Optimizing Neural Networks for Urban Visual Geo-Localization: A Comparative Approach

Matteo Bracco,Francesca Geusa, Christian D'Alleva
s319845@studenti.polito.it, s329174@studenti.polito.it, s331883@studenti.polito.it

Abstract

Visual Geo-localization (VG) involves determining the geographical location where a specific photograph was taken by matching it against a vast collection of images with known coordinates. The aim of this project is to explore the use of neural networks to enhance localization accuracy in complex urban environments. The proposed approach is based on the implementation of the paper "GSV-Cities: Toward appropriate supervised visual place recognition" [3], adapted to a reduced version of the GSV-Cities dataset in the training phase, and to a reduced version of Tokyo [10] and San Francisco [2] datasets in the testing phase. The networks showed a remarkable ability to distinguish between similar urban areas. The proposed approach includes the comparison between two different architectures, ConvAP and MixVPR [4], considering both their recall performances and qualitative results.

Code available at https://github.com/fracrumatte/geoloc-fcm-gsv-cities.

1. Introduction

Visual geo-localization (VG), also called Visual Place Recognition (VPR), refers to the process of identifying a place depicted in a query image using only computer vision. In order to determine whether the location depicted in a query image has already been visited, the system uses a database of images of previously-visited locations, and compares the query against them.

Several datasets have been released to train and evaluate different techniques of place recognition. The dataset used for the training phase is **GSV (Google Street View) - CITIES**[GSV-Cities] [3], a large-scale dataset which includes highly-accurate GPS coordinates and viewing direction for each image of 23 cities all around the world, covering a time-span of 14 years. All locations are geographically distant and uniformly distributed in every city, with at least four images for each location (o place). Training with GSV-CITIES improves performance of existing VPR techniques.

The dataset used for the validation phase is **San Francisco** [2], composed of a large database collected by a car-mounted camera; although being one of the largest with a database of 1 million images, still covers only 9% of the city of San Francisco.

For the testing phase two different datasets are used, San Francisco and Tokyo, in order to understand how changes to the model affect the geo-localization for different locations.

Tokyo [10] dataset has a quite large database (from Google Street View) and a smaller number of queries; the query images of Tokyo dataset capture the same locations in three different illuminations (day, sunset, night) and contain structural changes in the scene.

The model implemented in this project explores several techniques in the training phase in order to find which performs best.

The backbone architecture used is **ResNet-18** [5], a deep convolutional neural network (CNN) known for its high performance on image recognition tasks. The configuration includes an initial convolutional layer, followed by multiple residual blocks, and concludes with a final average pooling layer followed by a fully connected layer that produces the final output. The aim of this project is to study the behaviour of the AdamW [8] optimizer in the presence of different aggregators such as AVG [1], GeM [9] and ConvAP[cite] and then comparing the best among them with the MixVPR [4] aggregator architecture.

Paper outline In Chapter 2, we introduce the tools used in this project, analyzing their structural characteristics. Chapter 3 explains how the analysis was performed, including the aggregators and optimizers used to build the models, and their evaluation. Chapter 4 discusses a qualitative analysis of the results given by the top two models tested, considering their behavior in different scenarios. Chapter 5 presents the conclusions and potential future improvements.

2. Methodologies

In this chapter, we delve into the core components and methodologies crucial to our research on visual place recognition. We begin by exploring the Average Pooling (**Avg**) [1], it works by dividing the input into rectangular regions and computing the average value of the values within each region. This operation reduces the dimensionality of the input and helps to retain important information while discarding less important details.

$$\text{AvgPooling}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where X is the set of values in the pooling window and n is the number of values.

Then, different aggregation models are used in order to understand if there are better alternatives to the average pooling layer, for example GeM, ConvAP and MixVPR:

• Generalized Mean Pooling (GeM) [9] is a flexible pooling method that generalizes max pooling and average pooling.

$$\text{GeM}(X) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}}$$

, where X is the set of values in the pooling window, n is the number of values, and p is the parameter controlling the pooling type.

• ConvAP is a new efficient aggregation technique that performs channel-wise pooling on the feature maps followed by spatial-wise adaptive pooling. Given an image I_i , this technique passes it through the pre-trained backbone to obtain its feature maps F_i , and pass them to the aggregation layer to obtain the final representation $\mathbf{z}_i$:

$$\mathbf{z}_i = \text{AAP}_{s_1 \times s_2}\left(\text{Conv}_{1\times 1}\left(\mathbf{F}_i\right)\right)$$

•As detailed in the original work MixVPR [4]: Feature Mixing for Visual Place Recognition is a holistic feature aggregation technique that processes flattened feature maps from intermediate layers of pre-trained backbones to create global features. It employs successive Feature-Mixer blocks, composed solely of multi-layer perceptrons (MLPs), to integrate global relationships into individual feature maps, eliminating the need for local or pyramidal aggregation. The final output is projected into a compact representation space and used as a global descriptor.

Later, different training hyperparameters are exploited to decide what configurations can significantly influence the performance and convergence of the model. During this phase, common optimizers are used to adjust the weights of the neural network to minimize the loss function:

- ADAM (Adaptive Moment Estimation) [6]is one of the most popular optimization algorithms in deep learning due to its adaptive learning rate and momentum properties.
- ADAMW (Adam with Weight Decay) [8] is a variant of Adam that decouples the weight decay regularization from the gradient update. In traditional Adam, weight decay is incorporated in the update step, which can lead to suboptimal regularization. AdamW [8] addresses this issue by applying weight decay directly to the weights.
- SGD (Stochastic Gradient Descent) is an iterative method for minimizing an objective function, typically a loss function, by updating the model's parameters (weights) in the direction that reduces the loss. The term "stochastic" refers to the randomness in the gradient estimation process: each update is based on a randomly selected subset of data.

In the configuration of the optimizer, also different scheluders are explored. The scheduler is an algorithm used to adjust the learning rate during the training process. The learning rate controls how much the model's parameters are updated with respect to the gradient of the loss function. Properly adjusting the learning rate can significantly impact the model's convergence and overall performance. The schedulers used in this project are the following ones:

- CosineAnnealingLR adjust the learning rate of an optimizer according to a cosine annealing schedule. The idea is to periodically decrease the learning rate following a cosine curve, allowing the model to explore different regions of the loss landscape.
- ReduceLrOnPlateau reduces the learning rate when a monitored metric has stopped improving. This type of learning rate scheduling helps to adapt the learning rate dynamically based on the model's performance, allowing for finer convergence and potentially better generalization.
- MultistepLR adjusts the learning rate of an optimizer by reducing it at specified epochs. At each epoch specified in the milestones, the learning rate is reduced by multiplying it with the factor gamma. This helps to refine the training process by reducing the learning rate gradually, which can help in stabilizing the training and achieving better convergence.

We examined the MultisimilarityLoss function among the pool of loss functions. It is an approach in deep learning for enhancing feature discrimination and similarity learning. It ensures that similar examples are closer together in the embedding space while dissimilar examples are farther apart. This loss function incorporates advanced pair weighting schemes.

Method	SF - val			SF - test			Tokyo - test		
	R@1	R@10	R@25	R@1	R@10	R@25	R@1	R@10	R@25
AVG	47.22	68.75	76.33	12.80	31.80	39.70	22.54	46.35	52.70
GEM	55.77	77.43	84.46	20.70	41.10	49.80	35.56	62.86	72.70
ConvAP	71.71	83.39	87.25	41.20	61.50	67.30	55.87	79.68	84.44
MixVPR	78.83	89.77	93.06	53.10	70.50	75.70	70.16	84.13	91.11

Figure 1. Comparison between aggregators

Aggregator = ConvAP Optimizer = AdamW		Learning rate					
		0.01			0.001		
		R@1	R@10	R@25	R@1	R@10	R@25
Weight decay	0.01	71.65	82.83	86.98	71.71	83.39	87.25
	0.001	70.99	82.75	86.84	71.37	82.51	86.54

Figure 2. Hyperparameter tuning

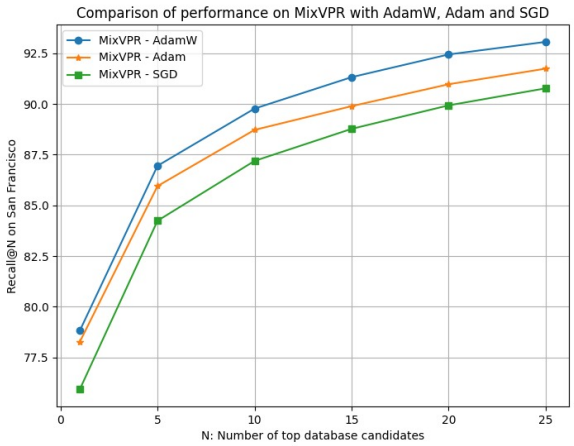

Figure 3. MixVPR performances with different optimizers

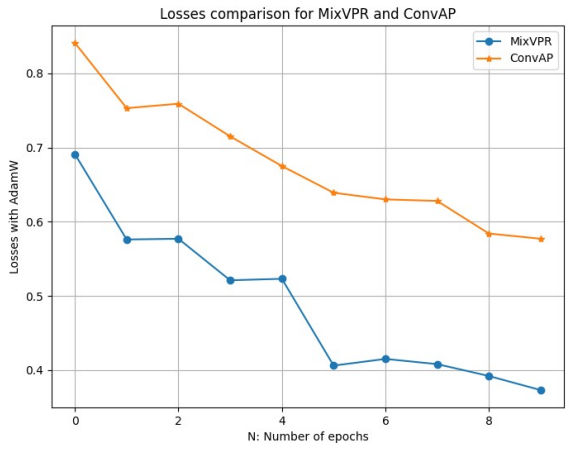

Figure 4. Losses on 10 epochs of MixVPR and ConvAP

3. Experiments

In this section, we conduct experiments to establish the best configuration for visual place recognition. Our approach focuses on the optimizer AdamW [8], linked to various aggregators. Although we considered other optimizers like SGD and Adam [6], AdamW [8] showed remarkable results and has proven to be on par with SGD. Our analysis follows two main branches:

1. The first aims to construct the best architecture that maximizes recall. To achieve this, we compared different aggregators such as AVG [1], GeM [9], and ConvAP.

2. The second branch involves comparing the performance of the SGD and AdamW [8] optimizers applied to MixVPR [4]aggregator.

The two branches share a pre-trained ResNet [5] backbone, trained on ImageNet [7]. For all architectures, the CNN backbone is cropped at the last convolutional layer, specifically at the fourth layer of ResNet18 [5]. All training has been performed on the dataset composed of 60K places and 500K images introduced by GSV-Cities [3]. For all training, we used a batch size of 32 with a minimum of 4 images per place. Due to hard-ware limitations, we ran the training for only 10 epochs.

Implementation of the first branch First, we analyzed the AVG [1], GeM [9], and ConvAP aggregators. We compared them based on the recall percentages of the first k predictions. From this comparison, it emerged that the ConvAP aggregator outperforms the others, providing better recall values for every k value. The results of this comparison are shown in the Table Fig. 1. Next, we performed hyperparameter tuning on the ConvAP architecture, taking into account parameters such as learning rate and weight decay, as shown in Table Fig. 2. The best results were obtained with a learning rate of 0.001 and a weight decay of 0.01. We chose these parameters by applying reasonable variations to the standard values. Additionally, we used a fixed momentum of 0.9.

Our approach relied on MultiStepLR as the scheduler. In this context, we also analyzed two alternatives: ReduceLrOnPlateau and CosineAnnealingLR. The results obtained from the alternatives were not satisfactory, so we decided to retain the initial scheduler.

Implementation of the second branch Our second analysis focused on a recently proposed aggregator called MixVPR [4], which has shown remarkable results in visual recognition tasks. We trained this aggregator using both SGD, as proposed by the authors, and the best version of

AdamW [8], derived from the hyperparameter tuing, to ensure a fair comparison. The results obtained were remarkable and are summarized in the table Fig. 3. The losses on ConvAP and MixVPR [4], with the configuration of the best model of the grid for ConvAP and AdamW with learning rate of 0.001 and weight decay of 0.0003, results are in Table Fig. 4.

3.1. Evaluation

The evaluation was carried out on three benchmarks: San Francisco Val xs, San Francisco Test xs [2], and Tokyo xs [10]. A query image is considered successfully retrieved if at least one of the top-k retrieved reference images is located within 25 meters of the query image.

4. Qualitative Results

Qualitative analysis of predictions involves evaluating the model's performance based on appearance information. This type of analysis focuses on understanding how and why the network makes certain decisions by examining its outputs in specific scenarios. This approach can be used to compare two or more models, highlighting their strengths and weaknesses in particular scenarios, and offering insights into their practical effectiveness and areas for improvement. We analyzed the following scenarios:

- **Repetitive Structures**: Observing if the model can correctly identify distinct places with similar architectural features.
- **Viewpoint Changes**: Evaluating the model's ability to recognize places from different angles or perspectives.
- **Illumination Changes**: Testing the model's performance when there are significant lighting variations between the query image and the reference images.
- **Occlusions**: Assessing how the model handles images where key features are partially blocked by objects or people.
- **Skyline**: Checking if the model can rely on less prominent features like skylines when other distinctive features are absent.
- **Failed cases**: Observing in which cases the model was not able to perform a correct prediction.

We compared the predictions of the architectures based on ConvAP and MixVPR [4]. The results showed that the two models often behave similarly, but in critical situations, ConvAP rarely outperforms MixVPR [4]. Even though both models demonstrated remarkable results in terms of recall, they tend to suffer in scenarios with occlusions and limited information, such as skylines.

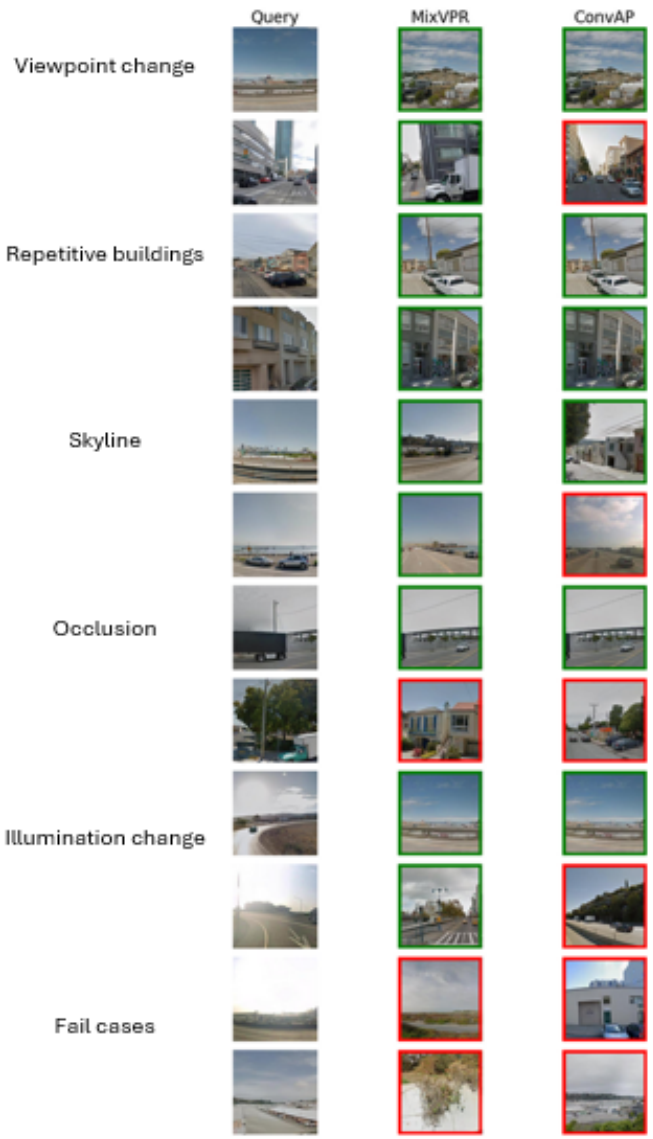

Figure 5. Qualitative analysis

5. Conclusion

In this project, we implemented several architectures based on ConvAP and MixVPR [4]. Through our experimentation, we showcased the effectiveness of pairing the AdamW [8] optimizer with state-of-the-art aggregators like MixVPR [4], achieving remarkable results. Our findings underscore the robust performance and synergy between these architectures and optimization strategies, highlighting their potential in advancing the field of visual recognition.

Future development The next steps could involve several approaches. Firstly, conducting a thorough hyperparameter tuning that includes a broader range of values for learning

rate, weight decay, and batch size. Additionally, exploring different loss functions could provide insights into their respective outcomes.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition, 2016. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1, 2, 3

[2] Berton, G., Mereu, R., Trivigno, G., Masone, C., Csurka, G., Sattler, T., and Caputo. Deep visual geo-localization benchmark, 2022. CVPR. 1, 4

[3] Brahim Chaib-draa, Philippe Giguère, Ali-bey, and Amar. Gsv-cities: Toward appropriate supervised visual place recognition., 2022. Neurocomputing 513 (2022): 194-203. 1, 3

[4] Brahim Chaib-draa, Philippe Giguère, Ali-bey, and Amar. Mixvpr: Feature mixing for visual place recognition., 2023. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 1, 2, 3, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2016. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1, 3

[6] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. International Conference on Learning Representations. 2, 3

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks, 2012. Advances in neural information processing systems. 3

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. International Conference on Learning Representations. 1, 2, 3, 4

[9] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Fine-tuning cnn image retrieval with no human annotation, 2018. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1, 2, 3

[10] A. Torii, R. Arandjelovi éc, J. Sivic, M. Okutomi, and T.Pajdla. 24/7 place recognition by view synthesis., 2018. IEEETransactions on Pattern Analysis and Machine Intelligence. 1, 4