Md Didarul Alam Suvra

Application of Visual Place Recognition for Vehicle Positioning in a Rural Environment

School of Technology and Innovation
Master's Thesis in Computer Science
Sustainable and Autonomous Systems

Vaasa 2025

Contents

Figures

Images

Tables

Abbreviations

CNN	Convolutional Neural Network
VPR	Visual Place Recognition
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GLONASS	Global Navigation Satellite System (Russian system)
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
PCA-SIFT	Principal Component Analysis-SIFT
FAISS	Facebook AI Similarity Search
HOG	Histogram of Oriented Gradients
AR	Augmented Reality
GEM	Generalized Mean
NetVLAD	Neural Network Vector of Locally Aggregated Descriptors
TransVPR	Transformer-Based Visual Place Recognition
OTA-VPR	Optimal Transport Aggregation for Visual Place Recognition
TF-VPR	Temporal and Feature Neighborhoods for Visual Place Recognition
SeqNet	Sequence Network
SF-XL	San Francisco Extra Large
VG	Visual Geo-Localization
CosPlace	Cosine Similarity-Based Place Recognition
GPU	Graphics Processing Unit
ResNet	Residual Network
ResNet50	Residual Network with 50 Layers
ResNet18	Residual Network with 18 Layers
ResNet101	Residual Network with 101 Layers
MSLS	Mapillary Street-Level Sequences
Pitts30k	Pittsburgh 30k Dataset
Pitts250k	Pittsburgh 250k Dataset
APGeM	Average Precision Generalized Mean Pooling
MixVPR	Mixed Features for Visual Place Recognition
OTA-VPR	Optimal Transport Aggregation for Visual Place Recognition
TF-VPR	Temporal and Feature Neighborhoods for Visual Place Recognition
InLoc	Indoor Localization Dataset
GSV-Cities	Google Street View Cities Dataset
TB-Places	Tromsø Bridge Places Dataset
Habitat-Sim	Habitat Simulator

AnyLoc	Any Location
OpenCV	Open-Source Computer Vision Library
CUDA	Compute Unified Device Architecture
FFmpeg	Fast Forward Moving Picture Experts Group
RGB	Red, Green, Blue
VAL	Validation Set
GSV	Google Street View

UNIVERSITY OF VAASA
School of Technology and Innovations **Master's thesis in Computer Science**
Author: Md Didarul Alam Suvra
Title of the thesis: Visual Place Recognition
Degree: Master of Computer Science
Discipline: Sustainable and Autonomous Systems
Supervisor: Jani Boutellier, Heidi Kuusniemi
Year: 2025 **Pages: 58**

ABSTRACT:

This thesis evaluates visual place recognition (VPR) for autonomous navigation in a rural-road scenario representative of GNSS-denied conditions. A fixed image database was constructed from a car-mounted video recorded along the Nordmela–Stave route in Norway and organized according to the Pitts30k test structure within the Deep Visual Geo-Localization Benchmark framework. Ground truth was defined in WGS84 / UTM zone 33N (EPSG:32633). Two query sources were examined: in-domain car-camera frames from the same route and cross-domain Google Street View (GSV) images sampled along the same corridor. A pretrained ResNet-18 network with Generalized Mean (GeM) pooling (ResNet18-GeM) from the benchmark model zoo was used as the feature extractor, and standard retrieval metrics (Recall@k) were reported at multiple distance radii using the benchmark's evaluation tooling.

In the in-domain setting, performance reached 100% for Recall@1, Recall@5, Recall@10, and Recall@20 at a 100m radius. This outcome is consistent with near-duplicate views captured within short temporal offsets along the drive and confirms that data formatting and evaluation were correctly implemented. In the cross-domain setting, recall remained low at strict radii and increased only modestly as the success radius was relaxed. At 100 m the results were R@1/R@5/R@10/R@20 = 1/1/3/4%, and at 800 m they were 7/17/20/23%. Qualitative inspection showed that retrieved images were often visually plausible yet geographically incorrect. It reveals a substantial domain gap between GSV images and car-mounted imagery in terms of viewpoint, focal length, camera geometry, and appearance.

The contribution of this work is a benchmark-compatible evaluation of VPR in a rural-road, GNSS-denied use case, combining a real-world car-video database with GSV queries and reporting results under a transparent and reproducible protocol. The study indicates that, without adaptation, a Pitts30k-pretrained ResNet18-GeM generalizes poorly from car-camera imagery to GSV queries in this setting. A practical direction to improve cross-domain performance is focal-length and field-of-view correction of GSV queries so that their effective intrinsics match those of the car-camera database images. If resources permit, light fine-tuning using focal-length-corrected GSV crops could further adapt the descriptor. The results emphasize the importance of dataset compatibility, explicit coordinate handling, and geometric alignment when assessing VPR for autonomous navigation in GNSS-denied rural environments.

KEYWORDS: VPR, Deep Visual Geo-Localization Benchmark, CNN, Deep learning, Geo-localization, Pitts30k, ResNet18

1. Introduction

GPS, Galileo, GLONASS, and BeiDou are the backbone of modern navigation and positioning systems. These systems are called Global Navigation Satellite Systems (GNSS). GNSS signals are sometimes exposed to interference. One serious risk is GNSS jamming, which prevents receivers from accessing positioning signals (Humphreys et al., 2008). It was found that the weaknesses of the GNSS signal creates risks to the safety of critical applications. Losing location data can lead to operational failures. An example related to signal jamming occurred in early 2024 when an aircraft experienced navigation failure. It happened because of jamming signals in the Baltic region and Norway. The pilots managed the situation safely in the end (Reuters, 2024). These incidents show the need for a secondary navigation method.

Visual Place Recognition (VPR) is now seen one approach to tackle this challenge. It uses freshly taken images. This approach can function independently of external infrastructure. As a result, it is ideal for conditions where satellite signals are unavailable. This thesis studies the potential of visual localization in GNSS-denied conditions. Here, real-world video data is used. The video was recorded during a controlled GNSS jamming test, JammerTest 2024, Norway.

.

1.1. Background

Global Navigation Satellite Systems (GNSS) provide the primary source of absolute position for many navigation and positioning applications. In normal conditions, systems such as GPS, Galileo, GLONASS, and BeiDou offer sufficient accuracy for everyday use. However, GNSS signals are vulnerable to interference, which can significantly degrade positioning performance (Meng, Yang, Yang, & Zhang, 2022). This vulnerability has renewed interest in complementary localization methods that can support or verify GNSS-based solutions.

One important direction is to use on-board sensors that do not depend on external radio-frequency infrastructure. Cameras are particularly attractive because they are low-cost, already

widely used in vehicles, and capture rich visual information. Using camera images, it is possible to recognise where the vehicle is by comparing the current view to previously recorded, geo-referenced images. This concept is known as Visual Place Recognition (VPR)**.**

Visual Place Recognition is the task of determining whether a query image corresponds to a previously seen place and, if so, identifying that place by matching it against a database of reference images (Lowry et al., 2016).

Although deep learning has significantly improved VPR performance, achieving reliable operation across diverse conditions and environments remains an open challenge (Masone & Caputo, 2021). Garg, Fischer, and Milford (2021) further highlight that VPR requirements are strongly shaped by three factors: the operating environment, the agent (platform and sensors), and the downstream task. This means that there is no single universal setting; instead, methods and evaluation protocols must be designed with the specific application scenario in mind (Garg et al., 2021).

1.2. Motivation: GNSS-denied rural roads and the need for VPR

More than 50% of the body's sensory receptors are located in the eyes, and a significant portion of the cerebral cortex is devoted to processing visual information (Rehman, Hazhirkarzar, & Patel, 2025).Improving AI-based autonomous systems is closely related to advancing vision-based capabilities of the machine.

The signals jamming incident mentioned in the introductory part highlights the need for reliable fallback methods that can provide position information without relying on satellite signals. Therefore, methods based on readily available sensors are of growing interest.

In many situations, such as in cities with tall buildings, underground areas, or conflict zones, GNSS may not work well. In war zones, signals are sometimes intentionally blocked with a jammer. This creates a risk for systems that depend only on GNSS. In this situation, though solutions like

inertial sensors exist, they are often expensive. Also, inertial sensors suffer from accuracy issues. Conventional inertial navigation systems are prone to sensor noise, which makes regular calibration essential to maintain accuracy (Cohen & Klein, 2024). As an INS integrates accelerometer and gyroscope measurements, any small bias is also integrated, so the position error grows with time; if the INS runs long without external updates, the estimated position drifts from the true position (Capuano, Xu, & Estrada Benavides, 2023; Hou et al., 2025). Image-based positioning does not suffer from such drift.

Established VPR benchmarks, and experimental studies focus on urban environments, where stable, man-made features are abundant (Lowry et al., 2016; Masone & Caputo, 2021; Garg et al., 2021). In contrast, environments relevant to GNSS-challenged operation—such as rural areas, or natural landscapes—often provide fewer distinctive landmarks and exhibit strong appearance changes, making visual recognition difficult. At the same time, these environments can be affected by GNSS interference. This combination motivates the investigation of whether modern VPR techniques can be used as a practical, complementary localization method under GNSS-denied conditions, which is the focus of this thesis.

To explore this question, this thesis utilizes the University of Vaasa 2024 JammerTest dataset. Data were collected on a village road in Norway. Images from the unjammed drives are processed using an existing pre-trained VPR model.

1.3. Research objectives

This research has three main objectives:

1. Investigate Visual Place Recognition as an alternative for localization in GNSS-denied rural environments for autonomous driving.
2. Testing a pre-trained model on a novel dataset.
3. Evaluate the model's performance.

1.4. Scope of the thesis

This thesis focuses on Visual Place Recognition (VPR) using image-based deep learning methods. The study utilizes a rural road dataset to evaluate the performance of a pre-trained VPR model. The scope is limited to visual feature extraction and matching. Topics such as sensor fusion, simultaneous localization and mapping (SLAM) are not covered. The work does not address hardware implementation.

2. Theoretical foundations

Reliable localization in GNSS-denied rustic roads requires systems that can interpret visual information with high robustness to appearance changes, environmental conditions, and viewpoint variations. To understand the methods applied in this thesis, this chapter reviews the core concepts that enable image-based localization. It begins with explaining the concepts of computer vision, and topics such as image classification, object detection, and segmentation are discussed. The impact of deep learning on solving visual tasks is also presented in this section. And finally, Image retrieval techniques used in geo-localization are also included - how image descriptors are extracted and compared using tools like Faiss has been described.

2.1. Computer vision

Computer vision is an area within AI. It focuses on enabling machines to understand visual information from the world. Computer vision develops computational methods to interpret visual data, sometimes inspired by human vision but not restricted to it. The primary objective is to extract meaningful features from images and use them for specific tasks, such as image classification, localization, or retrieval (Szeliski, 2022).

After getting a raw input from the camera, the system defines the image in terms of meaningful features like edges, textures, and patterns. Earlier methods used handcrafted features such as SIFT (Lowe, 2004) and HOG (Dalal & Triggs, 2005). Early methods, such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded-Up Robust Features), were robust but not always accurate (Ke & Sukthankar, 2004). Modern approaches use deep learning, where Convolutional Neural Networks (CNNs) learn features directly from data. The extracted features are provided to a classifier, such as a softmax layer, to assign the image to a class. Image classification serves as the basis for more advanced tasks like object detection, segmentation, and visual place recognition.

Object detection goes further by identifying the object and locating it within the image. Semantic segmentation labels each pixel with its class.

Image matching is a difficult task. Changes in viewpoint, scale, lighting, and occlusion make it challenging. Visual place recognition must deal with all these variations. For example, an image taken in the summer may look very different from the same place in winter. Lowry et al. (2016) highlight that visual place recognition is challenging because the appearance of the same location can change significantly with viewpoint, illumination, weather, and season. To improve robustness, modern systems depend on deep features. These are learned automatically from large datasets.

A key concept in visual recognition is the descriptor. This is a numeric summary of an image or part of it. These summaries are compared to find similar images. If a query image has a similar numeric summary to a database image, it is likely that both images show the same location. This process is called image recovery. Principal Component Analysis-SIFT (PCA-SIFT) reduces feature dimensionality and improves robustness (Ke & Sukthankar, 2004). Classical descriptors often struggle with a large viewpoint change. That is why learning-based procedures are now more common. Tools like Faiss (Facebook AI Similarity Search) speed up the matching by using approximate nearest neighbour search (Johnson et al., 2017). These tools support large-scale operation.

Szeliski (2022) notes that the structure of images and scenes is essential. These help to understand how vision systems interpret environments. The spatial layout and texture provide context. Modern methods can learn this information directly from data. A 2013 study by Gronat et al. emphasised the importance of calibrating surface predictions at the pixel level, which improves 3D scene understanding—a useful attribute in place recognition by vision (Gronat et al., 2013).

Computer vision is growing rapidly due to advances in computational power. Sensors and cameras become cheaper and more accurate. The role of vision systems in daily life will increase gradually.

2.2. Deep learning and CNNs

Deep learning has transformed computer vision in recent years. The rise of deep learning introduced a new approach. Instead of manually designing features, deep neural networks can learn them from data. This change improved accuracy and generalisation in many tasks. Convolutional neural networks (CNNs) are hierarchical models and gained widespread attention after demonstrating breakthrough performance on the ImageNet Large Scale Visual Recognition Challenge in 2012 (Krizhevsky et al., 2017; Goodfellow et al., 2016).

These vision networks process images using layers. Each layer learns to detect patterns from raw pixels. The first layers capture simple shapes like edges. It learns more abstract and complex representations as the network goes deeper. The convolutional layers apply filters over the image. Pooling compresses the representation. The pooling layer keeps only the important information. Fully connected layers take all the learned representations and make a prediction. Layered structure makes convolution-based architectures powerful tools for feature extraction and image understanding (Szeliski, 2022).

In the context of image-based localization, representation learning is important. The system needs to recognise the same location under different conditions. CNNs learn visual representations that are more robust to variations. Once such a model encodes an image, the result is often a compact embedding. An effective embedding should be discriminative between different locations while robust to variations in viewpoint and appearance (Lowry et al., 2016; Zeng et al., 2018).

Several studies have used convolutional neural networks for place recognition with high success. Some systems compare deep convolutional embeddings using similarity metrics such as Euclidean or cosine distance (Schroff et al., 2015; Arandjelović et al., 2016). Other methods extend convolution-based models with attention mechanisms, enabling the network to capture richer spatial and temporal information, which makes these hybrid architectures suitable for dynamic scenes (Wang et al., 2022; Mereu et al., 2022; Wang et al., 2023).

The advantages of CNNs are clear. They perform well across different datasets. However, there are also some drawbacks. Deep convolutional models need large labelled datasets to train properly, and their training is slow, requiring strong hardware such as GPUs (Alzubaidi et al., 2021). Also, such networks often work like black boxes, and it is not always easy to understand why a CNN made a certain decision (Alzubaidi et al., 2021). Despite these challenges, deep convolutional models remain the most widely used architecture in computer vision. They have been improved in many ways, such as using residual connections, normalization layers, and different activation functions (He et al., 2016; Ioffe & Szegedy, 2015; Nair & Hinton, 2010).

2.3. Image retrieval for geo-localization

Image retrieval is an important method in visual geo-localization. It works by comparing a query image to a large database of geotagged images. The system finds the most similar image and uses its location as the estimated position. This process does not need GPS support. The basic idea is simple. A query image is taken, for example, by a camera. Then, the system searches a database for the closest match. If the matching image has a known location, that location is used as the prediction.

Convolution-based neural architecture systems extract compact embeddings from each image. These embeddings are stored in a database. When a query image is given, the system compares its embedding to the stored ones. The most similar one is the best match. One challenge is database coverage. If the database does not include images near the query location, the system may fail. Some solutions use Google Street View to build dense databases. Others combine ground and aerial images to improve accuracy, especially in remote areas (Wang et al., 2023).

There are also techniques to refine the results. One process is geometric verification. After finding a match, the system checks if the spatial layout of the scene is consistent between images. This reduces false matches. Other systems use re-ranking to boost performance. Some methods

combine image retrieval with other sensors; this can improve localization in complex scenes. Research explores the use of transformers, graph neural networks, and attention-based architectures to learn discriminative feature representations for visual place recognition (Zhu et al., 2018; Wang et al., 2022; Hu & Bors, 2023; Lezama et al., 2021).

Image retrieval for geo-localization is a fast, effective strategy. With deep learning, it has become more reliable and scalable. Despite challenges like appearance change and database coverage, ongoing research continues to improve accuracy and robustness.

3. Literature review

Visual place recognition (VPR) is the task of determining whether a place has been visited before using visual information, typically formulated to operate using only images rather than GPS or other external sensors (Zaffar et al., 2021). This method is related to image retrieval for geo-localization, but they differ in goals. Both utilise similar tools, but place recognition only needs to determine if a place is known. Geo-localization needs geographic coordinates. It matches images with geotagged references to output a location. The literature survey presents an analysis of recent contributions in the field. It emphasises methods that improve robustness to viewpoint changes, efficiency, and data scalability.

3.1. Feature mixing for robust representations: Mix VPR

Visual Place Recognition aims to match images of the same place taken at different times , under different conditions. Ali-Bey, Chaib-Draa, and Giguère (2023) proposed MixVPR. It is a method that enhances vision-based localization by combining pattern maps. This procedure mixes representations from different layers to capture both low-level and high-level information. Unlike traditional models that use only one level of representations, this strategy blends them using a learnable strategy. The authors showed that this mixing improves resilience to viewpoint and appearance changes. Their solution outperformed several baselines on common benchmarks. The network architecture remains lightweight and does not rely on external labels. Embedding diversity improves place recognition and also helps avoid overfitting by balancing representations.

3.2. Viewpoint robust training: EigenPlaces

Berton et al. (2023) proposed EigenPlaces, a training method that improves viewpoint robustness in visual place recognition.The authors identified that visual localization models struggle when scenes are viewed from different angles. They proposed a training approach based on eigenvalue analysis. This procedure helps the model learn more stable characteristics. It does not rely on traditional contrastive loss. EigenPlaces uses a directional loss function. This encourages the feature representations to align in specific directions. The paper also introduces a proxy-based formulation. This Proxy-based formulation simplifies training and maintaining performance. Their model achieves high accuracy on several benchmarks, particularly under large viewpoint changes, and does not require additional manual supervision beyond the standard geo-referenced training data.

The work is significant because it improves the resilience without making the network heavier. It also generalises well to new environments. This makes EigenPlaces useful for applications like robotics as well as autonomous driving. The method balances performance, efficiency, and practicality.

3.3 Unified retrieval and reranking: R2Former

Zhu et al. (2023) proposed R2Former, a transformer-based framework for visual localization. The model combines two important tasks: image retrieval and reranking. These tasks were handled separately in earlier systems. R2Former unifies them into a single pipeline. This integration improves both speed and accuracy. The procedure uses cross-attention to refine features from the query and database images. These refined features help the system to compare images more effectively. R2Former also applies a two-stage training strategy. It trains the retrieval module, then it fine-tunes the reranking module. This staged approach gives better performance than joint training.

The model was tested on several benchmarks where it outperformed many existing methods in both recall and precision. R2Former is important because it simplifies the image-based localization process. It reduces the gap between finding candidate matches. This leads to reliable place recognition.

3.4 Aggregation with optimal transport: OTA-VPR

Izquierdo and Civera (2024) proposed a new feature aggregation procedure. They used optimal transport theory to improve how patterns are combined. Unlike average or max pooling, their strategy preserves spatial relationships. This helps the model understand scenes from different viewpoints. Their technique applies Sinkhorn iterations, and it is mathematically grounded. The method builds better global descriptors from local representations. These signatures support more accurate place matching. The authors tested their model on multiple datasets, and the results showed improved recall and precision compared to prior frameworks. It outperforms NetVLAD and GeM on several benchmarks, and its DINOv2 backbone improves accuracy but runs slower than typical ResNet-based baselines.

3.5. Data-efficient supervision: graded similarity learning

Leyva-Vallina et al. (2023) addressed the challenge of training image-based location recognition models with less labelled data. They proposed a graded similarity supervision approach which moves beyond binary labels. Instead of just matching or non-matching, it assigns similarity scores, and the scores represent how close two places are. This helps the model learn subtle visual differences. Their method is more data-efficient than traditional setups. The authors used a new loss function to support this learning. It takes graded labels into account. They tested the model on large-scale benchmarks, and the results showed better performance than standard contrastive learning. Graded Similarity Learning was especially helpful with limited annotations. The model learned resilient visual patterns and handled viewpoint and appearance changes well. The

framework also reduced overfitting, and their research supports more practical visual geolocalization systems. It shows how supervision can be improved for scale.

3.6. Traffic sign recognition as a proxy task

Zhu and Yan (2022) explored deep learning procedure for traffic sign recognition. While the paper focuses on signs, its techniques are useful for location recognition. The authors used convolutional neural networks (CNNs) to extract representations. They trained their model on a large dataset, which included traffic signs under different conditions. This helped the model become consistent, as places may appear different due to weather, time, or viewpoint. The paper also used data augmentation. This improved model generalisation. In place matching, augmented training data can help the system adapt better in real-world scenarios.

Their results showed high accuracy, which confirms that deep learning works well for visual tasks. Though recognising traffic signs is different from place recognition, the methods used in the paper give useful insights for place identification research.

3.7 Self-supervised learning via temporal and feature neighborhoods: TF-VPR

Chen et al. (2025) introduced TF-VPR. It is a self-supervised framework for visual place recognition. The method uses temporal neighborhoods and learnable feature neighborhoods to discover spatial neighbors without relying on ground-truth poses. It learns from sequential panoramic RGB images and 3D point clouds, which reduces the need for costly manual annotation. Temporal neighbors, i.e., observations that are close in time, are treated as likely positives for the same place, while feature similarity in the learned embedding space is used to expand the set of candidate positives. An iterative process of labeling, training, expansion, and contraction refines these pseudo labels using geometric verification. Experiments on simulated indoor RGB data and real LiDAR scans from KITTI-360 show that TF-VPR outperforms other self-

supervised baselines and is competitive with supervised methods when the domain gap between training and testing data is small. This demonstrates that exploiting temporal and featural structure in sensory data can substantially reduce the need for pose-dependent supervision in visual place recognition.

3.8. Deep learning in autonomous driving: general insights

Fujiyoshi, Hirakawa, and Yamas (2019) reviewed deep learning-based localization for autonomous vehicles and reported that convolutional neural networks are effective at extracting robust features from complex real-world environments. They emphasized that real-time deployment is challenging due to two factors. First, computational constraints, and then rapidly changing traffic conditions. They argued that reliable localization requires more efficient architectures, like large-scale training data and the fusion of visual information with complementary onboard sensors.

3.9 Forest road recognition using deep learning

Lee, Lee, and Kim (2024) present a deep learning method for recognising forest roads in images. Their goal is to improve navigation in natural settings by identifying roads in a reliable way. They use convolutional neural networks to train the model. They try to separate the road surface from trees, bushes, and shadows. The authors point out why this task is hard: lighting shifts under the canopy, shadows move through the day, and seasonal change alters texture and colour; branches, leaves, and ground cover often block the view. To cope with this, they train on a large, varied dataset of forest scenes. The system performs strongly, holding up under occlusion and visual noise, and it lifts the accuracy of place recognition in outdoor, less structured environments. The work has clear uses in autonomous forestry vehicles and in environmental monitoring, where dependable road detection matters. It also underlines the value of specialised datasets tailored

to natural areas rather than urban streets. Taken together, the results show how deep learning can extend location recognition beyond cities and into challenging forest terrain.

3.10. NetVLAD and the Pitts30k dataset

Arandjelović et al. (2016) introduced the Pitts30k dataset and also proposed NetVLAD, a neural network designed for large-scale visual place recognition. The Pitts30k dataset is a subset of the larger Pitts250k dataset. Pitts30k contains about 30,000 images, while Pitts250k includes around 250,000 images. The authors proposed a novel trainable generalised VLAD (Vector of Locally Aggregated Descriptors) layer. Which could be integrated into a CNN and trained end-to-end using weak supervision. This procedure allowed the model to learn reliable image representations that could handle significant changes in viewpoint, lighting, and occlusion. To evaluate their procedure, the authors collected and curated the Pitts30k dataset. This dataset consists of street-level imagery from Pittsburgh, USA, captured using Google Street View. The dataset includes well-organised query and reference images. NetVLAD demonstrated superior performance over earlier CNN-based methods. And, both the architecture and dataset have since become widely adopted in the visual place recognition community. In this thesis, the Pitts30k dataset is used.

3.11. SeqNet: learning descriptors for sequence-based hierarchical place recognition

Garg and Milford (2021) proposed SeqNet, a deep neural network for sequence-based hierarchical place recognition. It is different from traditional location recognition systems. SeqNet creates descriptors from image sequences. This makes it more powerful in changing conditions. The model uses a two-stage approach. First, it processes individual frames to extract image-level representations. Then it aggregates information from a short sequence of images to form a stronger, context-aware descriptor. This structure allows the system to recognise places more

accurately in environments where visual appearance changes. The authors tested SeqNet on multiple public datasets. The results show that incorporating temporal information helps distinguish similar places and improves recognition consistency. The method is particularly useful in long-term localization tasks.

3.12. SeqNetVLAD vs PointNetVLAD

Garg and Milford (2021) conducted a comparative study to evaluate the effectiveness of an image sequence-based place recognition process. They compare with 3D point cloud-based approaches. This method focuses on day-night place recognition scenarios. They compared SeqNetVLAD, an image sequence-based solution, with PointNetVLAD, a 3D point cloud-based process. They tested to evaluate their performance under varying environmental conditions. The study experimented with whether 3D structure-based place representations perform better than spatial representations learned from RGB image sequences. The authors found that image sequence-based procedures, such as SeqNetVLAD, can match or even surpass point cloud-based solutions for certain metric spans. The difference in performance stems from the amount of data the sensors provide and how mobile robots collect and combine it.

3.13. Rethinking visual geo-localization for large-scale applications

Berton, Masone, and Caputo (2022) offer a detailed look at visual geo-localization in city-scale settings. Their main contribution is the San Francisco eXtra Large (SF-XL) dataset—a city-wide benchmark that is about thirty times larger than earlier sets. It packs in a wide mix of tough, real-world cases and lets researchers test methods at a true urban scale. The authors point to two sticking points in today's systems: scalability and efficiency. Classic pipelines buckle as data grows, largely. Because contrastive learning is compute-heavy and depends on costly hard-negative mining. To tackle this, they introduce CosPlace, a training scheme that casts VG as a classification

task. By doing so, it removes the need for expensive mining and cuts both training time and memory use.

In practice, CosPlace delivers state-of-the-art results across several benchmarks. During training, it uses roughly 80% less GPU memory, and it produces descriptors that are about eight times smaller while keeping strong accuracy. That mix of speed, compactness, and performance makes CosPlace well-suited to real, city-wide deployments.

Table 1. CosPlace datasets detail analysis

Datasets	**Type of Image**	**Camera Type**	**Depth Info**	**Number of Images**	**Area type**	**Notes**	**Image size**	**Dataset size**
MSLS	Composed of frontal view images taken with cars	Monocular(RGB)	no	1.7M	Urban + Suburban	1. The test set labels have not been publicly released yet. It was tested on the validation set 2. Do not provide heading labels for their images, making CosPlace not trainable	480×640	56 GB
St Lucia	Composed of frontal view images taken with cars	Monocular(RGB)	no	33K	Suburban	From car-mounted camera in Australia, driving around same area repeatedly	480×640	124 MB

Pitts30k	Google StreetView imagery	Monocular(RGB)	no	30K	Urban	Do not provide heading labels, making CosPlace not trainable	480×640	2 GB
SFXL test v1	Google StreetView imagery	Monocular(RGB)	no	2.8M (1000 query,flickr)	Urban	Urban images	480×640	
Pitts250k	Google StreetView imagery (Larger version of Pitts30k)	Monocular(RGB)	no	278K	Urban	Urban images	480×640	
Tokyo 24/7	Google Street View	Monocular(RGB)	no	189K	Urban	Urban images(Urban Japan, various lighting conditions (day/night)	480×640	4 GB
SFXL test v2	Google StreetView imagery	Monocular(RGB)	no	2.8 M (from San Francisco Landmark Dataset ,598 query images)	Urban	Urban images	480×640	

3.14. Deep visual geo-localization benchmark

Berton et al. (2022) introduced an open-source benchmarking framework for visual place recognition. The goal was to standardise how these location recognition systems are evaluated. The framework allows researchers to build, train, and test models. It also shows how design choices affect performance measures like recall@N, speed, and memory use. The authors

emphasise the importance of considering system requirements alongside accuracy. These factors are crucial for real-world applications.

The study conducted a lot of experiments and highlighted key points for pipeline design. For example, reducing image resolution to 80% gave similar accuracy while reducing processing time and storage. The framework also tested components such as backbone networks, aggregation methods, and negative mining strategies. These results give guidance on how to optimise systems for specific tasks and hardware limits. In this thesis, this framework, called the Deep Visual Geo-Localization Benchmark, is used to conduct the experiments.

Table 2. Analysis on deep visual geo localization benchmark

Aspect	**Details**
How the Benchmark Works	Provides a full pipeline for training**,** validating, and evaluating visual geo-localization models. It compares performance based on how well query images are matched to correct locations in a reference dataset.
Main Components Supported	train.py: Train models on supported datasets test.py: Evaluate retrieval accuracy infer.py: Run inference on new data datasets: Dataset preparation utilities
Training Options	Train from scratch or fine-tune using: - CosPlace (classification-based) - NetVLAD/APGeM (retrieval-based)
Datasets	Pitts30k, Pitts250k, MSLS, Tokyo 24/7, San Francisco (R-SF), Eynsham, St Lucia, SVOX, Nordland, AmsterTime, SPED
Model Zoo	Provides pretrained models trained by the authors on standard datasets like Pitts30k, MSLS. Models include: NetVLAD, CosPlace, APGeM aggregation methods with ResNet50, ResNet 18, Resnet 101 backbones.

3.15 Chapter summary

Recent work aims to make image-based localization more reliable, efficient, and scalable. MixVPR widens feature diversity, while EigenPlaces improves resilience to viewpoint changes. Transformer models such as R2Former bring retrieval and re-ranking into a single pipeline and

deliver higher accuracy. OTA-VPR uses optimal transport to aggregate features more effectively. On the label-efficiency front, self-supervised methods like TF-VPR exploit temporal links and feature neighbourhoods to cut down on annotations.

Benchmarks like NetVLAD remain important, and sequence-based models like SeqNet show the value of temporal context. The Deep Visual Geo-Localization Benchmark has further standardised evaluation and accelerated research progress. These advances highlight steady progress toward more adaptable VPR systems.

Table 3. Literature survey summary table

Paper	**Year**	**Code (GitHub link)**	**Language / Framework**	**Pretrained model available on GitHub(y/n)**	**Supported dataset**	**Library / Software**
MixVPR: Feature Mixing for Visual Place Recognition	2023	https://github.com/amaralibey/MixVPR	Python/ PyTorch	Yes (ResNet50)	GSV-Cities dataset	PyTorch,numpy, matplotlib,pandas, faiss-gpu, torchvision
Visual Place Recognition: A Tutorial [Tutorial]	2023	https://github.com/stschubert/VPR_Tutorial	Python	no	specific datasets are not detailed	Ipywidgets, Matplotlib, Numpy,Pillow, scikit_image, tensorflow, tensorflow_hub,torch,torchvision,tqdm
EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition	2023	https://github.com/gmberton/EigenPlaces	Python/ PyTorch	Yes (ResNet101)	Pittsburgh30k, Tokyo24/7, Nordland, Mapillary Street Level Sequences	Python, PyTorch, Torchvision, NumPy, OpenCV
AnyLoc: Towards Universal	2024	https://github.com/AnyLoc/AnyLoc	Python/ PyTorch for deep learning	Yes	Baidu Mall, Gardens Point, Oxford Robotcar,	OpenCV, PyTorch, NumPy, Torchvision

Visual Place Recognition			functionalities/ Jupyter Notebooks		Pitts-30k, Laurel Caverns, Mid-Atlantic Ridge	
R2Former: Unified Retrieval and Reranking Transformer for Place Recognition	2023	https://github.com/bytedance/R2Former	Python/ PyTorch for deep learning	Yes	MSLS (Microsoft Landmark Dataset), Pitts30k, Tokyo24/7, Nordland, Mapillary Street Level Sequences, Aachen Day-Night, InLoc	Python, PyTorch, Torchvision, NumPy, OpenCV, Matplotlib, Scikit-learn, Gradio, Jupyter
Optimal Transport Aggregation for Visual Place Recognition	2024	https://github.com/serizba/salad	Python/ PyTorch for deep learning	Yes (DINOv2 SALAD)	Pitts30k, Tokyo24/7, Mapillary Street Level Sequence, Aachen, Day-Night InLoc	Torchvision, NumPy, OpenCV, Matplotlib, Scikit-learn
Data-efficient Large Scale Place Recognition with Graded Similarity Supervision	2023	https://github.com/marialeyvallina/generalized_contrastive_loss	Python/ PyTorch for deep learning	No	Mapillary Street Level Sequences (MSLS), TB-Places, 7Scenes	Torchvision, NumPy, OpenCV, Matplotlib, Scikit-learn
Traffic sign recognition based on deep learning	2022	https://github.com/ultralytics/yolov5	Python/ PyTorch	Yes	own dataset for the experiments, comprising 2,182 images	NumPy, OpenCV, Matplotlib
Self-Supervised Place Recognition by Refining Temporal and Featural Pseudo Labels from	2025	https://github.com/ai4ce/TF-VPR	Python/ PyTorch	No	2D simulated point cloud datasets., Habitat-Sim RGB dataset, NYU-VPR-360 dataset	NumPy, OpenCV, Matplotlib

Panoramic Data						
Deep learning-based image recognition for autonomous driving	2019	no	Python, C++/ TensorFlow or PyTorch	no	KITTI Cityscapes COCO ImageNet	NumPy, OpenCV, Matplotlib, CUDA (for GPU acceleration)
Developing Forest Road Recognition Technology Using Deep Learning-Based Image Processing	2024	no	Python/ PyTorch or TensorFlow	no	images taken 1.2 km of forest roads, 633 images	NumPy, OpenCV, Matplotlib, CUDA (for GPU acceleration)
NetVLAD: CNN Architecture for Weakly Supervised Place Recognition	2016	https://github.com/Relja/netvlad?	MATLAB Alternative Implementations: Python	Yes	Pittsburgh 250k, Pittsburgh 30k, Tokyo 24/7	MatConvNet, TensorFlow, PyTorch
SeqNet: Learning Descriptors for Sequence-based Hierarchical Place Recognition	2021	https://github.com/oravus/seqNet	Python	no	Nordland, Oxford ,RobotCar, Pitts250k Tokyo 24/7	PyTorch, Torchvision, NumPy, Faiss, scikit-learn, TQDM, TensorBoardX
SeqNetVLAD vs PointNetVLAD: Image Sequence vs 3D Point Clouds for Day-Night Place Recognition	2021	https://github.com/oravus/seqNet	Python	no	Oxford RobotCar	PyTorch, Torchvision, NumPy, Faiss, scikit-learn, TQDM, TensorBoardX, h5py

Rethinking Visual Geo-localization for Large-Scale Applications	2022	https://github.com/gmberton/CosPlace	Python	Yes	San Francisco eXtra Large (SF-XL)	PyTorch , Torchvision, Faiss (CPU), NumPy, Pillow, scikit-learn, TQDM, UTM.
Deep Visual Geo-localization Benchmark	2022	https://github.com/gmberton/deep-visual-geo-localization-benchmark	Python	Yes	Pitts30k, Pitts250k, MSLS, Tokyo 24/7, San Francisco (R-SF), Eynsham, St Lucia, SVOX, Nordland, AmsterTime, SPED	PyTorch, Torchvision, Faiss (CPU/GPU), NumPy, scikit-learn, Pillow, TQDM, psutil, timm, transformers, einops, requests, googledrivedownloader, torchscan

4. Methodology

The experimental pipeline consists of four main stages: (i) dataset acquisition and formatting, (ii) model selection, (iii) descriptor extraction and retrieval, and (iv) evaluation. The Pitts30k dataset was downloaded and formatted using the VPR-datasets-downloader and the format_pitts30k.py script to match the directory structure required by the Deep Visual Geo-localization Benchmark framework. A pretrained ResNet18-GeM model from the official model zoo was used as the feature extractor, and the provided eval.py script was employed to compute standard retrieval metrics (e.g., Recall@k). All steps were carried out to ensure compatibility with the Deep Visual Geo-Localization Benchmark, which serves as the main framework for this research. The Pitts30k dataset structure provided by the benchmark was used as a reference. The original .mat files of Pitts30k were modified to incorporate with the customized University of Vaasa 2024 JammerTest image dataset. The overall structure was kept the same to ensure consistency with the benchmark.

4.1 Benchmark framework and dataset integration

Two videos were recorded during a jammer test event in Norway. One was recorded with a intentional GPS jamming situation. The other was recorded under normal conditions. Frames extracted from the video recorded under non-jammed GNSS conditions were used to construct the test database and query sets. FFmpeg was used for video-to-image conversion.

The CosPlace datasets were reviewed to understand data types and features. Based on this review, the Pitts30k dataset structure was selected as the template. Pitts30k is a subset of Pitts250k and is natively supported by the Deep Visual Geo-Localization Benchmark. Pitts30k was adopted as the reference dataset structure for integrating the University of Vaasa 2024 JammerTest images. Query and database images from both Pitts30k and the University of Vaasa 2024 JammerTest dataset were organized according to the same directory and naming

conventions. This ensured that the custom dataset could be evaluated directly within the benchmark framework.

4.2 Model architecture

This study employs the ResNet18-GeM configuration provided by the Deep Visual Geo-Localization Benchmark model zoo. The network uses a ResNet-18 backbone truncated at the conv4 stage as the feature extractor. A Generalized Mean (GeM) pooling layer is used. This aggregates the convolutional feature maps into a single global descriptor. The descriptor is L2-normalized and used for image retrieval. The model weights are taken from the pretrained ResNet18-GeM checkpoint released with the benchmark; no new architecture is designed in this work.

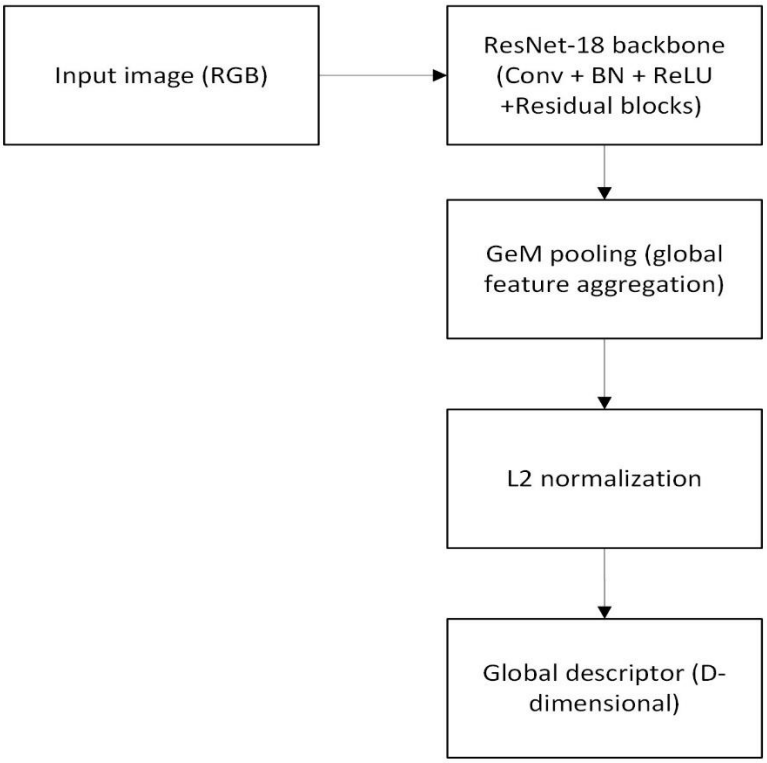

Figure 1: Architecture of the pretrained ResNet18-GeM feature extractor.

4.3 Retrieval pipeline and evaluation protocol

Within the Deep Visual Geo-Localization Benchmark, visual place recognition is implemented as a global-descriptor retrieval pipeline. All database images are first preprocessed (resizing and normalization) and passed through the pretrained ResNet18-GeM network to obtain L2-normalized global descriptors, which are stored in a FAISS index (descriptor database). Each query image is processed using the same preprocessing steps and the same ResNet18-GeM model to compute its corresponding L2-normalized query descriptor. The query descriptor is compared with the database descriptors using Euclidean distance. The top-k nearest neighbors is returned as candidate matches. A retrieval is correct if at least one of the top-k candidates lies within a predefined ground-truth distance threshold. Retrieval performance is reported using Recall@ k, following the standard evaluation protocol of the Deep Visual Geo-Localization Benchmark.

This retrieval and evaluation pipeline is applied consistently to both the original Pitts30k configuration and the integrated University of Vaasa 2024 Jammer Test, test dataset. This ensures that all results are comparable within the same framework.

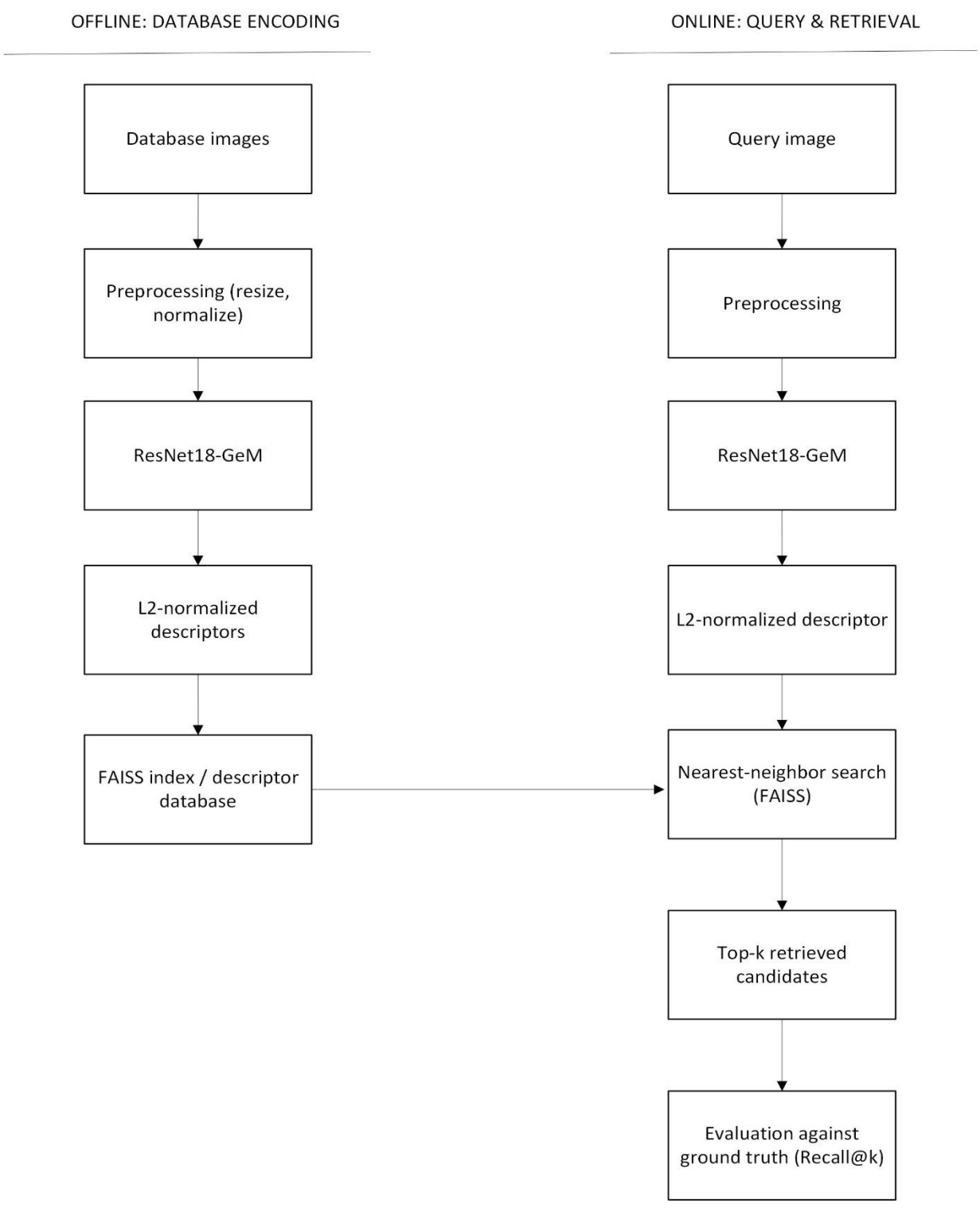

Figure 2. Retrieval pipeline within the deep visual geo-localization benchmark

5. Experiments

This section describes how the custom dataset was prepared, integrated into the Deep Visual Geo-Localization Benchmark framework, and evaluated. The objective of this study is to evaluate visual place recognition in a GPS-denied rural road environment for potential use in autonomous driving, using the Deep Visual Geo-Localization Benchmark framework.

5.1. Data sources and data preprocessing

A 12-minute “GNSS no-jam” dash-cam video from a 12–13 km stretch in Norway (Nordmela to Stave) is used. The route used for the database images is shown in Image 1.

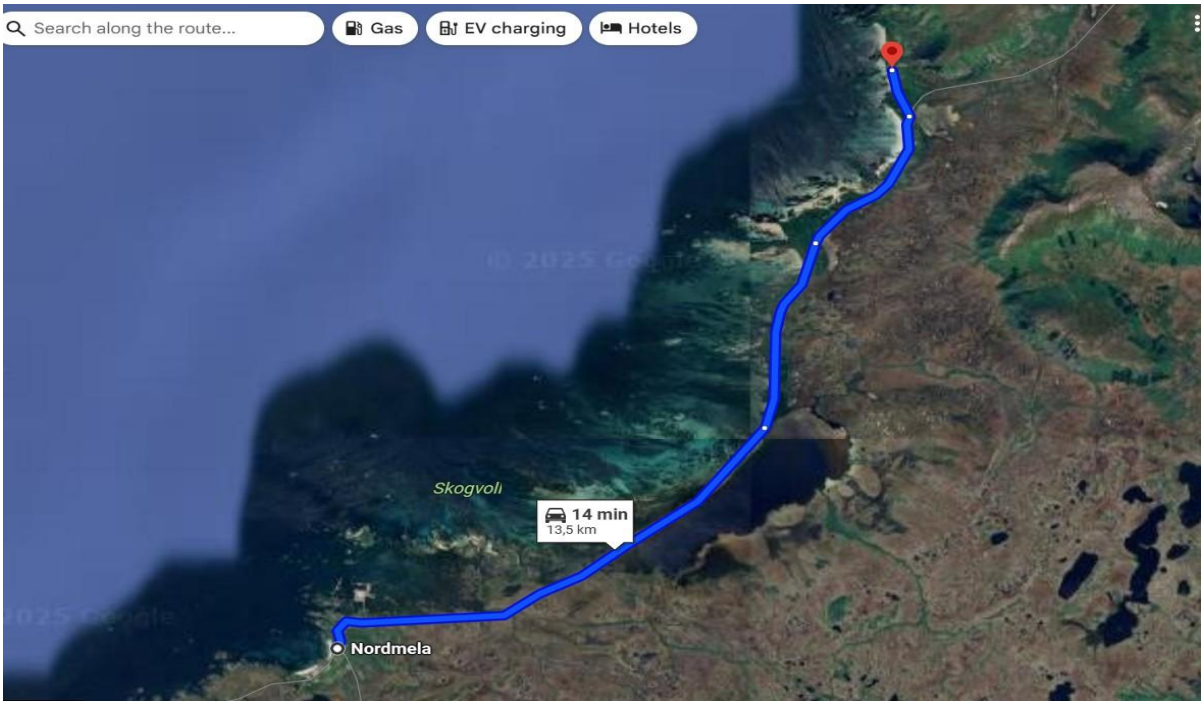

Image 1. Route of the jammertest 2024 car

To reduce redundancy, frames are extracted while every second frame is skipped. This keeps the original order but halves the data volume. The database and query images are partitioned according to a fixed index rule: frames with indices 0, 1, 2, 4, 5, 6, 8, 9, 10, … are assigned to the database, while frames 3, 7, 11, … are assigned as queries. All images are resized to 640×480 (width × height) and renamed (e.g., 000000_nojammer_db.jpg for database and 000000_nojammer_q.jpg for queries). Each frame is associated with GPS latitude and longitude in WGS84. These coordinates are transformed to the WGS84 / UTM zone 33N coordinate reference system (EPSG:32633), yielding Easting and Northing in metres within a single UTM zone covering the route.

The dataset is organised to match the Pitts30k test layout: database images under images/test/ and queries under queries_real/009/. A pitts30k_test.mat file stores the dbStruct, including relative filenames (dbImageFns, qImageFns), UTM coordinates (utmDb, utmQ), and counts (numImages, numQueries). A verification script checks that the listed filenames exist on disk and that UTM coordinates line up with the source GPS. Figure 3 summarises the preprocessing and Pitts30k formatting, and Figure 4 shows the evaluation pipeline.

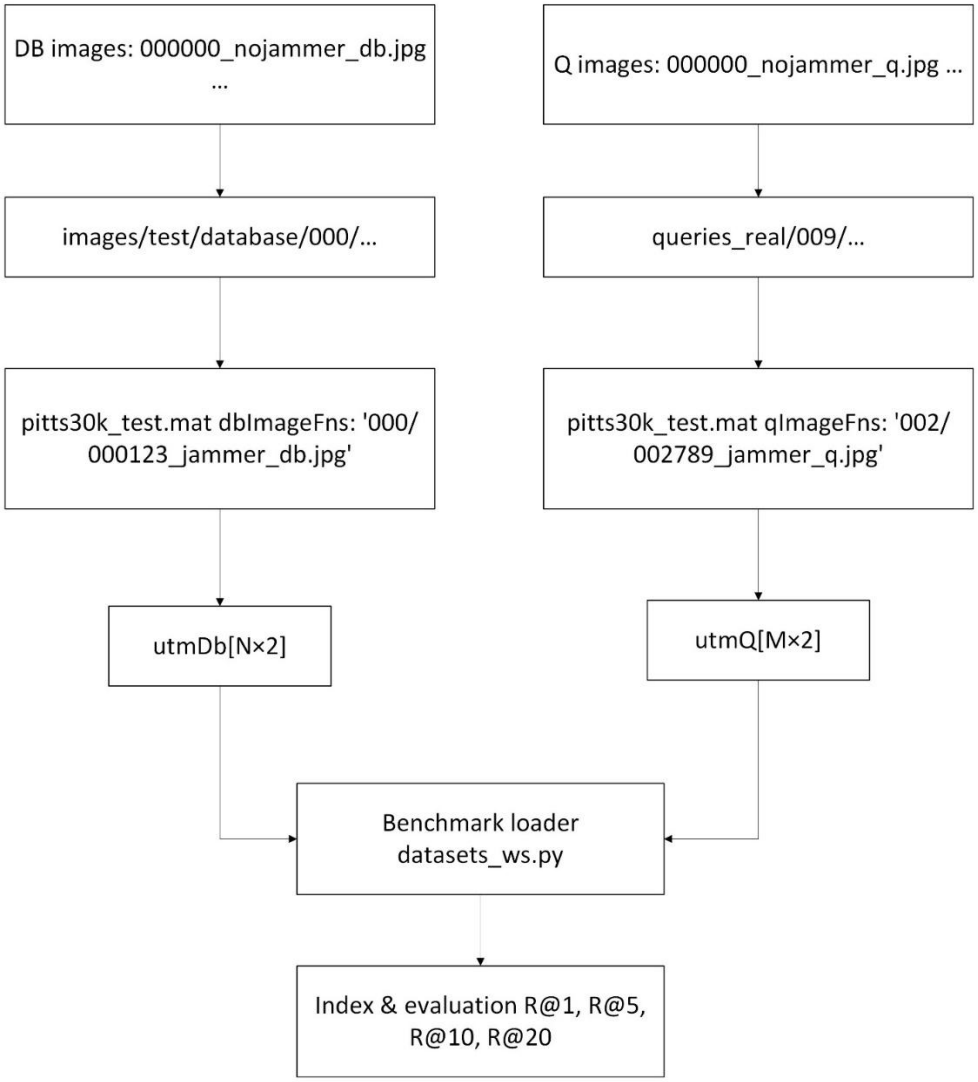

Figure 3. Test structure and the role of pitts30k_test.mat fields

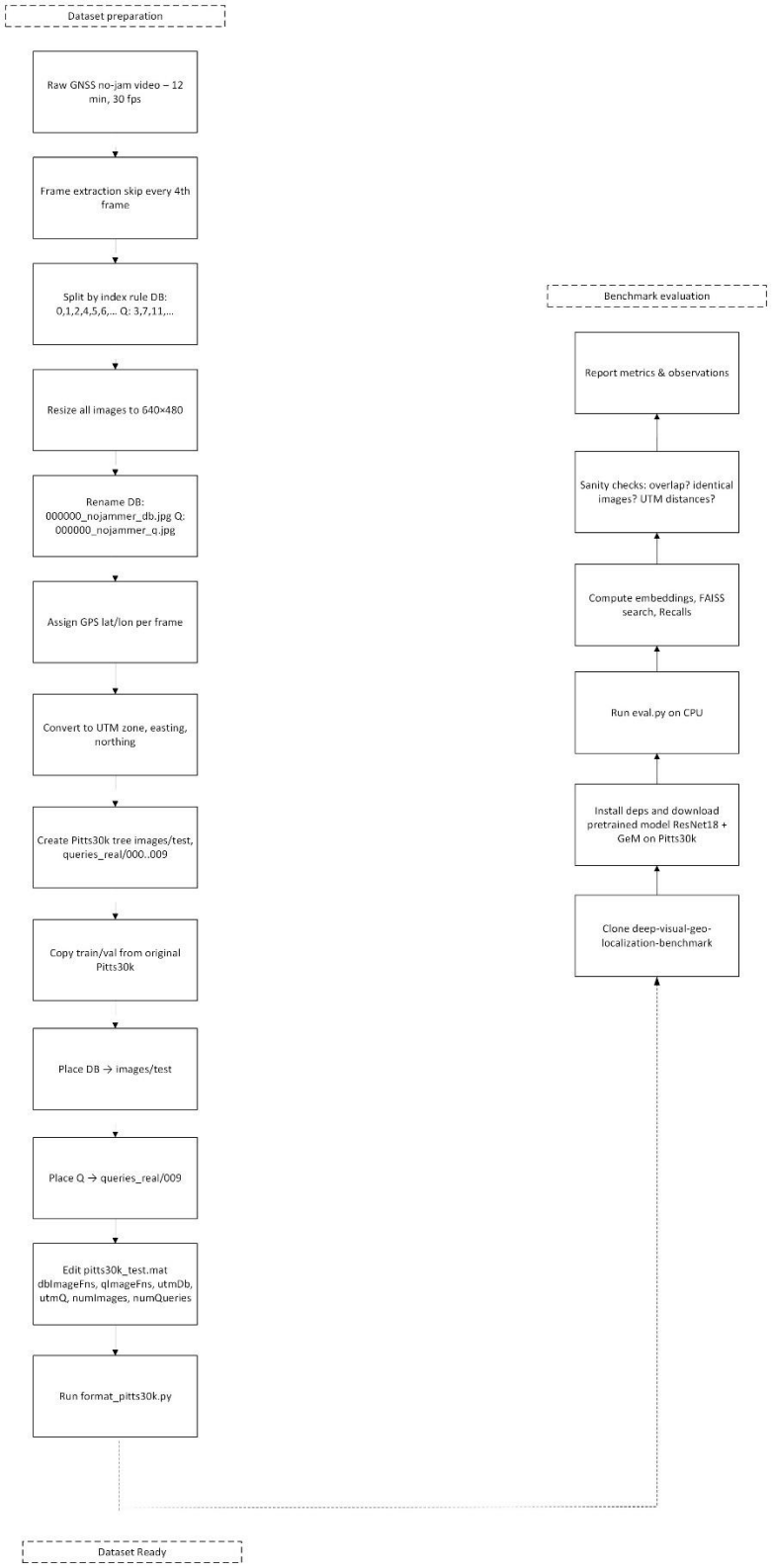

Figure 4. End-to-end evaluation pipeline

5.2. Model and benchmark setup

A pretrained ResNet-18, truncated at conv4 and equipped with a GeM pooling head, is used. The model is taken from the release provided with the Deep Visual Geo-localization Benchmark and is pretrained on the Pitts30k dataset.Inference resizes inputs to 480×640 with the hard_resize test mode, runs with small batches on CPU, and reports Pitts30k-style Recall@kkk with a 25 m success radius, where a query is counted as correct if any of its top-kkk retrieved database images lie within 25 m in WGS84 / UTM zone 33N (EPSG:32633).

5.3. Sanity checks and controls

Overlap between the database and query sets is checked by matching filenames, and no identical entries are found. Planar distances in WGS84 / UTM zone 33N (EPSG:32633) from each query to its closest database frame are then computed. Distances were effectively zero (median 0.00066 m, maximum 0.22 m), indicating the pairs are co-located, which is expected given the small temporal gap. From this, the perfect scores are most likely due to near-duplicate pairing (same place, tiny time offset). These checks show the setup is internally consistent, but they do not guarantee performance in new conditions. Figure 4 outlines the flow for embedding extraction, FAISS search, recall calculation, and these checks.

5.4. Queries from google street view (GSV)

To test robustness, we built an extra query set from Google Street View along the same Nordmela to Stave route. We sampled 20 positions covering the road and captured five headings at each, giving 100 screenshots. For each image, we logged a CSV row (idx, location_id, lat, lon,

heading_deg) using latitude/longitude from the Street View URL; UTM was derived for evaluation. The spatial distribution of the Google Street View query locations along the route is shown in Figure 2.

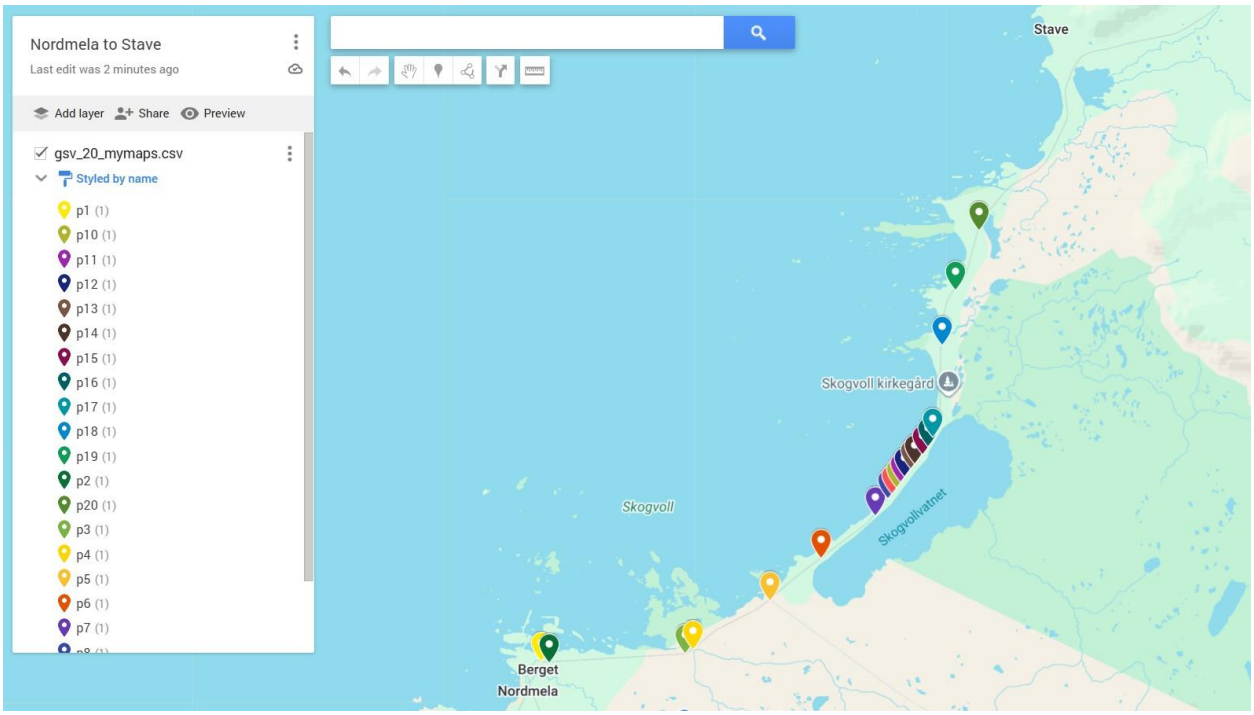

Image 2. Google street view query locations (p1–p20) .

Images were resized to 640×480, renamed consistently (e.g., 000000_gsv_q.jpg), placed under queries_real/009/, and the .mat was updated (qImageFns, utmQ, numQueries). Figure 3 outlines selection, capture, logging, and integration.

5.5. Results summary

Under the standard setup (25 m radius; ResNet-18 conv4 + GeM pretrained on Pitts30k; 480×640; hard_resize; CPU) and using car-video frames as the database, the in-domain car-video queries achieved Recall@1/5/10/20 = 100/100/100/100. Replacing the queries with GSV screenshots

against the same database reduced Recall@K, as expected. It happened due to viewpoint shifts, different cameras, lighting, and seasonal changes. The 100% in-domain result aligns with the centimetre-scale nearest-neighbour UTM distances. The table below reports the scores for the GSV queries.

Table 4. Results for the first GSV-query experiment (five images per location).

Radius (m)	R@1 (%)	R@5 (%)	R@10 (%)	R@20 (%)
100	1	1	3	4
200	1	1	6	9
500	4	10	13	17
800	7	17	20	23

Table 4 reports the scores for the first GSV experiment, where Google Street View (GSV) images were used as queries against the same car-camera database. In this experiment, at each of 20 locations along the Nordmela–Stave route, five GSV images were captured in different viewing directions, resulting in 100 query images in total. For each GSV query, a retrieval is counted as correct if at least one of the top-kkk retrieved database images lies within the specified success radius (in metres) from the query position in WGS84 / UTM zone 33N. At a 100 m radius, only 1% of queries have a correct match at rank 1, and only 4% have any correct match within the top-20 results, indicating that almost all GSV queries fail under a strict localisation criterion. Even when the radius is relaxed to 800 m, Recall@20 increases only to 23%, meaning that more than three-quarters of the GSV queries still do not retrieve a geographically correct database image within their top-20 candidates.

Table 5. Results for the second GSV-query experiment (one image per location).

Radius (m)	R@1 (%)	R@5 (%)	R@10 (%)	R@20 (%)
100	0	0	0	0
200	0	0	0	0
500	2	2	2	2
800	8	8	8	8

Table 5 shows results for a second GSV experiment, where a new query set was collected. This time, 50 GSV images were captured at 50 distinct locations, with exactly one image per location (unlike the previous experiment, where five images were taken per point). Using the same car-camera database, Recall@1/5/10/20 is 0/0/0/0 for success radii of 100 m and 200 m. At a 500 m radius, Recall@1/5/10/20 reaches only 2%, and at 800 m it reaches 8% for all values of k. In this experiment, manually cropping the GSV query images to visually match the car-camera views was attempted. However, this did not significantly improve Recall@K. This outcome indicates that visual cropping is insufficient to overcome the underlying domain gap. In addition, geometric analysis showed that only 4 out of 50 GSV locations have any database frame within 800 m, so many queries do not have a valid positive in the database under the chosen evaluation radius. Combined, the limited spatial overlap and the strong domain shift explain why the model still performs poorly on GSV queries, even when the images appear visually similar to a human observer.

5.6. Results analysis

(a) GSV query image (Google, 2022)

(b) Rank-1 retrieved database image

(c) Rank-2 retrieved database image

(d) Rank-3 retrieved database image

(e) Rank-4 retrieved database image

(f) Rank-5 retrieved database image

Image 3. Example of a failure case for the first GSV-query experiment.

GSV query image (a) and (b–f) Rank-1 to Rank-5 retrieved database images produced by the ResNet18-GeM model. Images (b), (c), and (d) are visually similar to the query and appear to lie on a nearby road segment, whereas (e) and (f) are clearly from different locations and differ in scene layout. However, under the strict evaluation criterion (WGS84 / UTM zone 33N, 100 m radius, top-20 results), none of these candidates is counted as a correct match. So, this query is classified as a failure at R@20. This illustrates how the model can return plausible-looking but geographically incorrect places under cross-domain conditions.

DB_q46_r1, UTM:33N 533996 7678 250

Q_q46, UTM:33N 534172.0 7677474.6

DB_q47_r1, UTM 33N 533996 7678250

Q_q47, UTM 33N 534165.6 7677515

DB_q48_r1, UTM 33N 533996 7678250

Q_q48 ,UTM 33N 534140.1 7677696.9

DB_q49_r1, UTM 33N 533996 7678250

Q_q48 UTM 33N 534170.7 7677818.4

Image 4 . Successful database- query pair from the second GSV-query experiment

Qualitative inspection of the retrieval results reveals that the network often focuses on large-scale background structures rather than the specific foreground objects that humans would consider distinctive. For instance, in successful query in the images , the input image shows both small houses and a hillside in the background. Although the database contains a frame that clearly depicts the same house and hillside combination. The top-ranked results consist mainly of images showing only the hillside from slightly different positions. Under the UTM-based evaluation protocol, these results are still counted as correct because they fall within the 500–800 m success radius. However, from a human perspective, the retrieved views do not capture the most informative foreground structure (the house). This suggests that, in this environment, the ResNet18–GeM descriptor tends to represent the broader landscape context (e.g., hills and terrain) more strongly than small man-made structures, and that the ranking is driven by those dominant background cues.

DB_q00_r1, UTM 33N 533995 7678250

Q_q00, UTM 33N 526950.8 7669727.5

DB_q01_r1, UTM 33N 533995 7678250

Q_q01, UTM 33N 526950.84 7669727.47

DB_q02_r1, UTM 33N 533995 7678250

Q_q02, UTM 33N 526992.5 7669716.5

DB_q03_r1, UTM 33N 533995 7678250

Q_q03, UTM 33N 527059.5 7669716.0

DB_q04_r1, UTM 33N 533995 7678250

Q_q04, UTM 33N 527089.7 7669717.2

DB_q05_r1, UTM 33N 533995 7678250

Q_q05, UTM 33N 527099.93 7669717.65

Image 5. Failed query–database pair from the second GSV-query experiment

Here, all four successful pairs share the same database image and UTM coordinate. And,all six failed pairs share another database image and UTM coordinate. In all ten cases, the query positions are distinct. This behaviour is a consequence of the spatial distribution of the database frames. Because the car-camera images are concentrated around a few locations, the nearest-neighbour search in UTM space repeatedly selects the same database frame for multiple GSV queries. For the four successful queries, the nearest database frame lies within an 800 m radius and is counted as a correct match. For the six failed queries, the corresponding nearest database frame lies outside the radius and is therefore labelled as a localisation failure.

(a) Car-mounted camera query image (University of Vaasa, 2024).

(b) Rank-1 retrieved database image from the same sequence (University of Vaasa, 2024).

Image 6. In-domain retrieval example

Car-mounted camera query image (a) and (b) is Rank-1 retrieved database image from the same sequence, visually almost identical and co-located, illustrating correct behavior under matched conditions (R@1 = 100%). In this case, the query set is considerably larger (2,718 images) compared to the Google Street View query set (only 100 images). As a result, the domain gap was minimal, and the query successfully matched. The two images appear almost identical.

5.7. Compute environment and reproducibility

Experiments were executed in Google Colab (CPU). Key packages were: PyTorch 2.3.1+cpu, Torchvision 0.18.1+cpu, FAISS-CPU, timm 0.9.x, NumPy/SciPy, and gdown for weights. The benchmark repository (gmberton/deep-visual-geo-localization-benchmark) was freshly cloned per run, and the Pitts30k ResNet18+GeM checkpoint was stored in weights. Evaluation logs, the final .mat, the GSV metadata, CSV, flow diagrams, and outputs from the UTM sanity checks were archived.

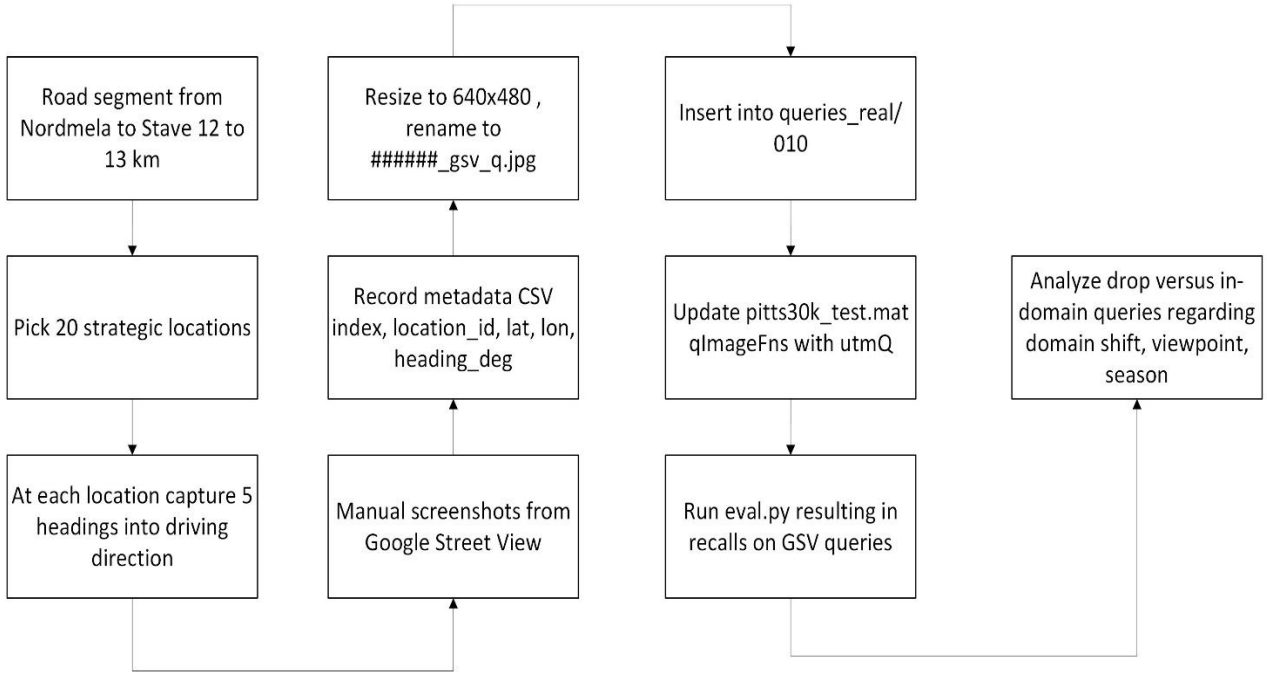

Figure 5. Creating a street view–based query set aligned to the same road for domain-shift evaluation.

6. Conclusion

Two experiments were conducted using a fixed database of car-camera images and two types of query data. In the in-domain setting (car-to-car; DB = 8,157; Q = 2,718), the system achieved Recall@1, Recall@5, Recall@10, and Recall@20 of 100% at a 100 m radius. Consistent with near-duplicate views captured within small temporal offsets. In the cross-domain setting (GSV-to-car), recall remained low under strict radii and increased only modestly as the radius was relaxed. It's indicating that the dominant failure mode is the domain gap between Google Street View and car-mounted camera imagery (differences in viewpoint, focal length/field of view, camera geometry, and appearance), rather than a limitation of the implementation or database coverage.

6.1 Improving the localization accuracy with GSV images

A practical way to improve localization with GSV queries is to reduce the geometric mismatch between GSV images and the car-camera database. One clear approach is focal-length correction. In this approach, GSV query images are reprojected so that their focal length and field of view match those of the Pitts30k database images.

In practice, the GSV panorama is first converted to a rectilinear view using the car-camera intrinsics as the target. The view is then leveled to align the horizon. After that, it is cropped and resized to the evaluation resolution (e.g., 480×640) and normalized in the same way as the benchmark.

This focal-length/field-of-view correction reduces perspective and scale differences that degrade global descriptors such as ResNet18-GeM. It also integrates into the current pipeline without changing the model.

Additional small refinements are possible. Photometric harmonization can reduce appearance differences. Small roll/pitch adjustments can better align the road plane. Evaluating a few nearby headings when selecting the GSV crop can further stabilize retrieval.

If resources allow, light fine-tuning on Pitts30k together with a small set of focal-length-corrected GSV crops can adapt the descriptor to the target domain, while keeping the retrieval protocol unchanged.

6.2 Limitations and future work

The in-domain 100% result reflects near-duplicate imagery and does not imply robustness across domains. The cross-domain study used single-image, single-crop GSV queries and a Pitts30k-pretrained network; broader improvements are expected once focal-length/field-of-view correction is implemented and evaluated. Future work will implement this reprojection to the car-camera intrinsics, report its impact on Recall@ k at 100–200 m, and examine small sets of alternative GSV headings with horizon alignment.

References:

Ali-Bey, A., Chaib-Draa, B., & Giguère, P. (2023). MixVPR: Feature Mixing for Visual Place Recognition. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2998–3007. https://doi.org/10.1109/WACV56688.2023.00301

Garg, S., Fischer, T., & Milford, M. (2021). Where is your place, visual place recognition? *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21)*, 4615–4622. https://doi.org/10.24963/ijcai.2021/603

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., & Milford, M. J. (2016). Visual place recognition: A survey. *IEEE Transactions on Robotics, 32*(1), 1–19. https://doi.org/10.1109/TRO.2015.2496823

Masone, C., & Caputo, B. (2021). A survey on deep visual place recognition. *IEEE Access, 9*, 19516–19547. https://doi.org/10.1109/ACCESS.2021.3054937

Meng, L., Yang, L., Yang, W., & Zhang, L. (2022). A survey of GNSS spoofing and anti-spoofing technology. *Remote Sensing, 14*(19), 4826. https://doi.org/10.3390/rs14194826

Capuano, V., Xu, L., & Estrada Benavides, J. (2023). *Smartphone MEMS accelerometer and gyroscope measurement errors: Laboratory testing and analysis of the effects on positioning performance.* Sensors, 23(17), 7609. https://doi.org/10.3390/s23177609

Hou, Z., Liang, H., Wu, S., Zhang, X., & Hu, W. (2025). *A review of inertial positioning error suppression and accuracy improvement methods for underground pipelines.* Buildings, 15(21), 3904. https://doi.org/10.3390/buildings15213904

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)* (Vol. 1, pp. 886–893). IEEE. https://doi.org/10.1109/CVPR.2005.177

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84–90. https://doi.org/10.1145/3065386

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from http://www.deeplearningbook.org

Zeng, Z., Hu, J., Huang, K., & Tan, T. (2018). Place recognition: An overview of vision perspective. *Applied Sciences, 8*(11), 2257. https://doi.org/10.3390/app8112257

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 815–823). https://doi.org/10.1109/CVPR.2015.7298682

Mereu, R., Trivigno, G., Berton, G., Masone, C., & Caputo, B. (2022). Learning sequential descriptors for sequence-based visual place recognition. *IEEE Robotics and Automation Letters, 7*(4), 10383–10390. https://doi.org/10.1109/LRA.2022.3194310

Wang, Y., Qiu, Y., Cheng, P., & Zhang, J. (2023). Hybrid CNN–Transformer features for visual place recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 33*(3), 1109–1122. https://doi.org/10.1109/TCSVT.2022.3212434

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*, 53. https://doi.org/10.1186/s40537-021-00444-8

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (pp. 448–456). https://arxiv.org/abs/1502.03167

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In

Proceedings of the 27th International Conference on Machine Learning (ICML) (pp. 807–814). https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf

Zhu, Y., Wang, J., Xie, L., & Zheng, L. (2018). Attention-based pyramid aggregation network for visual place recognition. *Proceedings of the 26th ACM International Conference on Multimedia* (pp. 99–107). https://doi.org/10.1145/3240508.3240525

Hu, Z., & Bors, A. G. (2023). Co-attention enabled content-based image retrieval. *Neural Networks, 164*, 245–263. https://doi.org/10.1016/j.neunet.2023.04.009

Lezama, F., González, G. G., Larroca, F., & Capdehourat, G. (2021). Indoor localization using graph neural networks. In *2021 IEEE URUCON* (pp. 51–54). https://doi.org/10.1109/URUCON53396.2021.9647082 OUCI

Rehman, I., Hazhirkarzar, B., & Patel, B. C. (2025). Anatomy, head and neck, eye. In StatPearls . StatPearls Publishing. https://pubmed.ncbi.nlm.nih.gov/29494035

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 886–893.

Schubert, S., Neubert, P., Garg, S., Milford, M., & Fischer, T. (2024). *Visual place recognition: A tutorial*. IEEE Robotics and Automation Magazine, 31(3), 103–120. https://doi.org/10.1109/MRA.2023.3322852

Berton, G., Trivigno, G., Caputo, B., & Masone, C. (2023). EigenPlaces: Training viewpoint robust models for visual place recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11080–11090. https://openaccess.thecvf.com/content/ICCV2023/papers/Berton_EigenPlaces_Training_Viewpoint_Robust_Models_for_Visual_Place_Recognition_ICCV_2023_paper.pdf

Zhu, S., Yang, L., Chen, C., Shah, M., Shen, X., & Wang, H. (2023). *R2Former: Unified retrieval and reranking transformer for place recognition*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13238–13247). https://openaccess.thecvf.com/content/CVPR2023/papers/Zhu_R2Former_Unified_Retrieval_and_Reranking_Transformer_for_Place_Recognition_CVPR_2023_paper.pdf

Izquierdo, S., & Civera, J. (2024). Optimal transport aggregation for visual place recognition.

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, 9312-9321. https://openaccess.thecvf.com/content/CVPR2024/papers/Izquierdo_Optimal_Transport_Aggregation_for_Visual_Place_Recognition_CVPR_2024_paper.pdf

Leyva-Vallina, M., Strisciuglio, N., & Petkov, N. (2023). Data-efficient large scale place recognition with graded similarity supervision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3901-3910. https://openaccess.thecvf.com/content/CVPR2023/papers/Leyva-Vallina_Data-Efficient_Large_Scale_Place_Recognition_With_Graded_Similarity_Supervision_CVPR_2023_paper.pdf

Zhu, Y., & Yan, W. Q. (2022). Traffic sign recognition based on deep learning. *Multimedia Tools and Applications, 81*(3), 17779–17791. https://doi.org/10.1007/s11042-022-12163-0

Chen, C., Cheng, Z., Liu, X., Li, Y., Ding, L., Wang, R., & Feng, C. (2025). Self-supervised place recognition by refining temporal and featural pseudo labels from panoramic data. *IEEE Robotics and Automation Letters, 10*(1), 248–255. https://doi.org/10.1109/LRA.2024.3495584

Fujiyoshi, H., Hirakawa, T., & Yamas, T. (2019). Deep learning-based image recognition for autonomous driving. *IATSS Research, 43*(4), 244–252. https://doi.org/10.1016/j.iatssr.2019.11.008

Lee, H.-S., Lee, S.-M., & Kim, M.-K. (2024). Developing forest road recognition technology using deep learning-based image processing. *Forests, 15*(8), 1469. https://doi.org/10.3390/f15081469

Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., O'Hanlon, B. W., & Kintner, P. M. (2008). *Assessing the spoofing threat: Development of a portable GPS civilian spoofer*. Proceedings of the Institute of Navigation GNSS Conference.

Reuters. (2024, February 14). *Airlines report GPS signal loss over Baltic amid jamming concerns*. Retrieved from https://www.reuters.com

Cohen, N., & Klein, I. (2024). Inertial navigation meets deep learning: A survey of current trends and future directions. *Results in Engineering, 24*, 103565. https://doi.org/10.1016/j.rineng.2024.103565T

Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5297–5307. https://arxiv.org/pdf/1511.07247

Garg, S., & Milford, M. (2021). SeqNet: Learning Descriptors for Sequence-Based Hierarchical Place Recognition. *arXiv preprint arXiv:2102.11603*. https://arxiv.org/abs/2102.11603

Garg, S., & Milford, M. (2021). SeqNetVLAD vs PointNetVLAD: Image sequence vs 3D point clouds for day-night place recognition. *Proceedings of the CVPR 2021 Workshop on 3D Vision and Robotics (3DVR)*. https://arxiv.org/abs/2106.11481

Berton, G., Masone, C., & Caputo, B. (2022). Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4868–4878). IEEE. https://doi.org/10.1109/CVPR52688.2022.00483

Berton, G., Mereu, R., Trivigno, G., Masone, C., Csurka, G., Sattler, T., & Caputo, B. (2022). Deep visual geo-localization benchmark. *arXiv preprint arXiv:2204.03444*. https://arxiv.org/abs/2204.03444

Gronat, P., Matas, J., Obdrzalek, S., & Pajdla, T. (2013). Learning and calibrating per-pixel surface prediction. *CVPR 2013*. Retrieved from https://openaccess.thecvf.com/content_cvpr_2013/papers/Gronat_Learning_and_Calibrating_2013_CVPR_paper.pdf

Ke, Y., & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)* (Vol. 2, pp. 506–513). IEEE. https://doi.org/10.1109/CVPR.2004.1315206

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer. Retrieved from https://szeliski.org/Book/

Zaffar, M., Garg, S., Milford, M., Kooij, J., Flynn, D., McDonald-Maier, K., & Ehsan, S. (2021). VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. *International Journal of Computer Vision, 129*(7), 2136–2174. https://doi.org/10.1007/s11263-021-01469-5

Johnson, J., Douze, M., & Jégou, H. (2017). Billion-scale similarity search with GPUs. *arXiv preprint*. https://arxiv.org/abs/1702.08734

Wang, H., Hammarstrand, L., & Felsberg, M. (2023). An efficient approach to scalable visual localization using compressed descriptors and satellite imagery. *Linköping University Electronic Press*. https://www.diva-portal.org/smash/get/diva2:1890661/FULLTEXT01.pdf

Wang, R., Shen, Y., Zuo, W., Zhou, S., & Zheng, N. (2022). TransVPR: Transformer-based place recognition with multi-level attention aggregation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13638–13647. https://doi.org/10.1109/CVPR52688.2022.01328

Google.(2024). Screenshot (q_000078@532289@7672770@_gsv_q) [Google Street View image]. Google Maps. https://www.google.com/maps

University of Vaasa. (2024). *Jammertest dataset – video and images recorded during the Jammertest event in Norway* [Research data]. University of Vaasa. https://nc.uwasa.fi/index.php/s/8jMyeoMjrExCjif