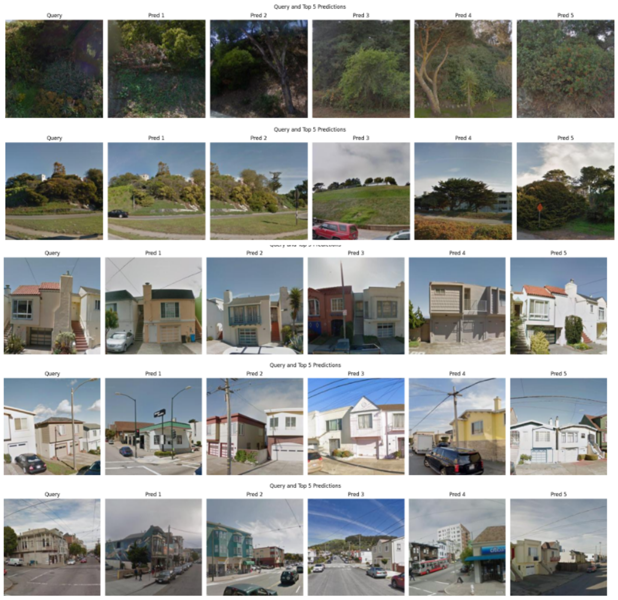

Figure 1. Each test image receives multiple predictions from our model, ranked by their likelihood to match.

### 3.3. Tokyo

Unlike the other two datasets, this one was created using a mobile camera. The Tokyo dataset, used as a test set, includes 1125 query images taken from 125 different locations in Tokyo, Japan. What makes this dataset unique is that each location is shown from three different viewpoints and at various times of the day, offering multiple perspectives.

## 4. Methodology

### 4.1. Base model

In this project, we follow a series of carefully planned steps to achieve the desired results. First, we base our approach on the methodologies and findings presented in the papers GSVCities [1] and MixVPR [2], starting under similar conditions but with a notable difference in the network architecture. We replace the original network with ResNet-18, which provides a lighter and potentially more efficient alternative. Its effectiveness in addressing the vanishing gradient problem lies in the introduction of residual connections, enabling the network to learn residual mappings. These connections facilitate the propagation of gradients throughout the network, simplifying the training of deep models. ResNet-18 comprises 18 layers and achieves

|  |  | SF-XS |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Loss function |  | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | test |
| contrastive | R@1 | 36.7 | 46.6 | 49.9 | 52.1 | 53.6 | 54.3 | 55 | 55.5 | 56.2 | 56.6 | 21.3 |
|  | R@5 | 51.3 | 62.0 | 64.6 | 66.6 | 67.7 | 68.5 | 69.3 | 69.5 | 69.9 | 70.4 | 35.4 |
| triplet | R@1 | 28.7 | 33 | 35.6 | 37.1 | 38.9 | 40 | 40.9 | 41.3 | 42 | 43.1 | 10.1 |
|  | R@5 | 43.4 | 48.0 | 50.6 | 52.3 | 54.9 | 55.3 | 56.7 | 57.2 | 58.3 | 59.1 | 19.4 |
| multisimilarity | R@1 | 38.6 | 47.6 | 51.7 | 54.3 | 55.9 | 57.6 | 58.6 | 59.6 | 60.2 | 60.8 | 25.1 |
|  | R@5 | 54.3 | 63.3 | 67.0 | 69.2 | 70.9 | 72.2 | 72.7 | 73.3 | 74.2 | 74.5 | 40.8 |

Table 1. Recall@1/5 results for different loss functions

impressive performance while maintaining a lightweight architecture compared to deeper variants. Additionally, we truncate the network at the con3 layer, reducing the depth of the network. This modification decreases the model's computational load and training time, making it more suitable for environments with limited resources or when quicker inference is required. Subsequently, we apply an average pooling layer to obtain the final embedding required for our analysis. These modifications are crucial to tailoring the project to meet our specific needs and goals. Regarding the base configuration, we employ the Adam optimizer with a learning rate of 1e-5. The loss function used is ContrastiveLoss, introduced in PyTorch metric learning, with the parameters pos-margin set to 0 and neg-margin set to 1. We begin by implementing the project under baseline conditions, incorporating the GeM Layer into the network. Additionally, we aim to compare and analyze the variations resulting from different loss functions and optimizers. To achieve this, we adjust the configuration accordingly and examine the outcomes (see Table 1).

## 4.2. GeM layer

As we aim to improve the robustness and performance of the results we achieve, we employ the Adam optimizer and incorporate the GeM layer in the second steps of our project. The reason behind this choice is that the GeM layer provides a highly flexible pooling mechanism that strikes a balance between average pooling by adjusting its parameters accordingly. This flexibility is crucial as it enables the network to capture more detailed and distinctive features in images, which plays an essential role in accurately recognizing places under varying conditions. When we compare the results between using the same Adam optimizer but switching from average pooling to the GeM layer, we observe a significant improvement in performance in the final results. This enhancement can be attributed to the GeM layer's ability to fine-tune the pooling process, thereby allowing the network to discern finer details that would otherwise be overlooked. Figure 2 illustrates the comparative results, highlighting the improvements achieved by adopting this new strategy over the prior approach.

## 4.3. Loss Function

In this phase of our project, we integrated the GEM (Geometric Margin) layer into our model and continued using the Adam optimizer, while focusing on exploring different loss functions to optimize Visual Place Recognition (VPR). Our goal was to enhance the model's ability to accurately differentiate between places. We evaluated the effectiveness of three specific loss functions: contrastive loss, triplet loss, and multi-similarity loss.

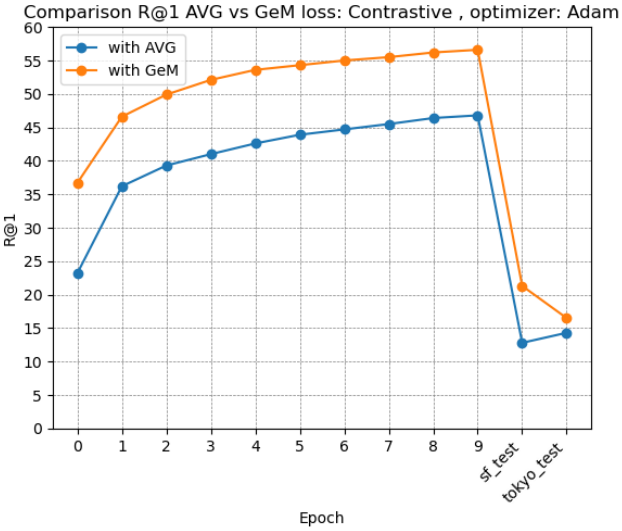

Figure 2. Performance Comparison with GeM Layer vs. Average Pooling using Adam optimizer. The GeM layer shows notable improvement by fine-tuning pooling to capture finer details, as illustrated by the results

### 4.3.1 Contrastive Loss

We began with the contrastive loss function using pos-margin=0 and neg-margin=1, which minimizes the distance between similar pairs and maximizes the distance between dissimilar pairs. This function is effective for tasks requiring the distinction between different classes. By training our model to minimize the distance between similar pairs and maximize the distance between dissimilar pairs, we aimed to enhance its capability to differentiate between places. The performance metrics, including recall@1 and recall@5, indicated high accuracy and robust differentiation between places.

### 4.3.2 Triplet Loss

Building upon our successful implementation with the contrastive loss, we made a strategic adjustment by transitioning to triplet loss with a margin of 1.0, and subsequently incorporating the GEM layer. Unlike contrastive loss which pairs an anchor with a positive or negative sample, triplet loss involves training with an anchor, a positive sample, and a negative sample simultaneously. This method aims to enforce closer proximity of the anchor to the positive sample

|  |  | SF-XS |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Loss function |  | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | test |
| contrastive | R@1 | 36.7 | 46.6 | 49.9 | 52.1 | 53.6 | 54.3 | 55 | 55.5 | 56.2 | 56.6 | 21.3 |
|  | R@5 | 51.3 | 62.0 | 64.6 | 66.6 | 67.7 | 68.5 | 69.3 | 69.5 | 69.9 | 70.4 | 35.4 |
| triplet | R@1 | 28.7 | 33 | 35.6 | 37.1 | 38.9 | 40 | 40.9 | 41.3 | 42 | 43.1 | 10.1 |
|  | R@5 | 43.4 | 48.0 | 50.6 | 52.3 | 54.9 | 55.3 | 56.7 | 57.2 | 58.3 | 59.1 | 19.4 |
| multisimilarity | R@1 | 38.6 | 47.6 | 51.7 | 54.3 | 55.9 | 57.6 | 58.6 | 59.6 | 60.2 | 60.8 | 25.1 |
|  | R@5 | 54.3 | 63.3 | 67.0 | 69.2 | 70.9 | 72.2 | 72.7 | 73.3 | 74.2 | 74.5 | 40.8 |

Table 2. Recall@1/5 results for different loss functions

than to the negative sample, thus refining the model's ability to discern similarities and differences within the dataset. Despite these modifications, our evaluation metrics, such as recall@1 and recall@5, indicated a decrease in performance compared to the contrastive loss.

### 4.3.3 Multi-Similarity Loss

After observing the limitations of the previous loss functions, we employed the multi-similarity loss function. Multi-similarity loss evaluates multiple similarity measures simultaneously, offering a nuanced approach to distinguishing between similar and dissimilar pairs. It aims to improve feature learning and overall model performance. This method allowed the model to simultaneously consider various similarity measures, providing a more comprehensive training approach. The performance metrics, including recall@1 and recall@5, demonstrated significant improvement over both contrastive and triplet loss functions. (see Table 2)

Our experiments highlighted the critical role of selecting the appropriate loss function for VPR tasks. The contrastive loss function provided a solid baseline with high accuracy, while the triplet loss did not meet our expectations for this specific project. Ultimately, the multi-similarity loss function delivered the best performance, as evidenced by higher recall rates at both recall@1 and recall@5 ranks, underscoring its suitability for enhancing feature learning and place recognition accuracy in VPR applications. The comparative graph 4 illustrating the performance of contrastive, triplet, and multi-similarity loss functions further emphasizes the superiority of multi-similarity loss in our VPR project. This visual representation supports our findings and reinforces the importance of methodically evaluating different loss functions to optimize model performance.

## 4.4. Optimizer

In the context of optimizing neural networks, the choice of optimizer is crucial for adjusting model parameters to minimize the loss function, thereby impacting efficiency, convergence speed, and overall performance. Initially, we selected the Adam optimizer with a learning rate of 1e-5 due to its adaptive learning rate and ability to handle sparse gradients, making it a popular choice for many neural network applications. However, we observed limitations in the model's ability to accurately recall relevant images, prompting us to explore alternative optimizers.

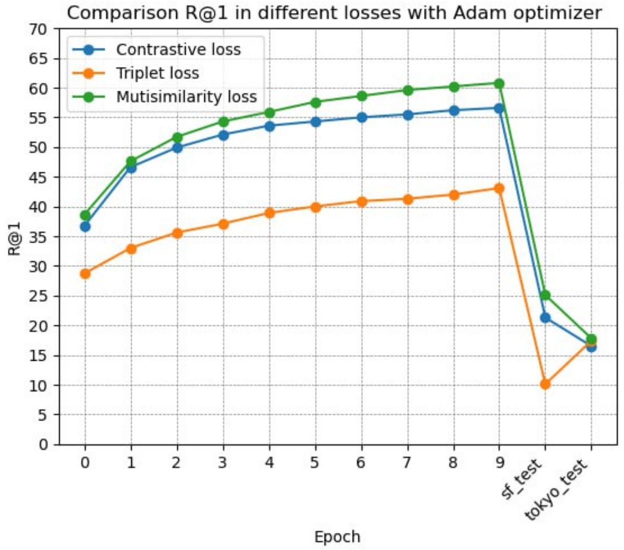

Figure 3. The results of our evaluations highlight the impact of integrating the multi-similarity loss function in contrast to using contrastive and triplet loss functions.

### 4.4.1 AdamW

We then switched from the Adam optimizer to AdamW with a learning rate of 1e-5 and weight decay of 0.01. The choice of AdamW was motivated by its decoupled weight decay regularization, which helps in preventing overfitting and can lead to better generalization. Our experiments demonstrated that AdamW surpassed Adam, effectively improving the model's recall accuracy. This improvement indicates that AdamW's approach to weight decay, which is more effective at controlling the complexity of the model, was beneficial for our image recognition task.

### 4.4.2 SGD

To further explore optimization strategies, we adopted the SGD optimizer with a learning rate of 0.01, weight decay of 0.0005, and momentum of 0.9. The selection of SGD was driven by its simplicity and effectiveness in training large-scale machine learning models, particularly for tasks that require fine-tuning of the model parameters. Momentum helps in accelerating gradient vectors in the right direction, thus leading to faster convergence.

Figure 4 illustrates significant improvements in our results, indicating that SGD excels in image recognition compared to both Adam and AdamW. This shift underscores the importance of empirical validation in determining the op-

|  |  | SF-XS |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| optimizer |  | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | test |
| Adam | R@1 | 38.6 | 47.6 | 51.7 | 54.3 | 55.9 | 57.6 | 58.6 | 59.6 | 60.2 | 60.8 | 25.1 |
|  | R@5 | 54.3 | 63.3 | 67.0 | 69.2 | 70.9 | 72.2 | 72.7 | 73.3 | 74.2 | 74.5 | 40.8 |
| AdamW | R@1 | 40.6 | 49.2 | 53.1 | 55.4 | 56.8 | 58.0 | 59.2 | 60.0 | 60.6 | 61.4 | 24.23 |
|  | R@5 | 56.4 | 64.9 | 67.8 | 70.2 | 71.3 | 72.1 | 72.9 | 73.4 | 73.9 | 74.4 | 39.3 |
| SGD | R@1 | 53 | 59.6 | 61.6 | 63 | 62.8 | 64.2 | 64.4 | 65.1 | 65.9 | 66.2 | 28.79 |
|  | R@5 | 68.2 | 72.7 | 73.7 | 75.5 | 75.7 | 76.8 | 77.1 | 77.6 | 78.0 | 78.6 | 43.8 |

Table 3. Recall@1/5 comparison for different optimizer

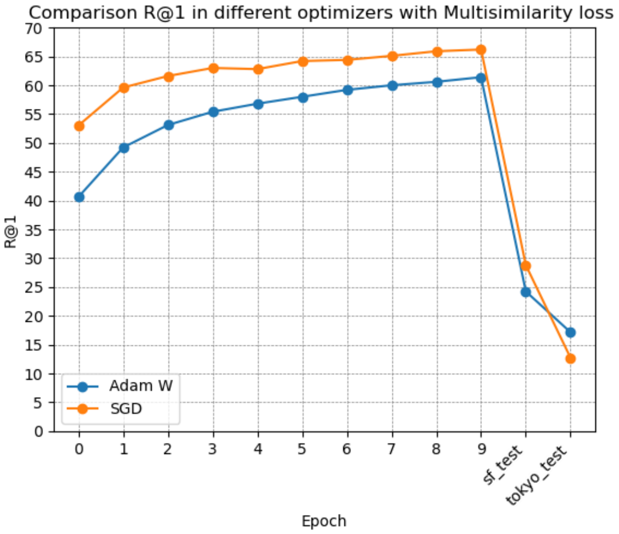

Figure 4. Impact of optimizer Selection on Recall Metrics in Visual Place Recognition Tasks. Comparing SGD, Adam, and AdamW reveals significant performance improvements with SGD, highlighting its effectiveness in image recognition.

timal configuration for achieving superior performance in Visual Place Recognition tasks.3

## 5. Conclusion

In this project, we systematically explored various methodologies to enhance Visual Place Recognition (VPR) tasks, focusing on optimizing network architecture, loss functions, and optimizers. We based our approach on the methodologies presented in GSVCities [1] and MixVPR [2], integrating findings from the Deep Visual Geo-localization Benchmark [4] and Rethinking Visual Geo-localization for Large-Scale Applications [3] to build upon existing knowledge.

Starting with ResNet-18, we utilized its lightweight design and residual connections to mitigate the vanishing gradient problem, enhancing training efficiency while maintaining robust performance. The incorporation of the GeM layer significantly improved our model's ability to capture intricate image details that average pooling might overlook, as supported by our results (Figure 2). Our exploration of loss functions revealed nuanced performance differences: while triplet loss initially showed promise, contrastive loss ultimately proved more effective for our specific VPR application. The adoption of the multi-similarity loss function further enhanced feature learning, significantly improving place recognition accuracy under diverse conditions, as evidenced in Figure 3. Optimization strategies were crucial, with AdamW and SGD optimizers demonstrating notable advantages over traditional Adam in improving recall metrics. Notably, SGD emerged as particularly effective for image recognition tasks, aligning with findings from Berton et al. [4] and emphasizing the importance of aligning optimizers with task-specific requirements. Figure 4 illustrates these improvements, highlighting SGD's superiority in image recognition compared to Adam and AdamW. Throughout our experimentation, empirical validation consistently guided our decisions, echoing the iterative nature of neural network optimization. Our findings underscore the critical role of selecting optimal configurations tailored to specific VPR tasks, ensuring superior performance and robustness in real-world applications. This study advances the field of Visual Place Recognition by providing insights into effective methodologies and configurations, paving the way for future enhancements in image recognition and related domains, as detailed in the contributions of Ali-bey et al. [2] and the foundational work in geo-localization benchmarks [4].

## References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguere. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 1, 2, 5

[2] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe Giguere. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2998–3007, 2023. 1, 2, 5

[3] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022. 1, 2, 5

[4] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep visual geo-localization benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5396–5407, 2022. 1, 2, 5

[5] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 726–743. Springer, 2020. 1

[6] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *CVPR*, volume 1, page 3, 2017. 1, 2

[7] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic attraction-repulsion embedding for large scale image local-

ization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2570–2579, 2019. 1

[8] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021. 1

[9] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018. 1

[10] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1, 2

[11] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE/CVF International conference on Computer Vision*, pages 11772–11781, 2021. 2