Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (37.th cycle)

Towards Robust Visual Geo-Localization: Cross-Domain, Sequential, and Fine-Grained Approaches for Place Recognition

Gabriele Trivigno

* * * * *

Supervisors
Prof. Barbara Caputo
Prof. Carlo Masone

Doctoral Examination Committee:
Prof. P.Garza, Politecnico di Torino,
Prof. M.Milford, Referee, QUT Centre for Robotics
Prof. N.Strisciuglio, Referee, University of Twente
Prof. J.Kooij, TU Delft
Prof. A.Furnari, University of Catania

Politecnico di Torino
September 11, 2025

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Gabriele Trivigno

Summary

Determining the geographic origin of an image, *i.e.* answering the fundamental question "Where was this picture taken?" constitutes a foundational challenge in computer vision and robotics with far-reaching implications. This capability to localize visual content across diverse environments underpins critical applications spanning from large-scale place recognition (with meter-level precision) to fine-grained visual localization (achieving centimeter-level accuracy). The advancement of these fields has been propelled by the proliferation of camera-equipped devices, the growing availability of geotagged imagery, and increasing demands for autonomous systems operating in unstructured environments. Applications range from consumer technologies like augmented reality and intelligent photo organization to robotic navigation for autonomous vehicles, assistive tools for the visually impaired, and large-scale geospatial analysis. Within this landscape, the literature has evolved to address two principal instantiations of the problem: Visual Place Recognition (VPR), which identifies coarse locations through image retrieval, and Visual Localization, which estimates precise 6-degrees-of-freedom camera poses, often within pre-built 3D maps.

This thesis makes several contributions advancing both paradigms. We first establish a comprehensive benchmarking framework for VPR, analyzing architectural choices, feature aggregation methods, and training strategies to derive practical insights for real-world deployment. Our findings reveal that lightweight CNNs often outperform complex models when optimized efficiently, while careful pipeline design can drastically reduce computational costs without sacrificing accuracy.

Recognizing the fundamental limitations of single-image retrieval approaches in Visual Place Recognition, we conduct a systematic investigation of sequence-based methods that leverage temporal information for improved localization robustness. While conventional VPR systems process frames independently, we demonstrate that sequential analysis of image streams yields significant performance gains, particularly in challenging scenarios affected by perceptual aliasing or viewpoint variations. To this end, we establish a comprehensive taxonomy of sequential descriptor architectures, analyzing their frame aggregation mechanisms and inherent design trade-offs, extending beyond traditional metrics to assess practical deployment factors including computational efficiency and memory requirements across diverse

datasets. Secondly, we introduce SeqVLAD, a novel sequential descriptor that holistically encodes multi-frame inputs through an innovative spatiotemporal aggregation layer. The proposed method addresses critical limitations of existing sequence matching approaches, mainly their quadratic complexity scaling and sensitivity to motion assumptions, while maintaining computational efficiency. We further demonstrate that local-feature based re-ranking, often overlooked in VPR, is decisive in overcoming domain shifts (e.g., day-night variations) and occlusions. To support our analysis, we introduce two challenging datasets that purposefully contain heavy domain shifts, and severe occlusion in the crowdsourced queries that we collected. These new datasets (SF test-night and SF test-occlusions) remain to this day unsolved by state-of-the-art methods.

Furthermore, having underlined the main bottleneck in scalability of conventional paradigms, we challenge them by proposing a scalable classification framework for VPR that replaces contrastive learning with an Additive Angular Margin Classifier, enabling fast, database-size-agnostic inference while maintaining fine-grained precision.

Finally, for precise 6-DoF localization, we develop a training-free pose refiner that generalizes across scene representations (e.g., point clouds, neural radiance fields) by leveraging pre-trained deep features in a render-and-compare paradigm, achieving state-of-the-art accuracy without per-scene optimization.

Together, these contributions present a holistic advancement across the spectrum of image localization capabilities. The thesis systematically addresses various facets of the localization spectrum from single-image place recognition to sequence-based methods and finally precise 6-DoF pose estimation, establishing new state-of-the-art performance at each level of spatial granularity. By unifying innovations in representation learning, temporal modeling, and geometric verification within a coherent framework, this work provides both theoretical insights and practical solutions for real-world visual localization systems operating under varying precision requirements and environmental constraints.

Contents

Chapter 1

Introduction

1.1 Image Localization

Determining the geographic origin of an image means in essence to answer the question "Where was this picture taken?". Such quest is a foundational challenge in computer vision and robotics, enabling systems to localize visual content across diverse environments. This capability underpins applications ranging from large-scale place recognition (with meter-level precision) to fine-grained visual localization (achieving centimeter-level accuracy). Over the years, advances in this field have been driven by the proliferation of camera-equipped devices, the growing availability of geotagged imagery, and the demand for autonomous systems operating in unstructured environments. Applications span consumer technologies like augmented reality and photo organization, robotic navigation for self-driving vehicles, assistive tools for individuals with visual impairments, and offline analysis of large data collections. Depending on the use case, localization granularity can vary significantly: a "place" may refer to a named landmark, a coarse geographic region, or an exact 6-degree-of-freedom camera pose. This versatility has attracted interdisciplinary research, with contributions from computer vision, robotics, and machine learning each emphasizing distinct aspects, from single-image retrieval to real-time, multi-modal localization in dynamic settings. Given the breadth of the localization problem, the literature has evolved to encompass several related tasks. This thesis focuses on two instantiation of this problem: Visual Place Recognition (VPR), and Visual Localization. A visual exemplification of these two tasks is depicted in Fig. 1.1.

1.1.1 Visual Place Recognition

Visual Place Recognition (VPR) is among the most popular sub-tasks under the umbrella of Image Localization. It addresses the fundamental challenge of determining where a photographic image was captured within a known map, based

solely on visual clues. Most commonly it is approached as a retrieval problem, in which a VPR system compares a query image against a database of geo-tagged reference images to identify the most likely location match. The success criterion for this task is typically defined by a spatial threshold (25 meters in mainstream literature [10, 306, 265, 96]).

In order to thoroughly map the area of interest in which queries need to be localized, the reference databases has to be densely sampled to be practically viable, making scalability an inherent challenge of the task. To be effective in practice, systems must handle databases ranging from tens of thousands [266, 265] to several million images [168, 289, 25], covering extensive urban and natural environments. This scale precludes the use of exhaustive pairwise comparison methods [191, 318], leading the field to develop sophisticated retrieval architectures. Modern approaches typically employ deep learning techniques to generate compact yet discriminative global descriptors, coupled with efficient approximate nearest-neighbor search algorithms to enable real-time operation even with massive databases.

The technical challenges in VPR are manifold and reflect the complexity of real-world environments. Viewpoint variation represents one of the most significant hurdles, where the same location may appear dramatically different when observed from alternative perspectives or viewing directions [30, 10]. Illumination changes, both daily and seasonal, can radically alter the visual appearance of scenes [265, 8, 164]. In these cross-domain scenarios, query images exhibit fundamentally different characteristics from the reference database. Examples include day-night transitions, weather variations (sunny vs. rainy conditions), or even cross-modal matching (e.g., satellite to ground-level perspectives) [26, 20, 29]. Dynamic elements such as moving vehicles, pedestrians, and changing vegetation further complicate the recognition task.

In robotic applications, VPR assumes particular importance as a critical component of simultaneous localization and mapping (SLAM) systems, where it enables loop closure detection and provides global localization capabilities in GPS-denied environments. The continuous nature of robotic perception introduces the additional consideration that temporal coherence and multi-frame observations can be leveraged to improve recognition, rather than processing single images in isolation. This temporal dimension allows to robustly embed the appearance of a scene, improving performance over single-frame approaches [174, 202, 170, 92].

The wide applicability of VPR techniques is demonstrated by its employment for applications such as loop closure in SLAM [7, 108, 61, 184], 3D reconstruction [234, 196, 207] and the closely related task of Visual Localization [231, 266, 267].

1.1.2 Visual Localization

Visual localization, also referred to as camera pose estimation, extends the capabilities of visual place recognition by estimating the precise six-degree-of-freedom

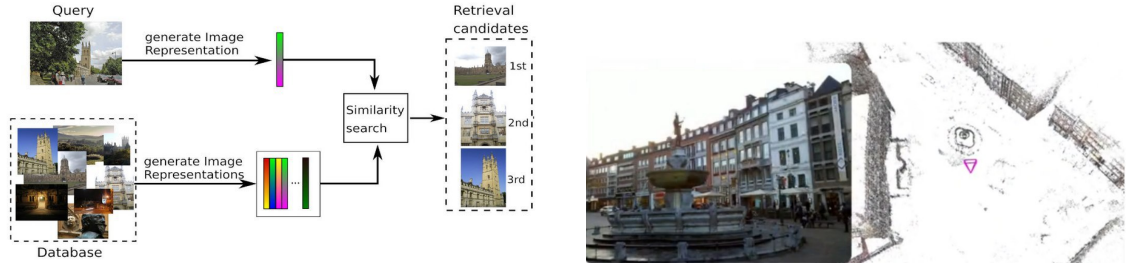


Figure 1.1: **Left**: The Visual Place Recognition task aims at finding the most similar image in geo-tagged database for a given query, to recover its location. **Right**:The Visual Localization tasks aims at recovering a query image camera pose against a known map of the environment.

(6-DoF) camera pose of a query image relative to a known environment. While place recognition retrieves the most similar reference images, pose estimation aims to compute accurate spatial coordinates and orientation, even for viewpoints not explicitly present in the reference dataset. Notably, these two tasks are frequently addressed jointly, as place recognition can effectively constrain the search space for pose estimation and is typically employed to this end in hierarchical localization pipelines [225, 224, 228, 113]. This approach proves particularly advantageous in large-scale maps.

Structure based approaches. The problem of 6-DoF camera pose estimation has been extensively studied, with traditional methods relying on geometric principles and sparse 3D point clouds. The fundamental component of these methods is the establishment of 2D-3D correspondences, matching image pixels coordinates to 3D point collected across different images.

To obtain the 3D representation of the scene, a mapping phase is necessary, during which image correspondences are computed to discover co-visible points. Camera calibration parameters and 3D scene structure are obtained accordingly, typically represented as a sparse point cloud [234, 117]. This mapping process can be performed either on unordered image sets (referred to as Structure-from-Motion (SfM) [234, 241]) or on input video sequences (common in robotics scenarios, referred to as SLAM).

Once the map has been obtained, a query image can be localized by first finding matches between its 2D pixels and 3D points of the scene [229, 227], and then by fitting a suitable camera pose [143]. Matching pipelines typically rely on *local features*, consisting of two components: (i) *keypoints*, which are discriminative image regions (e.g., corners) that should ideally be repeatable across viewpoints [229, 220, 74], and (ii) *descriptors*, which are high-dimensional vectors encoding local image content through intensity statistics or other handcrafted representations [9, 161,

220].

Learning based methods. As in many research field, the advent of deep learning has opened up new opportunities for visual localization.

Initially, research focused on replacing components of handcrafted pipelines with learnable alternatives. Popular works include neural network based keypoint detectors and descriptors [62, 220, 74, 314, 272], and learnable matchers to replace simple mutual neighbor search [226, 254, 316, 79]. These learned components often outperform traditional methods by leveraging data-driven priors, such as appearance invariance across viewpoints, keypoint stability, motion patterns, and scene geometry regularities.

Another research direction sought to replace entirely the geometric pipeline with end-to-end learnable models. It is the case of Absolute or Relative Pose Regressors [187, 130, 178, 242, 66] which direcly encode appearance and geometry of the scene inside a neural networks and thus allows to localize queries by directly regressing camera poses from images. Alternatively, Scene Coordinate Regressors predict 3D points corresponding to each image patch [39, 37, 45, 46]. Such approaches offer simplicity and the potential to learn powerful scene priors, improving robustness under challenging conditions. However, they suffer from several limitations: (i) reliance on large, annotated training datasets, which are expensive to acquire; (ii) poor generalization to unseen camera models, environments, or visual conditions [131]; (iii) reduced accuracy compared to geometric methods. Moreover, many end-to-end localization methods encode the map within their parameters, preventing generalization to novel scenes [271].

Pose Refinement. Pose refinement is a relevant research area in visual localization, aiming to improve an initial camera pose estimate through iterative optimization. Unlike standalone localization methods, refinement techniques are designed to complement existing approaches, acting as a post-processing step that boosts accuracy without requiring fundamental changes to the initial estimation. For instance, after a retrieval-based system provides a rough pose from a database image, refinement can align the query more precisely using direct (photometric/featuremetric) or indirect (geometric/reprojection) optimization [19, 210]. Similarly, in structure-based localization, where a 3D model provides an initial pose via feature matching, refinement can further reduce drift or misalignment by minimizing residuals in either pixel or geometric space [223, 80]. Recent advances have explored implicit scene representations for refinement, either through differentiable rendering of radiance fields [299] or feature field optimization [97], though these often face scalability limitations. Pose refinement serves as a powerful complement to traditional localization pipelines: while coarse localization methods provide an initial estimate, refinement techniques can recover precise camera poses by leveraging either geometric constraints or dense appearance matching, with the choice of approach often dictated by the application's requirements for accuracy, robustness,

and computational efficiency.

1.2 Contributions

This thesis proposes advances to the field of Visual Place Recognition (VPR), Visual Localization, and related tasks. Starting from an analysis of the VPR landscape, we outline the key challenges, and propose contributions towards achieving better scalability to large scale databases, as well as robustness to domain shifts through the exploitation of temporal information and matching of local features. At the fine grained end of the spectrum, for precise 6 DoF localization, we construct a pose refiner that generalizes across different domains and representations, without the need for specialized training.

Benchmarking Visual Place Recognition Pipelines. We introduce a thorough experimental framework to comprehensively analyze existing VPR systems. Through our open-source software, we define a standardized pipeline that encompasses all the core components both at training at inference time, including architectural choices, feature aggregation methods, and training strategies in geolocalization pipelines. We systematically evaluate all these aspects while accounting for practical constraints such as computational efficiency and memory usage. Our analysis reveals key insights for real-world deployment: (1) lightweight CNNs like ResNet-50 provide an optimal balance between accuracy and resource consumption, whereas vision transformers excel only with large-scale training data; (2) strategic optimizations—including image downscaling and selective negative mining—can drastically reduce computational costs (up to 60%) without sacrificing performance, sometimes even improving generalization by acting as implicit regularization. By establishing a standardized evaluation protocol, our work not only enables reproducible comparisons but also challenges the assumption that complex pipelines universally outperform carefully tuned, efficient alternatives. We also outline the main bottlenecks of existing pipelines, such as the mining algorithm necessary at training time, that requires careful design to avoid incurring in unfeasible computational requirements, and the impact of descriptor dimensionality. Overall, these findings provide actionable guidelines for future research and towards designing VPR systems tailored to specific hardware and dataset constraints.

Learning Sequential Descriptors for VPR. While single-image descriptors dominate visual place recognition (VPR), robotics and autonomous systems often provide multi-frame sequences that inherently contain richer spatial-temporal information. Current approaches typically handle sequences with post-hoc aggregation techniques through exhaustive similarity matrix searches, which are computationally inefficient and fail to fully exploit the temporal coherence in sequential data. Recent work has begun addressing this limitation by introducing *sequential descriptors* as compact representations that encode entire sequences holistically,

offering improved robustness against perceptual aliasing while being more scalable than traditional sequence matching. However, the field remains underdeveloped, with no systematic analysis of architectural trade-offs or practical deployment considerations. This work advances sequential place recognition by providing first a taxonomy categorizing sequence aggregation methods (e.g., recurrent, attention-based, or learned pooling) based on their ability to preserve temporal structure while maintaining discriminativity. Then, we construct a comprehensive benchmark evaluating not only accuracy but also computational efficiency and scalability across diverse datasets. Lastly, we introduce *SeqVLAD*, a novel aggregation layer that explicitly models temporal order through learnable feature binding, achieving state-of-the-art results while remaining lightweight. Our findings provide both theoretical insights into sequence encoding and practical guidelines for real-world deployment, particularly in robotics where sequential data is inherently available.

Cross Domain Robustness and Novel Challenging Datasets. We investigate the untapped potential of image matching methods for re-ranking in visual place recognition (VPR), motivated by the hypothesis that their reliance on local features grants inherent robustness to domain shifts (e.g., day-night variations) and occlusions, which represent two critical challenges for real-world systems. While spatial verification techniques have long been used for geometric validation in structure-from-motion and fine grained localization tasks, their systematic application to re-ranking predictions of VPR models remains unexplored, with prior studies limited to ad-hoc comparisons across incompatible retrieval pipelines. To address this gap, we conduct the first controlled benchmark of image matching methods for re-ranking in VPR, ensuring fair evaluation by: (1) using identical candidate pools for all methods, (2) standardizing feature backbones where feasible, and (3) quantifying computational costs under uniform hardware. Our experiments reveal that modern re-ranking can achieve near-perfect recall on existing datasets when combined with state-of-the-art retrieval. To support further research and validate our hypothesis, we introduce two novel datasets targeting the most demanding VPR conditions: *SF test-night* and *SF test-occlusions*, comprising Flickr-sourced nighttime and dynamically occluded queries paired with the city-scale SF-XL database. These contributions collectively demonstrate that local-feature-based re-ranking is not merely complementary to retrieval but often decisive in overcoming perceptual aliasing, while our datasets provide the necessary tools to study these effects under realistic, adversarially-chosen conditions. Notably, to this day, 2 years after the publications of these datasets, it has been shown that recent advances in VPR have essentially solved all existing benchmarks, except for *SF test-night* and *SF test-occlusions*.

Place Recognition is not necessarily a retrieval problem. Building on previous insights drawn from our benchmark, we challenges the prevailing reliance

on contrastive learning for visual place recognition (VPR) by proposing a scalable classification framework that circumvents the computational bottlenecks of mining-based training while ensuring fast inference, that does not depend on the database-size. Existing classification methods tackle the problem of global geolocation, and suffer from inadequate precision for city-scale localization (errors $\leq$25m) due to their inability to model dense visual overlaps in urban environments. Our approach specifically addresses the fine-grained nature of metropolitan VPR through two key innovations: (1) an **Additive Angular Margin Classifier (AAMC)** that leverages prototype learning from angular margin losses to achieve discriminative yet efficient place categorization, and (2) a **Divide & Conquer** strategy that dynamically partitions the classification space to handle urban density while maintaining real-time performance. Crucially, we demonstrate that our classification outputs can optimally constrain the search space for subsequent retrieval stages, creating a hybrid pipeline that achieves superior accuracy to pure retrieval systems, particularly under strict latency budgets, while keeping inference times fixed independently w.r.t. database scale. Our framework establishes classification as a viable paradigm for city-scale VPR, offering substantial speedup over contrastive methods during training while providing deterministic inference costs critical for large-scale deployment.

Training-free Pose Refinement. Further on, we devote our attention to the fine-grained localization problem. We advance the field of visual localization by introducing a novel pose refinement framework that overcomes two fundamental limitations of existing approaches: (1) dependency on specific scene representations (e.g., SfM point clouds or implicit neural fields), and (2) reliance on per-scene or task-specific feature optimization. At the core of our method lies the key insight that generic, pre-trained deep features, which are well known to be robust estimators of perceptual similarity, turn out to possess an inherent capability to measure pose similarity as well when used in a *render-and-compare* paradigm. This discovery allows us to develop a highly generalizable refinement system that operates agnostically across diverse scene representations (meshes, point clouds, or radiance fields) without requiring specialized training. Our framework employs a particle filter-based optimizer to efficiently explore the pose space, based on the observation that hierarchical structure of deep features can be leveraged to achieve an optimal trade-off between geometric sensitivity and robustness to appearance variations. Extensive experiments demonstrate that this conceptually simple approach outperforms modern pose regression networks and matches or exceeds the accuracy of implicit refinement methods, while being uniquely scalable to large-scale environments. The versatility of our framework is further evidenced by its ability to serve as either a standalone refiner, a pose prior enhancer, or a post-processing step for feature-based methods, thus establishing a new paradigm for representation-agnostic visual localization.

All methodologies presented in this thesis are validated through extensive experiments across diverse datasets. We provide comprehensive ablation studies to justify design choices, comparative analyses against state-of-the-art baselines, and investigations of failure cases to delineate the boundaries of each approach. To ensure reproducibility and accelerate progress in the field, we have released open-source implementations of all key components—including our benchmarking frameworks, sequence-aware descriptors, hybrid classification-retrieval pipelines, and representation-agnostic pose refinement systems, as well as releasing the novel datasets. We believe that open source implementations enable future researchers to build upon our work while maintaining consistent evaluation standards across the community.

1.3 Thesis outline

In Chapter 2 we present a detailed overview of the literature related to all the topics that are subject of study in the thesis. In Chapter 3, Chapter 5, Chapter 4 and Chapter 6 we tackle the various challenges related to Visual Place Recognition. Lastly, in Chapter 7 we cover the finer end of the localization spectrum and discuss the problem of camera pose estimation.

Chapter 3 "Deep Visual Geo-localization Benchmark" [27] introduces the first comprehensive framework for evaluating visual geo-localization (VG) techniques, uncovering that advancements in the field have been predominantly incremental while exposing key shortcomings in current methodologies. By analyzing critical components of VG systems, ranging from backbone architectures to feature aggregation strategies, the benchmark underscores the substantial influence of design decisions on overall performance. Furthermore, this work proposes standardized evaluation protocols and metrics, addressing the previously unresolved challenge of consistent and systematic comparison within the domain.

Chapter 4 "Learning Sequential Descriptors for Sequence-based Visual Place Recognition" [170] builds on the observation that for several VPR applications, especially in robotics, multi-frame sequences are readily available, and hence their informative content can be used to enhance robustness in *sequential descriptors*. This chapter addresses key gaps in sequential VPR by first introducing a taxonomy of temporal aggregation techniques, then presenting a comprehensive benchmark evaluating their accuracy and efficiency. Finally, we propose SeqVLAD, a novel lightweight aggregation layer that explicitly models temporal order.

Chapter 5 "Are local features all you need for cross-domain visual place recognition?" [20] challenges conventional VPR paradigms by demonstrating how local-feature-based re-ranking can overcome critical domain shifts and occlusions that

confound global descriptor methods. We establish the first rigorous benchmark for image matching in VPR re-ranking, eliminating evaluation biases through standardized candidate pools, feature backbones, and computational cost metrics. To sustain future research on challenging domain gaps, we introduce two novel datasets, which remain unsolved to this day, providing essential testbeds for robust place recognition under adverse conditions.

Chapter 6 "Divide&Classify: Fine-Grained Classification for City-Wide Visual Geo-Localization" [270] rethinks the dominant contrastive learning paradigm in VPR by introducing a novel classification framework that overcomes fundamental scalability limitations in both training and inference. We address the precision gap of traditional classification approaches through two key innovations: (1) an Additive Angular Margin Classifier (AAMC) that enhances discriminative power for dense urban environments, and (2) a Divide&Conquer strategy that ensures robust learning. The resulting hybrid pipeline, where classification optimally constrains subsequent retrieval, achieves higher accuracy than pure retrieval systems while providing deterministic, database-size-invariant inference times. Our framework establishes classification as a practical solution for large-scale deployment under strict latency constraints.

Chapter 7 "The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement" [271] introduces a paradigm shift in visual localization by proposing a scene-agnostic pose refinement framework that eliminates the need for specialized training or scene-specific optimization. At its core, our method exploits a fundamental discovery: pre-trained deep features are robust pose-similarity estimators. Hence, we develop a particle filter-based optimizer that strategically leverages the hierarchical nature of these features to balance geometric precision with robustness to appearance variations. The resulting system operates across diverse scene representations, from point clouds to neural radiance fields, without requiring architectural modifications or fine-tuning. Experimental results demonstrate that this training-free approach outperforms learned pose regression networks and rivals state-of-the-art implicit refinement methods, while offering unprecedented scalability for large environments. The framework's versatility enables usage as either a standalone refiner, pose prior enhancer, or feature-based post-processing step, thereby establishing a new direction for representation-agnostic visual localization.

1.4 Publication list

We list here the publications of the author in chronological order.

- G. Berton, R. Mereu, G. Trivigno, C. Masone, G. Csurka, T. Sattler, B. Caputo. **Deep visual geo-localization benchmark** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR) 2022 [27];
- R. Mereu(*), G, Trivigno (*), G, Berton, C. Masone, B. Caputo, Barbara. **Learning Sequential Descriptors for Sequence-Based Visual Place Recognition** In *IEEE Robotics and Automation Letters* (RAL) 2022 [170];
- M. Planamente, G. Goletto, G. Trivigno, G. Averta, B. Caputo. **Toward human-robot cooperation: Unsupervised domain adaptation for egocentric action recognition** In *International Workshop on Human-Friendly Robotics* (HFR) 2022 [212];
- G. Berton(*), G. Trivigno (*), B. Caputo, C. Masone. **EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition** In *IEEE/CVF International Conference on Computer Vision* (ICCV) 2023;
- G. Trivigno (*), G. Berton(*), J. Aragon, B. Caputo, C. Masone. **Divide&Classify: Fine-Grained Classification for City-Wide Visual Geo-Localization** In *IEEE/CVF International Conference on Computer Vision* (ICCV) 2023;
- G. Berton (*), G. Trivigno (*), B. Caputo, C. Masone. **Jist: Joint image and sequence training for sequential visual place recognition** In *IEEE Robotics and Automation Letters* (RAL) 2023;
- G. Barbarani, M. Mostafa, H. Bayramov, G. Trivigno, G. Berton, C. Masone, B. Caputo. **Are local features all you need for cross-domain visual place recognition?** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition - Image Matching Workshop* (CVPRW) 2023
- E. Giglio, G. Luzzani, G. Terranova, G. Trivigno, A. Niccolai, F. Grimaccia. **An efficient artificial intelligence energy management system for urban building integrating photovoltaic and storage** In *IEEE Access* (2023);
- M. Dutto, G, Berton, D. Caldarola, E. Fanì, G. Trivigno, C. Masone. **Collaborative Visual Place Recognition through Federated Learning** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition - FedVision Workshop* (CVPRW) 20224;

- C. Cuttano, G. Rosi, G. Trivigno, G. Averta. **What does CLIP know about peeling a banana?** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition - Multimodal Reasoning Workshop* (CVPRW) 2024;

- G. Berton, G. Goletto, G. Trivigno, A. Stoken, B. Caputo, C. Masone. **EarthMatch: Iterative Coregistration for Fine-grained Localization of Astronaut Photography** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition - Image Matching Workshop* (CVPRW) 2024;

- G. Trivigno, C. Masone, B. Caputo, T. Sattler. **The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR) 2024;

- G.Barbarani, F. Vaccarino, G. Trivigno, M. Guerra, G. Berton, C. Masone. **Scale-Free Image Keypoints Using Differentiable Persistent Homology** In *International Conference on Machine Learning* (ICML) 2024;

- C. Cuttano, G. Trivigno, G. Rosi, C. Masone, G. Averta. **SAMWISE: Infusing wisdom in SAM2 for Text-Driven Video Segmentation** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR) 2025;

- D. Sferrazza, G. Berton, G. Trivigno, C. Masone. **To Match or Not to Match: Revisiting Image Matching for Reliable Visual Place Recognition** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition - Image Matching Workshop* (CVPRW) 2025.

- C. Cuttano (*), G. Trivigno (*), G. Averta, C. Masone. **SANSA: Unleashing the Hidden Semantics in SAM2 for Few-Shot Segmentation** arXiv under submission;

Chapter 2

Background and related works

The purpose of this chapter is to give an overview of the different facets of the problem of localizing images, which has been analyzed in different literature contexts.

2.1 Finding image locations

Visual place recognition (VPR), i.e. the task of estimating the geographic location of a query image using only its visual content, is a cornerstone problem in computer vision and robotics. However, the definition of "localization" varies widely depending on context, leading to a diverse set of methodologies tailored to different applications, scales, and precision requirements. At one end of the spectrum, coarse-grained retrieval-based approaches identify semantically relevant regions (e.g., "Paris" vs. "Tokyo"), while fine-grained techniques aim for metric-accurate pose estimation within a pre-mapped environment. The choice of method often hinges on factors such as available reference data (e.g., GPS-tagged images, 3D maps, or satellite imagery), computational constraints, and the desired output granularity. Over time, these varying demands have given rise to a rich taxonomy of solutions, each addressing distinct challenges in visual geo-localization. As this thesis is focused not only on Place Recognition, but also Visual Localization, the rest of the section will cover both, as well as an overview of related usages of place recognition methods. In the following, we survey the key paradigms that have shaped this field:

- **Coarse Global-scale Localization** treats geolocalization as a large-scale classification task, where the Earth is discretized into geographic cells, and the predicted class centroid serves as the estimated location. Early works like PlaNet [292] and RevIm2GPS [275] demonstrated the feasibility of this approach, offering efficient inference—often just a single forward pass—without costly retrieval or matching steps. However, accuracy is inherently tied to the partitioning strategy: finer grids improve resolution but exacerbate data

sparsity and model complexity. To address this, hierarchical methods [183, 260] leverage geographic structures (e.g., city → region → country), while combinatorial [239] or learned partitioning [138, 118] optimizes cell boundaries. Though scalable, these methods trade precision for efficiency, with error thresholds ranging from 1 km to continent-level (1000 km) for ambiguous scenes (e.g., deserts or generic landscapes). Notably, fine-grained urban variants exist [100], but their applicability remains limited to small areas (1.56 km^2);

- **Visual Place Recognition** is typically cast as a retrieval problem, where a query image is localized by comparing it to a database of geo-tagged images. The problem is usually framed at the city-level, with an accuracy of ∼ 25 m. Initially addressed with Bag-of-Words representations based on local handcrafted features [248, 124] the field evolved to exploit global representations extracted from deep neural networks [10, 216, 14]. A closely related branch of literature has studied how to exploit multi-frame information to enhance localization accuracy, which is especially suited for robotic scenarios [92, 174, 170]. In modern years the field has evolved to leverage large scale datasets [25, 30, 31], and more recently with the rise of Visual Foundation Models [195] research has shown that these can be adapted to the VPR task to obtain remarkable generalization capabilities [120, 119];

- **Camera Pose Estimation**: At the fine-grained end of the spectrum, visual localization aims to recover the precise 6-DoF camera pose (position and orientation) of a query image within a pre-mapped environment. This task is foundational for applications like augmented reality and robotic navigation, where metric accuracy (often centimeter-level) is critical. The canonical pipeline leverages Structure-from-Motion (SfM) [234] to construct a sparse 3D point cloud, where scene geometry is encoded through triangulated local features from reference images. During inference, 2D-3D matches between the query and the model are established via feature matching [229, 227, 226], followed by perspective-n-point (PnP) solvers [143] to estimate the pose. However, exhaustive matching is computationally prohibitive at scale; thus, hierarchical approaches first use VPR techniques [10, 25] to filter candidate reference images with covisible regions [225, 113], drastically reducing search space.

  In robotics, this pipeline is tightly coupled with SLAM systems, where VPR serves dual roles: (1) initial coarse localization for bootstrapping pose tracking, and (2) loop closure detection to correct accumulated drift by recognizing revisited locations [7, 108, 246]. Recent advances explore alternatives to sparse SfM models, such as dense point clouds (e.g., from LiDAR or multi-view stereo [255, 241]), meshes [197, 312], or neural radiance fields (NeRFs)

[171, 157]. These representations enable direct pixel- or feature-wise alignment, bypassing traditional matching bottlenecks.

Despite the popularity of matching-based approaches, pose refinement techniques either direct (e.g., photometric alignment [80]) or indirect (e.g., reprojection error minimization [210]), have been studied for correcting initial estimates. While direct methods struggle with appearance changes, their integration with learned features [276, 223] or implicit maps [51] promises robustness. This synergy between VPR (for efficient hypothesis generation) and geometric optimization (for precision) underscores the interdependence of retrieval and pose refinement in modern localization systems.

In the rest of the chapter we delve into the two main topics on which this thesis is focused, i.e. Visual Place Recognition Sec. 2.2 and Visual Localization Sec. 2.3

2.2 Visual Place Recognition

Handcrafted Feature-Based Methods for Visual Place Recognition. Pioneering works for Visual Place Recognition (VPR) explored the usage of handcrafted visual features designed to capture invariant scene properties. Initially, influential works were based on local feature descriptors such as SIFT [161] and its follow up SURF [22], focused on efficiency. SIFT, in particular, revolutionized feature extraction by detecting keypoints using a Difference-of-Gaussians approach and describing them via gradient histograms, achieving robust invariance to rotation, scale, and moderate illumination changes. These descriptors became cornerstones for early VPR systems, as demonstrated by [238, 6, 185] who leveraged them for localization and mapping taks.

To improve computational efficiency, later works introduced binary descriptors like ORB [222] and CenSurE [1] which traded some discriminative power for real-time performance [137]. However, matching individual local features remained computationally expensive, leading to the widespread adoption of Bag-of-Words (BoW) representations [248]. BoW approaches clustered visually similar features into a pre-trained visual vocabulary (which could contain hundreds of thousands of *visual words*), enabling efficient image retrieval by comparing histograms of visual word occurrences. This paradigm was successfully applied in VPR by [7, 108].

Further refinements to feature aggregation included VLAD [124] which stored first-order statistics of descriptor residuals relative to cluster centers, and Fisher Vectors [203] which incorporated higher-order statistics for greater discriminability. DenseVLAD [267] extended this idea by densely sampling SIFT descriptors across images, proving competitive even against early deep learning methods [228].

Alongside local features, **global descriptors** provided computationally efficient

alternatives by encoding entire images into compact signatures. The GIST descriptor [193] for instance, captured scene layout using oriented edge filters and was employed in panoramic VPR systems by [186, 246]. Other global variants, such as Whole-Image SURF (WI-SURF) [17] offered trade-offs between discriminability and speed.

Despite their successes, handcrafted methods faced fundamental limitations under severe appearance changes (e.g., day-night or seasonal transitions), as they relied on low-level visual patterns rather than semantic understanding. Recent efforts like CoHOG [305] attempted to bridge this gap by focusing on entropy-rich regions and employing convolutional-regional matching, but the field ultimately shifted toward data-driven deep learning approaches for greater robustness.

Deep Learning-based VPR. The advent of deep learning marked a pivotal shift in visual place recognition (VPR), as Convolutional Neural Networks (CNNs) rapidly surpassed traditional hand-crafted features in robustness and discriminative power. A seminal study [16] revealed that off-the-shelf CNN features, pretrained on ImageNet for classification tasks, inherently encoded semantically rich representations that generalized remarkably well to place recognition. This discovery catalyzed a wave of research into more sophisticated feature aggregation techniques, designed to harness the spatial and hierarchical nature of CNN activations.

Among the earliest strategies were global descriptor pooling methods such as MAC [218], which exploits max-pooling to capture the strongest responses across convolutional feature maps, and its extension, R-MAC [264], which improved discriminability by aggregating regional maxima at multiple scales. Concurrently, SPoC [14] demonstrated that sum-pooling deep features could enhance translation invariance while preserving critical spatial information. GeM [216] proposed a learnable pooling and a mining strategy, based on 3D correspondences via SfM, to obtain compact and informative descriptors.

However, the field underwent a paradigm shift with the introduction of NetVLAD [10], which redefined end-to-end trainable architectures for VPR. By integrating a differentiable VLAD (Vector of Locally Aggregated Descriptors) layer into a CNN backbone, NetVLAD enabled learnable feature aggregation with soft-assignment mechanisms, effectively clustering deep features in a weakly supervised manner using only geotagged images. This innovation significantly boosted performance and inspired subsequent advances in attention-driven and multi-scale feature learning.

To further refine descriptor discriminability, later works introduced spatial and contextual priors. The Contextual Reweighting Network (CRN) [134] employed spatial attention to dynamically emphasize semantically salient regions, while SFRS [96] combined multi-scale feature extraction with adaptive cropping to mitigate viewpoint and appearance variations. Meanwhile, training stability was addressed by the Stochastic Attraction-Repulsion Embedding (SARE) loss [158], which explicitly optimized descriptor space by simultaneously pulling matching pairs together

and pushing non-matching pairs apart across multiple negatives.

These approaches predominantly relied on architectures like VGG16 [245], often coupled with NetVLAD-derived descriptors. However, the high dimensionality of these representations (e.g., 32K-D before PCA compression) introduced practical challenges in memory and computational efficiency, prompting later research into compact yet discriminative alternatives [27, 25].

Modern Methods for VPR. Recent progress in visual place recognition has been driven by innovations in network architecture, the availability of increasingly large-scale datasets, and the adoption of transformer-based models. In particular, the introduction of SF-XL [25], a comprehensive dataset containing 40 million images spanning the entire city of San Francisco, highlighted critical inefficiencies in existing methods. Traditional NetVLAD-based descriptors exhibit prohibitive memory and computational requirements during inference, while the conventional contrastive learning paradigm with hard-negative mining proves increasingly ineffective as dataset sizes grow to such unprecedented scales. These facts underscored the need for more scalable approaches in visual place recognition systems. To this end, [25] introduced novel training paradigm that allowed to handle such large scale datasets, while also maintaining compact descriptor size. Thanks to the ability to learn from large scale data collections, this methods showcased great generalization capabilities across diverse datasets. Concurrently, [148] proposed a general formulation of the contrastive loss that does not require hard pair mining. In later chapter of this thesis, we show a novel approach for VPR that exploits a classification framework to both (i) enable training time scalability, and (ii) obtain efficient inference pipelines avoiding the need to search through the entire database at test time.

The release of novel and more challenging dataset drove the research landscape, as it was the case for GSV-Cities [2], a dataset containing 1.6M street-view images. By capturing diverse geographic locations, lighting conditions, and seasonal variations, it enabled models to learn more generalizable features, reducing overfitting to specific environments. The key feature of this dataset is that it divided images into *places*, wherein each place was depicted from different perspectives. This enabled a discretization of the dataset into classes, making it suitable for training with retrieval losses such as the MultiSimilarity [2], which require the dataset to be divided into discrete categories. This shift toward data-centric training paralleled advances in efficient feature learning, such as MixVPR [3], which replaced traditional descriptor aggregation with a lightweight MLP-based feature-mixing approach. Remarkably, MixVPR achieved state-of-the-art accuracy while being trainable in under a day—a stark contrast to the computationally intensive pipelines of earlier deep learning methods.

The rise of vision transformers further expanded the VPR toolkit, with models

like R2Former [318] and TransVPR [284] leveraging self-attention to capture long-range spatial relationships, proving particularly effective for viewpoint-invariant recognition. Meanwhile, foundation model adaptation emerged as a promising direction, with AnyLoc [129] repurposing the recently released VFM Dinov2 [195] for VPR without task-specific fine-tuning. The subsequent research direction has been to further exploit the potential of DinoV2. SALAD [120] showed that, by finetuning the last layers of Dino, coupled with a flexible replacement for NetVLAD, one could achieve unprecedented generalization across datasets. Its follow up, CliqueMining [119] introduced a scalable hard-negative mining to enable training on multiple datasets, further improving performances.

These innovations have collectively pushed VPR toward real-time, city-scale deployment, even under extreme appearance changes (e.g., day-night shifts or seasonal variations). Practical adoption has been further accelerated by efficient retrieval techniques, including product quantization [123] for memory-efficient descriptor storage, inverted file indexing [15] for rapid candidate filtering, and hierarchical navigable small-world (HNSW) graphs [166] for approximate nearest-neighbor search at scale. Together, these advances have transformed VPR from a computationally demanding task into a viable component of real-world navigation systems.

Lastly, VPR based approaches have been explored under different perspectives. Examples include localization from remote-sensing images [29, 28, 72, 317], combined with multi-modal sensory information such as Lidar and point clouds [311, 163, 273, 32], for localization and mapping inside the human body [179, 180], for lightweight biologically inspired network [11, 47], for localization in indoor scenarios [304, 255, 271] and in federated settings [76].

2.2.1 Sequence-based Visual Place Recognition

Traditional methods for Place Recognition on sequence of images relied on the well established paradigm of Sequence-based matching [108, 174], particularly effective in addressing extreme appearance variations [237]. SeqSLAM [174] was among the pioneering works that introduced sequence-based matching on handcrafted local features, later extended by [202] to handle variable platform speeds, and by [173] to use learned descriptors. The traditional approach operates in two stages: first constructing a similarity matrix by pairwise comparison of query and database frame descriptors, then aggregating these scores under simplifying assumptions (e.g., constant velocity or no stops). While this framework has demonstrated success, its reliance on these restrictive assumptions limits generalization to real-world scenarios [237].

Subsequent research has sought to relax these constraints through various innovations. Some approaches incorporate egomotion information [189], while others develop more sophisticated matching mechanisms [202]. Additional improvements

include velocity-robust sequence searching [281, 278], hashing-based match selection [280], route-invariant matching or temporal diffusion processes [279, 310], and trajectory-aware attention mechanisms for SLAM [199]. However, as noted in [93], these methods fundamentally depend on single-image descriptors typically trained without consideration for their downstream sequential aggregation, potentially limiting their effectiveness.

Subsequent works identified two fundamental limitations of traditional sequence matching: (i) susceptibility to false positives from misleading single-image matches that could be detected through sequential analysis, and (ii) computational complexity scaling linearly with both database size and sequence length [92]. These limitations have spurred the development of sequential descriptor methods that compactly represent entire sequences through single descriptors, simultaneously capturing temporal information while enabling efficient sequence-to-sequence matching.

The concept of sequence-level descriptors has been explored in adjacent fields like video re-identification [297, 24, 295] and 3D-based localization [273, 71]. In VPR, Pioneering work by Facil et al. [82] introduced three simple baselines as foundational techniques: descriptor concatenation, fully-connected feature fusion, and LSTM-based temporal integration. These approaches were subsequently extended and evaluated on large-scale datasets like MSLS [289]. Alternative approaches include non-learnable discrete convolutional aggregators [94] and learnable 1D temporal convolutions to aggregate frame-level descriptors [92].

While existing works demonstrate the viability of sequential descriptors through diverse architectural approaches [92, 82, 289, 94], the literature lacks a systematic comparison and categorization of these methods. This thesis addresses that gap by presenting: (1) a comprehensive taxonomy categorizing sequential descriptors by their temporal fusion mechanisms, and (2) an extensive experimental evaluation that not only analyzes existing approaches but also introduces novel architectures including Transformer-based models and SeqVLAD - the first aggregation layer specifically designed for sequence-based VPR.

The work of SeqVLAD was later extended by [313] with spatio-temporal attention, and [31, 149] proposed to exploit large-scale single image dataset to boost generalization in sequence-based tasks.

2.2.2 Image Matching for Retrieval

In retrieval systems, a common strategy to improve performance is to adopt a post-processing step. The idea is that retrieval methods are optimized for recall, and are thus good at filtering out outliers from large data collections. However, they can suffer from perceptual aliasing issue and can thus benefit from more expensive post-processing methods to improve precision of the predicted candidates [20, 191, 257, 43]. To this end, various strategies can be employed from the literature on image-matching, which aims to establish correspondences between images and can

thus be used for spatial verification between a query and a retrieved images. This section provides an overview of these methods.

Keypoint Detection and Description. The identification of repeatable keypoints along with their corresponding descriptors has been a fundamental challenge in computer vision for many years. Early methods relied on handcrafted techniques following a detect-then-describe strategy, often utilizing local image derivatives [161, 22, 184]. With the rise of deep learning, data-driven approaches became increasingly prevalent. Initial efforts [244, 262] utilized contrastive learning to train convolutional networks for local descriptor extraction. SuperPoint [62] introduced a self-supervised approach by generating synthetic shapes for neural network training. Later developments shifted toward a unified detect-and-describe framework, where keypoints emerge as local maxima within the feature space [74, 220, 272, 314]. DeDoDe [77] presents an alternative approach by decoupling detection and description optimization, enhancing repeatability through 3D consistency constraints. Its successor, Steerers [36], incorporates rotation-invariant descriptors, broadening applicability in spatial and medical domains [29, 250, 209].

Image Matching. The goal of image matching is to establish pixel-level correspondences across different views of a scene. Traditional techniques relied on mutual nearest-neighbor searches over local keypoint descriptors [220], though this approach can be error-prone due to the lack of global context. Solutions include geometric verification via RANSAC [83] or learned matchers like SuperGlue [226], which employs graph neural networks. Unlike SuperGlue, which operates post-matching, LoFTR [254] eliminates the detection stage entirely, presenting a detector-free method that leverages transformer-based attention mechanisms to incorporate global context, thereby improving performance in low-texture and repetitive-pattern scenarios. Inspired by LoFTR, several subsequent works have adopted this detector-free paradigm [283, 259, 111, 49, 35, 316]. Alternatively, dense feature matching methods aim to estimate all possible pixel correspondences, enabling dense image warping [78, 79]. While these methods operate purely in 2D, recent approaches like Dust3r [285] and Mast3r [146] ground matches in 3D by solving uncalibrated 3D reconstruction before deriving correspondences.

2.3 Visual Localization

Visual localization addresses the problem of estimating the precise 6-DoF camera pose of a query image within a known environment, in the form of a collection of images representing the scene, or a 3D structure. A simple approach to solve this task is to utilize image retrieval to find the closest match to a query image in a database of images, and use the retrieved image as an approximation of the query pose. This technique, referred to as **image-based localization**, was employed in early approaches. Employing methods for Place Recognition [162, 53, 10] and loop

closure [88, 61, 184] proved effective for localization at scale [231, 266, 267], while being robust to domain changes [10, 265, 253]. In particular, popular choices were either compact VLAD-based representations such as DenseVLAD [265], which aggregates densely extracted SIFT descriptors, and NetVLAD, which employs learned CNN aggregated features. Similarly, FAB-MAP [61], a Bag-of-Words (BoW) retrieval approach, has been widely adopted in robotics for long-term navigation due to its probabilistic modeling of visual word co-occurrence [81, 155]. While efficient, these retrieval-based methods are inherently limited by their reliance on discrete database poses, leading to coarse localization accuracy.

To achieve finer pose estimation, structure-based localization methods leverage sparse 3D models reconstructed via Structure-from-Motion (SfM) [234]. These models associate triangulated 3D points with visual features, enabling precise pose estimation through 2D-3D matching followed by Perspective-n-Point (PnP) solvers [143]. To mitigate the computational cost of exhaustive matching, hierarchical pipelines first retrieve candidate database images using place recognition techniques [273, 25], then establish geometric correspondences only within covisible regions [224, 225, 113]. Alternatively, researchers experiment with different scene representations other than SfM point cloud, moving towards more flexible dense representations (such as dense clouds from Multi-View Stereo (MVS) or Lidar [241, 235, 255] or meshes [197, 41, 312]), or even implicit representations stored inside neural networks [171, 299, 157, 165]. In implicit representations, geometry and appearance of the scene are encoded within neural networks. Early implementation of this idea cast the problem as direct Pose Regression [131, 130, 242, 178]. Alternatively, Scene Coordinate Regressors predict 3D scene coordinates per patch [39, 37, 45, 46].

While matching based methods obtain state-of-the art results, a relevant research direction is that of Pose Refinement methods, based on the idea of obtaining direct pixelwise alignment without exhaustive feature matching. Recent trends show that renderable scene representations can be exploited to refine poses efficiently, complementing traditional pipelines with minimal computational overhead. Pose refinement techniques iteratively optimize an initial estimate by minimizing photometric, featuremetric, or reprojection errors. Direct methods, such as those in SLAM systems [80] align pixels or features using gradient-based optimization [147, 167], while indirect methods minimize geometric residuals from 2D-3D matches [210]. Hybrid approaches like PixLoc [223] train features end-to-end for featuremetric alignment, The advent of NeRFs [171] further enabled view synthesis and pose refinement via photometric optimization [299, 154] or feature-field rendering [97, 177] However, these methods often require per-scene training and struggle to scale. Other NeRF-based refiners [299, 51] invert the field and backpropagate rendering errors through implicit models. In a later chapter of this thesis we extend these ideas by showing that generic deep features inherently capture fine-grained pose discrepancies, echoing findings from perceptual similarity studies [308] while

generalizing to geometric alignment. The proposed approach leverages generalizable pre-trained features, eliminating the need for scene-specific optimization while maintaining compatibility with arbitrary scene representations such as meshes [197], which can be efficiently rendered in less than a millisecond, thanks to mature graphics primitives.

Chapter 3

Benchmarking Deep Neural Networks for Visual Place Recognition

In this chapter, we present a comprehensive benchmarking framework for Visual Place Recognition that addresses critical gaps in current evaluation methodologies. While recent years have seen rapid progress in VPR techniques, the field lacks standardized protocols to fairly compare different approaches and properly assess their real-world applicability. Our framework provides researchers with tools to systematically evaluate architectural choices, training strategies, and computational trade-offs under controlled conditions.

The benchmark reveals several important insights that challenge common assumptions in the field. Contrary to prevailing trends, we find that CNN architectures like ResNet-50 can achieve competitive performance with significantly lower resource requirements compared to more complex alternatives. We also demonstrate that vision transformers, while computationally demanding, exhibit superior scaling properties with larger training datasets. Through careful experimentation, we identify and propose solutions to two major bottlenecks affecting practical deployments: the computational overhead of hard negative mining during training and the storage demands of high-dimensional descriptors at inference time.

Available as an open source toolkit at [1]this link, our framework enables reproducible evaluation while providing actionable guidelines for developing efficient VPR systems. The included analysis of engineering considerations—from optimal input resolutions to effective data augmentation strategies—offers practical recommendations for balancing accuracy and efficiency in real-world applications.

The work described in this chapter has been previously published in a paper:

[1]https://github.com/gmberton/deep-visual-geo-localization-benchmark

- G. Berton, R. Mereu, G. Trivigno, C. Masone, G. Csurka, T. Sattler, B. Caputo. **Deep visual geo-localization benchmark** In *Conference on Computer Vision and Pattern Recognition* (CVPR) 2022 [27]

	Vanilla	Resize (80%)	Data augm. (brightness = 2)	Pred. refinement (*nearest crop*)	PCA (2048)	CRN [134]
R@1	63.4	64.3	68.6	67.0	56.6	68.8

Table 3.1: Example of how results can be influenced by little train or test time changes to the VG pipeline. Recall@1 for a ResNet-18 with NetVLAD trained on Pitts30k and tested on Tokyo24/7. Results are thoroughly discussed in later sections.

3.1 Introduction

Visual Place Recognition (VPR) has emerged as a fundamental capability for numerous real-world applications, ranging from autonomous navigation systems and augmented reality to large-scale image organization and robotic mapping. The ability to determine a camera's location by comparing visual input against a reference database enables crucial functionalities such as loop closure in SLAM systems, drone homing, and assistive navigation for the visually impaired. However, the field's rapid progress has led to a proliferation of approaches with varying architectures, training protocols, and evaluation metrics, making objective comparisons challenging. Establishing best practices in this domain requires rigorous comparison of methods under standardized conditions that control for implementation variables while isolating their true algorithmic contributions. This chapter addresses that need by introducing a comprehensive benchmark framework derived from an extensive analysis of existing literature, enabling researchers to systematically evaluate different approaches and identify the most effective solutions for specific application scenarios.

The problem of coarsely determining the location where an image was captured, using a database of images from known places, is referred to as *Visual (Image) Geolocalization* (VG) [134, 158, 307] or *Visual Place Recognition* (VPR) [162, 91]. This task is typically approached through image matching and retrieval techniques.

The field has experienced significant growth in recent years, as evidenced by the rising volume of publications [267, 162, 134, 10, 208, 265, 158, 216, 133, 96, 289, 105, 309, 169, 26, 109, 92, 278, 94, 288, 290, 296, 53]. However, this expansion is accompanied by two key challenges:

i) **Overemphasis on single-metric optimization**. Current evaluations often focus narrowly on recall rates while neglecting critical factors such as computational efficiency, hardware demands, and scalability. These aspects are vital for real-world VG systems. For example, a modest reduction in accuracy (e.g., 5%) may be acceptable if it leads to substantial savings in descriptor size (e.g., 90%), enabling better scalability. Similarly, descriptor dimensionality and inference speed are decisive for real-time applications on resource-constrained platforms.

ii) **The absence of a standardized evaluation framework**. Comparisons between methods are frequently made using inconsistent setups (*e.g.* , data augmentation, initialization, training data) [240, 134, 306], obscuring the true impact of algorithmic contributions and making it difficult to assess individual components. Tab. 3.1 illustrates how minor engineering choices can significantly alter performance metrics.

While prior benchmarks for VPR [306] and Visual Localization [211, 228] provide valuable insights, they do not resolve these limitations. To address these gaps, this chapter presents an open-source benchmark designed as a comprehensive toolkit for developing, training, and evaluating diverse VG architectures, with modular control over each pipeline component. This framework enables systematic analysis of how design choices affect performance while providing real-time metrics such as parameter counts, FLOPs, and descriptor dimensions.

Using this framework, we conduct extensive experiments to identify optimal configurations for real-world scenarios, offering practical guidelines tailored to dataset characteristics and hardware constraints. Regarding practical insights, Our findings suggest that ResNet-50 [107] strikes a favorable balance between accuracy, computational cost, and model size, despite being overlooked in the literature. We also find that Visual Transformers, when trained on larger datasets, can surpass CNN backbones in geo-localization performance. Additionally, we identify hard negative mining and high descriptor dimensionality as the two main bottlenecks affecting the scalability of VPR pipelines at training and test time, respectively. Throughout our analysis, we demonstrate techniques to alleviate such issues, through strategies like partial negative mining and reduced input resolution. We show how this design choices can significantly lower computational overhead with minimal—or even positive—effects on accuracy. These findings provide insight on direction for future research towards more scalable and robust VPR systems. In particular, the issue of training time scalability will be further deepened in Chapter 6 later in the thesis.

3.2 Related Work

Representation learning for visual retrieval and localization. Visual Geo-localization (VG), Visual Localization (VL), and Landmark Retrieval (LR) represent three established computer vision tasks aimed at associating images with spatial coordinates, though each differs in specific objectives. VG seeks to determine the approximate geographic position of a query image, considering predictions valid if they lie within a reasonable proximity to the true location [10, 134, 289, 158, 156, 96, 33, 287]. VL, in contrast, targets the precise estimation of a query image's 6 DoF camera pose within a known environment. While VG techniques can be integrated into VL pipelines alongside additional refinement stages, evaluations focused on VL [228, 266, 211] may not accurately reflect VG performance, motivating the need for dedicated benchmarks. LR constitutes a specialized form of Image Retrieval (IR), where queries depict landmarks and the objective is to retrieve all database images showing the same landmark, irrespective of visual overlap. As VG is commonly framed as a retrieval problem—predicting query coordinates via the GPS tags of top-retrieved matches—numerous methods initially developed for LR (or general IR) have been adapted for VG applications. LR datasets, whether city-scale (Oxford and Paris Buildings [206, 205]) or global (Google Landmarks [191, 293]), feature discrete landmark collections, whereas VG datasets typically span continuous geographic regions.

Traditional IR [251, 60, 9] relies on nearest-neighbor search using fixed-dimensional image representations [59, 232, 203, 121, 124, 265, 122], derived from aggregating discriminative local [161, 23, 9] or global [193, 305] features. Convolutional neural networks (CNNs) now dominate feature extraction for IR, employing diverse concatenation [16, 217] or pooling techniques [13, 14, 264] to generate image descriptors. NetVLAD [10], a differentiable adaptation of VLAD [124] trained end-to-end with a CNN backbone for place recognition, has emerged as a particularly effective deep learning approach for VG. This layer has been widely adopted in subsequent works [26, 87, 92, 96, 105, 158, 288, 289]. A limitation of NetVLAD is its high-dimensional descriptors, which impose significant memory demands in VG systems. This challenge has spurred investigations into more compact representations, either through dimensionality reduction [16, 215, 98, 319, 43] or alternative pooling layers like GeM [216] and R-MAC [99]. Attention mechanisms have also been utilized to prioritize salient scene regions during feature extraction and aggregation for geo-localization [134, 175, 156, 43]. The Contextual Reweighting Network (CRN) [134], for instance, enhances NetVLAD by incorporating contextual modulation to generate a weighting mask based on semi-global context. Visual Transformers leveraging self-attention, such as ViT [69] and DeiT [268], have seen applications in IR [44, 192] but remain unexplored in VG. These VG representation-learning architectures are typically trained with metric embedding objectives common in learning-to-rank, including contrastive loss [194, 215, 216], triplet loss [10, 99, 134], and SARE loss

[158].

This chapter examines how the interplay of widely-used backbone networks, pooling methods, data augmentation strategies, and implementation choices influences geo-localization accuracy, computational efficiency, and memory usage.

Benchmarking. VPR-Bench [306] stands as the sole existing benchmark tailored specifically to VG/VPR. Unlike our approach, [306] (and [211] for VL) evaluates pre-trained models directly, prioritizing real-world applicability where fine-tuning may be impractical. In contrast, this chapter emphasizes quantifying the effects of algorithmic modifications, necessitating controlled comparisons where variables are isolated. To facilitate this, we present a modular framework enabling equitable assessment of each VG system component under consistent conditions, ensuring interpretable and reproducible findings.

Although [306] discusses descriptor dimensionality and retrieval speed, our analysis centers on hardware-agnostic metrics like FLOPs, model size (Sec. 3.4.1), training complexity (Sec. 3.4.4), and storage demands (Sec. 3.4.6).

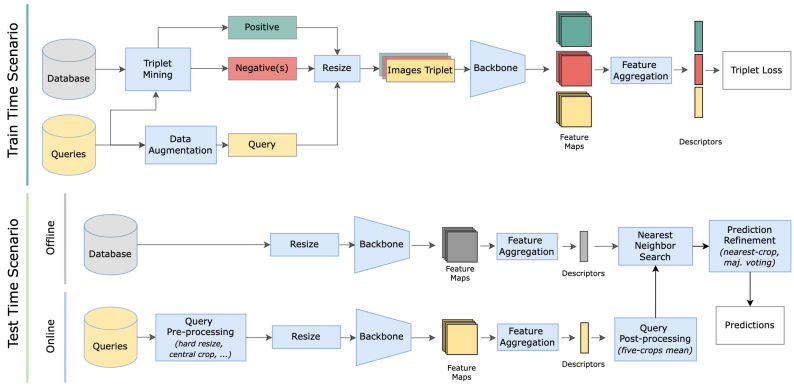


Figure 3.1: **Diagram of a visual geo-localization system.** Throughout this work, we rigorously and fairly analyze each component of a visual geo-localization system (the light blue blocks) comparing a variety of different implementations, both for train and test time.

3.3 Methodology

This chapter outlines the Visual Geo-localization (VG) pipeline employed in the benchmark framework (*cf* Fig. 3.1) along with the experimental setup adopted for evaluation. The objective is to define a consistent and formally sound evaluation protocol that enables systematic comparisons across VG systems. Additionally, this section introduces the datasets used throughout the experiments, chosen to reflect diverse and realistic visual geo-localization scenarios.

3.3.1 Visual Geo-localization System

The VG task is typically addressed via an image retrieval-based approach: given an input image (*query*) whose location is unknown, the system estimates its position by identifying the most similar images from a database of geo-referenced photos. A VG system, therefore, consists of two core stages: descriptor extraction—performed offline for the database and online for the query—and similarity search in the resulting feature space using nearest neighbors techniques.

As illustrated by the orange components in Fig. 3.1, constructing a VG system involves a number of critical design decisions, including the selection of neural architectures, strategies for negative mining, as well as various implementation details such as input image resolution and augmentation policies. Each of these factors influences the system's effectiveness and computational efficiency.

This chapter introduces a benchmark designed to systematically examine how different components impact the performance of VG systems. The modular architecture in Fig. 3.1 serves as a foundation for implementing and evaluating a wide array of methods, primarily those based on CNN backbones, as well as models built upon Visual Transformers.

The proposed abstract pipeline comprises multiple elements that can be independently modified during both training and inference phases: the backbone (Sec. 3.4.1), feature aggregation (Sec. 3.4.2), training sample mining (Sec. 3.4.4), image resizing (Sec. 3.4.6), and data augmentation (Sec. 3.4.5). A dedicated set of experiments investigates each of these components in isolation to assess their individual contributions.

Due to space constraints, this chapter includes only a subset of the results, while additional findings—including experiments on pre/post-processing, impact of pre-training, and other relevant dimensions—are provided in the Supplementary Material.

The benchmark codebase implements the modular structure depicted in Fig. 3.1, allowing easy customization of each pipeline component. It includes utilities for dataset downloading and formatting, and for conducting training and evaluation with minimal friction. This infrastructure supports extensive experimentation while maintaining consistency and reproducibility.

The framework is capable of reproducing many prominent VG architectures [10, 134, 216, 264, 221, 99, 158, 289] and widely adopted training protocols [10, 289, 158]. Further implementation details are provided in the Supplementary Material.

3.3.2 Datasets

The benchmark utilizes six diverse datasets (Tab. 3.2 and maps in the Supplementary Material), selected to represent a wide spectrum of real-world environments. These datasets vary in terms of geographic coverage, image variability, and capture devices.

For training, the Pitts30k dataset [10] and Mapillary Street-Level Sequences (MSLS) [289] are used, representing relatively small and large training sets, respectively. Pitts30k is characterized by consistent conditions—uniform resolution, weather, and camera—while MSLS encompasses multiple cities and a variety of environmental conditions.

Due to the lack of annotated test data in MSLS, we adopt the approach of [105] and evaluate model performance on the validation split. To analyze generalization and cross-dataset performance, four additional datasets are used for testing: Tokyo 24/7 [265], Revisited San Francisco (R-SF) [48, 152, 266], Eynsham [60], and St Lucia [172]. Supplementary Material contains more information about their characteristics and geographical scope.

3.3.3 Benchmark Protocol

Unless explicitly stated otherwise, all experiments employ the recall@N (R@N) metric, which computes the percentage of queries whose top-N retrieved results include an image taken within a certain distance from the query's actual location.

	# train/val datab./queries	# test datab./queries	Dataset size	Database type	Database img. size	Queries type
Pitts30k	20K / 15K	10K / 6.8K	2.0 GB	panorama	480×640	panorama
MSLS	934K / 514K	19K / 11K	56 GB	front-view	480×640	front-view
Tokyo 24/7	0 / 0	75K / 315	4.0 GB	panorama	480×640	phone*
R-SF	0 / 0	1.05M / 598	36 GB	panorama	480×640	phone*
Eynsham	0 / 0	24K / 24K	1.2 GB	panorama	512×384	panorama
St Lucia	0 / 0	1.5K / 1.5K	124 MB	front-view	480×640	front-view

Table 3.2: **Summary of the datasets:** "panorama" means images are cropped from a 360° panorama (including undistortion); "front-view" means that only one (forward facing) view is available; "phone" means photos were collected with a smartphone. "panorama" and "front-view" images were taken with car-rooftop cameras. * Variable resolution.

R@1 is the primary focus, and, in accordance with established practice [10, 134, 158, 200, 201, 26, 33, 289, 105], the standard threshold of 25 meters is adopted. The Supplementary Material includes analyses with varying thresholds and N values.

To ensure robust evaluation, each experiment is repeated three times, with average results reported in the main text. Standard deviations and additional experiment details are available in the Supplementary Material.

Training proceeds until there is no improvement in recall@5 on the validation set for three consecutive epochs. Given the varying sizes of the datasets (Tab. 3.2), an epoch is defined as 5,000 query samples. The Adam optimizer [136] is used, as it generally provides faster convergence and improved performance compared to alternatives such as SGD.

In line with the training scheme outlined in [10], we employ a batch size of 4 triplets per iteration. Each triplet includes a query (anchor), a positive sample, and ten negatives. Following conventional strategies [10, 289, 105, 287, 26, 158], positive images are selected as the closest features within a 10-meter radius of the query, while negatives are drawn from images located beyond 25 meters.

Depending on the dataset size, full database mining is applied for Pitts30k, whereas MSLS uses partial mining strategies. Details regarding the mining procedures are discussed in Section 3.4.4.

3.4 Results

In this section, we analyze the impact of each component in our framework (Fig. 3.1) on overall performance. We begin by evaluating architectural choices, focusing on backbones (Sec. 3.4.1), aggregation methods (Sec. 3.4.2), and Transformer-based networks (Sec. 3.4.3). We then turn to training-time components, such as negative mining (Sec. 3.4.4) and data augmentation (Sec. 3.4.5), before analyzing the effect of image resolution (Sec. 3.4.6) and the role of efficient nearest neighbor search (Sec. 3.4.7). We also report additional results, such as extended experiments, and more detailed metric analyses.

Backbone	Aggregation Method	Features Dim	FLOPs	Model Size	Training Dataset	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18 *conv4_x*	GeM	256	17.29 GF	10.63 MB	Pitts30k	77.8 ± 0.2	35.3 ± 0.5	35.3 ± 1.1	34.2 ± 1.7	64.3 ± 1.2	46.2 ± 0.4
ResNet-18 *conv4_x*	NetVLAD	16384	17.27 GF	10.76 MB	Pitts30k	**86.4 ± 0.3**	**47.4 ± 1.2**	**63.4 ± 1.2**	**61.4 ± 1.5**	**76.8 ± 1.2**	**57.6 ± 3.3**
ResNet-18 *conv5_x*	GeM	512	22.33 GF	42.67 MB	Pitts30k	77.9 ± 0.3	34.4 ± 0.4	34.4 ± 0.6	36.9 ± 0.3	59.1 ± 1.3	51.2 ± 1.3
ResNet-18 *conv5_x*	NetVLAD	32768	22.28 GF	42.92 MB	Pitts30k	79.6 ± 0.5	47.1 ± 1.8	48.9 ± 2.5	49.1 ± 3.6	70.5 ± 1.0	54.4 ± 2.7
ResNet-50 *conv4_x*	GeM	1024	40.61 GF	32.71 MB	Pitts30k	82.0 ± 0.3	38.0 ± 0.1	41.5 ± 1.8	45.4 ± 2.0	66.3 ± 2.5	59.0 ± 1.4
ResNet-50 *conv4_x*	NetVLAD	65536	40.51 GF	33.21 MB	Pitts30k	**86.0 ± 0.1**	**50.7 ± 2.0**	**69.8 ± 0.8**	**67.1 ± 2.3**	**77.7 ± 0.4**	**60.2 ± 1.6**
ResNet-50 *conv5_x*	GeM	2048	50.54 GF	89.88 MB	Pitts30k	79.8 ± 0.5	41.5 ± 0.7	48.0 ± 2.5	44.3 ± 1.0	65.2 ± 1.4	57.5 ± 1.5
ResNet-50 *conv5_x*	NetVLAD	131072	50.35 GF	90.88 MB	Pitts30k	79.6 ± 0.2	46.2 ± 0.5	54.7 ± 2.6	51.2 ± 2.5	69.8 ± 1.0	53.0 ± 4.1
ResNet-18 *conv4_x*	GeM	256	17.29 GF	10.63 MB	MSLS	71.6 ± 0.1	65.3 ± 0.2	42.8 ± 1.1	30.5 ± 0.8	80.3 ± 0.1	83.2 ± 0.9
ResNet-18 *conv4_x*	NetVLAD	16384	17.27 GF	10.76 MB	MSLS	**81.6 ± 0.5**	**75.8 ± 0.1**	**62.3 ± 1.6**	**55.1 ± 0.9**	**87.1 ± 0.2**	**92.1 ± 0.7**
ResNet-18 *conv5_x*	GeM	512	22.33 GF	42.67 MB	MSLS	73.5 ± 0.5	68.4 ± 0.8	41.0 ± 0.8	38.6 ± 1.8	79.4 ± 0.5	84.7 ± 0.7
ResNet-18 *conv5_x*	NetVLAD	32768	22.28 GF	42.92 MB	MSLS	75.7 ± 0.7	75.7 ± 0.6	49.9 ± 1.6	41.3 ± 0.2	84.1 ± 0.4	91.3 ± 0.4
ResNet-50 *conv4_x*	GeM	1024	40.61 GF	32.71 MB	MSLS	77.4 ± 0.6	72.0 ± 0.5	55.4 ± 2.5	45.7 ± 1.0	83.9 ± 0.6	91.2 ± 0.7
ResNet-50 *conv4_x*	NetVLAD	65536	40.51 GF	33.21 MB	MSLS	**80.9 ± 0.0**	**76.9 ± 0.2**	**62.8 ± 0.9**	**51.5 ± 1.2**	**87.2 ± 0.3**	**93.8 ± 0.2**
ResNet-50 *conv5_x*	GeM	2048	50.54 GF	89.88 MB	MSLS	74.7 ± 0.4	70.6 ± 0.6	46.3 ± 1.3	42.1 ± 0.5	82.5 ± 0.5	89.8 ± 0.4
ResNet-50 *conv5_x*	NetVLAD	131072	50.35 GF	90.88 MB	MSLS	74.7 ± 0.2	75.2 ± 0.5	52.4 ± 0.8	44.0 ± 1.1	85.5 ± 0.4	91.3 ± 0.7

Table 3.3: **ResNets:** The advantages of cropping the ResNets at *conv4_x* for visual geo-localization.

3.4.1 CNN Backbones

The CNN backbone plays a central role in extracting informative visual features. We assess four standard architectures—VGG16 [245], ResNet-18, ResNet-50, and ResNet-101 [107]—combined with two aggregation methods: GeM [216] and NetVLAD [10]. Despite the limited set, these cover a broad range of widely used VG and retrieval models [10, 134, 264, 221].

Backbone	Aggregation Method	Features Dim	FLOPs (GF)	Model Size (MB)	Extraction Time (ms)	Training on Pitts30k						Training on MSLS					
						R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
VGG-16	GeM	512	188.01	56.13	12.3	78.5	43.4	39.9	40.4	70.2	46.4	70.2	66.7	43.6	32.1	80.4	79.9
ResNet-18	GeM	256	17.29	10.63	4.1	77.8	35.3	35.3	34.2	64.3	46.2	71.6	65.3	42.8	30.5	80.3	83.2
ResNet-50	GeM	1024	40.61	32.71	6.7	82.0	38.0	41.5	45.4	66.3	**59.0**	**77.4**	72.0	**55.4**	45.7	**83.9**	91.2
ResNet-101	GeM	1024	86.29	105.36	9.6	**82.4**	**39.6**	**44.0**	**52.5**	**69.0**	57.6	77.2	**72.5**	51.0	**46.9**	83.6	**91.6**
VGG-16	NetVLAD	32768	188.09	56.38	13.0	83.2	50.9	61.4	64.6	74.4	50.1	79.0	74.6	61.9	57.1	84.2	86.7
ResNet-18	NetVLAD	16384	17.27	10.76	4.4	86.4	47.4	63.4	61.4	76.8	57.6	**81.6**	75.8	62.3	55.1	87.1	92.1
ResNet-50	NetVLAD	65536	40.51	33.21	8.5	86.0	50.7	69.8	67.1	**77.7**	60.2	80.9	76.9	**62.8**	51.5	**87.2**	93.8
ResNet-101	NetVLAD	65536	86.06	105.86	11.5	**86.5**	**51.8**	**72.2**	**67.5**	74.0	**63.6**	80.8	**77.7**	59.0	**56.1**	86.7	**95.1**

Table 3.4: Results and computational requirements with different convolutional **backbones**. Extraction time is the average over a 1000 forward passes.

For ResNet backbones, whose architecture is made up of identical blocks with depth scaling, it is not straightforward to choose whether or not to truncate the backbone. It is well known that deeper blocks encode progressively more semantic representation, loosing spatial details. Given that spatial structure can play an important role in VPR, we conduct a preliminary study in Tab. 3.3 to assess the effect of truncating the backbone. As the channel dimensionality increases with depth, this choice also impacts the computational and storage cost. From this analysis, we find that for ResNet-like backbones, the feature maps from `conv4_x` layer in general yield a better trade-off between recall and efficiency compared to using `conv5_x`. For VGG16, we keep the original structure and remove only the final pooling layer. Results are reported in Tab. 3.4

Discussion. Deeper ResNets (e.g., ResNet-50/101) outperform shallower variants. ResNet-50 achieves similar recall to ResNet-101 with substantially fewer FLOPs and smaller model size, offering a better efficiency-accuracy tradeoff. While ResNet-18 underperforms, it offers unmatched efficiency, making it ideal for resource-constrained scenarios. Training data significantly affects performance: the same model trained on Pitts30k vs. MSLS yields up to 30% recall@1 differences on the St. Lucia test set. This highlights how comparing methods trained on different datasets, as in [306], can be misleading.

3.4.2 Aggregation and Descriptor Dimensionality

Aggregation layers convert spatial feature maps into compact descriptors. We evaluate several aggregation methods, including SPOC [14], MAC [218], R-MAC [264], RRM [138], GeM [216], NetVLAD [10], and CRN [134]. We first conductive a thorough screening of all these methods in Tab. 3.5. When large differences exist, in terms of descriptor dimensionality, we resort to PCA or FC layers to obtain fair comparisons. From the table, it is apparent that the learnable GeM pooling outperforms its non-parametric counterparts, and that NetVLAD-based aggregations in general perform better. Hence, in Tab. 3.6 we focus on the top-performing methods, namely GeM, NetVLAD, and CRN.

Discussion. Performance depends strongly on the training dataset. On the small-scale Pitts30k, CRN delivers the best results, even after dimensionality reduction. On the larger MSLS, however, GeM surpasses both CRN and NetVLAD, especially on datasets with heterogeneous image types (e.g., Tokyo, R-SF), likely due to GeM's robustness to modality shifts. PCA severely degrades performance for NetVLAD and CRN, whereas GeM paired with a learned FC layer maintains performance. Despite CRN's robustness, it has practical downsides: a two-stage training procedure, additional hyperparameters, and potential convergence issues depending on initialization.

Backbone	Aggregation Method	Features Dim	Training Dataset	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18	SPOC [14]	256	Pitts30k	60.6 ± 0.9	16.5 ± 0.5	15.2 ± 1.1	10.4 ± 0.3	41.0 ± 2.0	29.0 ± 1.5
ResNet-18	MAC [218]	256	Pitts30k	57.3 ± 0.5	25.6 ± 0.4	15.2 ± 1.3	15.5 ± 0.3	49.6 ± 0.7	26.6 ± 1.0
ResNet-18	RMAC [264]	256	Pitts30k	63.2 ± 0.4	28.7 ± 0.6	22.7 ± 2.3	30.5 ± 1.4	64.0 ± 0.7	42.8 ± 1.3
ResNet-18	RRM [138]	256	Pitts30k	68.2 ± 0.5	21.4 ± 0.8	25.4 ± 1.4	21.7 ± 1.8	51.9 ± 0.8	33.7 ± 0.3
ResNet-18	GeM [216]	256	Pitts30k	77.8 ± 0.2	35.3 ± 0.5	35.3 ± 1.1	34.2 ± 1.7	64.3 ± 1.2	46.2 ± 0.4
ResNet-18	GeM + FC 256	256	Pitts30k	72.4 ± 0.7	26.4 ± 0.5	27.5 ± 1.2	29.0 ± 1.2	59.3 ± 1.0	39.1 ± 0.8
ResNet-18	NetVLAD + PCA 256	256	Pitts30k	80.7 ± 0.7	38.3 ± 1.2	41.7 ± 0.8	35.9 ± 1.8	68.9 ± 1.1	45.4 ± 2.2
ResNet-18	CRN + PCA 256	256	Pitts30k	**82.0 ± 0.7**	**43.6 ± 0.7**	**47.7 ± 0.9**	**45.1 ± 0.3**	**71.3 ± 0.8**	**51.3 ± 3.4**
ResNet-18	GeM + FC 2048	2048	Pitts30k	75.0 ± 0.4	29.9 ± 0.6	34.5 ± 0.4	36.1 ± 0.2	63.7 ± 0.3	45.1 ± 2.1
ResNet-18	NetVLAD + PCA 2048	2048	Pitts30k	85.0 ± 0.4	45.0 ± 1.5	56.6 ± 0.7	53.2 ± 2.4	75.4 ± 1.1	54.6 ± 3.0
ResNet-18	CRN + PCA 2048	2048	Pitts30k	**85.7 ± 0.3**	**50.6 ± 0.6**	**61.0 ± 1.6**	**62.8 ± 1.2**	**77.4 ± 0.5**	**61.1 ± 2.7**
ResNet-18	NetVLAD [10]	16384	Pitts30k	86.4 ± 0.3	47.4 ± 1.2	63.4 ± 1.2	61.4 ± 1.5	76.8 ± 1.2	57.6 ± 3.3
ResNet-18	CRN [134]	16384	Pitts30k	**86.8 ± 0.1**	**53.2 ± 0.7**	**68.8 ± 1.0**	**69.0 ± 0.6**	**79.1 ± 0.3**	**64.8 ± 3.2**
ResNet-50	SPOC [14]	1024	Pitts30k	60.9 ± 0.5	19.2 ± 0.4	14.0 ± 0.5	9.0 ± 0.7	40.5 ± 2.3	27.1 ± 1.5
ResNet-50	MAC [218]	1024	Pitts30k	77.6 ± 0.2	36.2 ± 0.7	36.2 ± 1.4	34.8 ± 0.7	72.9 ± 0.3	51.3 ± 2.4
ResNet-50	RMAC [264]	1024	Pitts30k	74.9 ± 1.0	34.8 ± 0.8	41.8 ± 0.6	46.4 ± 1.0	73.1 ± 0.7	**68.7 ± 0.5**
ResNet-50	RRM [138]	1024	Pitts30k	72.8 ± 0.2	27.9 ± 0.6	28.3 ± 0.8	28.6 ± 1.0	65.9 ± 0.9	45.1 ± 1.7
ResNet-50	GeM [216]	1024	Pitts30k	82.0 ± 0.3	38.0 ± 0.1	41.5 ± 1.8	45.4 ± 2.0	66.3 ± 2.5	59.0 ± 1.4
ResNet-50	NetVLAD + PCA 1024	1024	Pitts30k	83.9 ± 0.7	46.5 ± 2.0	59.4 ± 1.2	53.2 ± 3.8	72.5 ± 0.3	57.7 ± 2.0
ResNet-50	CRN + PCA 1024	1024	Pitts30k	**84.1 ± 0.4**	**49.9 ± 0.8**	**64.6 ± 1.2**	**58.8 ± 0.1**	**74.3 ± 0.2**	63.4 ± 0.4
ResNet-50	GeM + FC 2048	2048	Pitts30k	80.1 ± 0.2	33.7 ± 0.3	43.6 ± 1.6	48.2 ± 1.2	70.0 ± 0.3	56.0 ± 1.7
ResNet-50	NetVLAD + PCA 2048	2048	Pitts30k	84.4 ± 0.4	47.9 ± 2.0	62.6 ± 1.7	56.0 ± 2.9	74.1 ± 0.4	58.9 ± 1.6
ResNet-50	CRN + PCA 2048	2048	Pitts30k	**84.7 ± 0.3**	**51.2 ± 0.8**	**67.1 ± 0.7**	**62.3 ± 0.3**	**75.8 ± 0.2**	**65.0 ± 0.1**
ResNet-50	NetVLAD [10]	65536	Pitts30k	**86.0 ± 0.1**	50.7 ± 2.0	69.8 ± 0.8	67.1 ± 2.3	77.7 ± 0.4	60.2 ± 1.6
ResNet-50	CRN [134]	65536	Pitts30k	85.8 ± 0.2	**54.0 ± 0.8**	**73.1 ± 0.3**	**70.9 ± 0.2**	**79.7 ± 0.1**	**65.9 ± 0.4**
ResNet-18	SPOC [14]	256	MSLS	44.2 ± 1.0	39.5 ± 0.5	20.3 ± 1.3	9.5 ± 0.9	62.3 ± 0.6	58.8 ± 0.8
ResNet-18	MAC [218]	256	MSLS	60.4 ± 1.1	54.7 ± 1.8	20.4 ± 2.6	18.9 ± 2.0	76.3 ± 1.2	69.2 ± 1.2
ResNet-18	RMAC [264]	256	MSLS	58.1 ± 1.2	48.9 ± 2.0	29.1 ± 2.0	34.3 ± 1.4	73.3 ± 1.1	63.7 ± 2.7
ResNet-18	RRM [138]	256	MSLS	60.8 ± 1.5	54.9 ± 2.6	**44.4 ± 2.1**	30.9 ± 2.8	75.7 ± 1.5	68.7 ± 1.4
ResNet-18	GeM [216]	256	MSLS	71.6 ± 0.1	65.3 ± 0.2	42.8 ± 1.1	30.5 ± 0.8	80.3 ± 0.1	83.2 ± 0.9
ResNet-18	GeM + FC 256	256	MSLS	68.6 ± 1.1	59.6 ± 2.6	41.9 ± 2.7	31.3 ± 0.5	78.5 ± 2.0	76.1 ± 3.4
ResNet-18	NetVLAD + PCA 256	256	MSLS	74.2 ± 0.2	70.6 ± 0.3	43.6 ± 0.5	34.7 ± 1.7	84.4 ± 0.4	89.8 ± 0.5
ResNet-18	CRN + PCA 256	256	MSLS	**74.5 ± 0.8**	**72.1 ± 0.1**	44.1 ± 1.4	**35.1 ± 2.4**	**84.8 ± 0.3**	**91.6 ± 0.4**
ResNet-18	GeM + FC 2048	2048	MSLS	71.9 ± 1.0	64.0 ± 1.2	51.8 ± 0.9	37.6 ± 1.3	81.1 ± 0.9	79.2 ± 0.9
ResNet-18	NetVLAD + PCA 2048	2048	MSLS	**80.4 ± 0.4**	74.6 ± 0.2	55.6 ± 1.2	47.4 ± 1.1	86.4 ± 0.3	92.2 ± 0.3
ResNet-18	CRN + PCA 2048	2048	MSLS	80.1 ± 0.8	**75.8 ± 0.1**	**57.2 ± 2.3**	**47.8 ± 2.7**	**86.8 ± 0.3**	**93.2 ± 0.4**
ResNet-18	NetVLAD [10]	16384	MSLS	**81.6 ± 0.5**	75.8 ± 0.1	62.3 ± 1.6	**55.1 ± 0.9**	87.1 ± 0.2	92.1 ± 0.7
ResNet-18	CRN [134]	16384	MSLS	81.3 ± 0.7	**76.8 ± 0.0**	**63.8 ± 1.4**	53.9 ± 2.0	**87.5 ± 0.2**	**93.7 ± 0.1**
ResNet-50	SPOC [14]	1024	MSLS	47.5 ± 1.3	47.9 ± 1.5	20.6 ± 1.6	8.9 ± 1.0	68.3 ± 0.5	68.6 ± 1.4
ResNet-50	MAC [218]	1024	MSLS	76.0 ± 0.2	67.4 ± 1.6	45.3 ± 1.0	44.4 ± 2.6	84.6 ± 0.4	86.0 ± 0.7
ResNet-50	RMAC [264]	1024	MSLS	70.1 ± 0.8	62.0 ± 0.5	52.1 ± 2.3	**54.3 ± 1.8**	80.6 ± 0.5	85.9 ± 1.0
ResNet-50	GeM [216]	1024	MSLS	**77.4 ± 0.6**	72.0 ± 0.5	**55.4 ± 2.5**	45.7 ± 1.0	83.9 ± 0.6	91.2 ± 0.7
ResNet-50	NetVLAD + PCA 1024	1024	MSLS	**77.4 ± 0.2**	74.8 ± 0.3	51.3 ± 1.3	39.0 ± 1.3	85.2 ± 0.3	92.9 ± 0.3
ResNet-50	RRM [138]	1024	MSLS	69.3 ± 1.0	67.4 ± 0.4	53.7 ± 0.8	43.7 ± 1.0	84.3 ± 0.5	84.8 ± 1.1
ResNet-50	CRN + PCA 1024	1024	MSLS	77.3 ± 0.3	**75.6 ± 0.0**	51.8 ± 1.1	38.8 ± 1.0	**85.7 ± 0.3**	**94.1 ± 0.2**
ResNet-50	GeM + FC 2048	2048	MSLS	**79.2 ± 0.6**	73.5 ± 0.8	**64.0 ± 3.9**	**55.1 ± 2.4**	86.1 ± 0.7	90.3 ± 1.0
ResNet-50	NetVLAD + PCA 2048	2048	MSLS	78.5 ± 0.2	75.4 ± 0.2	52.8 ± 0.4	42.6 ± 1.3	85.8 ± 0.3	93.4 ± 0.4
ResNet-50	CRN + PCA 2048	2048	MSLS	78.3 ± 0.3	**76.3 ± 0.1**	54.3 ± 0.7	42.8 ± 1.6	**86.2 ± 0.4**	**94.4 ± 0.2**
ResNet-50	NetVLAD [10]	65536	MSLS	**80.9 ± 0.0**	76.9 ± 0.2	62.8 ± 0.9	51.5 ± 1.2	87.2 ± 0.3	93.8 ± 0.2
ResNet-50	CRN [134]	65536	MSLS	80.8 ± 0.2	**77.8 ± 0.1**	**63.6 ± 0.5**	**53.4 ± 1.4**	**87.5 ± 0.4**	**94.8 ± 0.3**

Table 3.5: **Aggregation methods.** Full table of aggregation methods, grouped by backbone and features dimension.

Backbone	Aggregation Method	Features Dim	Training on Pitts30k						Training on MSLS						
			R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia	R@1 Average
ResNet-50	GeM	1024	82.0	38.0	41.5	45.4	66.3	59.0	**77.4**	72.0	**55.4**	**45.7**	83.9	91.2	63.2
ResNet-50	NetVLAD + PCA 1024	1024	83.9	46.5	59.4	53.2	72.5	57.7	**77.4**	74.8	51.3	39.0	85.2	92.9	66.2
ResNet-50	CRN + PCA 1024	1024	**84.1**	**49.9**	**64.6**	**58.8**	**74.3**	**63.4**	77.3	**75.6**	51.8	38.8	**85.7**	**94.1**	**68.2**
ResNet-50	GeM + FC 2048	2048	80.1	33.7	43.6	48.2	70.0	56.0	**79.2**	73.5	**64.0**	**55.1**	86.1	90.3	65.0
ResNet-50	NetVLAD + PCA 2048	2048	84.4	47.9	62.6	56.0	74.1	58.9	78.5	75.4	52.8	42.6	85.8	93.4	67.7
ResNet-50	CRN + PCA 2048	2048	**84.7**	**51.2**	**67.1**	**62.3**	**75.8**	**65.0**	78.3	**76.3**	54.3	42.8	**86.2**	**94.4**	**69.9**
ResNet-50	GeM + FC 65536	65536	80.8	35.8	45.6	49.0	72.5	59.6	79.0	74.4	**69.2**	**58.4**	86.2	90.8	66.8
ResNet-50	NetVLAD	65536	**86.0**	50.7	69.8	67.1	77.7	60.2	**80.9**	76.9	62.8	51.5	87.2	93.8	72.1
ResNet-50	CRN	65536	85.8	**54.0**	**73.1**	**70.9**	**79.7**	**65.9**	80.8	**77.8**	63.6	53.4	**87.5**	**94.8**	**73.9**

Table 3.6: **Aggregation methods:** we report results with different aggregation methods downscaled or upscaled to equivalent dimensionality.

Backbone	Aggreg. Method	Feat. Dim	FLOPs (GF)	Training on MSLS R@1 Pitts30k	R@1 MSLS	R@1 Tok. 24/7	R@1 R-SF	R@1 Eyns.	R@1 St Lucia
ResNet-18	GeM	256	17.29	71.6	65.3	42.8	30.5	80.3	83.2
ResNet-50	GeM	1024	40.61	77.4	72.0	55.4	45.7	83.9	91.2
ViT	CLS	768	82.31	**82.9**	**73.5**	**59.9**	**65.0**	84.5	93.6
CCT	CLS	384	22.34	79.6	71.1	52.0	49.9	85.6	**94.0**
CCT	SeqPool	384	26.19	81.4	71.0	59.1	60.5	**86.1**	92.4
CCT	GeM	384	22.36	78.7	72.0	48.8	48.6	83.9	92.9
ResNet-18	NetVLAD	16384	17.27	81.6	75.8	62.3	55.1	87.1	92.1
ResNet-50	NetVLAD	65536	40.51	80.9	76.9	62.3	51.5	87.2	93.8
CCT	NetVLAD	24576	18.53	**85.1**	**79.9**	**70.3**	**65.9**	**87.4**	**98.4**

Table 3.7: **Transformers** Comparison of traditional CNN architectures with novel Transformers-based approaches.

Backbone	Aggregation Method	Features Dim	FLOPs [GF]	Model Size [MB]	Training Dataset	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18	GeM	256	17.29	10.63	Pitts30k	77.8 ± 0.2	35.3 ± 0.5	35.3 ± 1.1	34.2 ± 1.7	64.3 ± 1.2	46.2 ± 0.4
ResNet-50	GeM	1024	40.61	32.71	Pitts30k	**82.0 ± 0.3**	38.0 ± 0.1	41.5 ± 1.8	45.4 ± 2.0	66.3 ± 2.5	59.0 ± 1.4
ViT	CLS	768	82.31	350.96	Pitts30k	79.2 ± 1.5	39.0 ± 0.8	44.5 ± 3.2	48.3 ± 2.5	67.6 ± 1.2	**69.6 ± 2.0**
CCT	CLS	384	22.34	190.39	Pitts30k	76.3 ± 1.4	39.5 ± 0.4	39.0 ± 1.7	44.4 ± 0.4	50.8 ± 2.1	57.3 ± 2.6
CCT	SeqPool	384	26.19	221.92	Pitts30k	81.1 ± 1.0	46.9 ± 1.2	51.5 ± 0.8	57.8 ± 1.5	**75.2 ± 1.1**	63.6 ± 2.6
CCT	GeM	384	22.36	191.24	Pitts30k	79.6 ± 0.3	**47.8 ± 0.7**	**52.3 ± 2.0**	**61.3 ± 0.1**	71.0 ± 0.8	59.1 ± 2.0
ResNet-18	NetVLAD	16384	17.27	10.76	Pitts30k	**86.4 ± 0.3**	47.4 ± 1.2	63.4 ± 1.2	61.4 ± 1.5	76.8 ± 1.2	57.6 ± 3.3
ResNet-50	NetVLAD	65536	40.51	33.21	Pitts30k	86.0 ± 0.1	50.7 ± 2.0	**69.8 ± 0.8**	67.1 ± 2.3	**77.7 ± 0.4**	**60.2 ± 1.6**
CCT	NetVLAD	24576	18.53	160.08	Pitts30k	84.6 ± 0.3	**52.5 ± 1.9**	69.1 ± 0.4	**73.5 ± 1.4**	72.6 ± 0.6	56.1 ± 3.3
ResNet-18	GeM	256	17.29	10.63	MSLS	71.6 ± 0.1	65.3 ± 0.2	42.8 ± 1.1	30.5 ± 0.8	80.3 ± 0.1	83.2 ± 0.9
ResNet-50	GeM	1024	40.61	32.71	MSLS	77.4 ± 0.6	72.0 ± 0.5	55.4 ± 2.5	45.7 ± 1.0	83.9 ± 0.6	91.2 ± 0.7
ViT	CLS	768	82.31	350.96	MSLS	**82.9 ± 0.6**	**73.5 ± 0.6**	**59.9 ± 4.4**	**65.0 ± 1.1**	84.5 ± 1.0	93.6 ± 0.7
CCT	CLS	384	22.34	190.39	MSLS	79.6 ± 0.3	71.1 ± 0.4	52.0 ± 1.1	49.9 ± 1.8	85.6 ± 0.1	**94.0 ± 0.3**
CCT	SeqPool	384	26.19	221.92	MSLS	81.4 ± 0.8	71.0 ± 0.9	59.1 ± 3.2	60.5 ± 1.5	**86.1 ± 0.6**	92.4 ± 1.1
CCT	GeM	384	22.36	191.24	MSLS	78.7 ± 0.6	72.0 ± 0.6	48.8 ± 1.2	48.6 ± 2.9	83.9 ± 0.1	92.9 ± 0.7
ResNet-18	NetVLAD	16384	17.27	10.76	MSLS	81.6 ± 0.5	75.8 ± 0.1	62.3 ± 1.6	55.1 ± 0.9	87.1 ± 0.2	92.1 ± 0.7
ResNet-50	NetVLAD	65536	40.51	33.21	MSLS	80.9 ± 0.0	76.9 ± 0.2	62.8 ± 0.9	51.5 ± 1.2	87.2 ± 0.3	93.8 ± 0.2
CCT	NetVLAD	24576	18.53	160.08	MSLS	**85.1 ± 0.2**	**79.9 ± 0.3**	**70.3 ± 2.0**	**65.9 ± 1.3**	**87.4 ± 0.2**	**98.4 ± 0.2**

Table 3.8: **Transformers** Comparison of traditional CNN architectures with novel Transformers-based approaches.

3.4.3 Visual Transformers

We examine two Transformer-based architectures: ViT [69], which processes image patches as sequences, and CCT [103], which introduces CNN-style inductive bias. For ViT, we use the `CLS` token as a global descriptor, as it was proposed in the

work of [192], which uses ViT for image retrieval. For CCT, we evaluate its native aggregation SeqPool [103]. Further, we adapt the popular GeM and NetVLAD aggregations, which were designed for CNNs, to work with tokens as input rather than feature maps.

Discussion. Reading from Tab. 3.7, it is clear that Transformer-based backbones perform competitively with CNNs, even without additional aggregation by directly relying on the global CLS token, demonstrating strong generalization. Combining them with aggregators like GeM or NetVLAD further boosts performance. Notably, ViT rivals NetVLAD with much smaller descriptors but higher compute. CCT, despite its low cost (comparable to ResNet-18), consistently outperforms it, and often exceeds ResNet-50. SeqPool improves robustness, and NetVLAD+CCT achieves state-of-the-art results. However, Transformers require per-case tuning (e.g., where to truncate/freeze layers), unlike CNNs, which work reliably up to `conv4`. Upon further investigation in Tab. 3.8, it can be seen that when trained on the smaller scale of Pitts30k, using the CLS token compares unfavorably to using learnable aggregations. Overall, the main takeaway from this analysis is that the token based representation of Transformer architectures is a viable alterative to traditional backbones for the future, as the availability of training data is expected to grow. This can be intuitively explained with the fact that Transformers do not contain the inductive bias of CNNs, which makes the latter perform better for the task when trained on small datasets.

Backbone	Aggregation Method	Mining Method	Space & Time Complexity	Training on Pitts30k						Training on MSLS					
				R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18	GeM	Random	$\mathcal{O}(1)$	73.7	30.5	31.3	24.0	58.2	41.0	62.2	50.6	28.8	17.1	70.2	71.4
ResNet-18	GeM	Full database	$\mathcal{O}(\#db + \#q)$	**77.8**	**35.3**	**35.3**	**34.2**	**64.3**	**46.2**	70.1	61.8	**42.8**	**31.3**	79.3	81.0
ResNet-18	GeM	Partial database	$\mathcal{O}(k_{db} + k_q + \#pos)$	76.5	34.2	33.9	32.9	64.0	45.6	**71.6**	**65.3**	**42.8**	30.5	**80.3**	**83.2**
ResNet-18	NetVLAD	Random	$\mathcal{O}(1)$	83.9	43.6	55.1	53.8	76.3	53.5	73.3	61.5	45.0	34.8	84.9	79.7
ResNet-18	NetVLAD	Full database	$\mathcal{O}(\#db + \#q)$	**86.4**	**47.4**	**63.4**	61.4	**76.8**	**57.6**	-	-	-	-	-	-
ResNet-18	NetVLAD	Partial database	$\mathcal{O}(k_{db} + k_q + \#pos)$	86.2	47.3	61.2	**62.9**	76.6	57.1	**81.6**	**75.8**	**62.3**	**55.1**	**87.1**	**92.1**

Table 3.9: **Negative mining methods.** "Space & Time Complexity" refers to the complexity of building the cache, which normally is done after iterating over 1000 triplets [10, 289]. $\#db$ and $\#q$ are the numbers of database and query images, k_{db} and k_q are chosen constants (usually set to 1000), and $\#pos$ is the number of positives for the considered queries, which depends on the queries and database density.

3.4.4 Negative Mining

We compare three mining strategies: full database mining [10], partial mining [289], and random sampling. The choice of the mining protocol is fundamental in a VG system. Its main purpose is to select hard negative examples, *i.e.* images of places that are geographically far apart, but that share visual similarities. If the negative samples are too *easy*, essential this would give little to no supervision

Backbone	Aggregation Method	Mining Method	Training Dataset	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18	GeM	Random	Pitts30k	73.7 ± 0.7	30.5 ± 0.5	31.3 ± 0.8	24.0 ± 1.2	58.2 ± 1.4	41.0 ± 1.2
ResNet-18	GeM	Full database mining	Pitts30k	**77.8 ± 0.2**	**35.3 ± 0.5**	**35.3 ± 1.1**	**34.2 ± 1.7**	**64.3 ± 1.2**	**46.2 ± 0.4**
ResNet-18	GeM	Partial database mining	Pitts30k	76.5 ± 0.3	34.2 ± 1.3	33.9 ± 1.4	32.9 ± 0.7	64.0 ± 2.4	45.6 ± 0.9
ResNet-18	NetVLAD	Random	Pitts30k	83.9 ± 0.5	43.6 ± 0.5	55.1 ± 1.3	53.8 ± 1.1	76.3 ± 0.6	53.5 ± 1.4
ResNet-18	NetVLAD	Full database mining	Pitts30k	**86.4 ± 0.3**	**47.4 ± 1.2**	**63.4 ± 1.2**	61.4 ± 1.5	**76.8 ± 1.2**	**57.6 ± 3.3**
ResNet-18	NetVLAD	Partial database mining	Pitts30k	86.2 ± 0.3	47.3 ± 0.4	61.2 ± 0.5	**62.9 ± 0.3**	76.6 ± 0.5	57.1 ± 1.6
ResNet-50	GeM	Random	Pitts30k	77.9 ± 1.0	34.3 ± 1.3	40.1 ± 1.0	35.5 ± 3.0	63.8 ± 0.9	52.3 ± 1.4
ResNet-50	GeM	Full database mining	Pitts30k	82.0 ± 0.3	38.0 ± 0.1	41.5 ± 1.8	45.4 ± 2.0	66.3 ± 2.5	59.0 ± 1.4
ResNet-50	GeM	Partial database mining	Pitts30k	**82.3 ± 0.0**	**39.0 ± 0.4**	**43.5 ± 0.2**	**45.5 ± 1.7**	**67.7 ± 1.4**	**61.0 ± 2.0**
ResNet-50	NetVLAD	Random	Pitts30k	83.4 ± 0.6	45.0 ± 0.3	61.9 ± 2.1	55.8 ± 1.5	75.0 ± 1.8	52.6 ± 1.2
ResNet-50	NetVLAD	Full database mining	Pitts30k	**86.0 ± 0.1**	**50.7 ± 2.0**	**69.8 ± 0.8**	**67.1 ± 2.3**	**77.7 ± 0.4**	**60.2 ± 1.6**
ResNet-50	NetVLAD	Partial database mining	Pitts30k	85.5 ± 0.3	48.6 ± 3.1	66.7 ± 4.1	65.0 ± 4.3	77.6 ± 1.3	59.0 ± 4.1
ResNet-18	GeM	Random	MSLS	62.2 ± 0.3	50.6 ± 0.6	28.8 ± 0.8	17.1 ± 1.0	70.2 ± 0.6	71.4 ± 1.0
ResNet-18	GeM	Full database mining	MSLS	70.1 ± 1.1	61.8 ± 0.5	**42.8 ± 1.4**	**31.3 ± 1.2**	79.3 ± 0.2	81.0 ± 0.9
ResNet-18	GeM	Partial database mining	MSLS	**71.6 ± 0.1**	**65.3 ± 0.2**	**42.8 ± 1.1**	30.5 ± 0.8	**80.3 ± 0.1**	**83.2 ± 0.9**
ResNet-18	NetVLAD	Random	MSLS	73.3 ± 0.7	61.5 ± 1.4	45.0 ± 1.5	34.8 ± 0.2	84.9 ± 0.3	79.7 ± 1.7
ResNet-18	NetVLAD	Full database mining	MSLS	-	-	-	-	-	-
ResNet-18	NetVLAD	Partial database mining	MSLS	**81.6 ± 0.5**	**75.8 ± 0.1**	**62.3 ± 1.6**	**55.1 ± 0.9**	**87.1 ± 0.2**	**92.1 ± 0.7**
ResNet-50	GeM	Random	MSLS	69.5 ± 1.2	57.4 ± 1.1	43.5 ± 3.3	31.1 ± 0.9	78.8 ± 0.5	78.3 ± 1.2
ResNet-50	GeM	Full database mining	MSLS	77.3 ± 0.3	69.7 ± 0.2	52.4 ± 1.7	45.3 ± 0.2	**84.2 ± 0.0**	91.0 ± 0.2
ResNet-50	GeM	Partial database mining	MSLS	**77.4 ± 0.6**	**72.0 ± 0.5**	**55.4 ± 2.5**	**45.7 ± 1.0**	83.9 ± 0.6	**91.2 ± 0.7**
ResNet-50	NetVLAD	Random	MSLS	74.9 ± 0.4	63.6 ± 1.3	41.9 ± 1.6	34.6 ± 2.3	85.5 ± 0.2	80.9 ± 0.4
ResNet-50	NetVLAD	Full database mining	MSLS	-	-	-	-	-	-
ResNet-50	NetVLAD	Partial database mining	MSLS	**80.9 ± 0.0**	**76.9 ± 0.2**	**62.8 ± 0.9**	**51.5 ± 1.2**	**87.2 ± 0.3**	**93.8 ± 0.2**

Table 3.10: **Mining methods.**

signal for the model to learn. The authors of NetVLAD [10] relied on full mining, that requires every N iterations to compute features for all the database images in order to choose suitable negatives. Such strategy, although effective, hardly scales with the database size. Hence, in MSLS [289], which has a database of more than 1 million images, the authors find this technique to be impractical and resort to partial mining. The idea of partial mining is that every N training iterations, with N being typically set to 1000, it is sufficient to sample only a small subset of database images among which negatives should be chosen. This allows to keep the cost of mining fixed w.r.t. the database of choice. We report in Tab. 3.9 the main results.

Discussion. As expected, random sampling underperforms—5% lower on Pitts30k and >10% on MSLS. Full mining performs best but offers only marginal gains (1%) over partial mining, which is significantly more scalable. On large datasets, full mining becomes impractical, making partial mining a cost-effective and robust choice. In Tab. 3.10 we deepen our analysis. The results show how the choice of mining strategy is tied to the characteristics of the dataset. On the small set of Pitts30k, even random sampling provides good performances while being cost-free. On MSLS, which contains a much higher variety, the performance drop is higher. Lastly, while full mining is still feasible on Pitts30k, and performs comparably to partial mining, results on MSLS clearly indicate how this choice scales poorly. For methods with high dimensionality (the NetVLAD-based aggregations), full mining is unfeasible not only because of prolonged training times, but also because it

would require to store in memory all the database descriptors, which resulted in intractable memory requirements.

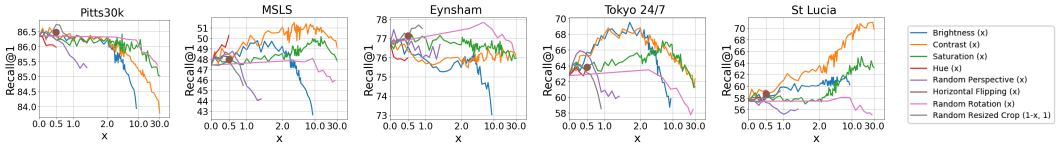


Figure 3.2: **Data Augmentation.** Results obtained applying popular augmentation techniques during training. We used PyTorch's transforms, and the x axis relates to the parameter passed to the class; the higher the parameter, the heavier the transform effect (*i.e.* $x = 0$ equals to the identity transformation). Refer to Supp. Mat. for further details on the transforms.

3.4.5 Data Augmentation

We evaluate various augmentations strategy, to assess whether they improve domain generalization. In general, to adhere to a realistic use case, augmentations are applied only to the query. In the case of horizontal flipping, we apply it on entire triplets. We experiment with a ResNet-18 trained on Pitts30k with NetVLAD aggregation. Results are shown in Fig. 3.2.

Discussion. The influence of data augmentation exhibits significant variation across different test datasets. In the case of Pitts30k, augmentation procedures consistently lead to performance degradation, which may be attributed to the inherent similarity between training and testing data distributions.

However, certain augmentation strategies demonstrate measurable improvements in cross-dataset generalization. Color jittering techniques, which modify image brightness, contrast and saturation parameters, prove particularly effective. As a concrete example, when contrast adjustment is increased to a factor of 2[2], we observe recall@1 improvements of 3% on MSLS, 5% on Tokyo 24/7, and 5% on St Lucia datasets, while performance on Pitts30k and Eynsham experiences only minimal reduction (less than 1%).

Among the various augmentation methods evaluated, two approaches yield consistent benefits: random horizontal flipping with 50% application probability, and random resized cropping where image regions as small as 50% of the original dimensions are extracted and subsequently rescaled to full resolution.

[2]Implemented using PyTorch's `ColorJitter()` function.

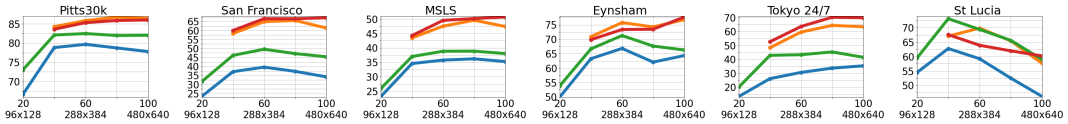


Figure 3.3: **Changing the images' resolution.** On the x-axis is the train and test resolution (N%), on the y-axis is the recall@1. Regarding the curves, red refers to ResNet-50 + NetVLAD, orange to ResNet18 + NetVLAD, green to ResNet-50 + GeM, and blue to ResNet-18 + GeM. In many cases, full resolution is not the optimal choice. NetVLAD's initial clusters computation breaks with low resolutions.

3.4.6 Resolution

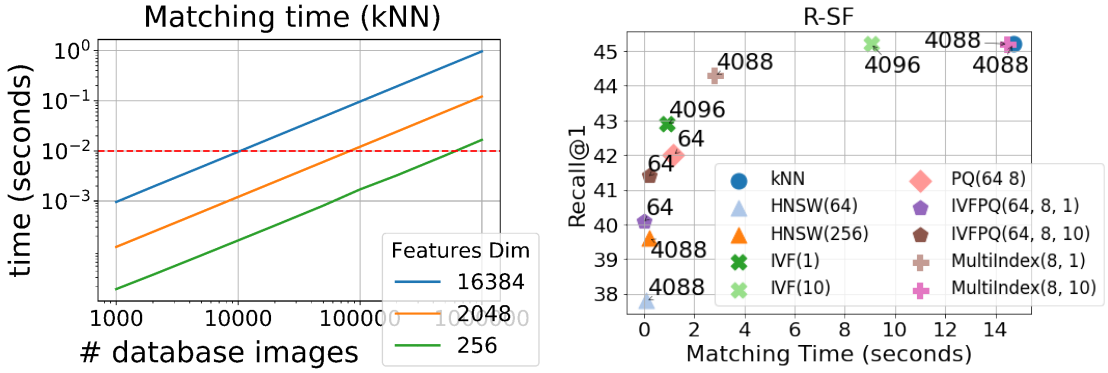
Most visual geo-localization datasets standardize on 480×640 pixel resolutions, yet the impact of resolution reduction merits systematic investigation. We evaluate this by progressively resizing images from 80% down to 20% of original dimensions during both training and testing phases on Pitts30k, employing CNN architectures with GeM or NetVLAD pooling that inherently support variable input sizes. We show results in Fig. 3.3.

Discussion. This analysis reveals counterintuitive findings: maximum resolution frequently proves unnecessary and sometimes harmful. NetVLAD descriptors exhibit greater robustness to resolution changes compared to GeM. Performance gains emerge at reduced resolutions, with 40% scaling showing particular benefits for cross-domain scenarios like St Lucia (comprising solely forward views versus Pitts30k's omnidirectional coverage), where it achieves peak recall@1. This suggests that downsampling attenuates domain-specific artifacts like distinctive textures, effectively regularizing the representation. A 60% resolution generally offers the best trade-off, implying that geo-localization relies more on structural appearance than fine details.

The computational benefits are substantial: 40% resolution (which means 256 pixels on the longest side) reduces FLOPs to 16% of original requirements through quadratic scaling. Storage demands follow similar reductions. Infact, while a retrieval systems does not require in theory to store images, storing the images can remain valuable for spatial verification and user-facing visualization tasks, and thus lower resolutions can be advantageous for cases where high fidelity visualizations are non-essential.

3.4.7 Nearest Neighbor Search and Inference Time

For deployed visual geo-localization systems, inference latency t_i becomes the critical performance metric affecting user experience. This total processing time comprises two distinct components: (1) feature extraction time t_e, determined by



(a) (b)

Figure 3.4: (a) **Matching time for one query.** The plot shows, with exact search, linear dependency on database size and features dimensionality. The red line marks the extraction time of an image for ResNet-101 + GeM; above, the bottleneck is matching time, below it is extraction time. As a rule of thumb, kNN is the bottleneck if database size times the features dimension exceeds 200M.
(b) Analysis of the **Recall-Speed-Memory trade-off** using optimized indexing techniques for neighbor search. Dots refer to a ResNet-50 + GeM (feat. dim. 1024) trained on Pitts30k. On the x axis is matching time in seconds for all queries in the dataset, on the y axis recall@1. The numbers next to the dots represent the RAM requirements in MB.

model architecture and input resolution, and (2) matching time t_m required for k-nearest neighbor search, which scales with database size, descriptor dimensionality, choice of k, and search algorithm selection.

The linear relationship between matching time and database/descriptor dimensions is demonstrated in Fig. 3.4a, while Fig. 3.4b compares various approximate nearest neighbor methods in terms of computational efficiency and memory requirements. Our evaluation considers exhaustive search alongside several optimized approaches: inverted file indexes (IVF) [248], product quantization variants (PQ, IVFPQ) [123], inverted multi-index [15], and hierarchical navigable small world graphs (HNSW) [166]. All experiments employ ResNet-50 with GeM pooling on the R-SF dataset. A more complete analysis of all methods is in Fig. 3.5.

Discussion Analysis of Fig. 3.4a reveals that matching operations dominate inference time as database size increases, while feature extraction remains consistently efficient at approximately 10ms. The algorithm comparison in Fig. 3.4b demonstrates that optimized search methods can achieve substantial efficiency gains with minimal accuracy trade-offs. Notably, IVFPQ reduces both matching time and memory usage by 98.5% while only decreasing recall from 45.4% to 41.4%.

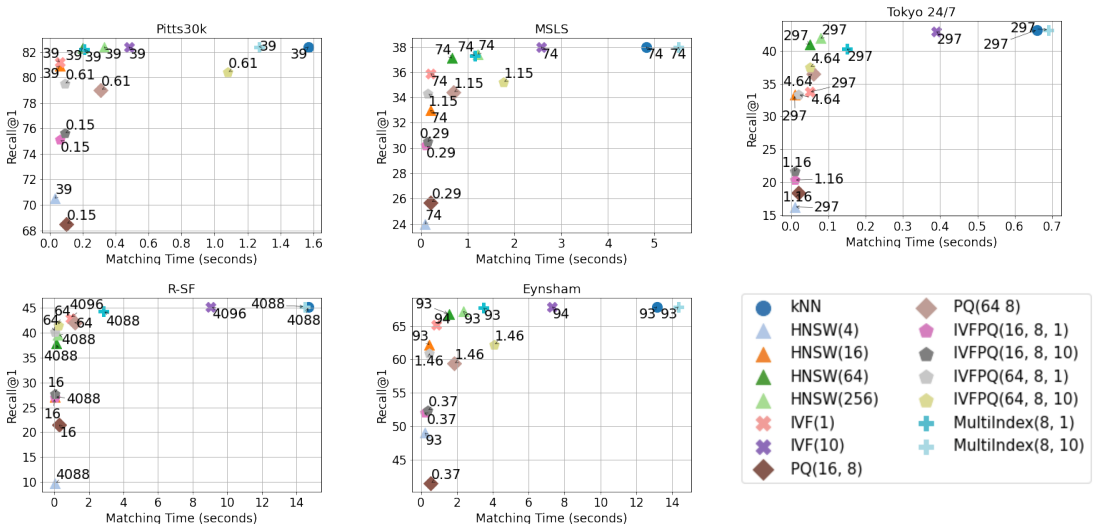


Figure 3.5: **Optimized kNN indexing: faster search & lower memory footprint.** The plots shows a number of efficient kNN variants, on different datasets, which are applied on features extracted with a ResNet-50 + GeM (features dimension 1024) trained on Pitts30k. On the x-axis is the matching time in seconds for all the queries in each dataset, while on the y-axis is the recall@1. The numbers next to the dots represent the RAM requirements of the method (memory footprint) in MB. Besides exhaustive kNN, we employ inverted file indexes (IVF) [248], product quantization (with and without inverted indexes, respectively PQ and IVFPQ) [123], the inverted multi index (MultiIndex) [15] and hierarchical navigable small world graphs (HNSW) [166]. In the legend, the parameters are shown for each method. The last parameter of IVFPQ, MultiIndex and IVF, which is either 1 or 10, represents the percentage of Voronoi cells to search, given that the search space has been split into 1000 Voronoi cells.

Memory efficiency proves particularly crucial for large-scale deployment, as descriptor vectors must reside in RAM for optimal performance. For instance, storing NetVLAD descriptors (65,536 dimensions) for the R-SF dataset's 1.05 million images requires approximately 256GB of memory. While inverted multi-index methods maintain similar memory footprints, they offer 80% faster matching with merely 0.9% recall reduction compared to exact search. These findings strongly suggest that (1) recall metrics alone provide insufficient performance characterization, and (2) search algorithm optimization represents an essential consideration for practical system implementation.

Backbone	Aggregation Method	Pre/Post-Processing Method	Pre-Proc.	Post-Proc.	Batch Parall.	Training Dataset.	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18	GeM	Hard Resize	Y	N	Y	Pitts30k	**77.8 ± 0.2**	35.3 ± 0.5	31.8 ± 0.9	33.2 ± 2.1	**64.3 ± 1.2**	**46.2 ± 0.4**
ResNet-18	GeM	Single Query	Y	N	N	Pitts30k	**77.8 ± 0.2**	**35.6 ± 0.6**	35.3 ± 1.1	34.2 ± 1.7	**64.3 ± 1.2**	**46.2 ± 0.4**
ResNet-18	GeM	Central Crop	Y	N	Y	Pitts30k	**77.8 ± 0.2**	34.8 ± 0.5	**36.4 ± 1.1**	32.6 ± 1.4	**64.3 ± 1.2**	**46.2 ± 0.4**
ResNet-18	GeM	Five Crops Mean	Y	Y	Y	Pitts30k	75.4 ± 0.3	30.2 ± 0.2	35.9 ± 0.5	34.4 ± 2.0	59.1 ± 0.7	43.3 ± 0.8
ResNet-18	GeM	Nearest Crop	Y	Y	Y	Pitts30k	74.8 ± 0.1	28.3 ± 0.3	33.8 ± 1.3	**35.7 ± 1.6**	55.5 ± 0.8	39.4 ± 0.5
ResNet-18	GeM	Majority Voting	Y	Y	Y	Pitts30k	75.1 ± 0.0	29.1 ± 0.4	34.8 ± 1.5	35.3 ± 1.3	51.8 ± 0.2	41.3 ± 0.5
ResNet-18	NetVLAD	Hard Resize	Y	N	Y	Pitts30k	**86.4 ± 0.3**	47.4 ± 1.2	58.3 ± 1.4	58.9 ± 1.1	76.8 ± 1.2	**57.6 ± 3.3**
ResNet-18	NetVLAD	Single Query	Y	N	N	Pitts30k	**86.4 ± 0.3**	47.5 ± 1.3	63.4 ± 1.2	61.4 ± 1.5	76.8 ± 1.2	**57.6 ± 3.3**
ResNet-18	NetVLAD	Central Crop	Y	N	Y	Pitts30k	**86.4 ± 0.3**	**48.0 ± 1.3**	63.2 ± 0.2	57.8 ± 0.4	76.8 ± 1.2	**57.6 ± 3.3**
ResNet-18	NetVLAD	Five Crops Mean	Y	Y	Y	Pitts30k	85.1 ± 0.2	45.3 ± 1.3	63.0 ± 0.7	60.9 ± 1.7	**78.9 ± 0.9**	54.6 ± 2.8
ResNet-18	NetVLAD	Nearest Crop	Y	Y	Y	Pitts30k	84.8 ± 0.2	46.0 ± 1.5	**67.0 ± 1.4**	**64.8 ± 0.7**	75.7 ± 1.4	53.0 ± 2.5
ResNet-18	NetVLAD	Majority Voting	Y	Y	Y	Pitts30k	84.8 ± 0.3	45.2 ± 1.4	66.9 ± 1.1	64.7 ± 0.7	77.1 ± 1.1	53.4 ± 2.3
ResNet-50	GeM	Hard Resize	Y	N	Y	Pitts30k	**82.0 ± 0.3**	38.0 ± 0.1	34.6 ± 1.4	40.7 ± 1.8	**66.3 ± 2.5**	**59.0 ± 1.4**
ResNet-50	GeM	Single Query	Y	N	N	Pitts30k	**82.0 ± 0.3**	**38.2 ± 0.3**	41.5 ± 1.8	45.4 ± 2.0	**66.3 ± 2.5**	**59.0 ± 1.4**
ResNet-50	GeM	Central Crop	Y	N	Y	Pitts30k	**82.0 ± 0.3**	37.5 ± 0.3	40.4 ± 0.9	41.0 ± 2.6	**66.3 ± 2.5**	**59.0 ± 1.4**
ResNet-50	GeM	Five Crops Mean	Y	Y	Y	Pitts30k	80.4 ± 0.1	33.2 ± 0.1	39.8 ± 2.0	43.8 ± 0.9	65.0 ± 2.4	54.4 ± 1.3
ResNet-50	GeM	Nearest Crop	Y	Y	Y	Pitts30k	79.2 ± 0.2	30.8 ± 0.2	**43.5 ± 1.4**	**46.9 ± 1.4**	63.5 ± 2.2	52.6 ± 1.4
ResNet-50	GeM	Majority Voting	Y	Y	Y	Pitts30k	79.7 ± 0.0	31.5 ± 0.1	43.0 ± 2.0	44.8 ± 1.2	62.9 ± 2.3	52.8 ± 0.9
ResNet-50	NetVLAD	Hard Resize	Y	N	Y	Pitts30k	**86.0 ± 0.1**	50.7 ± 2.0	64.3 ± 1.9	64.3 ± 1.2	77.7 ± 0.4	**60.2 ± 1.6**
ResNet-50	NetVLAD	Single Query	Y	N	N	Pitts30k	**86.0 ± 0.1**	50.6 ± 1.9	69.8 ± 0.8	67.1 ± 2.3	77.7 ± 0.4	**60.2 ± 1.6**
ResNet-50	NetVLAD	Central Crop	Y	N	Y	Pitts30k	**86.0 ± 0.1**	**50.9 ± 1.9**	68.3 ± 1.4	64.6 ± 2.2	77.7 ± 0.4	**60.2 ± 1.6**
ResNet-50	NetVLAD	Five Crops Mean	Y	Y	Y	Pitts30k	84.7 ± 0.1	47.4 ± 1.9	68.0 ± 2.2	66.5 ± 1.5	**78.6 ± 0.3**	54.3 ± 2.8
ResNet-50	NetVLAD	Nearest Crop	Y	Y	Y	Pitts30k	84.2 ± 0.2	47.0 ± 1.7	72.3 ± 1.3	**68.4 ± 0.8**	76.8 ± 0.5	52.3 ± 2.3
ResNet-50	NetVLAD	Majority Voting	Y	Y	Y	Pitts30k	84.3 ± 0.2	47.1 ± 1.7	**72.8 ± 0.8**	68.1 ± 1.3	77.5 ± 0.4	53.4 ± 2.2

Table 3.11: **Query pre/post-processing.** Results with different pre/post-processing methods are shown in the table. The batch parallelization column indicates if images have to be processed one by one or if they can be stacked in a batch for parallel computation.

3.4.8 Query pre/post-processing and Predictions Refinement

Practical geo-localization systems must handle query images with resolutions differing from database references, a scenario addressed in few datasets like R-SF [266, 152] and Tokyo 24/7 [265]. Prior solutions [10, 216, 264, 99, 221] typically process queries individually (batch size=1), ensuring accuracy at the expense of computational efficiency. We systematically evaluate alternative batch-compatible approaches that may simultaneously improve retrieval performance, categorizing them by pipeline stage (pre-processing, post-processing, prediction refinement) as illustrated in Fig. 1.

Table 3.11 presents comprehensive experimental results, with methodological details as follows. Pre-processing variants include:

- **Hard Resize**: Anisotropic scaling to database dimensions (identity transformation when resolutions match)
- **Single Query**: Isotropic resizing preserving aspect ratio (prevents batch processing for mixed resolutions)
- **Central Crop**: Isotropic scaling followed by center cropping to database dimensions

- **Five Crops**: Generation of five square regions sized to the database's shortest edge

Post-processing techniques comprise:

- **Mean**: Descriptor averaging across five crops
- **Nearest Crop**: Selection based on minimal descriptor distance to any crop
- **Majority Voting**: Consensus mechanism considering top-20 predictions per crop

Discussion. Tab. 3.11 confirms identical outcomes for *Hard Resize*, *Single Query*, and *Central Crop* when query resolutions match database images (Pitts30k, Eynsham, St Lucia). For Tokyo 24/7 and R-SF containing vertical queries, advanced methods (*Nearest Crop*, *Majority Voting*) demonstrate superior performance, particularly with robust networks, while enabling batch processing through dimension standardization.

The optimal strategy depends on application constraints. For instance, applications such as robotics use-case, where pictures may come from the same device and hence share details such as intrinsics and resolution could simply rely on a *Hard Resize*. On the other hand, application serving users in unconstrained platforms, should resort to *Single Query* if simplicity of the pipeline is important. In case efficiency is a priority, *Nearest Crop* grants both the best performance and allows to batch together multiple queries.

Source	Loss	Training Dataset	Backbone	Aggregation Method	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
[216]	Triplet	GLDv1	ResNet-50	GeM + FC 2048	**84.1**	69.5	**77.8**	**76.4**	61.8	77.3
[216]	Triplet	Sfm120k	ResNet-50	GeM + FC 2048	83.4	64.5	75.2	75.6	68.8	73.9
-	Triplet	Pitts30k	ResNet-50	GeM + FC 2048	80.1	33.7	43.6	48.2	70.0	56.0
-	Triplet	MSLS	ResNet-50	GeM + FC 2048	79.2	**73.5**	64.0	55.1	**86.1**	**90.3**
[216]	Triplet	GLDv1	ResNet-101	GeM + FC 2048	**85.1**	72.4	**77.8**	**79.8**	61.6	83.4
[216]	Triplet	Sfm120k	ResNet-101	GeM + FC 2048	83.9	64.7	77.5	78.3	62.8	76.3
-	Triplet	Pitts30k	ResNet-101	GeM + FC 2048	82.4	40.0	47.2	57.5	75.9	61.7
-	Triplet	MSLS	ResNet-101	GeM + FC 2048	79.1	**75.3**	61.9	54.9	**86.0**	**92.5**

Table 3.12: **The role of the training dataset.** The table shows results with models trained on large scale landmark retrieval datasets.

3.4.9 The role of the training dataset

This section examines the critical influence of training data selection on model performance. We evaluate publicly available SOTA models for image retrieval, comparing architectures trained on large-scale landmark recognition datasets against

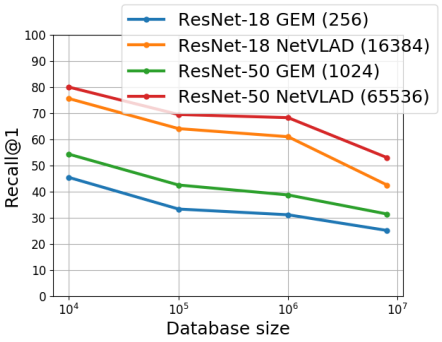


Figure 3.6: **Scaling datasets with distractors.** The plot shows the effects of exponentially increasing the size of the database up to 8M. In the legend, the descriptors dimensionality is shown between parentheses.

equivalent networks trained specifically on Pitts30k and MSLS. While NetVLAD with PCA dimensionality reduction could potentially achieve higher recall rates, we employ GeM with fully-connected layers to maintain consistency with third-party benchmark models trained on landmark datasets. Our software seamlessly integrates pretrained models from [216] and [221].

Discussion. The comparative results presented in Tab. 3.12 demonstrate the substantial impact of training data selection on model generalization. Models pretrained on extensive landmark collections, particularly GLDv1 [191], serve as effective off-the-shelf solutions across multiple visual geo-localization scenarios. This robustness stems from the diverse image content in landmark datasets, which promotes learning of transferable visual features.

When considering domain-specific training, MSLS-trained models consistently outperform their Pitts30k-trained counterparts across most datasets, benefiting from MSLS's greater scale and variability. The exception occurs with R-SF and Tokyo 24/7, where 360° panoramic views diverge significantly from MSLS's forward-facing training imagery. Pitts30k-trained models exhibit particularly poor generalization, likely due to overfitting caused by the high-parameter FC layer relative to the dataset's limited size. This interpretation is supported by Tab. 3.5, where parameter-efficient NetVLAD+PCA aggregation demonstrates superior generalization performance.

These findings suggest two key practical guidelines: first, prioritize training datasets with viewpoint characteristics matching the target application when known; second, select model complexity proportional to available training data volume.

Model	Trained on Pitts30k		Trained on MSLS	
	R@1 Single DB	R@1 Multi DB	R@1 Single DB	R@1 Multi DB
ResNet-18 + GeM	57.9	42.2 (-27.1%)	74.4	65.1 (-12.5%)
ResNet-50 + GeM	60.9	53.4 (-12.3%)	79.4	71.6 (-9.82%)
ResNet-18 + NetVLAD	70.0	67.4 (-3.7%)	83.0	79.0 (-4.1%)
ResNet-50 + NetVLAD	71.4	68.7 (-3.8%)	83.2	79.0 (-5.0%)

Table 3.13: **All-data benchmark.** Using all queries from the six datasets, *Single DB* indicates the average result from matching the queries only to their respective database, *Multi DB* refers to matching the the queries to all six databases merged.

3.4.10 Scaling datasets with distractors

Despite significant advances in both software and hardware capabilities in recent years, existing datasets for visual geo-localization remain limited in scale and coverage compared to real-world requirements (Tab. 3.15). For instance, while the San Francisco dataset represents one of the largest available collections with 1 million reference images, it captures merely 9% of the city's total area. To better understand the challenges of large-scale deployment, we constructed an expanded evaluation benchmark incorporating up to 8 million distractor images.

Our methodology begins with the 315 queries from Tokyo 24/7, first creating a baseline dataset of 10,000 images by combining relevant positives with randomly selected images from Tokyo 24/7. We then progressively scaled this collection by factors of 10, sequentially incorporating the complete Tokyo 24/7, Pitts30k, and MSLS test sets. The largest configuration combines the San Francisco database with additional images from Google Landmark v2 [293], Places 365 [315], and the MSLS training set to reach the 8 million image benchmark.

Discussion. The performance trends illustrated in Fig. 3.6 demonstrate a consistent degradation in accuracy with increasing database size, highlighting the ongoing challenges in scaling visual geo-localization systems to real-world conditions.

3.4.11 Testing on an ensemble of datasets

Real-world applications often require handling queries from diverse geographic regions and data distributions (e.g., Tokyo, San Francisco). To address this practical scenario, our benchmark framework supports evaluation on a composite dataset incorporating all test queries and reference databases: Pitts30k, MSLS, Tokyo 24/7, R-SF, Eynsham, and St Lucia. We specifically compare two evaluation protocols: (1) *Single DB* where queries are matched only against their corresponding database, and (2) *Multi DB* where queries are searched against the unified collection of all six databases.

Discussion. Tab. 3.13 presents recall@1 metrics for both evaluation protocols. The Multi DB configuration consistently shows degraded performance compared

to the averaged Single DB results, revealing the challenges of cross-dataset generalization. MSLS-trained models outperform those trained on Pitts30k across both settings, confirming the benefits of MSLS's greater scale and diversity for learning robust features. Notably, the performance degradation between Single DB and Multi DB conditions is more pronounced for GeM-based methods compared to NetVLAD variants, suggesting architectural differences in handling heterogeneous data distributions.

Backbone	Aggregation Method	Dataset	Training Dataset	R@1 Pitts30k	R@1 MSLS	R@1 Tokyo 24/7	R@1 R-SF	R@1 Eynsham	R@1 St Lucia
ResNet-18	GeM	ImageNet	Pitts30k	77.8 ± 0.2	35.3 ± 0.5	**35.3** ± 1.1	**34.2** ± 1.7	64.3 ± 1.2	46.2 ± 0.4
ResNet-18	GeM	GLDv2	Pitts30k	74.2 ± 0.4	30.9 ± 0.6	22.3 ± 1.9	20.4 ± 1.7	55.0 ± 2.0	43.3 ± 0.7
ResNet-18	GeM	Places 365	Pitts30k	**78.1** ± 1.0	**36.2** ± 0.9	31.8 ± 0.7	32.8 ± 1.6	**65.0** ± 2.1	**48.8** ± 2.1
ResNet-18	NetVLAD	ImageNet	Pitts30k	**86.4** ± 0.3	**47.4** ± 1.2	**63.4** ± 1.2	**61.4** ± 1.5	76.8 ± 1.2	**57.6** ± 3.3
ResNet-18	NetVLAD	GLDv2	Pitts30k	83.3 ± 0.5	39.9 ± 0.9	54.2 ± 2.3	41.1 ± 3.6	71.4 ± 2.6	46.8 ± 1.9
ResNet-18	NetVLAD	Places 365	Pitts30k	85.9 ± 0.4	**47.4** ± 0.6	57.9 ± 1.4	59.9 ± 3.2	**78.7** ± 0.7	50.4 ± 1.0
ResNet-50	GeM	ImageNet	Pitts30k	82.0 ± 0.3	38.0 ± 0.1	**41.5** ± 1.8	**45.4** ± 2.0	66.3 ± 2.5	59.0 ± 1.4
ResNet-50	GeM	GLDv2	Pitts30k	77.9 ± 0.5	35.2 ± 0.8	27.6 ± 2.1	37.2 ± 1.0	62.7 ± 1.6	48.4 ± 1.7
ResNet-50	GeM	Places 365	Pitts30k	**82.5** ± 0.4	**40.8** ± 0.3	41.3 ± 0.7	45.3 ± 0.6	**66.9** ± 1.3	**60.8** ± 1.6
ResNet-50	NetVLAD	ImageNet	Pitts30k	86.0 ± 0.1	**50.7** ± 2.0	**69.8** ± 0.8	**67.1** ± 2.3	**77.7** ± 0.4	**60.2** ± 1.6
ResNet-50	NetVLAD	GLDv2	Pitts30k	81.7 ± 0.6	43.5 ± 1.0	56.7 ± 0.9	54.1 ± 1.8	71.4 ± 0.6	42.3 ± 2.5
ResNet-50	NetVLAD	Places 365	Pitts30k	**86.2** ± 0.5	49.9 ± 2.0	66.3 ± 3.3	59.7 ± 3.5	75.4 ± 2.0	57.2 ± 5.5
ResNet-18	GeM	ImageNet	MSLS	**71.6** ± 0.1	**65.3** ± 0.2	**42.8** ± 1.1	**30.5** ± 0.8	**80.3** ± 0.1	**83.2** ± 0.9
ResNet-18	GeM	GLDv2	MSLS	60.7 ± 0.5	64.5 ± 0.7	30.9 ± 3.3	21.5 ± 0.8	79.2 ± 0.6	78.1 ± 1.0
ResNet-18	GeM	Places 365	MSLS	**71.6** ± 0.9	64.8 ± 1.1	36.6 ± 2.2	25.5 ± 0.3	80.1 ± 0.5	82.4 ± 0.6
ResNet-18	NetVLAD	ImageNet	MSLS	**81.6** ± 0.5	**75.8** ± 0.1	**62.3** ± 1.6	**55.1** ± 0.9	**87.1** ± 0.2	**92.1** ± 0.7
ResNet-18	NetVLAD	GLDv2	MSLS	73.3 ± 0.6	75.3 ± 0.3	53.4 ± 1.3	40.7 ± 2.9	86.1 ± 0.1	87.6 ± 0.9
ResNet-18	NetVLAD	Places 365	MSLS	79.7 ± 0.5	75.6 ± 0.2	61.5 ± 0.7	48.6 ± 1.5	86.5 ± 0.1	90.4 ± 0.4
ResNet-50	GeM	ImageNet	MSLS	77.4 ± 0.6	72.0 ± 0.5	**55.4** ± 2.5	**45.7** ± 1.0	83.9 ± 0.6	**91.2** ± 0.7
ResNet-50	GeM	GLDv2	MSLS	71.1 ± 1.7	72.4 ± 0.2	47.6 ± 0.4	35.8 ± 1.6	84.0 ± 0.4	86.1 ± 1.1
ResNet-50	GeM	Places 365	MSLS	**78.2** ± 1.1	**72.7** ± 0.6	51.8 ± 2.7	41.8 ± 2.2	**84.4** ± 0.2	89.3 ± 0.8
ResNet-50	NetVLAD	ImageNet	MSLS	**80.9** ± 0.0	76.9 ± 0.2	**62.8** ± 0.9	**51.5** ± 1.2	**87.2** ± 0.3	**93.8** ± 0.2
ResNet-50	NetVLAD	GLDv2	MSLS	74.7 ± 1.0	**77.4** ± 0.4	55.0 ± 1.7	45.4 ± 1.5	85.1 ± 0.5	87.7 ± 0.8
ResNet-50	NetVLAD	Places 365	MSLS	80.0 ± 1.1	75.6 ± 0.1	51.3 ± 3.3	44.8 ± 2.3	86.9 ± 0.1	91.3 ± 0.2

Table 3.14: **Pretraining the backbone on other datasets.**

3.4.12 Pretraining the backbone on other datasets

This section examines the potential benefits of using alternative pretraining datasets beyond ImageNet for initializing our visual geo-localization system's backbone network. We evaluate two specialized datasets: Places 365 [315] for scene recognition and Google Landmark v2 (GLDv2) [191, 292] for large-scale landmark identification. The Places 365 models employ conventional classification training, while GLDv2 utilizes ArcFace Loss [63], implementing the methodology suggested by [302].

Discussion. The comprehensive experimental results presented in Tab. 3.14 demonstrate that ImageNet pretraining remains superior in most scenarios. While

performance differences are generally marginal, notable decreases occur specifically for R-SF and Tokyo 24/7 datasets when using alternative pretraining sources. Even in the limited cases where GLDv2 or Places365 achieve slightly better results, the practical significance of these minor improvements is questionable. Considering both performance outcomes and the widespread availability of ImageNet-pretrained models compared to their specialized counterparts, ImageNet emerges as the clearly preferable option for backbone initialization.

	# database	# queries	Dataset size	Area (Km²)	Perimeter (Km)	Environment	Day/night changes	Long-term variations
Pitts30k	30K	21.8K	2.0 GB	0.615	3.42	Urban	N	Y
MSLS	973K	541K	56 GB	N/A	N/A	Urban + Suburban	Y	Y
Tokyo 24/7	75K	315	4.0 GB	2.1	5.8	Urban	Y	Y
R-SF	1.05M	598	36 GB	13.6	14.0	Urban	N	Y
Eynsham	24K	24K	1.2 GB	N/A	N/A	Urban + Suburban	N	N
St Lucia	1.5K	1.5K	124 MB	0.69	3.5	Suburban	N	N

Table 3.15: **Summary of datasets used.** Long-term variations refers to images taken at least one year apart.

3.4.13 Dataset specifications

In this section we provide an overview of all the considered benchmarks, providing details on how they were built, their characteristics and distributions as well as query/database examples.

Pitts30k [10] represents a curated subset of the larger Pitts250k dataset [267], partitioned into training, validation, and test subsets. This dataset derives from Google Street View imagery captured in Pittsburgh, where equirectangular panoramas undergo tile-based cropping followed by gnomonic projection. Notably, database and query images exhibit a temporal separation of two years while maintaining consistent weather conditions throughout.

Mapillary Street Level Sequence (MSLS) [289] offers extensive geographical coverage spanning urban environments across six continents, encompassing diverse domains, camera types, and seasonal variations. Similar to Pitts30k, MSLS follows a tripartite division of training, validation, and test sets. However, due to the unavailability of test set ground truths, this chapter follows established practice [105] by reporting validation recall metrics. Among available datasets, only Pitts30k and MSLS provide temporally varied training data essential for visual geo-localization model training [10].

Tokyo 24/7 [265] features an asymmetric distribution with an extensive database (sourced from Google Street View) contrasted against a limited set of query images. These queries are equally distributed across three illumination conditions: daylight, sunset, and nighttime, with the latter captured manually using mobile devices. Previous research [10, 156, 303] has employed Tokyo Time Machine (Tokyo TM) as complementary training data for this benchmark.

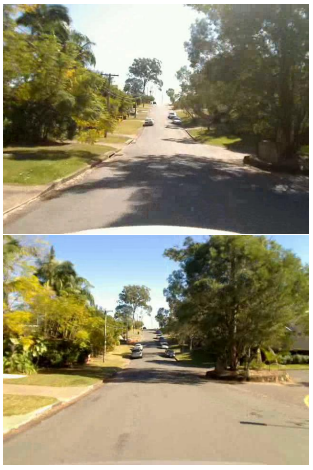
(a) Pitts30k

(c) San Francisco

(e) Eynsham

(b) Tokyo 24/7

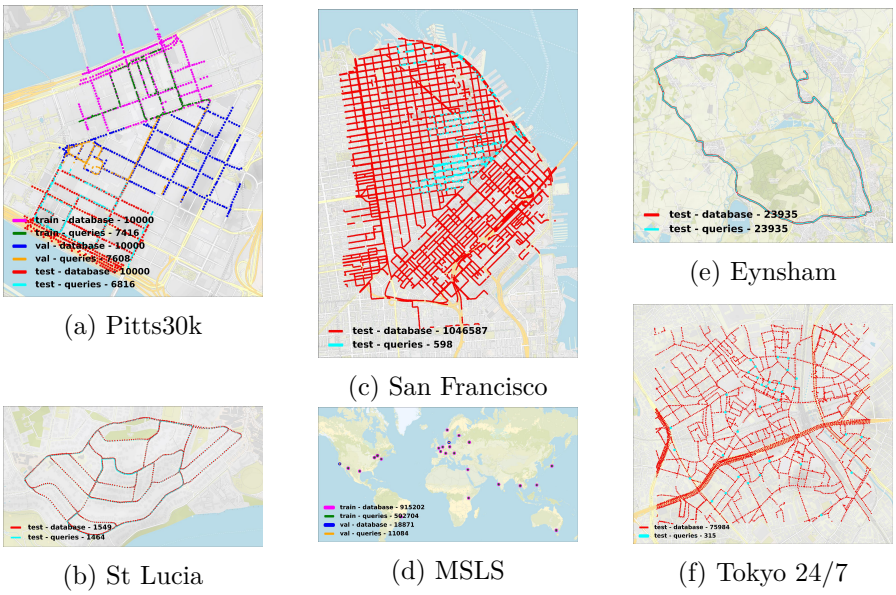
(d) MSLS



(f) St Lucia

Figure 3.7: **Examples of a query and a positive** for each of the used dataset.

San Francisco [48] mirrors Tokyo 24/7's structural paradigm, comprising a vehicle-acquired database alongside substantially fewer mobile-captured queries.



(a) Pitts30k (b) St Lucia (c) San Francisco (d) MSLS (e) Eynsham (f) Tokyo 24/7

Figure 3.8: **Maps of used datasets,** self-generated with our open source codebase.

From available Structure from Motion reconstructions, this chapter adopts the version from [266, 152], which designated the Revisited San Francisco version with superior label accuracy by relying on 6-degree-of-freedom query poses.

Eynsham [60] consists of grayscale imagery captured by vehicle-mounted cameras traversing urban and rural Oxfordshire routes. The dataset organization utilizes the first circuit as reference database and the subsequent traversal as queries. Original equirectangular panoramas undergo segmentation into five distinct crops per frame.

St Lucia [172] contains video-derived imagery captured during repeated vehicular traversals through Brisbane's riverside suburb. This chapter employs the initial and final routes as database and query sets respectively. To address frame density concerns from video sampling, a spatial subsampling strategy retains one frame per 5-meter interval. All preprocessing operations, including data retrieval, are implemented through our publicly available framework.

Figure 3.7 presents representative query-database image pairs that demonstrate the challenges of cross-dataset generalization, highlighting variations in viewpoint, environmental conditions, and acquisition parameters. Quantitative characteristics including image counts and geographical coverage are summarized in Table 3.15, while Figure 3.8 visualizes the spatial distribution of imagery across different locations.

Method	Feat. Dim.	R@1 Pitts30k	R@1 Pitts250k	R@1 Tokyo 24/7
VGG16 + NetVLAD + PCA [200]	4096	85.2	86.5	68.9
VGG16 + NetVLAD [200]	32768	-	84.1	60.0
SRALNet (ICRA21) [201]	4096	-	87.8	72.1
SRALNet (ICRA21) [201]	32768	85.1	85.8	68.6
APPSVR (ICCV21) [200]	4096	87.4	88.8	77.1
APPSVR (ICCV21) [200]	32768	-	86.6	68.3
ResNet-18 + NetVLAD + PCA (Ours)	4096	86.8	87.9	72.2
ResNet-18 + NetVLAD (Ours)	16384	87.2	88.1	73.7

Table 3.16: Comparison between recent SOTA methods, and a simple ResNet-18+NetVLAD where we use all the insight gained from the benchmark to find its optimal configuration: training with data augmentation, resize 80%, and majority voting post-processing for Tokyo 24/7 (since queries have different resolutions).

3.5 Discussions and Findings

This chapter has presented a modular framework designed to build, train, and evaluate a wide array of VG architectures, with the ability to interchange each module of a geo-localization pipeline. Through extensive experiments, we offer insights into how various design and engineering decisions, both at training and inference stages, influence performance and resource requirements (in terms of FLOPs, storage, and time).

Architecture. Our evaluation indicates that ResNet-50 strikes a strong balance between accuracy and computational efficiency when used as a CNN backbone. Additionally, we explore the application of Visual Transformers to VG for the first time, showing they can deliver strong results when compared to traditional CNNs. Among these, CCT emerges as particularly promising, achieving better performance than ResNet-50 while maintaining a lightweight profile similar to ResNet-18. Regarding feature aggregation strategies, CRN typically achieves the highest accuracy, albeit with high training costs. Conversely, GeM pooling, which is significantly more efficient, demonstrates superior generalization—especially when trained on large, diverse datasets. The most effective overall configuration in our experiments is CCT in combination with NetVLAD.

Negative mining. Negative mining remains a key component in metric learning for retrieval tasks. Our results reinforce its importance, showing that partial mining strategies can often match or even surpass full mining in performance, while significantly lowering computational overhead.

Training dataset. As anticipated, training on a large-scale, diverse dataset collected across various cities and conditions leads to markedly better performance. This underscores the critical role of the training set and highlights the pitfalls

of comparing models trained on different datasets—an issue still prevalent in the literature [134, 306] and one that should be avoided for fair evaluation.

Image size and data augmentation. Data augmentation continues to play a beneficial role for deep learning models. In our assessments, color jittering proved to be dataset-dependent in its effectiveness, whereas horizontal flipping and resized cropping consistently offered moderate improvements. Notably, we found that using full-resolution images (typically 480x640) is often unnecessary—resizing to 60% not only decreases FLOPs, but also maintains, or even improves, performance on average.

Inference time and kNN search. This chapter includes a detailed analysis of kNN search strategies within VG, offering a rare, in-depth comparison of efficient neighbor search algorithms and compact descriptor representations. Our findings show that selecting an appropriate search method can drastically improve time and memory efficiency, with minimal effects on retrieval accuracy. Moreover, we demonstrate that advanced kNN techniques can bridge the performance gap between compact and high-dimensional descriptors in terms of matching time and memory usage.

Final remarks. The insights gathered from these experiments are highly valuable for tailoring VG systems to specific applications and constraints. As illustrated in Tab. 3.1, it is possible to fine-tune a basic architecture using the strategies discussed to reach performance levels comparable to much more sophisticated—yet unoptimized—approaches (refer to Tab. 3.16).

Limitations. Despite its flexibility, the presented framework has some inherent constraints. It is currently tailored for VG systems in outdoor urban scenarios, limited to single-image localization, and does not analyze robustness to viewpoint or illumination changes, unlike other works [306]. Additionally, certain recent approaches [96, 200] and loss functions [158] have not yet been integrated or assessed. Nevertheless, we intend to maintain and enhance the software and associated platform, broadening its scope to encompass more techniques, use-cases, and pipeline components in future iterations.

Chapter 4

Learning Sequential Descriptors for Sequence-based Visual Place Recognition

This chapter presents a comprehensive investigation into sequence-based approaches for Visual Place Recognition (VPR), addressing the critical need for robust localization systems that can effectively leverage temporal information. While conventional VPR systems process individual frames independently we demonstrate how sequential processing of image streams can significantly enhance performance. Our work develops sequence-aware models, incorporating novel transformer architectures and innovative spatiotemporal aggregation mechanisms.

The dominant paradigm for sequence processing in VPR, known as sequence matching operates through independent frame processing followed by similarity matrix construction. While effective in fixed route scenarios, this approach suffers from two fundamental limitations: computational complexity that scales linearly with both sequence length and map size and sensitivity to assumptions about motion dynamics that often fail in real-world deployments. These limitations motivate our exploration of alternative architectures that can more effectively capture temporal relationships while maintaining computational efficiency.

While the concept of sequential descriptors (*i.e.* an embedding representing a whole sequence) has been introduced, the field remains underdeveloped.

To address these gaps, this chapter makes several key contributions. First, we establish a taxonomy of sequential descriptor architectures, categorizing methods by their frame aggregation mechanisms and analyzing the inherent trade-offs of each design paradigm. Our empirical evaluation framework extends beyond traditional recall metrics to assess practical deployment factors including computational efficiency, memory requirements, and scalability across datasets of varying sizes, incorporating both established and novel architectural variants. We provide particular focus on Transformer-based architectures, investigating their efficacy as

both feature extraction backbones and complete solutions for spatiotemporal modeling. Finally, we introduce SeqVLAD, an innovative temporal aggregation layer that effectively captures spatiotemporal patterns in image sequences, demonstrating state-of-the-art performance across multiple benchmark datasets. Code and models are available at https://github.com/vandal-vpr/vg-transformers.

The work described in this chapter has been previously published in a paper:

- R. Mereu(*), G, Trivigno (*), G, Berton, C. Masone, B. Caputo, Barbara. **Learning Sequential Descriptors for Sequence-Based Visual Place Recognition** In *IEEE Robotics and Automation Letters* (IROS/RAL) 2022 [170];

4.1 Introduction

In this chapter, we explore whether visual localization performance can be enhanced by leveraging sequences of images instead of relying solely on individual frames. Many applications of Visual Place Recognition (VPR), particularly in the context of mobile robotics, naturally involve the acquisition of image streams as a robot moves through its environment. Tasks such as loop closure detection in SLAM, re-localization in GPS-denied environments, or navigating under changing conditions can benefit significantly from reasoning over temporal continuity rather than isolated observations. Building upon the insights derived from the benchmark presented in the previous chapter, we train and evaluate robust sequence-based models. In doing so, we experiment with transformer-based architectures and introduce novel design elements for aggregating visual and temporal cues over sequences.

In mobile robotics, recognizing previously visited locations, known as Visual Place Recognition (VPR) [169], plays a crucial role both for loop closure in Simultaneous Localization And Mapping (SLAM) and for providing coarse localization when GPS signals are unavailable. This task involves processing a continuous stream of images captured by the robot during navigation, raising the question of how to best utilize temporal information within the data. Currently, the most common approach to leverage sequential information is through *sequence matching* [108, 174] (see Fig. 4.1 top). Here, each frame from the input sequence is independently compared against a database of reference images to construct a similarity matrix, which is then analyzed to identify the most probable trajectory by aggregating similarity scores. While effective, this method faces limitations in generalizability, as searching the similarity matrix depends heavily on assumptions about the robot's motion model, and in efficiency, since computational costs increase linearly with both sequence length and map size [92].

A handful of recent studies have acknowledged these limitations of sequence matching and instead presented approaches based on *sequential descriptors*, which encode entire sequences into compact representations [92, 82, 289, 94] (see Fig. 4.1 bottom). This direction is promising for two key reasons: i) it offers greater efficiency and scalability compared to sequence matching, and ii) sequential descriptors inherently capture temporal relationships within video streams, leading to improved robustness against high-confidence false matches relative to single-image descriptors [92]. However, research on sequential descriptors remains in early stages, with limited comparative analysis of different architectural approaches. This chapter addresses this gap through the following contributions:

- A taxonomy classifying sequential descriptors according to their method of combining information across frames, emphasizing the strengths and weaknesses of each design choice. Note that sequential matching techniques are not included in this taxonomy.

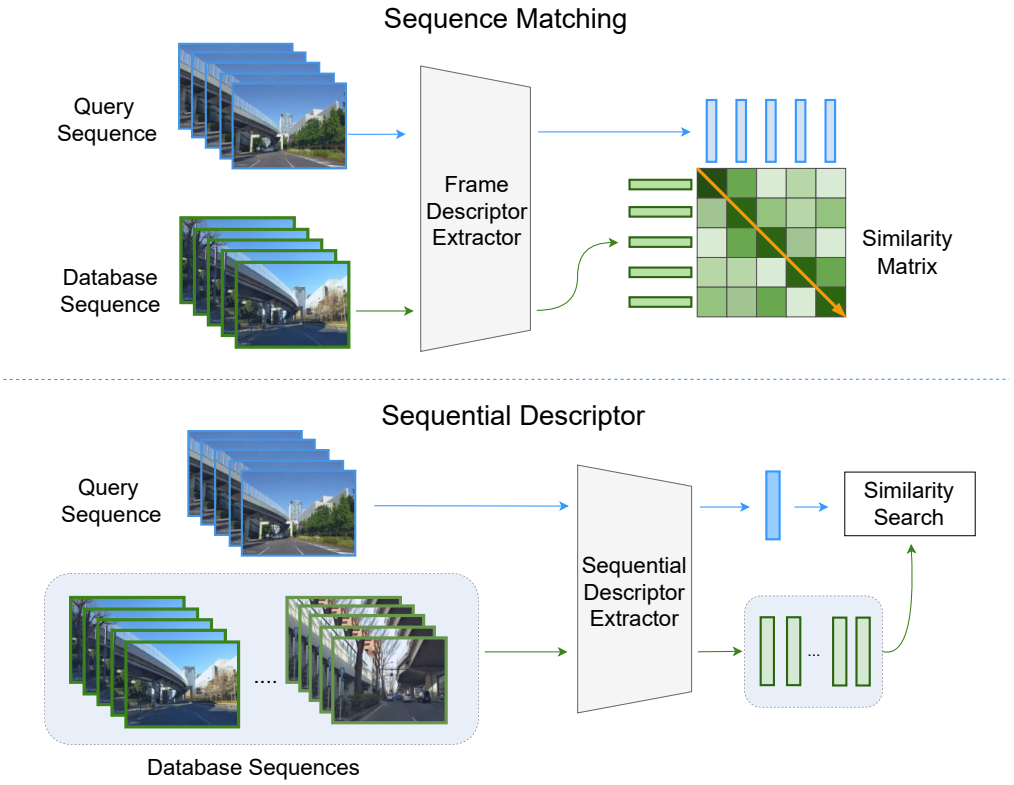


Figure 4.1: (Top) **Sequence matching** processes each frame in the sequences separately to generate single-image descriptors. Frame-to-frame similarity scores form a matrix, and the best-matching sequence is identified by aggregating these scores. (Bottom) **Sequential descriptors** encode entire sequences into vectors, allowing sequence-to-sequence similarity comparisons for direct matching.

- An empirical assessment evaluating sequential descriptors not only in terms of recall@N performance but also considering scalability, hardware demands, and real-world applicability across small and large-scale datasets. The evaluation extends beyond existing architectures to include additional methods.
- An investigation into the use of Transformers for computing sequential descriptors, examining their role both as backbones and standalone solutions.
- The development of *SeqVLAD*, a novel aggregation layer that leverages temporal cues within sequences, achieving state-of-the-art results across multiple datasets.

4.2 Related Works

Sequence matching. The sequence matching approach is a widely recognized paradigm [108, 174] that consists of two main stages. Initially, a similarity matrix is constructed by comparing descriptors from each query frame against all database sequence frames. Subsequently, the most suitable database sequence is identified by combining individual similarity scores while making simplified assumptions, such as constant robot velocity or no pauses [237], limiting applicability in real-world scenarios. Numerous sequential matching approaches have been developed to mitigate these constraints, either by utilizing egomotion data or implementing sophisticated techniques [188, 202, 189]. SeqMatchNet [93] has additionally tackled the limitation that such methods depend on single-image descriptors trained independently of the subsequent score aggregation process. Nevertheless, sequence matching still exhibits inherent limitations as outlined in [92]: i) erroneous single-image descriptor matches can lead to false positives, which could potentially be identified through sequential analysis, and ii) the computational requirements increase proportionally with both database size and sequence length.

Sequential descriptors. Methods based on sequential descriptors represent entire sequences with a single compact descriptor. This enables the integration of temporal information directly into the descriptors and facilitates sequence-to-sequence similarity comparisons, thereby improving matching efficiency. While this concept has been investigated in areas somewhat related to VPR, including Video Re-Identification [297, 24, 295] and 3D-based localization [273, 71], its adoption for VPR remains limited. Facil et al. [82] were among the first to explore sequential descriptors for VPR, proposing three distinct approaches: descriptor concatenation, feature fusion through fully connected layers, and temporal integration using LSTM networks. These findings were later expanded upon in [289] using the MSLS dataset. Garg et al. [94] developed a sequential descriptor by applying fixed discrete convolutions to individual frame descriptors. SeqNet [92] further advanced this direction by employing learned 1D temporal convolutions to pool frame-level features into sequence-level descriptors.

Current approaches to sequential descriptors [92, 82, 289, 94] utilize diverse architectural designs for learning sequence embeddings. However, the literature lacks a systematic comparison and classification of these different design choices. This chapter offers a thorough examination of various architectures for learning sequential descriptors, presented as both a taxonomy categorizing descriptor fusion methods and an extensive experimental assessment. Furthermore, this study not only analyzes existing solutions from [92, 82, 289, 94], but also broadens the methodological scope with additional architectural variations, including Transformer-based approaches and SeqVLAD, the first aggregation layer specifically designed for sequence-based VPR.

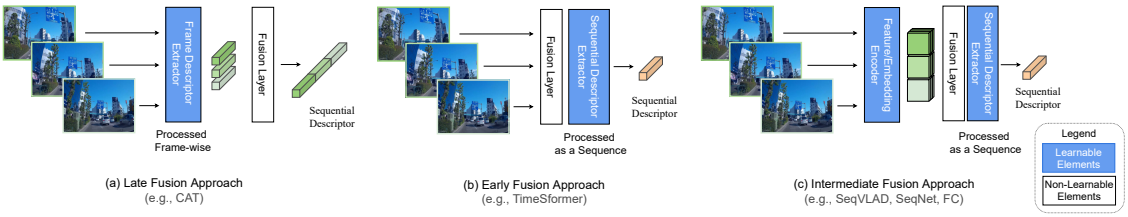


Figure 4.2: Taxonomy of the sequential descriptors architectures, obtained by composing learnable networks and parameterless fusion.

4.3 Taxonomy

4.3.1 Problem Setting

The sequential VPR task is defined according to the *seq2seq* framework established by [289]. Given a query sequence requiring localization, the task is formulated as a retrieval problem where success is achieved if any sequence within the top-N results correctly matches the query (measured by recall@N). A correct match is defined as any frame in the query sequence being within 25 meters of a frame in the database sequence.

This chapter presents a taxonomy of architectures designed for learning sequential descriptors in the context of sequential VPR. For a sequence I^L comprising L images, these architectures implement a learnable mapping $f_\theta : I^L \to \mathbb{R}^D$, producing a single D-dimensional descriptor that represents the entire sequence. The focus is on place recognition in outdoor sequences captured from vehicles, with particular attention to *short sequences*, as explored in prior work [92, 93], to avoid the computational delays associated with processing longer sequences.

4.3.2 Architectures for Learned Sequential Descriptors

Current architectures for sequential descriptors [92, 82, 289] integrate learnable functions with parameterless fusion mechanisms, such as vector concatenation, to achieve the mapping $f_\theta : I^L \to \mathbb{R}^D$. This chapter categorizes these architectures based on the stage at which fusion occurs. Three distinct paradigms are identified, illustrated in Fig. 4.2: late fusion, early fusion, and intermediate fusion. Below, these categories are detailed, focusing on how existing methods align with them, while also introducing additional approaches for each.

Late fusion

Late fusion methods employ a trainable network that processes each image in the sequence independently, merging the outputs only at the final stage to form a single sequential descriptor (see Fig. 4.2a). This approach is straightforward to implement, as it can utilize pre-existing models for single-image processing. However, this

simplicity introduces limitations, as the network can only access one frame at a time, restricting its ability to leverage temporal information. Many existing methods fall into this category [82, 289, 94], employing established convolutional networks like NetVLAD [10] and GeM [216] for single-image processing, followed by fusion via concatenation (CAT) or pooling.

In the experiments detailed in Sec. 4.4, these methods are extended by incorporating more recent architectures based on visual Transformers, which treat images as token sequences. Two visual Transformer backbones are evaluated: ViT [69] and CCT [103]. ViT includes a learnable *CLS token*, whose output embedding serves as an effective representation for tasks such as classification [69] or image retrieval [192]. This embedding is used here as the single-image descriptor. CCT [103], in contrast, employs a specialized pooling layer called SeqPool, which aggregates all token embeddings into a single vector representation. In both cases, the fusion of single-image embeddings is performed via concatenation.

Early fusion

Early fusion adopts the opposite approach, employing a learnable model that processes all frames simultaneously (see Fig. 4.2b). This allows full access to temporal information but may demand greater computational resources during training. To our knowledge, no existing sequential descriptors currently implement this strategy. To explore this paradigm, architectures from the Action Recognition domain are adapted. TimeSformer [24] extends self-attention to spatial and temporal dimensions, treating input frames as patches converted into tokens. Similar to ViT [69], it incorporates a CLS token, whose output embedding serves as the sequential descriptor. Additionally, the $2D + 1$ ResNet [269] is tested, replacing 3D convolutions with a factorization into spatial and temporal convolutions.

Intermediate fusion

Intermediate fusion strikes a balance between early and late fusion. An initial learnable stage processes individual frames, followed by a fusion operation and a final learnable component that produces the sequential descriptor (see Fig. 4.2c). This approach offers greater flexibility than late fusion, as the post-fusion model can access information from all frames, while being less resource-intensive than early fusion. Examples of this paradigm include [92] and [82], where a CNN extracts frame-level features, followed by reshaping and a final learnable layer (a FC layer in [82] or a 1D temporal convolution in SeqNet [92]).

This chapter introduces **SeqVLAD**, an intermediate fusion method that generalizes NetVLAD [10] to handle sequences of arbitrary length. NetVLAD interprets CNN feature maps as a set of local descriptors, clusters them into visual words, and generates a global descriptor based on cluster statistics. SeqVLAD extends this by

reshaping frame-level features into a unified set of M D-dimensional descriptors prior to aggregation. Two variants are presented to accommodate both CNN and Transformer backbones (see Fig. 4.3).

SeqVLAD for CNNs For an input sequence I^L, the CNN backbone produces a tensor of shape $L \times H \times W \times D$, where L is the sequence length and $H \times W \times D$ represents the feature maps per frame. SeqVLAD treats this as $M = L \cdot H \cdot W$ local descriptors.

SeqVLAD for Transformers For I^L, a Transformer backbone outputs T D-dimensional embeddings per frame, which SeqVLAD interprets as $M = L \cdot T$ embeddings.

Following fusion, SeqVLAD aggregates embeddings using soft-assigned residuals relative to cluster centroids:

$$SeqVLAD(k) = \sum_{f=1}^{L} \sum_{i=1}^{M} \bar{a}_k(\mathbf{x}_{fi}) \cdot (\mathbf{x}_{fi} - \mathbf{c}_k) \tag{4.1}$$

where $\mathbf{x}_{fi}$ is a feature or embedding, $\mathbf{c}_k$ is the k-th centroid, and $\bar{a}_k(\mathbf{x}_{fi})$ is the soft-assignment:

$$\bar{a}_k(\mathbf{x}_{fi}) = \frac{e^{\mathbf{w}_k^T \mathbf{x}_{fi} + b_k}}{\sum_{k'=1}^{K} e^{\mathbf{w}_{k'}^T \mathbf{x}_{fi} + b_{k'}}} \tag{4.2}$$

Like other intermediate fusion methods [92, 82], SeqVLAD incorporates temporal information. However, unlike [92, 82], which use generic learnable layers, SeqVLAD adapts an aggregation layer specifically designed for VPR. Key advantages of SeqVLAD include:

- *Scalability*: parameter count and output dimensionality are independent of sequence length;
- *Flexibility*: supports comparison of sequences with varying lengths, making it suitable for real-world applications where sequence lengths may differ.

4.4 Experiments

4.4.1 Experimental setup

Datasets. The experiments in this chapter utilize two datasets: MSLS [289] and Oxford RobotCar [164] (see Tab. 4.1).

MSLS is a large-scale dataset specifically designed for visual place recognition (VPR) research, consisting of diverse street-level image sequences captured across

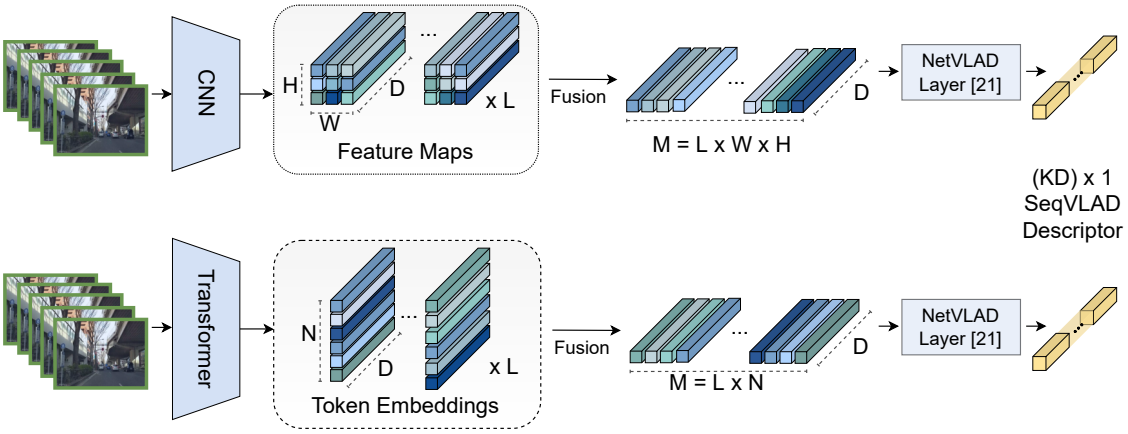


Figure 4.3: Representation of the SeqVLAD layer, in conjunction with a CNN backbone (top) and a Transformer backbone (bottom).

multiple cities and under varying environmental conditions. However, the ground truth labels for the test split were never released by the dataset authors, nor have there been any announcements regarding their availability. To address this limitation, many studies, following [105], compute recall metrics directly on the validation set. For a more rigorous evaluation, this chapter advocates for a structured train-validation-test split and presents the following partitioning:

- *MSLS Test set*: *Copenhagen, San Francisco*—previously the official validation set;
- *MSLS Validation set*: *Amsterdam, Manila*—these two cities were originally part of the training set in [289]. The selection criteria were (i) comparable size to the test split and (ii) geographic diversity, mirroring the original validation set;
- *MSLS Train set*: identical to the official training set from [289], excluding the two cities designated for validation.

Given the substantial size of the MSLS training set (see Tab. 4.1), training certain methods on it is computationally impractical (discussed further in Sec. 4.4.2). To facilitate comparisons with less scalable approaches, an additional experiment is conducted where models are trained solely on Melbourne, following the setup of [92, 93], while retaining the same validation and test sets.

Oxford-RobotCar [164] consists of repeated traversals along a single route in Oxford, captured under different seasonal and lighting conditions. Due to the absence of a standardized split in prior work [94, 92, 93], this chapter introduces a lap-based partitioning scheme to evaluate model robustness under domain shifts. The split incorporates Day/Night variations for training and testing, while validation

is performed on Snow/Overcast conditions to mitigate overfitting and account for limited nighttime data. The sequences, sampled every 2m, are divided as follows:

- *RobotCar Test set*: queries: 2014-12-16-18-44-24 (winter night); database: 2014-11-18-13-20-12 (fall day).
- *RobotCar Validation set*: queries: 2015-02-03-08-45-10 (winter day, snow); database: 2015-11-13-10-28-08 (fall day, overcast).
- *RobotCar Train set*: queries: 2014-12-17-18-18-43 (winter night, rain); database: 2014-12-16-09-14-09 (winter day, sun).

Methods. Experiments are conducted using sequential descriptors from all three paradigms outlined in Sec. 4.3. For late fusion, the concatenation (CAT) of NetVLAD and GeM from [289] is evaluated, along with extensions incorporating Transformer-based backbones such as ViT [69] and CCT [103] (in two variants, CCT224 and CCT384). Delta Descriptors [94] are also tested using their official implementation. For intermediate fusion, results from SeqNet [92] are included, alongside experiments employing a Fully-Connected (FC) layer to process flattened and fused frame-level representations from NetVLAD and ViT. SeqNet's official implementation was used, but its scalability is limited due to the need to load all single-image descriptors into memory. Thus, on Melbourne, training was restricted to the subset defined in [92]. Additionally, SeqNet's results from [92] are extended to include various backbones, including CNNs and Transformers. Results for SeqVLAD are also provided, covering multiple backbone architectures. For early fusion, TimeSformer [24] and 2D+1 convolutions [269] are evaluated. Transformer-based models require resized inputs: TimeSformer, ViT, and CCT224 use 224×224 resolution, while CCT384 operates at 384×384. CNNs are tested at 480×640. To ensure comparability, PCA is applied to standardize descriptor dimensionality, which also enhances retrieval efficiency. PCA is computed post-training and used solely during inference. For reference, results from established sequence matching methods—SeqSLAM [174], HVPR [92], and SeqMatchNet [93]—are included, using their official implementations. Delta Descriptors and SeqSLAM rely on pre-computed single-frame descriptors without training, marked with ** in the results.

Training. Training follows the protocol from [289], adapting NetVLAD's triplet loss [10] to sequential data. Hard-negative mining, as described in [289], is employed, caching 1000 randomly sampled negative sequences to avoid full database feature computation. The cache updates every 1000 triplets. Early stopping is applied if recall@5 on the validation set does not improve for 5 epochs. An epoch is defined as 5000 queries, with training performed using the Adam optimizer [136]. Batch size is set to 4 triplets, each comprising a query sequence, its positive, and 5 negatives. At inference, candidates are retrieved via exhaustive kNN search, consistent with [289].

Table 4.1: Number of sequences composing the datasets.

Dataset Name	Image Resolution	Seq. Length	# train db/queries	# validation db/queries	# test db/queries
MSLS	480×640	5	733k / 393k	8.1k / 5.8k	13.6k / 8k
		10	555k / 303k	5.2k / 4.1k	8.2k / 4.5k
		15	478k / 259k	4k / 3.3k	6.1k / 3.3k
Melbourne	480×640	5	95k / 76k	8.1k / 5.8k	13.6k / 8k
Oxford RobotCar	1280×960	5	3.6k / 3.3k	4k / 3.7k	3.6k / 3.9k

4.4.2 Results and Discussion

Table 4.2: Evaluation of sequential descriptors and sequence matching. SL stands for Sequence Length, CAT indicates concatenation of descriptors, FC stands for Fully Connected layer. ** denotes a non-trained method. * refers to a single descriptor.

Category	Method	Backbone	Descriptor Dimension	GPU Memory Occupation (GB)	Train on Melbourne (R@1)	Train on MSLS (R@1)	Train/Test on Oxf.RobotCar (R@1)
Sequence Matching	SeqSLAM** [174]	VGG-16	4096*	2.68	45.9	45.9	34.7
	HVPR [92]	VGG-16	4096*	2.68	51.0	-	56.8
	SeqMatchNet[93]	VGG-16	4096*	2.68	44.8	-	51.9
Late Fusion	Delta Descriptors**[94]	VGG-16	4096*	2.68	43.0	43.0	18.0
	GeM + CAT [289]	ResNet-18	1280	2.04	66.7	76.8	75.4
	GeM + CAT [289]	ResNet-50	5120	2.25	63.4	68.6	81.3
	NetVLAD + CAT[289]	ResNet-18	16384 · SL	2.04	74.3	84.3	80.3
	NetVLAD + CAT[289]	ResNet-50	65536 · SL	2.26	73.3	85.6	92.1
	NetVLAD + CAT + PCA	ResNet-18	4096	2.04	75.5	83.7	67.8
	NetVLAD + CAT + PCA	ResNet-50	4096	2.26	74.9	85.3	89.3
	CLS + CAT	ViT	768 · SL	2.19	68.2	78.1	69.8
	SeqPool + CAT	CCT224	384 · SL	1.91	65.7	74.2	67.6
	SeqPool + CAT	CCT384	384 · SL	2.01	69.9	77.8	77.4
Intermediate Fusion	SeqNet [92]	VGG-16	4096	2.68	50.1	-	60.5
	SeqNet [92]	ResNet-18	4096	2.04	45.2	-	31.3
	SeqNet [92]	ResNet-50	4096	2.26	45.6	-	61.3
	SeqNet [92]	CCT224	4096	1.91	45.6	-	41.3
	SeqNet [92]	CCT384	4096	2.02	45.4	-	58.8
	CLS + FC	ViT	4096	2.25	66.8	78.1	67.8
	NetVLAD + FC	ResNet-18	4096	3.39	55.5	68.5	44.7
	NetVLAD + FC	ResNet-50	4096	7.63	-	-	-
	SeqVLAD + PCA	ResNet-18	4096	2.04	78.2	85.5	86.5
	SeqVLAD + PCA	ResNet-50	4096	2.26	74.5	85.2	85.9
	SeqVLAD + PCA	VGG-16	4096	2.68	78.5	85.8	79.8
	SeqVLAD + PCA	ViT	4096	2.19	71.9	84.0	67.7
	SeqVLAD + PCA	CCT224	4096	1.91	75.8	85.5	69.6
	SeqVLAD	CCT384	24576	2.02	**81.7**	**89.4**	92.8
	SeqVLAD + PCA	CCT384	4096	2.02	81.4	89.2	**93.3**
Early Fusion	**ResNet-50 2D +1**	-	1024	3.05	63.9	75.2	80.9
	TimeSformer	-	768	2.34	73.8	81.5	74.9

Sequential descriptors vs. sequence matching. Table 4.2 and Fig. 4.4 compare all methods on MSLS and Oxford-RobotCar. Sequence matching approaches consistently underperform compared to sequential descriptors. On Melbourne,

HVPR [92] achieves the highest recall among sequence matching methods (51.0% R@1), while CCT384+SeqVLAD, the best sequential descriptor, reaches 81.7%, a 30% improvement. Sequence matching methods also face scalability issues with large datasets like MSLS, as indicated by missing entries ("-") in Tab. 4.2.

Train-time scalability. Table 4.2 demonstrates that training on the full MSLS dataset, rather than the Melbourne subset, improves test set performance by 9% on average, highlighting the importance of scalability in sequence VPR training.

FC vs. CAT. Simple concatenation (CAT) outperforms learned linear projections (FC), likely due to FC's susceptibility to overfitting with high-dimensional descriptors like NetVLAD. This issue is exacerbated on smaller datasets like RobotCar. FC layers also become computationally prohibitive as descriptor size grows—e.g., ResNet-50 + NetVLAD + FC demands 240 GB of GPU memory.

SeqVLAD. SeqVLAD consistently achieves superior performance across datasets and backbones, validating its effectiveness as a lightweight, sequence-aware layer. PCA further reduces descriptor dimensionality with minimal recall loss, enabling SeqVLAD to outperform prior work with compact descriptors.

Early fusion. TimeSformer, the top early-fusion method, produces compact descriptors but struggles on RobotCar due to limited data and low resolution (224×224). ResNet-50 2D+1 outperforms it on RobotCar, likely due to CNNs' data efficiency. Despite these challenges, TimeSformer's strong results with low-resolution inputs suggest promise for future work.

Backbone. Backbone choice significantly impacts performance, with efficient architectures like ResNets [107] and CCTs delivering the best results. This analysis reveals that costly backbones like VGG-16 [245], common in prior work, are unjustified by their performance.

Image resolution. Low-resolution (224×224) methods suffer recall drops under domain shifts, as seen on RobotCar's day/night test set. Transformers' data hunger may also contribute, as they lack the inductive biases of CNNs.

Precision-Recall curves. Figure 4.4 displays precision-recall curves for select methods from Tab. 4.2. CCT384+SeqVLAD and TimeSformer generally achieve the highest curves, while lower-recall methods like HVPR exhibit interesting trends.

Test-time GPU memory. For real-world applications, GPU memory constraints are critical, especially on embedded systems. Figure 4.5 shows that SeqVLAD with PCA and CCT backbones offers the best trade-off: CCT224 is lighter, while CCT384 achieves higher recalls. For instance, CCT384+SeqVLAD+PCA attains 89.2% R@1 using only 2.02 GB of GPU memory, underscoring the potential of CCT and SeqVLAD.

Inference time. In robotics, inference speed is crucial. Sequence-based VPR involves two steps: descriptor extraction (GPU) and matching (CPU). Extraction

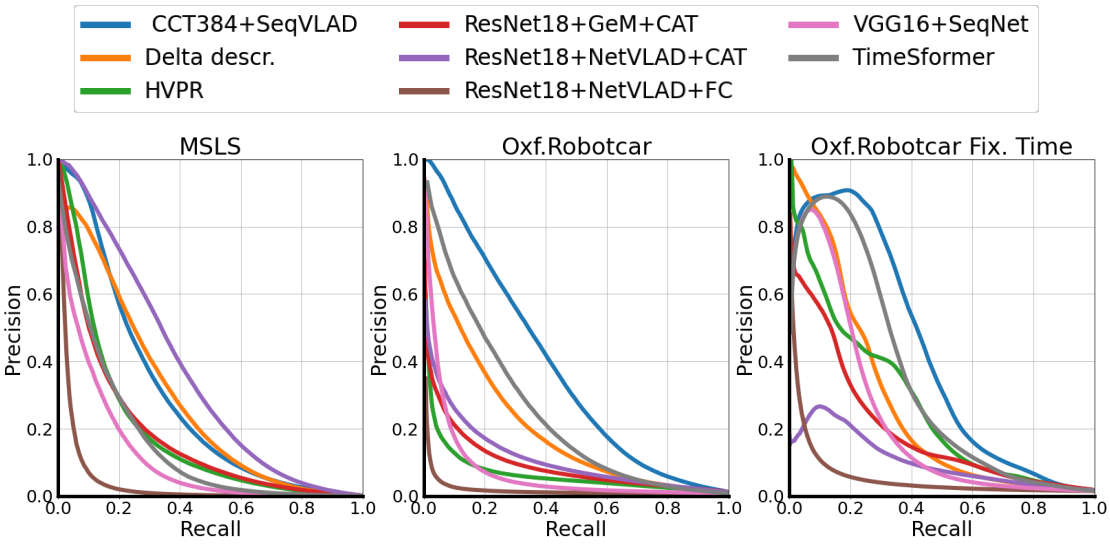


Figure 4.4: Precision-recall curves for main methods, computed with SL = 5.

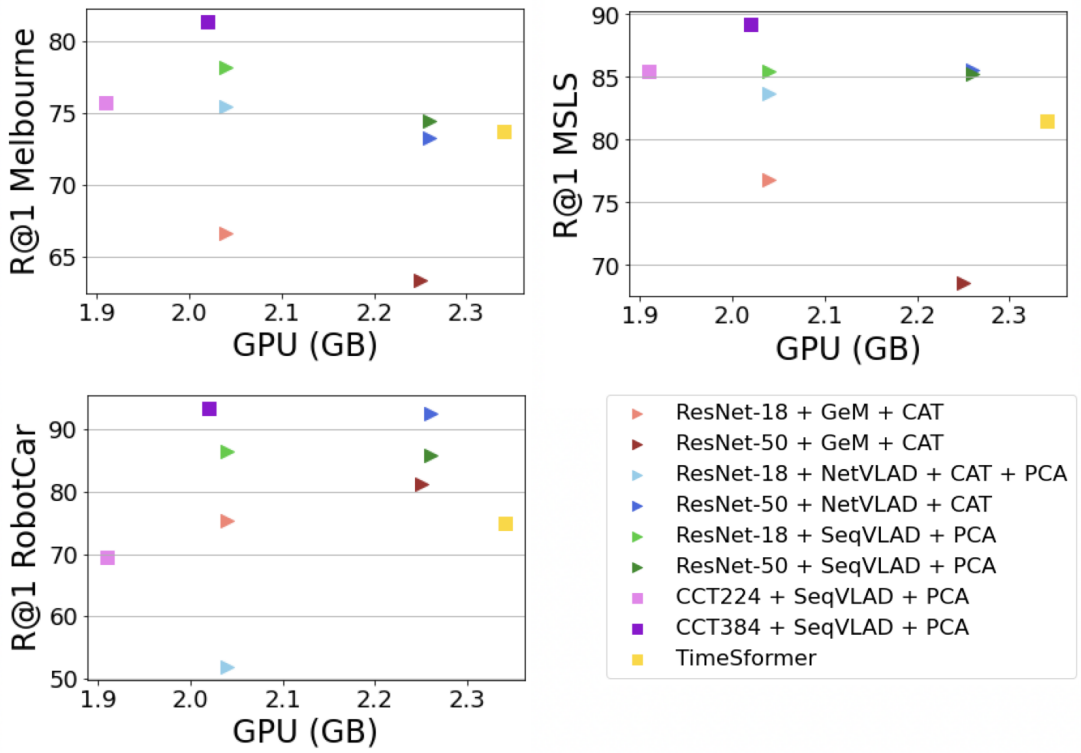


Figure 4.5: **Trade-off between GPU memory and recall@1** for best-performing methods

time depends on the backbone, while fusion adds negligible overhead. Figure 4.6 reports extraction times on an NVIDIA RTX 3090, with top backbones requiring

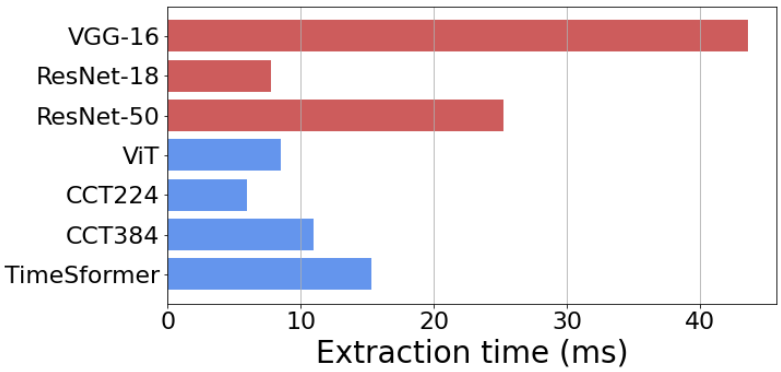


Figure 4.6: **Extraction time with different backbones** for a sequence of 5 frames. Red and blue bars refer to CNNs and Transformers, respectively. Measured on a NVIDIA Geforce RTX 3090

Table 4.3: Robustness to frame sampling at a fixed time (every 3.5s) on Oxford RobotCar.

Method	SeqNet	HVPR	SeqMatchNet	R18+GeM+CAT	R18+NetVLAD+CAT	CCT384+SeqPool+CAT	R50+SeqVLAD	CCT384+SeqVLAD	TimeSformer
R@1	59.3	59.4	59.1	63.6	79.0	76.3	83.2	**84.1**	62.3

10 ms per sequence. Matching time scales linearly with database size. Sequence matching methods, which compute per-frame distances, have complexity $O(N_{db} \cdot SL \cdot D)$, whereas sequential descriptors rely on kNN ($O(N_{db} \cdot D)$). HVPR [92] accelerates matching via kNN-based shortlisting ($O(N_{db} \cdot D + K \cdot SL)$). For example, kNN on 50k descriptors (d=4096) takes 10 ms on an i9-10940X CPU. Scaling to larger databases makes extraction time negligible, and approximate kNN methods [248, 16, 166] can further optimize matching.

Memory footprint. Efficient retrieval requires storing all database descriptors in RAM, consuming $N_{db} \times D$ memory. Thus, descriptor dimensionality (see Tab. 4.2) is as critical as recall, impacting both memory and matching time.

Varying camera speed. Results in Tab. 4.2 use fixed-distance sampling. To assess robustness to varying camera speeds, additional experiments are conducted with sequences sampled at fixed time intervals (3.5s), introducing speed variations. Table 4.3 shows consistent results, with SeqVLAD maintaining its superiority.

Frame ordering. To determine if sequential descriptors depend on frame order, database sequences in the MSLS test set are inverted. Table 4.4 shows that sequence matching suffers due to its assumption of fixed direction/speed. Late fusion (e.g., CAT) also declines, as it encodes frame order. Intermediate fusion, particularly SeqVLAD, proves more robust, capturing order-invariant representations. TimeSformer, while order-agnostic, underperforms SeqVLAD.

Robustness to sequence length variations. For practical deployment, descriptors should handle variable-length sequences without architectural changes.

Table 4.4: Robustness to the inversion of the frames.

Method	Backbone	Dim.	Forw.	Back.	Diff
SeqSLAM* [174]	VGG-16	4096	45.9	22.9	-50 %
Delta D.*[94]	VGG-16	4096	43.0	11.7	-73 %
HVPR [92]	VGG-16	4096	51.0	28.5	-44 %
SeqMatchNet [93]	VGG-16	4096	44.8	24.2	-46 %
GeM + CAT [289]	ResNet-18	1280	76.8	67.8	-12 %
NetVLAD + CAT + PCA	ResNet-18	4096	83.7	79.9	-5 %
SeqNet [92]	VGG-16	4096	50.1	42.0	-16 %
NetVLAD + FC	ResNet-18	4096	68.5	65.1	-5 %
SeqVLAD + PCA	ResNet-18	4096	85.5	85.2	-0.4 %
SeqVLAD + PCA	CCT384	4096	89.2	89.2	**0.0** %
TimeSformer	-	768	81.5	81.5	**0.0** %

Table 4.5: Flexibility to variations in test-time sequence length (SL).

Method	Backbone	Descr. Dim.	SL 1	SL 3	SL 5	SL 10	SL 15
GeM + CAT [289]	ResNet-18	256 · SL	63.7	74.6	76.8	79.8	82.2
GeM + CAT [289]	ResNet-50	1024 · SL	59.6	67.3	68.6	70.4	72.2
NetVLAD + CAT [289]	ResNet-18	16384 · SL	74.4	82.2	84.3	86.2	86.9
NetVLAD + CAT [289]	ResNet-50	65536 · SL	76.0	83.1	85.6	88.6	90.2
SeqPool + CAT	CCT224	384 · SL	58.4	72.7	74.2	76.2	75.8
SeqPool + CAT	CCT384	384 · SL	62.2	74.6	77.8	79.9	78.8
SeqVLAD + PCA	ResNet-18	4096	75.0	83.7	85.5	89.3	92.7
SeqVLAD + PCA	ResNet-50	4096	74.2	83.6	85.2	88.7	91.8
SeqVLAD + PCA	CCT224	4096	72.2	83.5	85.5	87.8	91.2
SeqVLAD + PCA	CCT384	4096	**78.2**	**87.6**	**89.2**	**92.1**	**95.2**
TimeSformer	-	768	62.2	78.8	81.5	86.3	89.9

Table 4.5 evaluates CAT, SeqVLAD, and TimeSformer on sequences of lengths differing from the training length (5 frames). TimeSformer and SeqVLAD produce fixed-dimensional descriptors, enabling direct comparison across lengths, while CAT's dimensionality varies with sequence length. Longer sequences generally improve recall, as they (i) provide more informative descriptors and (ii) increase the likelihood of frame overlap in seq2seq tasks [289].

4.5 Conclusions

This chapter explores the field of sequential descriptors for VPR, offering the first taxonomy and an extensive experimental comparison of these methods. Through this analysis, we highlight the strengths and weaknesses of various architectural

approaches. We also contribute with additional solutions, demonstrating the effectiveness of Transformers for this task and presenting a layer, SeqVLAD, that can be integrated with both types of backbones. SeqVLAD establishes a state-of-the-art benchmark while reducing computational cost and inference time compared to earlier methods. Despite the extensive experiments, the analysis has certain limitations that we aim to address in future work: i) the current evaluation focuses solely on outdoor sequences captured on roads, and we intend to expand this to include indoor random paths; ii) the robustness of these architectures to variations in camera speed remains uncertain. While some preliminary findings are provided in Tab. 4.3, we aim to examine this issue more comprehensively in future studies.

Chapter 5

Are Local Features All You Need for Cross-Domain Visual Place Recognition?

Visual Place Recognition (VPR) is conventionally tackled as a retrieval problems, where global descriptors are employed for coarse localization. A predominant challenge that VPR systems face is the fact that queries often diverge substantially from reference data (typically made up of daytime Street View imagery). Examples are nighttime captures, occlusions, or adverse weather. This chapter investigates the underutilized potential of image matching techniques as a re-ranking mechanism in VPR pipelines. While prior work has demonstrated the inherent robustness of local feature methods and matching algorithms to domain shifts, their systematic application to VPR remains limited. Existing studies suffer from inconsistent evaluation protocols, where variations in retrieval backbones mislead the assessment of re-ranking efficacy.

To address these limitations, we present a controlled benchmarking framework that isolates the contribution of re-ranking strategies by providing all methods with identical candidate pools. Our evaluation reveals that state-of-the-art retrieval combined with optimal re-ranking can achieve near-human performance on challenging benchmarks like Tokyo-night, reducing error rates below 5%.

Complementing our benchmarking efforts, this chapter introduces two novel datasets specifically designed to evaluate VPR systems under critical failure conditions: low-light nighttime environments and heavy occlusions from dynamic objects. These manually verified query sets, sourced from Flickr and paired with the comprehensive SF-XL reference database, provide targeted benchmarks for assessing robustness to real-world perceptual challenges. Through these contributions this chapter advances the understanding of image matching's role in building robust VPR systems capable of operating under diverse environmental conditions. Code and datasets have been released at https://github.com/gbarbarani/

re-ranking-for-VPR.

The work described in this chapter has been previously published in the following paper:

- G. Barbarani, M. Mostafa, H. Bayramov, G. Trivigno, G. Berton, C. Masone, B. Caputo. **Are local features all you need for cross-domain visual place recognition?** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition - Image Matching Workshop* (CVPRW) 2023

5.1 Introduction

Visual Place Recognition (VPR) addresses the fundamental question of determining the location where an image was captured. Typically, this task is approached as an image retrieval problem, where a query image is localized by comparing it against a database of georeferenced images [306, 27, 10, 134, 96, 25, 3, 2, 318]. A query is deemed correctly localized if its predicted position lies within 25 meters of the ground truth. As a precursor to more precise visual localization, VPR has diverse applications in autonomous navigation, SLAM systems, and augmented reality. Due to these use cases, the task is predominantly evaluated in large-scale outdoor environments, with databases often compiled automatically using Street View imagery [267, 265, 26, 25], which primarily consists of daytime captures. However, real-world deployments face significant appearance variations in query images, including nighttime conditions, occlusions, and adverse weather, creating a persistent domain gap between queries and reference data [265, 8, 288, 26, 289, 115, 301, 208, 309, 169].

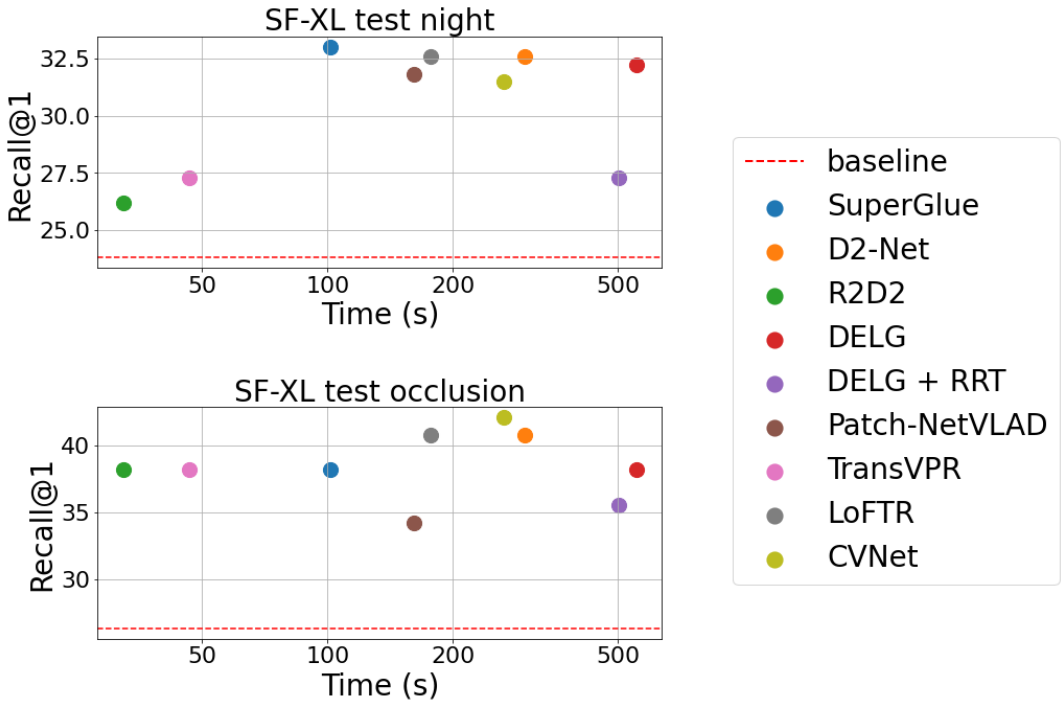


Figure 5.1: **Plot showing the Recall@1 and latency for different methods on multiple datasets.** Latency is to re-rank 100 candidates of a single query, considering local features extraction to be performed online. We can see that there is no single method that outperforms all others on all scenarios, and the ideal choice of a re-ranking method for a VPR system depends on multiple factors, such as time requirements and expected domain shifts.

Prior research has demonstrated that local feature extraction [191, 43, 144, 284] and image matching techniques [220, 74, 254, 226] exhibit inherent robustness to domain shifts. These methods can enhance performance by re-ranking candidate matches (often via spatial verification) initially retrieved through global descriptors, yielding substantial accuracy improvements [105, 284].

This chapter investigates the efficacy of image matching methods for re-ranking top-N candidates in VPR pipelines. Although such techniques have gained attention, their application to VPR remains underexplored, with existing studies limited to narrow comparisons [105, 284]. Moreover, prior evaluations are confounded by inconsistent retrieval backbones—for instance, [284] employs distinct retrieval methods for each re-ranking approach, obscuring whether performance gains stem from re-ranking or retrieval components.

To identify optimal re-ranking strategies for real-world VPR scenarios, we conduct a comprehensive evaluation of image matching pipelines, emphasizing domain shift robustness. Our benchmark ensures fair comparisons by supplying all methods with identical candidate pools. Where feasible, we standardize feature extraction backbones and hardware configurations for consistent efficiency analysis. Results demonstrate that combining state-of-the-art retrieval and re-ranking can nearly solve longstanding benchmarks such as Tokyo-night with an error rate of less than 5%.

To further advance research, we present two challenging datasets targeting critical VPR failure modes: nighttime conditions and heavy occlusions from dynamic objects. Both query sets are sourced from Flickr and manually verified, while the reference database utilizes the San Francisco eXtra Large (SF-XL) dataset.

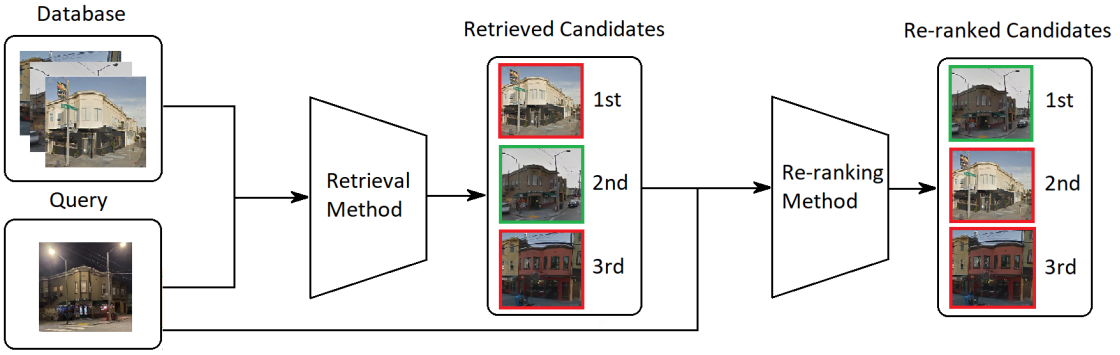


Figure 5.2: **Overview of the re-ranking pipeline.** First, a retrieval method performs a similarity search on the global descriptors extracted from query and database to output a set of top-k candidates. Then re-ranking is applied to refine the retrieved candidates.

Key contributions include:

- Two curated query sets for evaluating VPR under nighttime and occlusion

conditions, collected from Flickr and manually validated against a city-scale database.

- A systematic benchmark assessing spatial verification techniques for VPR re-ranking. Our controlled setups isolate method performance and quantify improvements over existing approaches.
- Empirical evidence that re-ranking significantly boosts retrieval accuracy, with method effectiveness varying across scenarios—highlighting the absence of a universally superior solution.

Database	Database source	Database # images	Query set	Queries source	# queries
Tokyo 24/7 [265]	Google StreetView	75k	Tokyo night (night queries from Tokyo 24/7)	Collected with a smartphone by [265]	105
SVOX [26]	Google StreetView	17k	SVOX Night	Oxford RobotCar	823
SF-XL [25]	Google StreetView	2.8M	SF-XL test v1	Flickr	1000
			SF-XL test v2	Collected with a smartphone by [48]	598
			SF-XL test night (**ours**)	Flickr	466
			SF-XL test occlusion (**ours**)	Flickr	76

Table 5.1: **Summary of the datasets considered in our experiments.** The table reports the raw data sources, every author has cleaned and processed them in customized way, refer to their papers for the details. In general streetview panoramas have been cropped in patches and turned in multiples suitable references for the databases. Flickr queries have been filtered and manually checked for positive references. Oxford RobotCar [164] data have been collected and processed with a modality analogues to streetview panoramas, although Tokyo 24/7 [265] queries have been collected with smartphone devices they are scenes compatible with a moving vehicle point of view. While Flickr and San Francisco Landmark [48] data contains a broader range of point of views and camera types.

5.2 Related Work

Visual Place Recognition through Image Retrieval.

The literature on Visual Place Recognition has already been described extensively in previous chapters of this thesis. This chapter focuses in particular on the issue of domain adaptation in VPR. To this end, [288, 8, 26] addressed this challenge proposing specific domain adaptation techniques. However, they are focused on the specific problem of adapting from a single source to a single target domain, and are thus lacking a study on generalization across data distributions.

Regarding re-ranking techniques for image retrieval, they have not been studied in depth for VPR. Existing VPR benchmarks focus instead on pure retrieval performance [228, 211, 306, 27]. On the other hand, we focus specifically on the opportunities that re-ranking and image matching can provide, while considering they computational trade-offs.

Local features for spatial verification.

Spatial verification based on local features is a well-established approach that has been widely adopted in various computer vision applications, including structure from motion (SfM) [153, 234, 159], simultaneous localization and mapping (SLAM) [226, 105, 61], and visual localization [153, 228, 254, 265]. For a long time, hand-crafted feature detectors and descriptors set a highly competitive standard [161, 22], while early learning-based techniques demonstrated significant potential for improvement, particularly under varying lighting conditions and viewpoints [300, 65]. More recently, there has been a growing body of work on trainable detectors and descriptors that leverage local features for tasks such as pose estimation and homography recovery [65, 62, 74, 220, 226, 254, 125]. While these approaches

are not explicitly tailored for retrieval tasks, they can be naturally adapted to re-rank retrieval candidates by prioritizing image pairs with a higher number of feature correspondences.

Nevertheless, techniques optimized for outdoor image matching may exhibit reduced robustness to dynamic elements (*e.g.* matching vehicles instead of static structures), compared to methods specifically developed for retrieval and re-ranking [105, 43].

Typically, local features consist of keypoint-descriptor pairs, where the keypoint denotes the pixel location and the descriptor is a fixed-length vector. For a given image pair, these features are cross-matched to establish correspondences, a process referred to as spatial verification. While traditional methods often rely on heuristics such as RANSAC [83], data-driven solutions like SuperGlue [226] have been introduced for this purpose.

SuperGlue employs graph neural networks to learn match priors conditioned on input keypoint sets. This framework has been extended by LoFTR [254], which eliminates the need for a separate detector by leveraging cross-attention transformers to directly identify keypoint matches between images.

Several of these techniques have been assessed on outdoor datasets that are not standard in the VPR community. Examples include Oxford5k [206] and Paris6k [205], which are less suitable for VPR due to their sparse coverage. Others, such as Aachen [312, 230, 228], focus on limited urban areas and are primarily intended for pose estimation. Similar limitations apply to datasets like Madrid Metropolis, Gendarmenmarkt, and Tower of London, introduced in [294].

Local features for re-ranking.

Local features have also been investigated for image retrieval, particularly for refining the top-N candidates identified by the retrieval stage [105, 257, 284, 43]. In this context, the matching process must be adapted to produce a similarity score, often based on the count of feature matches between images. Alternatively, some approaches integrate local features directly into global descriptors or employ them for reference image matching, bypassing explicit re-ranking [298, 191, 291, 263]. DELG [43] combines a global descriptor trained with a large margin cosine loss with locally refined features, selected based on unsupervised criteria for distinctiveness and reliability. Patch-NetVLAD [105] generates dense local features by applying VLAD aggregation to image patches, whereas TransVPR [284] employs a transformer architecture to select relevant patches via multi-scale attention mechanisms—both designed specifically for VPR. Most re-ranking techniques rely on RANSAC, using inlier counts to score candidates [191, 43, 105, 284].

Recent work has explored end-to-end trainable models that predict pairwise image similarity as an alternative to RANSAC. For instance, [257] processes local and global features from DELG using a transformer, framing the task as binary classification. Similarly, CVNet [144] constructs a hierarchy of 4D correlation maps

from CNN feature maps, which are then condensed into a similarity score via 4D convolutions. Unlike homography estimation methods, the approaches discussed in this section are trained without requiring patch-level annotations.

5.3 Dataset

Previous datasets.

Various datasets featuring query splits specifically designed to evaluate domain shift adaptation have been introduced in the VPR literature [265, 26, 164, 25].

One notable example is Tokyo 24/7 [265], which includes three distinct query sets captured at different times of day—daytime, sunset, and night—making it a commonly utilized benchmark for cross-domain VPR studies. Although Tokyo 24/7 offers a carefully curated dataset and has been extensively referenced in prior work [10, 158, 96, 156], its limited scale becomes apparent: with only 105 queries per domain and a database of 75k images, it covers just a small portion of the city.

The work by [26] describes SVOX, another cross-domain dataset where the database consists of Google StreetView imagery. The queries, sourced from RobotCar [164], span multiple domains including snow, rain, sun, night, and overcast conditions. However, SVOX's database is even smaller than Tokyo 24/7's, and it exclusively includes frontal-view images, meaning the camera faces directly along the road.

San Francisco eXtra Large (SF-XL) [25] represents another cross-domain dataset, featuring a large-scale database that encompasses the entire city of San Francisco with 2.8M StreetView images. SF-XL contains two query sets, referred to as *test v1* and *test v2*: (1) *test v1* comprises 1000 Flickr images distributed uniformly across the city, exhibiting some domain variations (primarily viewpoint changes and a small number of night images); (2) *test v2* consists of 598 smartphone-captured images from the city center, displaying mild to moderate viewpoint differences relative to the database. Despite SF-XL's relevance due to its large scale, the absence of clearly defined query splits across multiple domains complicates the assessment of method performance under specific domain shifts.

To address these limitations, this chapter introduces two additional sets of queries—night and occluded images—intended for use with the SF-XL database.

Datasets presented in this chapter. In order to gather realistic and varied queries, we collected hundreds of thousands of Flickr images covering the San Francisco area, following a methodology similar to [25, 216, 205, 206].

Using trained classifiers, we filtered out indoor images and subsequently generated two challenging query sets:

- *SF-XL test night* contains nighttime images automatically identified by a trained classifier.
- *SF-XL test occlusion* comprises images with significant occlusions, selected via an object detection model that retained those with dynamic objects (*e.g.* cars, trucks, people) occupying more than 50% of the image width and 30% of the height.

Figure 5.3: **Examples of queries from a) SF-Night and b) Sf-occlusion.**

Due to inaccuracies in Flickr's geotagging, we manually verified the location of each image, resulting in 466 and 76 images for *SF test night* and *SF test Occlusion*, respectively. Example queries are illustrated in Fig. 5.3.

Given the existence of an open-source large-scale dataset covering San Francisco—SF-XL [25]—we aligned our query sets with this database, effectively using SF-XL as the reference dataset.

A summary of the datasets utilized in our experiments is provided in Tab. 5.1.

Retrieval Method	Descriptors Dimension	SF-XL test v1		
		R@1	R@5	R@10
NetVLAD	4096	33.1	45.0	50.4
TransVPR	256	9.7	16.6	20.3
CVNet	2048	70.1	81.2	84.6
DELG	2048	64.3	73.0	76.1
CosPlace	512	**76.7**	**82.5**	**85.6**
Conv-AP	4096	49.1	60.6	65.6
MixVPR	4096	72.3	79.5	81.4

Table 5.2: **Recalls with different retrieval methods.** We used only global descriptors for this table (*i.e.* no re-ranking is applied to DELG and CVNet). NetVLAD uses a VGG-16 [245] (and PCA), TransVPR a custom transformer model, while for all other methods we used the author's ResNet-50 [107] implementation.

5.4 Experiments

Features Extractor	Features Matching	Tokyo night R@100 = 96.2			SVOX night R@100 = 90.3			SF-XL test v1 R@100 = 92.5			SF-XL test v2 R@100 = 97.7			SF-XL test night R@100 = 41.6			SF-XL test occlusion R@100 = 60.5		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
-	-	80.0	88.6	91.4	51.6	68.8	76.1	76.7	82.5	85.6	89.0	95.3	96.3	23.8	29.0	31.5	26.3	38.2	46.1
SuperPoint	SuperGlue	**95.2**	95.2	95.2	77.9	**85.2**	**86.5**	**88.6**	**91.6**	**91.9**	92.8	96.7	**97.7**	**33.0**	38.0	39.1	38.2	44.7	50.0
D2-net	RANSAC	92.4	**96.2**	**96.2**	78.9	85.1	86.4	87.5	90.3	90.8	94.0	96.3	97.0	32.6	38.2	39.5	40.8	48.7	51.3
R2D2	RANSAC	86.7	90.5	92.4	72.5	80.7	82.9	85.1	88.2	89.6	**94.1**	**96.8**	96.8	26.2	32.2	33.9	38.2	47.4	50.0
DELG	RANSAC	94.3	95.2	**96.2**	**80.1**	84.1	86.0	88.5	91.2	91.5	93.8	96.2	97.0	32.2	37.6	39.2	38.2	50.0	53.9
DELG	RRT	84.8	94.3	95.2	66.3	81.7	85.7	85.3	89.6	90.4	88.6	96.0	97.2	27.3	35.6	38.6	35.5	48.7	52.6
Patch-NetVLAD	RANSAC	90.5	94.3	94.3	67.2	80.6	83.6	77.0	84.7	87.0	91.0	95.2	96.2	31.8	37.3	38.4	34.2	47.4	52.6
Patch-NetVLAD	Rapid Scoring	73.3	87.6	92.4	42.2	66.3	73.1	69.3	80.3	84.1	90.0	94.6	95.8	21.7	31.3	35.4	25.0	38.2	42.1
TransVPR	RANSAC	88.6	95.2	95.2	63.8	79.2	83.2	84.0	87.6	89.1	92.5	96.2	96.7	27.3	34.3	36.7	38.2	46.1	52.6
	LoFTR	93.3	95.2	95.2	80.0	84.0	85.3	87.9	89.8	90.7	93.3	96.3	97.2	32.6	37.6	38.2	40.8	48.7	51.3
	CVNet	94.3	**96.2**	**96.2**	74.6	**85.2**	**86.5**	84.8	91.0	91.6	88.0	95.8	97.0	31.5	**39.3**	**39.9**	**42.1**	**52.6**	**56.6**

Table 5.3: **Recalls before and after applying re-ranking.** The shortlist of candidates to be re-ranked is obtained with CosPlace, and the results with such shortlist are shown in the first row. Re-ranking has been applied to the first 100 candidates (*i.e.* $K = 100$). Next to each dataset's name, we show the R@100, which in practice sets the upper bound of the maximum recalls achievable after re-ranking. Best results are in **bold**, second best are underlined.

5.4.1 Benchmark Methodology

The need for this benchmark becomes evident when considering how spatial verification techniques are typically applied to visual place recognition. Many existing

Model	Descriptors size (num. × dim.)	Backbone	Designed for re-ranking	Sparse Keypoints
DELG	1000 x 128	ResNet-50	✓	✓
Patch-NetVLAD	2826 x 4096	VGG-16	✓	✗
TransVPR	522 x 256	Custom CNN+transformer	✓	✓
R2D2	4126 x 128	custom L2-Net[262]	✗	✓
D2Net	2775 x 512	VGG-16	✗	✓
SuperPoint	1034 x 256	custom VGG	✗	✓

Table 5.4: **Characteristics of local features extractors.** The descriptors size was computed for all methods on the same image of resolution 480x640. For Patch-NetVLAD descriptors size depends only on the resolution, because it uses dense keypoints/features, whereas for all other methods the number of descriptors depends on the visual content of the image.

approaches [226, 254] rely on training data derived from 3D structure-from-motion models [153], which provide precise matching labels. However, in place recognition scenarios, matches are defined more loosely (within 25 meters [10, 27, 211]), meaning positive matches might only share limited scene overlap. This discrepancy raises questions about how well these methods handle significant viewpoint changes and transient objects. This chapter examines these previously unaddressed research aspects.

Our experimental framework follows a two-stage process (illustrated in Fig. 5.2):

1. initial candidate selection using global descriptor methods (identifying K nearest neighbors in feature space);
2. reordering these candidates through a re-ranking algorithm.

While extensive research exists on global descriptor-based image retrieval for VPR [10, 306, 27, 25, 2, 3], our focus lies on the re-ranking phase. For initial candidate selection, we employ CosPlace [25] (with ResNet-50 backbone), which demonstrates superior performance on SF-XL test v1 (see Tab. 5.2). We then evaluate multiple re-ranking approaches: SuperGlue, D2-Net, R2D2, DELG, Patch-NetVLAD, TransVPR, LoFTR and CVNet. Using identical candidate sets enables us to isolate the impact of local features from global descriptor performance, providing clear insights into the potential benefits of spatial verification when integrated into existing VPR systems.

The choice of K significantly impacts performance - higher values may improve accuracy but reduce speed (analyzed in Sec. 5.4.4). Following standard practice [105, 284, 43, 257], we set $K = 100$ for primary experiments, with additional analysis of varying K values.

Consistent with VPR literature [10, 96, 158, 318, 25, 27], we employ Recall@N (R@N) as our evaluation metric, measuring the percentage of queries with at least

one correct match among the top N predictions (within 25 meters). We also examine performance at 50 and 100 meter thresholds in Fig. 5.4. While positive matches might lack visual overlap with queries (e.g., opposite viewpoints within 25 meters), the likelihood of random matches decreases with database size, and obtaining unbiased covisibility ground truth would otherwise require relying on tested methods themselves.

5.4.2 Implementation details

Our comprehensive evaluation incorporates multiple approaches, including methods designed for re-ranking alongside those developed for spatial verification and image matching. Specifically, we examine SuperGlue [226] (using SuperPoint [62] features), D2-Net [74], R2D2 [220], DELG [43], Reranking Transformers (RRT) [257] (with DELG features), Patch-NetVLAD [105] (both RANSAC and Rapid Scoring variants), TransVPR [284], LoFTR [254] and CVNet [144].

All methods utilize official implementations with author-provided weights, without fine-tuning. For configurations offering multiple options, we selected those achieving best performance on *SF-XL test v2*, preferring ResNet50 over ResNet100 backbones where applicable.

Several approaches employ multi-scale processing, including DELG, R2D2, D2-Net, Patch-NetVLAD and CVNet. Most spatial verification methods use standard 8-parameter homography RANSAC, except DELG which employs 6-parameter affine RANSAC. Patch-NetVLAD applies RANSAC at multiple scales and offers Rapid Scoring as a faster, non-iterative alternative to RANSAC. Preliminary tests showed Rapid Scoring performed poorly outside its intended context, being particularly sensitive to outliers and effective only for minor viewpoint changes. Following original implementations, we resize images to 480x640 resolution for Patch-NetVLAD and TransVPR.

5.4.3 Quantitative evaluations of results

Tab. 5.3 presents our experimental outcomes. Most methods demonstrate performance improvements over the baseline, with LoFTR and SuperGlue showing average boosts of 21% and 13% respectively on night and occlusion benchmarks. This supports our primary motivation - demonstrating local feature methods' potential to address global descriptors' limitations in challenging conditions. Key observations include:

- Image matching methods prove surprisingly effective for VPR. SuperGlue achieves top recalls on *Tokyo night*, *SF-XL test v1* and *SF-XL test night*, despite lacking explicit training for transient object rejection. D2-Net and LoFTR show comparable performance across datasets.

- Among dedicated re-ranking approaches, DELG with RANSAC shows strongest versatility, achieving best performance on *SVOX night* and competitive R@1 elsewhere. CVNet excels in R@5 and R@10 metrics despite some R@1 drops, while TransVPR and Patch-NetVLAD show limited robustness to nighttime conditions.
- CVNet demonstrates particular effectiveness on *SF-XL test occlusion*, validating its Hide-and-Seek [247] augmentation strategy for occlusion handling.
- Rapid scoring delivers modest results, failing as a viable RANSAC alternative. Similarly, RRT scoring improves over baseline but underperforms RANSAC variants on night datasets.
- The presented datasets remain far from solved, offering substantial challenges that could drive future research directions.

5.4.4 Ablation on K and different positive threshold distances

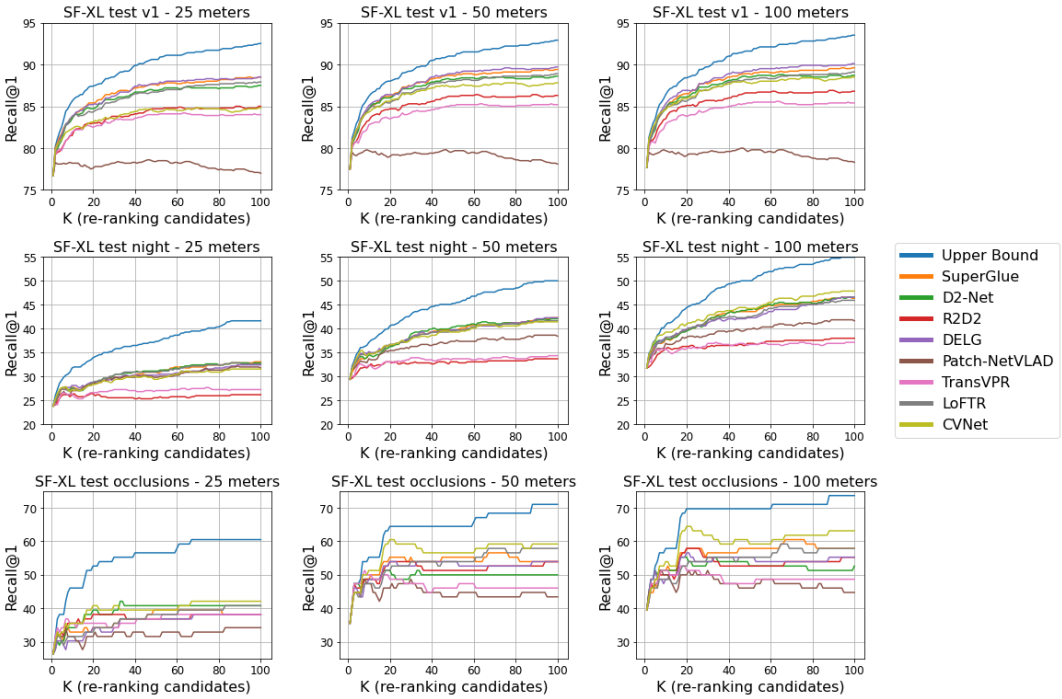


Figure 5.4: **Re-ranking with different values of K, from 1 to 100.** The "Upper Bound" is the Recall@K without applying re-ranking (*i.e.* with CosPlace). For DELG and Patch-NetVLAD we used the version with RANSAC.

Given re-ranking's computational expense compared to standard retrieval, understanding optimal candidate numbers is crucial. We evaluate top-performing methods from Tab. 5.3 across $K \in \{1, 2, 3, ...100\}$, examining Recall@1 at 25, 50 and 100 meter thresholds. We focus on *SF-XL test v1* and *SF-XL test night* as representative real-world scenarios.

Key findings include:

- Increasing the threshold to 100m yields more correct matches, particularly beneficial for applications not requiring precise localization. More challenging datasets show greater improvements.
- DELG and CVNet excel at 100m thresholds, likely due to training on Google Landmark Dataset's distant building photos. DELG performs best on *SF-XL test v1* while CVNet handles night conditions and occlusions better.
- D2-Net, SuperGlue and DELG provide most precise matches under domain shifts, achieving top scores at 25m on *SF-XL test night*.
- While higher K generally improves results, gains diminish earlier on challenging datasets, especially at lower thresholds. *SF-XL test occlusion* shows increased false positives with larger K, necessitating careful parameter selection given linear computational scaling.
- The performance gap between re-ranking methods and CosPlace's Recall@K indicates substantial room for improvement.

5.4.5 Qualitative evaluations of results

To illustrate strengths and weaknesses of SuperGlue, DELG and CVNet, Fig. 5.5 shows example queries with each method's top predictions.

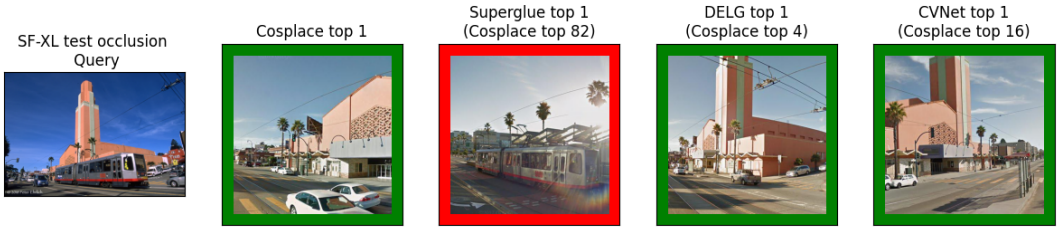
5.4.6 Is the night domain a real challenge?

We investigate whether nighttime errors stem primarily from illumination or other factors. Analyzing 101 *Tokyo night* queries where CosPlace provides at least one positive match in the top 100 candidates, CVNet succeeds on all cases while DELG and SuperGlue fail once (shown in Fig. 5.5 (d)). This suggests *SF-XL test night*'s difficulty extends beyond illumination, incorporating realistic challenges like viewpoint shifts and artificial lighting effects (see Fig. 5.3). These results demonstrate state-of-the-art local features' robustness to illumination changes, while highlighting the complex challenges present in real-world photography.

5.4.7 Computational cost

Fig. 5.1 analyzes computational requirements versus performance for re-ranking methods on *SF-XL test night* and *SF-XL test occlusion*. We measure time to re-rank top-100 candidates per query, including online local feature extraction. While some works assume offline extraction, storing features like SuperPoint for SF-XL would require approximately 1TB without quantization. Since quantization requires method-specific optimization [191, 27], we maintain general applicability with online extraction.

Lighter backbones (R2D2, TransVPR) show fastest performance, while DELG (with RANSAC or RRT) proves most computationally intensive. Considering performance, SuperGlue, LoFTR and CVNet offer best trade-offs. However, hundred-second delays may prove impractical for many applications, necessitating careful consideration alongside K value analysis in Fig. 5.4. While common practice uses $K = 100$ or more [191, 105, 284, 43, 257], substantial speed improvements can often be achieved by reducing K with minimal accuracy loss.



(a) A failure of SuperGlue due to a dynamic object (a tram), which SuperGlue (unlike DELG and CVNet) has not been trained to ignore. We can also see that CVNet finds a positive with very different viewpoint than the query, even though candidates closer to the query are available.

(b) DELG and CVNet failures for this case are most likely due to those methods using a combination of local and global scoring system. The global features see trees and a red line (which for the query is on the bus, and for the predictions is an awning).

(c) This example shows the robustness of CVNet to strong occlusions, which is learned thanks to its use of Hide-and-Seek data augmentation [247].

(d) The only example from *Tokyo night* where DELG and SuperGlue fail to find the correct prediction.

Figure 5.5: **Qualitative examples of 3 queries and the first prediction with 4 relevant methods**, namely CosPlace (retrieval baseline) SuperGlue, DELG and CVNet. Predictions are in green if they are less than 100 meters away from the query's ground truth.

5.5 Conclusion

In this chapter, we examined whether re-ranking techniques can enhance results in visual place recognition, and analyze the trade-offs they entail. In particular, we focus on the scenario where the queries originate from a domain distinct from the database, emphasizing cases involving nighttime and occluded queries. We presented two challenging query sets, demonstrating that even the highest performing methods struggle in this scenario, achieving low recall scores.

Extensive experiments were conducted to identify which approaches perform best for cross-domain re-ranking in VPR, revealing that numerous methods provide pareto-optimal solutions when taking into account for evaluation both inference time and recall. Additionally, we observe that different types of domain shifts demand tailored strategies, and no single method consistently outperforms others across all datasets, even when latency constraints are disregarded.

Overall, this chapter offers insights into designing highly effective VPR systems under diverse conditions, and we believe that the datasets introduced here will encourage further advancements in the field to push the boundaries of the state of the art.

Chapter 6

Divide&Classify: Fine-Grained Classification for City-Wide Visual Place Recognition

Visual Place Recognition (VPR) addresses the fundamental challenge of determining a photograph's capture location with meter-level precision, typically framed as an image retrieval problem where queries are matched against geo-tagged reference databases. While retrieval-based approaches have proven effective for small-scale environments, their application to city-wide deployments faces significant scalability limitations due to computational and memory requirements that grow proportionally with database size. This chapter explores an alternative paradigm by formulating VPR as a classification problem, offering the potential for constant-time inference regardless of map scale.

Current solutions to mitigate retrieval's scaling challenges, such as Approximate Nearest Neighbor (ANN) methods, achieve computational efficiency but at the cost of reduced localization accuracy. As demonstrated in our experiments, these approaches exhibit an inherent trade-off between performance and scalability that becomes particularly pronounced in dense urban environments covering hundreds of square kilometers. While classification-based methods have shown promise for coarse global-scale geo-localization, existing techniques prove inadequate for city-scale VPR due to their inability to handle fine-grained spatial distinctions between visually similar locations.

This chapter presents Divide&Classify (D&C), a novel classification framework specifically designed for dense urban environments that bridges the gap between classification efficiency and retrieval-level accuracy. Our approach introduces several key innovations: an Additive Angular Margin Classifier (AAMC) that learns discriminative location prototypes through angular margin optimization, and a hybrid pipeline that strategically combines classification predictions with targeted

retrieval to achieve superior performance. The proposed system maintains constant inference times regardless of map scale while delivering localization precision comparable to state-of-the-art retrieval methods.

Through extensive experimentation, we demonstrate that D&C not only overcomes the limitations of existing classification approaches for fine-grained VPR but also enables new operational paradigms. The framework's ability to constrain search spaces for subsequent retrieval stages yields a synergistic combination of classification speed and retrieval accuracy, establishing a new direction for scalable urban-scale localization systems. Code and models have been released at https://github.com/ga1i13o/Divide-and-Classify.

Part of the work described in this chapter has been previously published in the following paper:

- G. Trivigno (*), G. Berton(*), J. Aragon, B. Caputo, C. Masone. **Divide&Classify: Fine-Grained Classification for City-Wide Visual Geo-Localization** In *IEEE/CVF International Conference on Computer Vision* (ICCV) 2023;

6.1 Introduction

Visual Place Recognition (VPR) refers to identifying the location where a photograph was captured, with an accuracy typically within a few meters [267, 10, 134, 265, 158, 67, 96, 26, 115, 289, 170, 105, 266, 306, 54, 53, 90, 133, 104]. Also known as visual geo-localization [25, 27, 134] or image localization [158, 96], this problem is frequently framed as an image retrieval task: a query image is matched against a database of geo-tagged images using k-nearest neighbor (kNN) in feature space, with the retrieved images serving as predictions for the query's location. While retrieval techniques are highly effective for VPR tasks limited to smaller maps [27, 306, 25], their scalability to large, densely mapped regions, such as an entire city, becomes challenging due to the computational and memory demands that scale with the size of the database [306, 27, 211, 15].

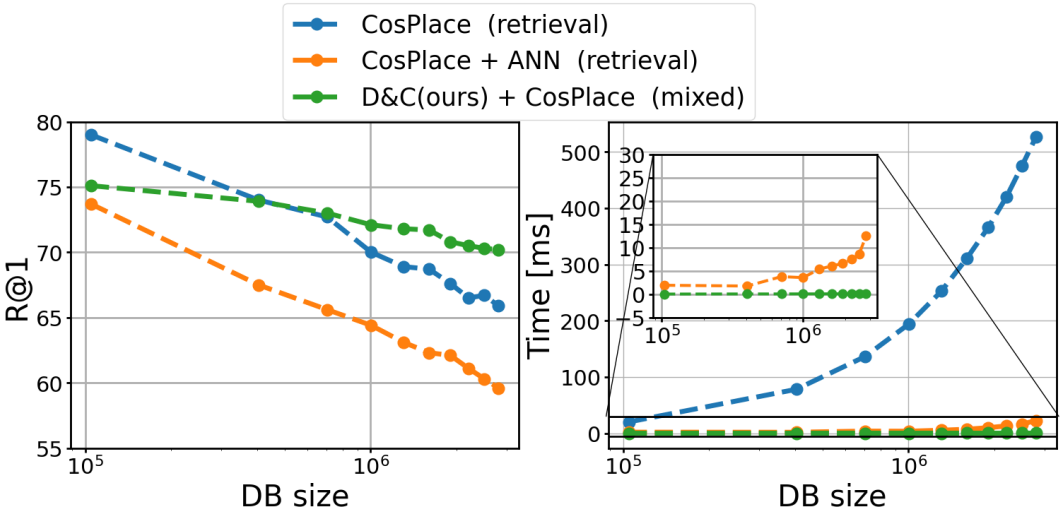


Figure 6.1: **Experiments** demonstrating the scalability problem of retrieval-based VPR methods, using the state-of-the-art CosPlace (with exhaustive kNN and with Approximate Nearest Neighbor - ANN - search through Inverted File Index with Product Quantization, IVFPQ). Combining our Divide&Classify method with the retrieval approach yields an optimal performance-efficiency trade-off when scaling up to the city-wide setting.

The resource requirements of kNN can be mitigated by employing Approximate Nearest Neighbors (ANNs) algorithms [121, 166, 15, 248], which offer significant speed improvements and reduce memory usage. However, these gains often come at the expense of accuracy, which remains capped by the exhaustive kNN method. This trade-off is evident in an experiment demonstrated in Fig. 6.1, where the state-of-the-art retrieval approach CosPlace [25] is applied to the SF-XL dataset [25].

SF-XL covers the city of San Francisco, encompassing nearly 170km^2 and over 40 million images. When scaling from a smaller version of SF-XL to its full scale, inference times for exhaustive kNN increase dramatically, whereas ANNs exhibit a slower, linear growth relative to database size, albeit with a significant reduction in accuracy.

An alternative approach to address scalability issues involves framing VPR as a classification problem, enabling location predictions without relying on similarity searches in a database. This formulation has primarily been explored in the context of coarse, planet-scale geo-localization [183, 292, 138, 138], where the globe is divided into large, unevenly distributed classes spanning hundreds of kilometers. We question whether these classification strategies can be adapted for fine-grained, city-scale VPR, which demands localization precision within a few meters. Existing global-scale classification techniques [183, 292, 138, 138] are unsuitable for this purpose due to their coarse granularity and inability to handle the visual overlap between densely sampled classes.

To address this, we present a classification-based VPR method tailored for densely mapped urban environments. Our approach, named **Divide&Classify** (D&C), combines the speed of classification techniques with the accuracy of retrieval methods. Crucially, we show that D&C's predictions can effectively limit the search space for retrieval approaches, enabling a hybrid pipeline that surpasses previous methods in accuracy while maintaining faster and consistent inference times (see Fig. 6.1). By contrast, current global-scale classification techniques fail to offer comparable benefits due to their limited accuracy.

Contributions. In summary, the key contributions of this chapter are:

- This chapter addresses the fine-grained (error $\leq$ 25m) and city-wide (map area >100 km^2) VPR challenge from a classification perspective, identifying the limitations of existing global-scale approaches and presenting the first viable solution (D&C).
- We introduce a classifier named Additive Angular Margin Classifier (AAMC), which employs prototypes learned via Additive Angular Margin Loss for classifying images. AAMC is scalable and delivers robust performance.
- We demonstrate that our method not only achieves results comparable to retrieval techniques but also enables the development of a fast, scalable, and accurate hybrid pipeline that leverages the strengths of both classification and retrieval.

Code and trained models are publicly available.

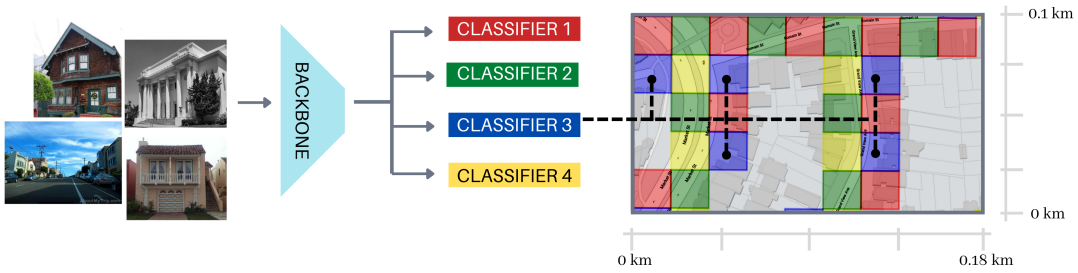


Figure 6.2: **Architecture of D&C**.The different groups, and their relative classifiers, are color coded. The picture explicates how cells are distributed into groups.

6.2 Related Work

Retrieval-based VPR. The majority of VPR methods rely on retrieval-based techniques, where location predictions are derived through similarity searches over a database of pre-computed embeddings, using either local [59, 121, 124, 203] or global features [10, 134, 158, 305, 306]. In more recent approaches, deep feature-extractors have become the dominant choice, often combined with aggregation or pooling layers. NetVLAD [10] is a prominent example, sometimes enhanced with attention mechanisms [134]. Common pooling strategies include [218, 264, 216]. A comprehensive overview of retrieval-based techniques is provided in [27, 169]. These methods typically employ contrastive learning with triplet loss, necessitating a mining procedure to generate suitable triplets, which must be periodically updated. Due to this setup, retrieval pipelines face scalability challenges: during *training*, as the database expands, the computational cost of mining overshadows the actual training process; at *test time*, the similarity search time increases linearly with the database size, potentially causing impractical delays for real-world applications. CosPlace [25] mitigates the training-time scalability issue by adopting an alternative contrastive learning strategy, enabling large-scale database training and achieving competitive performance across multiple datasets. However, it still relies on retrieval during inference, leaving the test-time scalability problem unresolved. Parallel to our work, [148] presents a generalized contrastive loss formulation that eliminates the need for hard pair mining.

Classification-based VPR. An alternative approach in the literature formulates place recognition as a classification task on a global scale. Early examples in this domain include RevIm2GPS [275] and PlaNet [292]. These methods divide the earth into predefined geo-classes, with the predicted class's geographic center serving as the output location. This classification-based strategy offers significant advantages in terms of computational efficiency. However, accuracy heavily depends on the

chosen partitioning scheme. While fine-grained partitions improve localization resolution, increasing the number of classes introduces challenges, such as parameter growth and reduced samples per class. Hierarchical Geolocation Estimation (HGE) employs a geographic hierarchy to alleviate these issues [183], while CPlaNet [239] utilizes a combinatorial partitioning of multiple geoclass sets. Other approaches [138, 118] focus on learning optimal class centers. More recently, [260] introduced a partitioning scheme based on interpretable hierarchies (e.g., city, region, country). For fine-grained urban environments, the classification-based idea was explored as early as 2013 [100] using SVM classifiers, though limited to a small area (≈ 1.56 km^2) rather than a full city-wide scope.

Relationship with prior works. This chapter bridges retrieval and classification methods, drawing insights from prior state-of-the-art (SOTA) works in both areas to develop a tailored solution for city-wide localization.

Compared to CosPlace [25] (**retrieval-based** SOTA), we adopt the concept of **Groups**, which represent distinct geoclass partitions. While CosPlace uses only a subset of groups for feature learning, our approach trains on all groups to model a geographically comprehensive distribution. Furthermore, CosPlace discards its classifiers during inference, relying instead on similarity searches that demand substantial memory and time. In contrast, we retain all classifiers and leverage their *collective predictions* for efficient inference.

Among **classification-based** methods, no clear SOTA exists: widely cited works like PlaNet [292] and CPlaNet [239] rely on private datasets, making fair comparisons difficult, and lack public implementations. Previous comparisons are further complicated by inconsistent backbones across methods. Our work provides the first equitable evaluation of classification techniques in a fine-grained, city-scale context. CPlaNet shares some similarities with our method, as it combines predictions from multiple classifiers: the authors define five discrete earth partitions and infer locations using their intersections via a combinatorial scheme. These partitions overlap and consist of geographically adjacent classes. However, we demonstrate that in dense urban settings (unlike sparse global-scale scenarios), adjacent class partitions hinder learning, and our partitioning strategy avoids this limitation. Sec. 6.3.1 elaborates on this reasoning, and Sec. 6.4 supports this finding empirically. Another distinction is CPlaNet's irregularly shaped classes, formed by intersecting partitions and assigned logits combinatorially, leading to slower inference. Our method employs a simpler partitioning scheme, improving speed while significantly outperforming alternatives.

Finally, Sec. 6.4.2 shows that our approach can integrate with retrieval methods into a unified pipeline, achieving faster inference and higher accuracy. Unlike existing classification-based methods, which are too imprecise for such combinations, the accuracy of our method makes it effective to be used as a distractor-filter for retrieval.

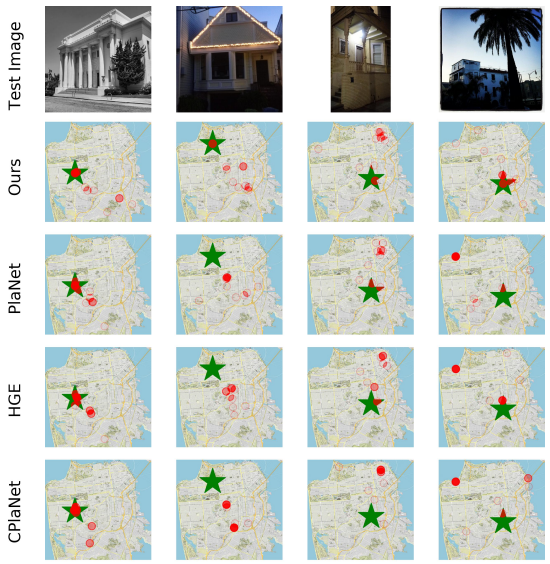


Figure 6.3: **Qualitative examples of predictions for each method.** The green star shows the ground truth of each test image, while the red circles represent the first 10 predictions. Brighter red indicates multiple predictions close to each other.

6.3 Method

This chapter addresses the VPR task in large urban environments (covering areas exceeding 100km^2), where a place is considered correctly recognized when the predicted location falls within 25m of the ground truth, following established practices in the field [10, 289, 158, 96, 26, 105, 284, 27, 306]. The approach assumes access to a training set of geo-referenced urban images $\mathcal{T} = \{(I_i, east_i, north_i)\}$, where each image I_i is associated with UTM coordinates $(east_i, north_i) : \mathbb{UTM}_e \times \mathbb{UTM}_n$, providing a local flat-surface approximation of GPS coordinates. To enable precise localization, $\mathcal{T}$ is constructed through dense sampling across the urban area (approximately one image per meter of roadway).

The objective is to leverage $\mathcal{T}$ to train an effective classifier for VPR. This is accomplished through two main components. First, $\mathcal{T}$ is divided into distinct sets of cells (classes) ensuring no visual overlap between classes within the same partition. Second, an ensemble of classifiers is employed, with one classifier dedicated to each partition. Specifically, we present an Additive Angular Margin Classifier (AAMC), which builds upon the discriminative capabilities of the Additive Angular Margin Loss [282]. The partitioning methodology is detailed in Sec. 6.3.1, while the AAMC is described in Sec. 6.3.2.

6.3.1 Partitioning method

Discretizing the continuous label space (GPS/UTM coordinates) of training set $\mathcal{T}$ into classes presents significant challenges. The high, uniform sampling density further complicates learning with conventional classification frameworks, as adjacent cell boundary images may exhibit nearly identical appearances and embeddings despite belonging to different classes. This similarity generates noisy gradients and impairs model learning. To overcome this limitation, the adopted partitioning strategy creates multiple mutually exclusive splits of $\mathcal{T}$, guaranteeing that no two classes within the same split share boundaries. This approach effectively eliminates *perceptual aliasing* between classes belonging to the same partition.

The partitioning method operates by *constructing multiple splits composed of* ***non-adjacent*** *geographic cells, with a dedicated classifier trained for each split* (see Fig. 6.2). Each cell (functioning as a class) is assigned to precisely one classifier, while its neighboring cells are allocated to different classifiers. The cell creation scheme, inspired by CosPlace [25], employs equal-sized square cells determined by a single hyperparameter M representing cell side length. Each cell corresponds to a distinct class. For coordinate prediction in geo-localization, a Class2UTM function maps each class to its cell's center coordinates. Formally, class $C_{e_i,n_j} \in \mathcal{S}$ is defined as:

$$C_{e_i,n_j} = \left\{ (east, north) : \left\lfloor \frac{east}{M} \right\rfloor = e_i, \left\lfloor \frac{north}{M} \right\rfloor = n_j \right\} \tag{6.1}$$

with the Class2UTM function given by:

$$\begin{aligned} &\text{Class2UTM} : \mathcal{S} \to \mathbb{UTM}_e \times \mathbb{UTM}_n, \\ &C_{e_i,n_j} \mapsto \Big((e_i + 0.5) \cdot M,\ (n_j + 0.5) \cdot M \Big) \end{aligned} \tag{6.2}$$

The cells are then organized into Groups through another hyperparameter N, with each Group defined as:

$$G_{uv} = \left\{ C_{e_i n_j} : (e_i \bmod N = u) \ \wedge \ (n_j \bmod N = v) \right\} \tag{6.3}$$

N determines the minimum separation (in cell units) between neighboring classes within a split and governs the total number of groups $|G| = N^2$. Through empirical analysis in Sec. 6.4.3, optimal hyperparameter values are established as $M = 20$ meters and $N = 2$. Unlike CosPlace's approach [25], this method requires no heading angle annotations, enhancing its general applicability.

6.3.2 Additive Angular Margin Classifier (AAMC)

The Groups formulation yields $|G|$ independent cell partitions, enabling a *Mixture-of-Experts* framework with dedicated classifiers for each partition (see Fig. 6.2). Recent investigations [63, 282, 25] demonstrate that large-margin losses improve

siamese network embeddings for retrieval tasks, promoting more discriminative features and better-structured latent spaces [176, 286]. However, these techniques have traditionally been limited to feature extractor training, with their learned prototypes remaining unused during classification. To capitalize on these benefits, this chapter describes a classifier that effectively utilizes the informative prototypes acquired during training.

Building upon the Additive Angular Margin Loss (ArcFace) [63], previously applied in large-scale retrieval systems [282, 249, 25], the AAMC enhances inter-class separation by maximizing angular distances through cosine similarity between normalized embeddings and learnable class prototypes. The ArcFace loss is formulated as:

$$\mathcal{L}_{arc} = \frac{1}{N}\sum_{i} -\log\frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{i\neq j} e^{s\cos\theta_j}} \tag{6.4}$$

with constraints:

$$\begin{gathered}\cos\theta_j = W_j^T x_i \\ W = \frac{W^*}{||W^*||}, \quad x = \frac{x^*}{||x^*||}\end{gathered} \tag{6.5}$$

where x_i denotes the i-th embedding for ground-truth class y_i and W_j represents the j-th class prototype vector.

While prior works [282, 249, 25] employed angular margin losses solely for embedding extraction, discarding the prototypes, this work demonstrates that these prototypes establish meaningful embedding-to-class mappings suitable for direct classification. To our knowledge, this represents the first implementation of ArcFace prototypes during inference.

The AAMC approach associates each k-th classifier (corresponding to the k-th Group) with a prototype matrix $W_k \in \mathbb{R}^{S_k \times d}$, where S_k indicates class count in group k and d denotes embedding dimensionality. During inference, given test image embedding $x \in \mathbb{R}^d$, the method generates $|G|$ normalized predictions: $\{p_k = \text{softmax}(W_k^T x) \in \mathbb{R}^{S_k}\}_{k=1}^{|G|}$.

The final prediction selects the highest-confidence output across all classifiers:

$$\begin{gathered}\text{D\&C} : \mathbb{R}^{S_1} \times \ldots \times \mathbb{R}^{S_{|G|}} \rightarrow \mathbb{UTM}_e \times \mathbb{UTM}_n, \\ \{p_k\}_{k=1}^{|G|} \mapsto \text{Class2UTM}(\underset{k\in 1..|G|}{\text{argmax}}\ \underset{c\in S_k}{\text{argmax}}\ p_k(c))\end{gathered} \tag{6.6}$$

where $p_k(c)$ refers to the c-th element of vector p_k. In essence, each classifier produces logits distributions for its assigned group's classes, with the final prediction determined by selecting the highest-confidence output across all AAMC classifiers.

The *mixture-of-AAMC* approach operates on the principle that while a test image's true class resides in only one classifier, neighboring classifiers will likely predict proximate classes due to visual similarity among nearby locations. Fig. 6.7 provides additional insight into this behavior by examining prototype agreement and correlation across different AAMC classifiers.

6.4 Experiments

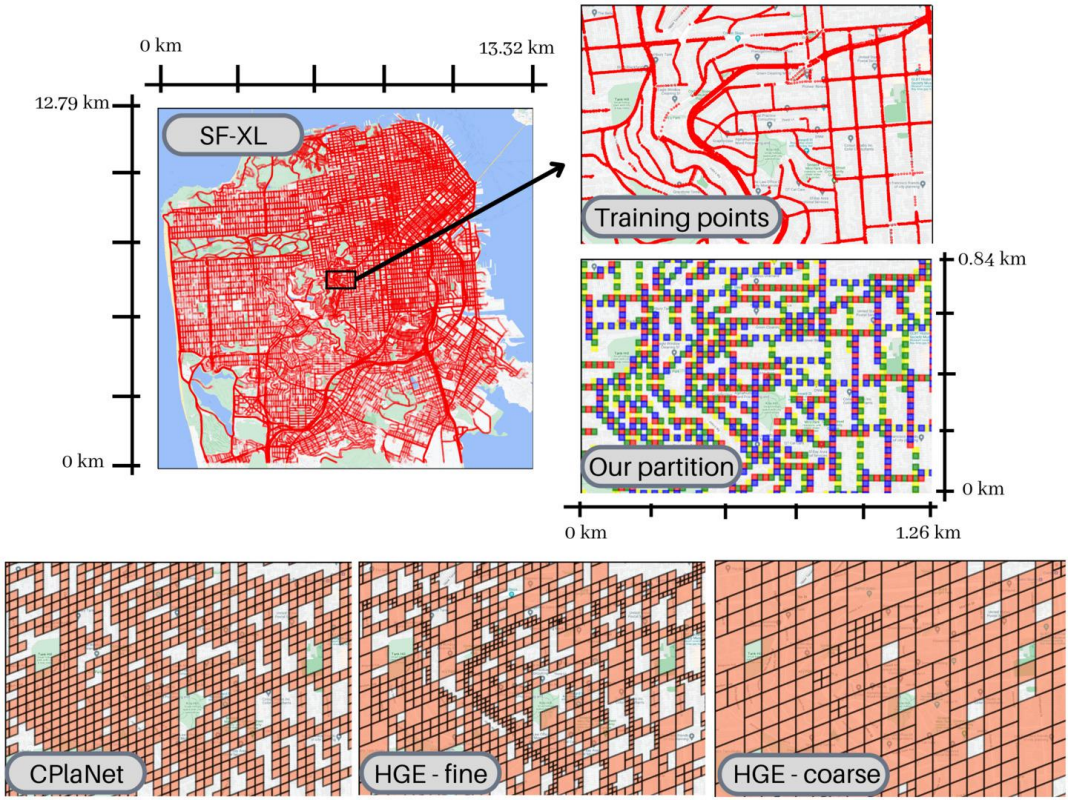


Figure 6.4: **Maps showing multiple partition methods over the training data of SF-XL.**

6.4.1 Experimental Setting

Implementation details. The dataset is divided into classes using $M = 20$, resulting in square cells with 20-meter sides. These classes are then organized into $N = 2$ groups, yielding 4 distinct classifiers in our framework. This partitioning scheme generates more than 113k classes for the SF-XL training set, visualized in Fig. 6.4. Additional details regarding class generation across different datasets are available in the Supplementary Material.

Each classifier undergoes independent training once per epoch. The training process consists of 1 million iterations with a batch size of 64. Since there are 4 classifiers, each one is trained every 4 epochs using the Adam optimizer [136] with a learning rate of $1e^{-4}$. Following [138], we employ an EfficientNet backbone. To ensure fair comparisons, all classification methods use identical batch sizes, data

augmentation techniques, backbone architectures, and training datasets. This approach differs from prior works [292, 239, 183, 138, 118] that utilized varying backbones and training data (see Sec. 6.2), making it challenging to determine whether performance improvements stemmed from methodological advances or architectural differences. For our approach, the backbone is supplemented with average pooling and a whitening layer [216] before feeding into the classifiers.

We evaluate different techniques for city-wide visual place recognition using the San Francisco eXtra Large dataset [25], containing 41.2 million training images covering over 100 km^2 of urban area. The dataset includes two smartphone-captured test sets (*test set v1* and *test set v2*). Currently, this remains the only dataset suitable for fine-grained, city-scale VPR applications matching our target scenario. Additionally, Sec. 6.4.7 examines the application of our method to smaller urban datasets (under 3 km^2) to discuss its reduced effectiveness in such contexts.

Metrics. VPR techniques are evaluated using various metrics depending on their approach. Retrieval-based methods for fine-grained VPR typically employ recall@N (R@N) with a 25-meter threshold [10, 158, 134, 96, 25, 27], while classification methods for global-scale localization use Great Circle Distance (GCD) [106]. To enable direct comparison between these approaches, this chapter introduces **Localization Radius@N** (LR@N), defined as: *The percentage of queries correctly localized within 25 meters of ground truth in at least one of the top-N predictions.*

While LR@N matches R@N @25m for retrieval systems, its formulation makes it applicable to classification pipelines. Notably, LR@1 corresponds directly to GCD@25m.

Baselines overview. We categorize baselines into three groups:

1) Existing *classification methods* originally developed for global geolocalization, adapted for city-scale use. These include PlaNet [292], Hierarchical Geolocation Estimation (HGE) [183], CPlaNet [239], and MvMF [118]. Unlike our approach, these methods utilize the S2 Sphere library with hyperparameters optimized for global classification. For fair comparison, we conducted extensive hyperparameter tuning on SF-XL (results in Tab. 6.1, with final partitions shown in Fig. 6.4). Where implementations were unavailable, we reproduced results from global datasets and adapted the code for city-scale evaluation. Implementations for PlaNet, CPlaNet, and MvMF will be made publicly available. Supplementary materials contain additional analysis of optimal partitioning parameters.

2) *Retrieval methods* employing deep networks to generate matching descriptors. Our experiments include NetVLAD [10], CRN [134], SARE [158], SFRS [96], and CosPlace [25].

3) *Retrieval-ANN hybrids* combining CosPlace with top-performing approximate nearest neighbor techniques: Hierarchical Navigable Small Worlds (HNSW) [166] and Inverted File Index with Product Quantization [248, 123]. We selected configurations optimized for either speed or LR@1 performance, with additional

ANN experiments detailed in the Supplementary.

Method	Infer. time	LR@1	
		test v1	test v2
Classification			
PlaNet [292]	12 ms	24.5	53.1
HGE [183]	15 ms	27.0	56.4
CPlaNet [239]	17 ms	27.4	64.1
MvMF [118]	12 ms	22.6	52.2
D&C (ours)	12 ms	61.0	79.1
Retrieval			
NetVLAD [10]	12117 ms	40.0	71.1
CRN [134]	12117 ms	45.8	76.4
SARE [158]	12117 ms	45.5	78.8
SFRS [96]	12117 ms	51.2	83.1
GeM [216]	1514 ms	21.7	43.1
CosPlace [25]	1514 ms	64.7	83.4
Retrieval + Approximate Nearest Neighbor			
CosPlace [25] + HNSW [166]	4 ms	52.5	77.8
CosPlace [25] + IVFPQ [123] *	8 ms	55.1	82.6
CosPlace [25] + IVFPQ [123] *	141 ms	63.7	83.3
Mixed pipeline			
D&C (ours) + CosPlace [25]	30 ms	**71.4**	**87.6**

Table 6.1: **Comparison of results for a large number of methods using different approaches.** All inference times measures are averaged over 1000 queries, on a system with a RTX 3090 GPU and i9-10940X CPU. The FAISS library [126] is used for all nearest neighbor implementations. *Mixed pipeline* is the best configuration from Tab. 6.2, which performs retrieval on the top-100 classes obtained through D&C. *We show two versions of the Inverted File Index with Product Quantization, one tuned for speed and one for recall.

6.4.2 Main Results

Comparison with existing approaches. Table 6.1 presents quantitative comparisons between our method and existing baselines, revealing several key findings:

- Existing **classification methods**, designed for global datasets with uneven distributions, perform poorly on SF-XL (qualitative results in Fig. 6.3).
- Our approach, specifically developed for dense urban environments, surpasses previous classification methods and approaches retrieval-based state-of-the-art performance (3.7 points lower LR@1 on *test v1*).
- **ANN** techniques accelerate retrieval by 10-400x, with corresponding LR@1 decreases of minimal to 12 points on *test v1*.
- A **combined pipeline** integrating our classification method with CosPlace achieves speeds comparable to classification alone while significantly outperforming all other approaches. Further details appear in the following section.

Classification + retrieval

This section examines how classification and retrieval methods can be combined to enhance both accuracy and efficiency. The approach restricts kNN search to cells receiving highest confidence from the classification model. For instance, using Top-10 predictions limits retrieval to images from the 10 most probable locations. Figure 6.5 illustrates how this filtering reduces irrelevant matches.

We evaluate various classification models (HGE, CPlaNet, and our method) paired with retrieval approaches (NetVLAD [10] and CosPlace [25]). Results in Tab. 6.2 break down inference time into classification, descriptor extraction, and kNN components (database descriptors are pre-computed). Top-N=All indicates standard retrieval without filtering.

The results demonstrate significant benefits from combining classification with retrieval. Speed improvements stem from reduced search spaces, while accuracy gains result from eliminating low-probability regions. Notably, using Top-100 predictions with CosPlace achieves state-of-the-art performance (6% LR@1 improvement) while being 500x faster than pure retrieval. Similarly, NetVLAD gains 20% LR@1 and 10,000x speedup when restricted to Top-10 predictions. Figure 6.1 visually demonstrates how our method offers superior scalability compared to retrieval approaches, even when enhanced with optimal ANN techniques.

6.4.3 Ablations

Class partition analysis. Figure 6.6 investigates M and N parameter choices. With $|G| = N^2$ classifiers (see Sec. 6.3.1), each trained every $|G|$ epochs (Sec. 6.4.1),

Retrieval Method	Top-N	kNN Time (ms)	Classification Method & Time		
			D&C (12 ms)	HGE (15 ms)	CPlanet (17 ms)
			LR@1	LR@1	LR@1
NetVLAD	All	12117	40.0	40.0	<u>40.0</u>
	1000	42	50.6	<u>42.8</u>	38.7
	100	4	56.7	41.1	37.4
	10	0.6	**62.6**	35.6	34.6
	1	0.06	56.1	24.8	26.3
CosPlace	All	1514	65.9	<u>65.9</u>	<u>65.9</u>
	1000	8	70.3	62.0	57.5
	100	1	**71.4**	52.7	51.0
	10	0.1	70.2	40.9	42.4
	1	0.03	57.0	25.0	26.7

Table 6.2: **Results of classification + retrieval pipelines** on SF-XL test v1. The *Top-N* column represents the number of cells within which we compute retrieval. The rows with *Top-N: All* are equivalent to using retrieval only, while the other rows employ the classification filter, reducing the search space by orders of magnitude. For retrieval, we use a VGG16 backbone. NetVLAD's dimensionality is 4096-D PCA (extraction time 2.1 ms), while CosPlace has 512-D (extraction time 5.1 ms).

excessive classifiers receive insufficient training, while $N = 1$ suffers from the learnability issues discussed in Sec. 6.3.1. $N = 2$ emerges as the optimal choice.

For cell size (M), 20 meters performs best. Larger values (50m, 100m) yield inferior results due to increased intra-class variability and learning difficulty.

Loss function analysis. While cross-entropy (CE) loss represents a natural choice for classification, Tab. 6.3 shows that AAMC consistently outperforms CE-trained linear classifiers. This advantage stems from AAMC's margin maximization, which not only separates classes but pushes them beyond a minimum margin, creating better-structured feature spaces (visualized via t-SNE in Supplementary materials).

Classifier behavior analysis. Our method employs multiple AAMC classifiers trained on disjoint sets, raising questions about prototype relationships. Ideally,

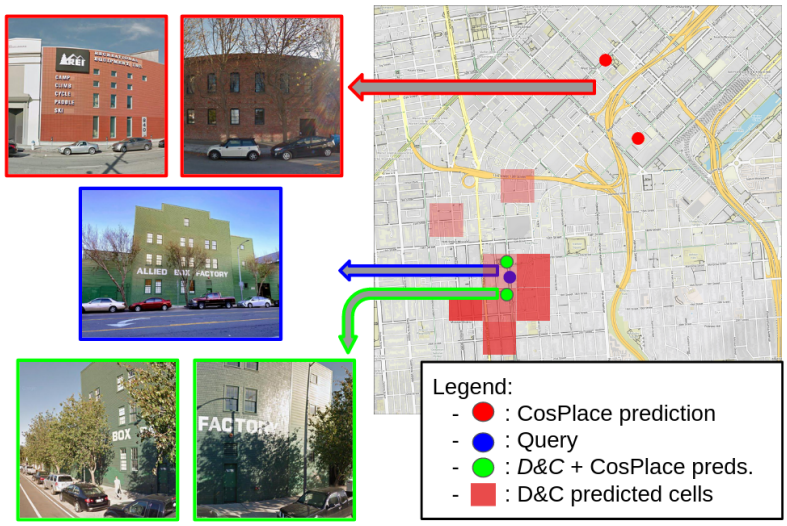


Figure 6.5: Example of our **mixed pipeline**. Thanks to the reduction in the search space obtained via the predictions of D&C, the retrieval module correctly localizes the query

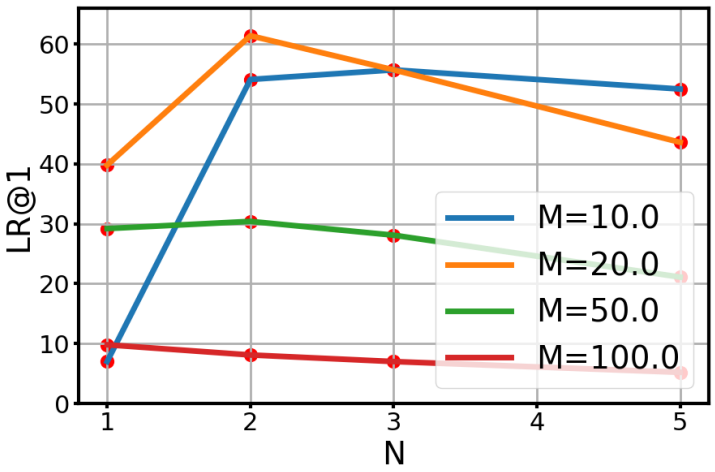


Figure 6.6: **Ablation** on the values of M and N. M determines cell size, N is the distance between cells in a group.

prototypes should exhibit geographic correlation despite being learned independently. Figure 6.7 confirms this behavior by analyzing similarity distributions among 500 neighboring prototypes across groups.

This emergent property stems from our partitioning strategy: while adjacent cells share visual features that could confuse a single classifier, independent learning allows prototypes to naturally align with geographic proximity. Additionally,

Classifier	LR@N (at 25 m)			
	LR@1	LR@5	LR@10	LR@20
Cross-Entropy	48.2	62.6	68.0	72.0
AAMC	**61.4**	**73.6**	**77.1**	**79.6**

Table 6.3: **Ablation AAMC vs Cross-Entropy classifier.** This table clearly presents the benefit of our AAMC classifiers, which largely outperform standard linear classifiers trained with a cross-entropy loss.

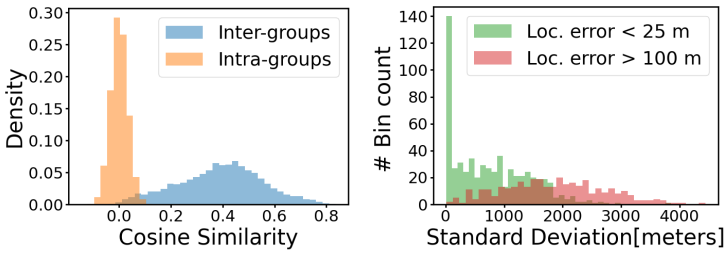


Figure 6.7: **Coordination of prototypes across groups.** (a) The left plot samples 500 neighboring prototypes (across all 4 groups), and shows their inter- and intra-group cosine similarity. It shows high correlation among inter-groups neighboring prototypes. Prototypes within a single classifier (intra-group) are well separated. (b) In the right plot we study the standard deviation (STD) among the prediction of each classifier. We can see that when the $N^2 = 4$ predictions are close to each other, the localization error is likely to be low (< 25 meters), proving that the STD between D&C's predictions from each expert is a good confidence measure, which is a very important value in real-world applications.

Fig. 6.7 (right) examines classifier agreement by analyzing coordinate prediction standard deviations. Correct localizations (<25m) show concentrated predictions, while errors display greater dispersion, indicating prediction reliability correlates with classifier consensus.

6.4.4 Additional analyses

Sec. 6.4.5 presents additional ablation studies to gain deeper insights into the functioning of our presented method.

In Sec. 6.4.6, we examine in detail how the partitioning scheme from prior work, initially developed for planet-scale localization, was adapted for city-wide localization scenarios.

6.4.5 Further Ablations

Embedding comparison: AMCC versus Cross Entropy.

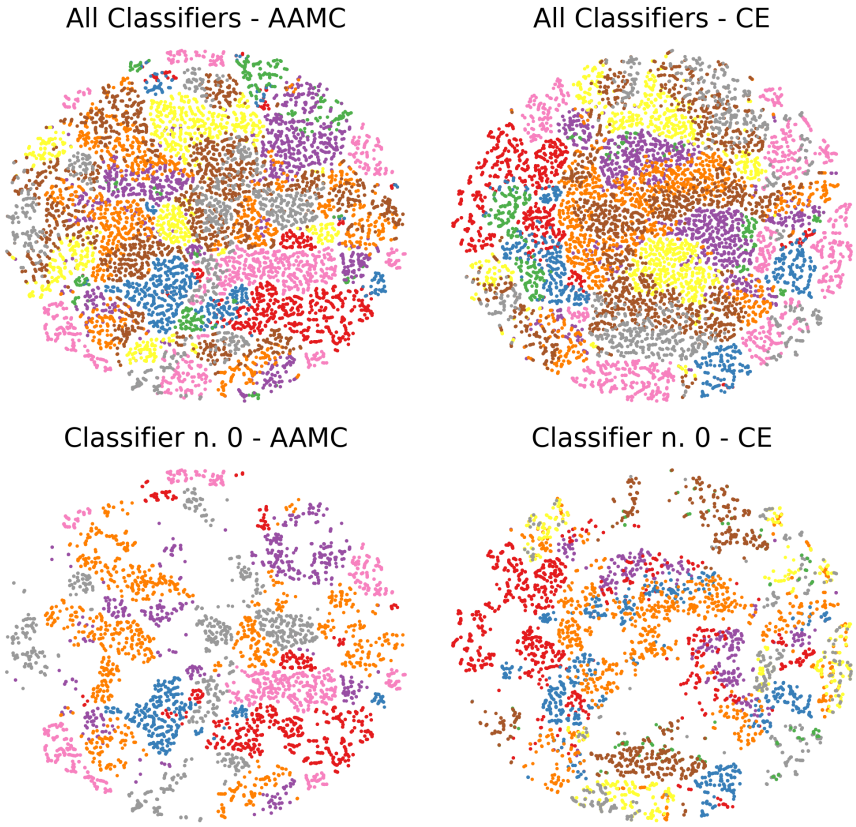


Figure 6.8: **t-SNE analysis** of embeddings in a 100m x 100m square. Each color codifies a different 20m cell.

The top row of Fig. 6.8 displays t-SNE visualizations of embeddings within a 100m square area, comparing models trained with either AAMC or a standard fully connected layer using cross-entropy loss; different colors represent distinct 20m cells. While certain patterns emerge, some overlap is observable, which is expected given the visual similarity between neighboring cells at this fine resolution. The bottom row illustrates why D&C enables each classifier to learn meaningful distributions: within each group, the non-adjacent cell arrangement creates well-defined classes. Specifically, these plots demonstrate how AAMC produces better-organized embedding spaces through its large margin approach.

Training dynamics of LR across different methods. Fig. 6.9 examines the evolution of LR@1 across training epochs for various approaches (considering the dataset size, we define an epoch as 2k iterations). We observe that existing methods - PlaNet [292], CPlaNet [239], and Hierarchical Geolocation Estimation [183] - show

limited LR improvements during initial epochs compared to our D&C approach, which exhibits rapid progress from the outset. MvMF initializes its mixture weights using a pretrained PlaNet model and concludes training before reaching 100 epochs.

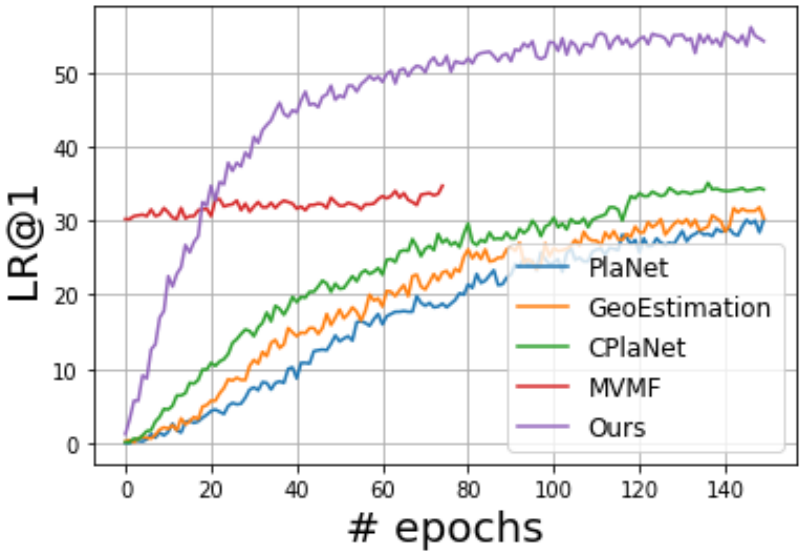


Figure 6.9: **Behaviour of LR@1 during training** for each of the methods. Note that MVMF [138] starts with a high LR because it uses the weights of a trained PlaNet model.

Impact of N on classification accuracy during training. To investigate how N influences training stability, we generated plots comparing $N = 2$ and $N = 3$, tracking training set accuracy after each epoch. Fig. 6.10 reveals that early training produces accuracy patterns with periodicity matching $|G| = N \times N$, where $|G|$ corresponds to both the number of groups and classifiers. This phenomenon occurs because each classifier undergoes training once every $|G|$ epochs, causing noticeable accuracy jumps at $|G|$-epoch intervals as previously trained classifiers are revisited.

AAMC hyperparameter analysis.

Our AAMC classifier incorporates two key hyperparameters from the ArcFace formulation: s, controlling hypersphere projection radius, and m, governing the cosine space margin between prototypes.

Approximate Nearest Neighbor Search Evaluation. Fig. 6.12 presents results for the top-performing configurations of Approximate Nearest Neighbor search algorithms in our experiments. The visualization includes only the most effective combinations. Additional tested methods include standard Product Quantization [123], Inverted File Indexes (combinable as IVFPQ), and Inverted File MultiIndex [15], which demonstrated inferior performance compared to those shown.
For Table 1 in the main text, we selected two configurations from this pareto

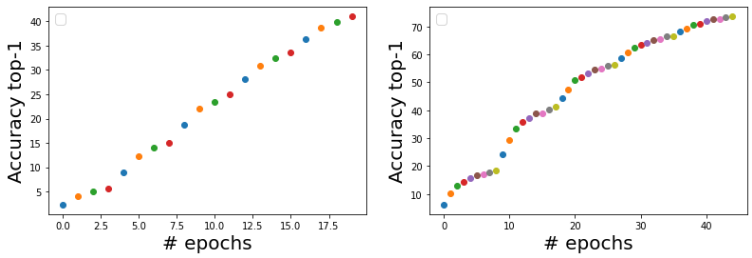


Figure 6.10: **Evolution of classification accuracy during training with different values of N.** We can see that in the first epochs of training, the accuracy on the train set presents waves with period length of size $|G|$. Each color represents a different classifier being trained at the given epoch for a total of $|G|$ colors.

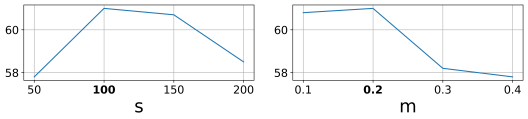


Figure 6.11: **Analysis of s (left) and m (right).** Optimal values are highlighted in bold.

front: one prioritizing accuracy (IVFPQ(128,50)), chosen for delivering at least 10x speedup with maximum performance, and another emphasizing speed (IVFPQ(128,2) and HNSW(512)), selected for achieving minimum 100x speedup.

6.4.6 Experiments on small datasets

The main text discusses how retrieval methods surpass classification approaches on small datasets due to insufficient positive training samples. However, the inference time difference becomes less significant with smaller datasets. Tab. 6.4 provides quantitative results demonstrating how the presented methods perform on datasets 1000x smaller than SF-XL, covering areas under $3km^2$ with half SF-XL's density.

Implementation Details of Baseline Methods. While prior works employ varying dataset partitioning strategies, we meticulously adjusted partitioning hyperparameters to enable fair method comparisons. Some approaches divide geographical areas based on training point density [292, 183], while others fix cell dimensions and merge them until reaching target region counts [239].

Tab. 6.5 shows the optimal class counts for each method, with subsequent paragraphs explaining our empirical determination process for these values.

Method	C-Pitts30k (30k images)		C-Tokyo 24/7 (76k images)	
	LR@1	Inf. time	LR@1	Inf. time
Classification				
PlaNet [292]	31.5	12 ms	19.5	12 ms
HGE [183]	33.6	15 ms	22.0	15 ms
CPlaNet [239]	33.0	17 ms	21.5	17 ms
MvMF [118]	31.5	12 ms	19.9	12 ms
D&C (ours)	40.5	12 ms	33.7	12 ms
Retrieval		(kNN time)		(kNN time)
NetVLAD [10]	86.1	58 ms	62.2	130 ms
CRN [134]	86.3	58 ms	62.8	130 ms
SARE [158]	87.2	58 ms	74.8	130 ms
SFRS [96]	88.7	58 ms	78.5	130 ms
GeM [216]	77.9	16 ms	46.4	25 ms
CosPlace	88.5	16 ms	82.8	25 ms
Mixed pipeline				
D&C (ours) + CosPlace	81.9	1 ms	74.9	3.5 ms

Table 6.4: **Comparison of LR@1** of different methods for *Pitts30k* and *Tokyo24/7* using *EfficientNet-B0* as backbone

Partitioning Schemes in HGE, PlaNet and MVMF. PlaNet [292], Hierarchical Geolocation Estimation (HGE) [183], and MVMF [118] share partitioning approaches, with HGE additionally employing two coarser splits (*medium* and *coarse*) alongside the standard *fine* partition. Using Google's S2 Sphere library, these methods require two parameters (τ_{min} and τ_{max}) defining cell image count bounds. Through empirical evaluation on SF-XL, we identified optimal parameter values (Tab. 6.6), selecting partitions yielding best HGE LR@1 performance and applying the same fine partition to PlaNet and MVMF. This process led to $\tau_{min} = 100$ and $\tau_{max} = 2500$ (see Tab. 6.7 for other partition values), following [183]'s proportional relationships between partition sizes.

We optimized cell density parameters on SF-XL as the most representative dataset. For consistency across other datasets (C-Pitts30k, C-Tokyo24/7), we scaled τ_{min} and τ_{max} according to relative density differences. Our method maintains consistent $20m$ cells across all datasets, as its partitioning depends solely on localization granularity.

CPlaNet Partitioning Approach.

For CPlaNet [239] partitions, we adhered closely to the original implementation: creating five *geoclass sets* per experiment, where *geoclass set*$_1$ and *geoclass*

Partition method	SF-XL	C-Pitts30k	C-Tokyo 24/7
PlaNet / MVMF	65k	486	1840
HGE	19k / 35k / 65k	158 / 272 / 486	508 / 961 / 1840
CPlaNet	54k	369	1236
Ours	114k	687	2492

Table 6.5: **Number of classes in different datasets** using different partitioning methods.

HGE Num. Classes			
coarse	medium	fine	LR@1
65.3k	119k	200k	19.0
35.0k	65.3k	119.0k	21.2
18.5k	35.0k	65.3k	27.0
9.4k	18.5k	35.0k	25.3
3.8k	9.4k	18.5k	19.2
1.8k	3.8k	9.4k	10.6

Table 6.6: **Results with different partitions** using HGE on SF-XL.

set_2 evaluate proximity using solely geographical or visual features respectively, while remaining sets combine these modalities stochastically. The method includes an additional hyperparameter controlling class counts in each *geoclass set* (the partitioning termination condition). Final predictions consider intersections across all five *geoclass sets*. Tab. 6.8 displays results using various parameter combinations, with the first column showing unique cell counts from partition intersections. We maintained consistent average cell sizes across datasets when transferring these hyperparameters.

6.4.7 Limitations

Our method excels in large, densely-mapped areas (e.g., SF-XL's 170 km^2 coverage). On smaller, sparser datasets like Pitts30k [267] and Tokyo 24/7 [265] (both under 3 km^2 with less than half SF-XL's density), retrieval methods achieve superior performance (over 80% recall@1 [96, 25]), while classification methods struggle (LR@1 below 50%). This occurs because retrieval only requires one matching database image per query, whereas classification needs sufficient examples per class.

hyperparams	fine	HGE-medium	HGE-coarse
τ_{min}	100	100	100
τ_{max}	2500	5000	10000

Table 6.7: **Chosen hyperparameters** for previous methods partitioning. Note that Planet, HGE-fine and MvMF use the same partitioning.

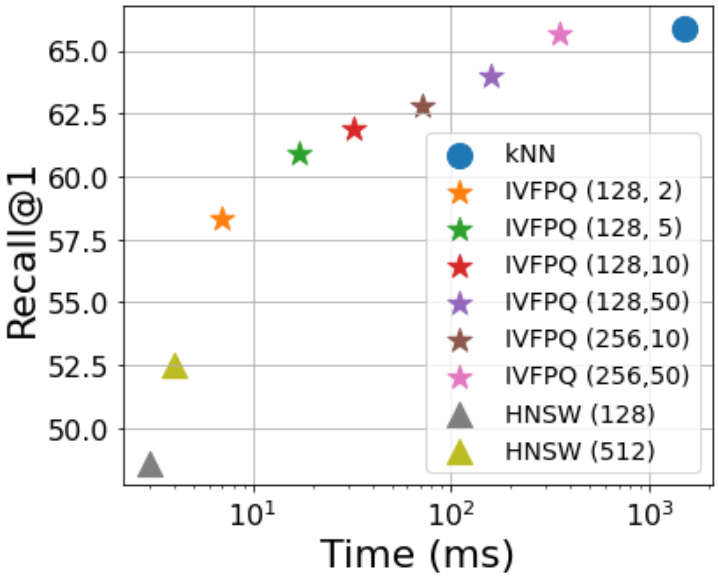


Figure 6.12: **Comparisons of best-performing Approximate Nearest Neighbor search algorithms.** We show only the pareto-optimal results, which are computed with an Inverted File Index with Product Quantization (IVFPQ) [123] and Hierarchical Navigable Small Worlds (HNSW) [166]. The parameters in parenthesis for IVFPQ indicate the number of subquantizers and the *nprobe*, *i.e.* the number of Voronoi cells to be searched (out of 1000). The parameters in parenthesis for HNSW indicates the number of connections each vertex has within the HNSW graph.

Cells per geoclass						
# classes	gcs 1	gcs 2	gcs 3	gcs 4	gcs 5	LR@1
58233	30k	30k	39k	36k	33k	27.6
54144	20k	20k	26k	24k	22k	27.7
47412	10k	10k	13k	12k	11k	25.7

Table 6.8: **CPlaNet preliminary results** on SF-XL.

6.5 Conclusions

This chapter demonstrates the effectiveness of approaching fine-grained VPR in urban settings through a classification-based framework. To our knowledge, this represents the first successful attempt at tackling this demanding scenario, establishing that precise localization can be achieved while maintaining dependable confidence estimation in the results. We present an inference pipeline designed to harness the combined knowledge of multiple trained classifiers, which surpasses previous classification-based localization approaches in performance. Additionally, we illustrate how the presented framework can be integrated with retrieval-based techniques to achieve an optimal balance between computational efficiency and localization accuracy, enabling the development of more efficient and precise VPR systems.

Chapter 7

The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement

Camera pose estimation constitutes a fundamental capability for numerous computer vision applications, from robotic navigation to augmented reality systems. While structure-based approaches utilizing sparse 3D point clouds have established the current state-of-the-art in visual localization, these methods exhibit inherent limitations regarding flexibility to the scene representation, which is tied to the feature representation of choice. This chapter explores an alternative paradigm through pose refinement techniques, presenting a versatile framework that complements existing localization pipelines while overcoming several key limitations of traditional approaches.

Contemporary localization systems predominantly rely on establishing 2D-3D correspondences between query images and preconstructed scene models. While effective, this methodology depends heavily on the specific feature representations used during scene reconstruction, creating a tight coupling between mapping and localization stages. Furthermore, the sparse nature of structure-from-motion point clouds limits their utility for other vision tasks beyond pure localization. Mesh-based representations offer an attractive alternative, providing feature-agnostic 3D models that support diverse applications while maintaining efficient rendering capabilities crucial for real-time systems.

The core contribution of this chapter lies in proposing a render-and-compare paradigm for pose refinement. Rather than relying on specialized, task-optimized features as in previous approaches, we demonstrate that generic deep features provide sufficient discriminative power for effective pose estimation when combined with an efficient particle filter optimization framework. This insight significantly simplifies the pose refinement pipeline by eliminating the need for scene-specific training or differentiable pose estimation modules. Our proposed MCLoc system

leverages this observation to deliver robust performance across diverse scenarios while maintaining computational efficiency.

Through our experimentation, we establish that this approach not only matches but often surpasses the accuracy of more complex refinement methods that require per-scene optimization. The framework's flexibility enables seamless integration with various scene representations and compatibility with different feature extraction backbones. Furthermore, the method scales effectively to large environments and demonstrates consistent performance across both indoor and outdoor scenarios. This versatility positions MCLoc as both a standalone localization solution and a complementary component that can enhance existing pose estimation pipelines through pre-processing or post-refinement stages.

Key aspects of this chapter's contribution include: a computationally efficient particle filter implementation that robustly explores the pose hypothesis space; systematic analysis of generic deep features for pose refinement tasks; and an open framework that supports experimentation with different scene representations and similarity metrics. The resulting system provides practitioners with a practical tool for camera pose estimation that balances accuracy, efficiency, and ease of deployment across diverse application scenarios. Code and models have been released at `https://github.com/ga1i13o/mcloc_poseref`.

Part of the work described in this chapter has been previously published in the following paper:

- G. Trivigno, C. Masone, B. Caputo, T. Sattler. **The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement** In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR) 2024;

7.1 Introduction

Visual localization involves determining the position and orientation of a camera within a given environment. This capability is crucial for numerous applications, including Simultaneous Localization and Mapping (SLAM) [73, 18], Structure-from-Motion (SfM) [234, 235], autonomous navigation [189, 61], robotics [85, 84], and Augmented/Virtual Reality (AR/VR) [214, 101].
Current leading techniques adopt a structure-based methodology [224, 225], where a 3D scene map is available, and a query image is localized by establishing 2D-3D correspondences. These matches are typically derived by comparing local features [220, 75, 64] between the query image and the 3D points in the map, often represented as an SfM [235, 234] sparse point cloud. The matches are then used to compute the camera pose via minimal solvers [102, 204] combined with robust optimization techniques [83, 57]. Constructing the point cloud through SfM requires detecting and describing local features across reference images, matching these features, and triangulating points visible in multiple images [234, 102]. The resulting 3D points are subsequently linked to visual descriptors extracted from the reference images.
Although SfM-based point clouds facilitate accurate and reliable localization [224, 228, 226, 254], they exhibit limited flexibility as they depend on the specific features employed during reconstruction and are primarily suited for localization tasks [190, 34, 197].

An alternative to traditional map representations is the use of meshes [197, 198, 274], which are feature-agnostic and support various tasks within pose estimation ecosystems, such as SLAM [190, 252, 34, 320], tracking [142], path planning [112], and relocalization [12, 274], while retaining the 3D information needed for visual localization. These models are straightforward to acquire [41, 181, 128] and can be rendered efficiently (*e.g.* , within 1 ms or less) even for large, textured models [197], leveraging well-established graphics pipelines.

Another strategy for visual localization involves refining an initial pose estimate. This can be applied either to enhance a pose derived from 2D-3D matches or to improve an initial hypothesis provided by image retrieval [225, 113]. Consequently, these methods are largely complementary to the matching-based approaches discussed earlier. Such techniques typically follow a *render&compare* paradigm [142, 151]: in each iteration, a synthetic view (either an image [142, 299, 312] or a sparse feature projection [223, 97, 277]) generated from the current pose estimate is compared to the query image. Based on this comparison, the pose is adjusted to better align the query with the scene representation. Existing methods often employ task-specific features [223, 97, 177, 51], sometimes jointly optimized with the scene representation [50, 178, 157].

This chapter suggests that in a *render&compare* framework, the key requirement

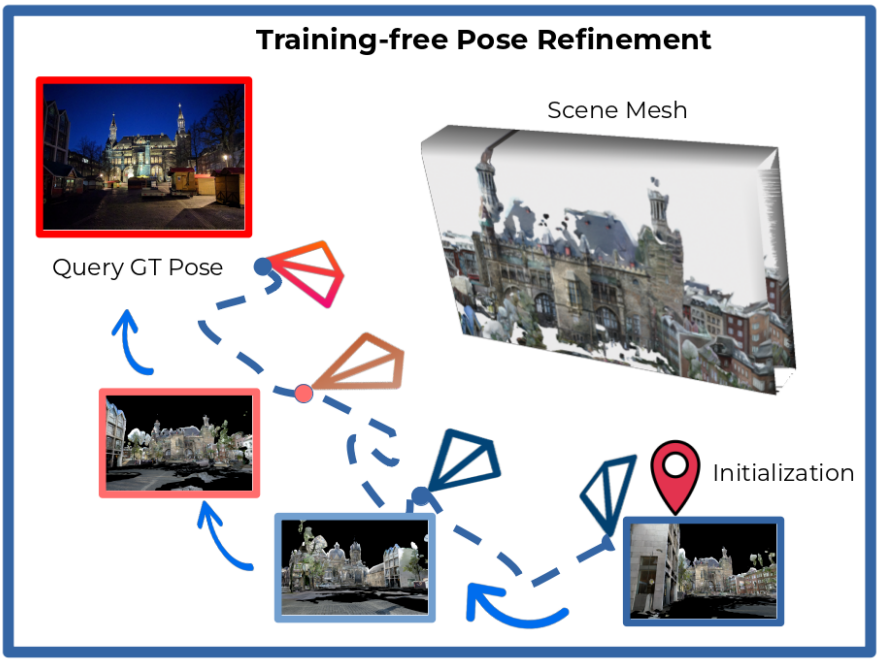


Figure 7.1: **MCLoc** estimates camera poses using a *render&compare* approach. Starting from an initial hypothesis, a particle filter generates perturbations and samples new candidates, which are rendered and compared to the query using generic pre-trained features.

is the ability to assess visual similarity between synthetic and real images. Prior work has demonstrated that generic deep features reliably estimate this similarity [95, 308, 135], making them suitable for pose re-ranking [255]. This contrasts with refinement approaches that depend on sparse, task-optimized features, raising the question of whether specialized feature training is necessary for localization, or if comparable results can be achieved using general-purpose, off-the-shelf dense features.

By avoiding feature optimization, there is no need for a differentiable feature-to-pose pipeline to compute gradients. Instead, this chapter employs a straightforward particle filter-based optimizer [261, 141] to efficiently explore the hypothesis space [55].

Despite its simplicity, the presented **MCLoc** approach surpasses modern pose regressors [178, 50] and performs comparably or better than refinement methods based on implicit fields [177, 97, 51], despite the latter being optimized per-scene. Additionally, the method scales effectively to large scenes.
While matching-based techniques remain state-of-the-art, this approach offers complementary advantages, as it can enhance matching-based methods as a pre- or

post-processing step. Extensive experiments across indoor, outdoor, and large-scale scenarios demonstrate its versatility, showing that pose refinement can generalize across domains and representations without requiring specialized training.

Contributions:

- A straightforward yet effective particle-filter optimization applicable to diverse scene representations and scoring functions
- An evaluation of general, pre-trained features at different network layers as a robust similarity measure
- A flexible pose-refinement framework requiring no per-scene training or fine-tuning, usable standalone, as a pose prior, or for refining existing estimates
- The accompanying code, supporting experimentation with various backbones, scoring functions, and scene representations, is available at `https://github.com/ga1i13o/mcloc_poseref`

7.2 Related works

Visual Localization. Visual Localization focuses on determining the camera pose of a query image within a known environment. A widely used approach involves utilizing sparse 3D models generated through SfM techniques [234], where point clouds link 3D positions to features extracted from database images. During inference, local features establish correspondences between queries and the 3D structure [229, 227, 152, 233, 224, 226, 41, 255, 256]. The camera pose is subsequently computed using PnP algorithms [143].

To reduce computational complexity, hierarchical strategies are often employed [225, 113, 114], incorporating Place Recognition networks [211, 27] to identify potentially relevant database images [10, 270, 25]. Alternative approaches replace SfM models with dense representations, including point clouds from Multi-View stereo or LIDAR [235, 241, 255], meshes [197, 41, 312, 271], or neural radiance fields [171, 299, 157]. This chapter demonstrates that renderable scene representations enable pose alignment through pixelwise feature comparison, eliminating the need for exhaustive matching.

While matching-based techniques achieve superior accuracy, our approach offers complementary advantages: it can seamlessly refine initial or final pose estimates while maintaining computational efficiency.

Implicit representations for Visual Localization. Unlike explicit geometric representations, implicit approaches encode scene information within neural network parameters. Early implementations included pose regression models [130, 178, 242, 66] and Scene Coordinate Regressors that predict 3D points from image

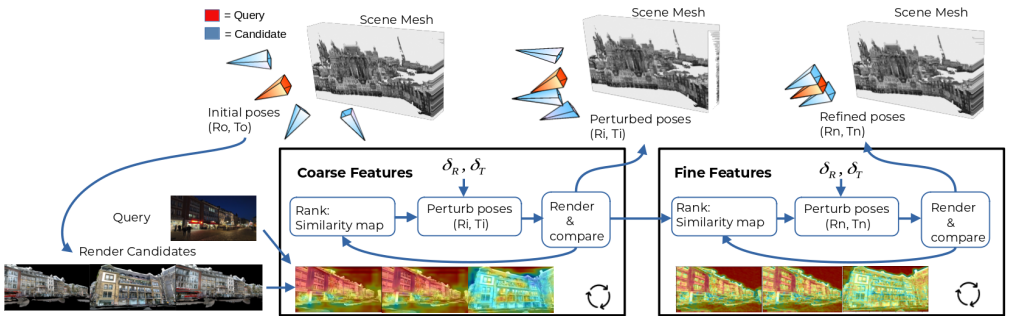


Figure 7.2: **MCLoc Architecture**. The diagram illustrates our iterative refinement process. Starting from an initial pose estimate, we generate perturbed variants and render corresponding views. These candidates are evaluated through dense feature similarity metrics. During optimization, we progressively utilize shallower network layers to capture finer details for precise alignment.

patches [39, 37, 45, 46]. Recent advancements have popularized neural radiance fields [171, 21, 182], which employ MLPs to model view-dependent appearance and geometry.

For localization tasks, these representations can be adapted by incorporating feature embeddings [97, 177, 157] or through field inversion techniques [299, 154]. Implicit models have additionally served as data generation tools for training pose estimation systems [178, 50].

Pose refinement and image alignment. Pose refinement techniques iteratively improve camera pose estimates by optimizing objective functions. Traditional Direct Alignment methods minimize photometric differences between projected scenes and current estimates [19, 80], employing gradient-based optimization [147, 167]. While prevalent in SLAM applications [236, 4], these photometric approaches struggle with appearance variations. Subsequent adaptations have applied similar principles to learned features [276, 277, 223].

Indirect methods instead minimize reprojection errors using geometric correspondences [210]. PixLoc [223] represents a notable direct method that learns features specifically for pose estimation. Recent work has explored refinement using implicit representations, where [299] employs rendered views for photometric error minimization.

Alternative approaches model feature fields rather than appearance. While FQN [97] utilizes reprojection errors, [177, 51] perform feature matching followed by descriptor field inversion. These methods require per-scene training and face scalability limitations. Our technique leverages pre-trained features applicable to any dataset and scene representation, including large-scale environments where meshes are available [197]. Our results align with [308], which demonstrated the superiority of learned similarity metrics, while extending this observation to pose

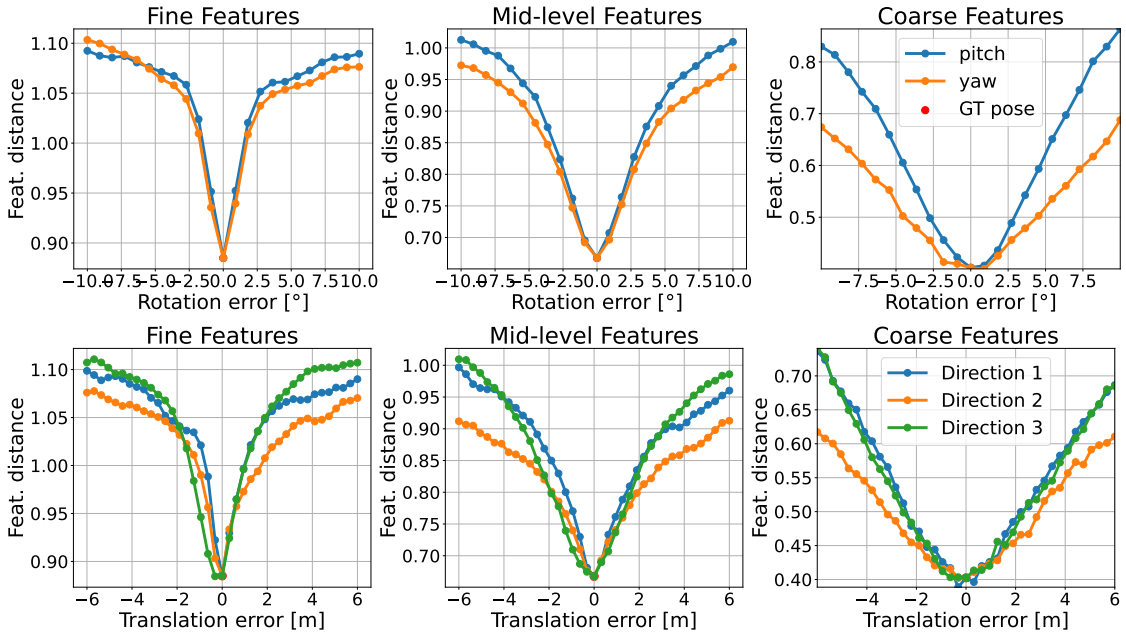


Figure 7.3: **Convergence Basin in Optimization Space at Multiple Scales**. The rotation and translation of a query from Aachen are perturbed, and dense feature distances are computed across various depths. First row: rotation adjustments along the yaw and pitch axes. Second row: displacement from the ground truth along three arbitrary directions.

discrimination tasks.

Localization with Particle filters. Previous work has explored particle filters for localization [213]. Similar optimization strategies appear in [154, 165], though both require neural radiance fields and depend on photometric error metrics.
Particle filters have found applications in mobile robotics for localization [127, 86, 116] and visual tracking [56], as well as in remote sensing for satellite image alignment [110]. Theoretical analyses of particle filters are available in [55, 141, 42, 140].

7.3 MCLoc

Overview. MCLoc estimates the pose of a query image within a *render-and-compare* framework, utilizing Monte Carlo simulation. The localization process involves iterative refinement of the pose, as illustrated in Fig. 7.2. Starting from an initial pose hypothesis (obtained through various means), perturbations are applied, and a renderable scene representation generates corresponding views. A generic feature extractor serves as a cost function to assess the similarity between candidates and the query. The pose estimates are modeled and adjusted using a

particle filter [261], functioning as a stochastic optimizer.
This approach is independent of the scene representation used for rendering and demonstrates that general-purpose feature extractors are suitable for evaluating pose alignment without requiring fine-tuning or scene-specific training.

Motivation.

This chapter investigates the following research question: *can generic features suffice for localization, or are specialized descriptors necessary?* This inquiry stems from the observation that deep network activations reliably estimate *perceptual similarity* [95, 308], while also exhibiting robustness to domain shifts, blur, and distortions [135]. Perceptual similarity appears well-suited for assessing pose similarity within a *render & compare* framework. We demonstrate that this property, combined with the inherent spatial structure of feature maps, provides an efficient means of measuring pose discrepancies. To achieve this, a perceptual metric is integrated into a particle-filter-based optimizer [261, 56], which generates new pose hypotheses for rendering and comparison.

Problem setting. The goal is to estimate the 6-DoF pose of a given query image I_q. Following [131, 113], the pose is parameterized as $T_q = (\mathbf{c}, \mathbf{q})$, where $\mathbf{c} \in \mathbb{R}^3$ denotes the camera center and $q \in \mathbb{R}^4$ is a unit quaternion. Quaternion-based representations offer numerical stability, compactness, and avoid gimbal lock [145]. This formulation separates translation and rotation updates, operating on the manifold of $\mathrm{SO}(3) \times \mathrm{T}(3)$ [42, 154]. The problem is formulated as the following optimization:

$$\hat{T}_q = \underset{T \in \mathrm{SO}(3) \times \mathrm{T}(3)}{\operatorname{argmin}} \mathcal{L}_{\mathcal{F}_\theta}(T | I_q, I_T) \tag{7.1}$$

where I_q and I_T are the query image and the candidate rendered at pose T, $\mathcal{F}_\theta$ is a feature extractor, and the loss function measures the feature-space distance between the query and the rendered candidate: $\mathcal{L}_{\mathcal{F}_\theta}(T | I_q, I_T) = ||\mathcal{F}_\theta(I_q) - \mathcal{F}_\theta(I_T)||_2$. This loss is optimized using a particle filter.

7.3.1 Pose alignment with Pre-trained features

To evaluate the loss in Eq. (7.1) for a candidate pose relative to the query, both are processed through a pre-trained CNN. Further details on the architecture are provided later. A hierarchy of feature volumes, $F_l \in \mathbb{R}^{C_l \times H_l \times W_l}$, is obtained for each level $l \in \{1..L\}$. These feature pyramids decrease in resolution while encoding richer semantic information as the receptive field expands. Prior work has shown that such hierarchies inherently measure perceptual similarities at varying conceptual levels [89, 308, 5]; to our knowledge, no previous methods leverage this property for assessing *pose similarity* in pose refinement. A straightforward scoring function exploits this behavior: at step s of the optimization, level $l(s)$ is selected

to compute the score between candidate I_T and query I_q as follows:

$$S(h,w|l) = \left\| \frac{F_l^{h,w}(I_q)}{||F_l^{h,w}(I_q)||_2} - \frac{F_l^{h,w}(I_T)}{||F_l^{h,w}(I_T)||_2} \right\|^2$$
$$\mathcal{L}_{\mathcal{F}_\theta}(T|I_q, I_T, l) = \frac{1}{h_l w_l} \sum_{h,w} S(h,w|l) \tag{7.2}$$

where $F_l^{h,w} \in \mathbb{R}^{C_l}$. In practice, pixelwise normalized descriptors are compared, yielding a spatial similarity map $S \in \mathbb{R}^{H_l \times W_l}$, which is then averaged. Early optimization stages handle large baselines, as initial hypotheses may deviate significantly from the ground truth. To widen the convergence basin, a hierarchical *Coarse-to-Fine* strategy is employed. Initially, deeper features are used due to their larger receptive fields, increasing the likelihood of overlapping regions between misaligned poses. These features also ignore low-level details and transient objects, enhancing robustness. As optimization progresses toward accurate poses, finer details and small displacements become critical. Shallower features, with higher spatial resolution and smaller receptive fields, are better suited for this stage. Pre-trained features, regardless of architecture or training method, exhibit an *unreasonably effective* convergence basin, as noted in [308]. This is experimentally validated in Fig. 7.3, which also shows how feature hierarchy depth controls the basin's width, with shallower features discerning even minor discrepancies.

7.3.2 Particle filter optimization

Overview. Particle filters are Monte Carlo methods that estimate system states based on observations and dynamics [261, 141]. These algorithms approximate diverse distributions efficiently by focusing on high-likelihood regions of the state space [86]. Their application to visual localization is well-established, with effectiveness demonstrated in [213, 154, 127, 165]. The core idea involves perturbing an initial state to generate new hypotheses, evaluating them with a cost function, and refining iteratively. Here, the state variable is the camera pose T, and the cost function is *perceptual similarity*. At each step, the particle filter models the posterior distribution $p(\mathbf{T}_q|\mathbf{Z}_i)$ of the query pose $\mathbf{T}_q$ using particles $\mathbf{Z}_i = \{(T_i^1, \pi_i^1), ..(T_i^n, \pi_i^n)\}$. Weights π_i^n represent likelihoods, estimated via Eq. (7.1). Since particle states $T_i^1, \ldots, T_i^n$ lie on the SO(3) $\times$ T(3) manifold, perturbations are applied using their Lie algebra, ensuring coordinate invariance [55, 140, 56].

Our approach. The loss in Eq. (7.1) is non-convex over the 7D optimization space, and convergence basins depend heavily on initialization. Key challenges include efficient hypothesis space exploration and enlarging the convergence basin. To address the former, multi-hypothesis tracking [56] is employed, where independent *beams* (sets of particles) are optimized in parallel. This allows broad exploration

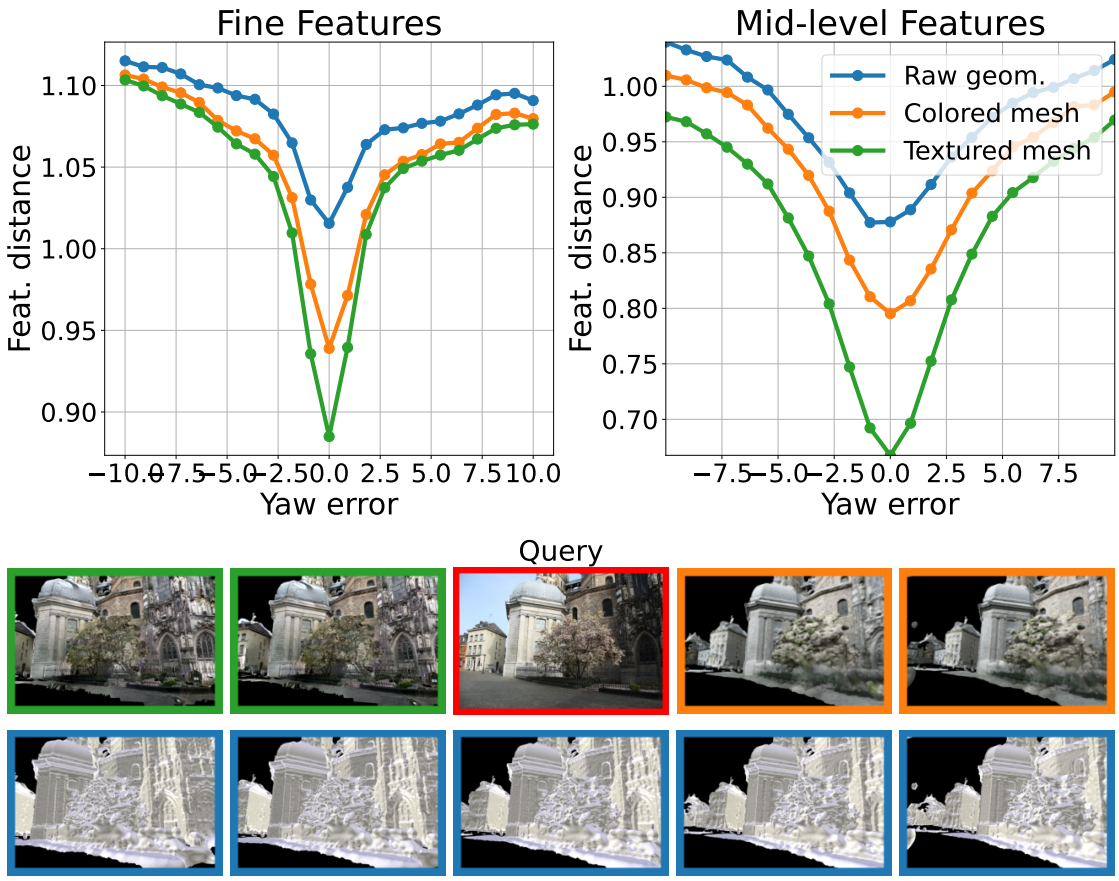


Figure 7.4: **Robustness of the Convergence Basin to the Rendering Domain**. Images are rendered while rotating along the yaw axis, using different meshes: Textured, Colored, and Raw Geometry. Feature distances are evaluated at varying depths, showing that domain shifts alter absolute values but preserve the basin shape.

of the state space [70], preventing local minima in one beam from affecting others. Initial steps use low-resolution renders (256×340) to reduce computational cost, as fine details are unnecessary early on. Every N_0 iterations, a *resampling step* pools the best candidates across beams, reinitializing new beams for further optimization. This halts unpromising hypotheses early. The *Coarse-to-Fine* strategy from Sec. 7.3.1 is applied to enhance convergence. After N_1 resampling steps, shallower feature maps are used. Additionally, image resolution increases gradually, while the number of beams and particles per beam decreases. This balances computational efficiency with the need for broad exploration early on and fine refinement later. Further implementation details are provided in Sec. 7.4.1, with pseudo-code available in the Supplementary. The code will be released upon acceptance.

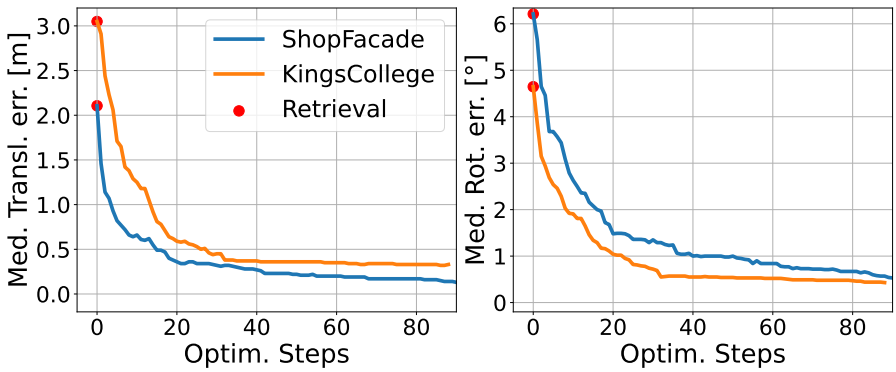


Figure 7.5: **Optimization trajectory**. Behavior of median errors over the iterations for 2 scenes from Cambridge Landmarks.

7.3.3 Adapting to different domains

Since the framework compares queries against rendered candidates, a natural question arises: is domain adaptation necessary to bridge the gap between real and synthetic data? Recent work [197, 312] suggests rendering domain shifts do not hinder matching performance. We extend this analysis, as our setup compares dense feature maps rather than local descriptors. Tests across rendering domains (textured, colored, raw geometry) reveal feature discrepancies, meaning the distance between a query and its rendered ground truth is non-zero. However, absolute values are irrelevant—only relative pose differences must correspond to similarity changes. Figure 7.4 illustrates this effect. In practice, domain shifts preserve relative differences, as the rendering domain is uniform. This underscores an advantage of our approach: independence from scene representation.

7.4 Experiments

Datasets. The proposed pose refinement approach is evaluated across several standard datasets. The Aachen Day-Night v1.1 dataset [312, 230, 228] serves as a standard benchmark for large-scale visual localization [223, 224], comprising 6,697 daytime reference images and 1,015 query images captured using handheld devices. This dataset covers extensive urban areas and includes challenging night-time queries along with significant viewpoint variations between reference and query images. Additional evaluations are conducted on two smaller but widely used datasets: Cambridge Landmarks [131] and 7scenes [243], containing 5 outdoor and 7 indoor environments respectively.

Both datasets feature query sequences captured along trajectories different from

the reference data. Following established protocols, we report recall rates at thresholds of $(25\text{cm}, 2°)$, $(50\text{cm}, 5°)$, and $(5\text{m}, 10°)$ for Aachen [228, 224], while median translation (meters) and rotation (°) errors are used for the remaining datasets [177, 157, 178].

Coarse Features	Fine Features	ShopFacade	OldHospital
ResNet-18			
CosPlace [25]	ImageNet	12 / 0.45	39 / 0.73
ImageNet	ImageNet	12 / 0.55	46 / 0.80
SimCLR [52]	SimCLR [52]	18 / 0.62	50 / 0.83
ALIKED [314]	ALIKED [314]	17 / 0.64	49 / 0.84
AlexNet [139]	AlexNet [139]	15 / 0.74	53 / 0.88

Table 7.1: **Feature extractor ablation study**, demonstrating that dense features consistently serve as robust estimators of *visual similarity* across various architectures and training approaches. Errors reported in $cm, °$.

7.4.1 Implementation details

The primary experiments utilize a lightweight ResNet-18 architecture [107] pre-trained for place recognition in [25]. This network was fine-tuned starting from the *conv3* layer, with earlier layers remaining frozen. Consequently, when switching to *conv2* during optimization, ImageNet features are actually being employed.

As shown in Tab. 7.1, this configuration yields marginally better results compared to vanilla ImageNet features. We posit that deeper layers trained for place recognition learn to focus on permanent structures like buildings while ignoring transient objects, which proves beneficial for localization tasks. After N_1 optimization steps, the process transitions to *conv2* features (ImageNet-only), with final refinement (after N_2 iterations) performed using *conv1*. The exact values of N_1 and N_2 are not crucial, provided that initial steps use coarser features and final steps employ finer ones, since *conv1* features exhibit a narrower convergence basin. Supplementary materials include additional experiments analyzing hyperparameter robustness and convergence behavior.

For camera center perturbation, Gaussian noise is applied with reduced standard deviation on the vertical axis (1/10 of other directions), reflecting the prior knowledge that height variations are typically constrained. Rotation perturbations occur around random axes with uniform noise. Noise magnitude decreases linearly

and resets every $N_0 = 20$ steps, implementing a *start-and-stop* schedule similar to CosineAnnealing [160].

Renderable scene representations. The method requires only a renderable scene model capable of generating views from arbitrary poses $(R, t) \in \mathrm{SE}(3)$. Recent work by [197, 198] has demonstrated the benefits of 3D meshes, particularly their flexibility across tasks and efficient rendering capabilities honed over decades of pipeline optimization.

As an alternative to meshes, neural radiance fields [171, 258] can produce photorealistic renderings, albeit with significantly higher computational costs. While recent advances have accelerated NeRF rendering [182, 219], they remain substantially slower than mesh-based approaches. We therefore experiment with Gaussian Splatting [132], which combines high-quality output with rendering speeds comparable to meshes.

For the large-scale Aachen dataset [230, 228, 312], we utilize compressed meshes from [197], enabling efficient rendering at low resolutions (500 μs per image). For Cambridge Landmarks [131] and 7scenes [243], we optimize 3D Gaussians from the provided point clouds following [132], requiring only 10 minutes of optimization time while achieving rendering speeds of 600-900 μs per frame (measured on RTX 4090 GPU).

7.4.2 Experimental results

This section presents ablation studies supporting the methodological choices, followed by comprehensive comparisons against state-of-the-art matching methods, pose regressors, and feature-based refinement approaches.

Ablation studies. Tab. 7.1 examines different architectural configurations. Dense features extracted from all tested architectures are employed as described in Eq. (7.2). While CosPlace+ImageNet achieves marginally better results, the method proves effective regardless of architecture, training protocol, or dataset. These findings extend the observations from LPIPS [308], demonstrating that generic features effectively measure not only perceptual similarity but also pose similarity.

The results indicate that dense features, whether trained via supervised or unsupervised objectives for classification, place recognition, or feature matching (ALIKED [314]), share the properties visualized in Fig. 7.3. Shallower layers provide precise alignment estimation, while deeper features accommodate wider baselines. These observations align with transfer learning principles [68], where high-quality features demonstrate broad applicability. Supplementary materials include additional ablations comparing different scoring functions, showing that simple dense pixelwise comparisons outperform more complex formulations.

Method		Cambridge Landmarks			
		King's	Hospital	Shop	St. Mary's
Retrieval					
DenseVLAD [265]	-	2.8/5.7	4.0/7.1	1.1/7.6	2.3/8.0
CosPlace [25]	-	3.1/4.4	4.5/6.7	2.1/6.2	3.2/7.2
SOTA					
AS [227]†	-	0.13/0.22	0.20/0.36	0.04/0.21	0.08/0.25
hloc [224]	TL	0.12/0.20	0.15/0.30	0.04/0.20	0.07/0.21
DSAC* [38]	TS	0.15 / 0.3	0.21 / 0.4	0.05 / 0.3	0.13 / 0.4
HACNet [150]	TS	0.18 / 0.3	0.19 / 0.3	0.06 / 0.3	0.09 / 0.3
PixLoc [223]	TL	0.14/0.24	0.16/0.32	0.05/0.23	0.10/0.34
Pose Regressors					
MS-Transformer [242]	TS	0.83 / 1.47	1.81 / 2.39	0.86 / 3.07	1.62 / 3.99
DFNet [50]	TS	0.73 / 2.37	2 / 2.98	0.67 / 2.21	1.37 / 4.03
LENS [178]	TS	0.33 / 0.5	0.44 / 0.9	0.25 / 1.6	0.53 / 1.6
Pose Refiners					
FQN [97]	TS	**0.28** / **0.4**	0.54 / 0.8	0.13 / 0.6	0.58 / 2.0
CROSSFIRE [177]	TS	0.47 / 0.7	0.43 / **0.7**	0.2 / 1.2	0.39 / 1.4
NeFeS (DFNet) [51]	TS	0.37 / 0.62	0.55 / 0.9	0.14 / 0.47	0.32 / 0.99
MCLoc (ours)	-	0.31 / 0.42	**0.39** / 0.73	**0.12** / **0.45**	**0.26** / **0.88**

Table 7.2: **Performance on Cambridge Landmarks dataset.** The presented approach outperforms methods requiring per-scene descriptor training. TM indicates feature matching methods, TL denotes localization-trained methods, TS represents scene-specific training.

Baselines. Performance comparisons are made against methods employing scene-specific training:

- **Pose Regressors**: Networks trained for direct pose prediction, including DFNet [50], LENS [178] and MS-Transformer [242]
- **Pose Refiners**: Most similar to our approach, including FQN [97], NeFeS [51] and CROSSFIRE [177] which optimize scene-specific descriptors in implicit fields (limited to small scenes). PixLoc [223] trains localization-specific features on MegaDepth [153]
- **Matching-based Methods**: Current state-of-the-art approaches, demonstrating how our method can enhance their performance

Method	Aachen Day-Night v1.1	
	Day	Night
Retrieval		
NetVLAD [10]	0.0 / 0.2 / 18.9	0.0 / 0.0 / 14.3
CosPlace [25]	0.0 / 0.4 / 27.1	0.0 / 0.0 / 24.1
Pose Refiners		
Pixloc [223]	63.2 / 67.8 / 75.5	38.7 / 47.1 / 60.7
MCLoc (ours)	55.8 / 73.3 / 89.7	42.4 / 66.5 / 86.9
Matching based		
AS [227]	85.3 / 92.2 / 97.9	39.8 / 49.0 / 64.3
hloc [224]	87.4 / **95.0** / 98.1	71.7 / **88.5** / **97.9**
+ PixLoc refine	86.2 / 94.9 / 98.1	70.8 / **88.5** / **97.9**
+ **(ours)** refine	**87.9** / 94.9 / **98.9**	**73.8** / **88.5** / **97.9**

Table 7.3: **Large-scale visual localization results on Aachen v1.1 dataset.** Competitive performance is demonstrated against PixLoc refinement, along with compatibility with state-of-the-art pipelines for improved results.

MeshLoc [197] pipeline	Top K Matched	Aachen Night v1.1
Textured Mesh		
LoFTR [254]	50	73.3 / 89.0 / 95.8
LoFTR [254]	20	71.2 / 89.0 / 94.8
LoFTR [254]	10	70.7 / 86.4 / 94.8
(ours) + LoFTR [254]	20	**74.3** / **91.1** / **99.5**
(ours) + LoFTR [254]	10	73.8 / **91.1** / 99.1
Raw Geometry		
P2P[316] + SG [226]	50	8.4 / 27.7 / 60.7
P2P[316] + SG [226]	10	6.8 / 20.4 / 52.4
(ours) + P2P[316] + SG [226]	1	16.8 / 37.7 / 66.0

Table 7.4: **Preprocessing results on Aachen Night**: The proposed method improves retrieval-based initial poses prior to expensive localization.

Comparison with scene-specific methods. Tab. 7.2 compares against refinement approaches [51, 97, 177] and pose regressors [50, 178] on Cambridge Landmarks [131]. While pose regressors offer faster inference, they generally underperform. Despite requiring no scene-specific tuning, our method outperforms all

implicit feature-based refiners except for a minor deficit on King's College where FQN performs slightly better. Scene Coordinate Regressors [38, 150] achieve the best performance among scene-specific methods.

Method	7 scenes: DSLAM ground truths median error in (cm/°) ↓						
	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
Retrieval							
DenseVLAD [265]	21/12.5	33/13.8	15/14.9	28/11.2	31/11.3	30/12.3	25/15.8
CosPlace [25]	31 / 11.4	45 / 14.6	23 / 13.7	43 / 11.2	52 / 11.4	48 / 11.1	46 / 14.8
SOTA							
AS [227]	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01
DSAC [39]	2/1.10	2/1.24	1/1.82	3/1.15	4/1.34	4/1.68	3/1.16
HACNet [150]	2/0.7	2/0.9	1/0.9	3/0.8	4/1.0	4/1.2	3/0.8
hloc [224]	2/0.85	2/0.94	1/0.75	3/0.92	5/1.30	4/1.40	5/1.47
Pose Regressors							
MS-Transf. [242]	11 / 4.7	24 / 9.6	14 / 12.2	17 / 5.66	18 / 4.4	17 / 6.0	17 / 5.9
DFNet[50]	5 / 1.9	17 / 6.5	6 / 3.6	8 / 2.5	10 / 2.8	22 / 5.5	16 / 2.4
LENS [178]	3 / 1.3	10 / 3.7	7 / 5.8	7 / 1.9	8 / 2.2	9 / 2.2	14 / 3.6
Pose Refiners							
FQN-PnP [97]	4 / 1.3	10 / 3.0	4 / 2.4	10 / 3.0	9 / 2.4	16 / 4.4	140 / 34.7
CROSSFIRE [177]	1 / 0.4	5 / 1.9	3 / 2.3	5 / 1.6	3 / 0.8	2 / 0.8	12 / 1.9
MCLoc (ours)	5 / 1.8	4 / 2.0	4 / 1.9	10 / 3.6	10 / 3.7	8 / 3.1	10 / 2.5
	SFM ground truths [40]						
MS-Transf. [242]	11 / 6.4	23 / 11.5	13 / 13.0	18 / 8.1	17 / 8.4	16 / 8.9	29 / 10.3
DFNet [50]	3 / 1.1	6 / 2.3	4 / 2.3	6 / 1.5	7 / 1.9	7 / 1.7	12 / 2.6
NeFeS [51]	2 / 0.8	2 / 0.8	2 / 1.4	2 / 0.6	2 / 0.6	2 / 0.6	5 / 1.3
MCLoc (ours)	2 / 0.8	3 / 1.4	3 / 1.3	4 / 1.3	5 / 1.6	6 / 1.6	6 / 2.0
(ours) w. DINOv2 [195]	3 / 0.9	4 / 1.8	3 / 1.5	6 / 1.4	7 / 2.1	8 / 1.8	9 / 2.2
(ours) w. RoMa [79]	2 / 0.7	3 / 1.2	2 / 1.0	3 / 1.1	4 / 1.0	5 / 1.4	6 / 1.5

Table 7.5: **Indoor localization results**. While challenging due to textureless surfaces, competitive performance is achieved on indoor scenarios.

Large-scale localization. Tab. 7.3 presents results on the Aachen v1.1 benchmark [312, 230, 228], demonstrating scalability where implicit field-based methods fail. Comparisons focus on PixLoc [223], another render-and-compare approach using feature-metric errors. Our method shows superior performance from retrieval initialization (except at finer thresholds for daytime queries), despite PixLoc's specialized feature training.

The method also complements state-of-the-art matching pipelines from hloc [224]. Using hloc poses as initialization, just 5 refinement steps yield improved accuracy with minimal overhead. Tab. 7.4 demonstrates another application: refining

retrieval poses to improve initialization for MeshLoc [197], reducing the number of candidates needed while boosting performance.

Indoor localization. On 7scenes [243], textureless surfaces pose challenges for perceptual similarity metrics. Despite this, comparable performance is achieved by increasing iteration counts. Ground truth inaccuracies identified by [40] affect evaluations, with better performance observed on their updated SFM labels. Experiments with DINOv2 [195] show comparable results to ImageNet features, with ViT-based models' fixed patch sizes yielding coarser features. The RoMA approach [79], which refines DINO features, matches or exceeds specialized pose refiners, suggesting potential for test-time optimization despite requiring feature matching training. Additional details are provided in supplementary materials.

7.5 Additional Experiments

In this section we analyze in more detail the following aspects:

- an ablation on different scoring functions to demonstrate the effectiveness of simple pixelwise comparison;
- a convergence analysis to test the robustness of our algorithm to initialization;
- additional insights on hyperparameters;
- a discussion on inference time;
- a pseudo-code version of our algorithm.

7.5.1 Scoring functions

In this chapter, we demonstrate the effectiveness of dense, pre-trained features for evaluating pose similarity, as compared to prior methods that rely on sparse, specialized features. The objective of these experiments is to examine whether dense features provide a tangible advantage or if similar outcomes would emerge from sparse comparisons. Consequently, we developed several alternative scoring functions for ranking candidates using sparse comparisons.
To recall, this scoring function is required to compute the loss from Eq. 1 of the main paper ($\mathcal{L}_{\mathcal{F}_\theta}(T|I_q, I_T)$), which is used at each step to compare the rendered candidates against the query and rank them. In Tab. 7.6, we compare the following cost functions:

- (1): the scoring function employed in our method, detailed in Eq. 2 of the main paper, namely the pixelwise L2 distance between feature maps, normalized across the channels;

- (2): a straightforward alternative to dense comparison involves exhaustive matching of detected keypoints. For this purpose, we use ALIKED [314] to obtain a set of keypoints and associated descriptors $\{k_i, f_i\}, k_i \in \mathbb{R}^2, f_i \in \mathbb{R}^d$ for each image. By computing the mutual nearest neighbors between the descriptors of the query I_q and a candidate I_T, we derive a set of matched keypoints $\{k_i, k_j\}, i \in K_{I_q}, j \in K_{I_T}$. The score is then the reprojection error among matched keypoints, *i.e.* , their spatial distance (in pixel space). Thus:

$$\mathcal{L}_{\mathcal{F}_\theta}(T|I_q, I_T) = \sum_{i \in K_{I_q}, j \in K_{I_T}} ||k_i - k_j||^2 \ . \tag{7.3}$$

- (3): exhaustive matching significantly increases the cost of computing the loss. Additionally, keypoints situated in opposite locations in the image pairs being analyzed do not provide meaningful signals for pose refinement. Consequently, a natural alternative to reduce the cost is to perform local keypoint matching. Here, the nearest neighbors are computed only for keypoints satisfying $||k_i - k_j||_2 \leq W$, where W represents the patch size defining the local window around each keypoint for computing matches.

- (4): implicit matching is an interesting concept explored in [58]. The main idea is to use a standard CNN to extract keypoints, rather than a dedicated keypoint detector. The underlying assumption is that these networks, through training, assign each channel to detect certain feature types. Thus, by identifying local maxima within each channel of the feature maps, these spatial locations can be compared across image pairs without descriptor matching. To evaluate this method, given a feature volume $F_l \in \mathbb{R}^{C,H,W}$, we determine for each channel: $k_c = \underset{h,w \in H,W}{argmax}\ F_l^{c,h,w}$. These are extracted for both the query k_c^q and a candidate k_c^T.
  To mitigate noise, we smooth these locations by applying a Gaussian filter over a window of size W, then compare them:

$$\mathcal{L}_{\mathcal{F}_\theta}(T|I_q, I_T, l) = \sum_{c \in C_l} ||k_c^q - k_c^T||^2 \ . \tag{7.4}$$

Results. The results presented in Tab. 7.6 indicate that among the scoring functions based on sparse comparisons (2, 3, 4), performance generally aligns with computational cost, with (2) being the most accurate but also the most computationally demanding. Overall, the simple dense comparison (1) emerges as the most effective, while also being lightweight and free of hyperparameters. This can be attributed to the discussion in Sec. 3.1 of the main paper: dense feature comparisons fully leverage the deep networks' ability to estimate *perceptual similarity* [308] and offer a smoother signal compared to sparse features, owing to the spatial structure

Scoring function	ShopFacade
(1) Dense Comparison	12 / 0.45
(2) Exhaustive Matching	20 / 0.93
(3) Patch-wise Matching	34 / 1.36
(4) Implicit Matching	85 / 1.92

Table 7.6: **Ablation on scoring function**. Demonstrates the effectiveness of dense feature comparisons versus more complex cost functions. Median errors are reported in $cm/°$.

Coarse Features	Fine Features	ShopFacade	OldHospital
CNN features: ResNet-18			
CosPlace [25]	ImageNet	12 / 0.45	39 / 0.73
ImageNet	ImageNet	12 / 0.55	46 / 0.80
SimCLR [52]	SimCLR [52]	18 / 0.62	50 / 0.83
Transformer: ViT small			
DINOv2 [195]	DINOv2 [195]	34 / 0.81	59 / 1.15

Table 7.7: **Ablation on feature extractors**, testing whether the robustness of dense features as estimators of *visual similarity*, typically associated with feature maps from CNN architectures, extends to state-of-the-art vision transformers. Median errors in $cm/°$.

of feature maps.
This characteristic of dense feature maps was central to our earlier work. While this effect has been primarily studied with CNN-derived feature maps [5, 135, 308], Tab. 7.7 explores state-of-the-art vision transformers trained with DINOv2 [195]. These architectures encode image patches as tokens. To adapt this model for our algorithm, we compute the distance between corresponding tokens in image pairs, using various encoder layers to retain our *Coarse-to-Fine* approach.
Although these tokens, along with positional encoding, preserve spatial information, our findings reveal that using them yields only moderate results, significantly below what is achievable with a simple ResNet-18. This discrepancy can be explained by

the receptive field of each token being constrained to be equal to or larger than the patch size (14 pixels), coupled with the self-attention mechanism embedding some global context into each patch. This perspective was recently supported in RoMa [79], which refines DINOv2 features using a specialized CNN architecture. In Tab.5 of the main paper, we evaluate this architecture and find that it matches or surpasses other specialized pose refiners, as illustrated in the table. It is worth noting that RoMa features rely on an architecture with approximately 80x more parameters than the ResNet-18 employed in our approach.

7.5.2 Optimization hyperparameters

In this section, we provide additional insights and analyses of some critical hyperparameters in our algorithm. As discussed in Sec. 4.1 of the main paper, a pivotal element for successful pose refinement is adopting a *Coarse-to-Fine* strategy, transitioning gradually from deeper features to shallower ones. With three feature levels (coarse-medium-fine), this involves determining two hyperparameters, N_1 and N_2. Here, N_1 specifies the number of steps before switching from coarse to medium features; N_2, given as a negative value, represents the number of final steps performed with the shallowest features.
Tab. 7.8 presents results for two Cambridge scenes, illustrating the impact of these two parameters. In these experiments, while varying N_1, we keep the number of steps after N_1 constant. Similarly, when adjusting N_2, the preceding steps are held fixed.

Another critical aspect of our method is multi-hypothesis tracking [56], which optimizes multiple *beams* independently. In principle, increasing the number of beams should enhance results, though this assumption may not hold if the total number of candidates sampled at each step remains constant—a desirable constraint to limit computational costs. Thus, we examine this trade-off in Fig. 7.6, where we test the effects of using no beams (*i.e.* , $nbeams = 1$) or varying numbers of beams. In our main experiments, we use 3 beams, starting with 50 candidates, which gradually reduce to 20 in the final steps.

Results. As mentioned in the main manuscript, our algorithm remains robust to the selection of (N_1, N_2) values, provided sufficient steps are performed with coarse features initially. This is because all feature levels exhibit a convex convergence basin around each pose, but the basin becomes narrower with shallower features. Since initialization through retrieval can result in large baselines, it is crucial to rely on coarse features for a sufficient number of steps to refine the pose adequately, enabling entry into the finer levels' convergence basin.
This is evident from Tab. 7.8, which shows that performing too few steps with

N_1	N_2	ShopFacade	OldHospital
15	-10	16 / 0.74	50 / 1.42
30	-10	<u>12</u> / <u>0.45</u>	<u>39</u> / <u>0.73</u>
50	-10	13 / 0.47	41 / 0.74
30	0	20 / 0.50	43 / 0.80
30	-20	12 / 0.43	37 / 0.72
30	-30	10 / 0.42	36 / 0.70

Table 7.8: **N. of steps before switching to coarser features**. We evaluate different N_1 values (coarse-to-mid switch) and N_2 values (final fine features). Default values are underlined. Median errors in $cm/°$.

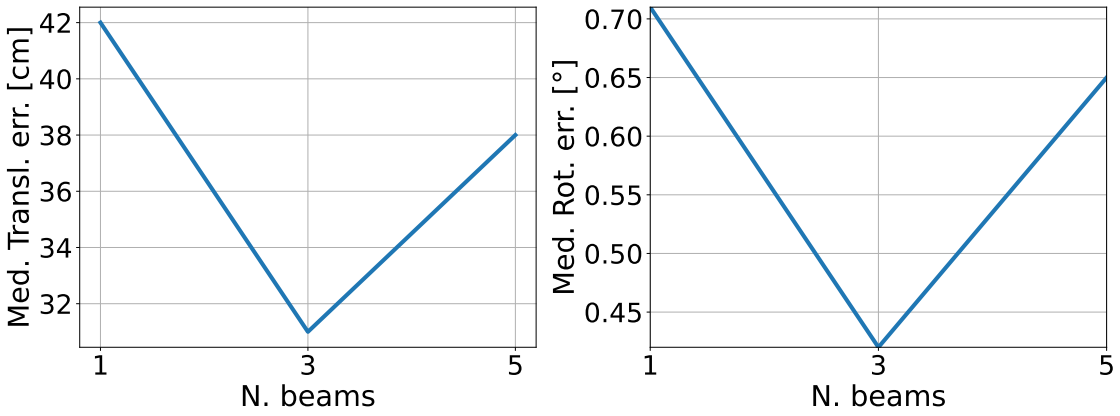


Figure 7.6: **Number of independent beams**. Results on KingsCollege.

conv3 features ($N_1 = 15$) has the most detrimental impact. Conversely, performing additional steps does not degrade performance but slows convergence. Regarding N_2, performing more steps with finer features improves results, though the gain is marginal. For this reason, we set N_2 to -10 to balance cost and performance. Fig. 7.6 highlights the advantage of employing multiple optimization

threads (*beams*) in parallel. However, increasing the number of beams excessively can be counterproductive; since the total number of candidates is fixed in these experiments, having more beams reduces the candidates each beam samples, diminishing their ability to explore the state space effectively, ultimately harming performance.

7.5.3 Convergence analysis

Similar to any refinement algorithm, the precision of the initial poses significantly influences the convergence rate and overall performance. To examine the sensitivity of our method to the initial error, we conduct an experiment analogous to the one described in [312]: we introduce random perturbations of varying magnitudes to the ground truth poses and then execute our algorithm for a set number of iterations. We employ magnitudes of $1, 5, 10, 15$ meters for translation and $5, 10, 20, 30$ degrees for rotation. For each magnitude, we repeat the sampling process 10 times to ensure a more reliable analysis.
These experiments are carried out on the ShopFacade dataset, and our optimization runs for 40 steps. It's worth noting that the results presented in the main chapter for Cambridge scenes were obtained using 80 steps. In this specific setup, we opted for fewer iterations due to the substantial number of combinations and repetitions involved in each experiment (totaling 160 runs). Nevertheless, the overarching trends and conclusions remain consistent.

Results. Fig. 7.9 presents the outcomes for each combination of translation and rotation errors in a matrix format, after 20 and 40 iterations. Observing the color map, it becomes clear that the achieved accuracy is more strongly correlated with the translation error. This observation is logical, as scene details might become less discernible from a greater distance, potentially leading to the pose falling outside the convergence basin. Conversely, at close range, our optimization can recover from significant rotation errors even with minimal overlap between the views.
In general, our algorithm demonstrates robustness to errors up to 5 meters, irrespective of the rotation, while performance starts to decline at 10 meters.

7.5.4 Inference cost

Inference speed is not a primary focus of our method. We observed a lack of consistent comparisons across different methods and implementations on identical hardware in the existing literature. Nevertheless, we provide a breakdown of the time required to optimize a pose over 80 iterations, which corresponds to the number of steps used to generate the results for the Cambridge scenes. The timings were measured on an RTX4090. Rendering the Gaussian cloud from [132] takes $0.8ms$. Across all optimization steps, we render a total of 2600 candidates for each query. Extracting features using a truncated ResNet-18, with FP16 precision and batch

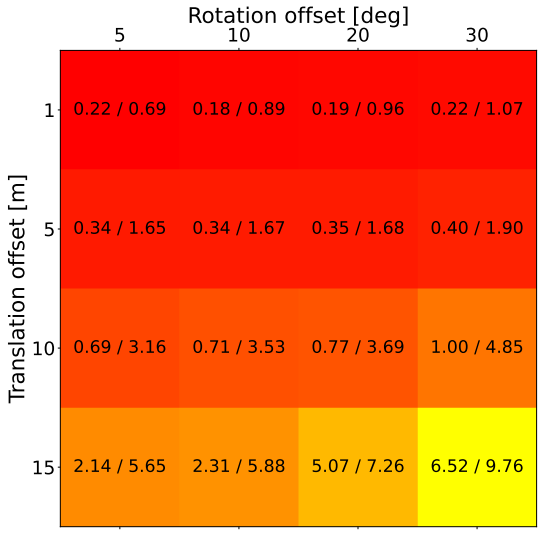


Figure 7.7: After 20 steps

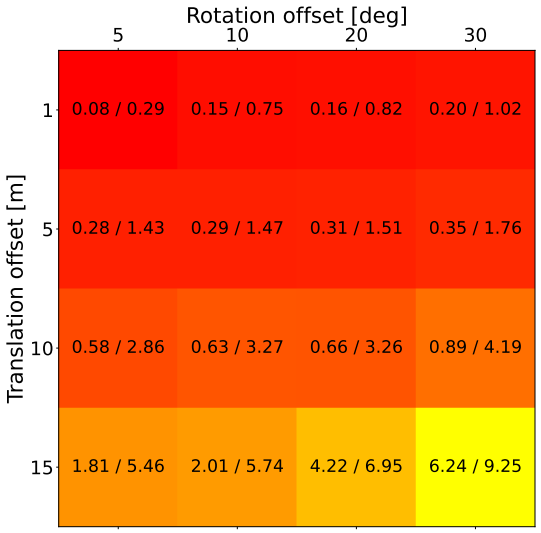


Figure 7.8: After 40 steps

Figure 7.9: **Convergence analysis.** We experiment on the ShopFacade scene by initializing the pose refinement with random perturbation of the ground truth, with varying magnitudes of perturbation. We then run the MCLoc pipeline to refine the pose for a fixed number of optimization steps. In this way we evaluate the robustness to different errors in the pose initialization. Numbers are reported as median errors, averaged over 10 runs, as $m/°$.

processing, requires $0.1ms$ per image at the lowest resolution (256×320). At the highest resolution we utilize (320×480), this increases to $0.2ms$. Considering the use of 3 beams, our approach takes approximately $2.4s$ on average for Cambridge and around $8.7s$ for Aachen (due to a higher number of iterations). Our optimization relies on independent beams, which allows for parallel processing, reducing the runtime to $1.1s$ and $4.5s$ respectively on the same hardware. When employed to refine HLoc poses, we only use 5 iterations, which takes as little as 200ms.

7.5.5 Comparison with PixLoc

On our RTX4090, PixLoc requires $3.1s$ per query, independent of the scene. Our method offers greater versatility as it does not necessitate any training and can be integrated with various dense scene representations. In contrast, PixLoc requires end-to-end training and a point cloud. Tab.4 of the main chapter illustrates how our method can be effectively used as a pre-processing step in the setup detailed in [197], with different types of meshes. Furthermore, our method performs better than PixLoc as a post-processing step on HLoc poses and for night queries. On indoor datasets and small outdoor scenes, PixLoc achieves superior results, albeit

at a slower speed.

7.6 Algorithm pseudocode

The pseudocode for our algorithm is provided below in Algorithm 1. It emphasizes: (i) the *render&compare* structure inherent in our approach, (ii) the dependency of the model on the current step, and (iii) the fact that the particle filter, including the noise applied during sampling, also varies with the step.
For simplicity, it omits details such as the parallel optimization of multiple beams and other lower-level implementation aspects.
More specifically, the pseudocode illustrates a loop for each query, starting from the initial estimate (est_center, est_qvec), where in each step:

- The number of candidates (N_cand) and the noise magnitude ($noise\_t, noise\_R$) are determined based on the current step.
- The particle filter, informed by the noise magnitude and the current pose estimate, is used to generate N_cand new hypotheses.
- The model, which is adapted based on the step (the backbone is truncated at a specific layer), is used to extract features from the query image (q_feats) and the sampled candidates ($rend\_feats$).
- Using the extracted features, the sampled candidates are assigned a score by the function *rank_poses*. Finally, these scores are used to update the current pose estimate.

7.7 Conclusion

This chapter explores the feasibility of transferring generic pre-trained features to the localization task, eliminating the need for training specialized descriptors. By leveraging the robustness of dense features as estimators of *perceptual similarity*, we establish a relationship between this property and *pose similarity*. We further illustrate how this connection can be utilized to develop a refinement method integrated into a render & compare framework, combined with MonteCarlo sampling.

Experiments demonstrate that the presented **MCLoc** approach is versatile, functioning effectively in both large and small environments. It can operate independently as a refiner or complement more precise localization systems. Additionally, the results indicate that it surpasses several competing methods relying on optimized descriptors, particularly in outdoor settings.

Algorithm 1 MCLoc pose refinement

```
N ← n_steps
renderer ← load_scene_model()
for query ∈ query_list do
  est_center, est_qvec ← init_pose()
  for step ∈ 1..N do
    N_cand ← get_N_cand(step)
    noise_t, noise_R ← get_perturb_pars(step)
    sampler ← part_filter(N_cand, noise_t, noise_R)

    poses ← sampler.sample(est_center, est_qvec)
    renders ← renderer(poses)

    model ← get_model(step)
    q_feats ← extract_features(query, model)
    rend_feats ← extract_features(renders, model)

    center, qvec ← rank_poses(q_feats, rend_feats)
    est_center, est_qvec ← update(center, qvec)
  end for
end for
```

Chapter 8

Conclusions and future opportunities

8.1 Summary

This thesis advances the field of **Visual Place Recognition (VPR)** and **Visual Localization** by addressing critical challenges in *scalability, cross-domain robustness, sequential data modeling, and fine-grained pose estimation.* Moving beyond conventional retrieval-based paradigms, we introduce novel methodologies that redefine efficiency, generalization, and precision in large-scale localization systems.

Our work begins with a **systematic benchmarking framework** for VPR pipelines, identifying key architectural trade-offs and optimization strategies that enable real-world deployment without sacrificing accuracy. We demonstrate that lightweight CNNs, when combined with strategic training optimizations, can match or surpass more complex models while drastically reducing computational overhead—a crucial insight for resource-constrained applications.

Recognizing the limitations of single-image descriptors, we explore **descriptor-based sequential place recognition** through SeqVLAD, a learnable aggregation layer that explicitly models temporal coherence. This approach not only outperforms traditional sequence-matching methods but also provides a scalable solution for robotics and autonomous systems where sequential data is inherent.

To tackle **domain shifts**, one of the most persistent challenges in VPR, we re-examine the role of local feature matching for re-ranking, showing its decisive impact on robustness to night-time conditions and occlusions. Our introduction of the *SF test-night* and *SF test-occlusions* datasets exposes the limitations of existing benchmarks, revealing that even state-of-the-art retrieval systems struggle under adversarially chosen conditions.

Challenging the status quo, we propose **VPR as a classification problem**,

leveraging an Additive Angular Margin Classifier (AAMC) to bypass the computational bottlenecks of contrastive learning. This framework achieves database-scale-invariant inference, enabling real-time city-scale localization without compromising accuracy—a paradigm shift for metropolitan deployment.

Finally, we redefine fine-grained localization with a **training-free pose refiner** that generalizes across diverse scene representations (meshes, point clouds, neural radiance fields). By exploiting the inherent pose-discriminative properties of pre-trained deep features in a **render-and-compare** framework, our method outperforms specialized regression networks while remaining agnostic to underlying 3D structures.

Collectively, these contributions push the boundaries of *where, how, and under what conditions* visual localization systems can operate reliably. By unifying insights from large-scale retrieval, sequential modeling, domain adaptation, and geometric refinement, this thesis lays a foundation for next-generation localization systems that are efficient, generalizable, and universally deployable.

8.2 Limitations and future opportunities

While this thesis has advanced the state of visual place recognition and localization, several important challenges remain open. While some are inherent to the nature of the problem, others emerge as new technologies reshape the research landscape. The field now stands at an interesting crossroads, where traditional task-spefic approaches must reconcile with the potential of the recent wave of foundation models.

The fundamental challenge of generalization persists, particularly when dealing with extreme domain gaps that go beyond the scenarios addressed in this work. Seasonal changes, weather extremes, and long-term urban evolution continue to test the limits of current systems. While our re-ranking framework with local features demonstrated improved robustness, there remains a need for end-to-end solutions that bake such invariance directly into the learned representations. This becomes especially pertinent with the recent emergence of foundation models like DINOv2, which have shown remarkable generalization capabilities out-of-the-box. While initial adaptations of such models for VPR look promising, critical questions remain about how best to fine-tune them for large-scale geo-localization tasks without losing their broad generalization abilities or requiring prohibitive computational resources.

Our proposed classification-based approach successfully decoupled inference time from database size, but this came at the cost of requiring dataset-specific training. This limitation points to an important research direction: developing more flexible classification frameworks that can adapt to new environments with minimal training data, perhaps through meta-learning techniques or by leveraging the few-shot

capabilities of foundation models. Similarly, while our pose refinement method achieved impressive cross-domain generalization without any training, its performance in texture-poor indoor environments reveals an interesting tension between generality and specialization that warrants further investigation. The challenge lies in enhancing performance for specific difficult cases without sacrificing the method's current strength of working across diverse scene representations.

The rapid progress in sequential place recognition, exemplified by our SeqVLAD work, has opened new possibilities for robotics applications. Howver, real-world robotic systems might have to deal with intermittent and noisy sensor data that breaks temporal continuity, a scenario where current sequential descriptors may struggle. Future systems might need to combine the strengths of sequential processing with more sophisticated attention mechanisms that can handle irregular temporal sampling. Moreover, the increasing availability of diverse sensor modalities suggests that the next breakthrough in robustness might come from multimodal fusion, yet the optimal ways to combine visual data with LiDAR, inertial measurements, or even radio signals remain largely unexplored for VPR.

Regarding future works, perhaps most exciting are the opportunities to rethink visual localization at a more fundamental level. The success of foundation models suggests that we may be approaching a paradigm shift in how visual representations are learned and applied. Such a shift should aim to integrate 3D understanding directly into learned representations, enabling models to not only perceive but also reason about the spatial and geometric structures of the world from images, bridging the gap between 2D observations and comprehensive spatial awareness.

Concluding, the field is rapidly evolving with foundation models and multimodal sensors redefining scalability. By addressing these limitations, future work could unlock universal visual localization—where systems generalize across domains, scales, and modalities with minimal supervision. Our open-source frameworks and datasets provide the tools to accelerate this progress.

Bibliography

[1] Motilal Agrawal, Kurt Konolige, and Morten Blas. "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching". In: *European Conference on Computer Vision*. Vol. 5305. Oct. 2008, pp. 102–115.

[2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. "GSV-Cities: Toward appropriate supervised visual place recognition". In: *Neurocomputing* 513 (2022), pp. 194–203.

[3] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. "MixVPR: Feature Mixing for Visual Place Recognition". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 2998–3007.

[4] Hatem Alismail, Brett Browning, and Simon Lucey. "Photometric bundle adjustment for vision-based slam". In: *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*. Springer. 2017, pp. 324–341.

[5] Seyed Ali Amirshahi, Marius Pedersen, and Stella X. Yu. "Image Quality Assessment by Comparing CNN Features between Images". In: *Image Quality and System Performance*. 2016. URL: https://api.semanticscholar.org/CorpusID:27643834.

[6] Henrik Andreasson and Tom Duckett. "Topological localization for mobile robots using omni-directional vision and local features". In: *IFAC Proceedings Volumes* 37.8 (2004). IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004, pp. 36–41. ISSN: 1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)31947-X.

[7] Adrien Angeli et al. "Incremental vision-based topological SLAM". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 1031–1036.

[8] Asha Anoosheh et al. "Night-to-day image translation for retrieval-based localization". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5958–5964.

[9] R. Arandjelović and Andrew Zisserman. "Three things everyone should know to improve object retrieval". In: 2012, pp. 2911–2918.

[10] Relja Arandjelović et al. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1437–1451. DOI: 10.1109/TPAMI.2017.2711011.

[11] Bruno Arcanjo et al. "An Efficient and Scalable Collection of Fly-Inspired Voting Units for Visual Place Recognition in Changing Environments". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2527–2534. DOI: 10.1109/LRA.2022.3140827.

[12] Eduardo Arnold et al. "Map-free visual relocalization: Metric pose relative to a single image". In: *European Conference on Computer Vision*. Springer. 2022, pp. 690–708.

[13] Hossein Azizpour et al. "Factors of Transferability for a Generic ConvNet Representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (Nov. 2015). DOI: 10.1109/TPAMI.2015.2500224.

[14] Artem Babenko and Victor Lempitsky. "Aggregating Deep Convolutional Features for Image Retrieval". In: *ICCV* (Oct. 2015).

[15] Artem Babenko and Victor Lempitsky. "The Inverted Multi-Index". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.6 (2015), pp. 1247–1260. DOI: 10.1109/TPAMI.2014.2361319.

[16] Artem Babenko et al. "Neural Codes for Image Retrieval". In: *ArXiv* abs/1404.1777 (2014).

[17] Hernán Badino, D Huber, and Takeo Kanade. "Visual topometric localization". In: *Intelligent Vehicles Symposium*. IEEE. 2011, pp. 794–799.

[18] Tim Bailey and Hugh Durrant-Whyte. "Simultaneous localization and mapping (SLAM): Part II". In: *IEEE robotics & automation magazine* 13.3 (2006), pp. 108–117.

[19] Simon Baker, Ralph Gross, and Iain Matthews. "Lucas-Kanade 20 Years On: A Unifying Framework". In: *IJCV* 56 (2003).

[20] Giovanni Barbarani et al. "Are local features all you need for cross-domain visual place recognition?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 6155–6165.

[21] Jonathan T Barron et al. "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5855–5864.

[22] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded up robust features". In: *ECCV*. 2006.

[23] Herbert Bay et al. "Speeded-up robust features (SURF)". In: *Computer Vision and Image Understanding* 110 (June 2008), pp. 346–359. DOI: 10.1016/j.cviu.2007.09.014.

[24] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. "Is space-time attention all you need for video understanding?" In: *ICML*. Vol. 2. 3. 2021, p. 4.

[25] Gabriele Berton, Carlo Masone, and Barbara Caputo. "Rethinking Visual Geo-localization for Large-Scale Applications". In: *CVPR*. 2022.

[26] Gabriele Berton et al. "Adaptive-Attentive Geolocalization From Few Queries: A Hybrid Approach". In: *IEEE Winter Conference on Applications of Computer Vision*. 2021, pp. 2918–2927.

[27] Gabriele Berton et al. "Deep Visual Geo-localization Benchmark". In: *CVPR*. 2022.

[28] Gabriele Berton et al. "Earthloc: Astronaut photography localization by indexing earth from space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 12754–12764.

[29] Gabriele Berton et al. "Earthmatch: Iterative coregistration for fine-grained localization of astronaut photography". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 4264–4274.

[30] Gabriele Berton et al. "EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 11080–11090.

[31] Gabriele Berton et al. "Jist: Joint image and sequence training for sequential visual place recognition". In: *IEEE Robotics and Automation Letters* 9.2 (2023), pp. 1310–1317.

[32] Gabriele Berton et al. "MeshVPR: Citywide Visual Place Recognition Using 3D Meshes". In: *European Conference on Computer Vision*. Springer. 2024, pp. 321–339.

[33] Gabriele Berton et al. "Viewpoint Invariant Dense Matching for Visual Geolocalization". In: *IEEE International Conference on Computer Vision*. 2021, pp. 12169–12178.

[34] Michael Bloesch et al. "Learning Meshes for Dense Visual SLAM". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 5854–5863. DOI: 10.1109/ICCV.2019.00595.

[35] Georg Bökman and Fredrik Kahl. "A case for using rotation invariant features in state of the art feature matchers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5110–5119.

[36] Georg Bökman et al. “Steerers: A framework for rotation equivariant keypoint descriptors”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 4885–4895.

[37] Eric Brachmann and Carsten Rother. “Learning Less is More-6D Camera Localization via 3D Surface Regression”. In: *CVPR*. 2018.

[38] Eric Brachmann and Carsten Rother. “Visual Camera Re-Localization from RGB and RGB-D Images Using DSAC”. In: *TPAMI* (2021).

[39] Eric Brachmann et al. “DSAC-Differentiable RANSAC for Camera Localization”. In: *CVPR*. 2017.

[40] Eric Brachmann et al. “On the limits of pseudo ground truth in visual camera re-localisation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6218–6228.

[41] Jan Brejcha et al. “LandscapeAR: Large Scale Outdoor Augmented Reality by Matching Photographs with Terrain Models Using Learned Descriptors”. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 295–312. ISBN: 978-3-030-58525-9. DOI: 10.1007/978-3-030-58526-6_18. URL: https://doi.org/10.1007/978-3-030-58526-6_18.

[42] Benjamin Busam, Tolga Birdal, and Nassir Navab. “Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions”. In: *Proceedings of the IEEE international conference on computer vision workshops*. 2017, pp. 2436–2445.

[43] B. Cao, A. Araujo, and J. Sim. “Unifying Deep Local and Global Features for Image Search”. In: *European Conference on Computer Vision*. Springer Int. Publishing, 2020, pp. 726–743. ISBN: 978-3-030-58564-8.

[44] Mathilde Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”. In: *IEEE International Conference on Computer Vision*. 2021, pp. 9650–9660.

[45] Tommaso Cavallari et al. “Let’s Take This Online: Adapting Scene Coordinate Regression Network Predictions for Online RGB-D Camera Relocalisation”. In: *3DV*. 2019.

[46] Tommaso Cavallari et al. “Real-Time RGB-D Camera Pose Estimation in Novel Scenes Using a Relocalisation Cascade”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10 (2020), pp. 2465–2477. DOI: 10.1109/TPAMI.2019.2915068.

[47] Marvin Chancán et al. “A Hybrid Compact Neural Architecture for Visual Place Recognition”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 993–1000. DOI: 10.1109/LRA.2020.2967324.

[48] D. M. Chen et al. "City-scale landmark identification on mobile devices". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 737–744. DOI: 10.1109/CVPR.2011.5995610.

[49] Hongkai Chen et al. "Aspanformer: Detector-free image matching with adaptive span transformer". In: *European Conference on Computer Vision*. Springer. 2022, pp. 20–36.

[50] Shuai Chen et al. "Dfnet: Enhance absolute pose regression with direct feature matching". In: *European Conference on Computer Vision*. Springer. 2022, pp. 1–17.

[51] Shuai Chen et al. "Refinement for Absolute Pose Regression with Neural Feature Synthesis". In: *arXiv preprint arXiv:2303.10087* (2023).

[52] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[53] Zetao Chen et al. "Deep learning features at scale for visual place recognition". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3223–3230. DOI: 10.1109/ICRA.2017.7989366.

[54] Zetao Chen et al. "Only look once, mining distinctive landmarks from convnet for visual place recognition". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 9–16.

[55] A. Chiuso and S. Soatto. "Monte Carlo filtering on Lie groups". In: *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*. Vol. 1. 2000, 304–309 vol.1. DOI: 10.1109/CDC.2000.912777.

[56] Changhyun Choi and Henrik Christensen. "Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features". In: *Intl. Jour. of Robotics Research* 33 (Apr. 2012). DOI: 10.1177/0278364912437213.

[57] Ondřej Chum, Jiří Matas, and Josef Kittler. "Locally optimized RANSAC". In: *Joint Pattern Recognition Symposium*. Springer. 2003, pp. 236–243.

[58] Titus Cieslewski, Michael Bloesch, and Davide Scaramuzza. "Matching features without descriptors: implicitly matched interest points". In: *arXiv preprint arXiv:1811.10681* (2018).

[59] Gabriela Csurka et al. "Visual categorization with bags of keypoints". In: *European Conference on Computer Vision*. Vol. Vol. 1. Jan. 2004.

[60] Mark Cummins and Paul Newman. "Appearance-only SLAM at large scale with FAB-MAP 2.0". In: *The International Journal of Robotics Research* 30.9 (2011), pp. 1100–1123. DOI: 10.1177/0278364910385483.

[61] Mark Cummins and Paul Newman. "FAB-MAP: Probabilistic localization and mapping in the space of appearance". In: *The International Journal of Robotics Research* 27.6 (2008), pp. 647–665.

[62] Tomasz Malisiewicz Daniel DeTone and Andrew Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *Computer Vision and Pattern Recognition Workshop*. 2018.

[63] Jiankang Deng, J. Guo, and S. Zafeiriou. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4685–4694.

[64] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *CVPR Workshop on Deep Learning for Visual SLAM*. 2018.

[65] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "Toward geometric deep slam". In: *arXiv preprint arXiv:1707.07410* (2017).

[66] Mingyu Ding et al. "CamNet: Coarse-to-fine retrieval for camera re-localization". In: *ICCV*. 2019.

[67] A.-D. Doan et al. "Scalable Place Recognition Under Appearance Change for Autonomous Driving". In: *IEEE International Conference on Computer Vision*. 2019, pp. 9319–9328.

[68] Carl Doersch, Abhinav Gupta, and Alexei A Efros. "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.

[69] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ArXiv* abs/2010.11929 (2021).

[70] Randal Douc and Olivier Cappé. "Comparison of resampling schemes for particle filtering". In: *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.* IEEE. 2005, pp. 64–69.

[71] Juan Du, Rui Wang, and Daniel Cremers. "DH3D: Deep Hierarchical 3D Descriptors for Robust Large-Scale 6DoF Relocalization". In: *Comput. Vis. – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham, Springer International Publishing, 2020, pp. 744–762. ISBN: 978-3-030-58548-8.

[72] Abhilash Durgam et al. "Cross-view geo-localization: a survey". In: *IEEE Access* (2024).

[73] Hugh Durrant-Whyte and Tim Bailey. "Simultaneous localization and mapping: part I". In: *IEEE robotics & automation magazine* 13.2 (2006), pp. 99–110.

[74] Mihai Dusmanu et al. “D2-Net: A Trainable CNN for Joint Detection and Description of Local Features”. In: *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[75] Mihai Dusmanu et al. “D2-Net: A Trainable CNN for Joint Detection and Description of Local Features”. In: *CVPR*. 2019.

[76] Mattia Dutto et al. “Collaborative Visual Place Recognition through Federated Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 4215–4225.

[77] Johan Edstedt et al. “DeDoDe: Detect, don’t describe—Describe, don’t detect for local feature matching”. In: *2024 International Conference on 3D Vision (3DV)*. IEEE. 2024, pp. 148–157.

[78] Johan Edstedt et al. “DKM: Dense Kernelized Feature Matching for Geometry Estimation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023.

[79] Johan Edstedt et al. “RoMa: Revisiting Robust Losses for Dense Feature Matching”. In: *arXiv preprint arXiv:2305.15404* (2023).

[80] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry”. In: *TPAMI* 40.3 (2017), pp. 611–625.

[81] Jakob Engel, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-Scale Direct Monocular SLAM”. In: *ECCV*. 2014.

[82] José M. Fácil et al. “Condition-Invariant Multi-View Place Recognition”. In: *ArXiv* abs/1902.09516 (2019).

[83] Martin A Fischler and Robert C Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[84] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. “Markov localization for mobile robots in dynamic environments”. In: *Journal of artificial intelligence research* 11 (1999), pp. 391–427.

[85] Dieter Fox et al. “Monte carlo localization: Efficient position estimation for mobile robots”. In: *Aaai/iaai* 1999.343-349 (1999), pp. 2–2.

[86] Dieter Fox et al. “Particle Filters for Mobile Robot Localization”. In: *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer New York, 2001, pp. 401–428. DOI: 10.1007/978-1-4757-3437-9_19. URL: https://doi.org/10.1007/978-1-4757-3437-9_19.

[87] Matthew Gadd, D. Martini, and P. Newman. "Look Around You: Sequence-based Radar Place Recognition with Learned Rotational Invariance". In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)* (2020), pp. 270–276.

[88] Dorian Galvez-López and Juan D. Tardos. "Bags of Binary Words for Fast Place Recognition in Image Sequences". In: *IEEE Transactions on Robotics* 28.5 (2012), pp. 1188–1197.

[89] Fei Gao et al. "DeepSim: Deep Similarity for Image Quality Assessment". In: *Neurocomputing* 257 (Feb. 2017). DOI: 10.1016/j.neucom.2017.01.054.

[90] S. Garg, N. Suenderhauf, and M. Milford. "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views". In: *The International Journal of Robotics Research* (2019). DOI: 10.1177/0278364919839761.

[91] Sourav Garg, Tobias Fischer, and Michael Milford. "Where is your place, visual place recognition?" In: *arXiv preprint arXiv:2103.06443* (2021).

[92] Sourav Garg and Michael Milford. "SeqNet: Learning Descriptors for Sequence-Based Hierarchical Place Recognition". In: *IEEE Robotics and Automation Letters* 6 (2021), pp. 4305–4312.

[93] Sourav Garg, Madhu Vankadari, and Michael Milford. "SeqMatchNet: Contrastive Learning with Sequence Matching for Place Recognition & Relocalization". In: *5th Annual Conference on Robot Learning*. 2021.

[94] Sourav Garg et al. "Delta Descriptors: Change-Based Place Representation for Robust Visual Localization". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5120–5127. DOI: 10.1109/LRA.2020.3005627.

[95] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[96] Yixiao Ge et al. "Self-supervising Fine-Grained Region Similarities for Large-Scale Image Localization". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 369–386.

[97] Hugo Germain et al. "Feature Query Networks: Neural Surface Description for Camera Pose Refinement". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, pp. 5067–5077. DOI: 10.1109/CVPRW56347.2022.00555.

[98] Albert Gordo et al. "Deep Image Retrieval: Learning Global Representations for Image Search". In: *ECCV*. 2016.

[99] Albert Gordo et al. "End-to-end learning of deep visual representations for image retrieval". In: *IJCV* 124.2 (2017), pp. 237–254.

[100] Petr Gronát et al. "Learning and Calibrating Per-Location Classifiers for Visual Place Recognition". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 907–914. DOI: 10.1109/CVPR.2013.122.

[101] Vladimir Guzov et al. "Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4318–4329.

[102] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[103] Ali Hassani et al. "Escaping the Big Data Paradigm with Compact Transformers". In: *ArXiv* abs/2104.05704 (2021).

[104] S. Hausler, A. Jacobson, and M. Milford. "Multi-Process Fusion: Visual Place Recognition Using Multiple Image Processing Methods". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1924–1931.

[105] Stephen Hausler et al. "Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14141–14152.

[106] James Hays and Alexei A. Efros. "im2gps: estimating geographic information from a single image". In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[107] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[108] Kin Leong Ho and Paul Newman. "Detecting Loop Closure with Scene Sequences". In: *International Journal of Computer Vision* 74.3 (2007), pp. 261–286. DOI: 10.1007/s11263-006-0020-1.

[109] Ziyang Hong et al. "TextPlace: Visual Place Recognition and Topological Localization Through Reading Scene Texts". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[110] Sixing Hu and Gim Hee Lee. "Image-based geo-localization using satellite imagery". In: *International Journal of Computer Vision* 128.5 (2020), pp. 1205–1219.

[111] Dihe Huang et al. "Adaptive assignment for geometry aware local feature matching". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 5425–5434.

[112] Nicolas Hudson et al. "Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO Data61's approach to the DARPA subterranean challenge". In: *arXiv preprint arXiv:2104.09053* (2021).

[113] Martin Humenberger et al. "Investigating the Role of Image Retrieval for Visual Localization: An Exhaustive Benchmark". In: *International Journal of Computer Vision* 130.7 (2022), pp. 1811–1836.

[114] Martin Humenberger et al. "Robust image retrieval-based visual localization using kapture". In: *arXiv preprint arXiv:2007.13867* (2020).

[115] Sarah Ibrahimi et al. "Inside Out Visual Place Recognition". In: *British Machine Vision Conference*. 2021.

[116] Wardah Inam. "Particle filter based self-localization using visual landmarks and image database". In: *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*. 2009, pp. 246–251. DOI: 10.1109/CIRA.2009.5423198.

[117] Arnold Irschara et al. "From structure-from-motion point clouds to fast location recognition". In: *CVPR*. 2009.

[118] Mike Izbicki, Evangelos E Papalexakis, and Vassilis J Tsotras. "Exploiting the earth's spherical geometry to geolocate images". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II*. Springer. 2020, pp. 3–19.

[119] Sergio Izquierdo and Javier Civera. "Close, But Not There: Boosting Geographic Distance Sensitivity in Visual Place Recognition". In: *European Conference on Computer Vision*. Springer. 2024, pp. 240–257.

[120] Sergio Izquierdo and Javier Civera. "Optimal transport aggregation for visual place recognition". In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2024, pp. 17658–17668.

[121] H. Jégou, M. Douze, and C. Schmid. "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search". In: *European Conference on Computer Vision*. Ed. by D. Forsyth, P. Torr, and A. Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 304–317.

[122] H. Jégou and Andrew Zisserman. "Triangulation Embedding and Democratic Aggregation for Image Search". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 3310–3317.

[123] Herve Jégou, Matthijs Douze, and Cordelia Schmid. "Product Quantization for Nearest Neighbor Search". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1 (2011), pp. 117–128. DOI: 10.1109/TPAMI.2010.57.

[124] Hervé Jégou et al. "Aggregating Local Image Descriptors into Compact Codes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (Dec. 2011). DOI: 10.1109/TPAMI.2011.235.

[125] Yuhe Jin et al. “Image matching across wide baselines: From paper to practice”. In: *International Journal of Computer Vision* 129.2 (2021), pp. 517–547.

[126] Jeff Johnson, Matthijs Douze, and Hervé Jégou. “Billion-scale similarity search with GPUs”. In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[127] Peter Karkus, David Hsu, and Wee Sun Lee. “Particle filter networks with application to visual localization”. In: *Conference on robot learning*. PMLR. 2018, pp. 169–178.

[128] Michael M. Kazhdan and Hugues Hoppe. “Screened poisson surface reconstruction”. In: *ACM Trans. Graph.* 32 (2013), 29:1–29:13. URL: https://api.semanticscholar.org/CorpusID:1371704.

[129] Nikhil Keetha et al. “Anyloc: Towards universal visual place recognition”. In: *IEEE Robotics and Automation Letters* 9.2 (2023), pp. 1286–1293.

[130] Alex Kendall and Roberto Cipolla. “Geometric loss functions for camera pose regression with deep learning”. In: *CVPR*. 2017.

[131] Alex Kendall, Matthew Grimes, and Roberto Cipolla. “PoseNet: A convolutional network for real-time 6-DoF camera relocalization”. In: *ICCV*. 2015.

[132] Bernhard Kerbl et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering”. In: *ACM Transactions on Graphics* 42.4 (2023). URL: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[133] A. Khaliq et al. “A Holistic Visual Place Recognition Approach Using Lightweight CNNs for Significant ViewPoint and Appearance Changes”. In: *IEEE Transactions on Robotics* 36.2 (2020), pp. 561–569.

[134] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. “Learned Contextual Feature Reweighting for Image Geo-Localization”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3251–3260.

[135] Jongyoo Kim and Sanghoon Lee. “Deep Learning of Human Visual Sensitivity in Image Quality Assessment Framework”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[136] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).

[137] Kurt Konolige and Motilal Agrawal. “FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping”. In: *IEEE Transactions on Robotics* 24 (2008), pp. 1066–1077. URL: https://api.semanticscholar.org/CorpusID:78652.

[138] Giorgos Kordopatis-Zilos et al. "Leveraging EfficientNet and Contrastive Learning for Accurate Global-scale Location Estimation". In: *ACM International Conference on Multimedia Retrieval* (2021).

[139] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105.

[140] Junghyun Kwon and Frank C. Park. "Visual tracking via particle filtering on the affine group". In: *2008 International Conference on Information and Automation*. 2008, pp. 997–1002. DOI: 10.1109/ICINFA.2008.4608144.

[141] Junghyun Kwon et al. "Particle filtering on the Euclidean group: Framework and applications". In: *Robotica* 25 (Nov. 2007), pp. 725–737. DOI: 10.1017/S0263574707003529.

[142] Yann Labbé et al. "Megapose: 6d pose estimation of novel objects via render & compare". In: *arXiv preprint arXiv:2212.06870* (2022).

[143] Viktor Larsson et al. "Revisiting Radial Distortion Absolute Pose". In: *ICCV*. 2019.

[144] Seongwon Lee et al. "Correlation Verification for Image Retrieval". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5374–5384.

[145] Vincent Lepetit and Pascal Fua. *Monocular Model-Based 3D Tracking of Rigid Objects: A Survey*. 2005.

[146] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. "Grounding image matching in 3d with mast3r". In: *European Conference on Computer Vision*. Springer. 2024, pp. 71–91.

[147] Kenneth Levenberg. "A method for the solution of certain non-linear problems in least squares". In: *Quarterly of applied mathematics* 2.2 (1944), pp. 164–168.

[148] Maria Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. "Data-efficient large scale place recognition with graded similarity supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 23487–23496.

[149] Heshan Li et al. "CaseVPR: Correlation-Aware Sequential Embedding for Sequence-to-Frame Visual Place Recognition". In: *IEEE Robotics and Automation Letters* (2025).

[150] Xiaotian Li et al. "Hierarchical Scene Coordinate Classification and Regression for Visual Localization". In: *CVPR*. 2020.

[151] Yi Li et al. "Deepim: Deep iterative matching for 6d pose estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 683–698.

[152] Yunpeng Li et al. "Worldwide pose estimation using 3D point clouds". In: *ECCV*. 2012.

[153] Zhengqi Li and Noah Snavely. "MegaDepth: Learning Single-View Depth Prediction from Internet Photos". In: *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[154] Yunzhi Lin et al. "Parallel inversion of neural radiance fields for robust pose estimation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 9377–9384.

[155] Chris Linegar, Winston Churchill, and Paul Newman. "Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 90–97.

[156] Dongfang Liu et al. "DenserNet: Weakly Supervised Visual Localization Using Multi-scale Feature Aggregation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), pp. 6101–6109.

[157] Jianlin Liu et al. "NeRF-Loc: Visual Localization with Conditional Neural Radiance Field". In: *arXiv preprint arXiv:2304.07979* (2023).

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

[159] Liu Liu, Hongdong li, and Yuchao Dai. "Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map". In: Oct. 2017, pp. 2391–2400. DOI: 10.1109/ICCV.2017.260.

[160] Ilya Loshchilov and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983* (2016).

[161] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *IJCV* 60.2 (2004), pp. 91–110.

[162] Stephanie Lowry et al. "Visual Place Recognition: A Survey". In: *IEEE Transactions on Robotics* 32.1 (2016), pp. 1–19. DOI: 10.1109/TRO.2015.2496823.

[163] Kan Luo et al. "3D point cloud-based place recognition: a survey". In: *Artificial Intelligence Review* 57.4 (2024). ISSN: 0269-2821.

[164] W. Maddern et al. "1 Year, 1000km: The Oxford RobotCar Dataset". In: *The International Journal of Robotics Research* (2017).

[165] Dominic Maggio et al. "Loc-nerf: Monte carlo localization using neural radiance fields". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 4018–4025.

[166] Yu A. Malkov and D. A. Yashunin. "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2020), pp. 824–836.

[167] Donald W Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441.

[168] Julieta Martinez et al. "Pit30m: A benchmark for global localization in the age of self-driving cars". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 4477–4484.

[169] Carlo Masone and Barbara Caputo. "A Survey on Deep Visual Place Recognition". In: *IEEE Access* 9 (2021), pp. 19516–19547.

[170] Riccardo Mereu et al. "Learning Sequential Descriptors for Sequence-Based Visual Place Recognition". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 10383–10390. DOI: 10.1109/LRA.2022.3194310.

[171] B Mildenhall et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *European conference on computer vision*. 2020.

[172] Michael Milford and G. Wyeth. "Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System". In: *IEEE Transactions on Robotics* 24 (2008), pp. 1038–1053.

[173] Michael Milford et al. "Place recognition with event-based cameras and a neural implementation of SeqSLAM". In: *arXiv preprint arXiv:1505.04548* (2015).

[174] Michael J. Milford and Gordon. F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights". In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 1643–1649. DOI: 10.1109/ICRA.2012.6224623.

[175] Eva Mohedano et al. "Saliency Weighted Convolutional Features for Instance Search". In: *2018 International Conference on Content-Based Multimedia Indexing (CBMI)* (2018), pp. 1–6.

[176] Jong Hak Moon, Wonjae Kim, and E. Choi. "Correlation between Alignment-Uniformity and Performance of Dense Contrastive Representations". In: *British Machine Vision Conference*. 2022.

[177] Arthur Moreau et al. "CROSSFIRE: Camera Relocalization On Self-Supervised Features from an Implicit Representation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 252–262.

[178] Arthur Moreau et al. "Lens: Localization enhanced by nerf synthesis". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1347–1356.

[179] Javier Morlana, Juan D Tardós, and JMM Montiel. "ColonMapper: topological mapping and localization for colonoscopy". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 6329–6336.

[180] Javier Morlana, Juan D Tardós, and José MM Montiel. "Topological SLAM in colonoscopies leveraging deep features and topological priors". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2024, pp. 733–743.

[181] Markus S. Mueller et al. "Image-to-image translation for enhanced feature matching, image retrieval and visual localization". In: *ISPRS annals* IV-2/W7 (2019), pp. 111–119. ISSN: 2194-9050. DOI: 10.5194/isprs-annals-IV-2-W7-111-2019.

[182] Thomas Müller et al. "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–15.

[183] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. "Geolocation estimation of photos using a hierarchical model and scene classification". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 563–579.

[184] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163.

[185] A. C. Murillo, J. J. Guerrero, and C. Sagues. "SURF features for efficient robot localization with omnidirectional images". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 3901–3907. DOI: 10.1109/ROBOT.2007.364077.

[186] A. C. Murillo and J. Kosecka. "Experiments in place recognition using gist panoramas". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. 2009, pp. 2196–2203. DOI: 10.1109/ICCVW.2009.5457552.

[187] Tayyab Naseer and Wolfram Burgard. "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments". In: *IROS*. 2017.

[188] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss. "Robust Visual Localization Across Seasons". In: *IEEE Transactions on Robotics* 34.2 (2018), pp. 289–302. DOI: 10.1109/TRO.2017.2788045.

[189] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss. "Robust visual localization across seasons". In: *IEEE Transactions on Robotics* 34.2 (2018), pp. 289–302.

[190] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *2011 International Conference on Computer Vision*. 2011, pp. 2320–2327. DOI: 10.1109/ICCV.2011.6126513.

[191] Hyeonwoo Noh et al. "Large-Scale Image Retrieval with Attentive Deep Local Features". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 3476–3485. DOI: 10.1109/ICCV.2017.374.

[192] Alaaeldin El-Nouby et al. "Training Vision Transformers for Image Retrieval". In: *ArXiv* abs/2102.05644 (2021).

[193] A. Oliva and A. Torralba. "Building the gist of a scene: the role of global image features in recognition." In: *Progress in brain research* 155 (2006), pp. 23–36.

[194] Eng-Jon Ong, Sameed Husain, and Miroslaw Bober. "Siamese network of deep fisher-vector descriptors for image retrieval". In: *arXiv preprint arXiv:1702.00338* (2017).

[195] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023.

[196] Linfei Pan et al. "Global structure-from-motion revisited". In: *European Conference on Computer Vision*. Springer. 2024, pp. 58–77.

[197] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. "MeshLoc: Mesh-Based Visual Localization". In: *European Conference on Computer Vision (ECCV)*. 2022.

[198] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. "Visual Localization using Imperfect 3D Models from the Internet". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13175–13186.

[199] Emilio Parisotto et al. "Global pose estimation with an attention-based recurrent network". In: *Proceedings of the ieee conference on computer vision and pattern recognition workshops*. 2018, pp. 237–246.

[200] Guohao Peng et al. "Attentional Pyramid Pooling of Salient Visual Residuals for Place Recognition". In: *IEEE International Conference on Computer Vision*. 2021, pp. 885–894.

[201] Guohao Peng et al. "Semantic Reinforced Attention Learning for Visual Place Recognition". In: *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*. IEEE, 2021, pp. 13415–13422.

[202] Edward Pepperell, Peter I. Corke, and Michael J. Milford. "All-environment visual place recognition with SMART". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 1612–1618. DOI: 10.1109/ICRA.2014.6907067.

[203] Florent Perronnin et al. "Large-scale image retrieval with compressed Fisher vectors". In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2010, pp. 3384–3391. DOI: 10.1109/CVPR.2010.5540009.

[204] Mikael Persson and Klas Nordberg. "Lambda twist: An accurate fast robust perspective three point (p3p) solver". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 318–332.

[205] James Philbin et al. "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2008.

[206] James Philbin et al. "Object retrieval with large vocabularies and fast spatial matching." In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2007. ISBN: 1-4244-1179-3. URL: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2007.html#PhilbinCISZ07.

[207] *Phototourism Challenge, CVPR 2019 Image Matching Workshop*. https://image-matching-workshop.github.io. Accessed November 8, 2019.

[208] Nathan Piasco et al. "A survey on Visual-Based Localization: On the benefit of heterogeneous data". In: *Pattern Recognit.* 74 (2018), pp. 90–109.

[209] Nicolas Pielawski et al. "CoMIR: Contrastive multimodal image representation for registration". In: *Advances in neural information processing systems* 33 (2020), pp. 18433–18444.

[210] Maxime Pietrantoni et al. "SegLoc: Learning Segmentation-Based Representations for Privacy-Preserving Visual Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 15380–15391.

[211] Noé Pion et al. "Benchmarking Image Retrieval for Visual Localization". In: *2020 International Conference on 3D Vision (3DV)*. 2020, pp. 483–494.

[212] Mirco Planamente et al. "Toward Human-Robot Cooperation: Unsupervised Domain Adaptation for Egocentric Action Recognition". In: *Human-Friendly Robotics 2022*. 2023.

[213] Christian Poglitsch et al. "[POSTER] A Particle Filter Approach to Outdoor Localization Using Image-Based Rendering". In: *2015 IEEE International Symposium on Mixed and Augmented Reality*. 2015, pp. 132–135. DOI: 10.1109/ISMAR.2015.39.

[214] Gerard Pons-Moll et al. "Interaction Replica: Tracking human-object interaction and scene changes from human motion". In: (2023).

[215] Filip Radenović, Giorgos Tolias, and O. Chum. "CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples". In: *ECCV*. 2016.

[216] Filip Radenović, Giorgos Tolias, and Ondřej Chum. "Fine-Tuning CNN Image Retrieval with No Human Annotation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7 (2019), pp. 1655–1668. DOI: 10.1109/TPAMI.2018.2846566.

[217] A. Razavian et al. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014), pp. 512–519.

[218] A. Razavian et al. "Visual Instance Retrieval with Deep Convolutional Networks". In: *CoRR* abs/1412.6574 (2015).

[219] Christian Reiser et al. "KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs". In: *International Conference on Computer Vision (ICCV)*. 2021.

[220] Jerome Revaud et al. "R2D2: Repeatable and Reliable Detector and Descriptor". In: *NeurIPS*. 2019.

[221] Jérôme Revaud et al. "Learning With Average Precision: Training Image Retrieval With a Listwise Loss". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 5106–5115.

[222] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF." In: *ICCV*. 2011.

[223] Paul-Edouard Sarlin et al. "Back to the feature: Learning robust camera localization from pixels to pose". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 3247–3257.

[224] Paul-Edouard Sarlin et al. "From Coarse to Fine: Robust Hierarchical Localization at Large Scale". In: *CVPR*. 2019.

[225] Paul-Edouard Sarlin et al. "Leveraging deep visual descriptors for hierarchical efficient localization". In: *Conference on Robot Learning*. PMLR. 2018, pp. 456–465.

[226] Paul-Edouard Sarlin et al. "SuperGlue: Learning Feature Matching with Graph Neural Networks". In: *CVPR*. 2020.

[227] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. "Efficient & effective prioritized matching for large-scale image-based localization". In: *TPAMI* 39.9 (2016), pp. 1744–1756.

[228] Torsten Sattler et al. "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions". In: *CVPR*. 2018.

[229] Torsten Sattler et al. "Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. USA: IEEE Computer Society, 2015, pp. 2102–2110. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.243. URL: https://doi.org/10.1109/ICCV.2015.243.

[230] Torsten Sattler et al. "Image Retrieval for Image-Based Localization Revisited." In: *BMVC*. 2012.

[231] Torsten Sattler et al. "Large-Scale Location Recognition And The Geometric Burstiness Problem". In: *CVPR*. 2016.

[232] Grant Schindler, Matthew Brown, and Richard Szeliski. "City-Scale Location Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2007.

[233] Johannes L Schönberger et al. "Semantic visual localization". In: *CVPR*. 2018.

[234] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *CVPR*. 2016.

[235] Johannes Lutz Schönberger et al. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *ECCV*. 2016.

[236] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. "BAD SLAM: Bundle Adjusted Direct RGB-D SLAM". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 134–144. DOI: 10.1109/CVPR.2019.00022.

[237] Stefan Schubert and Peer Neubert. "What makes visual place recognition easy or hard?" In: *ArXiv* abs/2106.12671 (2021).

[238] Stephen Se, David Lowe, and Jim Little. "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks". In: *The International Journal of Robotics Research* 21.8 (2002), pp. 735–758. DOI: 10.1177/027836402761412467.

[239] Paul Hongsuck Seo et al. "CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps". In: *ECCV*. 2018.

[240] Zachary Seymour et al. "Semantically-Aware Attentive Neural Embeddings for Image-based Visual Localization". In: *ArXiv* abs/1812.03402 (2018).

[241] Qi Shan et al. "Accurate Geo-Registration by Ground-to-Aerial Image Matching". In: *Proceedings of the 2014 2nd International Conference on 3D Vision - Volume 01*. 3DV '14. USA: IEEE Computer Society, 2014, pp. 525–532. ISBN: 9781479970001. DOI: 10.1109/3DV.2014.69. URL: https://doi.org/10.1109/3DV.2014.69.

[242] Yoli Shavit, Ron Ferens, and Yosi Keller. "Learning Multi-Scene Absolute Pose Regression with Transformers". In: *arXiv preprint arXiv:2103.11468* (2021).

[243] Jamie Shotton et al. "Scene coordinate regression forests for camera relocalization in RGB-D images". In: *CVPR*. 2013.

[244] Edgar Simo-Serra et al. "Discriminative learning of deep convolutional feature point descriptors". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 118–126.

[245] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[246] Gautam Singh. "Visual Loop Closing using Gist Descriptors in Manhattan World". In: 2010.

[247] Krishna Kumar Singh and Yong Jae Lee. "Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-Supervised Object and Action Localization". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 3544–3553.

[248] Sivic and Zisserman. "Video Google: a text retrieval approach to object matching in videos". In: *IEEE International Conference on Computer Vision*. 2003, 1470–1477 vol.2. DOI: 10.1109/ICCV.2003.1238663.

[249] Yash Srivastava, Vaishnav Murali, and Shiv Ram Dubey. *A Performance Comparison of Loss Functions for Deep Face Recognition*. 2019. arXiv: 1901.05903 [cs.CV].

[250] Alex Stoken and Kenton Fisher. "Find My Astronaut Photo: Automated Localization and Georectification of Astronaut Photography". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2023, pp. 6196–6205.

[251] Elena Stumm, Christopher Mei, and Simon Lacroix. "Probabilistic place recognition with covisibility maps". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 4158–4163.

[252] Edgar Sucar et al. "iMAP: Implicit mapping and positioning in real-time". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6229–6238.

[253] Niko Suenderhauf et al. “Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free”. In: *Robotics: Science and Systems XI*. Ed. by D Hsu. Robotics: Science and Systems Conference, 2015, pp. 1–10.

[254] Jiaming Sun et al. “LoFTR: Detector-free local feature matching with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8922–8931.

[255] Hajime Taira et al. “InLoc: Indoor Visual Localization with Dense Matching and View Synthesis”. In: *TPAMI* (2019).

[256] Hajime Taira et al. “Is this the right place? geometric-semantic pose verification for indoor visual localization”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4373–4383.

[257] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. “Instance-level Image Retrieval using Reranking Transformers”. In: *IEEE International Conference on Computer Vision*. 2021.

[258] Matthew Tancik et al. “Nerfstudio: A Modular Framework for Neural Radiance Field Development”. In: *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH ’23. 2023.

[259] Shitao Tang et al. “Quadtree attention for vision transformers”. In: *arXiv preprint arXiv:2201.02767* (2022).

[260] Jonas Theiner, Eric Müller-Budack, and Ralph Ewerth. “Interpretable Semantic Photo Geolocation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 750–760.

[261] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2005. URL: http://www.amazon.de/gp/product/0262201623/102-8479661-9831324?v=glance&n=283155&n=507846&s=books&v=glance.

[262] Yurun Tian, Bin Fan, and Fuchao Wu. “L2-net: Deep learning of discriminative patch descriptor in euclidean space”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 661–669.

[263] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. “Learning and aggregating deep local descriptors for instance-level recognition”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 460–477.

[264] Giorgos Tolias, R. Sicre, and H. Jégou. “Particular object retrieval with integral max-pooling of CNN activations”. In: *CoRR* abs/1511.05879 (2016).

[265] A. Torii et al. "24/7 Place Recognition by View Synthesis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.2 (2018), pp. 257–271.

[266] A. Torii et al. "Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2021), pp. 814–829.

[267] Akihiko Torii et al. "Visual Place Recognition with Repetitive Structures". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.11 (2015), pp. 2346–2359. DOI: 10.1109/TPAMI.2015.2409868.

[268] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *International conference on machine learning*. PMLR. 2021, pp. 10347–10357.

[269] Du Tran et al. "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 6450–6459.

[270] Gabriele Trivigno et al. "Divide&Classify: Fine-Grained Classification for City-Wide Visual Geo-Localization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 11142–11152.

[271] Gabriele Trivigno et al. "The Unreasonable Effectiveness of Pre-Trained Features for Camera Pose Refinement". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 12786–12798.

[272] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. "DISK: Learning local features with policy gradient". In: *NeurIPS*. 2020.

[273] Mikaela Angelina Uy and Gim Hee Lee. "PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4470–4479. DOI: 10.1109/CVPR.2018.00470.

[274] Jonathan Ventura et al. "P1ac: Revisiting absolute pose from a single affine correspondence". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 19751–19761.

[275] Nam Vo, Nathan Jacobs, and James Hays. "Revisiting IM2GPS in the Deep Learning Era". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2640–2649. DOI: 10.1109/ICCV.2017.286.

[276] Lukas Von Stumberg et al. "GN-Net: The Gauss-Newton loss for multi-weather relocalization". In: *RA-L* 5.2 (2020), pp. 890–897.

[277] Lukas Von Stumberg et al. "LM-Reloc: Levenberg-Marquardt Based Direct Visual Relocalization". In: *3DV*. 2020.

[278] O. Vysotska and C. Stachniss. "Effective Visual Place Recognition Using Multi-Sequence Maps". In: *IEEE Robotics and Automation Letters* 4 (2019), pp. 1730–1736.

[279] Olga Vysotska and Cyrill Stachniss. "Lazy Data Association for Image Sequences Matching Under Substantial Appearance Changes". In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 1–8. DOI: 10.1109/LRA.2015.2512936.

[280] Olga Vysotska and Cyrill Stachniss. "Relocalization under substantial appearance changes using hashing". In: *Proceedings of the IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles, Vancouver, BC, Canada*. Vol. 24. 2017.

[281] Olga Vysotska et al. "Efficient and effective matching of image sequences under substantial appearance changes exploiting GPS priors". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 2774–2779.

[282] Hao Wang et al. "CosFace: Large Margin Cosine Loss for Deep Face Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 5265–5274.

[283] Qing Wang et al. "Matchformer: Interleaving attention in transformers for feature matching". In: *Proceedings of the Asian Conference on Computer Vision*. 2022, pp. 2746–2762.

[284] Ruotong Wang et al. "TransVPR: Transformer-Based Place Recognition With Multi-Level Attention Aggregation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 13648–13657.

[285] Shuzhe Wang et al. "Dust3r: Geometric 3d vision made easy". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 20697–20709.

[286] Tongzhou Wang and Phillip Isola. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". In: *Advances in Neural Information Processing System* (2020).

[287] Xun Wang et al. "Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5022–5030.

[288] Ziqi Wang et al. "Attention-Aware Age-Agnostic Visual Place Recognition". In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 1437–1446. DOI: 10.1109/ICCVW.2019.00181.

[289] Frederik Warburg et al. “Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[290] Isaac Ronald Ward, M. Jalwana, and M. Bennamoun. “Improving Image-Based Localization with Deep Learning: The Impact of the Loss Function”. In: *PSIVT Workshops*. 2019.

[291] Weinzaepfel, Philippe and Lucas, Thomas and Larlus, Diane and Kalantidis, Yannis. “Learning Super-Features for Image Retrieval”. In: *ICLR*. 2022.

[292] Tobias Weyand, Ilya Kostrikov, and James Philbin. “PlaNet - Photo Geolocation with Convolutional Neural Networks”. In: *European Conference on Computer Vision*. 2016.

[293] Tobias Weyand et al. “Google Landmarks Dataset v2 – A Large-Scale Benchmark for Instance-Level Recognition and Retrieval”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2572–2581.

[294] Kyle Wilson and Noah Snavely. “Robust Global Translations with 1DSfM”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2014.

[295] Lin Wu et al. “3-D PersonVLAD: Learning Deep Global Representations for Video-Based Person Reidentification”. In: *IEEE Trans. Neural Netw. Learn. Syst.* 30.11 (2019), pp. 3347–3359. DOI: 10.1109/TNNLS.2019.2891244.

[296] Zhe Xin et al. “Visual place recognition with CNNs: From global to partial”. In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2017, pp. 1–6. DOI: 10.1109/IPTA.2017.8310121.

[297] Youjiang Xu et al. “Sequential Video VLAD: Training the Aggregation Locally and Temporally”. In: *IEEE Trans. Image Process.* 27.10 (2018), pp. 4933–4944. DOI: 10.1109/TIP.2018.2846664.

[298] Min Yang et al. “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*. 2021, pp. 11772–11781.

[299] Lin Yen-Chen et al. “inerf: Inverting neural radiance fields for pose estimation”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1323–1330.

[300] Kwang Moo Yi et al. “Lift: Learned invariant feature transform”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, pp. 467–483.

[301] B. Yildiz et al. "AmsterTime: A Visual Place Recognition Benchmark Dataset for Severe Domain Shift". In: *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society, 2022, pp. 2749–2755. DOI: 10.1109/ICPR56361.2022.9956049. URL: https://doi.ieeecomputersociety.org/10.1109/ICPR56361.2022.9956049.

[302] Shuhei Yokoo et al. "Two-stage Discriminative Re-ranking for Large-scale Landmark Retrieval". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2020), pp. 4363–4370.

[303] Jun Yu et al. "Spatial Pyramid-Enhanced NetVLAD With Weighted Triplet Loss for Place Recognition". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.2 (2020), pp. 661–674. DOI: 10.1109/TNNLS.2019.2908982.

[304] Dmitry Yudin et al. "Hpointloc: Point-based indoor place recognition using synthetic rgb-d images". In: *International Conference on Neural Information Processing*. Springer. 2022, pp. 471–484.

[305] Mubariz Zaffar et al. "CoHOG: A Light-Weight, Compute-Efficient, and Training-Free Visual Place Recognition Technique for Changing Environments". In: *IEEE Robotics and Automation Letters* 5 (2020), pp. 1835–1842.

[306] Mubariz Zaffar et al. "VPR-Bench: An Open-Source Visual Place Recognition Evaluation Framework with Quantifiable Viewpoint and Appearance Change". In: *International Journal of Computer Vision* 129.7 (2021), pp. 2136–2174.

[307] Amir R. Zamir et al., eds. *Large-Scale Visual Geo-localization*. Advances in Computer Vision and Pattern Recognition. Springer, 2016.

[308] Richard Zhang et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 586–595. DOI: 10.1109/CVPR.2018.00068.

[309] Xiwu Zhang, Lei Wang, and Yan Su. "Visual place recognition: A survey from deep learning perspective". In: *Pattern Recognition* 113 (2021).

[310] Xiwu Zhang et al. "Graph-Based Place Recognition in Image Sequences with CNN Features". In: *Journal of Intelligent & Robotic Systems* 95 (2018), pp. 389–403. URL: https://api.semanticscholar.org/CorpusID:116607093.

[311] Yongjun Zhang, Pengcheng Shi, and Jiayuan Li. "Lidar-based place recognition for autonomous driving: A survey". In: *ACM Computing Surveys* 57.4 (2024), pp. 1–36.

[312] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. "Reference pose generation for long-term visual localization via learned features and view synthesis". In: *International Journal of Computer Vision* 129 (2021), pp. 821–844.

[313] Junqiao Zhao et al. "Learning sequence descriptor based on spatio-temporal attention for visual place recognition". In: *IEEE Robotics and Automation Letters* 9.3 (2024), pp. 2351–2358.

[314] Xiaoming Zhao et al. "ALIKED: A Lighter Keypoint and Descriptor Extraction Network via Deformable Transformation". In: *IEEE Transactions on Instrumentation and Measurement* (2023).

[315] Bolei Zhou et al. "Places: A 10 million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[316] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. "Patch2pix: Epipolar-guided pixel-level correspondences". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 4669–4678.

[317] Sijie Zhu, Mubarak Shah, and Chen Chen. "Transgeo: Transformer is all you need for cross-view image geo-localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1162–1171.

[318] Sijie Zhu et al. "R2Former: Unified Retrieval and Reranking Transformer for Place Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023.

[319] Yingying Zhu et al. "Attention-based Pyramid Aggregation Network for Visual Place Recognition". In: *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*. ACM, 2018, pp. 99–107.

[320] Zihan Zhu et al. "Nice-slam: Neural implicit scalable encoding for slam". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12786–12796.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.