Efficient visual place recognition in changing environments for resource-constrained platforms

Bruno Rafael Queirós Arcanjo

A thesis submitted for the degree of

Doctor of Philosophy in Computer Science

School of Computer Science and Electronic Engineering

University of Essex

January 2025

Abstract

Visual Place Recognition (VPR) enables autonomous systems to localize themselves within their environment using image information, a crucial component of robotic navigation. The affordability, availability, and versatility of current camera sensors makes VPR an attractive option for many mobile robotic applications. However, performing highly reliable VPR is a complicated task, as a place's appearance can be significantly altered with changes in illumination, visiting viewpoint, seasons, weather, and movement of dynamic elements. Conversely, two distant places within the same environment can appear similar, a problem known as perceptual aliasing. In recent years, VPR techniques built upon Convolutional Neural Networks (CNNs) have consistently improved state-of-the-art performance. However, these approaches continuously tend to larger models, increasing computation and thus requiring more powerful hardware. While the training process of these models can take place in extremely powerful hardware, targeted mobile robotic applications often operate with low-end hardware, usually due to cost or size, rendering CNN based solutions unsuitable, and making lightweight VPR techniques highly desirable.

This thesis addresses the computational shortcomings of CNN-based methods by proposing novel algorithms to perform highly efficient VPR, primarily focusing on bio-inspired and multi-model approaches. The first two contributed algorithms are based on DrosoNet, an exceptionally compact model inspired by the odor processing abilities of the fruit fly. Using multiple of these small units in tandem, the proposed Region-DrosoNet VPR algorithm achieves comparable performance to expensive state-of-the-art techniques, with an AUC of 0.94 on the challenging Nordland Winter dataset, at a fraction of the computational cost, requiring only 9 milliseconds to perform a match. Moreover,

in an attempt to improve efficiency in standard multi-model VPR algorithms, an adaptive technique-switching mechanism is proposed which selects the most apt VPR techniques for the current environment, without access to ground-truth information. The system, dubbed A-MuSIC, achieves an average AUC of 0.88, a substantial increase from the 0.66 of the current multi-technique state-of-the-art, while requiring only 668 milliseconds to perform a match versus the state-of-the-art's 1698 milliseconds.

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others except as specified in the text and Acknowledgements.

Bruno Arcanjo

January 2025

To my grandmothers.

Acknowledgements

I consider myself to have been extremely lucky over the past four years, and I am afraid no amount of words can truly express my gratitude towards those who have supported me. Nevertheless, I leave here my kind thanks to both my supervisors, Dr. Shoaib Ehsan and Prof. Klaus McDonald-Maier, and close collaborator Prof. Michael Milford, who have provided me with all the resources needed during these years. I would also like to express my deepest gratitude to my close colleague, mentor, and friend Bruno Ferrarini for the many productive conversations on Visual Place Recognition, amongst other interesting topics. A heartfelt thank you to Maria Waheed, to whom I was able to relate on the many struggles of a starting research journey. And because I believe emotional support is just as important, I would like to thank my closest friends and family, who have been there for me on every misstep.

Contents

List of Figures

List of Tables

Abbreviations

ANN	Artificial Neural Network
BNN	Binary Neural Network
BoW	Bag of Words
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DL	Deep Learning
DOF	Degree of Freedom
EP	Extended Precision
FN	False Negative
FP	False Positive
GPU	Graphical Processing Unit
HOG	Histogram of Oriented Gradients
LCD	Loop Closure Detection
PR-Curve	Precision-Recall Curve
SLAM	Simultaneous Localization and Mapping
SNN	Spiking Neural Network

TN	True Negative
TP	True Positive
UAV	Unmanned aerial vehicle
ViT	Vision Transformer
VLAD	Vector of Locally Aggregated Descriptors
VPR	Visual Place Recognition

Chapter 1

Introduction

Automation is at the heart of robotics. Autonomous robots have the potential to revolutionize the way humans work, either by augmenting human capabilities through human-robot interaction [1] or by fully replacing people in performing dangerous tasks [2]. Indeed, the last decades have seen major developments in robot automation, ranging from space [3, 4] and marine [5] exploration to mundane, everyday applications [6].

Localization is a fundamental part of long-term, autonomous navigation, as robots need to know their current position to update their internal maps and plan future actions. Visual Place Recognition (VPR) completes the localization task using images, matching the robot's current camera view to a previously observed scene. VPR offers unique advantages compared to other common methods of self-localization, such as Global Positioning System (GPS), as it does not require a network signal, and thus being able to operate in remote environments. Furthermore, camera sensors are available in a wide variety of sizes, budgets, qualities, and optical characteristics, being an extremely versatile option for any robotic application. However, despite these advantages, performing robust and reliable VPR remains an unsolved problem which continues to receive substantial attention from the research community. The rest of this section defines the VPR problem, explores its current challenges, and details the contributions of this thesis to the VPR field.

1.1 The VPR Task

The VPR problem is easily defined: given the current view of the robot's camera, has this place been visited before? And, if so, which place is it [7]? This task proposition poses important considerations on what exactly a "place" is. While a human might consider two images to represent the same place even if they show completely opposite views of the physical location, such would not work for a purely image-based system. One must also define what scale to operate at: are places geometric shapes of pre-determined size? Streets? Cities? Countries? This decision has important implications when it comes to the training and deployment of VPR methods, and has recently received significant attention [8, 9]. To circumvent these issues, this thesis follows the widely used practice of simply framing VPR as matching images of the same location with some level of overlapping line of sight [10].

Despite its straightforward goal, performing long-term VPR remains a complicated task. The next section explores the main challenges in long-term Visual Place Recognition.

1.2 VPR Challenges

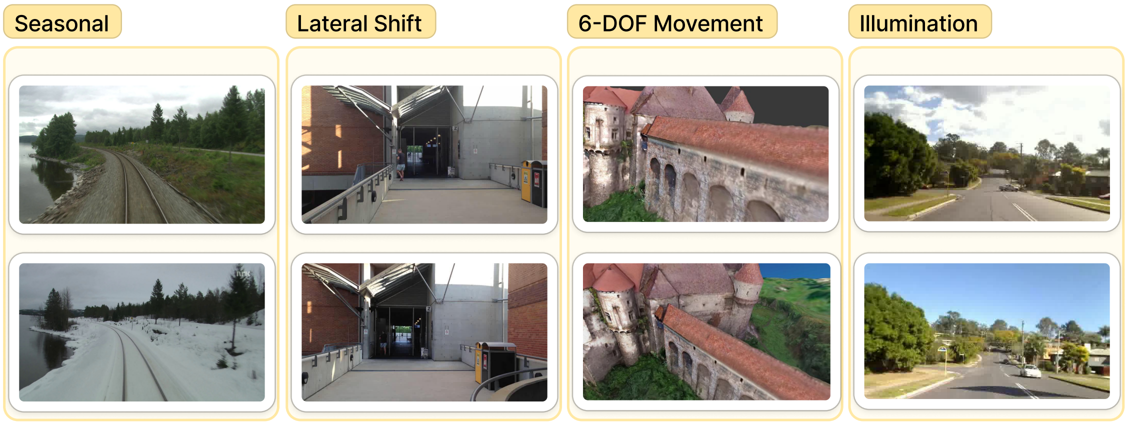


Fig. 1.1. Some of the different visual challenges faced by the VPR community.

VPR sits at the intersection of two major computer science fields: robotics and Computer Vision (CV), thus inheriting open research problems from both. Moreover, representing and matching places long-term comes with its own set of unique challenges.

Environments and their appearance change significantly over time and in a myriad of ways, as shown in Fig. 1.1. Seasonal variations, illumination changes due to daylight cycles, and dynamic elements such as cars or people, can substantially alter how a place looks at different time points. Conversely, different places, especially in the same environment, can look remarkably similar, a problem known as perceptual aliasing. Moreover, even if an environment stays relatively unchanged, revisiting it from a different perspective can make recognition extremely difficult. Given these visual challenges, a place representation must be sufficiently distinct to avoid perceptual aliasing while similar enough to be recognized when subjected to large appearance or viewpoint changes.

Further complications arise when considering practical VPR implementations for mobile robotics, as these platforms are often equipped with low-end hardware operating on battery power. Furthermore, VPR is just one of the many components of an autonomous navigation system, and freeing up resources for other tasks is of extreme importance. As such, addressing the above visual challenges is not a simple case of designing increasingly complex algorithms in the search of matching accuracy. VPR methods should strive to achieve a suitable trade-off between computational cost and VPR performance, the former being measured in terms of memory usage, power consumption, and matching latency. This thesis' main goal is to contribute VPR methods designed with a focus on computational efficiency, thus suitable for hardware constrained robotics.

1.3 Thesis Contributions

This thesis primarily offers the following contributions:

1. Firstly, this thesis addresses the problem of VPR in extremely hardware constrained platforms by presenting a novel lightweight VPR pipeline. The first component of this system is DrosoNet, a compact bio-inspired neural network classifier which achieves impressive VPR performance relative to its computational requirements. To further improve VPR performance, a Voting mechanism utilizing several DrosoNets is designed, taking advantage of DrosoNet's stochastic characteristics. The pipeline achieves a modest boost in VPR performance compared to other lightweight techniques on several datasets, with an AUC of 0.98 on the Nordland Fall dataset compared to the 0.92 of the next best lightweight technique. In computational efficiency, the improvement is more pronounced, with an inference time of only 3 milliseconds versus the 166 milliseconds of the next fastest technique.

2. The second contribution builds upon the previous work to vastly improve the VPR

performance of a multi-DrosoNet system while retaining a low-computational profile. A novel training approach is proposed where distinct DrosoNet groups are trained on different image regions. The method aims to increase differentiation between DrosoNets and to reduce the amount of information that each model must learn to recognize. To join the outputs of all groups, an improved voting protocol is also proposed. The overall system, dubbed RegionDrosoNet, achieves up to twice the VPR performance when compared to its predecessor and other lightweight techniques. Moreover, it competes with computationally demanding VPR methods on certain datasets while having a matching latency up to three orders of magnitude smaller.

3. The third contribution continues the work with multi-technique VPR solutions, addressing the problem of identifying the most suitable technique for a given query frame. While the previous two contributions address lightweight multi-technique VPR using multiple of the same model, this work uses several different methods in tandem. A method using sequential information to simultaneously improve VPR performance and switch between techniques in a set is proposed. This sequential metric is then used to design an adaptive mechanism that keeps only the most used techniques active, addressing the redundant computation problem of multi-technique VPR.

1.4 Thesis Structure

The rest of this thesis contains six chapters, organized as follows.

Chapter 2 provides an overview of the state of Visual Place Recognition literature. The chapter starts by exploring the general VPR problem, providing context to its motivation, challenges, historical and current approaches, and evaluation frameworks. The chapter then delves into the problem of computationally efficient VPR, and how bio-inspired algorithms present themselves as a viable solution. Finally, an overview of multi-method VPR approaches is given, covering technique fusion and technique switching.

Chapter 3 proposes a novel, highly computationally efficient VPR method, with two main innovative contributions. The first is DrosoNet, a small and extremely fast bio-inspired neural network classifier designed for VPR. DrosoNet closely resembles drosophila melanogaster's (the common fruit-fly) odor recognizing cognitive processes, achieving relatively competitive VPR performance. The second proposal is a voting mechanism that leverages multiple VPR classifier models to achieve more robust performance. When employed with DrosoNet as its baseline, the voting mechanism outperforms other lightweight VPR techniques on several benchmark datasets at a small fraction of their computational cost.

Chapter 4 aims to substantially improve the absolute VPR performance of a multi-DrosoNet system, while maintaining a low computational profile. The author proposes a novel training process in which different DrosoNet groups are trained on different regions of the reference traversal images. During inference, each group makes its prediction only on its assigned region of the query image, and all predictions are combined with an improved voting scheme. The overall method, dubbed RegionDrosoNet, not only outperforms other lightweight VPR methods but also competes with computationally expensive ones on some benchmark datasets.

Chapter 5 continues exploring multi-technique approaches to the VPR problem by proposing a VPR system which leverages sequential information to select the most suitable technique for a given query frame. On a per-frame basis, Multi-Sequential Information Consistency (MuSIC) analyses the frame-to-frame continuity of each technique in a set and selects the most cohesive technique with which to perform VPR. Unlike other multi-technique approaches, MuSIC does not rely on additional ground-truth information nor on brute-force fusion. The combination of sequential information with a multi-method paradigm provides a substantial increase in VPR performance across several benchmark datasets. This chapter also addresses the shortcoming of increased computational effort inherent to the multi-technique VPR paradigm. Adaptive Multi-Sequential Information Consistency (A-MuSIC) continues using temporal cohesion to select the most suitable VPR technique for a query, but it avoids running all VPR methods in the set for every query image. Instead, A-MuSIC tracks the techniques which are contributing the most in current navigation and disables the remaining ones. Upon a statistically significant change in sequential cohesion, the system issues a re-selection stage in which all techniques are evaluated and forms a new working subset.

Chapter 6 concludes this thesis with final reflections over the presented work and considerations for future research directions.

1.5 List of Publications

The author of this thesis is the main contributor for the following listed publications, and is thus listed as first author. He designed the proposed algorithms, developed the code to implement them, ran the necessary experiments, and wrote the submitted manuscripts as well as the necessary corrections for their publication. The remaining authors provided important feedback and discussion on each step, as well as guidance on the publication process.

1. B. Arcanjo, B. Ferrarini, M. Milford, K. D. McDonald-Maier and S. Ehsan, "An Efficient and Scalable Collection of Fly-Inspired Voting Units for Visual Place Recognition in Changing Environments," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2527-2534, April 2022, doi: 10.1109/LRA.2022.3140827

2. B. Arcanjo, B. Ferrarini, M. Fasli, M. Milford, K. D. McDonald-Maier and S. Ehsan, "Aggregating Multiple Bio-Inspired Image Region Classifiers for Effective and Lightweight Visual Place Recognition," in IEEE Robotics and Automation Letters, vol. 9, no. 4, pp. 3315-3322, April 2024, doi: 10.1109/LRA.2024.3367275

3. B. Arcanjo, B. Ferrarini, M. Milford, K. D. McDonald-Maier and S. Ehsan, "Multi-Technique Sequential Information Consistency For Dynamic Visual Place Recognition In Changing Environments," https://arxiv.org/abs/2401.08263, *Under Review.*

4. B. Arcanjo, B. Ferrarini, M. Milford, K. D. McDonald-Maier and S. Ehsan, "A-MuSIC: An Adaptive Ensemble System For Visual Place Recognition In Changing Environments," https://arxiv.org/abs/2303.14247, *Under Review.*

Chapter 2

Literature Review

This chapter aims to familiarize the reader with the crucial concepts explored in this thesis. The first section starts by addressing the fundamental task of Visual Place Recognition, presenting its definition and common implementation pipeline. It then draws focus to historical and current state-of-the-art solutions to the VPR problem, dividing the literature into several sub-topics: handcrafted feature descriptors, learned feature descriptors, visual transformers, re-ranking, and VPR as a pure classification task. The VPR overview finishes with an explanation of the usual VPR evaluation framework, touching on performance metrics and benchmark datasets. The chapter then delves into the topics of biological inspired and computational efficient VPR, both of major importance for this thesis' first two contributions. The final section motivates the remaining two contributions by exploring multi-technique approaches to VPR, a paradigm in which several VPR methods are combined to increase performance and reliability.

2.1 Visual Place Recognition

Visual Place Recognition is the ability of a system to determine its location in the runtime environment using only image information [7]. Such a capability is extremely desired in autonomous, long-term navigation, in which visual self-localization can play an essential role by performing loop closure detection in Simultaneous Localization and Mapping (SLAM) pipelines [11, 12, 13]. As aforementioned, VPR is particularly attractive to mobile robotics as camera sensors are highly versatile. Moreover, VPR is also strongly biologically motivated: humans and other mammals primarily use vision to navigate in their environment [14].

VPR is most often framed as an image retrieval task in which a database of image representations is searched for the best match to the robot's current camera view [15, 16]. As depicted in Fig. 2.1, the database is constructed using a reference traversal of the environment in an offline setting, i.e., before navigation starts [17]. During this stage, the employed VPR technique, or more specifically its image descriptor component, encodes each reference place into an image descriptor which is then stored for subsequent online matching. During actual navigation, the robot's view is taken as a query image, encoded using the same VPR technique, and compared to the descriptor database, retrieving the most similar reference place.

Given that a place's appearance can change dramatically when visited at different times or from different angles [18, 19], the core of VPR becomes an image representation problem [20]: How to represent a place such that it can be recognized when under different appearance conditions? Conversely, how to ensure that similar, yet distant, places in the same environment are properly distinguished? Moreover, a practical VPR implement-

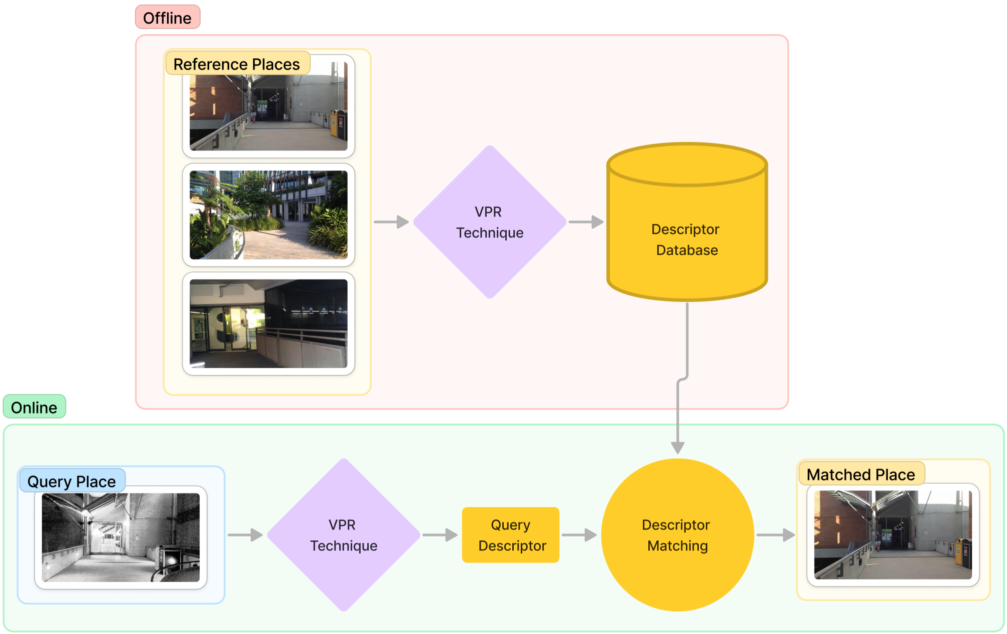


Fig. 2.1. High-level overview of a typical VPR pipeline. In the offline step, a database of image descriptors is constructed using a chosen VPR technique. During online navigation, the camera's view is used as a query image and encoded using the same VPR method. The query descriptor is then compared to the existing descriptor database and the most similar reference place is retrieved.

ation must make further considerations. In particular, on-board computational resources become a limiting factor to many VPR applications, as many mobile robots operate with low-end hardware [21, 22]. Thus, the search for adequate image representations cannot be trivialized to simply increasing the size and complexity of visual models.

The rest of this section presents an overview of the several approaches proposed to tackle the VPR representation task, starting from handcrafted descriptors and then moving to Convolutional Neural Network based approaches. The section also explores casting VPR as a stricter classification task, which has recently gained traction [9, 23] and is of importance to this thesis' contributions. The section finishes with an explanation on conducting VPR evaluation, covering common evaluation metrics and benchmark datasets.

2.1.1 Handcrafted Feature Descriptors

The appearance of a place can change dramatically from visit to visit, and thus highly robust image descriptors should encode only the place's most defining and persistent visual traits. Furthermore, these descriptors must store information in an easily comparable and searchable format as to facilitate downstream matching. Handcrafted feature descriptors are a category of deep learning free algorithms that aim to generate such resilient representations and were the basis for the first solutions to the VPR problem. The following discussion sticks to the standard practice of dividing handcrafted features into two subcategories: local and global descriptors.

Handcrafted Local Descriptors

Handcrafted local feature descriptors first detect the image's regions of interest (ROI) and then encode a pixel patch around (and including) the identified regions into a descriptor. SURF [24] offers both feature detection and encoding, and has been employed in several VPR applications [25, 26]. SIFT [27] also offers both a feature detector and descriptor, but it's less computationally efficient than SURF due to its larger descriptor size. Nevertheless, VPR solutions such as [28, 29] have successfully incorporated SIFT.

Comparing two images by computing the pairwise correspondence between all their local features would quickly become computationally unfeasible, as each image can contain thousands of descriptors [30]. Moreover, not all features are informative enough to be useful [31]. Local feature aggregation methods aim to both improve place matching efficiency and diminish the impact of distracting features. Visual Bag-of-Words (BoW) [32] forms a discrete feature space (dictionary) by clustering descriptors into centroids (visual words). The descriptors of an image can then be assigned to visual words, with the resulting histogram of the assignment being used as the image representation. FAB-MAP [25] successfully employs a BoW representation strategy to perform VPR, in conjunction with SURF feature detection and encoding. Methods like VLAD [33, 34] and Fisher Vectors [35] refine the BoW approach, improving upon representation strength and reducing dictionary size, respectively.

The main efficiency bottleneck of handcrafted local features is the demanding matching process [36]. In an effort to improve matching efficiency, DBoW [37] proposes the use of binary BoW representations clustered using k-means++ [38]. While DBoW substantially improves place matching time, its features lack rotation and scale invariance, reducing VPR reliability when dealing with large viewpoint variations. Moreover, its dictionary requires a long training process, a shortcoming that IBoW-LCD [39] addresses by building its word dictionary iteratively. Nevertheless, these algorithms continue to struggle when faced with large appearance changes.

Handcrafted Global Descriptors

Rather than encoding specific patches of an image, handcrafted global descriptors such as GIST [40] describe the whole image at once, potentially saving computational effort by skipping the ROI detection and local feature matching steps [16]. In [41], the authors perform visual loop closure detection by using GIST in tandem with the local feature descriptor BRIEF [42]. Standalone GIST has also been proposed for visual place matching in several works [43, 44]. While Histogram-Of-Oriented Gradients (HOG) [45] has been traditionally employed as a descriptor for local features [27, 46], it has also been explored as a whole-image descriptor for VPR [47]. Despite their efficiency and algorithmic simplicity benefits, global descriptors are more prone to matching errors when dealing with dynamic elements and viewpoint variations [48].

Between handcrafted local and global descriptors, the local version has been more successful in the VPR field, with FAB-MAP [25] having been one of the first and most popular VPR systems. Indeed, the high robustness to viewpoint variations showcased by local descriptors grants it a unique advantage over its global counterparts. However, perhaps the most interesting takeaway from the comparison is that learned feature descriptors evolved to combine local features into global representations, gaining advantages from both approaches. These will be promptly explored in the next Section.

2.1.2 Learned Feature Descriptors

In their 2016 VPR survey, Lowry *et al.* [7] conclude that the use of Convolutional Neural Networks (CNNs) in VPR solutions "is a worthwhile direction for future research". Just five years later, in 2021, two large surveys [49, 48] solely dedicated to deep learning (DL) based VPR were published, each with over 200 references. Indeed, as with many other robotic fields, deep learning [50] has become widely popular in the VPR community.

From Fully Connected to Convolutional Features

Convolutional Neural Networks are the most used subtype of neural network in computer vision applications [51], and also in VPR. While the introduction of CNNs dates back to 1992 [52], the technology started seeing massive adoption after the publication of AlexNet [53], in 2012. This delay in widespread adoption can be attributed to two main reasons [49]. Firstly, CNN architectures often contain millions of parameters, making the training and deployment of these models computationally expensive. Secondly, the training process requires large amounts of labeled data. The latest decade has seen massive advancements in graphical processing unit (GPU) computing [54], DL development software [55, 56], and large-scale data collection, allowing for deep learning research to thrive and expand.

Visual features extracted from the fully-connected layers of a trained CNN can be used in Computer Vision image recognition tasks [57, 58, 59, 60], even when not trained specifically for them [61]. This insight first led to the use of general purpose CNNs as feature extractors for Visual Place Recognition, even if they were not specifically trained or designed for VPR. Chen *et al.* [62] first validated this concept by using a CNN [63] pre-trained on the ImageNet dataset [64] as an off-the-shelf feature extractor, showing superior VPR performance when compared to handcrafted features. While later research [65] demonstrated that training a network explicitly for the VPR task improved place retrieval performance, the used fully-connected representations presented serious shortcomings, being computationally expensive, subject to dynamic elements and occlusions, and having a fixed input size [48].

Convolutional features are instead directly computed from the activations of the convolutional layers of a trained CNN. Given a convolutional layer consisting of K feature maps (channels), each of width W and height H, the flattened vector of $K \times W \times H$ elements can be used an image representation [66, 67]. HybridNet and AMOSNet [68] are methods based on this approach and are fine-tuned for VPR by being trained on the SPED dataset [68]. PlaceNet [69] follows the same pattern employing a VGG-16 [70] backbone and being trained on the Places365 dataset.

This approach is also less computationally intensive than fully-connected features, especially when paired with dimensionality reduction techniques [71] such as PCA [72]. However, flattening the feature map erases the spatial relationship information between features, making these representations suffer from the same lack of robustness that afflicts fully connected extractions. The intelligent aggregation of convolutional features was the breakthrough needed for substantially improved robustness.

Aggregation and Pooling of Convolutional Features

Interpreting the activations of a CNN layer as a $W \times H$ grid of K-dimensional descriptors was the key to unlock a variety of feature pooling and aggregation methods with impressive image retrieval performance.

Maximum activation of convolutions (MAC) [73] constructs an image representation by forming a vector in which elements are the maximum value of each descriptor in the feature grid. Regional-Mac (R-MAC) [74] applies the same principle over sub-regions of the feature map. By varying the size of the sub-regions, one can obtain feature vectors corresponding to different scales which are then post-processed into a compact image descriptor. R-MAC has been shown to outperform MAC in image retrieval [75] and has achieved impressive performance in VPR applications [76]. Cross-dimensional weighting (CroW) [77] also relies on pooling and post-processing of the feature map, but it introduces intra-channel and inter-channel feature weighting. Generalized mean pooling (GeM) [78] generalises maximum and average pooling using a parameter which weighs each feature map and can be set manually or fully learned. With GeM, maximum and average pooling become special cases defined by the weighting parameter. In [8], GeM is employed in a VPR pipeline with either a ResNet [79] or Transformer [80, 81, 82] based extractor. MixVPR [83] applies fully connected layers to flattened CNN activations, producing a two-dimensional matrix which is then pooled, normalized, and flattened into the final representation. NetVLAD [15] introduces a new learned pooling layer based on the VLAD [33] algorithm, achieving state-of-the-art VPR performance in 2016. Since its introduction, NetVLAD has been used as the backbone for more complex VPR methods, adding additional steps or information for improved recognition. Patch-NetVLAD [84] uses NetVLAD to identify several patches of interest in the input image and compute a descriptor for each patch which are subsequently used for place matching. Thus, while NetVLAD produces one descriptor per image, Patch-NetVLAD computes several descriptors that need to be spatially matched later in the pipeline. In [85], the authors use NetVLAD to first filter for a subset of suitable reference candidates and then use SuperPoint [86] features and the SuperGlue [87] matcher to select the final match amongst them.

2.1.3 Emergence of the Vision Transformer for VPR

Similarly to the shift from handcrafted to CNN-based features, the introduction of the vision transformer (ViT) [81] is driving VPR to attention-based visual features. The ViT self-attention mechanism is able to dynamically identify and aggregate task-relevant features, leaving out unimportant or even misleading visual information. TransVPR [88] uses multi-level attention to extract image features from several regions of an image. Akin to the usual CNN procedure, these representations are then aggregated together into a global representation and finally filtered with an attention mask. Foundational models [89] are large networks trained on massive datasets, producing extremely robust ViT feature extractors which can be directly used for VPR. AnyLoc [90] uses the DINOv2 [91] foundational model to achieve state-of-the-art performance across several benchmark datasets exhibiting different visual challenges.

While ViT-based approaches demonstrate substantially improved VPR performance, these models are significantly larger than even Convolutional Neural Networks, further increasing the computational cost of both training and inference [92].

2.1.4 Additional Information and Consistency Checks

So far, the discussion has focused on the simplified version of the VPR pipeline (Fig. 2.1), where the query descriptor is compared to the database descriptors and the most similar reference place is deemed a match. However, due to the difficulty of the VPR task, several methods propose the use of extra information pertinent to robotic navigation with the goal of improved performance. As observed in Fig. 2.2, the extra information or stages can be added at several points of the VPR pipeline, with some techniques directly infusing information into the descriptor similarity computation and others enhancing the similarity matrix itself.

Sequence-based Localization

Exploiting the sequential nature of robotic navigation is perhaps the most popular approach to improve VPR matching. Rather than matching a single frame, these methods perform localization over a sequence of query images. In [93], the authors use SIFT descriptors captured from image sequences to perform loop-closure in SLAM. RatSLAM [94], an algorithm inspired by the hippocampus of rodents, also performs SLAM over a sequence of frames. SeqSLAM [95] makes use of the Sum of Absolute Differences (SAD) algorithm to propose featureless sequential navigation. After using SAD to obtain a similarity matrix, SeqSLAM finds the optimal linear path over the similarity scores, that is, it finds the linear path which maximizes the sum of similarities along itself. Addressing the linear limitation, [96] proposes the use of a Hidden Markov Model (HMM) [97] to find non-linear trajectories in the similarity matrix. In [98], SeqSLAM is implemented with CNN features, substantially improving VPR performance. Other works such as [99] and [100] also propose methods to find the best fitting trajectory in the computed similarity matrix. Of course, the main drawback from sequential methods is that VPR performance is dependent on sequence-length, and the larger the sequence, the larger the matching delay.

VPR Re-ranking

A more general approach to enhance VPR performance with extra information is to use a hierarchical pipeline. In this paradigm, also referred to as re-ranking [101], place retrieval is divided into two or more tiers, and only a select few candidates from a tier are passed on to the next, with different filtering information being used at each stage. The core idea is for the first tier to perform a coarse search in a computationally inexpensive manner, and only apply an intensive fine-grained search to the retrieved subset of candidates [102]. Arshad et al. [36], inspired by the use of semantic information in VPR [103, 104, 105], propose an hierarchical VPR method which utilizes semantic information at various stages to filter for the best matching candidates. Hierarchical Multi-Process Fusion [106] operates a three-tier system, with each tier being composed of pre-selected VPR methods. Only the top-candidates from a given tier, obtained by the operation of its respective techniques, are allowed to flow to the next tier. SSM-VPR

[107] uses CNN features in its first stage and re-ranks the best candidates using anchor points for spatial comparison. As previously mentioned, Patch-NetVLAD [84] also performs additional spatial checks on the outputs of NetVLAD.

2.1.5 VPR as a Classification Task

Most of the recent state-of-the-art techniques frame VPR as an image retrieval task performed by comparing representations extracted with a CNN backbone [15, 108]. However, recent works [8, 83] have argued that the training mechanism these methods employ, based on contrastive learning [109] and triplet loss [110], is not scalable to real-world map sizes. As an alternative, the authors of [8] propose that during the training stage VPR be treated as a classification task, where places are grouped into different classes. Framing VPR as a classification task is not without precedent, being a popular approach in the realm of global geo-localization [111, 112]. When well executed, the class grouping process can even improve robustness to viewpoint variations [9]. FlyNet [23] acts as a neural network classifier both in training and place inference. Such a setup has large efficiency benefits, as no memory is used to store image descriptors and no similarity search is conducted. However, the number of places is fixed from the start of training, not allowing for further exploration during navigation.

2.1.6 VPR Evaluation

Benchmark Datasets

Performing reliable VPR requires resilience against a variety of visual challenges. Thus, when evaluating VPR methods, it is important to consider their performance on multiple types of visual changes. Addressing this need, many benchmark Visual Place Recognition datasets have been proposed in the literature. These datasets usually consist of multiple traversals of the same environment, most often recorded at different times or from different visiting viewpoints, simulating real-world VPR use. The usual benchmarking approach is to use one traversal as the reference environment, building a database of image representations, and images in other traversals as queries. The following datasets are used across this thesis to evaluate VPR performance.

The Nordland dataset [113] consists of four train traversals, each recorded in a different yearly season: Summer, Fall, Winter, and Spring. Each traversal consists of a video, i.e., a collection of frames, filmed in the corresponding season, using a camera mounted on the front of the train. The goal of the setup is to capture the same places under different seasonal variations, e.g., snowy in the Winter and Sunny in the Summer. This dataset is mainly used to assess performance against varying strengths of appearance change, depending on which traversals are used as reference and query. For example, using the Summer traversal as reference and Fall as query, one can assess VPR performance with moderate seasonal appearance variation. To test resilience against extreme seasonal changes, the Winter dataset should be used as query.

Also presenting moderate appearance changes, the St. Lucia dataset [114] provides car recorded traversals at different daily times in St. Lucia, Brisbane. These traversals, recorded at different times of the day, contain dynamic elements, mainly other cars, modest viewpoint variations, and some daily illumination changes.

The Oxford RobotCar dataset [115] was also recorded from a moving vehicle at different dates and times. This dataset provides numerous traversals addressing different visual challenges, such as changes in daily illumination, dynamic elements(cars and pedestrians), as well as moderate viewpoint variations.

The Gardens Point dataset [67] presents both changes in illumination and a lateral viewpoint shift, consisting of three sequences recorded in the Queensland University of Technology campus. One sequence filmed during the day from a left perspective, another daily one but from a right perspective, and a third recorded at night from a right perspective.

The Berlin dataset [116], car recorded in three locations of the city, Halense Strasse, Kudamn, and A100, shows stronger viewpoint variations than the previous datasets. As per usual with street recorded sequences, a significant amount of dynamic elements is also present.

Assessing challenging 6 degrees-of-freedom (DOF) viewpoint changes is of special interest to unmanned aerial vehicles (UAVs) [22]. This thesis makes use of two synthetic datasets to evaluate VPR performance in these conditions: Corvin and Lagout [117]. These datasets consist of several simulated flight traversals, each at a different view angle and exhibiting strong scale variations.

The 17 Places dataset [118] consists of several different challenging indoor environments, with one traversal made during the day and another during the night. The dataset can hence be used to assess VPR performance under indoor illumination changes, as well as small amounts of viewpoint shifts and dynamic elements.

These datasets, summarized in Table 2.1, are used in the various experiments conducted throughout this thesis.

TABLE 2.1: DATASET DETAILS

Dataset	Condition	Reference Traverse	Query Traverse	Number of Images
Nordland Winter	Extreme seasonal	Summer	Winter	1000
Nordland Fall	Moderate seasonal	Summer	Fall	1000
Berlin	Strong viewpoint	halen.-2, kudamm-1 and A100-1	halen.-1, kudamm-2 and A100-2	250
Night-Right	Outdoor Illumination; Lateral Shift	Day-Left	Night-Right	200
Day-Right	Lateral Shift	Day-Left	Day-Right	200
St. Lucia	Daylight; Dynamic Elements	Afternoon	Morning	1100
17 Places	Indoor Illumination	Day	Night	2000
Oxford RobotCar	Outdoor Illumination	Night	Sunny Day	200
Lagout 15	6 DOF	0 Degrees	15 Degrees	336
Corvin 30	6 DOF; Scale	0 Degrees	30 Degrees	1000

Evaluation Metrics

The desirability of performing robust and reliable VPR motivated the development of several VPR techniques, and thus the need for objective metrics assessing and comparing their performance also surged. This section introduces multiple VPR performance metrics and discusses their respective use cases.

Precision-Recall curves (PR-Curves) are widely used to evaluate VPR performance [119, 120]. The common setup for VPR evaluation often leads to an imbalanced dataset scenario, for which these curves are a robust metric [121, 122]. In this framing, VPR is evaluated as a classification task with two classes: correctly matched and incorrectly matched. For each query place, only a few reference places are considered to be a correct match, thus the dataset imbalance. Recall and Precision can be computed as:

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

where TP stands for true positive, FP for false positive, and FN for false negative. When performing VPR, techniques output a confidence value for each reference place, signifying its likelihood of being the correct match. By varying the confidence threshold at which a match is deemed correct, a set of Precision-Recall value pairs can be constructed [123, 124]. These value pairs can be plotted on a graph to construct the PR-curve.

Several metrics can be used to summarize and quantify a PR-curve, facilitating comparisons at the cost of detail. Each of these numeric metrics provides different insights into the performance of a VPR technique, evaluating its applicability to specific applications.

The area under the PR-curve (AUC) is a well-established and commonly used metric in VPR benchmarking [125, 124]. It compresses the entire PR-Curve into a numerical value [17], making it a practical comparison tool. However, it does not provide details on important points of the PR-Curve (e.g. the maximum Recall without the introduction of FPs). Moreover, it assumes that Recall and Precision are equally important, which is not always the case [10].

The maximum Recall at 100% Precision (R@100P) is computed by taking the largest Recall value for which Precision is at 100% [98]. Thus, R@100P represents the maximum Recall without the introduction of false positives and should be prioritized by applications where a single incorrect match would result in failure. This used to be important when performing SLAM [123], where a single false match could result in a wrong loop closure [126]. However, the introduction of graph optimizations into the SLAM pipeline has provided resilience against wrong loop closure candidates, and finding enough correct closure candidates is now of higher importance [17].

The R@100P metric has two main shortcomings. Firstly, it is undefined for scenarios where a technique never achieves 100% Precision. Secondly, it provides no information about the lower performance bound of the VPR method. Extended Precision (EP) [120] combines R@100P with Precision at Minimal Recall (PMR) to address these shortcomings. EP can be computed as

$$EP = \frac{PMR + R@100P}{2} \tag{2.3}$$

By definition, if $PMR < 1$, then R@100P is undefined, and is set to 0. In this scenario, according to Eq. 2.3, EP relies only on PMR, being able to provide some performance information. When $PMR = 1$, EP is similar to R@100P.

In the context of VPR, the Recall at K (R@K), also called Recall at N (R@N), measures the rate of queries with a correct match within the K most similar database images

[9, 127, 128]. If all queries have at least one matching database location, which is often the case for VPR, then R@1 and R@100P are equal [17]. Contrary to R@100P, this metric is most suitable for applications which perform additional place candidate validation.

Given its extensive use in the robotics community, the majority of the results in this thesis focus on the PR-AUC metric. A smaller set of experiments also offers EP results with the intent of enriching discussion on different use-cases for the proposed techniques.

2.2 Biological Inspiration for Efficient VPR

While the move to CNN based feature extractors has undoubtedly enabled impressive VPR performance, it significantly increased the computational requirements of the target platform. Indeed, CNN models have gotten increasingly complex, with larger and deeper networks being developed to push the state-of-the-art forward [129, 90]. The recent developments in ViT based VPR further exacerbate this problem, showing no slowdown to the ever increasing computational demands. This section starts by covering the motivation and current solutions for achieving efficient VPR models. It then discusses biological inspired models as a possible approach to provide highly-efficient Visual Place Recognition alternatives.

2.2.1 The Search for Efficiency

Mobile robotic platforms are most often equipped with highly constricted hardware [21, 22] and thus suffer from low compute and small memory packages. Moreover, these devices are battery powered, making power-efficient localization extremely desired for long-term navigation.

The push for more efficient neural networks initiated in the broader CV field, with the goal of making image processing methods available on mobile devices such as smartphones. MobileNets [129] replaces some of its layers with depth-wise separable convolutions, resulting in decreased model size and faster inference. Its second version, MobileNetsV2 [130], introduces inverted residual and linear bottleneck layers to further improve efficiency and image classification performance. In [131] the authors employed MobileNetsV2 enhanced with bilinear feature fusion [132] specifically for visual scene recognition. SqueezeNet [133] introduces several design strategies to reduce a network's size while retaining performance: smaller filters, shallower inputs, and pooling in the deeper layers. Pruning redundant weights in a network is another technique to improve efficiency [134, 135], albeit some concerns have recently been raised with this procedure [136]. Quantization [137, 138, 139] refers to the practice of lowering the precision of a network's parameters, achieving a lower computational footprint, often at the cost of classification performance [20]. Binary Neural Networks (BNNs) [140] are an extreme case of quantization where network weights are reduced to 1-bit precision. BNNs have been proposed specifically for the VPR task in [141, 142], showcasing impressive efficiency provided that the hardware supports binary operations. CALC [143] is a CNN specifically designed for lightweight VPR which learns to recreate a HOG [45] descriptor in an unsupervised fashion. CoHOG [46] also presents itself as a lightweight VPR method; it is a training-free algorithm that uses image-entropy [144] to find regions-of-interest in images and computes a HOG descriptor for each found region.

Given the high reliance on convolutional neural networks for feature extraction, the current state-of-the-art in lightweight VPR solutions is achieved by using a combination of the described approaches to lower the computational cost of the backbone CNN. Practical deployment on real-world systems requires tailoring the models to the available hardware to ensure the correct balance of computation cost to performance. Studies such as [20] show how different efficiency-gaining techniques, such as quantization and pruning, affect VPR performance and should be used as guidance.

2.2.2 Biologically Inspired Models for Efficient VPR

Visual Place Recognition often draws inspiration from the animal kingdom, with even the fundamental definition of "place" being based on the operation of spatial view cells [145] in [10] and place cells [146] in [7]. While it could be argued that any deep learned VPR approach is biologically inspired, since the artificial neuron is vaguely derived from the natural one [147], this is not the usual usage of the term. Instead, bio-inspired algorithms aim to mimic higher-level processes of biological brains. Copying the cognitive mechanisms of small animals such as insects or rodents is of particular interest to efficiency-aware VPR [48]. Indeed, ants [148, 149], bees [150, 151], fruit-flies [152, 153], and rodents [154] are all able to efficiently navigate and interact with their environment at a small fraction of the energy required by current state-of-the-art VPR methods.

RatSLAM [94] takes inspiration from the rodent brain, using a continuous attractor network (CAN) [155, 156] to track odometry and detected landmarks over sequential navigation. CANs have also been used in conjunction with FlyNet: a lightweight VPR model inspired by the brain of the fruit-fly [23]. Interestingly, FlyNet is derived from how the fruit-fly, drosophila melanogaster, processes odours [157], rather than how it navigates. The method in [158] trains several VPR models with inputs at different pre-selected scales, motivated by the discovery that mammals primarily navigate by storing spatial information at multiple scales [159]. The authors of [160] critique the fixed scales approach and propose an adaptive alternative. BioSLAM [161] addresses the problem of life-long SLAM [162] by emulating the short and long-term memory mechanisms of human brains [163, 164]. Human memory is also taken as inspiration in [165] which proposes a neurologically inspired approach based on Hierarchical Temporal Memory [166] for sequential Visual Place Recognition. Spiking Neural Networks (SNNs) [167] more closely emulate the inner workings of biological brains than conventional neural networks. Much like BNNs, SNNs benefit from incredible power efficiency when deployed with specific hardware [168, 169]. The application of SNNs for Visual Place Recognition is a relatively new field, but initial attempts show promising results. In [170] the authors introduce the first SNN-based VPR solution, proposing a lightweight network trained in an unsupervised manner. Deploying multiple, location-specific SNNs as a scalable solution to VPR has been proposed in [171].

2.3 Multi-Technique Visual Place Recognition

Despite significant advancements in VPR performance, it remains a difficult task due to how drastically a place's appearance can change from visit to visit. Moreover, the nature of these changes is also highly variable [172, 173, 18, 19]. As so far discussed, many VPR techniques have been proposed and recent work [174] shows how different approaches present different strengths and weaknesses regarding their robustness to visual challenges [124, 175, 176]. Combining multiple VPR techniques with the goal of offsetting the shortcomings of one technique with the strengths of another is becoming a popular approach for improving matching performance. The following discussion separates the topic into two categories which can be visualized in Fig. 2.3: technique fusion and technique switching.

2.3.1 Technique Fusion

In fusion, the output of multiple VPR techniques is combined into a single place matching structure, either at the descriptor or similarity matrix level. Note that this is different from sensor fusion [177, 178, 179] where the inputs of multiple sensors, such as cameras and LIDAR [180], are merged together. Instead, in technique fusion, the only sensor remains the camera (vision) and it is the technique's processing that creates variability in visual robustness.

Multi-Process Fusion (MPF) [181] uses a Hidden Markov Model [97] and a modified version of the Viterbi algorithm [182] to combine the similarity matrices of four VPR techniques over a small sequence of images. In [106], a hierarchical fusion approach is introduced, also acting at the similarity matrix level. In each of the three tiers of the hierarchy, only the top candidates from each technique are passed on to the next tier, eventually reaching a matching consensus. Similar to MPF, ROMS [183] also presents a sequential fusion approach, but it combines feature vectors from multiple image processing methods rather than acting on the similarity matrix. Directly aggregating feature vectors from different methods has been further explored in [184], where the authors propose an efficient framework for embedding extra information.

Despite increased performance in most scenarios, fusion approaches come with two main shortcomings. Firstly, fusion substantially increases computational effort. Not only do multiple VPR techniques need to be processed, but the aggregation stage itself can become resource intensive. Secondly, a performance increase is not guaranteed. When there is a large disparity of performance between the different techniques, it is possible that the fused output actually performs worse than the best individual method.

2.3.2 Technique Switching

Technique switching also aims to offset weaknesses of one technique with strengths of another from a given method ensemble. However, rather than combining outputs into a single representation, the switching process selects the output of the most suitable technique for a given query image. Designing an effective and practical selection mechanism is the main challenge to the switching approach, and few solutions have been proposed thus far.

SwitchHit [185] starts by running a set of four VPR techniques for each query image. Then, by using prior environmental and technique complementarity knowledge, it uses bayesian inference [186] to select the best technique for the query. As an evolution of SwitchHit, SwitchFuse [187] uses this scheme to select a sub-set of techniques to be fused, achieving a mixture of switching and fusion. In [171], the authors propose an ensemble of SNNs, with subsets being trained on different regions of the traversal. At navigation time, the non-informative (hyperactive) networks are identified and only the region-relevant agents perform VPR.

The major advantage of switching over fusion is that it avoids the situation where one poorly performing technique brings down the overall performance. However, it introduces a difficult selection stage where the method must identify the most suitable technique for the given query, often relying on prior information of the deployment environment for this step. Moreover, if all the techniques are required to run prior to a selection, the same computational intensity problem from fusion remains.

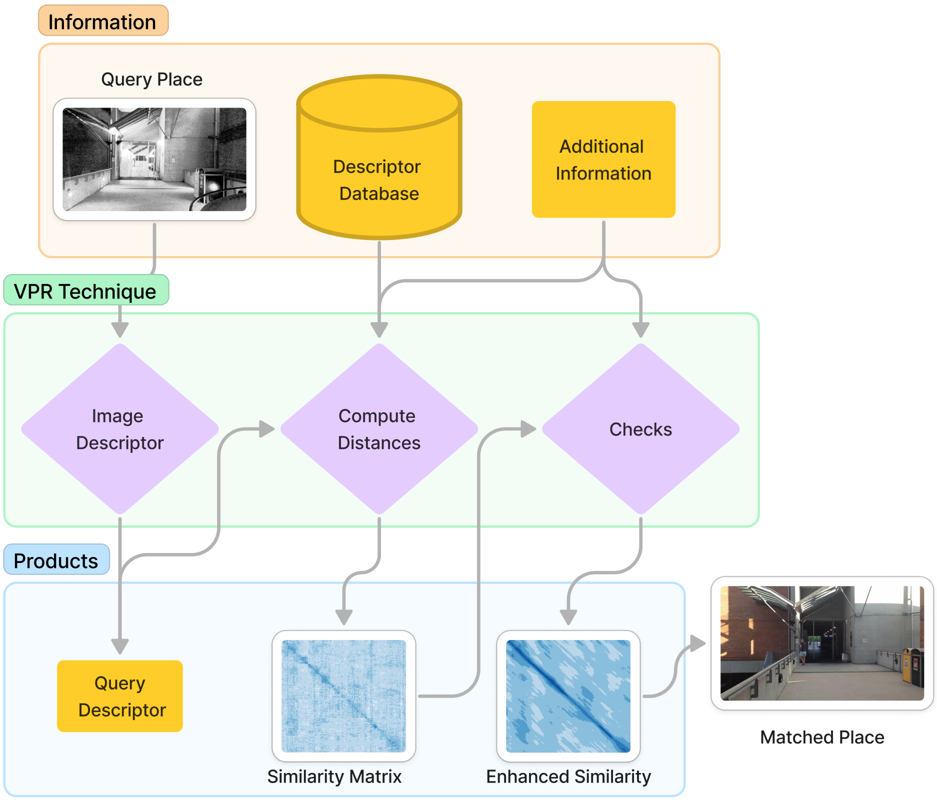


Fig. 2.2. In an effort to increase VPR performance, several methods include extra steps and/or information into their matching pipelines. Some techniques infuse information when performing the similarity computation itself while others enhance the similarity matrix.

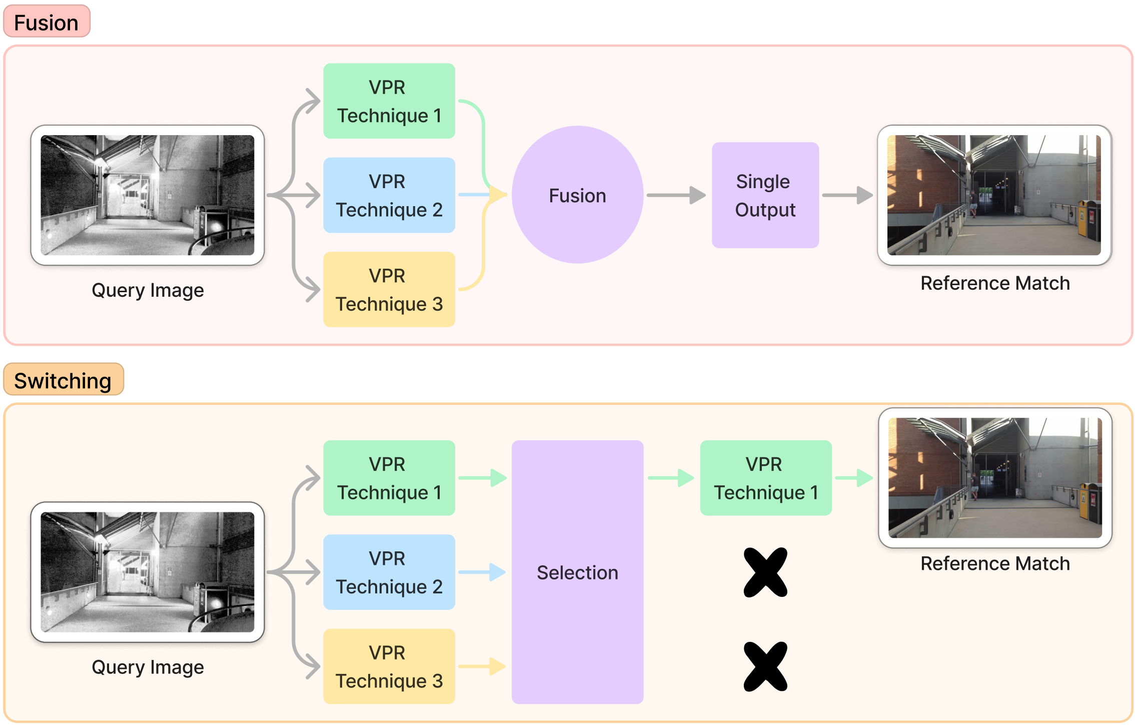


Fig. 2.3. In technique fusion, the outputs of several VPR methods are combined into one structured from which a place match result is derived. In switching, one VPR method is selected from the ensemble and the output of the remaining techniques is discarded.

Chapter 3

Fly-Inspired Voting Units for Efficient VPR[1]

The introduction of deep learning approaches, in the form of CNNs or more recently Transformers, has undoubtedly improved the state-of-the-art VPR performance. However, in chasing accuracy, these networks have become deeper and more complex, resulting in high memory and computational power needed to perform real-time matching [129]. As mobile robotics are often restricted to low-end hardware, mainly due to size, budget, or battery limitations, such models are unsuitable for several autonomous applications [21].

This chapter adds to the efficient VPR literature by presenting a bio-inspired, multimodel VPR pipeline, with two novel algorithms being proposed. The first, dubbed DrosoNet, is a lightweight neural network architecture inspired by the brain of the fruit fly processing of odors. It employs 8-bit quantization [138] to achieve a minimal memory

[1] This work is published in IEEE Robotic and Automation Letters, vol. 7, no. 2, pp. 2527-2534, Apr. 2022, and was presented at IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, USA, 2022. DOI: 10.1109/LRA.2022.3140827.

size and inference time. The second algorithm builds upon DrosoNet, exploiting its compact and randomness properties by using multiple DrosoNets in an ensemble, combining their outputs with a voting mechanism tailored for VPR. With the suggested hyperparameters, the system features a model size of 6 MBs and inference time of 18 ms, with no specialized hardware required.

3.1 The Voting Pipeline

This section details the two components of the proposed VPR pipeline: DrosoNet and the voting mechanism built on top of it. DrosoNet starts by computing a binary image representation of a given image. It is then trained as a classifier, where each place is a different class, learning to recognize these small image tags and associating them with their respective place. The voting mechanism exploits the stochastic nature of the DrosoNet initialization and training processes, making use of several DrosoNet individuals to perform more accurate and consistent VPR while remaining compact relative to state-of-the-art approaches.

3.1.1 DrosoNet

DrosoNet is a bio-inspired model that draws inspiration from the fruit fly's brain circuits. The brain of these small insects is extremely efficient at recognizing different odors, especially when considering its size. While the VPR problem deals with visual information, the algorithm uses a simplified version of the information processing that the fruit fly's brain uses for odor recognition.

A computer science focused implementation of the odor recognizing process was proposed in [157], presenting three main steps. Firstly, a sort of normalization occurs, centering the mean of the activation rates of the flies' neurons for all odors. Secondly, around 10% of the neurons that respond to an odor are evaluated and their activation rate is summed up. Finally, 5% of the summed up values are used to create a binary representation of the given odor - this compact representation is then used to compare and recognize odors.

The proposed DrosoNet algorithm makes use of the fly's schema to encode a compact image representation which is then fed to a fully connected layer for classification. The process aims at a low memory footprint by utilizing small image input sizes and other hyperparameters are chosen according to empirical data. Fig. 3.1 shows the operations that occur in the DrosoNet algorithm, displaying the dimensions of each matrix. The image is stretched into a row vector of size 1×2048. It is then multiplied by the matrix H, which is binary and sparse, with 10% of the elements of each column randomly set to 1 and the remaining to 0 on DrosoNet's initialization. This results in 10% random pixels of the input image being sampled when calculating the activation values, stored in F. The number of columns in H corresponds to the number of activations used - this value is set to 192. The top 50% higher values in F are then set to 1 while the bottom 50% are set to 0, resulting in a binary representation for the input image, vector O. O is then fed into a fully connected layer where the learning process takes place. The fully connected layer works as a classifier to predict the current place from the vector O, hence including exactly one neuron per map's location. Note that matrix W in the fully connected layer is the only element in DrosoNet which undertakes any sort of learning process. The highest value among the n elements of the output vector is regarded as the matching location. Finally, after the model is trained, the parameters' precision is reduced to 8-bit using dynamic quantization [138], further reducing the memory size of DrosoNet.

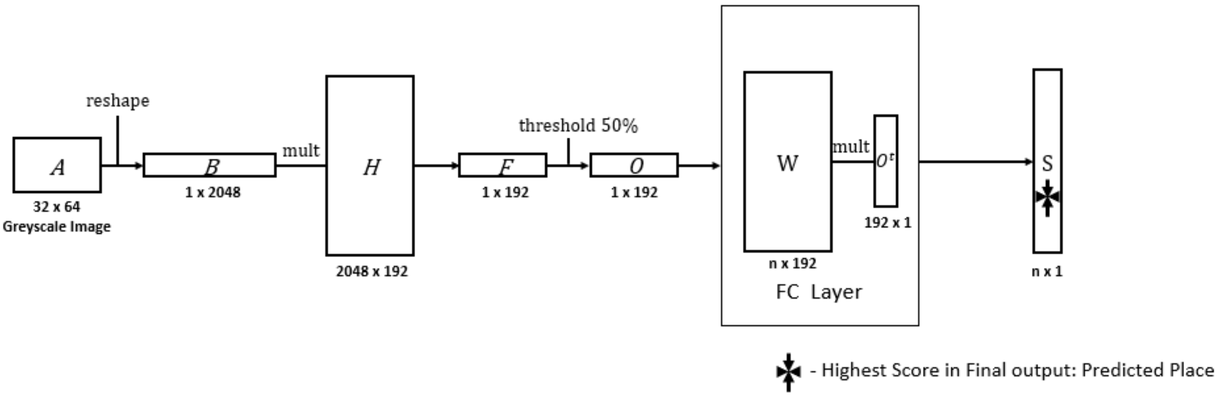


Fig. 3.1. DrosoNet implementation diagram. This process is repeated for each input image (each image corresponds to a place, n places). A is the input image; B is the input image reshaped to a row vector; H is a sparse, binary matrix where each column has only 10% of its values set to 1, the remaining to 0; F contains the 192 activation values; O is a binary image representation resulting from the threshold of the values of F; W is the weight matrix of the fully connected classifier; S is the final output of the DrosoNet, where each value corresponds to a place score. After training takes place, DrosoNet is dynamically quantized to 8-bit precision, further reducing its memory usage.

3.1.2 Voting Mechanism

The voting mechanism, illustrated in the diagram of Fig. 3.2, combines the output of several DrosoNets to perform more effective and consistent VPR. In practice, the random initialization of DrosoNet's H matrix, as well as the stochastic nature of training a fully connected layer, means that one particular DrosoNet might have an erroneous outcome for a given place while most other DrosoNets actually output an acceptable prediction. The voting exploits this observation and does not rely on any single DrosoNet to perform place matching. Instead, it selects the prediction that most DrosoNets agree on, following the ruling detailed in the remaining of this section.

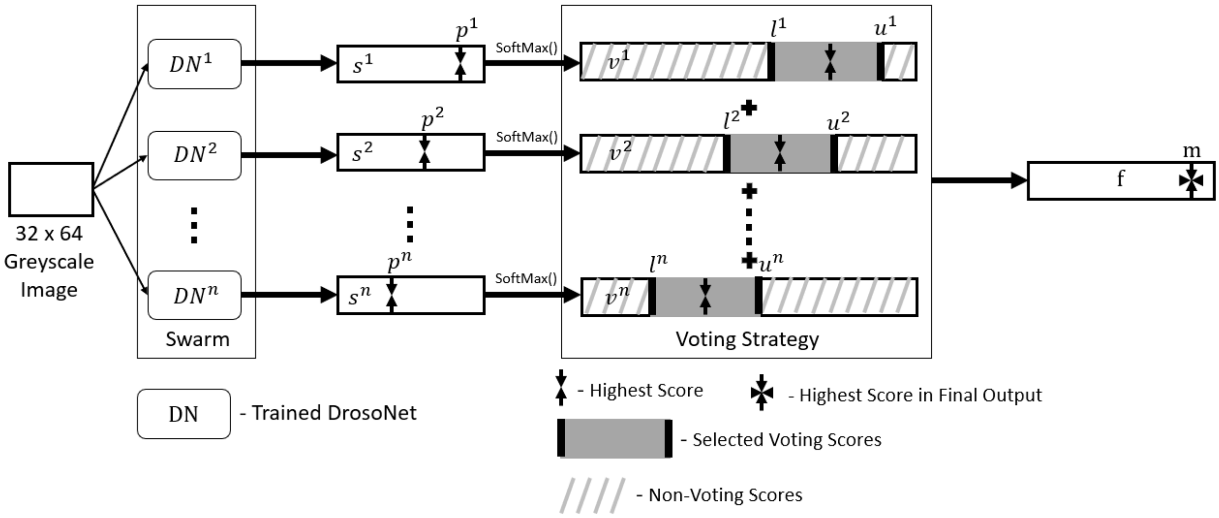


Fig. 3.2. Voting mechanism diagram, displaying the combination of several DrosoNets' outputs using the proposed voting method. Each DrosoNet is fed the same query image in the correct format. The output of each DrosoNet, a score vector s, passes through a softmax function before voting takes place, producing vector v. The voting mechanism takes each vector v and sets all elements to 0 except for the elements within a window around the largest score p in each respective vector. Finally, all vectors are summed element wise, producing the output vector f. The reference place associated with the largest element in f is deemed the correct match.

The pipeline assumes a pre-existing collection of n trained DrosoNets, each independently receiving the current query image as input and outputting their corresponding score vectors. For each DrosoNet, the voting strategy takes into account the largest element in its output (its predicted reference place) as well as all the scores within a range around that index. The selection of a score for a single DrosoNet, d, can be represented as

$$v_i^d = \begin{cases} s_i^d & \text{if } l^d \leq i \leq u^d \\ 0 & \text{else} \end{cases} \tag{3.1}$$

where s^d is the complete vector score given by the d^{ith} DrosoNet after being normalized by a soft max function pass, s_i^d denotes the i^{th} score in s^d, v_i^d is the i^{th} score that is either copied from s^d or set to 0 and stored in vector v^d. l^d and u^d denote for the lower and upper index bounds of the scores to be selected around the highest score d, for the d^{ith} DrosoNet, and are defined as

$$p^d = argmax(s^d) \tag{3.2}$$

$$l^d = max(0, p^d - r) \tag{3.3}$$

$$u^d = min(len(s^d) - 1, p^d + r) \tag{3.4}$$

where p^d is the index of the highest score (hence the place predicted by the d^{ith} DrosoNet), r is a hyperparameter for the chosen range of selection and $len(s)$ is the length of the score vector s^d (also corresponding to the number of places and hence is constant for all n DrosoNets). The max and min functions are used to avoid negative and out-of-bound indexing.

Using the above definitions, the vector v^d is constructed for each of the n DrosoNets in the ensemble, where each element is set to either 0 or the corresponding score in s^d. A single vector score f is obtained by summing element-wise over all n vectors v^d

$$f = \sum_{d=1}^{n} v^d \tag{3.5}$$

Finally, the index m of the highest score in f is selected as the matching place for the input image:

$$m = argmax(f) \tag{3.6}$$

The assumption of the above voting algorithm is that enough DrosoNets correctly match a query as to ignore the noise produced by less competent DrosoNets. Since wrong DrosoNets tend to disagree on their respective erroneous matches, and correct DrosoNets, by definition, agree on the correct match, the pipeline can achieve consensus on the correct reference place even without a majority of the DrosoNets being correct. However, it's always possible that too many or even all the DrosoNets are incorrect, and thus voting would not be able to generate a correct output.

3.2 Experiments

The proposed models are evaluated against several other VPR algorithms: CALC [143], HOG [45], CoHOG [46], FlyNet [23], HybridNet [68], ASMOSNet [68], NetVLAD [15] and GIST [40]. Different datasets are employed to assess the models' capacity to deal with different VPR challenges: moderate to extreme appearance changes and small to moderate point-of-view (POV) variations. Memory usage and time required to process a single image are presented to assess computational efficiency. The remaining of this section provides details on models' settings, datasets and evaluation metrics.

3.2.1 Model Settings

FlyNet is tested with the settings made available by the authors in [23]. The implementations provided in [16] are used for the HOG, CoHOG, CALC, HybridNet, ASMOSNet and NetVLAD. GIST is evaluated with the implementation from [188].

For the standalone DrosoNet and for DrosoNet in conjunction with the voting mechanism, a series of ablation studies guided the choices for the number of activations for DrosoNet and the number of models to be used in conjunction with the voting system. Fig. 3.3 shows the results of these studies in the Corvin (Fig. 3.3a), Nordland Fall (Fig. 3.3b) and Oxford (Fig. 3.3c) datasets. Rather than optimizing for a single dataset, the hyperparameters that achieve best average performance across all datasets are selected. By this criteria, an ensemble size of 32 DrosoNet models is chosen, each with a hidden size of 192, indicated by the blue arrows in the figures. Also by experimentation, the voting range r value is set to equal 50% of the total number of places in the dataset. Each DrosoNet is trained for 200 epochs, using cross-entropy loss with a fixed learning rate of 0.001.

3.2.2 Datasets

A subset of the datasets presented in Section 2.1.6 is used during testing, with the following ground-truth approach. For Fall and Winter, a margin error of 1 frame is allowed. Thus, for query image q and ground-truth image t, references $t-1$ to $t+1$ are considered correct matches. For Gardens Point Day-Right, the error margin is 2 frames. For Lagout 15, the direct ground-truth provided by the dataset is used. In Oxford RobotCar, 10 frames of error are allowed, as per [181, 103]. Finally, for Corvin 30, a margin of error of 20 frames is used.

3.2.3 Evaluation Metrics

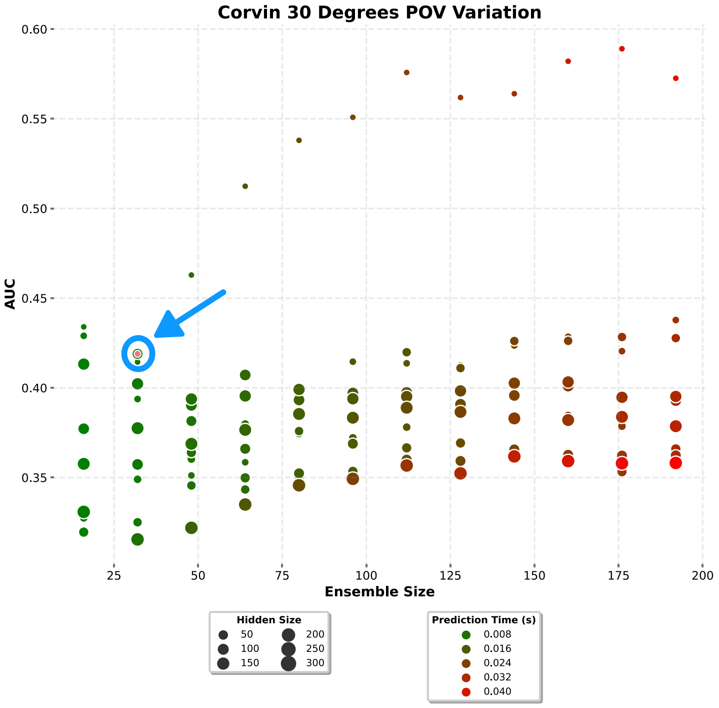
Visual place recognition performance is evaluated with PR curves and their respective AUC. Memory sizes and inference times are used to compare computational efficiency.

3.3 Analysing DrosoNet & Voting Results

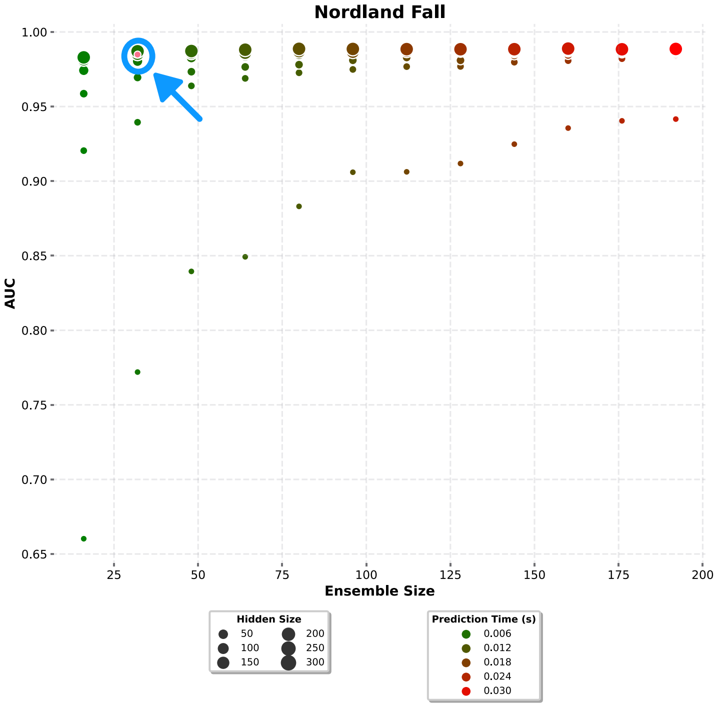
This section discusses the results obtained by experiments in both fronts: VPR performance and computational efficiency.

3.3.1 VPR Performance

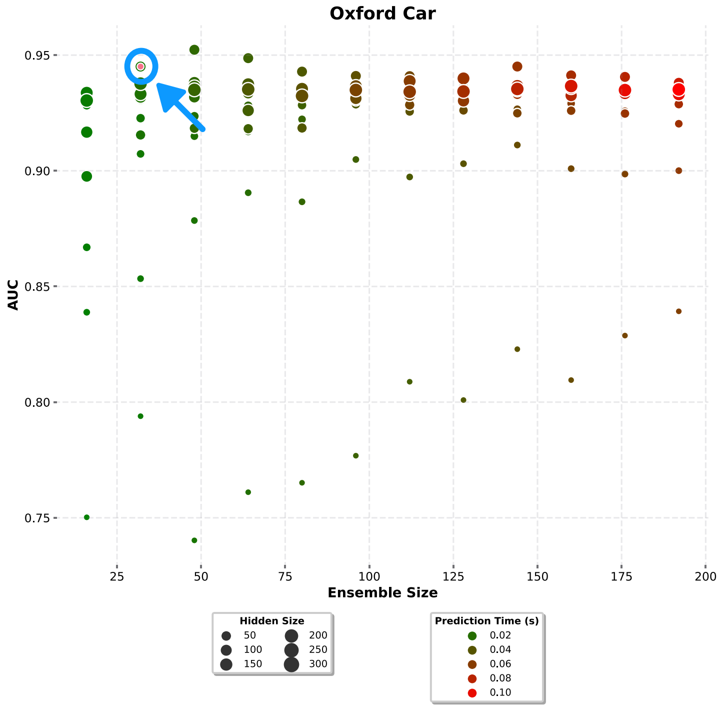
Performance results are organized as follows. The PR curves graphs in Fig. 3.4 present VPR performance for DrosoNet, the voting system and a subsection of the tested methods


Corvin 30 Degrees POV Variation
AUC
0.60
0.55
0.50
0.45
0.40
0.35
25
50
75
100
125
150
175
200
Ensemble Size
Hidden Size
50
100
150
200
250
300
Prediction Time (s)
0.008
0.016
0.024
0.032
0.040

(a)


Nordland Fall
AUC
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
25
50
75
100
125
150
175
200
Ensemble Size
Hidden Size
50
100
150
200
250
300
Prediction Time (s)
0.006
0.012
0.018
0.024
0.030

(b)



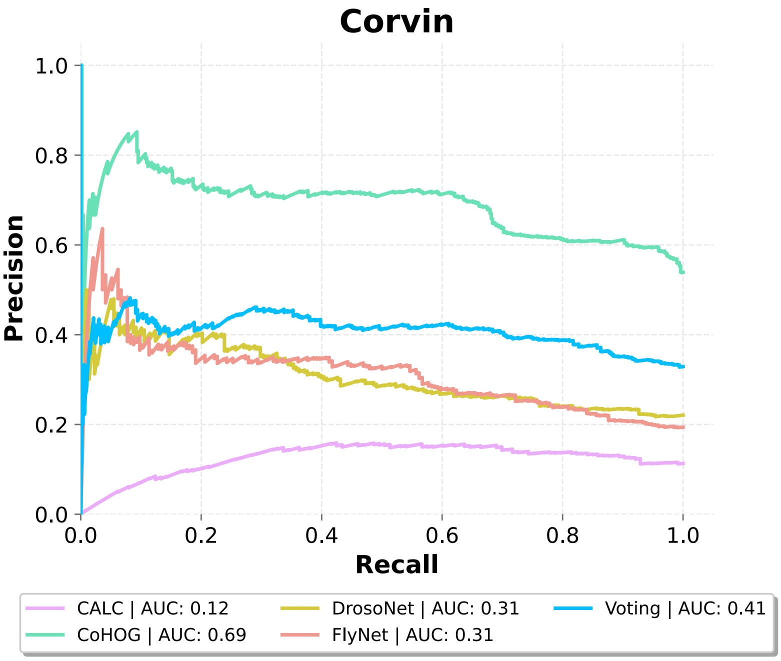
(c)

Fig. 3.3. Ablation studies to determine best hidden and ensemble size on the Corvin 30 (a), Nordland Fall (b) and Oxford Car (c) datasets

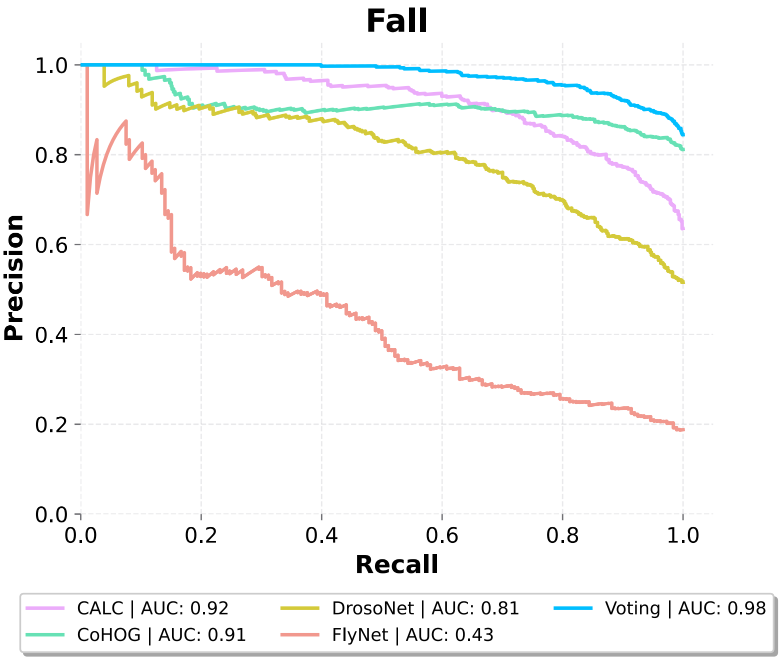
which claim to be lightweight and efficient: CoHOG, CALC, and FlyNet. For comparing VPR performance across all tested techniques and datasets, Fig. 3.5 displays a bar graph with all AUC results, including performance for NetVLAD, HybridNet, AMOSNet, GIST and HOG in addition to the aforementioned models.

Corvin 30 Degrees POV And Scale Variation

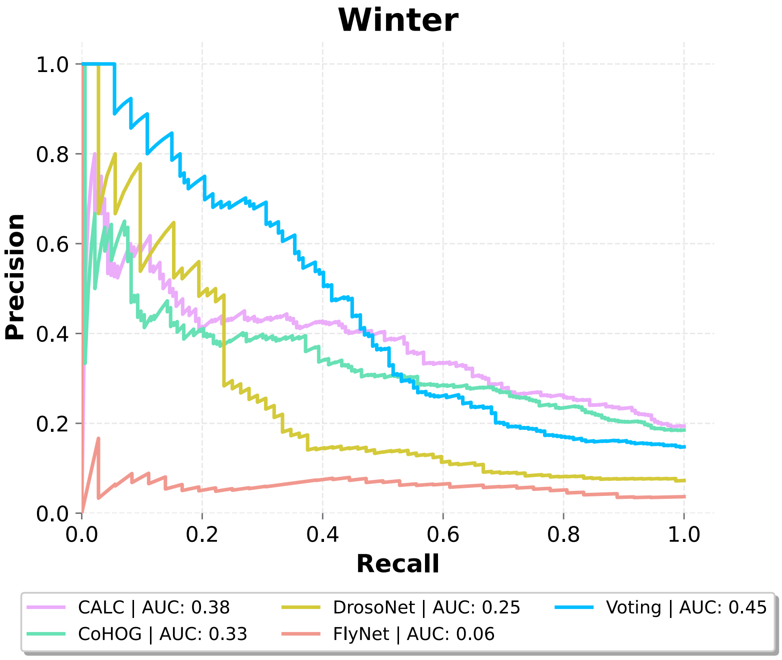
Fig. 3.4a shows how the voting mechanism is only outperformed by CoHOG amongst the lightweight methods. The Corvin dataset presents extreme point-of-view variations with 6 DOF, making it especially challenging for non-local feature techniques as is the case of DrosoNet and consequentially the voting system. As expected, all the top performers are


Corvin
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.12
CoHOG | AUC: 0.69
DrosoNet | AUC: 0.31
FlyNet | AUC: 0.31
Voting | AUC: 0.41

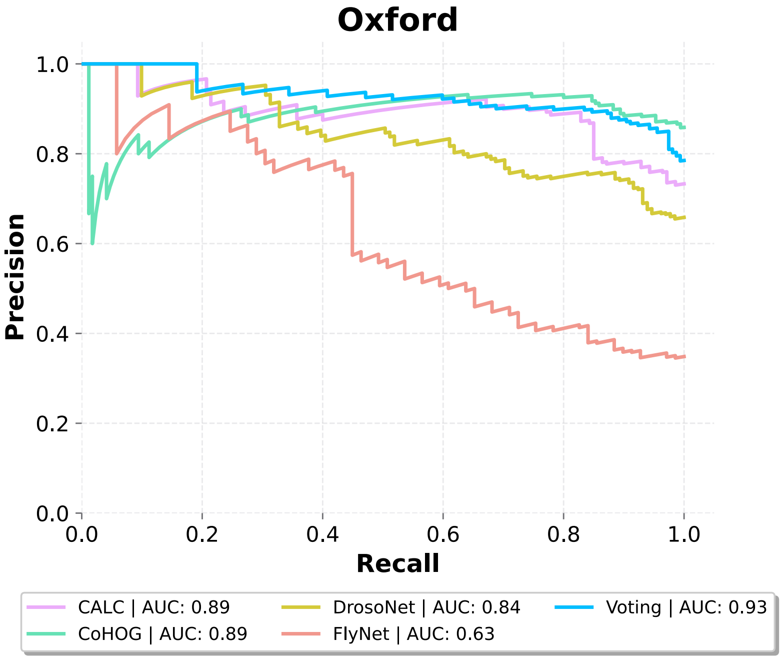
(a)


Fall
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.92
CoHOG | AUC: 0.91
DrosoNet | AUC: 0.81
FlyNet | AUC: 0.43
Voting | AUC: 0.98

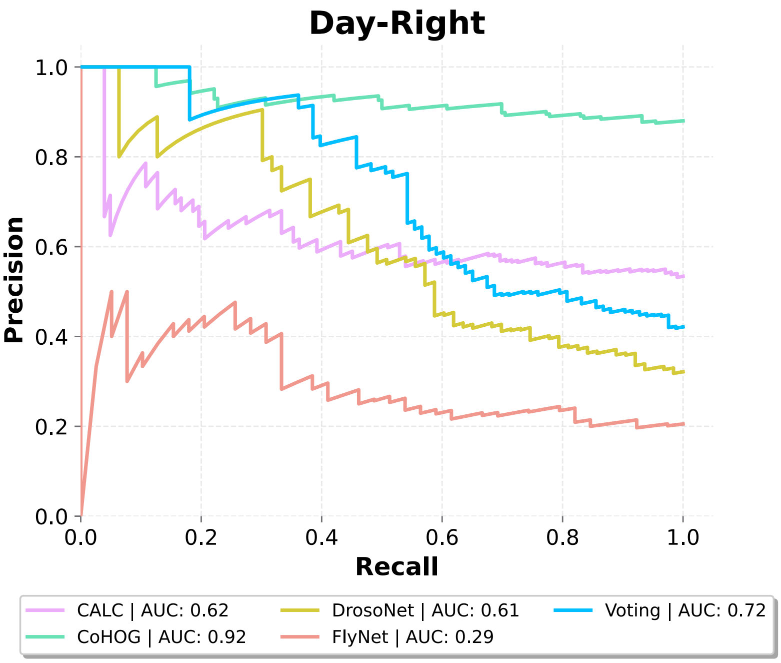
(b)


Winter
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.38
CoHOG | AUC: 0.33
DrosoNet | AUC: 0.25
FlyNet | AUC: 0.06
Voting | AUC: 0.45

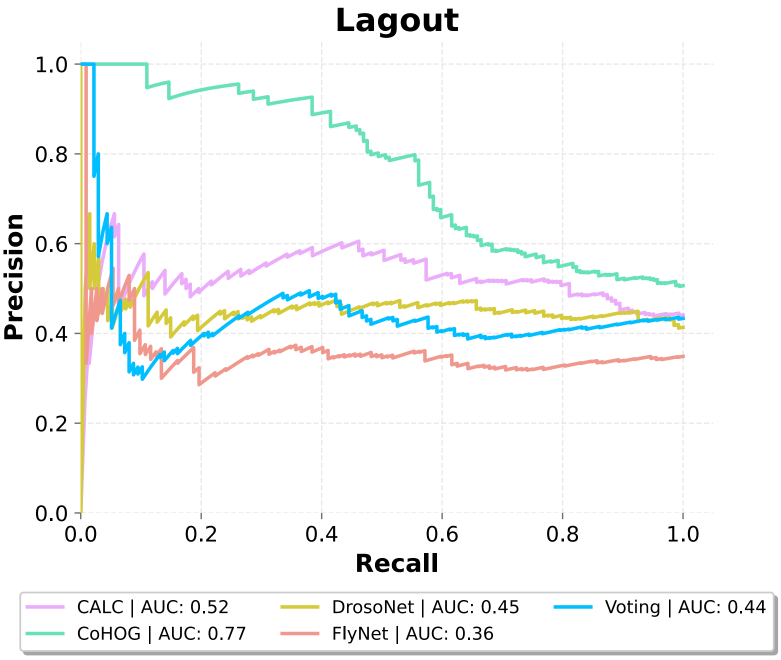
(c)


Oxford
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.89
CoHOG | AUC: 0.89
DrosoNet | AUC: 0.84
FlyNet | AUC: 0.63
Voting | AUC: 0.93

(d)



(e)



(f)

Fig. 3.4. Precision-recall curves and respective AUCs for lightweight VPR methods

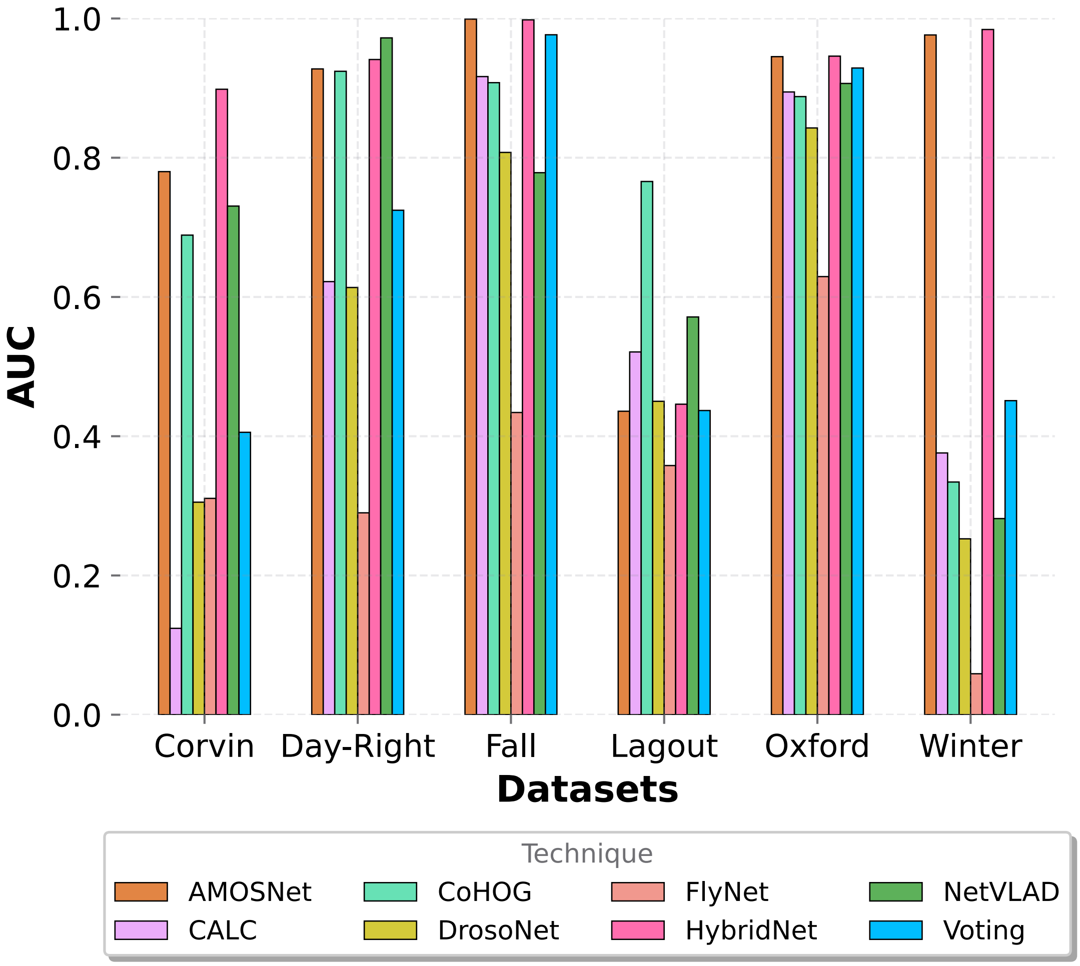


Fig. 3.5. Precision-recall AUC values for every tested model across all benchmark datasets.

region-based techniques: AMOSNet, HybridNet, CoHOG, and NetVLAD. Nevertheless, Fig. 3.5 shows that voting outperforms CALC, FlyNet, GIST, and HOG. Furthermore, when tuned specifically for this dataset, the voting ensemble is able to achieve better AUC values, as seen in Fig. 3.3a.

Nordland Fall

From the subset of models shown in Fig. 3.4b, the voting pipeline outperforms every other method. When considering all techniques in Fig. 3.5, it is only outperformed by more computational demanding algorithms such as HybridNet and AMOSNet, and only by a small difference of 0.02 AUC value.

Nordland Winter

For the more challenging Nordland Winter dataset, DrosoNet voting is again the top performer out of the lightweight methods, as seen in Fig. 3.4c. Once more, the only approaches which outperform voting are HybridNet and AMOSNet, albeit with a much larger AUC advantage.

Oxford RobotCar

The voting ensemble similarly outperforms other lightweight methods with the illumination changes of the Oxford dataset, as seen in Fig. 3.4d. The only techniques which perform better, by a small AUC margin, are HybridNet and AMOSNet (3.5).

Gardens Point

Fig. 3.4e shows that CoHOG is the best performer out of the considered lightweight algorithms, with an AUC value of 0.92. The voting ensemble comes in second at 0.72. As this dataset presents a lateral viewpoint shift, region-based approaches achieve the best performance, as observed in Fig. 3.5.

Lagout 15 Degrees POV Variation

Lagout is the only dataset where the proposed voting mechanism is outperformed by CALC, as shown in Fig. 3.4f. However, this dataset is challenging for every technique, including HybridNet and AMOSNet 3.5.

3.3.2 Computational Resources

This section covers computational efficiency results by analyzing inference times with respective frames-per-second rates and required memory allocation. Table 3.1 shows the aforementioned metrics, obtained while running the different models on the Corvin dataset with a Ryzen 7 4000 Series mobile processor. The choice to utilize a CPU rather than a GPU is motivated by the fact that CPUs are available even in the lowest-end of hardware, while GPUs are usually not present in extremely constrained mobile robots.

TABLE 3.1: PREDICTION TIMES AND MEMORY USAGE COMPARISON.

Model	Prediction time (ms)	FPS	Size (MBs)
HybridNet	1143.92	0.87	61.44
AMOSNet	1138.97	0.88	61.44
CoHOG	3627.18	0.28	123.01
CALC	73.62	13.58	4.26
GIST	225.04	4.44	4.53
HOG	208.71	4.79	142.88
NetVLAD	1435.11	0.70	16.38
FlyNet	1.00	1000	0.26
DrosoNet	1.00	1000	0.19
Voting	18.43	54.27	6.19

DrosoNet outperforms FlyNet in most datasets due to the increased size of its fully connected layer. Despite the increase in parameters, 8-bit quantization allows DrosoNet to remain more compact than FlyNet, as seen in Table 3.1. The multi-DrosoNet voting mechanism is the fastest model apart from FlyNet and DrosoNet. It shows a prediction time of 18ms, four times less than the next fastest model CALC, achieving a 54 FPS rate. The methods with overall best AUC results were HybridNet and AMOSNet, both having significantly larger model sizes and inferences times, and consequentially smaller FPS rates. For the Oxford RobotCar and Nordland Fall datasets, the voting pipeline achieves almost identical VPR performance for a fraction of the computational cost. Although CoHOG presents better VPR performance on 6 DOF viewpoint changes when compared to voting, it takes 200 times the processing time. For appearance change datasets, voting performs better and faster than CoHOG. NetVLAD is not only slower when compared to the voting ensemble but also performs worse when dealing with appearance changes. However, it does achieve better results with viewpoint variations. For CALC, HOG, and GIST, voting is faster and it performs better VPR across all tested datasets with the sole exception of Lagout 15.

3.4 The Limitations of DrosoNet

This chapter introduced a lightweight VPR pipeline composed of two novel techniques and suitable for the most hardware constrained of robotic platforms. The first contribution is DrosoNet, an extremely compact algorithm inspired by the brain of the fruit fly. Relative to its size and speed, it obtains respectable benchmark VPR results, especially when dealing with moderate appearance changes. However, most systems require more robust VPR than what DrosoNet is able to offer. Thus, a voting scheme is also proposed, using multiple of these small networks to achieve competitive VPR performance while remaining compact relatively to CNN based algorithms and even other lightweight methods. When comparing the trade-off between size and performance, the DrosoNet based voting model stands out as a compact VPR algorithm with strong performance, suitable for robotic platforms equipped with low-end hardware.

It is important to note the relatively small size of the datasets used for testing - the largest datasets used in the experiments comprise solely of 1000 images. This brings important considerations to the presented results, both in terms of computational efficiency and VPR performance. Firstly, regarding efficiency, DrosoNet's W matrix grows by a factor of $192 \times n$, where n is the number of images in the dataset. Thus, an increase in places leads to increased computation and memory requirements. While this is also the case for other descriptor-based algorithms, as additional places require additional representations in the map, efficiency optimizations for these approaches have been more thoroughly explored. Secondly, with respect to VPR performance, the small map size used for testing also bears considerations. As the map grows, the small image representation produced by DrosoNet (vector O) may begin struggling to sufficiently distinguish between different places. While such a problem could be addressed by increasing the size of O, this would further lead to increased computational demands. Given these points, future DrosoNet research could focus on testing the system with much larger map sizes, as new datasets become available [8].

Moreover, despite the strong relative performance showcased by DrosoNet and voting, both algorithms still substantially lack behind the more computational intensive methods. While this is, to some degree, expected, several VPR applications require stronger matching performance to be functional. One straightforward avenue for improving the performance of a multi-DrosoNet system is to increase differentiation across DrosoNets. Indeed, the work presented in this chapter fully relied on the stochastic behaviors of DrosoNet's initialization (H matrix) and training process (FC layer). While this simple approach is enough to justify the merging of multiple models, as proven by the performance increase of the voting pipeline over a single DrosoNet, a more intentional differentiation method is desirable. The next chapter explores the training of DrosoNets on different regions of the input images with the goal of increase model differentiation and, consequentially, improve absolute VPR performance.

Chapter 4

Aggregating Bio-Inspired Image Region Classifiers for Lightweight VPR[1]

Chapter 3 introduced DrosoNet, a highly compact and fast bio-inspired classifier which achieves impressive VPR results relatively to its computational cost. To further improve VPR performance, a voting system combining multiple DrosoNets was also proposed, exploiting the inherent randomness of DrosoNet's initialization and training processes. The overall system outperforms other lightweight VPR techniques while retaining inference times orders of magnitude lower. Nevertheless, despite its efficiency, the absolute VPR performance of the DrosoNet voting mechanism remains unsuitable for applications where more reliable VPR is required.

This chapter improves upon the VPR performance of a multi-DrosoNet setup by increasing differentiation between different voting units. Rather than solely relying on DrosoNet's stochastic behaviors, a novel training approach is explored where different

[1] This work is published in IEEE Robotic and Automation Letters, vol. 9, no. 4, pp. 3315-3322, Feb. 2022, and accepted to be presented at the IEEE International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, UAE, 2024. DOI: 10.1109/LRA.2024.3367275.

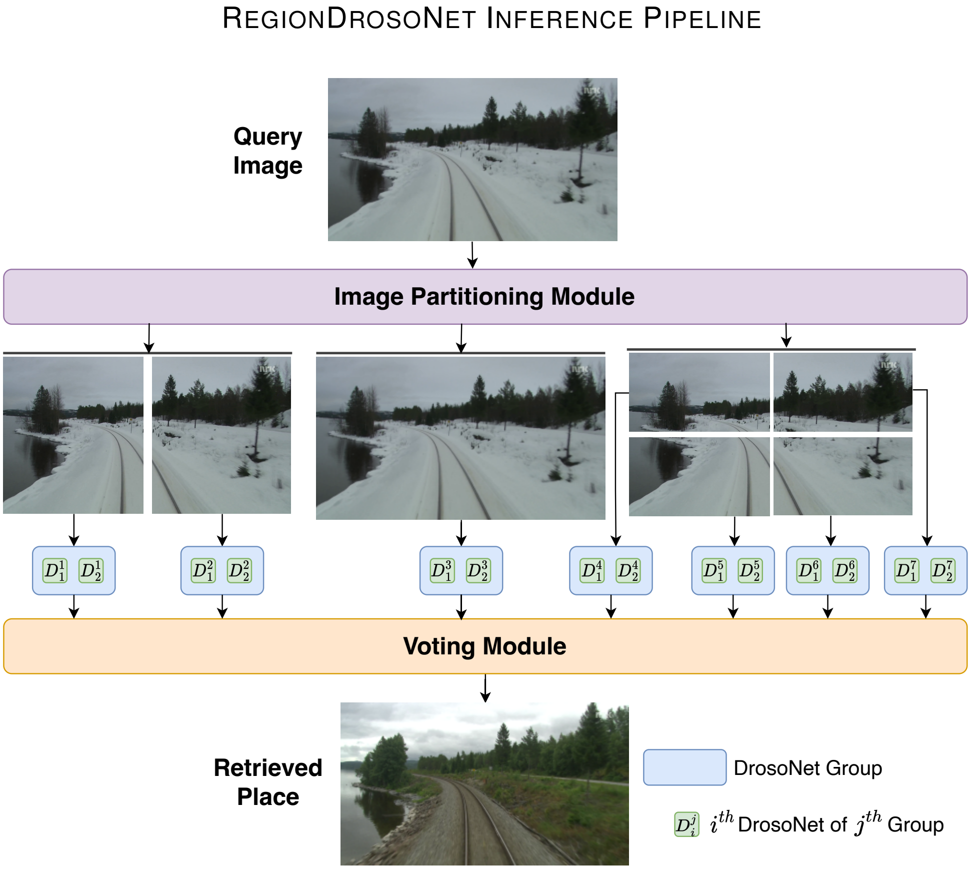


Fig. 4.1. The query image is divided into multiple heterogeneous regions. Each region is then fed as input into a specialized DrosoNet group which was trained only on that particular region from the training set images. Finally, the output of each group is aggregated in the voting module and a reference place is retrieved.

groups of DrosoNets are trained on different regions of training set images. Moreover, an improved voting system is also introduced. It considers multiple top place candidates from each DrosoNet, allowing the system to converge on the most generally agreed upon reference place and thus mitigating the individual DrosoNets failing to realize a correct match.

The proposed VPR system, dubbed RegionDrosoNet, outperforms other lightweight VPR techniques when dealing with both appearance changes and viewpoint variations. Moreover, it competes with computationally expensive methods on some benchmark datasets at a small fraction of their online inference time.

4.1 Differentiating DrosoNets

In the broader field of machine learning, the practice of combining multiple models is common and a widely accepted avenue for improved performance [189, 190]. For a combination of baseline classifiers to perform better than any individual one, it is necessary that their outputs, given the same input, are different [191]. To conform to this requirement, two approaches are viable, the most straightforward of which is to construct a set of diverse baseline models. The second and less obvious procedure is to use a set of multiple of the same baseline model, but ensuring that sufficient variation between models is introduced.

The usage of multiple DrosoNets as voting units proposed in Chapter 3 follows the second approach - all models in the set are DrosoNets, and thus share the same network structure and algorithmic steps. Solely two stochastic processes contribute to all the differences present between DrosoNets and their respective outputs. The first is the random initialization of the H matrix, which is untrained and thus completely fixed after its construction. The second is the initialization and training process of the weight matrix W, as it is common in neural networks for these steps to result in different learned parameters [192], possibly leading to different outputs.

This section details a novel multi-DrosoNet localization pipeline which achieves increased VPR performance across various visual challenges, while maintaining a low computational profile. The core approach, dubbed RegionDrosoNet, relies on introducing additional model differentiation by training specialized DrosoNet groups on different regions of the train set images. At inference time, as can be observed in Fig. 4.1, each partition of the query image is served as input to its respective group and each DrosoNet produces its reference place confidences. The training and inference processes are tailored to DrosoNet, taking full advantage of its peculiarities: it's extremely fast and compact, allowing for the use of multiple units; it's a neural network classifier, not requiring storage of an image descriptor for every map location as a reference for image matching; DrosoNet groups trained on different image regions benefit from additional model differentiation induced by different training data, while units within each group continue benefiting from DrosoNet's inherent differentiation.

4.1.1 Revising DrosoNet

Revising DrosoNet, it is a compact and fast neural network image classifier where each of the environment's total N places is a different class. The same configuration as in Chapter 3 is used, which can be seen in Fig. 4.2. A 64×32 grayscale image is first flattened into a one-dimensional vector, denoted as $\hat{i}$, followed by a matrix multiplication with H, producing vector F. H is a binary, sparse, and randomly initialized matrix, where 10% of each column's elements are initialized to 1 and the remaining to 0. Matrix H is untrained, and thus the random initial values are fixed from its construction. F is then binarized by the function th, where the top 50% of values are set to 1 and the bottom 50% are set to 0, resulting in the binary vector O. W is a fully connected layer which learns to map O to one of the N classes, i.e. reference places. The final output vector s stores the score distribution for each reference place, and the DrosoNet's prediction is the index of the largest score in s.

While DrosoNet is a fast algorithm, its standalone VPR performance is too unreliable.

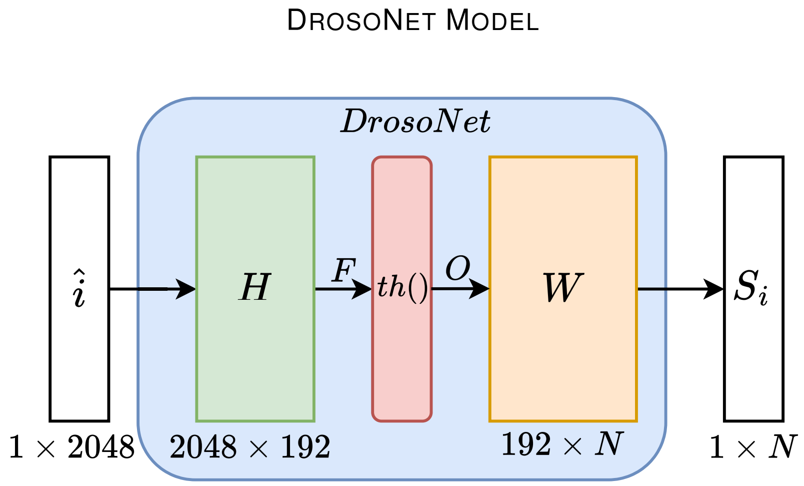


Fig. 4.2. DrosoNet model diagram.

Moreover, due to the randomness of its H matrix initialization and supervised training, different DrosoNets exhibit high variance in their VPR performance. Combining multiple DrosoNets was hence proposed in Chapter 3 as an avenue to improve overall VPR performance, relying only on the native stochastic behavior of the models for differentiation.

This chapter aims to increase DrosoNet differentiation by training distinct models on different partitions of the original images, producing region specialized DrosoNets. Moreover, by training multiple DrosoNets on each image region, the system continues taking advantage of the randomness associated with the initialization and training processes.

4.1.2 Image Partitioning

The image partitioning module receives as inputs an image i and grid dimensions (r, c), where r represents the number of rows and c the number of columns, outputting rc image regions. As detailed, DrosoNet operates with grayscale images with a resolution of 64×32, thus the produced regions are converted to grayscale and resized to the correct

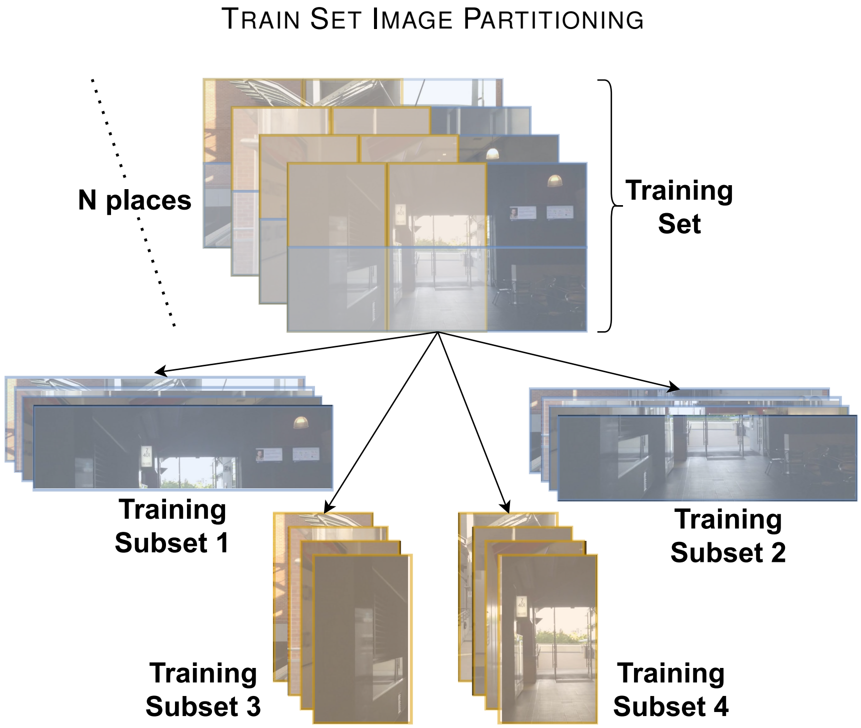


Fig. 4.3. A training subset is produced for each grid position. In this example, the grids $[(2 \times 1), (1 \times 3)]$ are used, with the blue regions highlighting the 2×1 grid and the yellow regions the 1×3 grid (the last column was omitted for visibility). The total number of regions is 5.

dimensions.

Section 4.3.4 shows how different grid setups can significantly impact the VPR performance of the overall system. Since it is not possible to predict which grid layout is best for the deployment environment without access to ground-truth information, the system uses a set of multiple, heterogeneous image regions. In this arrangement, the partitioning process is simply repeated for G different grid settings. The total number of image partitions P can thus be computed as follows:

$$P = \sum_{g=1}^{G} r_g c_g \tag{4.1}$$

where r_g and c_g represent the number of rows and columns associated with grid setup g, respectively.

4.1.3 Training and Inference

Each dataset contains N images, one per place, in their training traversal. Before the training process, P training subsets are constructed, each corresponding to one of the desired regions (Fig. 4.3). Each subset therefore also contains N image partitions.

A group of Z DrosoNets is assigned for each of the P training subsets, with each group being trained only on their respective grid position. The total number of DrosoNets in the system T is therefore given as:

$$T = PZ \tag{4.2}$$

At inference time, the query image is partitioned following the same G grids, and each DrosoNet is fed the corresponding region of its group, resulting in T score vectors for the query image. All these vectors are aggregated into a final prediction using the proposed voting module, detailed in Section 4.2.

4.2 Top Candidates Voting Module

The revised voting scheme combines all the output score vectors into a final score vector from which the reference place can be identified. Fig. 4.4 illustrates the matching process for a single query image.

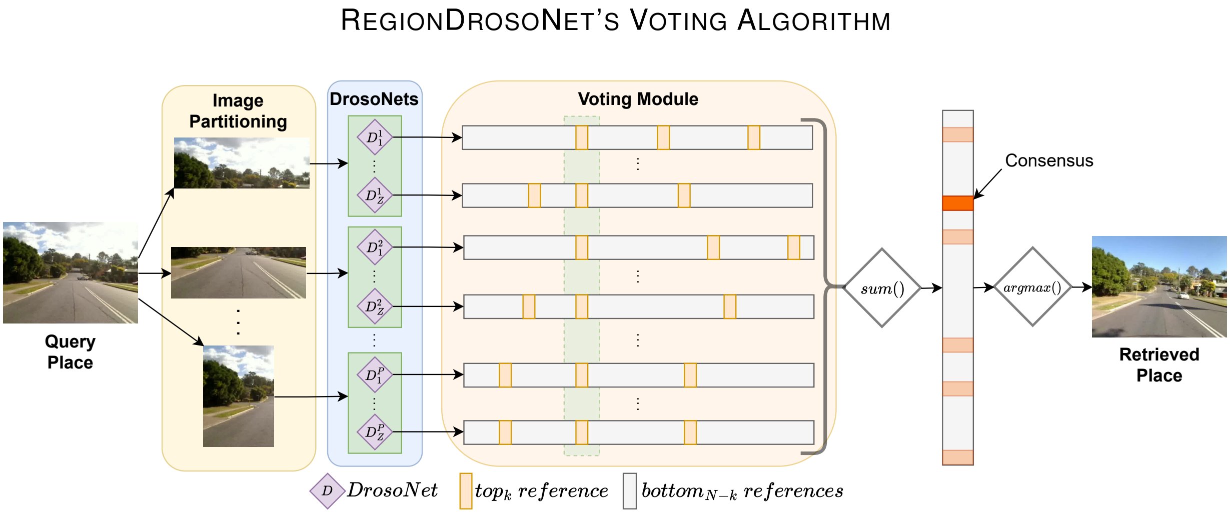


Fig. 4.4. The voting module receives all score vectors produced by each DrosoNet, with the largest K values being considered (in this case $K = 3$) and all remaining $N - K$ values being discarded.

The voting scheme combines all the output score vectors into a final score vector from which the reference place can be identified. Fig. 4.4 illustrates the matching process for a single query image.

For each of the T score vectors s, the voting vector $\hat{s}$ is constructed by setting each of the N elements $\hat{s}_n$ as:

$$\hat{s}_n = \begin{cases} s_n & \text{if } s_n \geq top_K(s) \\ 0 & \text{else} \end{cases} \tag{4.3}$$

where $top_K(S)$ represents the value of the K^{th} largest score in s, with K being a hyper-parameter. Fig. 4.4 shows an example of this operation with $K = 3$, where only the highest 3 scores per DrosoNet are considered and the remaining $N - K$ are set to 0. All the voting vectors are then summed element wise into the final score vector V:

$$v = \sum_{t=1}^{T} \hat{s}^t \tag{4.4}$$

and the retrieved reference place m is the most voted for index: $m = argmax(v)$.

4.3 Experimental Setup

This section details the experimental setup used to evaluate RegionDrosoNet, starting with a presentation of the benchmark datasets, followed by evaluation metrics, comparison VPR methods and implementation settings.

4.3.1 Datasets

The Nordland Fall, Nordland Winter, Corvin, and Gardens Point datasets were used with the same settings as in the previous Chapter. Additionally, the Berlin and St. Lucia datasets are used in these experiments, with a margin of error of 1 frame and 2 frames, respectively.

4.3.2 Evaluation Metrics

Similarly to the previous Chapter, AUC is used to evaluate VPR performance. In addition, Extended Precision (EP) is also used in this work to provide different VPR performance insights. Inference time (IT) is once again used to assess computational efficiency. IT is measured as the time elapsed from the technique receiving a query image to a match being computed. This includes the time required for any runtime image pre-processing, descriptor computation and descriptor matching. The St. Lucia dataset is used to compute the average IT over 1100 inferences, with an Intel 12900 k processor, running Ubuntu 20.03.

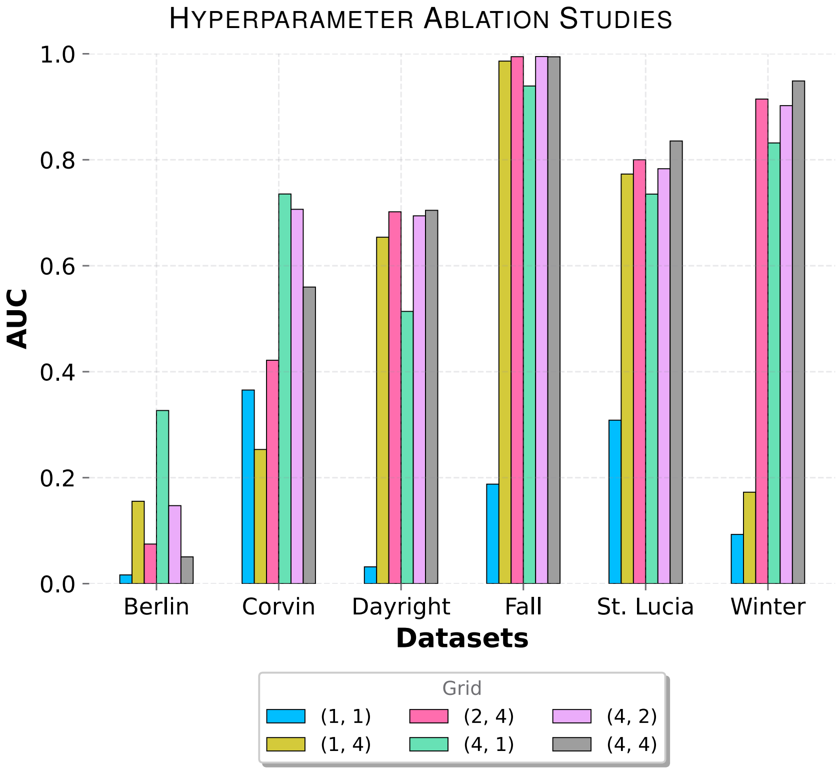
4.3.3 Comparison VPR Techniques

RegionDrosoNet is compared to several VPR techniques which claim computational efficiency as one of their main strengths: CALC, CoHOG, and the Voting mechanism introduced in the previous Chapter. Moreover, to better situate this work, a comparison against the computationally expensive techniques HybridNet and Patch-NetVLAD is performed. CALC, CoHOG and HybridNet are implemented as in [16] and Patch-NetVLAD as in [84]. Voting as is implemented as detailed in the previous Chapter and with an additional setup with 82 DrosoNets to match the same setup as RegionDrosoNet.

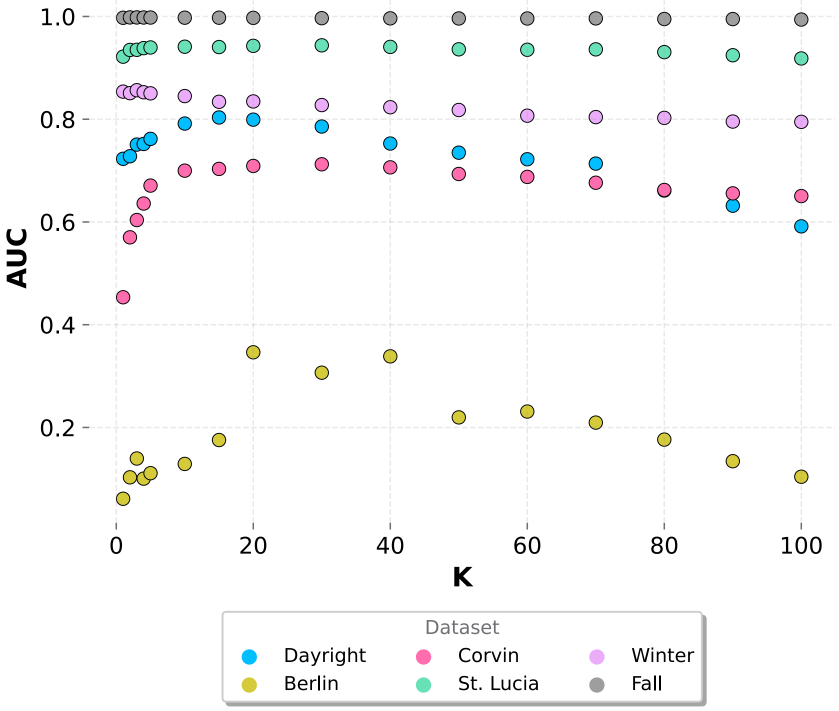
4.3.4 Ablation Studies & Implementation Details

RegionDrosoNet has three main hyperparameters: the grid setups used to construct image regions, the number of DrosoNets per group, with one group being assigned to each region, Z, and the number of top_K voted places per DrosoNet. Ablation studies are conducted to find optimal settings with the aim of providing a general setup that performs strongly across all datasets, rather than fine-tuning the system for each scenario. The results of these studies can be seen in Fig. 4.5.

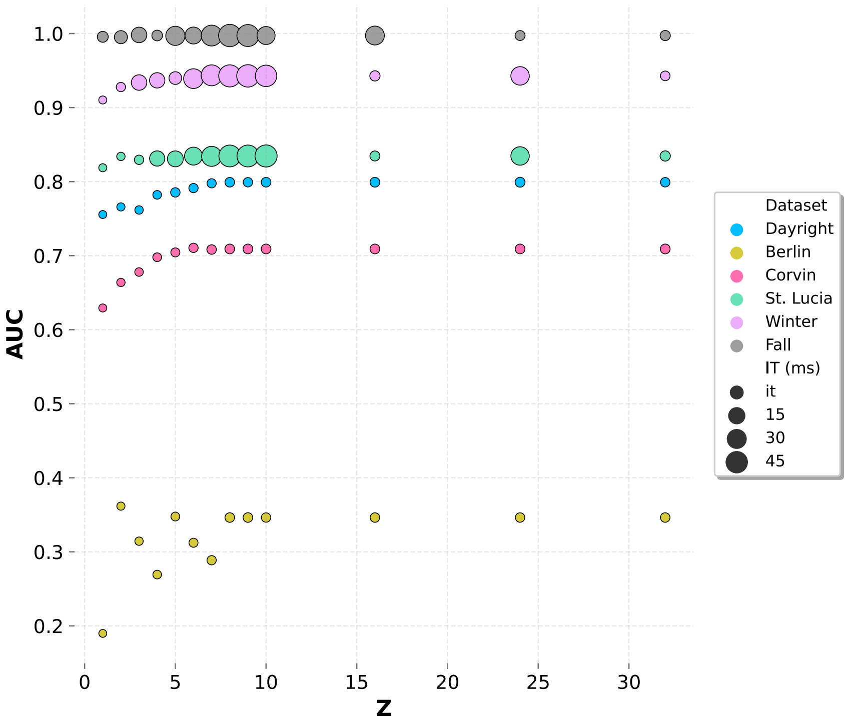
As can be seen in Fig. 4.5a, different grid settings significantly impact VPR performance, and the optimal individual grid setting varies from dataset to dataset. As such, a combination of all tested partitioning grids is used:


Hyperparameter Ablation Studies
AUC
1.0
0.8
0.6
0.4
0.2
0.0
Berlin
Corvin
Dayright
Fall
St. Lucia
Winter
Datasets
Grid
(1, 1)
(1, 4)
(2, 4)
(4, 1)
(4, 2)
(4, 4)

(a)


AUC
1.0
0.8
0.6
0.4
0.2
0
20
40
60
80
100
K
Dataset
Dayright
Berlin
Corvin
St. Lucia
Winter
Fall

(b)



(c)

Fig. 4.5. AUC impact of the region grid (4.5a), the top K voted places (4.5b) and the number of DrosoNets per region Z (4.5c).

$$[(1,1),(1,4),(4,1),(2,4),(4,2),(4,4)] \tag{4.5}$$

resulting in a total of 41 partitions, following the example scheme in Fig. 4.3.

The choice for K also has a substantial impact on VPR performance, as can be seen in Fig. 4.5b. $K = 20$ is used as it presents the best overall AUC performance across all datasets.

Finally, the number of DrosoNets per region Z has a significant impact on both AUC performance and inference time, observable in Fig. 4.5c. The system is set to $Z = 2$, as there are heavily diminishing AUC returns with higher Z values, even lowering VPR performance on Corvin and Berlin. With the choice of grids described in above, the total number of DrosoNets in the system becomes 82.

Each DrosoNet is trained for 200 epochs using the Adam optimizer [193] and with a learning rate of 0.001.

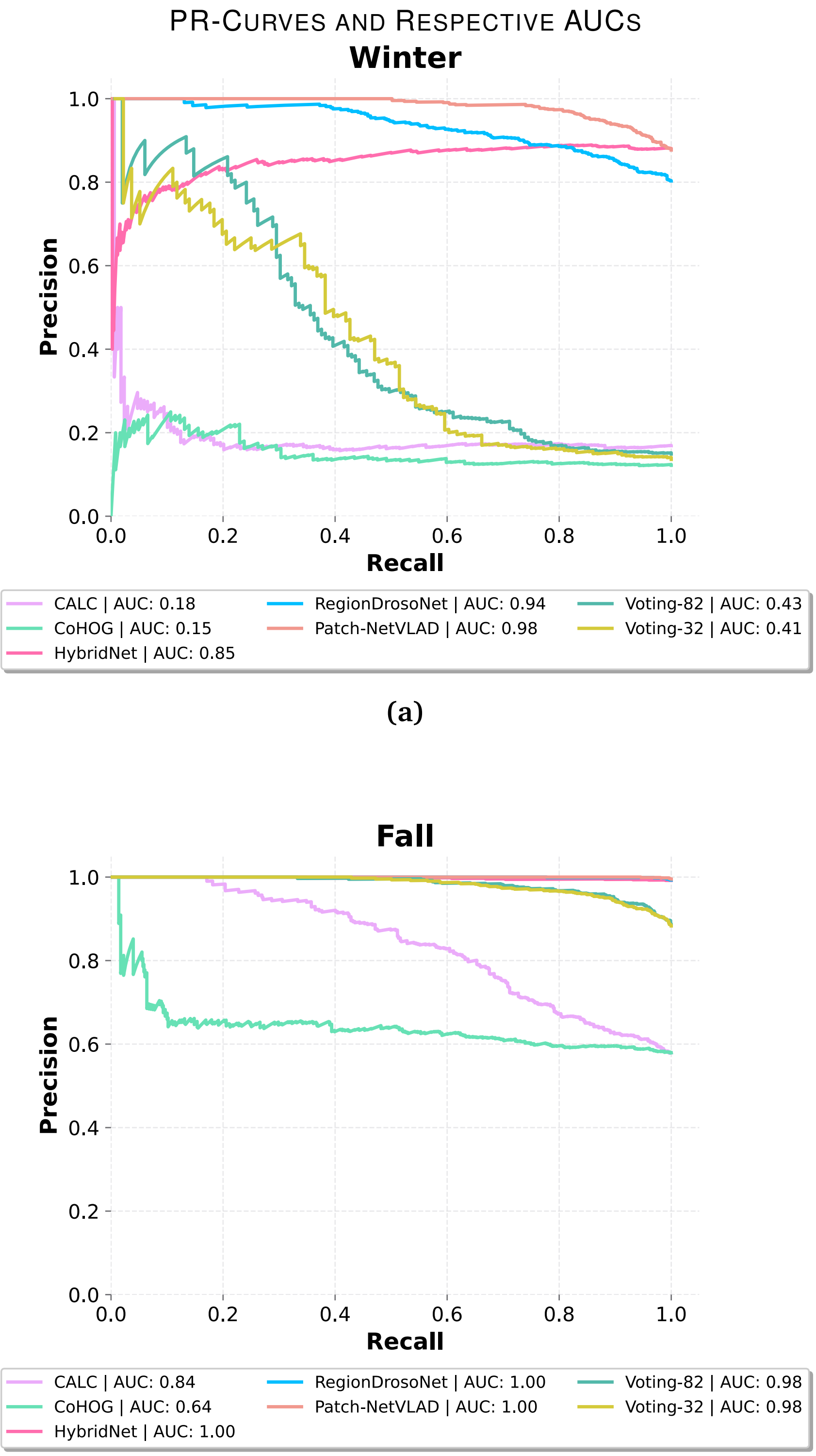
4.4 Results

This section presents and discusses results, firstly with a comparison of RegionDrosoNet versus other computationally efficiency VPR techniques, followed by a comparison against expensive methods and finalizing with a per-region performance analysis.

4.4.1 VPR Performance vs Lightweight Methods

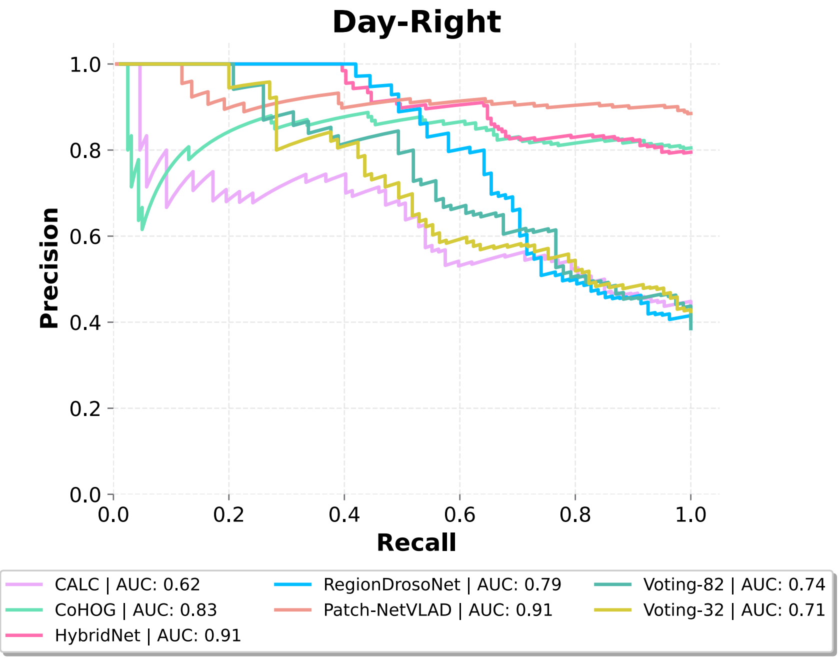
Fig. 4.6 showcases the VPR performance in terms of AUC for all tested techniques. RegionDrosoNet outperforms every other lightweight algorithm on all appearance-based datasets (Winter, Fall, St. Lucia). The performance advantage on the Winter dataset over other efficient methods is the most notable, with RegionDrosoNet more than doubling the AUC of the second best efficient technique (Voting-82). The strong robustness to appearance changes is unsurprising. DrosoNet acts as a global image descriptor when constructing its image representation (vector O) and global descriptors are better suited to deal with such changes than to deal with viewpoint variations. RegionDrosoNet is, after all, based on DrosoNet, and its performance is highly impacted by how well the underlying DrosoNet models perform.

Viewpoint performance on the Corvin dataset is also commendable, with RegionDro-

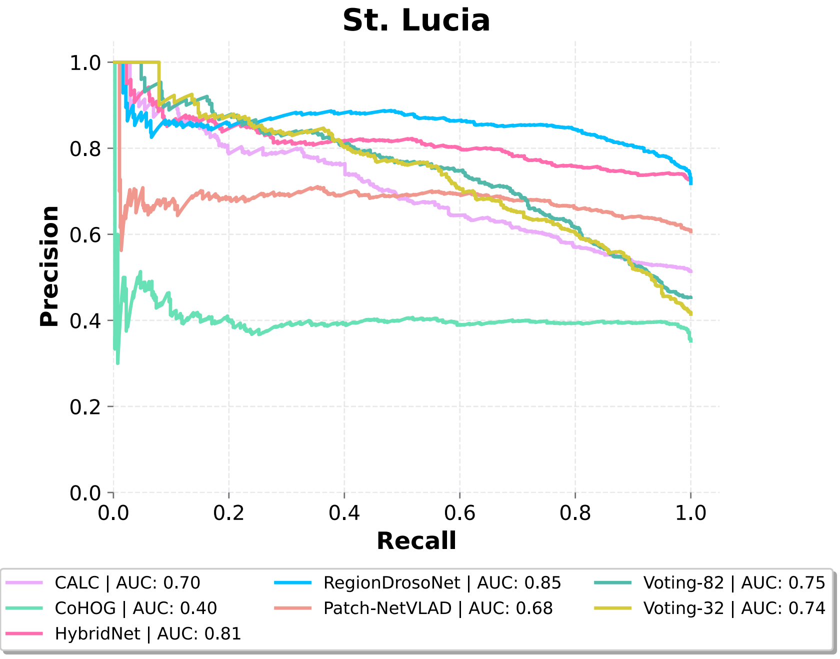

PR-Curves and Respective AUCs
Winter
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.18
CoHOG | AUC: 0.15
HybridNet | AUC: 0.85
RegionDrosoNet | AUC: 0.94
Patch-NetVLAD | AUC: 0.98
Voting-82 | AUC: 0.43
Voting-32 | AUC: 0.41
Fall
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.84
CoHOG | AUC: 0.64
HybridNet | AUC: 1.00
RegionDrosoNet | AUC: 1.00
Patch-NetVLAD | AUC: 1.00
Voting-82 | AUC: 0.98
Voting-32 | AUC: 0.98

(a)

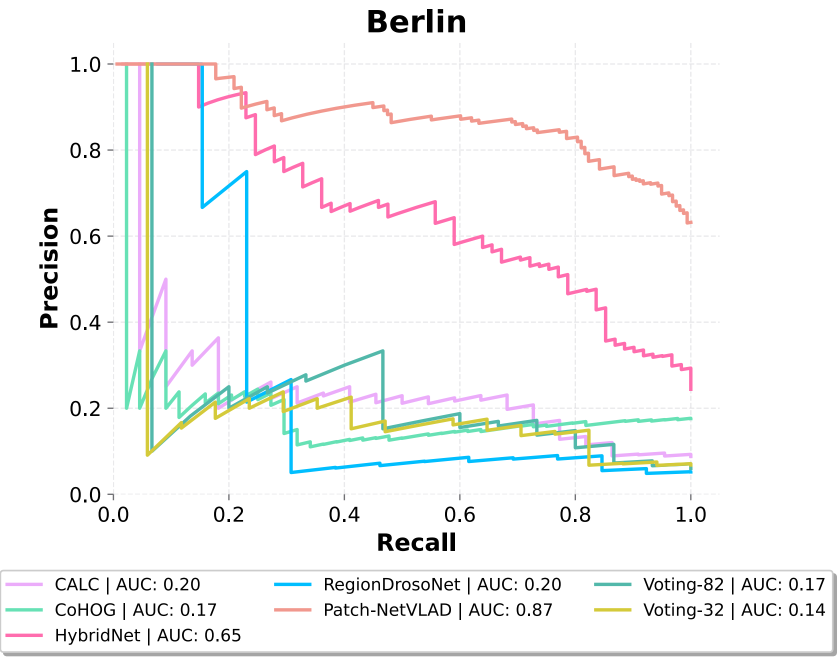
(b)


Day-Right
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.62
CoHOG | AUC: 0.83
HybridNet | AUC: 0.91
RegionDrosoNet | AUC: 0.79
Patch-NetVLAD | AUC: 0.91
Voting-82 | AUC: 0.74
Voting-32 | AUC: 0.71

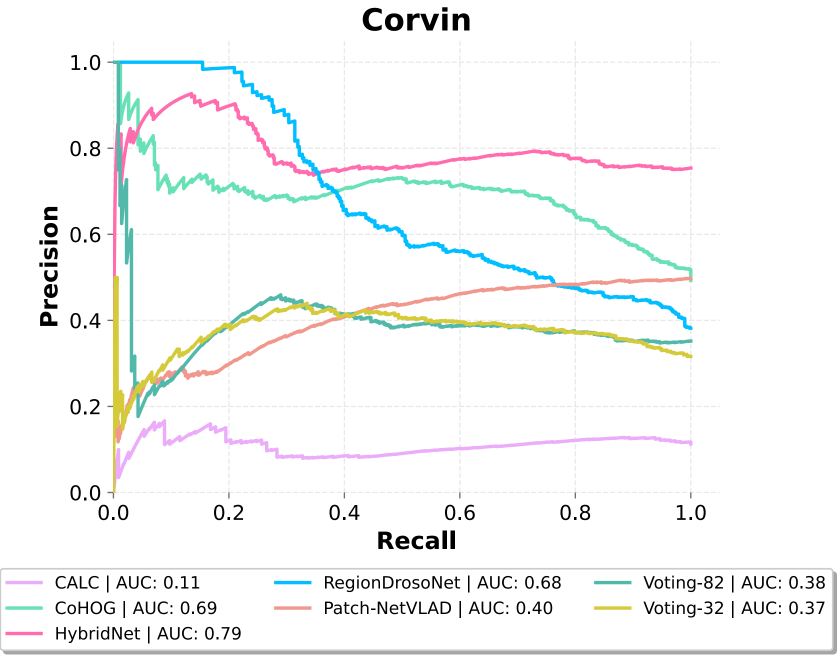
(c)


St. Lucia
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
CALC | AUC: 0.70
CoHOG | AUC: 0.40
HybridNet | AUC: 0.81
RegionDrosoNet | AUC: 0.85
Patch-NetVLAD | AUC: 0.68
Voting-82 | AUC: 0.75
Voting-32 | AUC: 0.74

(d)



(e)



(f)

Fig. 4.6. Precision-recall curves and respective AUC

TABLE 4.1: INFERENCE TIME (IT) & FRAMES PER SECOND (FPS)

Model	IT (ms)	FPS
CoHOG	671	1.49
CALC	166	6.02
Voting-32	3	333.33
Voting-82	8	125.00
HybridNet	3318	0.30
Patch-NetVLAD	2892	0.35
RegionDrosoNet	9	111.11

soNet achieving the highest EP result (Fig. 4.7) and matching CoHOG in AUC. While all lightweight techniques perform poorly on the Berlin dataset, RegionDrosoNet achieves the highest EP amongst them and ties with CALC for the highest AUC. The VPR performance of Voting-32 and Voting-82 is functionally indistinguishable, showing that simply increasing the number of DrosoNets does not contribute significantly to place matching. Conversely, the use of 82 units in the proposed pipeline provides significant improvements in VPR, as demonstrated by the performance gap between RegionDrosoNet and Voting-82. These results highlight the effectiveness of the region-based approach at improving VPR performance and the importance of increasing differentiation between the underlying DrosoNet models.

Table 4.1 shows the inference times at runtime for every tested technique on the Nordland Winter dataset, therefore testing efficiency with a map of size 1000. RegionDrosoNet is the third-fastest method, second only to Voting-32 and Voting-82, the latter due to the extra image pre-processing required by RegionDrosoNet. Nevertheless, it achieves substantially higher VPR reliability on both viewpoint and appearance-based visual challenges while remaining 18 times faster than CALC and over two orders of magnitude faster than CoHOG.

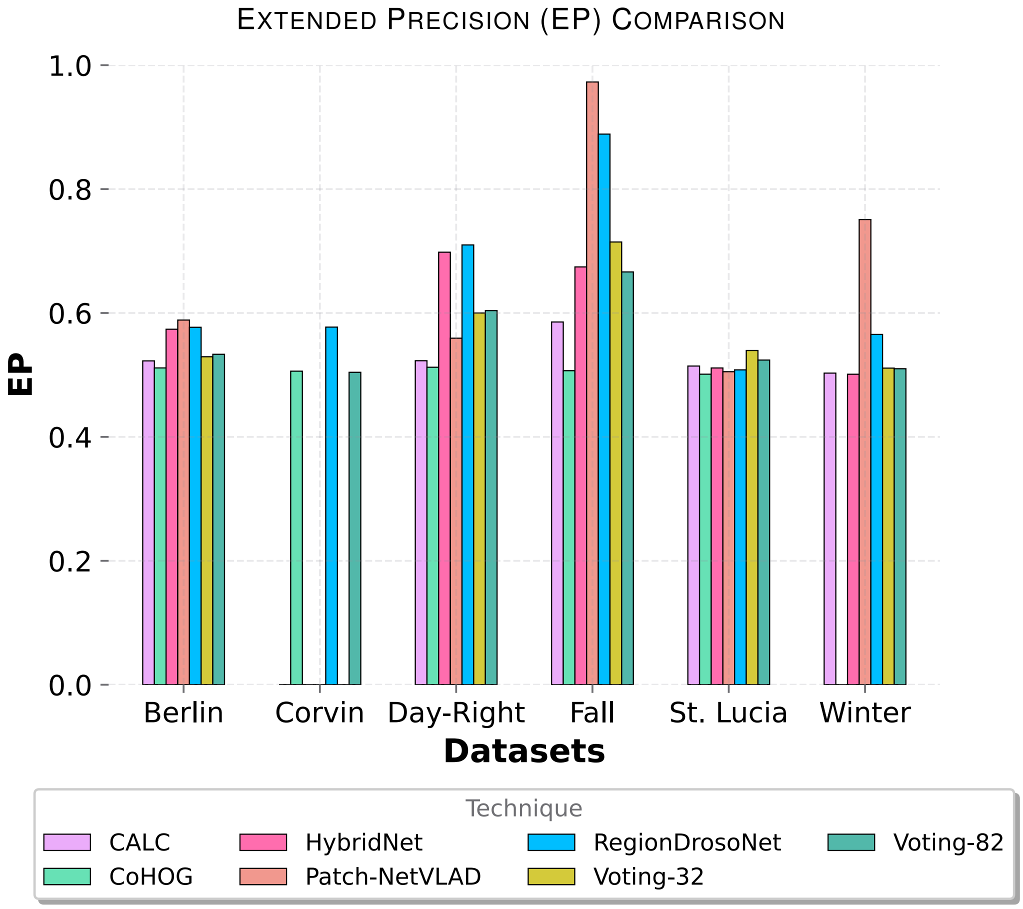


Fig. 4.7. Extended precision (EP) comparison.

Despite these efficiency advantages, it is worth noting that different methods can offer various benefits over each other. CoHOG, while requiring the reference traversal images for the reference map computation, is a trainless technique. CALC, while trained and also requiring the reference place images for the descriptor database, does not require environment specific training. RegionDrosoNet, while achieving better VPR performance and efficiency, does require environment specific training due to its dependency on DrosoNet. The choice of a VPR technique is highly application dependent and all factors such as data availability, hardware, deployment environment and risk of failure should be taken into account.

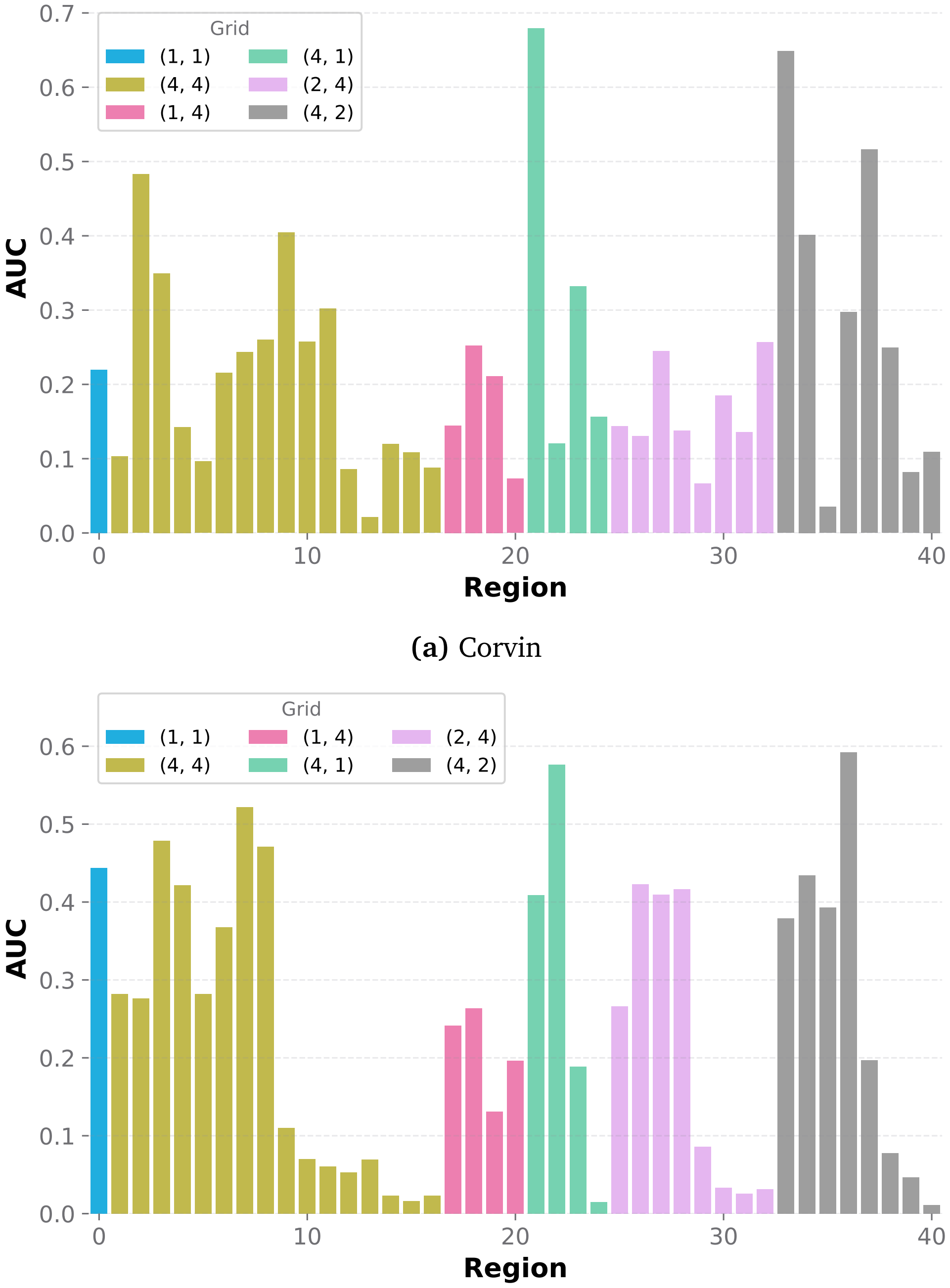
4.4.2 VPR Performance vs Expensive Methods

As can be seen in Table 4.1, HybridNet and Patch-NetVLAD are significantly slower than the lightweight methods. In particular, RegionDrosoNet achieves an inference time three orders of magnitude lower than these two methods, requiring only 9 milliseconds to perform a match versus the 3318 required by HybridNet and 2892 required by Patch-NetVLAD.

Despite its substantially lower computational requirements, RegionDrosoNet is able to compete with these expensive methods, even outperforming them on some datasets. On the Corvin dataset, RegionDrosoNet achieves higher EP (Fig. 4.7). In the challenging Winter dataset, it outperforms HybridNet in both EP and AUC. In the St. Lucia dataset, it outperforms both HybridNet and Patch-NetVLAD in AUC, and equals the two expensive competitors in EP. The highest performance drop from RegionDrosoNet is in Berlin, where it loses substantially in both AUC and EP to the two computationally costly techniques. Interestingly, in Fall and Winter, the two appearance based datasets, Patch-NetVLAD outperform RegionDrosoNet in terms of EP. This highlights the need for various performance metrics, and shows that each technique might be better or worse suited for different applications.

4.4.3 Per-Region Insights

Fig. 4.8 shows RegionDrosoNet's AUC per region on the Corvin (4.8a) and St. Lucia (4.5b) datasets. As per Eq. 4.1 and Eq. 4.5, the setup has a total of 41 regions, each



(a) Corvin

(b) St. Lucia

Fig. 4.8. AUC per region, with colour highlighting the associated grid dimensions. Within each grid, regions are placed from left-to-right, top-to-bottom. E.g., for grid $(4, 4)$, its first bar represents $row1$, $column1$, the second bar $row1$, $column2$, the fifth bar $row2$, $column1$, etc.

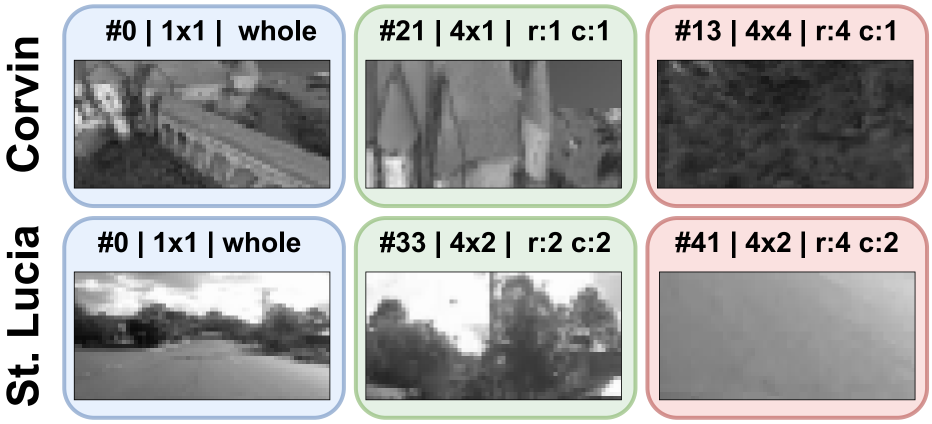


Fig. 4.9. Query regions: whole image in blue, best performing region in green, and worse performing region in red.

represented by a bar, where the colour code shows the corresponding grid arrangement from which it originated from. It is clear that some regions perform substantially better than others, and region performance is dataset dependant. Furthermore, the region corresponding to the whole query image (region 0, in blue) is not the best performing one.

Looking at Fig. 4.9, one can find visual insights for the large performance discrepancy. On Corvin, region 13 does not have enough visual detail for DrosoNet to specialize on, while 21 contains strong features. Region 13 also performs better than the whole query image, as the former has less non-detailed visual zones and less compression resulting from the image scaling pre-processing. Finally, St. Lucia follows the same pattern with its respective best and worst performing regions.

4.5 Reflecting on the Drawbacks Of RegionDrosoNet

This chapter proposes RegionDrosoNet: a novel localization system which significantly improves upon the VPR performance of current lightweight methods while remaining computational efficient. The approach relies on increasing the differentiation of different DrosoNets by training specialized groups on several image partitions. Moreover, it introduces a novel voting method which considers multiple top place candidates from each DrosoNet, allowing a correct consensus to be reached even if individual DrosoNets place an incorrect highest scoring match.

The main goal of this chapter was to improve upon the absolute VPR performance of the voting pipeline of Chapter 3 while retaining its strong efficiency characteristics. The approach of increasing model differentiation succeeded in this regard: RegionDrosoNet's performance is vastly improved over its predecessor and other lightweight techniques. Indeed, on datasets such as Nordland Winter and St. Lucia, RegionDrosoNet is even able to compete with computationally expensive methods.

Nevertheless, on datasets such as Berlin, the VPR performance results leave still a lot to be desired. Taking the insight that model differentiation is of extreme importance for multi-technique VPR, the next chapter explores the use of different baseline VPR techniques. While using VPR methods other than DrosoNet will undoubtedly increase computational cost, some navigation applications simply cannot operate below a certain degree of VPR reliability. It is thus important to investigate how to decrease the computational cost of more general ensemble based VPR.

Chapter 5

A-MuSIC: An Adaptive Ensemble System For VPR

The previous two chapters introduced the DrosoNet voting paradigm and RegionDrosoNet: two multi-model VPR approaches with a focus on computational efficiency. This chapter continues the exploration of the multi-technique VPR paradigm but with a greater emphasis on absolute VPR performance. Using multiple techniques in a VPR system will always introduce an efficiency penalty when compared to using a single backbone method. The work in this chapter aims to mitigate this shortcoming by limiting the usage of techniques to the minimum required for current navigation. Achieving such a flexible system without relying on additional ground-truth information at runtime is not trivial. The approach taken in the following sections exploits the sequential navigation assumption to infer the VPR quality of the set of techniques and then select the most suitable subset for the current environment.

5.1 Computing Sequential Information For Adaptive Technique Switching

Visual Place Recognition is often cast as an image retrieval task [10]. A database of reference template images, commonly image descriptors, is assumed to be pre-existing, and the goal of VPR is to match the currently observed place with one of these templates.

During navigation, a technique performs VPR for the current observed frame q. The output of the technique is a similarity vector S_q, where each element $S_{q,n}$ represents the similarity score associated with the n^{th} reference place in the map. In the standard image retrieval setting, the reference template which achieves the highest similarity is taken to be the correct match.

SIC performs a sequential search over recent score vectors, requiring these to be stored appropriately. The matrix S is therefore constructed by stacking similarity vectors by time of observation, that is, S_q is added immediately after S_{q-1} (Fig. 5.2a).

5.1.1 Sequential Information Consistency (SIC)

Inspired by the trajectory similarity score introduced in [95] and the analysis of query candidates with the highest similarity in [106], SIC performs a search over previous query similarity vectors for the K most similar templates. For each top scoring candidate k in the similarity vector S_q of query frame q, a sequential consistency θ value is calculated as follows:

$$\theta_k = \sum_{f=0}^{F} max(S_{q-f,k-f-W:k-f+W}) \tag{5.1}$$

where F denotes how many past queries should be evaluated and $k-f-W : k-f+W$ denotes a vector slice of size $W * 2 + 1$ around the center $k-f$. Fig. 5.1 exemplifies the computation with $K = 2, F = 2, W = 1$. The purpose of W is to relax the sequential navigation assumption and compensate for small navigation drifts. Fig. 5.2 shows the transformation from the regular similarity scores to the sequential consistency scores.

The candidate which achieves the highest θ, denoted as m, is considered to be the correct match for query q:

$$m = \underset{k}{\operatorname{argmax}}\, \theta \tag{5.2}$$

and θ_m therefore representing the maximum $theta$ value.

By evaluating only a fixed number of top matching candidates, SIC's computational cost is independent of the number of reference places in the internal map, as discussed later in Section 5.3. Conversely, typical sequence matching schemes suffer from increased computation times as the database of reference images increases [95]. Maintaining a low and stable computational profile is especially important for SIC, as it is intended to be used on multiple techniques in tandem.

5.1.2 Multi-Technique SIC (MuSIC)

MuSIC uses SIC's sequential consistency metric to choose amongst a set of VPR techniques with which to perform VPR for the current query image. As shown in Fig. 5.3, for a query image (q), MuSIC runs SIC with every available t in T, generating the corresponding θ_m^t values. The candidate achieving the highest θ is then selected as the correct

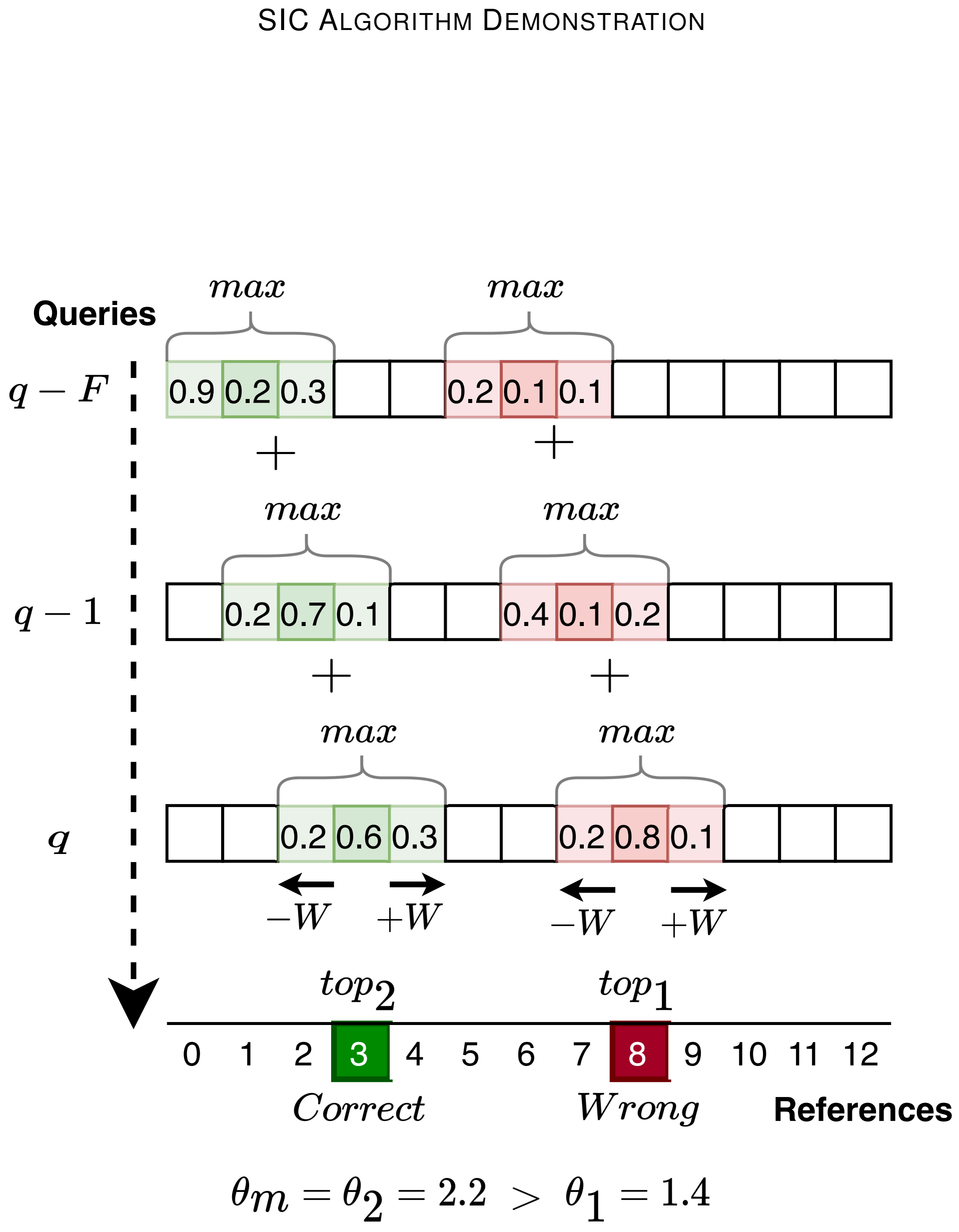
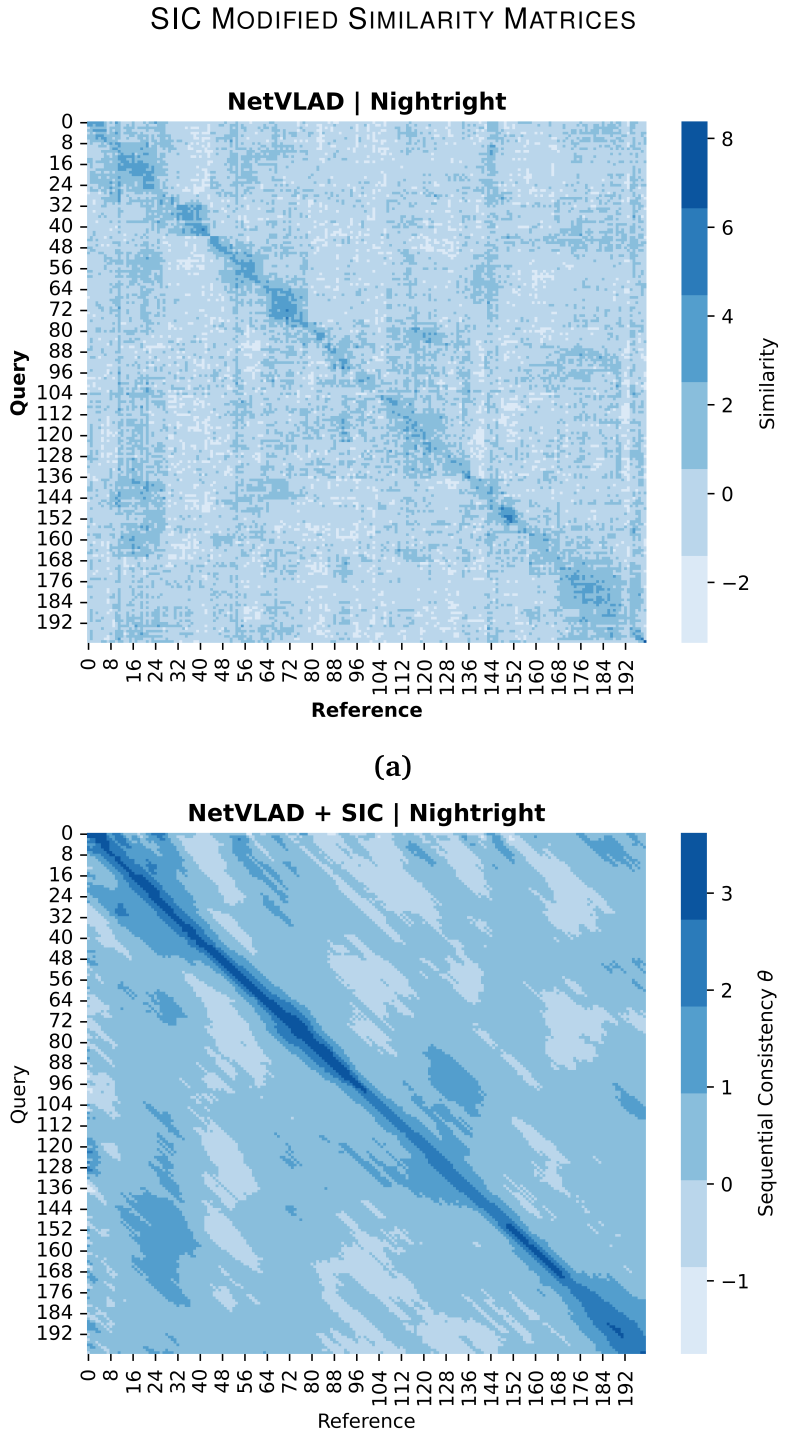


Fig. 5.1. SIC operating with $K = 2, F = 2, W = 1$. The correct match for the current query q is the reference place 3. However, the reference place 8 erroneously achieves the highest similarity. By computing the sequential consistency θ from the frame-to-frame continuity of previous similarity vectors, SIC identifies the correct match.



(a)

(b)

Fig. 5.2. (a) shows the usual similarity scores matrix produce, while (b) shows the sequential consistencies matrix computed by SIC (K=200, F=20, W=1)

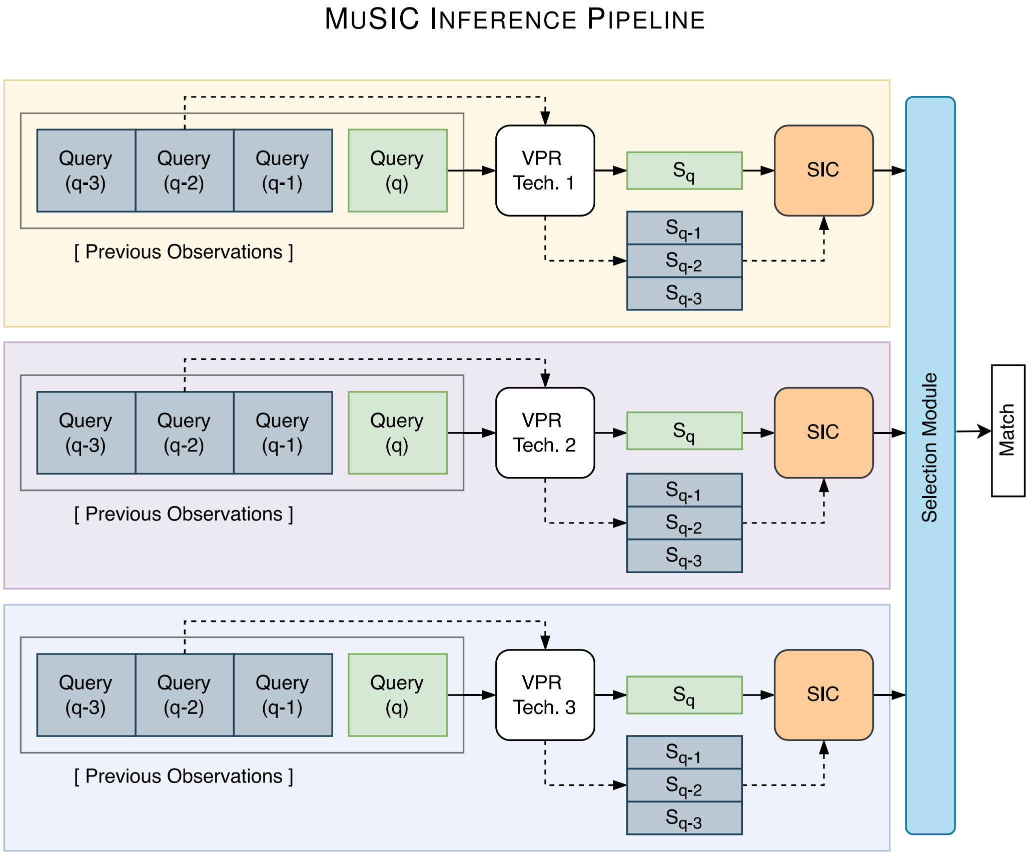


Fig. 5.3. MuSIC operating with 3 VPR techniques. For a given query image q, SIC analyses recent observation score vectors $S_{q-1}, S_{q-2}, S_{q-3}$ for each respective technique, outputting their sequential consistency scores. The output of SIC is then used to select which technique is used to deliver the match.

match.

Different techniques have different ranges of output similarity scores. Since θ is computed directly from these values, the vectors are scaled before applying SIC. Each score $S_{q,n}$ is normalized using the following equation:

$$\hat{S}_{q,n} = \frac{S_{q,n} - \mu}{\sigma} \tag{5.3}$$

where $\hat{S}_{q,n}$ is the scaled value, μ and σ are the mean and standard deviation of the similarity vector S_q currently being scaled, respectively.

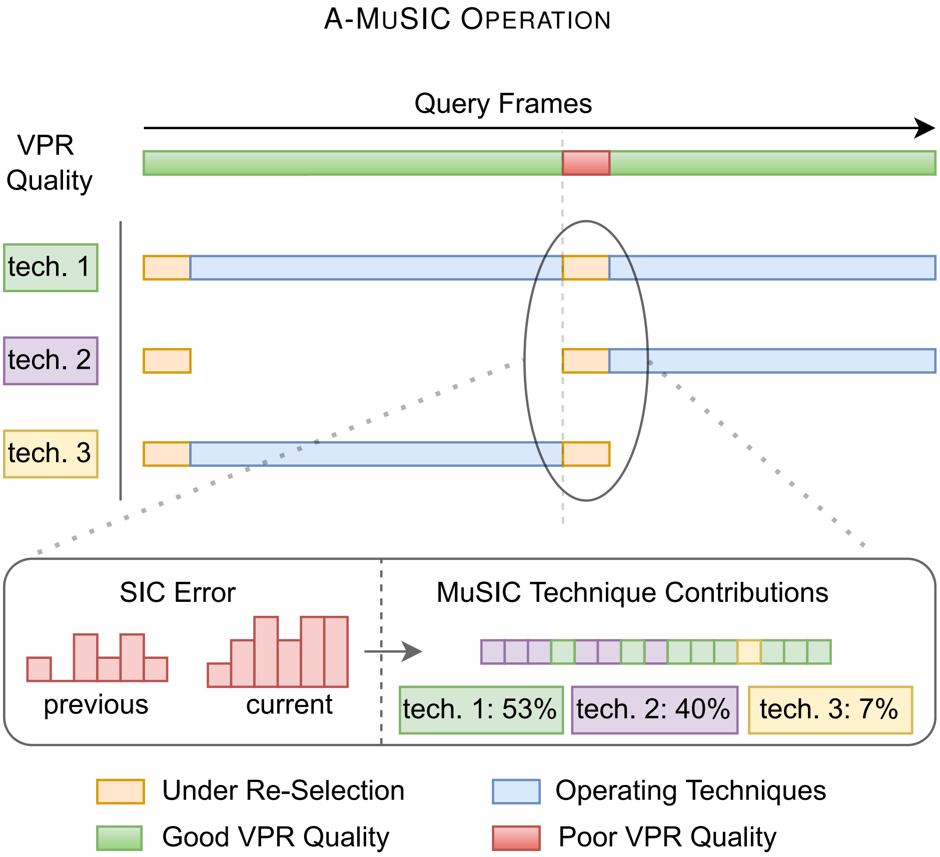


Fig. 5.4. From a set of three VPR techniques (tech. 1, tech. 2 and tech. 3), A-MuSIC selects a subset to remain active. The first selection occurs by default at the beginning of the navigation. The inner component MuSIC computes how much each technique is contributing in the current environment and only the most helpful techniques remain active (tech. 1 and tech. 2). When the VPR quality of the active subset degrades (denoted in red), the value of SIC error increases, triggering a re-selection of techniques, after which only tech. 1 and tech. 2 remain active.

SIC operates individually on each technique t, therefore requiring their respective F past observation score vectors $S^t_{q-F:q-1}$ and current observation vector S^t_q. The final output match is selected by taking the reference template which achieves maximum θ^t_m amongst all techniques.

5.1.3 Adaptive MuSIC (A-MuSIC)

The goal is to find the optimal, minimal subset of techniques $\hat{T}$ for the current navigation environment and only attempt to change this active subset when a change of sequential consistency is detected. Two distinct processes are introduced and illustrated in Fig. 5.4 to achieve this behavior. Firstly, a technique selection stage is presented, allowing the system to pick the most consistent subset of techniques in current navigation. Secondly, a re-selection trigger protocol is detailed, allowing the system to monitor ongoing sequential consistency and issuing a re-selection stage upon significant degradation.

Technique Selection

MuSIC uses the most consistent technique in T, according to SIC, to find a match for the current query frame. Over F queries, MuSIC constructs the technique history vector th

$$th = [t^*_q, t^*_{q-1}, ..., t^*_{q-F}] \tag{5.4}$$

where t^*_q is the chosen technique to place a match for query q.

The proportion of selection ρ of each technique t is then given by

$$\rho^t = \frac{|th^t|}{M} \tag{5.5}$$

where $|th^t|$ is the number of occurrences of technique t in th. Therefore, ρ^t ranges from 0 (t was never chosen) to 1 (t was always chosen). Techniques are then added to the active subset $\hat{T}$ by order of highest ρ until the cumulative ρ value reaches a designated threshold E. Lower E values increase the probability that more techniques will be excluded from $\hat{T}$.

Re-Selection Trigger

During navigation, sequential consistency information of the active techniques over F query frames is stored in the error history vector ch:

$$ch = [\epsilon_q^t, \epsilon_{q-1}^t, ..., \epsilon_{q-F}^t], \forall t \in \hat{T} \tag{5.6}$$

The stored ch vector contains the distribution of sequential consistency of the current $\hat{T}$. Over the next F' frames, a new vector ch' is computed. If the consistency values stored in ch are significantly different from the newly generated ch', a re-selection is triggered.

To assess if the distributions are significantly different, a paired-sample, one-tailed t test [194] is used at the significance level of P. The null hypothesis H_0 is that the mean of ch and ch' are equal. The alternative hypothesis H_1 is that the mean of ch' is statistically greater than the mean error of ch. A paired-sample version of the test is used as both ch and ch' are generated from the same set of techniques $\hat{T}$.

When H_0 is rejected, a re-selection process is conducted, using the same F' frames that triggered the re-selection to compute a new set of active techniques. Regardless of the outcome of the t test, the stored ch is always updated to equal ch'.

5.2 Experimental Setup

Several experiments are performed to evaluate the VPR performance and computational efficiency of the proposed methods. This section provides details on datasets, baseline VPR techniques, technical implementations and hyperparameter settings.

5.2.1 Datasets

All used datasets are described in Section 2.1.6 and share a margin of error of 1 frame except St. Lucia and 17 Places, for which 2 and 5 error frames are allowed, respectively.

5.2.2 Evaluation Metrics

As in the previous chapter, AUC and EP are used to evaluate VPR performance. Computational cost is also once again assessed by computing the time required to match a single query, measured in milliseconds (ms). However, as the focus of this chapter is on algorithms which act as an additional step on top of baseline techniques, the underlying technique's matching time is excluded from the computation.

5.2.3 Implementation Settings

The implementations available in [16] for HOG, CoHOG, CALC and NetVLAD are used, all with the provided default settings. MuSIC is not required to operate with a specific number or combination of VPR techniques, but this set provides some variety in baseline approaches: handcrafted descriptors (HOG and CoHOG), lightweight CNN (CALC), and costly CNN (NetVLAD). To allow for fair comparison between multi-technique systems,

TABLE 5.1: GRID-SEARCH SETTING COMBINATIONS

Parameter	Settings
K	1; 2; 3; 4; 5; 10; 50; 150; 200
F	5; 10; 25; 50; 75; 100
E	0.3; 0.5; 0.7; 0.9
P	0.01; 0.05; 0.1
W	3; 5; 7; 9; 11

MPF was deployed to use the same set of techniques as MuSIC, with the remaining settings as per [181]. SeqSLAM parameters are as given in [95].

As detailed in Section 5.1, MuSIC contains three hyperparameters: $K, F,$ and W. The same parameters are selected for all datasets with the goal of providing a general configuration. K, therefore, is set to 200: the number of images of the smallest benchmark dataset. For fair comparison with SeqSLAM, F is set to 20: the same number of searched past observations. Finally, W is set to 1, as it showed the best average performance across all datasets.

A-MuSIC contains several hyperparameters, some inherited from SIC and MuSIC, and some newly added. Note that A-MuSIC behaves differently from MuSIC and thus, to find suitable settings, a new grid-search ablation study is conducted using the benchmark datasets. Table 5.1 provides details on all the setting combinations tested. AUC performance is grouped by setting combination, averaging across datasets.

The 95% AUC percentile of combinations is further analyzed in terms of VPR performance, required technique runs, and the ratio of AUC per run. These results are shown in 5.2. The configuration resulting in the least technique usage is also the one achieving the best trade-off in VPR quality. A-MuSIC is therefore configured with these settings, highlighted in bold in the table.

TABLE 5.2: RELEVANT CONFIGURATIONS

Relevancy	K	F	E	W	P	AUC	Tech. Runs
Best AUC	200	50	0.9	3	0.01	0.91	1690
Least Tech. Runs	200	25	0.7	3	0.01	0.88	965
Best Ratio	**200**	**25**	**0.7**	**3**	**0.01**	**0.88**	**965**

5.3 Results

This section starts by presenting results comparing SIC applied to the individual baseline techniques and baselines plus embedded sequential information from SeqSLAM. It then focuses on the comparison between MuSIC, A-MuSIC, and MPF. Finally, a deeper analysis of the selection patterns of MuSIC and A-MuSIC is given, providing additional insight into the computational efficiency and VPR performance of the two systems.

5.3.1 Comparison To Baselines Plus Sequential Information

SIC's VPR performance results are summarized in Table 5.3. Fig. 5.5 displays the match computation times for increasing internal map sizes, including also MPF and MuSIC (but not A-MuSIC as such a scale does not make sense given its operation).

Regarding average VPR performance, Table 5.3 shows that, except for HOG, individual techniques plus SIC achieve higher average AUC than their SeqSLAM counterparts. The best performing baseline technique with infused sequential information is NetVLAd+SIC, at 0.83 AUC, with SeqSLAM achieving its maximum when pared with CALC, at 0.72 AUC. The results are similar in terms of EP. The highest average EP for single technique plus sequential step is achieved by CoHOG+SIC, at 0.63 EP, with the

TABLE 5.3: VPR PERFORMANCE: AREA UNDER PRECISION-RECALL CURVE (AUC) AND EXTENDED PRECISION (EP)

	Winter		Fall		Berlin	
	AUC	**EP**	**AUC**	**EP**	**AUC**	**EP**
HOG	0.28	0.54	0.85	0.59	0.03	0.00
CALC	0.29	0.51	0.88	0.5	0.06	0.01
CoHOG	0.22	0.52	0.84	0.59	0.26	0.51
NetVLAD	0.27	0.52	0.68	0.59	0.81	0.01
HOG+SeqSLAM	0.97	0.6	0.98	0.88	0.06	0.56
CALC+SeqSLAM	0.93	0.76	1.00	0.97	0.75	0.56
CoHOG+SeqSLAM	0.44	0.55	0.91	0.72	0.19	0.54
NetVLAD+SeqSLAM	0.52	0.57	0.99	0.78	0.97	0.71
HOG+SIC	0.66	0.51	0.98	0.82	0.04	0.03
CALC+SIC	0.92	0.66	1.00	0.94	0.58	0.27
CoHOG+SIC	0.72	0.53	1.00	0.73	0.91	0.77
NetVLAD+SIC	0.88	0.50	0.98	0.81	0.96	0.76

	Night-Right		17 Places		St. Lucia		Average	
	AUC	**EP**	**AUC**	**EP**	**AUC**	**EP**	**AUC**	**EP**
HOG	0.03	0.09	0.18	0.01	0.70	0.52	0.35	0.29
CALC	0.14	0.02	0.13	0.01	0.70	0.51	0.37	0.26
CoHOG	0.43	0.52	0.12	0.01	0.55	0.02	0.40	0.36
NetVLAD	0.53	0.51	0.19	0.01	0.46	0.5	0.49	0.36
HOG+SeqSLAM	0.42	0.71	0.2	0.03	0.70	0.51	0.56	0.55
CALC+SeqSLAM	0.57	0.52	0.3	0.03	0.76	0.52	0.72	0.56
CoHOG+SeqSLAM	0.24	0.54	0.15	0.03	0.49	0.55	0.40	0.49
NetVLAD+SeqSLAM	0.67	0.53	0.31	0.03	0.82	0.53	0.71	0.53
HOG+SIC	0.05	0.50	0.23	0.51	0.73	0.88	0.45	0.54
CALC+SIC	0.49	0.18	0.32	0.01	0.65	0.82	0.66	0.48
CoHOG+SIC	0.93	0.68	0.36	0.51	0.83	0.57	0.79	0.63
NetVLAD+SIC	0.98	0.79	0.32	0.01	0.87	0.52	0.83	0.57

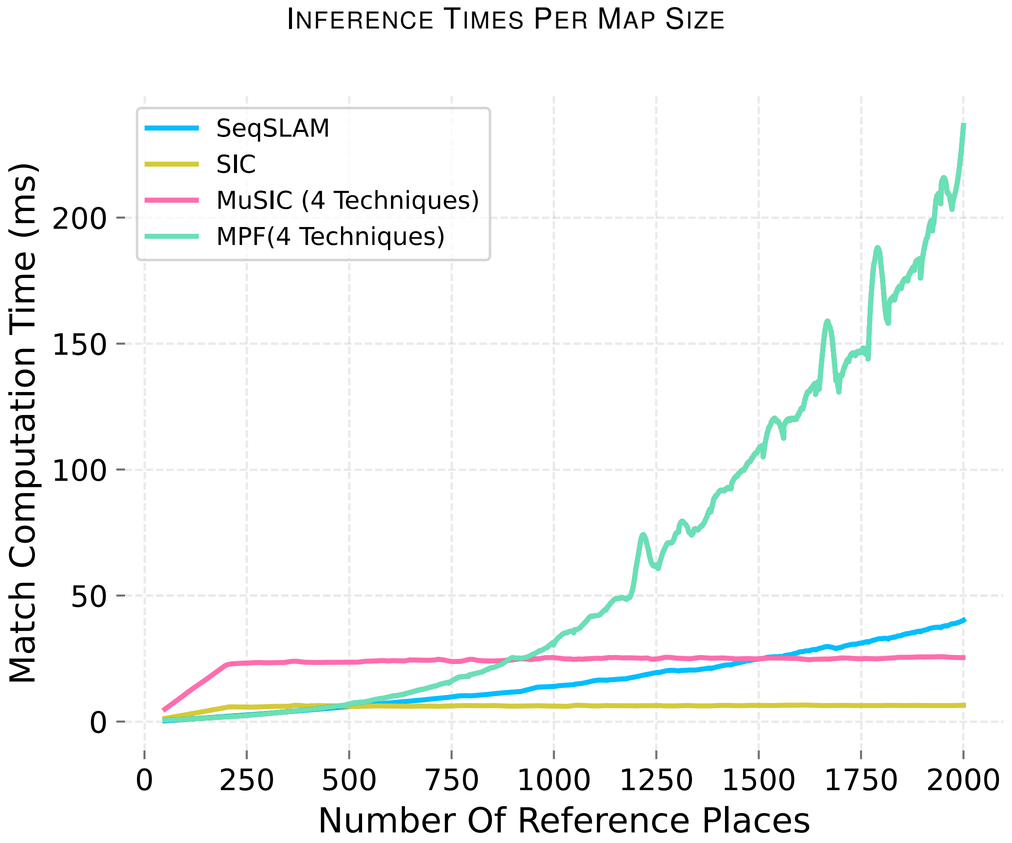


Fig. 5.5. Computational time, in milliseconds, required to match a single query frame at different map sizes, excluding baseline technique computation.

highest for SeqSLAM at 0.56, again when combined with CALC.

With respect to computational cost, Fig. 5.5 shows that, starting from internal map sizes containing more than 500 reference places, SIC computes a match faster than SeqSLAM. As explained in 5.1.1, this is due to the constant number of candidates evaluation K.

5.3.2 MuSIC, A-MuSIC, and Fusion

The results obtained by the multi-technique VPR systems can be observed in Table 5.4.

The AUC results in the Night-Right dataset illustrate the downside of fusion approaches such as MPF. While the underlying techniques NetVLAD and CoHOG perform well (Table 5.3), CALC and HOG bring the VPR performance of MPF down, even lower than the two best baseline techniques. On the other hand, MuSIC correctly identifies that best performance can be achieved by running NetVLAD (5.6d). In cases where fusion does improve overall performance, such as Winter and Fall, MuSIC still achieves better VPR performance.

However, despite the strong VPR performance exhibited by MuSIC, its computational time is unsurprisingly high, at roughly the sum of the underlying individual techniques. This is where the adaptive component of A-MuSIC shows its advantages. Referring back to Table 5.4, A-MuSIC cuts the average inference time of MuSIC by a factor of almost two thirds, while retaining an average AUC of 0.88, versus the 0.95 of MuSIC. When compared to MPF, A-MuSIC achieves a substantially average AUC (0.88 versus 0.88) while being three times faster.

Due to its operation, the benefits of the adaptive system vary greatly between datasets, depending on the number of re-selection triggers and active techniques. Section 5.3.3 provides a closer inspection of the selection patterns produced by A-MuSIC.

TABLE 5.4: VPR PERFORMANCE (AUC) AND PREDICTION TIME (MS)

	Winter		**Fall**		**Berlin**		**Night-Right**		**St. Lucia**		**Average**	
	AUC	**ms**	**AUC**	**ms**	**AUC**	**ms**	**AUC**	**ms**	**AUC**	**ms**	**AUC**	**ms**
MPF	0.58	1578	0.95	1595	0.47	1049	0.38	1020	0.43	1672	0.66	1698
MuSIC	0.95	1548	1.00	1554	0.96	1051	0.98	1007	0.87	1631	0.95	1547
A-MuSIC	0.86	246	0.98	382	0.92	682	0.97	689	0.66	618	0.88	668

5.3.3 Analyzing Technique Selection

This section presents an analysis into the selection patterns of MuSIC and A-MuSIC, with the goal of identifying key aspects of their operation and providing further insight into their obtained results.

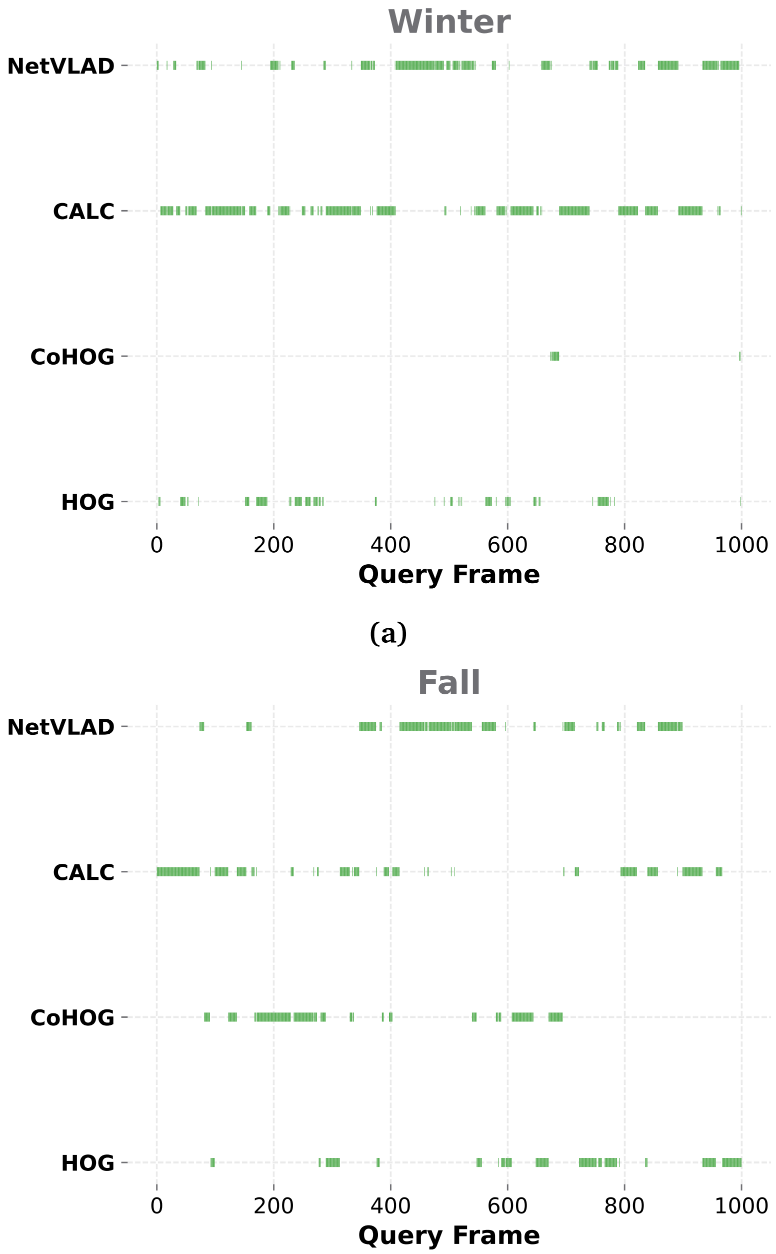
MuSIC Selections

The selection pattern produced by MuSIC is observable in Fig. 5.6, where two distinct cases stand out. In the plots 5.6c, 5.6d and 5.6e, NetVLAD is the clear dominant technique, being always selected by MuSIC. In the Berlin and Night-Right datasets, this is congruent with the VPR performance metrics, as NetVLAD+SIC obtain the highest AUC and EP (Table 5.3). This is not consistent in the 17 Places dataset, where, according to VPR performance, CoHOG should be the dominant technique.

The plots 5.6a, 5.6b and 5.6f show the second distinct case, where multiple techniques contribute to the VPR performance of the system. In terms of AUC, the lower performance bound is given by the best performant technique (St. Lucia), while the highest performance bound can be higher than any of the individual techniques (Winter). According to [174], the latter should occur when there is high complementarity between the underlying techniques.

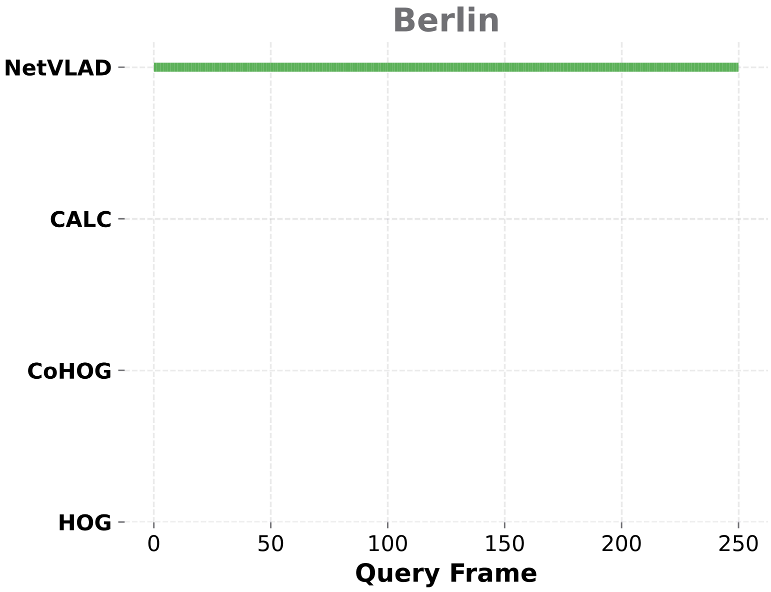
Overall, MuSIC successfully identifies the correct technique to be used per frame, allowing for a system which can be deployed on a wide range of environments without extra ground-truth information.

MuSIC Technique Selection Pattern

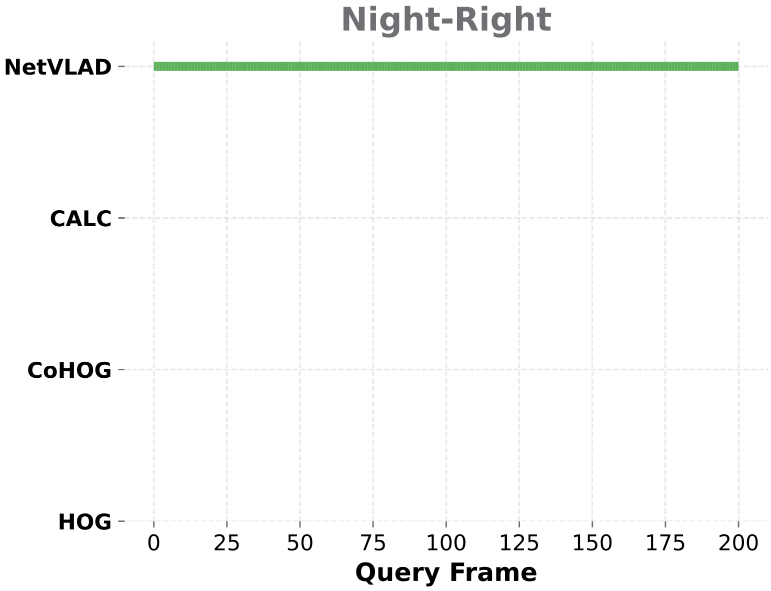


(a)

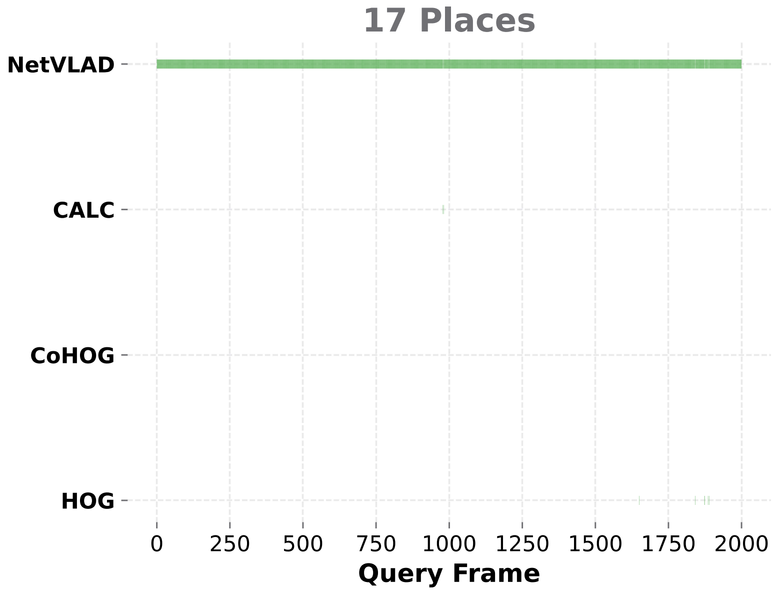
(b)


Berlin
NetVLAD
CALC
CoHOG
HOG
0
50
100
150
200
250
Query Frame

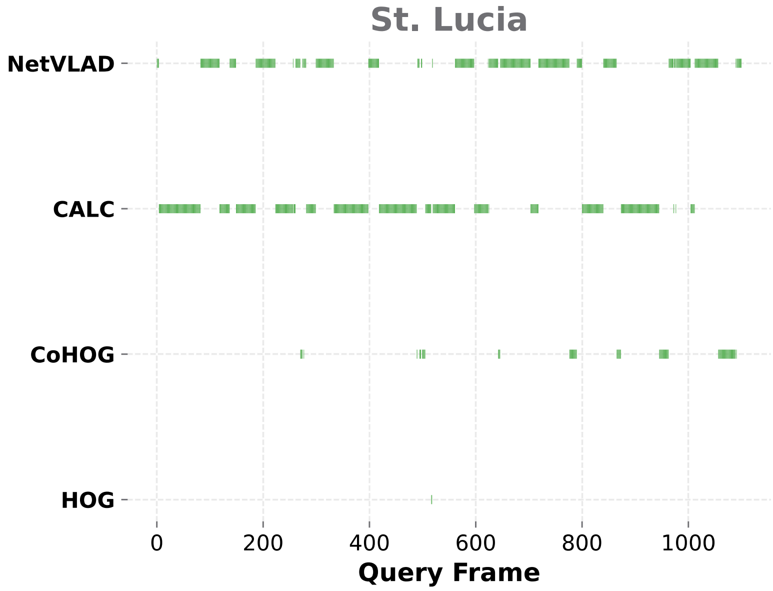
(c)


Night-Right
NetVLAD
CALC
CoHOG
HOG
0
25
50
75
100
125
150
175
200
Query Frame

(d)



(e)



(f)

Fig. 5.6. MuSIC selection patterns over all tested benchmark datasets.

A-MuSIC Selections

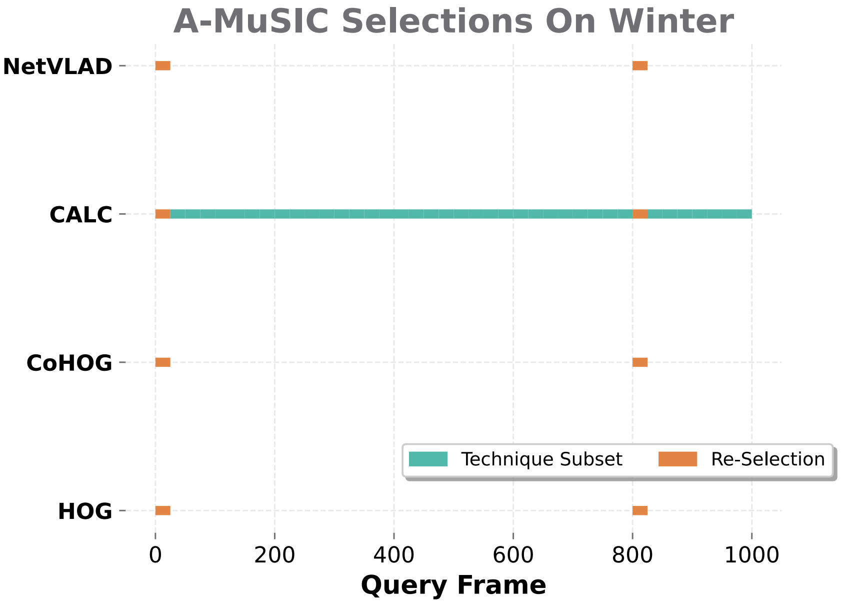
The graphs in Fig. 5.7 show the selection patterns of A-MuSIC on different datasets, highlighting the re-selection stages in orange. CALC and NetVLAD were by far the most used techniques, with CALC being mostly employed on appearance based datasets (Figs. 5.7a, 5.7b) and NetVLAD with viewpoint variations (Figs. 5.7c, 5.7d). Interestingly, the St. Lucia dataset is divided between these two techniques, as observed in Fig. 5.7e. While HOG makes a substantial appearance on the Fall dataset (Fig. 5.7b), CoHOG makes only a small standalone contribution also on the Fall dataset (apart from re-selection stages where all techniques are run).

These graphs also make clear where the efficiency gains come from. At the most extreme example, the fact that CALC was nearly the only technique used on the Winter dataset (Fig. 5.7a), together with a single re-selection, explains the sharp decrease in inference time between MuSIC and A-MuSIC. In the Night-Right dataset, Fig. 5.7d, the match time does not reduce nearly as much, as NetVLAD, the most costly technique, was selected for the entire test.

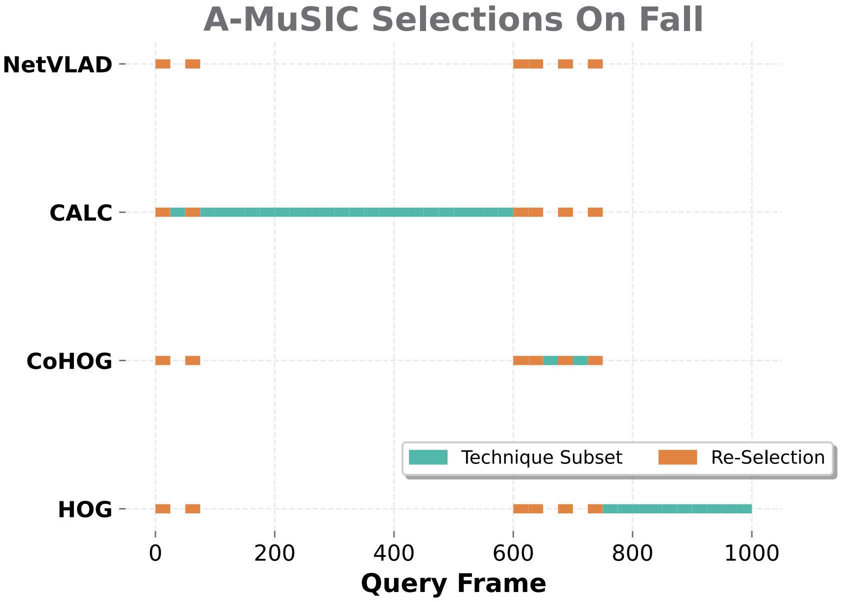
5.4 Conclusions and Considerations on A-MuSIC

This chapter starts by proposing a multi-technique VPR system which uses the frame-to-frame sequential continuity of several VPR techniques to select which should be used to perform VPR on the current query image. Then, with the goal of reducing the high computational costs of multi-technique pipelines, a mechanism to both select and re-select a subset of techniques is proposed.

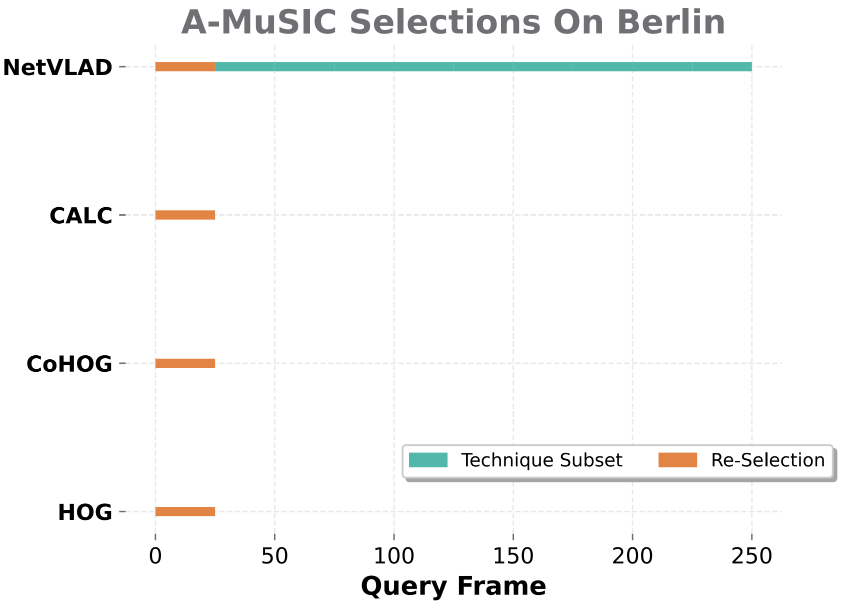
SIC is the first contribution, an algorithm which performs a search over recent ob-


A-MuSIC Selections On Winter
NetVLAD
CALC
CoHOG
HOG
Technique Subset
Re-Selection
0
200
400
600
800
1000
Query Frame

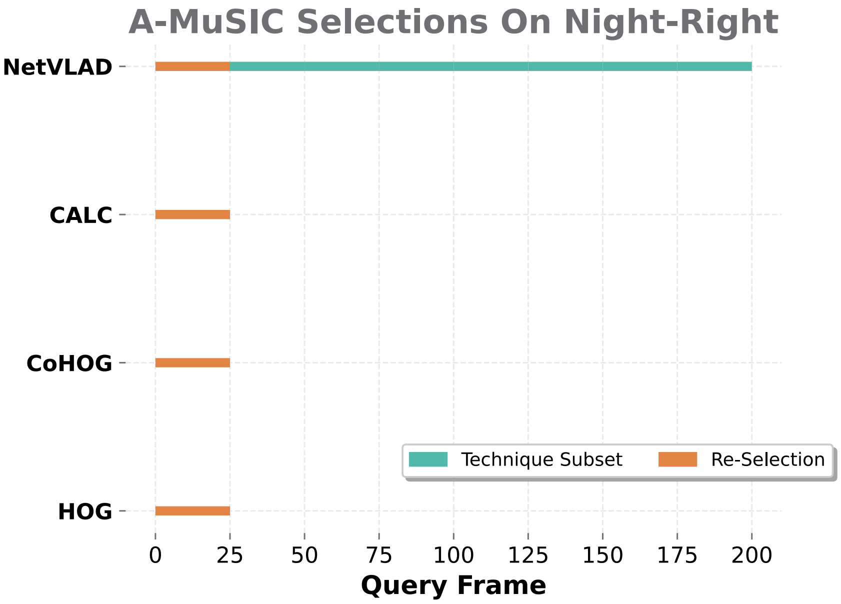
(a)


A-MuSIC Selections On Fall
NetVLAD
CALC
CoHOG
HOG
Technique Subset
Re-Selection
0
200
400
600
800
1000
Query Frame

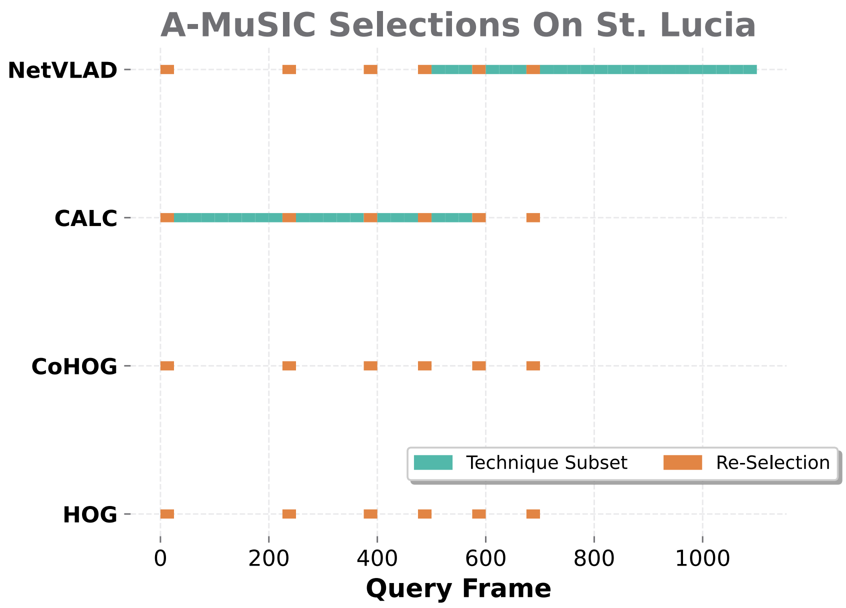
(b)


A-MuSIC Selections On Berlin
NetVLAD
CALC
CoHOG
HOG
Technique Subset
Re-Selection
0
50
100
150
200
250
Query Frame

(c)


A-MuSIC Selections On Night-Right
NetVLAD
CALC
CoHOG
HOG
Technique Subset
Re-Selection
0
25
50
75
100
125
150
175
200
Query Frame

(d)



(e)

Fig. 5.7. A-MuSIC selection patterns. Orange sections denote the frames under which a re-selection stage was issues, while the green frames indicate steady-state operation of the subset of optimal techniques.

servation score vectors produced by a single technique. SIC quantifies the sequential consistency of the top match candidates for the current query frame, resulting in a significant VPR performance improvement. MuSIC employs SIC on a set of VPR methods and, comparing their respective maximum sequential consistencies, selects which to select for performing VPR. Choosing from multiple techniques provides an additional VPR quality increase, allowing better overall performance across different datasets. MuSIC provides an alternative system to current fusion and switching methodologies, with the advantages of not requiring ground-truth information nor brute-force technique combination. Finally, A-MuSIC makes use of the sequential consistency computation produced by SIC, tracking it over a number of frames and using a statistical test to trigger re-selection stages. A-MuSIC trades some VPR performance for significant computational efficiency gains, with nearly three times lower inference time when compared to MuSIC while trumping MPF in VPR performance.

Despite the strong empirical results, several considerations and shortcomings are worth mentioning. The main limitation of the proposed methodology is the sequential navigation assumption. While this assumption holds in many navigation tasks, and an effort has been made to atone it in the SIC algorithm, the system still heavily relies on some degree of sequentiality. Moreover, the adaptive pipeline has clear limitations. Firstly, the match latency benefits are highly unpredictable - some environments might require the use of more techniques, and it is hard to estimate when this might happen. This makes it hard to design the remaining software components needed for navigation, as we can not rely on a frame being processed in constant time. Furthermore, hardware setups must account for the worth case scenario where all techniques are employed, and must be powerful enough to still have a reasonable matching time. Thus, while power usage benefits will still be present, downgrading the hardware isn't a straightforward task. Finally, several hyperparameters are involved these algorithms and choosing appropriate values is non-trivial.

From these shortcomings, it is clear that there is still much work to be done in achieving reliable and efficient VPR with multi-technique setups. In particular, a more sophisticated selection algorithm, which takes into account not only performance but also latency, is highly desirable. Being aware of each technique's cost, i.e. inference time, would also allow the user to specify a maximum acceptable match time, and designing the overall VPR pipeline would become much more feasible.

Chapter 6

Conclusions and Future Directions

Developing robust Visual Place Recognition methods is fundamental to achieve long-term, autonomous navigation. VPR allows a system to localize itself even in areas without network access, unlocking interesting applications in remote or dangerous environments. Indeed, the desire for effective VPR techniques is clearly visible in the research community, with hundreds of new papers on the topic every year.

Nevertheless, a universal VPR technique which excels in every visual challenge and over long-term navigation is yet to be found. The variety of possible visual changes makes VPR an extremely complicated task. Moreover, if VPR is to be deployed in mobile robotics, one must make extra considerations to the computational efficiency when designing VPR algorithms. While recent years have seen remarkable improvements to pure VPR performance, with the advent of CNNs and recently the ViT, the computational cost of said approaches is far more than what a mobile platform can withstand.

In this thesis, the author addresses the efficiency problem by proposing several lightweight VPR algorithms. The first approach relied on the combination of several bio-inspired models, DrosoNets, via a voting mechanism specifically tailored for VPR. As a neural network classifier, each DrosoNet possesses some amount of randomness from its initialization and training processes, and thus two DrosoNets could reach vastly different outputs for the same query even when trained on the same data. The voting system exploited this observation, combining multiple DrosoNets to offset individual failures. While this initial voting pipeline achieved strong VPR performance relative to its computational demands, its absolute VPR performance was not enough for most practical applications. Thus, the second method proposed in this manuscript, dubbed RegionDrosoNet, aimed to improve VPR performance while retaining a low-computational profile. The core of RegionDrosoNet is a novel multi-DrosoNet training protocol, where each DrosoNet group is trained on a different region of training images, further increasing differentiation between DrosoNets. At inference time, each group likewise only infers on its assigned region, and all outputs are brought together with an improved voting scheme. RegionDrosoNet's improved VPR performance outperforms other lightweight techniques while remaining an extremely fast algorithm. Continuing the exploration of multi-method VPR approaches, this thesis proposes Multi-Technique Sequential Information Consistency (MuSIC). By using sequential information, MuSIC selects the most consistent VPR technique amongst an initial pool, improving VPR performance without relying on additional ground-truth information or brute-force fusion which plague other similar approaches. Despite these advantages, MuSIC presents substantial computational cost, as is common for the multi-technique VPR paradigm. Addressing this shortcoming, the last work in this thesis incorporates an adaptive component into MuSIC. The full pipeline, dubbed A-MuSIC, uses sequential information to select a subset of VPR techniques to remain active, saving valuable computational resources.

6.1 Summary of Contributions & Key Findings

In this thesis, the author presents his research work in the field of Visual Place Recognition, with a focus on designing computationally efficient methods. This section aims to summarize the delivered contributions and extract the key insights the research produced.

- The adoption of deep learning approaches for VPR, such as CNNs and Transformers, has increased the computational demands of state-of-the-art methods. Chapter 3 addresses this shortcoming, providing an extremely efficient VPR pipeline consisting of multiple bio-inspired voting units, each dubbed DrosoNet. DrosoNet shows competitive VPR performance relative to its memory size and inference speed, even when compared to other lightweight VPR techniques. The results obtained by DrosoNet support the claim that bio-inspired algorithms are a worthwhile approach when designing lightweight VPR methods. Rather than relying on complex and costly learned feature extractors, the small image tag produced by DrosoNet is distinctive enough to be recognized by a simple perception layer. This finding casts doubt on the current brute-compute approach to increased VPR performance, motivating the development of simpler and more elegant solutions.

- While DrosoNet achieves a strong ratio of computation requirements to matching accuracy, its performance is not enough for performing reliable VPR. Thus, Chapter 3 also introduced a voting pipeline utilizing multiple DrosoNets in tandem. By taking advantage of the model's intrinsic randomness, the overall pipeline was able to obtain stronger VPR performance while retaining a small computationally footprint. The success of the multi-DrosoNet system brings important implications.

Firstly, it shows how having a small baseline model enables scalability, further incentivizing the design of lightweight algorithms and combining multiple if deemed necessary for the target application. Secondly, it demonstrates how even a moderate amount of baseline model differentiation is enough for substantially increased classification performance.

- Nevertheless, the absolute VPR performance of the voting system from Chapter 3, as well as of many other efficient VPR methods, remains unsatisfactory for many applications. Chapter 4 further improves place retrieval reliability by proposing a novel training and inference pipeline for a multi-DrosoNet system. The approach relies on increasing differentiation between models by using distinct DrosoNets groups on different image regions. The resulting system is significantly more performant than other efficiency-focused techniques, even competing with costly approaches on some datasets. This work highlights the benefits of increased model differentiation in ensemble methods. Even when using the same model architecture (DrosoNet), training on different portions of the same images is enough to substantially increase the encoded information by the overall system.

- Continuing the investigation of multi-technique VPR methods, three major shortcomings arise. Firstly, current switching approaches require ground-truth information of the deployment environment. Secondly, the fusion approach often relies on brute-force combination, potentially lowering performance when there are large performance discrepancies in the underlying technique set. Lastly, all multi-method approaches suffer from severely increased computation, as several techniques are run for each query image. Chapter 5 starts by addressing the first two shortcomings by proposing MuSIC, a VPR system which uses sequential information to select the optimal technique for a given query. Then, addressing the third shortcoming, an adaptive module is added on top of MuSIC, allowing the pipeline (A-MuSIC) to disable unhelpful techniques and save computational resources for downstream tasks. These algorithms bring into discussion the use of sequential information for more than the usual correction erroneous matches. Moreover, it shows how it is possible to achieve some level of efficiency in the realm of multi-technique VPR if some thought is put into the operation method.

6.2 Future Work

This section provides guidance for possible future steps building upon this thesis' work, both from its positive insights and shortcomings.

6.2.1 DrosoNet as a General Image Encoder

Chapter 3 and Chapter 4 introduce and develop the bio-inspired neural network DrosoNet. In these works, DrosoNet is treated as a classifier, requiring specific environment training and a fix number of places for inference. Future work should focus on improving the generalization of DrosoNet, perhaps allowing it to act as an image descriptor. Indeed, one can interpret the binary image tag produced by DrosoNet as an encoded image. Viewing DrosoNet as an image encoding algorithm opens interesting possibilities for the development of better performing techniques. One could, for instance, substitute the image encoding portion of a common Transformer architecture with DrosoNet and investigate whether this allows for a less computationally demanding attention mechanism. If successful, such a paradigm shift would significantly improve the practical deployment of DrosoNet.

6.2.2 Data-Driven Adaptability

The sequential mechanism introduced in Chapter 5 is key for both the technique selection and re-selection trigger of the adaptive module proposed in **??**. The obvious drawback from this approach is the reliance on sequential navigation. Furthermore, A-MuSIC fails to take advantage of easy to obtain information regarding a technique's computational cost: should a technique be included in the selection if it's too expensive to be run? Indeed, more data could be incorporated into the mechanisms presented in these chapters with the goal of making not only improving efficiency but also reducing the sequential assumption.

6.2.3 Determining Online VPR Performance

The fundamental issue being addressed by the sequential consistency metric from 5 is the inability of assessing VPR performance during online navigation. Interestingly, there is a pertinent parallel to this dilemma in the booming field of large language models (LLMs): how to determine when a model is hallucinating? As developments in the broader LLM field inevitably arise, one could implement their findings into the VPR pipeline to help detected when a given technique is underperforming.

6.2.4 Deployment On Embedded Systems

One glaring shortcoming of the work presented in this thesis is the lack of direct testing with embedded system hardware. Indeed, all experiments were conducted on either a relatively powerful and power hungry x86 laptop or desktop processor. While all algorithms were tested under the same conditions, thus allowing for fair comparison, computational bottlenecks vary greatly across different hardware configurations. Therefore, it is not possible to claim that the performance benefits showcased by the methods proposed in this work will translate linearly to extremely low-powered hardware. Hence, the work in this thesis can be substantially improved by deploying the proposed methods using embedded systems and collecting data regarding computational efficiency. Moreover, given their simplicity, custom implementations of these algorithms can be easily designed to maximize computation on a given hardware platform. Such practical implementations would be of great benefit to both this work and to practical VPR research.

6.3 Author's End Note

Visual Place Recognition is clearly an important component of robot navigation. While recent advancements have shown impressive VPR performance gains, many of these methods are incredibly computational intensive and are therefore not suitable for mobile platforms. While the work outlined in this thesis is but a small contribution to the lightweight VPR field, the author hopes that it inspires others to work on designing practical and deployable VPR methods. Moreover, focusing on real-time computation on low-end hardware enables researchers without access to expensive compute tools to contribute to this exciting field.

Bibliography

[1] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7989–7996, 2021.

[2] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník, "Artificial intelligence for long-term robot autonomy: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4023–4030, 2018.

[3] J. L. Bresina, A. K. Jónsson, P. H. Morris, K. Rajan *et al.*, "Activity planning for the mars exploration rovers." in *ICAPS*, vol. 2005, 2005, pp. 40–49.

[4] S. Chien, J. Doubleday, D. R. Thompson, K. L. Wagstaff, J. Bellardo, C. Francis, E. Baumgarten, A. Williams, E. Yee, E. Stanton *et al.*, "Onboard autonomy on the intelligent payload experiment cubesat mission," *Journal of Aerospace Information Systems*, vol. 14, no. 6, pp. 307–315, 2017.

[5] L. Chrpa, J. Pinto, M. A. Ribeiro, F. Py, J. Sousa, and K. Rajan, "On mixed-initiative planning and control for autonomous underwater vehicles," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1685–1690.

[6] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner *et al.*, “The strands project: Long-term autonomy in everyday environments,” *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 146–156, 2017.

[7] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[8] G. Berton, C. Masone, and B. Caputo, “Rethinking visual geo-localization for large-scale applications,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.

[9] G. Berton, G. Trivigno, B. Caputo, and C. Masone, “Eigenplaces: Training viewpoint robust models for visual place recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 11 080–11 090.

[10] S. Garg, T. Fischer, and M. Milford, “Where is your place, visual place recognition?” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 4416–4425, survey Track. [Online]. Available: https://doi.org/10.24963/ijcai.2021/603

[11] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[12] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[13] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, “Kimera: an open-source library for real-time metric-semantic localization and mapping,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1689–1696.

[14] A. D. Ekstrom and E. A. Isham, “Human spatial navigation: Representations across dimensions and scales,” *Current opinion in behavioral sciences*, vol. 17, pp. 84–89, 2017.

[15] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[16] M. Zaffar, S. Garg, M. Milford, J. Kooij, D. Flynn, K. McDonald-Maier, and S. Ehsan, “Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change,” *International Journal of Computer Vision*, pp. 1–39, 2021.

[17] S. Schubert, P. Neubert, S. Garg, M. Milford, and T. Fischer, “Visual place recognition: A tutorial,” *IEEE Robotics & Automation Magazine*, p. 2–16, 2024. [Online]. Available: http://dx.doi.org/10.1109/MRA.2023.3310859

[18] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, “Robust visual robot localization across seasons using network flows,” in *Twenty-eighth AAAI conference on artificial intelligence*, 2014.

[19] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3829–3836.

[20] O. Grainge, M. Milford, I. Bodala, S. D. Ramchurn, and S. Ehsan, "Design space exploration of low-bit quantized neural networks for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5070–5077, 2024.

[21] F. Maffra, Z. Chen, and M. Chli, "tolerant place recognition combining 2d and 3d information for UAV navigation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2542–2549.

[22] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. Milford, and K. D. McDonald-Maier, "Visual place recognition for aerial robotics: Exploring accuracy-computation trade-off for local image descriptors," in *2019 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2019, pp. 103–108.

[23] M. Chancán, L. Hernandez-Nunez, A. Narendra, A. B. Barron, and M. Milford, "A hybrid compact neural architecture for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 993–1000, 2020.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[25] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[26] 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*. IEEE, 2007, pp. 3901–3907.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] E. Stumm, C. Mei, and S. Lacroix, "Probabilistic place recognition with covisibility maps," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4158–4163.

[29] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 2006, pp. 33–40.

[30] S. Ehsan, A. F. Clark, and K. D. McDonald-Maier, "Rapid online analysis of local feature detectors and their complementarity," *Sensors*, vol. 13, no. 8, pp. 10 876–10 907, 2013.

[31] H. J. Kim, E. Dunn, and J.-M. Frahm, "Predicting good features for image geo-localization using per-bundle vlad," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1170–1178.

[32] Sivic and Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings ninth IEEE international conference on computer vision*. IEEE, 2003, pp. 1470–1477.

[33] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3304–3311.

[34] R. Arandjelovic and A. Zisserman, "All about vlad," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.

[35] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3384–3391.

[36] S. Arshad and T.-H. Park, "Svs-vpr: A semantic visual and spatial information-based hierarchical visual place recognition for autonomous navigation in challenging environmental conditions," *Sensors*, vol. 24, no. 3, p. 906, 2024.

[37] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[38] D. Arthur, S. Vassilvitskii *et al.*, "k-means++: The advantages of careful seeding," in *Soda*, vol. 7, 2007, pp. 1027–1035.

[39] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[40] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23–36, 2006.

[41] N. Sünderhauf and P. Protzel, "Brief-gist-closing the loop by simple means," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1234–1241.

[42] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 2010, pp. 778–792.

[43] 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*. IEEE, 2009, pp. 2196–2203.

[44] G. Singh and J. Kosecka, "Visual loop closing using gist descriptors in manhattan world," in *ICRA omnidirectional vision workshop*, 2010, pp. 4042–4047.

[45] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[46] M. Zaffar, S. Ehsan, M. Milford, and K. McDonald-Maier, "CoHOG: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1835–1842, 2020.

[47] C. McManus, B. Upcroft, and P. Newmann, "Scene signatures: Localised and point-less features for localisation," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[48] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.

[49] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[50] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[51] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[54] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 873–880. [Online]. Available: https://doi.org/10.1145/1553374.1553486

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito,

M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[56] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[57] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[58] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*. PMLR, 2014, pp. 647–655.

[59] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich,*

Switzerland, September 6-12, 2014, Proceedings, Part I 13. Springer, 2014, pp. 584–599.

[60] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are one," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 3–10.

[61] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[62] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *arXiv preprint arXiv:1411.1509*, 2014.

[63] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[65] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. Gonzalez-Jimenez, "Training a convolutional neural network for appearance-invariant place recognition," *arXiv preprint arXiv:1505.07428*, 2015.

[66] Y. Hou, H. Zhang, and S. Zhou, "Convolutional neural network-based image representation for visual loop closure detection," in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 2238–2245.

[67] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 4297–4304.

[68] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3223–3230.

[69] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[70] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[71] X. Zhang, Y. Su, and X. Zhu, "Loop closure detection for visual slam systems using convolutional neural network," in *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE, 2017, pp. 1–6.

[72] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[73] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 36–45.

[74] G. Tolias, R. Sicre, and H. Jégou, “Particular object retrieval with integral max-pooling of cnn activations,” in *ICLR 2016-International Conference on Learning Representations*, 2016, pp. 1–12.

[75] F. Radenović, G. Tolias, and O. Chum, “Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 3–20.

[76] X. Zhang, L. Wang, Y. Zhao, and Y. Su, “Graph-based place recognition in image sequences with cnn features,” *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 389–403, 2019.

[77] Y. Kalantidis, C. Mellina, and S. Osindero, “Cross-dimensional weighting for aggregated deep convolutional features,” in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 685–701.

[78] F. Radenović, G. Tolias, and O. Chum, “Fine-tuning cnn image retrieval with no human annotation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[79] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[81] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[82] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," *arXiv preprint arXiv:2104.05704*, 2021.

[83] A. Ali-Bey, B. Chaib-Draa, and P. Giguere, "Mixvpr: Feature mixing for visual place recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2998–3007.

[84] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 141–14 152.

[85] N. V. Keetha, M. Milford, and S. Garg, "A hierarchical dual model of environment- and place-specific utility for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6969–6976, 2021.

[86] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[87] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[88] R. Wang, Y. Shen, W. Zuo, S. Zhou, and N. Zheng, "Transvpr: Transformer-based place recognition with multi-level attention aggregation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 648–13 657.

[89] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. S. Chatterji, A. S. Chen, K. A. Creel, J. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. E. Gillespie, K. Goel, N. D. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. F. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. S. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. P. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. F. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. H. Roohani, C. Ruiz,

J. Ryan, C. R'e, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. P. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. A. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," *ArXiv*, 2021. [Online]. Available: https://crfm.stanford.edu/assets/report.pdf

[90] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "Anyloc: Towards universal visual place recognition," *IEEE Robotics and Automation Letters*, 2023.

[91] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[92] P. Yin, J. Jiao, S. Zhao, L. Xu, G. Huang, H. Choset, S. Scherer, and J. Han, "General place recognition survey: Towards real-world autonomy," *arXiv preprint arXiv:2405.04812*, 2024.

[93] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1180–1187.

[94] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 403–408.

[95] M. J. Milford and G. F. Wyeth, “Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[96] P. Hansen and B. Browning, “Visual place recognition using hmm sequence matching,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4549–4555.

[97] B. Mor, S. Garhwal, and A. Kumar, “A systematic review of hidden markov models and their applications,” *Archives of computational methods in engineering*, vol. 28, pp. 1429–1448, 2021.

[98] D. Bai, C. Wang, B. Zhang, X. Yi, and X. Yang, “Sequence searching with cnn features for robust and fast visual place recognition,” *Computers & Graphics*, vol. 70, pp. 270–280, 2018.

[99] C. Cadena, D. Gálvez-López, J. D. Tardós, and J. Neira, “Robust place recognition with stereo sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 871–885, 2012.

[100] T. Naseer, W. Burgard, and C. Stachniss, “Robust visual localization across seasons,” *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289–302, 2018.

[101] T. Pivoňka and L. Přeučil, “On model-free re-ranking for visual place recognition with deep learned local features,” *IEEE Transactions on Intelligent Vehicles*, pp. 1–12, 2024.

[102] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging deep visual descriptors for hierarchical efficient localization," in *Conference on Robot Learning*. PMLR, 2018, pp. 456–465.

[103] S. Garg, N. Suenderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," *arXiv preprint arXiv:1804.05526*, 2018.

[104] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard, "Semantics-aware visual localization under challenging perceptual conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2614–2620.

[105] S. Garg, A. Jacobson, S. Kumar, and M. Milford, "Improving condition-and environment-invariant place recognition with semantic place categorization," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6863–6870.

[106] S. Hausler and M. Milford, "Hierarchical multi-process fusion for visual place recognition," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3327–3333.

[107] L. G. Camara and L. Přeučil, "Visual place recognition by spatial matching of high-level cnn features," *Robotics and Autonomous Systems*, vol. 133, p. 103625, 2020.

[108] Y. Ge, H. Wang, F. Zhu, R. Zhao, and H. Li, "Self-supervising fine-grained region similarities for large-scale image localization," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 369–386.

[109] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1735–1742.

[110] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[111] T. Weyand, I. Kostrikov, and J. Philbin, "Planet-photo geolocation with convolutional neural networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 37–55.

[112] P. H. Seo, T. Weyand, J. Sim, and B. Han, "Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 536–551.

[113] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons," in *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*. Citeseer, 2013, p. 2013.

[114] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, "Fab-map+ ratslam: Appearance-based slam for multiple times of day," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 3507–3512.

[115] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[116] Z. Chen, F. Maffra, I. Sa, and M. Chli, "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.

[117] F. Maffra, L. Teixeira, Z. Chen, and M. Chli, "Real-time wide-baseline place recognition using depth completion," *IEEE Robotics and Automation Letters*, 2019.

[118] R. Sahdev and J. K. Tsotsos, "Indoor place recognition system for localization of mobile robots," in *2016 13th Conference on computer and robot vision (CRV)*. IEEE, 2016, pp. 53–60.

[119] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[120] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. J. Milford, and K. D. McDonald-Maier, "Exploring performance bounds of visual place recognition using extended precision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1688–1695, April 2020.

[121] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[122] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[123] T. L. Molloy, T. Fischer, M. Milford, and G. N. Nair, “Intelligent reference curation for visual place recognition via bayesian selective fusion,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, p. 588–595, Apr. 2021.

[124] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, and K. McDonald-Maier, “Levelling the playing field: A comprehensive comparison of visual place recognition approaches under changing conditions,” *ICRA 2019 Workshop on Database Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR*, 2019.

[125] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier, “A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes,” *IEEE Transactions on Robotics*, pp. 1–9, 2019.

[126] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, “Appearance-based loop detection from 3d laser data using the normal distributions transform,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 23–28.

[127] A. Ali-bey, B. Chaib-draa, and P. Giguère, “Gsv-cities: Toward appropriate supervised visual place recognition,” *Neurocomputing*, vol. 513, pp. 194–203, 2022.

[128] H. Jin Kim, E. Dunn, and J.-M. Frahm, “Learned contextual feature reweighting for image geo-localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[129] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[130] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[131] D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, “An efficient and lightweight convolutional neural network for remote sensing image scene classification,” *Sensors*, vol. 20, no. 7, p. 1999, 2020.

[132] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[133] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and$<$ 0.5 mb model size,” *arXiv preprint arXiv:1602.07360*, 2016.

[134] Y. LeCun, J. Denker, and S. Solla, “Optimal brain damage,” *Advances in neural information processing systems*, vol. 2, 1989.

[135] B. Hassibi and D. Stork, “Second order derivatives for network pruning: Optimal brain surgeon,” *Advances in neural information processing systems*, vol. 5, 1992.

[136] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, “Rethinking the value of network pruning,” *arXiv preprint arXiv:1810.05270*, 2018.

[137] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-s. Hua, "Quantization networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7308–7316.

[138] Y. Xu, S. Zhang, Y. Qi, J. Guo, W. Lin, and H. Xiong, "Dnq: Dynamic network quantization," *arXiv preprint arXiv:1812.02375*, 2018.

[139] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.

[140] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, p. 661, 2019.

[141] B. Ferrarini, M. J. Milford, K. D. McDonald-Maier, and S. Ehsan, "Binary neural networks for memory-efficient and effective visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2617–2631, 2022.

[142] B. Ferrarini, M. Milford, K. D. McDonald-Maier, and S. Ehsan, "Highly-efficient binary neural networks for visual place recognition," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 5493–5500.

[143] N. Merrill and G. Huang, "Lightweight unsupervised deep loop closure," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[144] M. Zaffar, S. Ehsan, M. Milford, and K. D. McDonald-Maier, “Memorable maps: A framework for re-defining places in visual place recognition,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[145] E. T. Rolls, “Spatial view cells and the representation of place in the primate hippocampus,” *Hippocampus*, vol. 9, no. 4, pp. 467–480, 1999.

[146] J. O’Keefe and J. Dostrovsky, “The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat.” *Brain research*, 1971.

[147] J.-W. Lin, “Artificial neural network related to biological neuron network: a review,” *Advanced Studies in Medical Sciences*, vol. 5, no. 1, pp. 55–62, 2017.

[148] A. Narendra, S. Gourmaud, and J. Zeil, “Mapping the navigational knowledge of individually foraging ants, myrmecia croslandi,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1765, p. 20130683, 2013.

[149] J. Dupeyroux, J. R. Serres, and S. Viollet, “Antbot: A six-legged walking robot able to home like desert ants in outdoor environments,” *Science Robotics*, vol. 4, no. 27, p. eaau0307, 2019.

[150] A. Cope, C. Sabo, E. Yavuz, K. Gurney, J. Marshall, T. Nowotny, and E. Vasilaki, “The green brain project–developing a neuromimetic robotic honeybee,” in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2013, pp. 362–363.

[151] J. Degen, A. Kirbach, L. Reiter, K. Lehmann, P. Norton, M. Storms, M. Koblofsky, S. Winter, P. B. Georgieva, H. Nguyen *et al.*, “Exploratory behaviour of honeybees during orientation flights,” *Animal Behaviour*, vol. 102, pp. 45–57, 2015.

[152] T. A. Ofstad, C. S. Zuker, and M. B. Reiser, “Visual place learning in drosophila melanogaster,” *Nature*, vol. 474, no. 7350, pp. 204–207, 2011.

[153] T. L. Warren, Y. M. Giraldo, and M. H. Dickinson, “Celestial navigation in drosophila,” *Journal of Experimental Biology*, vol. 222, no. Suppl_1, p. jeb186148, 2019.

[154] A. Arleo, “Spatial learning and navigation in neuro-mimetic systems: modeling the rat hippocampus,” Ph.D. dissertation, Verlag nicht ermittelbar, 2000.

[155] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, “Path integration and the neural basis of the’cognitive map’,” *Nature Reviews Neuroscience*, vol. 7, no. 8, pp. 663–678, 2006.

[156] L. M. Giocomo, M.-B. Moser, and E. I. Moser, “Computational models of grid cells,” *Neuron*, vol. 71, no. 4, pp. 589–603, 2011.

[157] S. Dasgupta, C. F. Stevens, and S. Navlakha, “A neural algorithm for a fundamental computing problem,” *Science*, vol. 358, no. 6364, pp. 793–796, 2017.

[158] Z. Chen, S. Lowry, A. Jacobson, M. E. Hasselmo, and M. Milford, “Bio-inspired homogeneous multi-scale place recognition,” *Neural Networks*, vol. 72, pp. 48–61, 2015.

[159] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, “Microstructure of a spatial map in the entorhinal cortex,” *Nature*, vol. 436, no. 7052, pp. 801–806, 2005.

[160] C. Fan, Z. Chen, A. Jacobson, X. Hu, and M. Milford, “Biologically-inspired visual place recognition with adaptive multiple scales,” *Robotics and Autonomous systems*, vol. 96, pp. 224–237, 2017.

[161] P. Yin, A. Abuduweili, S. Zhao, L. Xu, C. Liu, and S. Scherer, “Bioslam: A bioinspired lifelong memory system for general place recognition,” *IEEE Transactions on Robotics*, 2023.

[162] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[163] E. I. Moser, E. Kropff, and M.-B. Moser, “Place cells, grid cells, and the brain’s spatial representation system,” *Annu. Rev. Neurosci.*, vol. 31, pp. 69–89, 2008.

[164] J. Stretton and P. Thompson, “Frontal lobe function in temporal lobe epilepsy,” *Epilepsy research*, vol. 98, no. 1, pp. 1–13, 2012.

[165] P. Neubert, S. Schubert, and P. Protzel, “A neurologically inspired sequence processing model for mobile robot place recognition,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3200–3207, 2019.

[166] J. Hawkins, S. Ahmad, S. Purdy, and A. Lavin, “Biological and machine intelligence (bami),” 2016, initial online release 0.4. [Online]. Available: https://numenta.com/resources/biological-and-machine-intelligence/

[167] S. Ghosh-Dastidar and H. Adeli, “Spiking neural networks,” *International journal of neural systems*, vol. 19, no. 04, pp. 295–308, 2009.

[168] E. P. Frady, G. Orchard, D. Florey, N. Imam, R. Liu, J. Mishra, J. Tse, A. Wild, F. T. Sommer, and M. Davies, "Neuromorphic nearest neighbor search using intel's pohoiki springs," in *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop*, 2020, pp. 1–10.

[169] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[170] S. Hussaini, M. Milford, and T. Fischer, "Spiking neural networks for visual place recognition via weighted neuronal assignments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4094–4101, 2022.

[171] ——, "Ensembles of compact, region-specific & regularized spiking neural networks for scalable place recognition," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4200–4207.

[172] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3791–3798.

[173] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.

[174] M. Waheed, M. J. Milford, K. Mcdonald-Maier, and S. Ehsan, “Improving visual place recognition performance by maximising complementarity,” *IEEE Robotics and Automation Letters*, 2021.

[175] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, K. Alexis, and K. McDonald-Maier, “Are state-of-the-art visual place recognition techniques any good for aerial robotics?” *IEEE ICRA 2019 Workshop on Aerial Robotics*, 2019.

[176] F. Maffra, L. Teixeira, Z. Chen, and M. Chli, “Real-time wide-baseline place recognition using depth completion,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1525–1532, 2019.

[177] A. Jacobson, Z. Chen, and M. Milford, “Autonomous multisensor calibration and closed-loop fusion for slam,” *Journal of Field Robotics*, vol. 32, no. 1, pp. 85–122, 2015.

[178] ——, “Leveraging variable sensor spatial acuity with a homogeneous, multi-scale place recognition framework,” *Biological Cybernetics*, vol. 112, pp. 209–225, 2018.

[179] J. Collier, S. Se, and V. Kotamraju, “Multi-sensor appearance-based place recognition,” in *2013 International Conference on Computer and Robot Vision*. IEEE, 2013, pp. 128–135.

[180] U. Wandinger, “Introduction to lidar,” in *Lidar: range-resolved optical remote sensing of the atmosphere*. Springer, 2005, pp. 1–18.

[181] S. Hausler, A. Jacobson, and M. Milford, “Multi-process fusion: Visual place recognition using multiple image processing methods,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1924–1931, 2019.

[182] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[183] H. Zhang, F. Han, and H. Wang, “Robust multimodal sequence-based loop closure detection via structured sparsity.” in *Robotics: Science and systems*, 2016.

[184] P. Neubert and S. Schubert, “Hyperdimensional computing as a framework for systematic aggregation of image descriptors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 16 938–16 947.

[185] M. Waheed, M. Milford, K. McDonald-Maier, and S. Ehsan, “Switchhit: A probabilistic, complementarity-based switching system for improved visual place recognition in changing environments,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7833–7840.

[186] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011.

[187] M. Waheed, S. Waheed, M. Milford, K. McDonald-Maier, and S. Ehsan, “A complementarity-based switch-fuse system for improved visual place recognition,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 5195–5202.

[188] “GIST MATLAB implementation,” http://people.csail.mit.edu/torralba/code/spatialenvelope/, accessed: 2021-11-06.

[189] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[190] M. Pal, “Random forest classifier for remote sensing classification,” *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[191] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[192] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[193] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *3rd International Conference for Learning Representations, San Diego, 2015*, 2015.

[194] T. K. Kim, “T test as a parametric statistic,” *Korean journal of anesthesiology*, vol. 68, no. 6, pp. 540–546, 2015.