Are You Sure You Are in the Right Place?

Davide Sferrazza

1 Introduction to Visual Place Recognition

Visual Place Recognition (VPR) [3, 6, 16, 19] is the task of recognizing the location where an image was taken given only its visual content, *i.e.* the image itself. It is a building block of fundamental importance for applications in autonomous driving, augmented reality, robotics, absolute pose estimation, simultaneous localization and mapping, etc. The problem is commonly treated as an image retrieval task, where a summary description of an image of interest, called a **query**, is computed and compared against a **database** of descriptions of images with known locations, typically expressed in geographical coordinates.

To determine the location of the query image, a summary description is computed for both the query and the database images. A **similarity search** is then conducted over the database using a distance metric, often implemented with a K-Nearest Neighbors algorithm. This algorithm identifies the K most similar images to the query based on their representations. Once the K nearest neighbors are retrieved, they contribute to the final location prediction. Typically, the location is predicted by selecting the most similar database descriptor, though in some cases, additional aggregation techniques may be used to refine the prediction.

A full Visual Place Recognition pipeline, incorporating the steps outlined above, is shown in Fig. 1. This pipeline includes an optional refinement step, which seeks to re-rank the nearest neighbors by utilizing local image features. The query and the retrieved images are compared by counting the number of *inliers*, *i.e.*, matched keypoints between the query and a retrieved image that survive a geometric post-processing using RANSAC [11]. This step is time-consuming and has high computational cost, though it has been used during the years to increase localization performance. The figure also highlights the processing time of the descriptors.

To evaluate and compare the performance of Visual Place Recognition models, the standard metric used is **Recall@N** on a given dataset. For a specified distance threshold τ in meters, the Recall@N measures the percentage of queries for which at least one of the top-N (with $N \leq K$) retrieved database images–determined by the K-Nearest Neighbors algorithm–is within τ meters of the ground-truth location of the query. A common value for τ is 25 meters.

2 Uncertainty Estimation

A key limitation of current State-Of-The-Art methods in VPR is that they do not provide any indication of **uncertainty** regarding their predictions. This is especially important in *safety-critical scenarios*, such as autonomous driving, where incorrect predictions could have severe consequences,

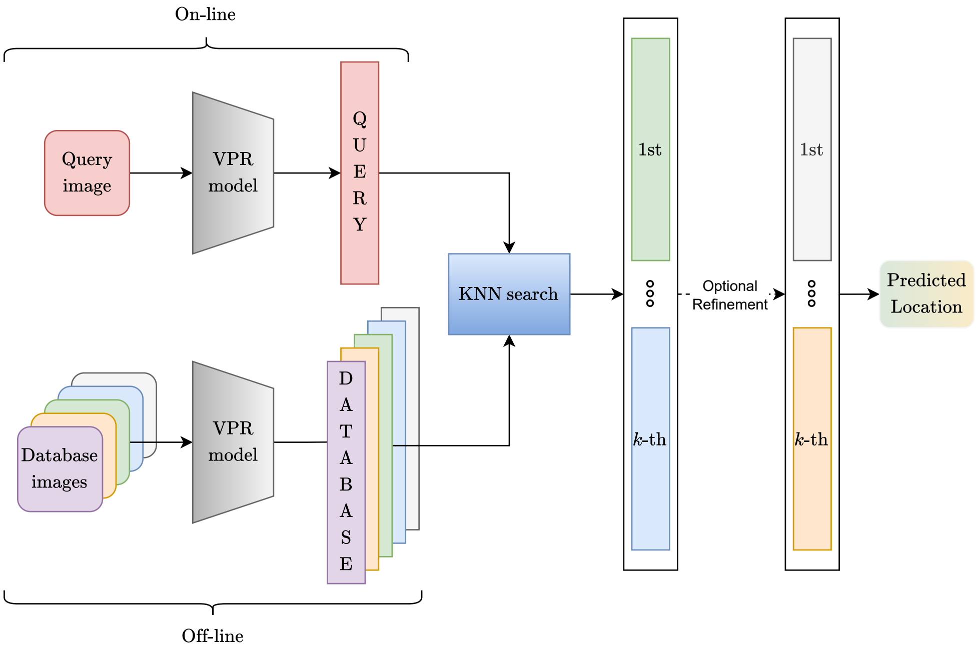

Figure 1: **Common Visual Place Recognition pipeline.**

including harm to people or environmental damage. In such contexts, providing uncertainty scores–**quantifying the model's confidence in its predictions**–is essential. These scores allow both humans and systems to identify uncertain predictions and avoid making risky decisions.

3 Goals

The goals of this project can be summarized as follows:

1. Get acquainted with the field of Visual Place Recognition, by understanding current common practices and why we need reliable VPR systems;
2. Conduct experiments using popular VPR and image matching methods to understand how an additional refinement step influences the performance-efficiency trade-off;
3. Propose your own extension to improve the system's reliability, efficiency, performance or interpretability.

4 Datasets

The datasets for this project can be found on Google Drive at the following two links:

1. GSV-XS, Tokyo-XS, SF-XS
2. SVOX

GSV-XS dataset is a small version (XS for *eXtra Small*) of the GSV-cities dataset [1], created for your convenience to adapt to Colab resources. This dataset is used for **training**.

Method	**Dataset 1**	**Dataset 2**	**...**	**Dataset N_{D}**
	R@20	R@20	...	R@20
VPR Method 1	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method 1 + Image Matching method 1	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method 1 + Image Matching method 2	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
⋮				
VPR Method 1 + Image Matching method N_{IM}	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method 2	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method 2 + Image Matching method 1	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method 2 + Image Matching method 2	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
⋮				
VPR Method 2 + Image Matching method N_{IM}	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
⋮				
VPR Method N_{VPR}	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method N_{VPR} + Image Matching method 1	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
VPR Method N_{VPR} + Image Matching method 2	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10
⋮				
VPR Method N_{VPR} + Image Matching method N_{IM}	R@1/R@5/R@10	R@1/R@5/R@10		R@1/R@5/R@10

Table 1: Example table for performance measurements using Recall@N.

San Francisco eXtra Small (SF-XS) is a subset of the SF-XL dataset [5]. It is used for **validation (SF-XS val)** and **testing (SF-XS test)**. Note that SF-XS also has a training set, but you will not use it in this project, and you can simply ignore it.

Tokyo eXtra Small (Tokyo-XS) is a subset of the Tokyo 24/7 [25] dataset. This is used only for **testing**.

SVOX [7] is a dataset which provides a robust test set for cross-domain VPR. For the project, queries from the Sun and Night subsets are used for **training** and **testing**.

5 Steps

5.1 Read and study the literature

As a preliminary step to get familiar with the task, especially the retrieval part, start by reading papers like [3, 6, 19]. [16] is a survey, which goes into extensive details; if you want, you can use it to get a global overview. During the project, you are going to use popular VPR methods like NetVLAD(4096) [3], CosPlace(512) [5], MixVPR(4096) [2] and MegaLoc [4]. Read the papers carefully; make sure you understand the essence of the task and how to approach it.

For the re-ranking part, instead, you are going to use methods like Superglue [21], LoFTR [24] and SuperPoint+LightGlue [9, 15]. Although it is not necessary to read the original papers, it is still essential to understand what Image Matching is (you can search online to get relevant resources and knowledge).

5.2 Run experiments using the provided code

Once you are familiar with the theory of Visual Place Recognition, you can start to run some experiments to understand how the whole VPR pipeline works.

First, you should evaluate the performance of the VPR methods cited in Section 5.1 on the test sets of Section 4 using only the retrieval part (pay attention to the image size for each method). In this phase, experiment with different distance metrics for the K-nearest neighbors search, with $K = 20$. Specifically, compare the L2 distance with the dot product as the distance measure and evaluate how the metric choice influences the retrieval results. Do you see any changes? Why?

After conducting these tests, select the metric that yields the best retrieval performance and use it for all subsequent experiments.

Next, you should run Image Matching methods over the retrieval predictions (fix the image size to 512×512 for input images to Image Matching methods) and analyze how performance changes (and at what price...). You should report your results in a table similar to the one displayed in Table 1 using the methods cited in Section 5.1 and the test sets in Section 4–report only the results for the chosen distance measure. In addition, you should report other metrics, such as the processing time for a single query, to better understand the trade-off involved.

Your coding, starting point can be found at the following link: FarInHeight/Visual-Place-Recognition-Project.

After running these first experiments, try to see if you can find something interesting, like a correlation between queries' correctness using only the retrieval part (consider only R@1) and the number of inliers with the first retrieved image. Try to plot some histograms of the number of inliers, differentiating between wrong and correct queries. Can you distinguish between wrong and correct queries using the number of inliers?

6 Extentions

6.1 Can Re-ranking be Adaptive? (both Retrieval and Re-ranking parts)

The VPR world needs effective but also *efficient* solutions. One contribution could be to make *adaptive* the re-ranking part. Instead of blindly applying re-ranking for each query image, try to propose a solution where the image matching method re-ranks only when the query is hard. What does *hard* mean? Well, based on the previous study, it could mean that the number of inliers between the query and the first retrieved image is low with respect to the entire inliers' distribution over the queries. Thus, one solution could be to apply re-ranking when the inliers count is below a certain threshold. How to decide the threshold? Can I avoid hard-thresholding and use instead a logistic regressor?

Here, you should propose a definition of *hard* query (you can use the definition given above) and an *adptive re-ranking* strategy. If you decide to use hard-thresholding, I expect to see plots showing how R@1 varies as a function of the threshold and how dataset choice influences the threshold computation and final performance on test sets. If you use a logistic regression, I expect the same dataset-based analysis. Additionally, you should calculate the cost savings of your strategy.

Pick two VPR methods and two Image Matching methods from your previous results. For training the logistic regressor or for threshold selection, use only the **training** sets–excluding GSV-XS. Use the **validation** sets to validate your hyperparameter selections. Then, evaluate on all the **test** sets.

6.2 How to estimate Uncertainty? (mainly Retrieval part)

As highlighted in Section 2, current VPR methods are deterministic, and as such cannot quantify the uncertainty in their predictions by design. In the literature, some attempts have been made to estimate uncertainty in a post-hoc fashion, *i.e.*, by keeping the VPR model frozen, like measuring L2 distance in feature space [18], PA-score [13], and SUE [28]. Even the number of inliers between the query and the first retrieved image can be used as an uncertainty measure [23]. Performance is assessed using AUPRC scores.

Here, you should produce results for the current solutions and propose your own. The most straightforward approach is to train a logistic regressor to predict the uncertainty from the number of inliers, where a lower number of inliers translates into higher uncertainty, and thus higher probability of wrongly localizing the query. In this case, you should evaluate how dataset choice for training influences the uncertainty results. Another possible solution instead involves using *generative models*, for example VAEs [14], which learn the data distribution and use the likelihood as an uncertainty estimation. In this setup, the model is trained on the VPR model's outputs from the GSV-XS dataset. Additionally, you should implement other metrics to evaluate the performance of uncertainty estimation, *e.g.*, the *Spearman's rank correlation coefficient* and the *coefficient of determination* in [26], and the *Area Under the Sparsification Curve* [27]. If you find other metrics in the literature, you can implement them as well.

Pick two VPR methods and two Image Matching methods from your previous results. If any training is required, use only the **training** sets. Use the **validation** sets to validate your hyperparameter selections. Then, evaluate on the **test** sets.

6.3 How to save Memory? (only Retrieval part)

Visual Place Recognition methods produce high-dimensional output vectors, which linearly contribute to the database memory usage. These vectors are composed of several features, but not all of them could be necessary to perform the task. A possible study thus could be how to reduce the output vectors' dimensionality by inspecting existing relationships among feature values and by discarding features which are highly correlated with others and features that are almost never activated.

Here, you should devise a strategy to label features as *unnecessary* or *essential*, and assess performance with or without unnecessary features. To detect useless features you could track the feature activation pattern over the GSV-XS dataset of each feature and discard features which are rarely activated (*i.e.*, non-zero values) or with nearly-uniform activation distributions. To eliminate features which carry redundant information, inspect the correlation matrix of feature activations and eliminate highly correlated features. In addition, you should visualize what parts of an image correspond to deleted features using Grad-CAM [22] or Activation Maximization techniques [10]. If discarded features have no clear visual patterns or show low discriminative power, then feature discardment is justified. Furthermore, you should measure the compression factor you achieved with your strategy.

For this study, pick two VPR methods from your previous results (excluding NetVLAD–can you guess why?) and use the **GSV-XS** dataset for **feature labeling**. Use the **validation** set to validate your hyperparameter selections like thresholds. The evaluation is conducted on the **test** sets.

6.4 How to obtain Mono-Semantic Features? (only Retrieval part)

A promising direction in *mechanistic interpretability*, *i.e.*, a field in *Deep Learning* which tries to understand the internal reasoning of *Neural Networks* (NNs) by analyzing their computations, lies in *Sparse Autoencoders* [8]. Under the *superposition hyphothesis*, where NNs encode more features than neurons they have, Sparse Autoencoders try to recover mono-semantic neurons by leveraging *overcomplete autoencoders* with *sparsity penalties* [20].

Here, you should study if VPR methods, as implemented through Neural Networks, can recover mono-semantic features, thanks to Sparse Autoencoders[1]. Recently, mono-semanticity scores have been proposed in the literature [12, 17]. You should evaluate Sparse Autoencoders trained on outputs' vectors for the GSV-XS dataset and study their mono-semanticity using one of the mono-semenaticity metrics in [17] or [12]. In addition, you should give examples of top-activating images for some of the identified mono-semantic features. Choose some neurons with different mono-semanticity scores and, if the Sparse Autoencoder is trained correctly, you should see coherent images for highly mono-semantic features, and decreased coherence with decreased mono-semanticity scores .

For this study, pick two VPR methods from your previous results and use only the **GSV-XS** dataset for both **training** and **visualization**.

6.5 Got Your Own Idea?

I encourage you to think outside the box and propose your own ideas! If you have a unique extension or modification in mind that is not covered in the previous extensions, please feel free to **discuss** it *in advance* **with me**. The goal here is to demonstrate a thoughtful approach and strong reasoning for your choice. Be sure to explain why you are pursuing this particular direction and what you hope to explore or discover through it.

7 Deliverables

To conclude the project you will need to:

- Deliver PyTorch scripts for the required steps as a zip file or by linking your GitHub repository (upload only scripts and code, no model weights or datasets);
- Write a complete PDF report following a standard paper format. The report should contain a brief introduction, a related works section, a methodological section for describing the algorithms you are going to use, an experimental section with all the results and discussions. End the report with a brief conclusion.

Report format and delivery instructions will be given on the teaching portal.

8 What do I expect from you?

Examples of questions you should be able to answer after you complete the project:

- What is VPR?

[1] https://github.com/KempnerInstitute/overcomplete

- How are VPR models trained?
- What is contrastive learning and mining?
- Is there a performance-computational cost trade-off in the current VPR pipeline?

References

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

[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.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[4] Gabriele Berton and Carlo Masone. Megaloc: One retrieval to place them all. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2861–2867, 2025.

[5] 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.

[6] 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.

[7] Gabriele Moreno Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: A hybrid approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2918–2927, 2021.

[8] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[10] Thomas Fel, Thibaut Boissin, Victor Boutin, Agustin Picard, Paul Novello, Julien Colin, Drew Linsley, Tom Rousseau, Rémi Cadène, Lore Goetschalckx, et al. Unlocking feature visualization for deep network with magnitude constrained optimization. *Advances in Neural Information Processing Systems*, 36:37813–37826, 2023.

[11] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[12] Ruben Härle, Felix Friedrich, Manuel Brack, Stephan Wäldchen, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. Measuring and guiding monosemanticity. *arXiv preprint arXiv:2506.19382*, 2025.

[13] Stephen Hausler, Tobias Fischer, and Michael Milford. Unsupervised complementary-aware multi-process fusion for visual place recognition. *arXiv preprint arXiv:2112.04701*, 2021.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023.

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

[17] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. *arXiv preprint arXiv:2504.02821*, 2025.

[18] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74: 90–109, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2017.09.013. URL `https://www.sciencedirect.com/science/article/pii/S0031320317303448`.

[19] 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.

[20] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, pages 444–461. Springer, 2024.

[21] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[22] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[23] Davide Sferrazza, Gabriele Berton, Gabriele Trivigno, and Carlo Masone. To match or not to match: Revisiting image matching for reliable visual place recognition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2849–2860, 2025.

[24] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[25] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015.

[26] Uddeshya Upadhyay, Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Probvlm: Probabilistic adapter for frozen vison-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1899–1910, 2023.

[27] Frederik Warburg, Marco Miani, Silas Brack, and Søren Hauberg. Bayesian metric learning for uncertainty quantification in image retrieval. *Advances in Neural Information Processing Systems*, 36:69178–69190, 2023.

[28] Mubariz Zaffar, Liangliang Nan, and Julian FP Kooij. On the estimation of image-matching uncertainty in visual place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17743–17753, 2024.