Domain shift-aware Ensemble-based Visual Place Recognition

Msc Thesis

Wouter de Leeuw

Delft University of Technology

TU Delft

Domain shift-aware Ensemble-based Visual Place Recognition

by

Wouter de Leeuw

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday March 16, 2023 at 13:00.

Student number: 4487753
Thesis committee: Dr. ir. J.F.P. Kooij, TU Delft 3mE, supervisor
Dr. ir. Seyran Khademi TU Delft ABE, external member
PhD. ir. Mubariz Zaffar, TU Delft 3mE, daily supervisor

TU Delft

Summary

VPR describes a task where an agent (e.g., a robot) attempts to recognize its current location by comparing the incoming visual data from its sensor(s) (query images), usually a camera, to geotagged reference images. Both query and reference images are described using a feature extractor, and the query descriptor is matched to its closest reference descriptor in the feature space. Within VPR there are many different VPR techniques that have been proposed throughout the years with many different types of architecture and trained on different datasets. With the many test datasets available, there exists no VPR technique that is able to reach state-of-the-art performance on all these datasets. For this reason, existing work has argued it can be beneficial to utilize an ensemble-based method to combine multiple VPR techniques and achieve better VPR performance.

Some of these Ensemble-based methods have already been proposed. These ensemble-methods combine individual VPR techniques and weigh their predictions using these same predictions to give an indication of their confidence. This calculation, however, is strictly based on predictions obtained from applying the VPR techniques on test data at inference time. Generally within VPR research, the dataset that was used to train the VPR technique is often different from the dataset it is tested on. This means there is a domain shift between the training and test data. This domain shift is not taken into account when weighting the predictions of VPR techniques in an ensemble using these existing methods.

In this work, we analyze how this degree of domain shift between train and test data, which can be observed by looking at the relative location of descriptors in the feature space, impacts downstream VPR performance. Intuitively, one would expect better VPR performance in a situation where the degree of train-test domain shift is minimal. Our analysis shows that this is indeed the case. We propose two different methods that utilize this degree of domain shift to calculate the weights given to the VPR techniques in an ensemble.

First, we propose a generative method. Here weights are given to the VPR techniques based on the likelihood that the query sample originated from the same distribution as the training dataset of the technique and is in distribution. This way each individual technique is given a weight.

Secondly, we propose a discriminative method. Here weights are given to the training datasets used to train the techniques in the ensemble. These training datasets are given weights based on relative proximity to a query sample in the feature space, an indicator for the degree of domain-shift between the training dataset and the query sample. all VPR techniques are given the weight corresponding to their training dataset.

We compare these proposed approaches to other ensemble-based baselines and individual VPR techniques. The quantitative results show that our proposed methods generally outperform the ensemble-based baselines and the individual VPR techniques.

We also propose further future work. One of the generative methods still delivers lower performance than could be possible, caused by applying this method to high-dimensionality descriptors. A solution for this issue should lead to higher VPR performance using this method. Additionally, we suggest future work to expand on the datasets used in this research, to strengthen the claims made, and verify that results and trends found to hold up when testing and training using other datasets.

Contents

1

Introduction

Visual Place Recognition (VPR) finds itself in the overlap between the robotics and computer vision communities. It describes a task where an agent (e.g., a robot) attempts to recognize it's current location by comparing the incoming visual data from its sensor(s), usually a camera, to geotagged images in a database [1–3]. While there are other applications such as visual navigation [4] and aerial robotics [5], one of the major applications for VPR is loop closure in SLAM systems [1–3, 6–8]. Here, a VPR system is used to mitigate the cumulative error caused by odometry sensors. By recognizing its place as one that the system previously visited, the error caused by the odometry sensors can be reset and the 'loop' is closed.

A VPR system works as follows: Of both the input image (the query image) and all the database images (reference images) descriptors are obtained by using a feature extractor. This feature extractor functions as a mapping function, mapping from the image space to a feature space, where this feature space is often designed to be robust to viewpoint and appearance (day/night, seasons, etc.) changes. In this feature space, distances are measured between the descriptor of the query image and the descriptors of the reference images. The inverse of the distance between the query descriptor and a reference descriptor is called the *similarity*. The reference image descriptor with the highest similarity value is chosen as the best match and the location of the best-matched image is chosen to be the location of the query image. A visual overview of a VPR system is shown in figure 1.1.

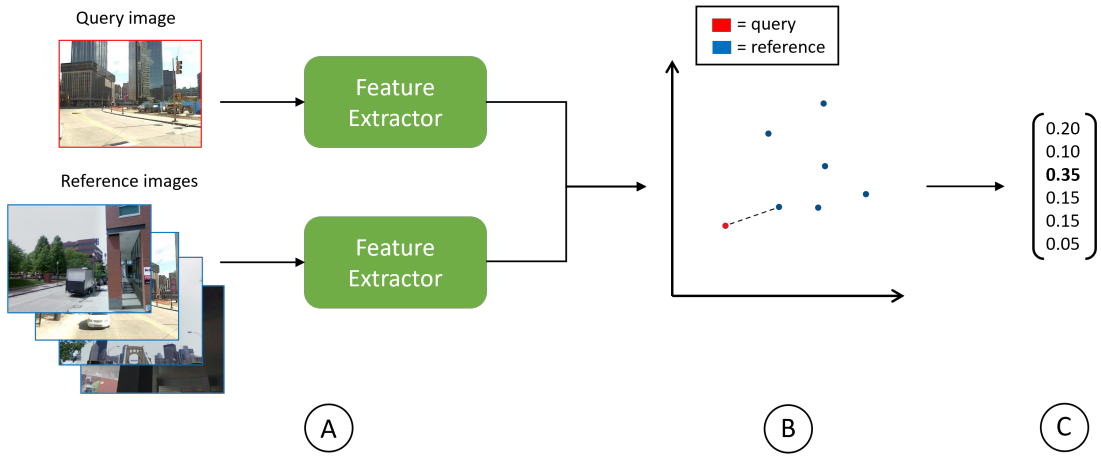

Figure 1.1: An overview of a generic VPR system. Feature extractors extract features from the query and reference images to form image descriptors (A). These descriptors all lie in a feature space (B), where the distance between the query descriptor and reference images can be calculated. The inverse of these distances, the similarities, are stored in a similarity vector (C). This vector contains the similarity score for each reference image. The reference image with the highest similarity score (of 0.35 in this case), and so the lowest distance to the query descriptor in the feature space, is the best match to the query image. The location of the query image is then predicted to be the known location of this reference image.

The field of VPR has produced research proposing many different VPR techniques [8–23] and a multitude of different datasets. Some of this research [7, 24] has focused on comparing existing techniques. These works have shown how there does not exist a single state-of-the-art technique across the different available VPR datasets. This in turn means that there is potential to achieve performance that surpasses any individual technique by applying some ensemble-based method. This leads us to the main research question of this work:

Could an ensemble-based solution lead to increased performance in Visual Place Recognition and if so, what should be the criterion used to combine the techniques in the ensemble?

In order to provide an answer to this question, we raise the following sub-research questions:

1. *Have ensemble methods already been utilized for VPR and what type of combination criteria do they use?*
2. *For data-driven VPR techniques, what is the effect of any train-test domain shift on downstream VPR performance?*
3. *How can the degree of train-test domain shift be utilized to create a criterion for an ensemble method?*
4. *How does our ensemble criterion compare to existing methods on public benchmark datasets and what are the respective limitations?*

Regarding research sub-question 1, existing literature shows that a few ensemble methods for VPR have already been proposed [25–27]. Interestingly, these existing methods all share a similar approach. where the predictions of the VPR techniques in the ensemble are weighted using weights that are calculated based on the similarity scores obtained by applying all techniques. Here similarity is used as a confidence score, meaning if a technique gives a certain reference image a high similarity score, this similarity score is considered equivalent to its confidence in retrieval's success. This is further explained in section 2.

Something else shown by existing research is that more and more focus has been put on proposing data-driven VPR techniques in pursuit of increased robustness [8, 14–16]. An observation that can be made here, is that in a lot of cases, these data-driven techniques are trained and tested on different datasets. What this means, is that a degree of domain shift between the train and test data exists. More explanation on the existing data-driven VPR techniques and the train-test domain shift is given in chapter 2. In this work, we also provide evidence that shows how the degree of this train-test domain shift is an indicator of downstream VPR performance, in order to answer research sub-question 2. The experiment that delivers this evidence is found in chapter 4.

We then propose two methods that utilize the degree of train-test domain shift as a criterion to combine the techniques in the ensemble, to answer research sub-question 3. These methods use probability theory to quantify the degree of train-test domain shift and use the magnitude of this domain shift as weights to weigh the predictions of individual techniques. Doing so results in a criterion that weighs the predictions of techniques more heavily if there is a relatively low degree of domain shift between the corresponding training data and the test data, which should indicate better potential performance. A general problem definition and further explanation of these two methods can be found in chapter 3.

We test our proposed methods by comparing them to a set of baselines. We obtain both quantitative and qualitative results, which we analyze to provide further insights. This way we can answer research sub-question 4. The experiments and their results and analysis can be found in chapter 4.

Finally, we draw our conclusions in chapter 5. There we summarize the answers to the research sub-questions in order to formulate an answer to the main research question. We also go into some of the limitations of our research and identify promising directions for future work.

2

Literature Review

This chapter explores existing literature to find the first answers to the research sub-questions posed in the introduction. We start off by showing how some existing works comparing VPR techniques have made the case for an ensemble-based solution in section 2.1. Then, we explain the few ensemble-based VPR methods that have already been proposed in section 2.2. In section 2.3 we cover some of the existing techniques to show a trend moving towards more data-driven techniques, often leading to testing setups with a train-test domain shift. Finally, in section 2.4 we summarize our literary findings and lay out how our research contributes to the works that already exist.

2.1. Comparing VPR methods

In this subsection, we show how frameworks have been proposed to compare existing VPR techniques since there are many different techniques and datasets. These works show how there is no single state-of-the-art technique.

When there are many different VPR techniques, it is important to have the possibility to compare them to each other to determine which one performs the best under which conditions. VPR-Bench [7] is a framework made available to directly compare VPR methods. It provides the ability to compare 8 different VPR techniques on 12 different datasets. These datasets cover different environments such as cities, nature, indoor, and university campuses under multiple conditions (e.g. day/night and seasonal changes). The results show that there is no single VPR method that is state-of-the-art on all these datasets. For that reason, the authors argue that potential exists for ensemble-based approaches to cover this gap.

Another framework to compare VPR methods is provided by the authors of [24]. They identify an issue in the existing VPR literature: Generally, when VPR methods are compared, they have different architectures but are also trained on different datasets. In this work, they separate these two factors and test different architectures trained on different training datasets on a set of test datasets, mainly focusing on the effects of changing the architecture (using different backbones, aggregators, etc.). One of the conclusions they draw is that there exists no single technique that is the best performer across their test datasets.

By showing how there is no state-of-the-art VPR technique, these works make a case for an ensemble-based solution, which could combine the strengths of different techniques.

2.2. Existing Ensemble-based approaches

The previous subsection showed how some comparison frameworks made the case for ensemble-based VPR. Since then, some ensemble-methods have been proposed. In this subsection, we will explain the workings of these few ensemble-based solutions.

In the work of [27] a multi-process fusion method is proposed that includes four existing VPR techniques, essentially making it an ensemble-based method. They framed using different feature extractors on the data from one sensor (camera) as an alternative to a sensor fusion setup. They used SAD

[28, 29], HOG [10], CNN with max-pooling [19] and CNN with spatial arg-max pooling [21] as feature extractors. These feature extractors are applied to the input to generate similarity vectors that contain the inverse of the distance between the test query descriptor and the test reference descriptors. These similarity vectors are then used in a Hidden Markov Model to provide the final predictions. They show an increase in performance compared to using any of these feature extractors individually.

The authors of [25] propose a hierarchical method. This method consists of multiple layers, where in the first layer a subset of the techniques in the ensemble is applied to the test query image and test reference images to generate similarity vectors. These similarity vectors are averaged and the top N images in the vector with the highest similarity scores are used as input in the next layer. In the next layer, another subset of techniques is applied, and once again the average similarity vector is reduced to a smaller number of candidates. This can be done for any number of layers until a final layer is used that provides the final prediction.

The authors of [26] propose another method: dynamic multi-process fusion. Here every *F* test query images, the current test query image is used as a calibration image. To this image, a set of VPR techniques is applied and an optimal pair of techniques is found. They do so by utilizing the degree of perceptual aliasing. Perceptual aliasing describes a situation where multiple locations look so similar to a VPR system, that it has difficulty distinguishing between them. In this work, the amount of perceptual aliasing experienced by combining any pair of VPR techniques from the ensemble is estimated, and the pair that minimizes this estimated perceptual aliasing is chosen. The argument used for this is that less perceptual aliasing leads to fewer false positives and thus higher accuracy. For every image until the next calibration image, this chosen pair of techniques is used for feature extraction. By applying this calibration step, the best pair of techniques can be found up to a frame-by-image basis. By changing the value of *F*, a balance between this adaptability and computational efficiency can be found. The paper shows this method outperforms single methods, as well as the other proposed ensemble-based methods.

This existing literature shows that some ensemble-based VPR methods already exist. Of these methods, the one proposed in [25] stands out, because with this method how heavily a technique is weighed (or more specifically, how high in the hierarchy a technique is placed) is determined a priori. This means it is manually decided beforehand which techniques should weigh more heavily than others. The other two methods algorithmically decide how heavily the prediction of each technique should be weighed. These two share a likeness in how this is done since in both cases this is done using similarity vectors obtained from the test data, so the training data used to train any data-driven techniques in the ensemble is not taken into account.

2.3. Existing VPR Techniques and Train-Test Domain Shift

Since the aim of this work is to find a well-performing criterion for ensemble-based VPR, it is important to know which VPR techniques exist, and how they work. In this subsection, we cover some of these existing techniques to give insight into the state of the VPR research field.

In the early days of VPR, mostly hand-crafted methods were used, including local feature descriptors such as scale-invariant feature transforms (SIFT) [9] and sped-up robust features (SURF) [11]. Histogram of Gradients (HOG) [10], is a global feature descriptor that was also used for the task of VPR.

After hand-crafted methods, early machine-learning methods started to appear with the goal of providing more robust feature descriptors. These used a Bag-of-Words approach to the problem. Such a model describes the features of an image using visual words. Examples of this sort of approach are FAB-MAP [12] and VLAD [13]. These methods can be seen as a bridge between hand-crafted and deep learning methods, as they use hand-crafted features, and apply the learnable bag-of-words model to them.

As the popularity of deep learning and convolutional neural networks started to grow, they were also being applied to the task of VPR since it was believed they could provide feature descriptors that are more robust against appearance changes in the environment and viewpoint [8, 14–16]. The authors of [17] for example, show that CNN-based features outperform hand-crafted features in situations where the illumination changes.

The authors of [16] were the first ones to train a CNN architecture on the task of VPR in an end-to-end manner. They based their architecture on CaffeNet [30] by using the first four layers and applying a fully connected layer at the end. While these layers were pre-trained, they are finetuned on a VPR dataset using triplets. Especially in situations with conditional changes in weather, lighting, and point of view, it strongly outperforms ORB-SLAM[31], a BoW method that was state-of-the-art at the time, on the Nordland [32], Malaga [33], and Alderley [34] datasets.

Another one of these deep learning-based methods is NetVLAD [18]. The authors implement VLAD pooling as a differentiable layer that can be applied to any CNN architecture to aggregate features for VPR. This way, the network could be trained end-to-end on the VPR task while using the VLAD feature aggregator. They showed how this increased performance compared to traditional hand-crafted methods as well as methods such as [8] that use pre-trained networks on the Pittsburgh [35] and Tokyo24/7 [36] datasets. NetVLAD is still considered one of the best-performing VPR methods, as shown in [7]. Of course, the performance achieved using this aggregator is dependent on the backbone it is combined with.

In [19] the lack of large-scale VPR datasets is identified, which are necessary to train deep neural networks properly and get the best and most robust feature descriptor possible. They create a new dataset: SPED. This dataset consists of 2.5 million images obtained from public outdoor cameras. They also propose two new network architectures: AMOSNet and HybridNet. AMOSNet is trained from scratch, while HybridNet is based on CaffeNet[30].

The authors of [20] attempt to provide extra robustness by using a mask-like binary semantic segmentation method. The network learns which parts of the image are salient (in their experiment: the parts displaying buildings) and which parts are not. Features are extracted from the salient and original images and used for VPR. They were the first to utilize the semantic segmentation task for VPR.

A downside of this method is that marking certain semantic classes as salient is a manual prior. The authors of [21] propose using an attention mechanism so the network can learn by itself which parts of the image are salient and which are not. They apply attention masks at three different layers in the CNN network, to utilize attention at multiple feature levels. These attention masks are then combined to form the final mask that shows which parts of the input image are important for place recognition.

In [22] the focus is put on providing robustness against seasonal changes. They treat this as a domain transfer problem, with summer and winter as their two domains. They propose a network architecture that uses Generative Adversarial Networks that translate images between the two domains. The discriminators are then used as feature extractors for VPR.

Some of the other VPR techniques proposed in the existing literature focus on dealing with the limited amount of available data. They do so by utilizing unsupervised learning methods. [37] implement a stacked denoising auto-encoder to learn image features for VPR in an unsupervised manner. They show performance comparable to FAB-MAP 2.0 [38] on the FAB-MAP and TUM datasets, while no manual labeling is required.

In [23] it is noted that this previous method cannot obtain enough condition-invariant features. As a solution, they propose a method using an auto-encoder that is forced to closely reconstruct a HOG feature extractor. They chose HOG, since it is illumination invariant. Additionally, instead of providing noise using pixel dropout as in [37], noise is added by warping images with perspective transformations since such noise is more realistic in VPR scenarios.

Examining the literature surrounding the existing VPR techniques as we have done in this section, shows a clear trend: Over time more research has focused on data-driven VPR techniques in order to provide better robustness. An interesting observation can be made here, as was done by the authors of [24]: In most of the research that covers data-driven VPR techniques, different datasets are used for training and testing. When doing so, a certain degree of domain shift takes place between the training and testing data.

2.4. Summary of the Current State and Contributions

In this chapter, we have shown how the potential for ensemble-based VPR has been identified, and how some ensemble-based solutions have been proposed. These existing ensemble methods all use the test data to form their criterion that is used to combine the techniques in the ensemble.

By examining the existing VPR techniques, we have shown how there is a trend toward more data-driven VPR techniques. These techniques are generally trained and tested on different datasets, leading to situations with a train-test domain shift. The existing ensemble-based VPR solutions do not take this domain shift into account, since only test data is used for their criteria. Because data-driven VPR techniques are becoming more common, it could be beneficial to implement an ensemble-based solution that takes the train-test domain shift of these methods into account, which is what we do in this work.

Following this analysis of the current state of the literature surrounding ensemble-based VPR, we can state the following about our contributions to this field:

- Our *first contribution* is to show that the degree of domain shift between train and test data for a data-driven VPR technique is an indicator of the downstream VPR performance. We do so by performing an experiment that provides proof for this intuitive concept.
- Our *second contribution* is that we propose two methods that utilize this degree of domain shift to form a criterion which is then used to perform ensemble-based VPR. The next chapter will give a formal explanation of the general ensemble-based VPR problem, and provide an in-depth explanation of the two methods we propose.

3

Methodology

This chapter will describe in detail the methodology of the ensemble methods we propose. First, section 3.1 will give a formal description of the general VPR problem and the ensemble-based VPR problem. Section 3.2 describes the shared fundamentals of our two proposed methods. Then, section 3.3 will describe our generative ensemble method, and finally, section 3.4 describes our discriminative ensemble method.

3.1. Problem description

Assuming an application such as loop closure in a SLAM system, the task of VPR is one where a single query image I_q is matched to a set of reference images, denoted by $\mathcal{R}$. The location of the best matching reference image is then predicted to be the location of the query image I_q. Matching the images at inference time is done by first describing these images in a feature space using a VPR technique τ, resulting in a query descriptor $\boldsymbol{f}_q^\tau$, with dimensionality N and a set of reference descriptors $\mathcal{F}_r^\tau = \{\boldsymbol{f}_r^{\tau,1}, \cdots, \boldsymbol{f}_r^{\tau,R}\}$ of size R, where r indicates this is a set of reference descriptors, and the superscript τ indicates it is described in the feature space of technique τ. The normalized distance between $\boldsymbol{f}_q^\tau$ and the descriptors in $\mathcal{F}_r^\tau$ is calculated and the inverse of this normalized distance is considered the similarity (see equation 3.1).

$$\boldsymbol{d}^{\tau,j} = 1 - ||\boldsymbol{f}_q^\tau - \boldsymbol{f}_r^{\tau,j}||_2, \quad \text{where} \quad \tau \in \mathcal{T}, j \in \mathcal{R} \tag{3.1}$$

The similarity values are stored in a similarity vector $\boldsymbol{d}^\tau = [d^{\tau,1}, \cdots, d^{\tau,R}]$ of size R where element $d^{\tau,1}$ describes how similar the first reference descriptor $\boldsymbol{f}_r^{\tau,1}$ is to the query descriptor $\boldsymbol{f}_q^\tau$. The reference descriptor with the highest similarity value (and thus the lowest distance to $\boldsymbol{f}_q^\tau$ in the feature space) is considered the closest match. The location of I_q is predicted to be the location of this reference image. Figure 3.1 gives an overview of a generic VPR system.

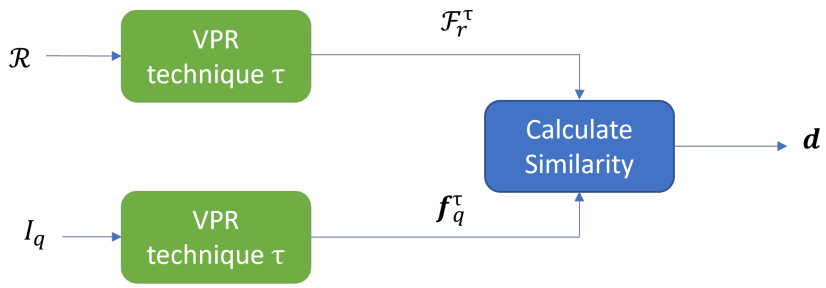

Figure 3.1: An overview of a generic VPR system. A query image I_q and the reference images $\mathcal{R}$ are described using a feature descriptor τ. The similarity vector $\boldsymbol{d}$ is then calculated by comparing the query descriptor f_q^τ with the reference descriptors in $\mathcal{F}_r^\tau$.

When the VPR system used is a data-driven VPR technique, the technique is first trained on a training dataset consisting of a training reference set and a training query set. At inference time, the system is tested on a dataset that might or might not be taken from the same distribution, again consisting of two sets: the test reference set and the test query set. This means there are four sets of data in total. As mentioned in the literature review chapter, the training and test sets are in practice usually taken from different datasets, resulting in a domain shift between the training and test sets.

In the case of ensemble-based VPR, we have a set $\mathcal{T} = \{\tau^1, \cdots, \tau^T\}$ of T VPR techniques. In this work, we refer to specific ensemble-based approaches to VPR as *ensemble methods* and to individual VPR models (e.g. a Resnet-18 [39] backbone with NetVLAD [18] layer trained on a dataset) as a *VPR technique*. An ensemble method prescribes a way of combining the techniques in $\mathcal{T}$, such that the result of the combination yields better performance than any single technique $\tau \in \mathcal{T}$. The techniques in the ensemble can be trained on different datasets. In this work we specifically focus on the reference set of these training datasets. The superset of these different reference sets is denoted as $\boldsymbol{\mathcal{X}}$, with size M where $M \leq T$. Here $\mathcal{X}_1$ refers to the first dataset in this list, and $\mathcal{X}_{tech\ \tau}$ refers specifically to the training reference set used to train technique τ.

Figure 3.2 gives a high-level overview of existing ensemble-based VPR systems. In such a system, all reference images in $\mathcal{R}$ are described using the techniques in $\mathcal{T}$, resulting in a reference set of descriptors $\mathcal{F}_r^\tau = \{f_r^{\tau,1}, \cdots, f_r^{\tau,R}\}$ for every technique τ.

Then for every query image I_q at test time, the query image is described using the techniques in $\mathcal{T}$, resulting in a feature descriptor $\boldsymbol{f}_q^\tau$ for every technique. Then, for each descriptor $\boldsymbol{f}_q^\tau$ of query image I_q the corresponding similarity vector $\boldsymbol{d}^\tau$ is calculated using $\boldsymbol{f}_q^\tau$ and the reference descriptors of the same technique ($\mathcal{F}_r^\tau$), see equation 3.1. This results in a list $\mathcal{D} = \{\boldsymbol{d}^1, \cdots, \boldsymbol{d}^T\}$ of similarity vectors for query image I_q, where each element in this list is a similarity vector $\boldsymbol{d}^\tau = \{\boldsymbol{d}^{\tau,1}, \cdots, \boldsymbol{d}^{\tau,R}\}$ obtained by technique $\tau \in \mathcal{T}$. Finally, the similarity vectors within $\mathcal{D}$ are fused to obtain the final similarity vector $\boldsymbol{d}$. How these similarity vectors are fused depends on the specific method that is used. However, the existing methods do share how this is done fundamentally. The next subsection will cover this in more depth.

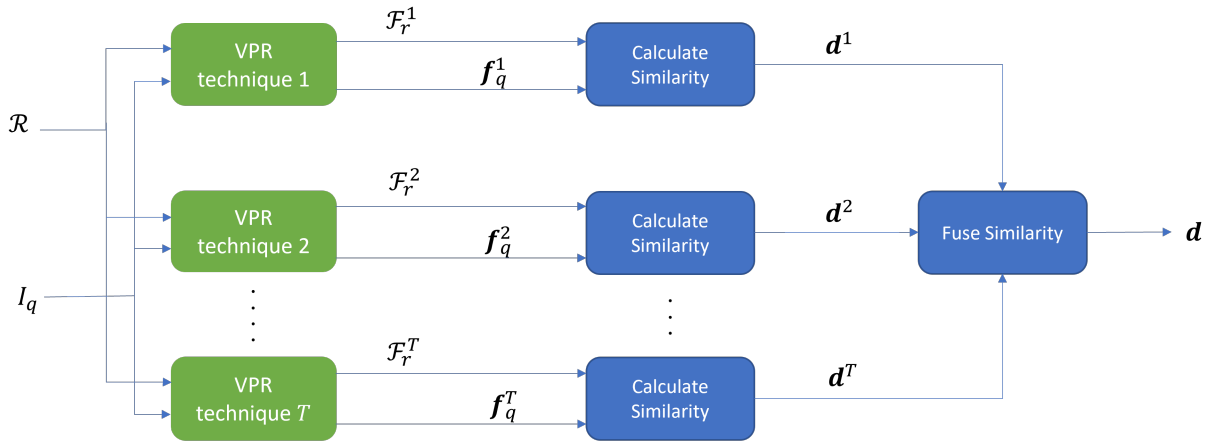

Figure 3.2: A high-level overview of existing ensemble-based VPR methods. Each technique in $\mathcal{T}$ describes the query image and reference images. For each technique, a similarity vector is then calculated and the similarity vectors are fused to form a final similarity vector $\boldsymbol{d}$

3.2. Domain shift-aware approach

In existing ensemble methods, the similarity vectors resulting from the VPR techniques are fused in a manner where each technique is given a certain weight. In the case of these ensemble methods, that weight is calculated using the similarity vectors of these techniques themselves. This is done by interpreting the similarity values as confidence scores [25–27], where a higher similarity score given to the top-scoring reference image means the VPR technique is more confident in its prediction that that reference image is the correct match for the query image. Since these similarity vectors are the result

of a similarity search between the test query and test reference set, this means that this calculation is based only on the test data. As a result, any domain shift between these two test sets is the only domain shift these ensemble methods are influenced by when the fusion of similarity vectors is performed.

As an alternative, we propose two methods that calculate the weights used in the fusion of the similarity vectors using the test query set *and the training reference set of each VPR technique*. The consequence of calculating weights using these sets is that the domain shift between the training and test data has an influence on the calculation of these weights. In practice, different datasets are usually used to train and test VPR methods, so a domain shift between the training and test data is common.

Intuition tells us that any Machine Learning or Deep Learning method achieves better performance if the degree of domain shift between training and test data is minimal or non-existent [40–44]. For this reason, one would expect that the degree of domain shift between the training reference set and the test query set would be an indicator of the downstream performance of a VPR technique. We find supporting evidence for this intuition in section 4.2. There we show that the degree of domain shift can be seen when visualizing the feature space of a VPR technique, where a high degree of train-test domain shift is visible as test and training descriptors lying further away from each other. We also show that in this situation with a high degree of domain shift, the VPR performance achieved is generally lower. Since this degree of domain shift can predict the VPR performance of an individual technique, it could be used to calculate the weight given to the techniques in $\mathcal{T}$ in the case of a query image I_q. We propose two methods that implement the following hypothesis:

A higher weight should be given to the prediction of technique τ if the test query descriptor $\boldsymbol{f}_q$ has a lower degree of domain shift compared to the descriptors of the training reference set $\mathcal{X}_{tech\ \tau}$ of technique τ in the feature space.

The degree of domain shift between a query descriptor $\boldsymbol{f}_q$ and the training reference descriptor set of technique τ, $\mathcal{X}_{tech\ \tau}$, can be described by the probability that the underlying distribution the descriptors of $\mathcal{X}_{tech\ \tau}$ were sampled from is the same as the one the query descriptor $\boldsymbol{f}_q$ was sampled from. The set of descriptors obtained when describing the training reference set $\mathcal{X}_{tech\ \tau}$ in the feature space of VPR technique τ is denoted as $\mathcal{F}^{\tau}_{\mathcal{X}_{tech\ \tau}}$. The underlying distribution of $\mathcal{F}^{\tau}_{\mathcal{X}_{tech\ \tau}}$ is denoted as Z^{τ}, see equation 3.2.

$$\mathcal{F}^{\tau}_{\mathcal{X}_{tech\ \tau}} \sim Z^{\tau} \tag{3.2}$$

The probability of this underlying distribution Z^{τ} also being the one the query descriptor was sampled from is then denoted as $P(Z^{\tau}|\boldsymbol{f}^{\tau}_q)$. If this probability is relatively high, the degree of domain shift between $\mathcal{F}^{\tau}_{\mathcal{X}_{tech\ \tau}}$ and $\boldsymbol{f}^{\tau}_q$ is relatively low and one can expect relatively good VPR performance from technique τ. For this reason, the weight w_{τ} given to technique τ should be proportional to $P(Z^{\tau}|\boldsymbol{f}^{\tau}_q)$ (equation 3.3). This then leads to the calculation of the final similarity vector $\boldsymbol{d}$ by multiplying the similarity vector $\boldsymbol{d}^{\tau}$ of each technique with their corresponding weight w_{τ}, followed by averaging (see equation 3.4).

$$w_{\tau} \propto P(Z^{\tau}|\boldsymbol{f}^{\tau}_q) \tag{3.3}$$

$$\boldsymbol{d} = \sum_{i \in \mathcal{T}}^{T} \boldsymbol{d}^{i} w_i \tag{3.4}$$

The two methods we propose can be related to one another using Bayes' Thereom, as shown in equation 3.5. Here $P(\boldsymbol{f}^{\tau}_q|Z^{\tau})$ describes the probability of observing $\boldsymbol{f}^{\tau}_q$ given that the same underlying distribution generates it as $\mathcal{F}^{\tau}_{\mathcal{X}_{tech\ \tau}}$, $P(Z^{\tau})$ describes the prior probability of $\boldsymbol{f}^{\tau}_q$ being generated by the same underlying distribution as $\mathcal{F}^{\tau}_{\mathcal{X}_{tech\ \tau}}$ and $P(\boldsymbol{f}^{\tau}_q)$ the prior probability of generating $\boldsymbol{f}^{\tau}_q$. Because $\boldsymbol{f}^{\tau}_q$ is obtained single given query image and the ensemble contains a finite amount of techniques, the normalization term can be removed from the equation. Additionally, we assume that the prior probability $P(Z^{\tau})$ of the query sample being sampled from the same underlying distribution as the training data of any specific technique to be the same for each technique, so this term can also be removed from the equation to form equation 3.6.

$$P(Z^{\tau}|\boldsymbol{f}^{\tau}_q) = \frac{P(\boldsymbol{f}^{\tau}_q|Z^{\tau})P(Z^{\tau})}{P(\boldsymbol{f}^{\tau}_q)} \tag{3.5}$$

$$P(Z^\tau|\boldsymbol{f}_q^\tau) \propto P(\boldsymbol{f}_q^\tau|Z^\tau) \tag{3.6}$$

These equations relate our two methods to one another, since the first method we propose is a *generative ensemble-method*, which estimates likelihood $p(\boldsymbol{f}_q^\tau|Z^\tau)$ and uses the normalized value as the weight w_τ. The second method is a *discriminative ensemble-method* that estimates $P(Z^\tau|\boldsymbol{f}_q^\tau)$ directly and uses that value as w_τ. The next two sections will explain the precise workings of these methods.

3.3. Generative ensemble method

In this section, we explain the workings of our first proposed method, the generative ensemble method. This method aims to estimate the probability $P(\boldsymbol{f}_q^\tau|Z^\tau)$, however, since the underlying distribution Z^τ is considered continuous, this method instead estimates the likelihood $p(\boldsymbol{f}_q^\tau|Z^\tau)$ and normalizes it. We first show the concept of this method, after which we give an in-depth explanation.

Figure 3.3 shows a high-level conceptual example of using this generative ensemble method. Here we assume an ensemble of VPR techniques, with all of them trained on either training dataset $\mathcal{X}_A$ or $\mathcal{X}_B$. The left plot shows a query sample $f_q^{\tau_A}$ and the training reference set $\mathcal{X}_A$ described in the feature space of a technique τ_A that was trained on $\mathcal{X}_A$. The plot on the right shows the *same* query sample, as well as training reference set $\mathcal{X}_B$ described in the feature space of technique τ_B trained on $\mathcal{X}_B$. These plots show that the degree of domain shift between the query sample and the descriptors of $\mathcal{X}_A$ is smaller than between the query sample and the descriptors of $\mathcal{X}_B$. For that reason, τ_A should be given a higher weight than τ_B. This is what our generative ensemble method aims to accomplish.

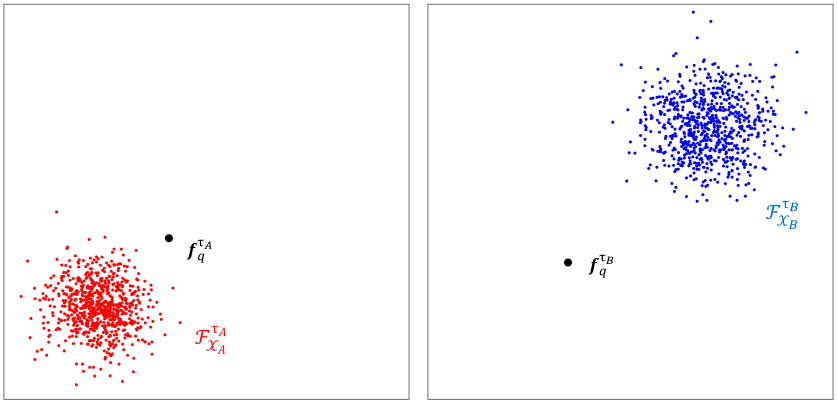

Figure 3.3: Training data and query $\boldsymbol{f}_q$ depicted in the feature spaces of τ_A and t_B respectively

The generative method calculates the weights for each technique τ based on the query descriptor $\boldsymbol{f}_q^\tau$ and the training reference set that was used to train technique τ, denoted as $\mathcal{X}_{tech\ \tau}$. As mentioned before, the training reference set descriptors $\mathcal{F}_{\mathcal{X}_{tech\ \tau}}^\tau$ are sampled from distribution Z^τ.

Since Z^τ is unknown, it can only be estimated. This is done using a probability estimator, where the type of estimator used determines what form the distribution is assumed to take, which is denoted by $\hat{Z}^\tau$. By fitting the estimator on $\mathcal{F}_{\mathcal{X}_{tech\ \tau}}^\tau$, estimated parameters $\hat{\theta}$ are found, giving the estimated distribution $\hat{Z}_{\hat{\theta}}^\tau$ with PDF $p_{\hat{\theta}}$.

$$\hat{\theta} = argmax_\theta \prod_{\boldsymbol{f}_{\mathcal{X},i}^\tau \in \mathcal{F}_{\mathcal{X}_{tech\ \tau}}^\tau} p_{\hat{Z}^\tau}(\boldsymbol{f}_{\mathcal{X},i}^\tau|\theta) \tag{3.7}$$

Given query descriptor $\boldsymbol{f}_q^\tau$, a likelihood value $\mathcal{L}_q^\tau$ can be obtained, which describes the likelihood that the estimated parameters $\hat{\theta}$ and thus distribution $\hat{Z}_{\hat{\theta}}^\tau$ describe the distribution $\boldsymbol{f}_q^\tau$ originated from (see equation 3.8). If the test query descriptor lies relatively close to the training reference set in the feature space and thus is more likely to be in-distribution for this set, the likelihood value $\mathcal{L}_q^\tau$ will be

higher. Here $\mathcal{L}_q^\tau$ is an approximation of the true likelihood value $p(\boldsymbol{f}_q|Z^\tau)$, where p is the true PDF of Z^τ.

$$\mathcal{L}_q^\tau = p_{\hat{\theta}}\left(\boldsymbol{f}_q^\tau|\hat{\theta}\right) \approx p(\boldsymbol{f}_q|Z^\tau) \tag{3.8}$$

Because a certain likelihood value does not mean the same across different feature spaces, the likelihood $\mathcal{L}_q^\tau$ should be normalized. This is done by mapping its value from its original range between 0 and the maximum possible likelihood value $\mathcal{L}_{max}^\tau$ to a range between 0 and 1. Theoretically, $\mathcal{L}_{max}^\tau$ can be found by finding the derivative of the PDF $p_{\hat{\theta}}$ with the estimated parameters and using that to find its maximum. Alternatively, this derivative can be estimated [45–48], and the maximum can be found that way.

However, to stay within the scope of this project, we employ a rather simple method that does not use the (estimated) derivative of $p_{\hat{\theta}}$ to estimate the maximum possible value of the likelihood under this PDF. We estimate the maximum likelihood $\mathcal{L}_{max}^\tau$ in the following manner: we make the assumption that $\mathcal{L}_{max}^\tau$ can be found in or around the center of the cluster of the training data descriptors $\mathcal{F}_{\mathcal{X}_{tech\ \tau}}^\tau$. We make this assumption based on the T-SNE plots as shown in chapter 4.2, where the descriptors of individual datasets take the form of clear clusters that quite closely resemble Gaussian or at least relatively symmetric distributions. We take the mean $\overline{\boldsymbol{f}_{\mathcal{X}}^\tau}$ of all the training reference set descriptors and calculate its likelihood value $\mathcal{L}_{\overline{\boldsymbol{f}_{\mathcal{X}}^\tau}}^\tau$. This value is then assumed to be a close enough approximation of $\mathcal{L}_{max}^\tau$.

$$\overline{\boldsymbol{f}_x^\tau} = \frac{1}{R}\sum_i^R \boldsymbol{f}_{\mathcal{X},i}^\tau \tag{3.9}$$

$$\mathcal{L}_{max}^\tau \approx \mathcal{L}_{\overline{\boldsymbol{f}_{\mathcal{X}}^\tau}}^\tau = p_{\hat{\theta}}\left(\overline{\boldsymbol{f}_x^\tau}|\hat{\theta}\right) \tag{3.10}$$

Then, linear conversion is used to map the likelihood value of $\boldsymbol{f}_q^\tau$ from its original range to one between 0 and 1. Since the lower bound is 0 in both ranges, this conversion is equal to the ratio between $\mathcal{L}_q^\tau$ and $\mathcal{L}_{\overline{\boldsymbol{f}_x^\tau}}^\tau$. The resulting value is then used as the weight applied to similarity vector $\boldsymbol{d}t$ of technique τ for query image I_q. Equation 3.11 shows how the linear conversion is used to calculate the weight w_τ given to VPR technique τ in the case of a query sample I_q. Equation 3.12 shows an example of how the weighted similarity vector is obtained for a technique τ.

$$w_\tau = \frac{(\mathcal{L}_q^\tau - \mathcal{L}_{min})(w_{max} - w_{min})}{(\mathcal{L}_{\overline{\boldsymbol{f}_x^\tau}}^\tau - \mathcal{L}_{min})} + w_{min} = \frac{(\mathcal{L}_q^\tau - 0)(1-0)}{(\mathcal{L}_{\overline{\boldsymbol{f}_x^\tau}}^\tau - 0)} + 0 = \frac{\mathcal{L}_q^\tau}{\mathcal{L}_{\overline{\boldsymbol{f}_x^\tau}}^\tau} \tag{3.11}$$

$$\boldsymbol{d}_w^\tau = \boldsymbol{d}^\tau \cdot w_\tau \tag{3.12}$$

Figure 3.4 gives a high-level overview of this generative approach. It shows how for each technique τ both test sets and the training reference set are described, similarity vectors and weights are calculated, and how the weighted similarity vectors ($\boldsymbol{d}_w^\tau$) are averaged to form the final similarity vector $\boldsymbol{d}$.

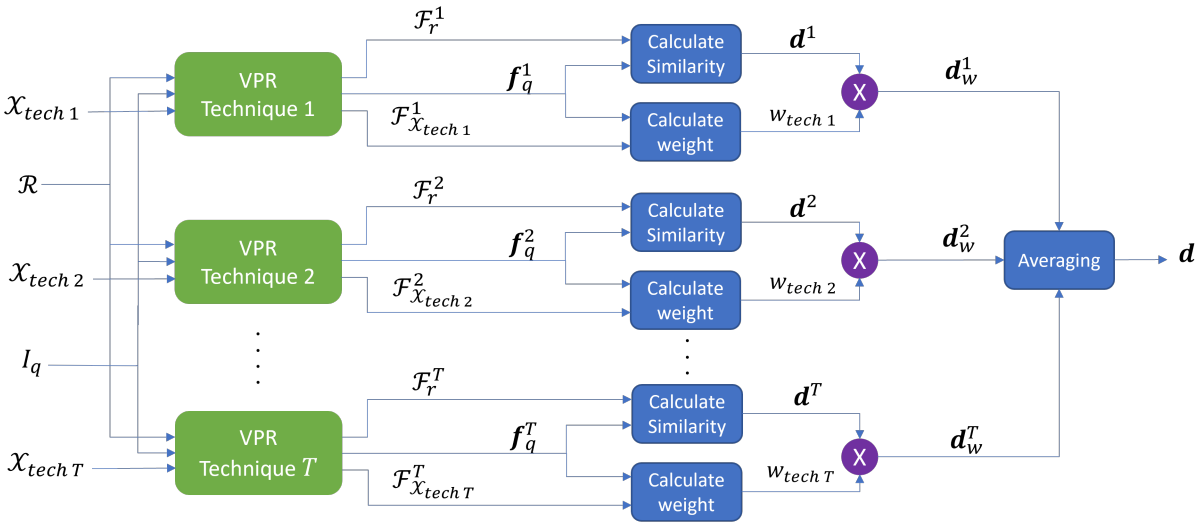

Figure 3.4: A high-level overview of our proposed generative approach. It shows how each technique in $\mathcal{T}$ describes the test query image, test reference images, and the corresponding training reference images. For each technique, the test query and reference descriptors are used to calculate a similarity vector, while the test query and training reference descriptors are used to calculate a weight. The similarity vector of the technique is multiplied with that weight, and the weighted similarity vectors of all the techniques are averaged to form the final similarity vector $\boldsymbol{d}$.

3.4. Discriminative ensemble method

In this section, we explain our discriminative ensemble method. This method aims to estimate the probability $P\left(Z^\tau | \boldsymbol{f}_q^\tau\right)$, which is the probability that given a query descriptor $\boldsymbol{f}_q^\tau$, the underlying distribution the training data descriptors $\mathcal{F}_{\mathcal{X}_{tech\ \tau}}^\tau$ of technique τ were sampled from, is also the distribution $\boldsymbol{f}_q^\tau$ was sampled from. This is done by considering this a classification problem with the training datasets in $\boldsymbol{\mathcal{X}}$ as classes. We first give a conceptual overview of the method, followed by an in-depth explanation.

Figure 3.5 gives a high-level conceptual example of this approach. Here we again assume an ensemble with VPR techniques that are all trained on either training reference set $\mathcal{X}_A$ or $\mathcal{X}_B$. In this figure, both these training reference sets are described in the feature space of a VPR technique τ_A, trained on $\mathcal{X}_A$. We consider this situation as a classification problem, with the task to classify $f_q^{\tau_A}$ as either belonging to $\mathcal{X}_A$ or to $\mathcal{X}_B$. The class probabilities obtained can be used to weigh the predictions of techniques trained on $\mathcal{X}_A$ or to $\mathcal{X}_B$.

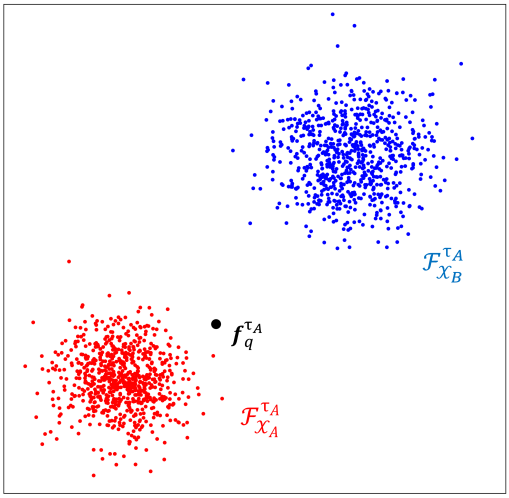

Figure 3.5: The descriptors of the training reference sets $\mathcal{X}_A$ and $\mathcal{X}_B$, and query descriptor $\boldsymbol{f}_q$ depicted in the feature space of a technique τ_A trained on dataset $\mathcal{X}_A$.

More specifically, calculating the weight w_τ of any technique τ is done in the following manner: First, the class probabilities in each feature space as shown in equation 3.14 are obtained. This equation shows an example scenario, where again a setup is assumed with two training reference sets: $\mathcal{X}_A$ and $\mathcal{X}_B$. The underlying distribution of the descriptors $\mathcal{F}^\tau_{\mathcal{X}_A}$ of set $\mathcal{X}_A$ is denoted as Z_A.

The probability of Z_A being the more likely underlying distribution of a query descriptor $\boldsymbol{f}_q$ originated from than Z_B is denoted as $P\left(Z_A|\boldsymbol{f}_q\right)$. In this setup, $P\left(Z_A|\boldsymbol{f}_q\right)$ and $P\left(Z_B|\boldsymbol{f}_q\right)$ sum to one (3.13).

$$P\left(Z_A|\boldsymbol{f}^\tau_q\right) + P\left(Z_B|\boldsymbol{f}^\tau_q\right) = 1 \tag{3.13}$$

Since these true class probabilities are unknown, they are estimated using a classifier. In each feature space, a classifier is trained on $\mathcal{F}^\tau_{\mathcal{X}_A}$ and $\mathcal{F}^\tau_{\mathcal{X}_B}$. At inference time, the query descriptor is classified in each feature space, and the obtained class probabilities are averaged (equation 3.14). By averaging the probability values obtained for $P\left(Z_A|\boldsymbol{f}^\tau_q\right)$ in each feature space for example, the weight $w_{\mathcal{X}_A}$ corresponding to training dataset $\mathcal{X}_A$ can be obtained. This weight is then applied to the prediction of any technique τ_A that was trained on dataset $\mathcal{X}_A$ (equation 3.15). Averaging the probabilities across the feature spaces is done since assuming a situation with no prior knowledge of the test dataset, it is impossible to know which feature space will lead to the estimation of more accurate class probabilities.

$$w_{\mathcal{X}_A} = \frac{1}{T}\sum_{i\in\mathcal{T}}^{T} P^i\left(Z_A|\boldsymbol{f}^i_q\right) \tag{3.14}$$

$$\boldsymbol{d}^{\tau_A}_w = \boldsymbol{d}^{\tau_A} \cdot w_{\mathcal{X}_A} \tag{3.15}$$

Figure 3.6 shows a high-level overview of the proposed discriminative approach. It shows how in the feature space of each technique τ a similarity vector is obtained, and weights for all training reference sets are calculated using a classifier. Then, in the *Averaging weights* block, the weights given in all feature spaces to a training reference set are averaged. This is done for each training reference set, and gives a separate weight for each of them (e.g. $w_{\mathcal{X}_A}$, $w_{\mathcal{X}_B}$). Finally, it shows how the similarity vector of each technique τ is multiplied by the weight $w_{\mathcal{X}_{tech\ \tau}}$ corresponding to the training reference set of τ, after which the weighted similarity vectors are averaged to from final similarity vector $\boldsymbol{d}$.

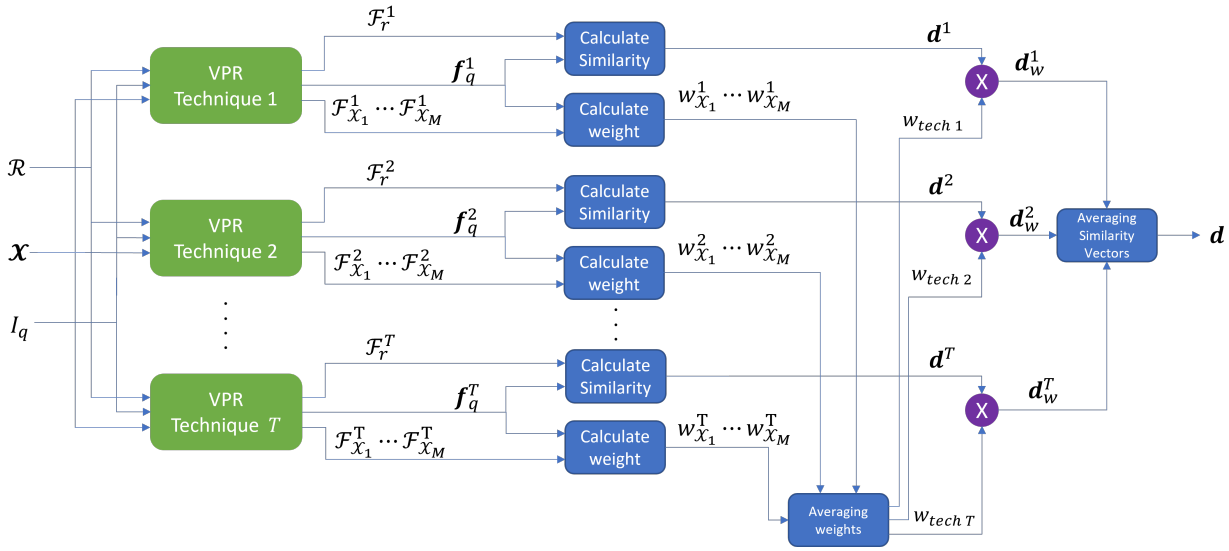

Figure 3.6: A high-level overview of our proposed discriminative approach. M denotes the size of the set $\mathcal{X}$ containing the training reference sets.

4

Experiments and Results

This chapter describes the experiments we performed, their results, and further analysis. First, section 4.1 explains the experimental setup we used to perform the experiments. Next, section 4.2 covers the first experiment, where we find experimental proof for some important intuitions that stand at the base of our methods. Then, section 4.3 shows the results of comparing our methods to certain baselines in a quantitative manner. Then, section 4.4 provides a qualitative analysis of the results achieved by our methods in the prior experiment. Finally, section 4.5 shows how we performed hyperparameter tuning on our methods.

4.1. Experimental setup

This subsection covers different aspects of the setup that was used to perform the experiments evaluated in the next subsections. We go over the used datasets (section 4.1.1), the evaluation metric used (section 4.1.2), tested methods and parametric choices (section 4.1.3), the ensemble that was used (section 4.1.4), the baselines used for comparison (section 4.1.5) and finally, we cover the training of the methods (section 4.1.6).

4.1.1. Datasets

For the experiments executed, we used two different training datasets (Pittsburgh30k [35] and MSLS [49]) and another four (Tokyo24/7 [36], San Francisco [50], St. Lucia [51] and Eynsham [52]) at test time.

- **Pittsburgh30k** is a dataset that contains Google Street View panorama images from the city center of Pittsburgh, USA. While it does not contain any major appearance changes, it does contain viewpoint variations.
- **MSLS** is a large dataset with images from different cities throughout the world, obtained using the Mapillary collaborative mapping platform. It contains both urban and suburban images, with variations in seasons and viewpoints.
- **Tokyo24/7** is a dataset containing city center images of the city of Tokyo. These images were obtained using Google Street View and contain variations in viewpoint. Additionally, there is a shift from day to night conditions between the reference and query sets.
- **San Francisco**, as the name suggests, contains images of the city center of San Francisco. Most images were obtained through a car-mounted panorama camera, these images contain changes in viewpoint. Additionally, some images in the dataset were obtained by mapping mobile phone pictures to a 3D map of the city.
- **St Lucia** is a dataset with images obtained in the St. Lucia suburb in Brisbane, Australia. These images were obtained using a dash-mounted camera in car. This dataset contains no conditional or viewpoint changes.
- **Eynsham** contains greyscaled images of Oxford city and countryside. These were obtained by a dash-mounted camera.

Table 4.1 shows an overview of the characteristics of these datasets, while figure 4.1 shows some representative samples.

Table 4.1: An overview of the training and test datasets used in this work and their characteristics. Query sets with an asterisk have seen their size reduced using sub-sampling as explained in subsection 4.1.6.

Dataset name	Test size query/ ref	Train size query/ ref	Environment type	Camera viewpoint	day-night shift	Additional comments
Pittsburgh30k	800*/ 84k	7824/91K	U.S. city center	Variable	no	
MSLS	1100*/ 18.9K	503K/915K	Worldwide city center and suburbs	variable	no	
Tokyo24/7	300/ 76K	-	Tokyo city center	Variable	yes	
San Francisco	600/ 1.1M	-	U.S. city center	Variable, car-mounted	no	
St. Lucia	1.5K/ 1.5K	-	Australian suburb	Dash-cam, front-view	no	
Eynsham	2.4K*/ 24K	-	Oxford city and countryside	Dash-cam, front-view	no	All images greyscaled

4.1.2. Evaluation Metric

In order to evaluate any methods in our experiments, we use the recall@1 score as our performance metric. This metric shows the percentage of cases where the top predicted match for a query image is indeed a correct match according to the ground truth. In our case, a query and reference image are considered a correct match if they lie within 15 meters distance from one another. Since this top prediction is the one that is used when a VPR system is used for localization, it is a good general performance metric to assess VPR performance.

4.1.3. Tested Methods and Parametric Choices

For both the generative and the discriminative ensemble methods we tested two separate instances.

We tested two different probability estimators for the generative approach: a Kernel Density Estimator (KDE) [53, 54] and a Gaussian Mixture Model (GMM). A kernel density estimator is a non-parametric density estimator and works by applying a kernel to training data. Because this is a non-parametric estimator, it can describe any distribution. The smoothness of the resulting PDF estimation is determined by the bandwidth hyperparameter. A GMM provides an estimation of the PDF and its parameters by describing the training data using one or more weighted multivariate Gaussian distribution functions. How many of these are used to describe the training data is given as a hyperparameter beforehand. GMMs are considered semi-parametric.

For the discriminative approach, we tested the use of a K-Nearest Neighbors classifier, as well as a deep neural network classifier. The neural network classifier consists of one or more fully connected layers. Table 4.2 shows the used architectures of the neural network.

Table 4.2: Input - output dimensions of the layers in the neural net classifier, for cases with 1, 2 or 3 layers. N denotes the dimensionality of the descriptors used as input.

	1 Layer Network	2 Layer Network	3 Layer Network
Layer 1	N - 2	N - $\frac{N}{4}$	N - $\frac{N}{2}$
Layer 2		$\frac{N}{4}$ - 2	$\frac{N}{2}$ - $\frac{N}{4}$
Layer 3			$\frac{N}{4}$ - 2

The hyperparameters of these Generative and Discriminative methods were tweaked as follows:

- KDE: The bandwidth between 0.01 and 1.0
- GMM: The number of components between 1 and 3
- KNN: The number of neighbors used between 1 and 10000
- NN: The number of layers in the network, between 1 and 3

(a) Pitts30k

(c) San Francisco

(e) Eynsham

(b) Tokyo 24/7

(d) MSLS

(f) St Lucia

Figure 4.1: Samples of the datasets used for training and testing.

4.1.4. The Ensemble

The ensemble of techniques used throughout this section consists of four individual techniques. An overview of these four techniques is shown in table 4.3. All techniques have either a ResNet-18 [39] or a VGG16 [55] backbone, a GeM [56] aggregator, and were trained on either Pittsburgh or MSLS [24].

Table 4.3: Techniques used in the ensemble of our experiments

technique nr.	Backbone	Aggregator	Training dataset
1	Resnet-18	GeM	Pittsburgh
2	Resnet-18	GeM	MSLS
3	VGG-16	GeM	Pittsburgh
4	VGG-16	GeM	MSLS

4.1.5. Baselines

When comparing the proposed methods to existing methods we do so using the following baselines:

- **Best individual technique oracle**: For each test dataset all the techniques in the ensemble are tested individually. The individual technique that achieves the highest recall@1 score, is noted in the results. Since it is impossible to know which VPR will perform the best beforehand in a practical scenario, this is considered an oracle baseline.
- **Average score**: The average score baseline is the simplest method of performing ensemble-based VPR. Here the similarity vectors resulting from applying the techniques in the ensemble to a query image are averaged to obtain the final similarity vector.
- **Maximum voting**: When maximum voting is used, each individual technique in the ensemble first predicts which reference image is the most likely match for the query image. The reference image that is chosen the most by these techniques (i.e. gets the most 'votes'), is chosen as the final prediction of the ensemble.
- **Maximum score**: For the maximum score approach, the similarity vector of the technique that contains the highest similarity value for its predicted best match is chosen as the final similarity vector.
- **Dynamical MPF [26]**: This method is currently the state-of-the-art method for ensemble-based VPR. Section 2 explains the workings of this method in further detail. For R the value of 2 was chosen after initial experiments showed this value leading to the best general performance.

4.1.6. Sub-sampling and ensemble-method training

In this subsection, we cover how our methods were trained, and how we applied some subsampling for this training.

The ensemble-methods we tested had to be trained on the training reference sets of MSLS and Pittsburgh. due to the size of these sets and computational constraints, we decided to implement a sub-sampling method to reduce the size of these sets, while still training on data that was representative of the original set. This was done as follows:

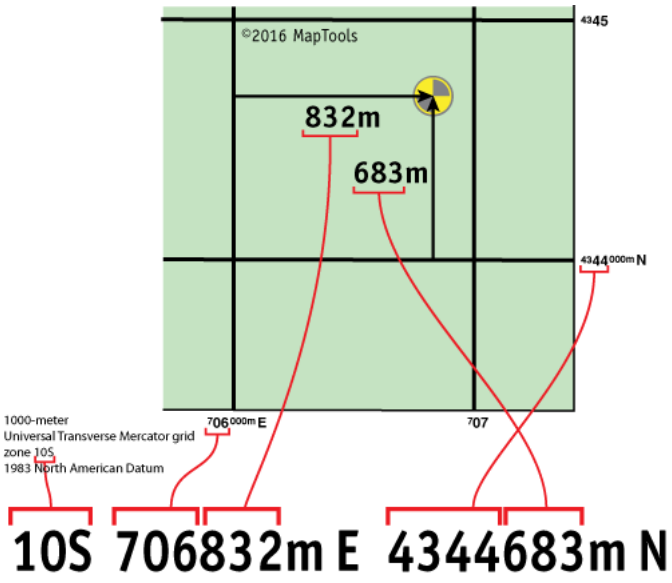

Figure 4.2: An overview of the workings of UTM coordinates [57]. It shows how the coordinates consist of a zone, zone letter, easting, and northing.

The label of each reference image contains UTM coordinates that consist of a zone, zone letter, easting and northing (see figure 4.2). The samples of the original set are first sorted by their UTM zone, a rectangular zone on the world map. Then, the samples of each zone are sorted based on their zone letter, which denotes a horizontal slice (from east to west) within their UTM zone. Finally, the samples of each zone-letter combination are sorted based on their easting (distance from the west border of their zone). By combining the lists of samples of each zone-letter combination again the result is a sorted set of samples, sorted by UTM zone, then zone letter, and then easting. Sub-sampling is

then done by saving one in every J samples. This way, the resulting subset is guaranteed to have a similar distribution over the world map as the original set. The representation of the original sets in their subsets was also evaluated in a qualitative manner for all datasets. For the Pittsburgh and MSLS training reference sets, this method was used to reduce their sample count to around 45,000.

These subsets of Pittsburgh and MSLS were used to train the generative and discriminative ensemble-methods. For the generative methods, an instance was trained in the feature space of each technique on the subset of the training dataset of that technique. This training was done using the built-in SKlearn *fit* method. The KNN discriminative method was trained in a similar manner but using the subsets of both training datasets in each feature space.

The Neural Net discriminative method was also trained in each feature space, using the Pittsburgh and MSLS training reference set subsets. on this combined training data a train/test split of 0.8 was applied. The Neural Net was trained using Stochastic Gradient Descent with a learning rate of 0.01 and momentum of 0.9. The training was done over 10 epochs, where the model with the best accuracy on the test split was saved.

The test datasets all have varying sizes of their query sets, and initial experiments showed that some larger query sets also lead to unfeasible computational times. For this reason, the query sets of the Eynsham, Pittsburgh and MSLS test datasets were reduced by a factor of 10 using the sub-sampling method explained above. Again qualitative evaluation was performed to make sure the new subset represented the original set properly.

4.2. Relating latent spaces, scene appearances and VPR Performance

Before comparing our methods to baselines, we first look into the intuition behind our research: how visually similar-looking datasets end up closer to one another in the feature space of any VPR technique, and how this closeness subsequently has a positive effect on the downstream VPR performance. To analyze this intuition, we perform two experiments. The first one shows how visually similar-looking images end up close together in the feature space, and the second one shows how the relative locations and closeness of test query and training sets in the feature space can be an indicator of resulting performance. These experiments were performed using the framework put forward in [24].

For this experiment, we will use the six datasets listed in the previous subsection. When looking at the samples of these datasets shown in figure 4.1, it becomes possible to divide these datasets up into two groups based on visual similarity:

- Group 1: **Pittsburgh30k**, **Tokyo24/7** and **San Francisco** all contain images of city centers, obtained through a camera looking at buildings from the street. This results in images containing lots of buildings, with viewpoints similar to each other.
- Group 2: **MSLS**, **St. Lucia** and **Eynsham** all mostly consist of images taken in suburbs, resulting in images with far fewer visible buildings than in the datasets of the other group. These datasets also contain a lot of images obtained through dashcams, resulting in a viewpoint that is similar between these three datasets.

When datasets look similar visually, one would expect the probability of these datasets being in distribution for each other to be higher. This then leads us to hypothesize that the data from datasets within the same group also lie close to each other in the feature space of any VPR method applied to them. In order to visualize these datasets in the feature space we first described them using two networks trained on Pittsburgh and MSLS respectively. Then, we applied a t-SNE algorithm to map the data from the high-dimensional feature space to a two-dimensional representation. This visualization can be found in figure 4.3.

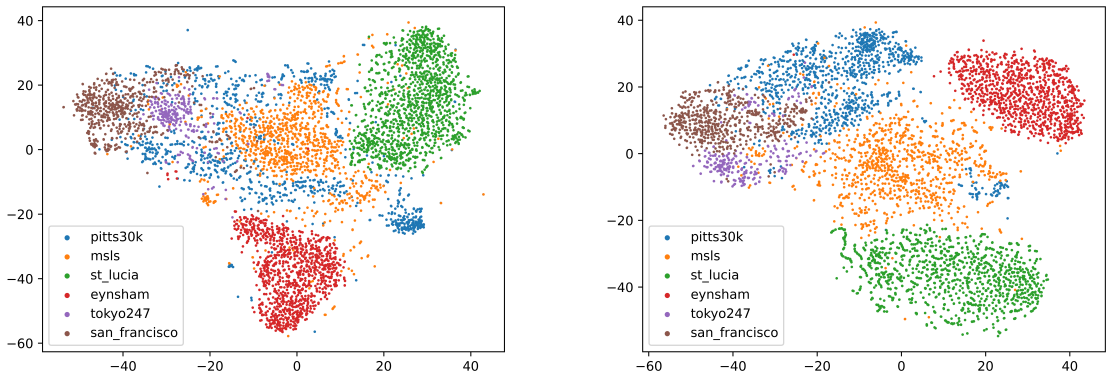

Figure 4.3: Features of the 6 datasets in [24] plotted in the feature space of VPR techniques trained on Pittsburgh30k (a) and MSLS (b).

In these figures, one can see the clusters of the descriptors of the different datasets. Both figures show similar relative locations of these clusters, albeit mirrored when compared to each other. In both cases, a decision boundary could be drawn between the clusters of the datasets of group 1 and the clusters of the datasets of group 2. Most of the data points of the test datasets lie closer to the cluster of the training dataset that looks more visually similar than the other training data. The descriptors of the San Francisco and Tokyo24/7 datasets form a single cluster with the Pittsburgh descriptors, while the descriptors of the St. Lucia and Eynsham datasets form their own separate clusters, lying closer to the MSLS descriptors. Thus, these figures show how visual similarity to the human eye translates to occupying the same or close locations in the feature space.

Generally, any deep learning or machine learning-based system works best on data that is in distribution, meaning data obtained from the same distribution as the training data [40–44]. The clusters in figure 4.3 represent the distributions of the training and test datasets. One would expect that if the data from a test dataset lies closer in the feature space to one of the training datasets, data from the test dataset is more likely to be in distribution for that specific training dataset, meaning a lower degree of domain shift between the two. As a result, the expected VPR performance should be higher if the VPR system is trained on this training dataset than if it is trained on another training dataset whose descriptors lie farther away from the test data in the feature space. This means that in the case of the six datasets we are using, we expect better performance by a VPR system trained on Pittsburgh when testing on San Francisco or Tokyo24/7, and better performance by a VPR system trained on MSLS when testing on St Lucia or Eynsham.

In order to test this relationship between the closeness of training and test sets and the downstream VPR performance, we performed three sub-experiments. For each sub-experiment VPR techniques trained on MSLS and VPR techniques trained on Pittsburgh were tested. While the used training datasets remained the same across these sub-experiments, other parts of the VPR techniques such as their backbones and aggregators were changed as follows:

- sub-experiment 1, Different backbones: Here a few different backbones (feature extractors) were tested.
- sub-experiment 2, Different Aggregators: After feature extraction by a backbone, these features are aggregated to form the description of the input. Here different aggregation methods were tested and compared.
- sub-experiment 3, Different mining methods: VPR systems are trained using image pairs (a query and a matching reference image), that need to be 'mined' from the training dataset. Here different mining methods are compared.

By changing the architectures in these sub-experiments, we are able to single out the effect of training on different datasets on VPR performance, regardless of the architecture used. For each sub-experiment, we then averaged the recall@1 scores achieved by the VPR techniques trained on MSLS

and the VPR techniques trained on Pittsburgh, as shown in table 4.4. What the results show, is that generally, a VPR technique trained on Pittsburgh performs better when testing on Tokyo24/7 or San Francisco, while a technique trained on MSLS is the better performer when testing on The St. Lucia or Eynsham datasets.

Table 4.4: Average recall@1 when trained on Pittsburgh30k/MSLS as shown in [24]

	Tokyo 24/7	San Francisco	Eynsham	St. Lucia
sub-exp. 1: changing backbone	53.4/**54.9**	**54.1**/46.9	71.6/**84.2**	55.1/**89.2**
sub-exp. 2: changing aggregator	**58.6**/58.4	**56.8**/47.5	73.7/**86.0**	60.6/**92.9**
sub-exp. 3: changing mining method	**46.7**/44.3	**44.9**/33.8	69.4/**80.4**	50.2/**81.5**

From this we can conclude that our analysis shows in two steps:

- Descriptors from datasets that look visually similar are located close together in the feature space of a VPR system.
- If the test query descriptor is closer to the training reference set of a VPR system in its feature space, better performance can be expected.

4.3. Comparing proposed approaches to baseline performance

In order to compare the performance of the proposed approaches we compare them to a set of baseline techniques. The results of this comparison can be found in table 4.5. When looking at this comparison between the individual method, ensemble-method baselines, and our proposed approaches certain observations can be made, which we will cover in this section.

Firstly, it shows that on all the test datasets performance can increase significantly when using an ensemble-method over any of the single VPR techniques. This means using an ensemble-method does not only increase average performance across multiple datasets but there exists a case of complementarity when combining VPR techniques that can lead to higher single-dataset performance as well. This can be explained by considering the similarity score given by a VPR technique to the closest matching reference image for a query image as an indicator of the confidence of the technique's prediction (e.g. if the highest similarity score given to any reference image is 0.8, the confidence in the prediction is higher than if the highest similarity score given is 0.4). If a technique is confident in its prediction for a certain query image, the difference in the similarity score of the closest matching reference image and the similarity scores of the other reference images will be relatively large. If a technique is less confident in its prediction, the similarity scores given to the best matching reference image will be closer to the score given to the other reference images. As a result, when combining the similarity vectors of these two techniques, the prediction of the more confident technique for a specific query image will be more prominently reflected in the resulting final similarity vector. This principle is what results in the complementarity shown by the ensemble-methods.

Secondly, It shows that when averaging performance across the four test datasets our proposed approaches outperform the Best individual method oracle, and are also able to perform to the same level as, and in the case of some datasets, slightly outperform all ensemble-method baselines. Additionally, when evaluating performance per individual dataset, one of our methods is always the best-performing method for each test dataset. This shows how utilizing the degree of domain shift between train and test data for an ensemble-method can lead to better VPR performance.

Table 4.5: Comparing recall@1 of our approaches and baselines.

	Pitts30k	MSLS	St Lucia	Eynsham	Tokyo24/7	San Fran-cisco	Average
Best individual method oracle	77.4	68.8	84.4	85.7	41.9	39.0	66.2
ResNet-18 - GeM- Pittsburgh	75.5	36.6	46.3	62.2	33.3	36.1	48.3
ResNet-18 - GeM - MSLS	66.7	67.4	84.4	85.7	37.8	28.3	61.7
VGG-16 - GeM - Pittsburgh	77.4	43.6	46.0	73.1	35.6	39.0	52.4
VGG-16 - GeM - MSLS	65.9	68.8	80.7	85.3	41.9	32.6	62.5
Average Score	**83.6**	65.6	80.0	88.4	49.5	54.7	70.3
Maximum Voting	79.3	56.9	66.2	80.7	38.4	39.8	60.2
Maximum Score	77.2	60.1	69.9	82.4	41.6	37.8	61.5
Dynamical MPF [26]	83.3	67.4	81.4	87.8	50.2	47.2	69.6
Ours, KDE (bandwidth=0.5)	**83.6**	65.6	80.0	88.4	49.5	54.7	70.3
Ours, GMM (n=3)	83.3	66.8	81.2	88.8	49.2	52.7	70.3
Ours, KNN (n_neighbors=10000)	83.3	69.5	**85.7**	88.5	47.6	**55**	71.6
Ours, NN (2 layers)	81.4	**71.4**	82.3	**89.0**	**53.0**	53.2	**71.7**

4.4. Qualitative Analysis of achieved performance

In order to further understand the achieved performance of our methods, we evaluate them in a qualitative manner. For this, we have created plots with training reference and test query image descriptors, with their dimensionality reduced using a TSNE algorithm. This was done for all of our own tested methods, on all the test datasets, with plots created for the feature space of each technique in the ensemble. Analyzing these plots leads to further insight into the results achieved when assessing our methods. In this section, we cover these insights and show plots that give examples of them. First, we make a general observation in subsection 4.4.1. Then, we analyze the results of the discriminative method in subsection 4.4.2, followed by an analysis of the results of the generative ensemble method in subsection 4.4.3. Finally, we summarize our main findings of the qualitative analysis at the end of this section.

4.4.1. General Observation

A general observation we make is that the query descriptors of the test datasets lie closer to the reference descriptors of the training dataset that is more visually similar in the feature space when testing our methods, which is in line with the intuition proven and plots generated in section 4.2. This behavior can be observed across the datasets and methods, with examples given in figure 4.4. Here it shows how the descriptors of the test query set of St. Lucia lie closer to the MSLS training reference descriptors in the feature space than to the training reference descriptors of Pittsburgh. Looking at images of these datasets, it is clear that St. Lucia is indeed also more visually similar to MSLS than to Pittsburgh.

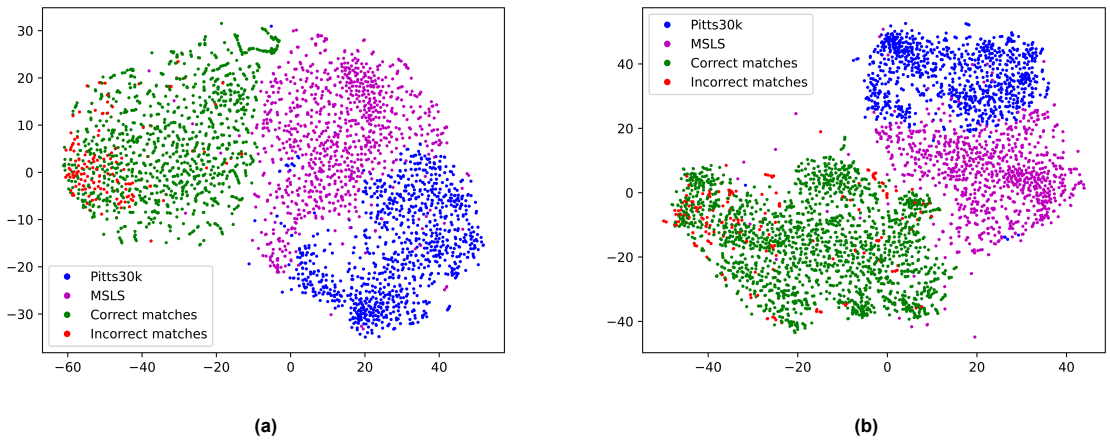

Figure 4.4: Correct/incorrect matches on the St. Lucia (4.4a) and Eynsham (4.4b) datasets using the discriminative method with KNN (n=1), displayed in the feature space of a Resnet-18-GeM VPR technique trained on MSLS. In both cases the test descriptors lie closer to the MSLS dataset (depicted in magenta) than the Pittsburgh dataset (depicted in blue)

4.4.2. Qualitative results of the Discriminative Ensemble method

In this subsection, we will explain and show the observations we made based on the results of the discriminative ensemble method.

The first observation we made for this method, is that when testing on the **St. Lucia** and **Eynsham** datasets, it is clear that the instances of this method follow the expectations. These two test datasets are more visually similar to MSLS than to Pittsburgh, and for that reason, one would expect higher weights to be given to the techniques that are trained on MSLS. Figure 4.5 shows the weights given when testing with the KNN on St. Lucia. As expected, higher weights are given to techniques trained on MSLS. This behavior is shown strongly, with weights given within almost the full range of 0 to 1. This behavior results in the high performance achieved on these datasets by the discriminative method, as can be seen in table 4.3.

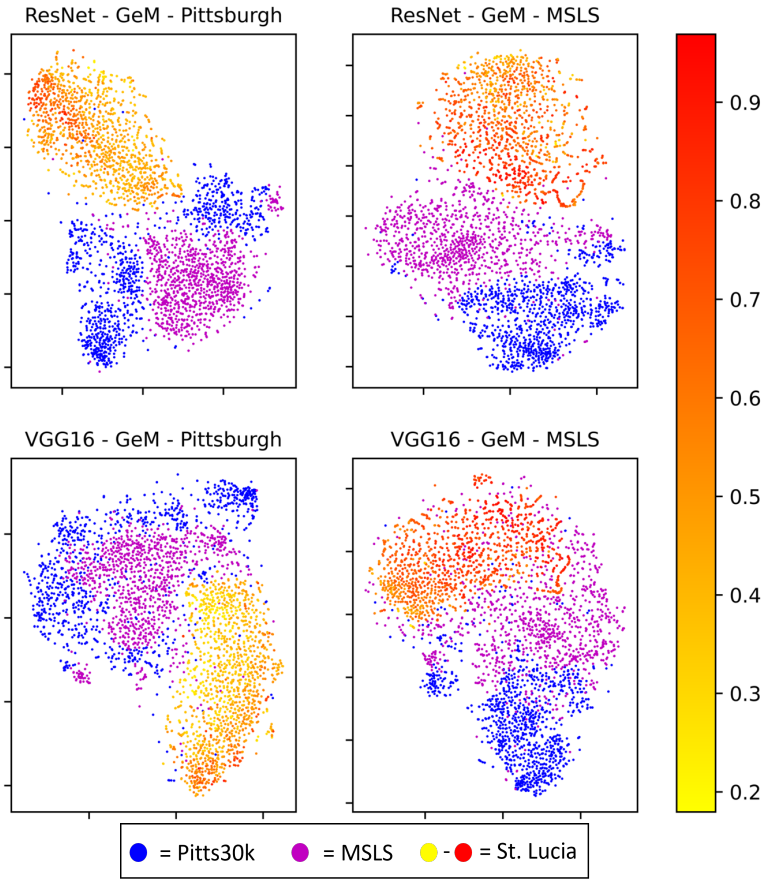

Figure 4.5: Weights given to the techniques when testing the Discriminative method with **KNN** (n_neighbors = 10000) on **St. Lucia**. Each subplot shows the weights given to the corresponding training dataset in that specific feature space. For example, the top-left subplot shows the weights given to the Pittsburgh training dataset for each of the query descriptors in the feature space of the ResNet-GeM-Pittsburgh technique. In the end, the weights given in each feature space are averaged to give the final weights given to the training datasets and their corresponding techniques. Pittsburgh data here is displayed in blue, MSLS data is displayed in magenta. These show how significantly higher weights are given to the techniques that were trained on MSLS. Additionally, A slight gradient in these weights can be observed in the feature spaces. Slightly more even weights are given to these techniques if the query descriptor is relatively far from both training datasets.

Additionally, When testing on the **St. Lucia** and **Eynsham** datasets, the KNN with a higher number of neighbors used and the Neural Net with more layers used show a gradient in weights given in the feature space, depending on the distance to the training data. The plots of these ensemble-methods still show higher weights being given to the VPR techniques trained on MSLS, but also show how

generally slightly more even weights are given to VPR techniques trained on MSLS/Pittsburgh when the test query descriptor is further away from both the MSLS and Pittsburgh reference set descriptors. This can be observed in figure 4.5.

Since these two test datasets are more visually similar to MSLS than to Pittsburgh, it is expected that techniques trained on MSLS are given higher weights than techniques trained on Pittsburgh. The gradient in these plots shows that this is less the case with descriptors that lie further away from both training datasets, which can be explained by the fact that this further distance shows that the degree of domain shift between the query sample and both training datasets is larger for these query samples. This also means that the probability of these descriptors strongly belonging to either training dataset is lower, leading to more even weights being given to the techniques trained on both datasets.

Where on the St. Lucia and Eynsham datasets significantly higher performance is achieved by all techniques trained on MSLS, this is not the case for **Tokyo24/7**. On this dataset, the results achieved by VPR techniques trained on MSLS and Pittsburgh are similar to each other (see table 4.5. For this reason, it is not surprising that the descriptors of this dataset do not clearly lie closer to MSLS or Pittsburgh descriptors, as can be seen in figure 4.6. This figure also shows that relatively even weights are given to techniques trained on Pittsburgh and MSLS because the query samples do not have a significantly higher probability of belonging to either training dataset.

Additionally, it shows that in each feature space, the weights given to the corresponding training dataset generally are not higher than somewhere between 0.5 and 0.6, while very low values are in some cases given. This tells us that according to these methods, any test query descriptor of Tokyo24/7 either has a similar degree of domain shift compared to either training reference set, or specifically has a high degree of domain shift compared to the training reference set that corresponds with the feature space. In the top-left subplot of figure 4.6 for example, every test query descriptor either has a similar degree of domain shift compared to Pittsburgh as compared to MSLS, or a higher degree of domain shift compared to Pittsburgh according to the method. In the feature space of the techniques trained on MSLS this is the other way around. A possible explanation for this is the fact that the Tokyo24/7 test query set consists of mostly nighttime images, which both training datasets do not or barely possess. This means that the Tokyo24/7 test query set indeed has a high degree of domain shift for both training reference sets. The low weights for a training reference set that corresponds to a feature space could then be caused by the method being more certain about th high degree of domain shift for this corresponding training set than for the other training set.

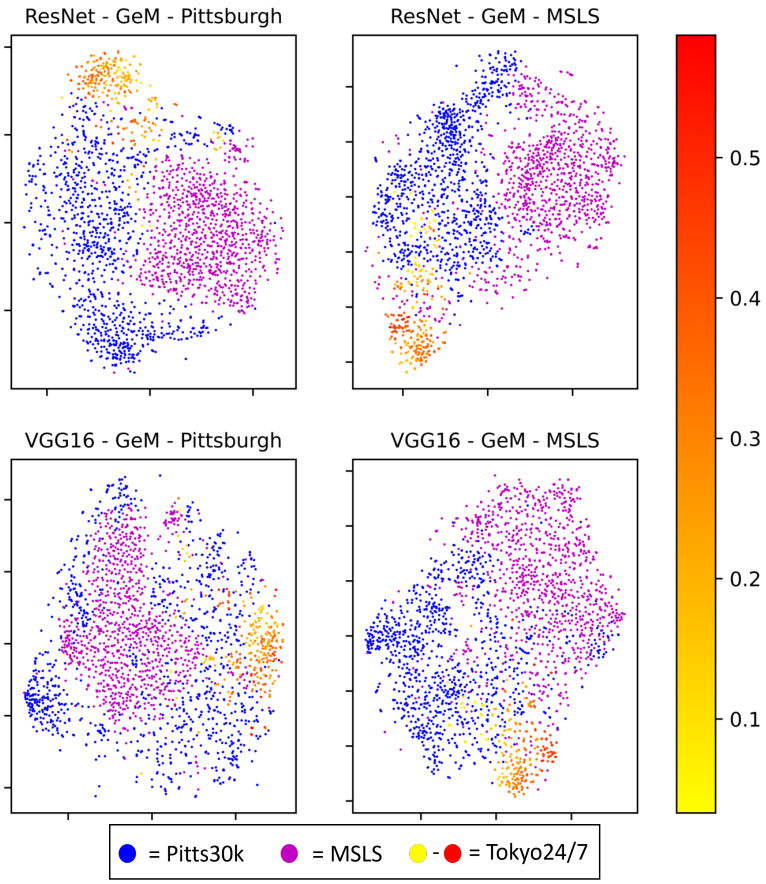

Figure 4.6: Weights given to the techniques when testing the Discriminative method with **KNN** (n_neighbors = 10000) on **Tokyo24/7**, shown in the feature space of all four VPR techniques. Pittsburgh data is displayed in blue, MSLS data is displayed in magenta. Unlike with St. Lucia and Eynsham, the test descriptors do not clearly lie closer to one training dataset than the other. As expected due to the relative location of the test data to the training datasets, no clearly higher weights are given to techniques trained on either Pittsburgh or MSLS.

In contrast to the other test datasets, VPR techniques trained on Pittsburgh outperform VPR techniques trained on MSLS when testing on the **San Francisco** dataset, as shown in table 4.5, although not by large margins. For this reason, one would expect the query descriptors of the San Francisco dataset to lie closer to the Pittsburgh descriptors than the MSLS descriptors, which should then lead to higher weights given to the VPR techniques trained on Pittsburgh. This is indeed what the qualitative results show as can be seen in figure 4.7. Additionally, Just as when testing on Tokyo24/7 we can see that these methods generally do not give weights of a higher value than around 0.5 in any feature space to the training dataset that corresponds to that feature space. Where this could be explained by a shift between day- and nighttime images in the case of Tokyo24/7, but this cannot be done for San Francisco. A possible cause here could be the high degree of variety in the MSLS dataset, which it has since it contains data from many different cities compared to the single city the Pittsburgh data was obtained from. This difference then makes it easier to confidently classify data that is similar to MSLS as such than to confidently classify data that is more similar to Pittsburgh as indeed being more similar to Pittsburgh.

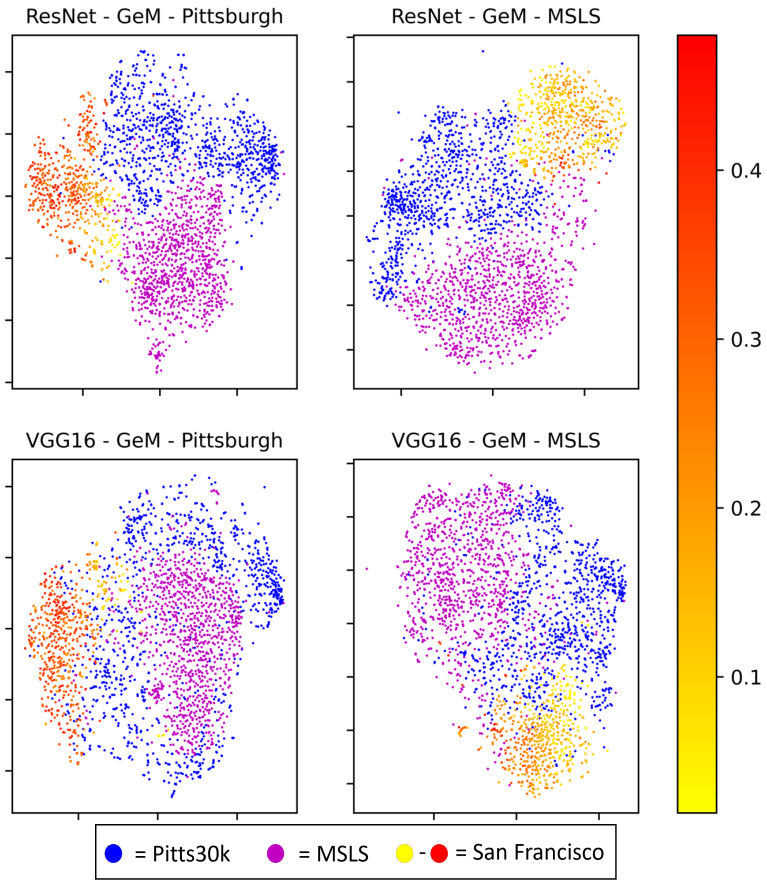

Figure 4.7: Weights given to the techniques when testing the Discriminative method with **KNN** (n_neighbors = 10000) on **San Francisco**, shown in the feature space of all four VPR techniques. In these feature spaces the San Francisco query samples generally lie closer to the Pittsburgh descriptors than the MSLS descriptors. The techniques trained on Pittsburgh are also given higher weights.

4.4.3. Qualitative Results of the Generative Ensemble method

In this subsection, we analyze the performance of the generative ensemble method. We first look at the performance when using the GMM, and then at the performance when using the KDE.

Figure 4.8 shows the weights given to the techniques when testing the generative method with the **GMM** (n=3) on St. Lucia. While the weights limit themselves to a small range, it is visible that higher weights are given to a technique when the query descriptor lies closer to the training data. At the same time, it is not able to match the discriminative methods in performance. This can be attributed to two reasons: Firstly, since weights are calculated per individual technique, techniques trained on MSLS getting a relatively high weight does not guarantee low weights given to the techniques trained on Pittsburgh. Secondly, judging by the quantitative and these qualitative results, the degree of domain shift between the St. Lucia test query set and the Pittsburgh training reference set might be larger than between the St. Lucia test query set and the MSLS training reference set, but the difference is not large enough for this method to give weights that are much lower to techniques trained on Pittsburgh than to techniques trained on MSLS.

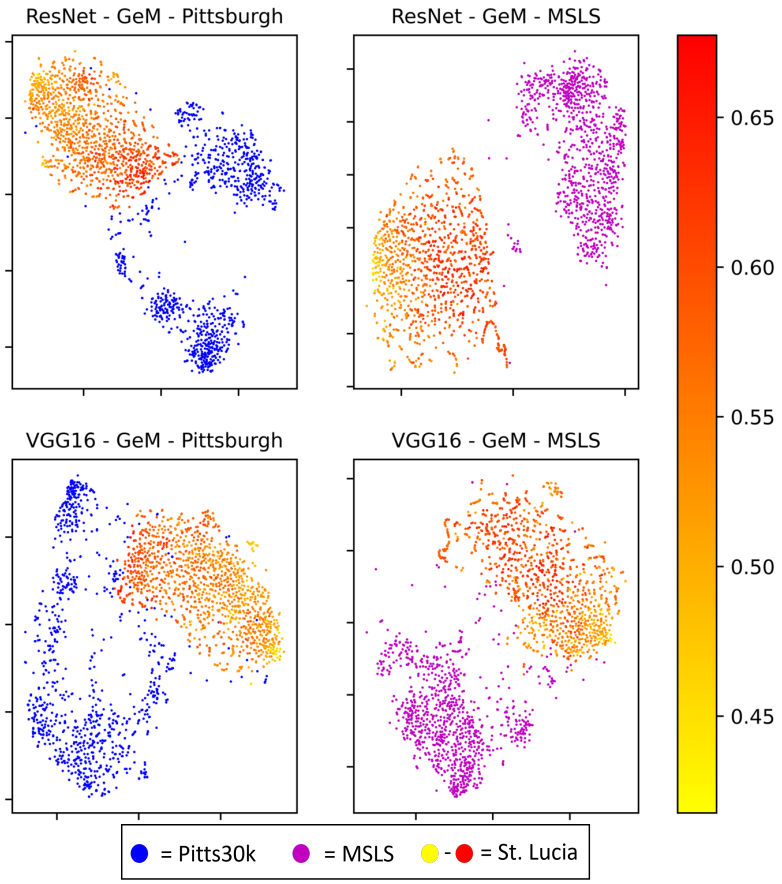

Figure 4.8: Weights given to the techniques when testing the **GMM** (n=3) on **St. Lucia**, shown in the feature space of all four VPR techniques. These show slightly higher weights given to the techniques if the query sample lies closer to the training data in the feature space. Do note the range of the colormap being adjusted in order to make the gradient in weights more visible.

The Generative method with a **KDE** shows certain issues, resulting in performance that is at maximum equal to the average voting baseline (see tables 4.5 and 4.6).

The first issue concerns the weights given to the VPR techniques. In the case of the St. Lucia and Eynsham datasets, for example, one would expect the techniques trained on MSLS to be given a higher weight than the techniques trained on Pittsburgh. However, this is not what the plots show (see figure 4.9). Instead, the weights given to the techniques are relatively equal, independent of their training dataset. Because the weights are so similar for all VPR techniques, the KDE method in practice ends up performing as the average voting baseline. This is also reflected in the quantitative results, where the KDE achieves the same VPR performance as this baseline (see table 4.5).

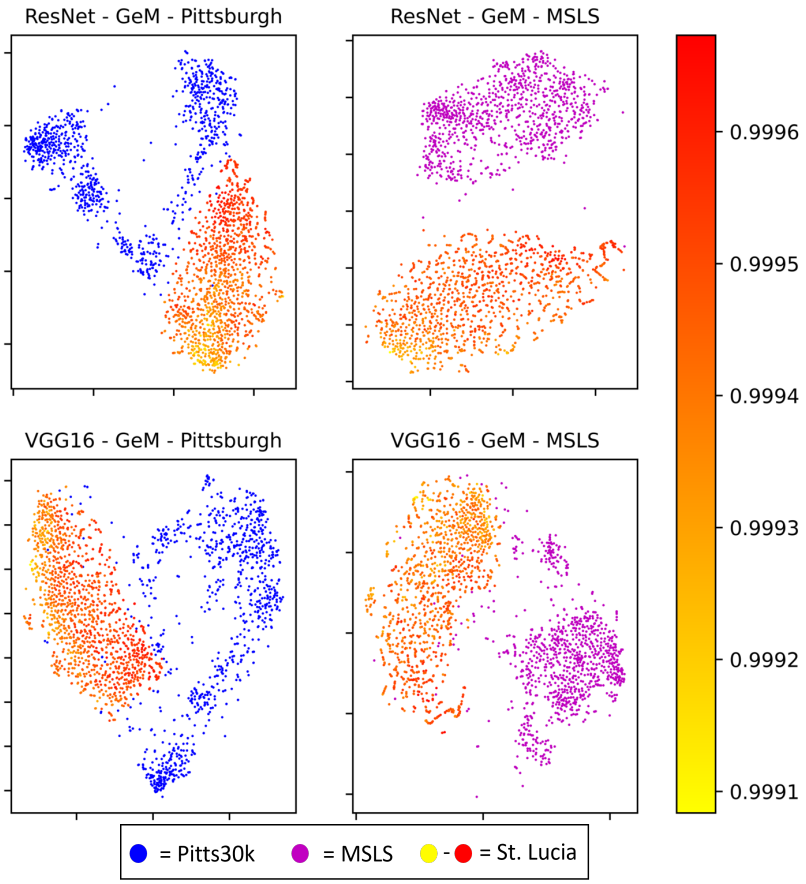

Figure 4.9: Weights given to the techniques when testing the Generative method with **KDE** (bandwidth=0.5) on **St. Lucia**. Pittsburgh data displayed in blue, MSLS data displayed in magenta. Based on the relative location of the train and test set descriptors, higher weights should be expected for the techniques trained on MSLS. Instead, relatively equal weights are given to all the techniques.

A possible explanation for the behavior the generative method with the KDE shows when assigning weights to the techniques is the curse of dimensionality. Existing literature describes that using a KDE in high dimensional space can lead to problems [58–60]. The probability density function becomes very even in this high-dimensional space, and the result is that any point in this space will have a likelihood value that is close to the maximum likelihood value. This leads to high normalized likelihood values for descriptors in the feature spaces of the VPR techniques, leading to relatively even weights for all of them.

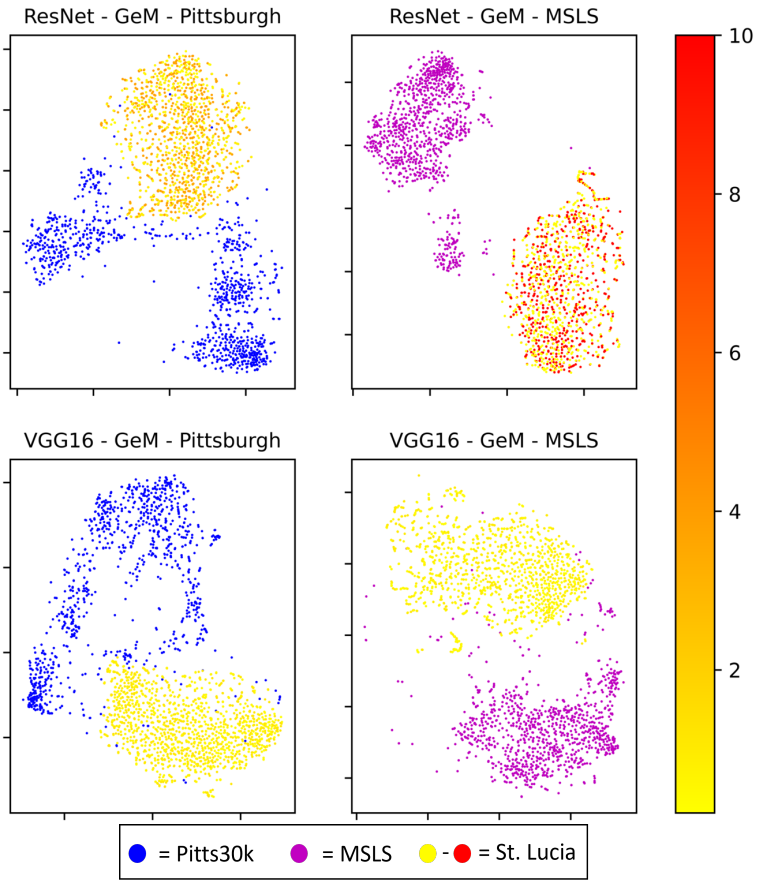

Figure 4.10: Weights given to the techniques when testing the Generative method with **KDE** (bandwidth=0.01) on **St. Lucia**. Pittsburgh data displayed in blue, MSLS data displayed in magenta. These plots show that when using a low bandwidth value, the likelihood values that should be normalized take on values far higher than 1 in some cases.

The second issue with the KDE concerns the normalization of the likelihood. As shown in figure 4.10, when the KDE is used with a small value for its bandwidth parameter, the normalized likelihoods sometimes exceed the value of 1. It turns out that in this situation the small bandwidth results in a PDF that varies a lot in value across the feature space and has steep slopes. As a result, it can happen that the likelihood value of the mean descriptor of the training data is not actually a good estimation of the highest value the PDF can take. This then results in a normalized likelihood that is not properly normalized.

This is not an issue when using the GMM, since it is a model that only contains a limited amount of Gaussian distributions (a maximum of three in our experiments). That means the resulting PDF only contains a few peaks and relatively gradual slopes.

To summarize, the most important findings of this qualitative analysis are as follows:

- If a VPR technique trained on either Pittsburgh or MSLS performs better on a test dataset than a VPR technique trained on the other training dataset, the descriptors of the training dataset that lead to better VPR performance generally lie closer to the test descriptors, which is in line with the analysis of the previous section.
- In the same scenario, the weights given by the discriminative methods tend to follow expectations. Higher weights are given to the techniques trained on the training dataset that is more similar and has a lower degree of domain shift to the test dataset. This behavior is shown most strongly when testing on datasets that are more similar to MSLS than to Pittsburgh. This behavior explains how these methods are able to outperform the baselines as shown in the previous section.

- The GMM generative method also shows expected behavior, with higher weights given to techniques if the test query descriptor is closer to the training reference set. However, the difference in weights between techniques trained on Pittsburgh and MSLS is much smaller than with the Discriminative methods. This could be the result of the weights given to the techniques being independent of one another and the difference in the degree of domain shift between the training datasets and the test data not being large enough to cause a significant difference in weights between the techniques.
- Finally, the generative method with the KDE is only able to perform to the level of the average voting baseline at a maximum, which is caused by two separate issues. Firstly, the relatively high dimensionality of the descriptors combined with most bandwidth values leads to even values for all weights, caused by the PDF as described by the KDE becoming relatively flat across the feature space. Secondly, with low bandwidth values, the likelihood normalization method we employed is not able to accurately estimate the maximum value the PDF can take, leading to badly normalized likelihood values.

4.5. Hyper-parameter tuning

In order to properly compare our methods with the baselines, a hyper-parameter was chosen for each method and tweaked in order to achieve the best possible performance. In this section, we show the effects of tuning these hyperparameters on the VPR performance of the ensemble-methods proposed in this work.

We start by evaluating the Generative approach using the KDE and tweaking the bandwidth. The results of this hyperparameter tuning can be seen in table 4.6. Noticeable is that increasing the bandwidth leads to slightly higher performance. The qualitative analysis of the previous section tells us that this slightly higher performance is caused by the method more closely resembling the average voting baseline.

Table 4.6: Recall@1 using KDE on the test datasets, while tweaking the bandwidth hyper-parameter

Bandwidth	St Lucia	Eynsham	Tokyo24/7	San Francisco	Average
0.01	73.7	86.1	41.3	43.5	61.2
0.05	79.8	88.4	48.6	54.5	67.8
0.1	79.8	88.4	49.5	54.5	68.1
0.3	80.0	88.4	49.5	54.5	68.1
0.5	80.0	88.4	49.5	54.5	68.1
1.0	80.0	88.4	49.5	54.5	68.1

Next, the GMM was tested for the generative approach, and the amount of Gaussians used was tweaked. The results can be seen in table 4.7. It shows that tweaking this hyper-parameter only has a slight impact on performance.

Table 4.7: Recall@1 using GMM on the test datasets, while tweaking the hyperparameter *n*

n	St Lucia	Eynsham	Tokyo24/7	San Francisco	Average
1	79.8	88.5	48.6	52.8	67.4
2	81.1	88.9	49.2	52.3	67.9
3	81.2	88.8	49.2	52.7	68.0

For the Discriminative approach, we first tested using the K-Nearest neighbor method. Tweaking its hyperparameter leads to the results shown in table 4.8. While the changes in recall score for different hyper-parameter values are not that large, differences can be found nonetheless. Interestingly testing on Tokyo24/7 or San Francisco shows an opposite trend compared to the first two test datasets. In the cases of St Lucia and Eynsham, the system delivers slightly higher performance with a lower value for *n_neighbors*, while a higher value leads to better performance on the other two test datasets.

Table 4.8: Recall@1 using KNN on the test datasets, while tweaking the *n_neighbors* hyperparameter

n_neighbors	St Lucia	Eynsham	Tokyo24/7	San Francisco	Average
1	86.5	88.6	45.1	51.0	67.8
10	86.7	88.8	44.4	52.2	68.0
100	86.7	88.8	44.8	52.2	68.1
1000	86.1	88.6	45.4	53.0	68.3
5000	85.9	88.6	47.0	54.5	69
10000	85.7	88.5	47.6	55.0	**69.2**

Finally, we test the Discriminative approach using the Neural Network. Here we tweaked the number of layers in the network. The resulting average recall@1 scores can be found in table 4.9. It shows that the number of layers only has a small impact on performance levels, although a higher amount of layers does lead to a slight increase.

Table 4.9: Recall@1 using NN classifier on the test datasets, while tweaking the number of layers

layers	St Lucia	Eynsham	Tokyo24/7	San Francisco	Average
1	82.4	88.8	47.9	54.7	68.45
2	82.3	89.0	53.0	53.2	69.4
3	76.7	88.6	49.8	54.0	67.3

5

Conclusion and Discussion

In this Chapter, we conclude and discuss the results of our research. In section 5.1 we go over our findings and conclude by giving our answers to the posed research sub-questions. In section 5.2 we discuss the limitations and challenges faced in our work, and propose some possibilities for future work.

5.1. Conclusion

In this work, we set out to answer the main research question by finding answers to sub-questions. The answer to sub-question 1 was found by showing in our literature review that existing work has identified the potential for ensemble-based VPR, and some ensemble methods have already been proposed. These existing methods have an important aspect in common: with all of them, their criterion is obtained from the similarity vectors that result from the test query and reference sets. This means any domain shift between training and test data is not taken into account.

This becomes an important point when answering sub-question 2. By examining existing literature we conclude that a trend can be identified where more research is being done on data-driven VPR techniques. These data-driven techniques are usually trained on one dataset and tested on another. This results in situations where a certain degree of domain shift between the training and test data is present. In the first experiment we perform, we show that this degree of domain shift is an indicator of downstream VPR performance. I.e. if there is a higher degree of domain shift between the data used to train a VPR technique and the test data, one can expect the VPR performance to be worse. For this reason, utilizing this degree of domain shift as the criterion of an ensemble-based method could potentially lead to better performance compared to the existing methods.

We proposed two ensemble methods that do just that, to give an answer to sub-question 3. The first method we propose is a generative method, where the underlying distribution of the training data is estimated, and the likelihood of observing the query sample given the estimated distribution of the training data is used as the weight given to a technique. The second method we propose is a discriminative one, where the weight given to a technique is equal to the estimated probability of the query sample originating from the same underlying distribution as the training data of the technique.

We also performed both quantitative and qualitative experiments to assess the performance of our methods (to answer sub-question 4). We compared two separate instances of each method to some baselines on four different datasets. Some of the key takeaways from these results and our analysis are as follows:

- Our best-performing method instance, the discriminative ensemble method with the Neural Net, was able to achieve an average recall@1 score of 69.4 across the test datasets, compared to 68.2 for the best-performing baseline method.
- When looking at the results on any of the test datasets, an instance of our discriminative ensemble method always achieves the best performance, better than any of the baselines.

- In general, the discriminative ensemble method gave higher weights to techniques trained on the training dataset that had a lower degree of domain shift to the query test set, as expected.
- The generative ensemble method with the GMM estimator generally gives higher weights to techniques if the query descriptor lies closer to the training reference set of the technique according to our qualitative analysis, which follows expectations. However, this effect is not strong enough to match the performance increase shown by the discriminative ensemble methods when compared to the baselines.
- Finally, the generative ensemble method with the KDE ends up functioning very similarly to the average voting baseline as shown by the quantitative and qualitative results. This could be caused by the curse of dimensionality, making it so that any point in the feature space has a likelihood value close to the estimated maximum possible likelihood value, resulting in almost the same weight given to all techniques.

To conclude, our research shows that there is indeed a benefit to utilizing the degree of train-test domain shift as a criterion for ensemble-based VPR with data-driven techniques. Still, not all method instances were able to outperform the baselines. The next subsection will go over some of the limitations of this work and provide some possible avenues for future research.

5.2. Limitations and Future Work

In this subsection, we will go over some of the challenges and limitations faced in this research and put forward how these create possibilities for future work.

Firstly, a tough challenge faced in this research was finding the correct and enough datasets for both training and testing. We settled on testing VPR techniques trained on either the Pittsburgh or MSLS dataset. We divided these two training datasets and the test datasets into two groups, representing images from city centers and suburbs respectively. Where we drew a clear border between these two groups, this distinction is more nuanced in reality. Apart from the difference between city center and suburban data, MSLS is also simply a much larger dataset containing images from a multitude of different cities around the world, whereas Pittsburgh is a much smaller dataset with images exclusively from one city. As a result of this, VPR techniques trained on MSLS generalize a lot better than when trained on Pittsburgh, which becomes clear when looking at the performance levels of the individual techniques (table 4.3). This had an effect on the results of our working methods, where the difference in weights between techniques was significantly larger if the test dataset was more similar to MSLS than if the test dataset was more similar to Pittsburgh. This also meant that the behavior we desired to see (with higher weights being given to techniques trained on data that had a lower degree of domain shift to the test data) was most clear when testing on St. Lucia and Eynsham, where techniques trained on MSLS clearly outperform the techniques trained on Pittsburgh.

Ideally, the datasets used to train the VPR techniques in the ensemble should be of similar size and have a single distinguishable difference between them (e.g. city center/suburban, night/day). That way, the differences in the degree of domain shift between the training and test data would be more defined, and one would expect the behaviors we observed to be even more clearly visible. While we were unable to gather such a specific set of datasets, it would be interesting to see if it could be done in the future, so the claims made here can be further verified and strengthened. This could potentially be done by creating and using synthetic datasets, where the user has more control over the different aspects that form any dataset.

Another challenge posed itself with the qualitative analysis of the acquired results. In order to visually assess behaviors we transformed the high-dimensional descriptors into a two-dimensional space using a TSNE algorithm. When applying dimensionality reduction, some information always gets lost. Especially because TSNE is a non-linear algorithm, the relative distances the transformed descriptors showed in the two-dimensional space are not guaranteed to be the same as they are in their original high-dimensional form. This makes it more difficult to draw conclusions on which sets of data are really closer or farther apart in the high-dimensional feature space. In section 4.2 for example, the quantitative results strongly support the hypothesis posed in that section, and while figure 4.3 also supports it, the relationship is not shown as strongly. Further analysis of these distances in other ways could make behaviors or results clearer in situations where the TSNE visuals only faintly show the expected behavior or result.

Finally, it is important to note a fundamental limit to the methods proposed in this work. As discussed, there are four separate sets of data present in data-driven VPR: the training reference set, training query set, test reference set, and test query set. The methods proposed in this work focus on the domain shift between the test query set and the training reference set. Domain shifts between other pairs of sets are not directly taken into account. The existing ensemble-methods base their weight calculation on the test query and test reference set, and with that implicitly take into account any domain shift between these two sets. Future research could find a way to combine these existing methods with the ones proposed here, to test if doing so could lead to even better VPR performance.

References

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

[2] Xiwu Zhang, Lei Wang, and Yan Su. “Visual place recognition: A survey from deep learning perspective”. In: *Pattern Recognition* 113 (2021), p. 107760. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107760. URL: https://www.sciencedirect.com/science/article/pii/S003132032030563X.

[3] Sourav Garg, Tobias Fischer, and Michael Milford. “Where Is Your Place, Visual Place Recognition?” In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence* (2021). DOI: 10.24963/ijcai.2021/603. URL: http://dx.doi.org/10.24963/ijcai.2021/603.

[4] Marvin Chancán and Michael Milford. *CityLearn: Diverse Real-World Environments for Sample-Efficient Navigation Policy Learning*. 2020. arXiv: 1910.04335 [cs.RO].

[5] Mubariz Zaffar et al. *Are State-of-the-art Visual Place Recognition Techniques any Good for Aerial Robotics?* 2019. arXiv: 1904.07967 [cs.CV].

[6] P. Newman and Kin Ho. “SLAM-Loop Closing with Visually Salient Features”. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 635–642. DOI: 10.1109/ROBOT.2005.1570189.

[7] Mubariz Zaffar et al. “VPR-Bench: An Open-Source Visual Place Recognition Evaluation Framework with Quantifiable Viewpoint and Appearance Change”. In: *International Journal of Computer Vision* 129.7 (2021), pp. 2136–2174. ISSN: 1573-1405. DOI: 10.1007/s11263-021-01469-5. URL: http://dx.doi.org/10.1007/s11263-021-01469-5.

[8] Zetao Chen et al. *Convolutional Neural Network-based Place Recognition*. 2014. arXiv: 1411.1509 [cs.CV].

[9] D.G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints.” In: *International Journal of Computer Vision 60* (2004), pp. 91–110.

[10] 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. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[11] Herbert Bay et al. “Speeded-Up Robust Features (SURF)”. In: *Computer Vision and Image Understanding* 110.3 (2008). Similarity Matching in Computer Vision and Multimedia, pp. 346–359. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2007.09.014. URL: https://www.sciencedirect.com/science/article/pii/S1077314207001555.

[12] Mark Cummins and Paul Newman. “FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance”. In: *The International Journal of Robotics Research* 27.6 (2008), pp. 647–665. DOI: 10.1177/0278364908090961.

[13] Hervé Jégou et al. “Aggregating local descriptors into a compact image representation”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 3304–3311. DOI: 10.1109/CVPR.2010.5540039.

[14] Manuel Lopez-Antequera et al. “Appearance-invariant place recognition by discriminatively training a convolutional neural network”. In: *Pattern Recognition Letters* 92 (2017), pp. 89–95. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2017.04.017. URL: https://www.sciencedirect.com/science/article/pii/S0167865517301381.

[15] Niko Sünderhauf et al. *On the Performance of ConvNet Features for Place Recognition*. 2015. arXiv: 1501.04158 [cs.RO].

[16] Ruben Gomez-Ojeda et al. *Training a Convolutional Neural Network for Appearance-Invariant Place Recognition*. 2015. arXiv: 1505.07428 [cs.CV].

[17] Yi Hou, Hong Zhang, and Shilin 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. DOI: 10.1109/ICInfA.2015.7279659.

[18] Relja Arandjelović et al. *NetVLAD: CNN architecture for weakly supervised place recognition*. 2016. arXiv: 1511.07247 [cs.CV].

[19] Zetao Chen et al. *Deep Learning Features at Scale for Visual Place Recognition*. 2017. arXiv: 1701.05105 [cs.CV].

[20] Tayyab Naseer et al. "Semantics-aware visual localization under challenging perceptual conditions". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 2614–2620. DOI: 10.1109/ICRA.2017.7989305.

[21] Zetao Chen et al. "Learning Context Flexible Attention Model for Long-Term Visual Place Recognition". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4015–4022. DOI: 10.1109/LRA.2018.2859916.

[22] Yasir Latif et al. "Addressing Challenging Place Recognition Tasks Using Generative Adversarial Networks". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 2349–2355. DOI: 10.1109/ICRA.2018.8461081.

[23] Nate Merrill and Guoquan Huang. *Lightweight Unsupervised Deep Loop Closure*. 2018. arXiv: 1805.07703 [cs.RO].

[24] Gabriele Berton et al. *Deep Visual Geo-localization Benchmark*. 2022. DOI: 10.48550/ARXIV.2204.03444. URL: https://arxiv.org/abs/2204.03444.

[25] Stephen Hausler and Michael Milford. *Hierarchical Multi-Process Fusion for Visual Place Recognition*. 2020. DOI: 10.48550/ARXIV.2002.03895. URL: https://arxiv.org/abs/2002.03895.

[26] Stephen Hausler, Tobias Fischer, and Michael Milford. *Unsupervised Complementary-aware Multi-process Fusion for Visual Place Recognition*. 2021. arXiv: 2112.04701 [cs.CV].

[27] Stephen Hausler, Adam Jacobson, and Michael Milford. "Multi-Process Fusion: Visual Place Recognition Using Multiple Image Processing Methods". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1924–1931. ISSN: 2377-3774. DOI: 10.1109/lra.2019.2898427. URL: http://dx.doi.org/10.1109/LRA.2019.2898427.

[28] Michael J Milford and Gordon F Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights". In: *2012 IEEE international conference on robotics and automation*. IEEE. 2012, pp. 1643–1649.

[29] Edward Pepperell, Peter I Corke, and Michael J Milford. "All-environment visual place recognition with SMART". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 1612–1618.

[30] Yangqing Jia et al. *Caffe: Convolutional Architecture for Fast Feature Embedding*. 2014. arXiv: 1408.5093 [cs.CV].

[31] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. ISSN: 1941-0468. DOI: 10.1109/tro.2015.2463671. URL: http://dx.doi.org/10.1109/TRO.2015.2463671.

[32] Niko Sünderhauf, Peer Neubert, and Peter 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)*. 2013, p. 2013.

[33] José-Luis Blanco-Claraco, Francisco-Angel Moreno-Duenas, and Javier González-Jiménez. "The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario". In: *The International Journal of Robotics Research* 33.2 (2014), pp. 207–214.

[34] Aude Oliva and Antonio Torralba. "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope". In: *International Journal of Computer Vision* 42 (2001), pp. 145–175.

[35] Akihiko Torii et al. “Visual place recognition with repetitive structures”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 883–890.

[36] Akihiko Torii et al. “24/7 place recognition by view synthesis”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1808–1817. DOI: 10.1109/CVPR.2015.7298790.

[37] Xiang Gao and Tao Zhang. “Unsupervised learning to detect loops using deep neural networks for visual SLAM system”. In: *Autonomous Robots* 41 (Jan. 2017). DOI: 10.1007/s10514-015-9516-2.

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

[39] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[40] Joaquin Quinonero-Candela et al. *Dataset shift in machine learning*. Mit Press, 2008.

[41] Swami Sankaranarayanan et al. “Learning from synthetic data: Addressing domain shift for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3752–3761.

[42] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagation”. In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.

[43] Swami Sankaranarayanan et al. “Generate to adapt: Aligning domains using generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8503–8512.

[44] Mingsheng Long et al. “Learning transferable features with deep adaptation networks”. In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

[45] José E Chacón, Tarn Duong, and MP Wand. “Asymptotics for general multivariate kernel density derivative estimators”. In: *Statistica Sinica* (2011), pp. 807–840.

[46] Tarn Duong. “ks: Kernel Density Estimation and Kernel Discriminant Analysis for Multivariate Data in R”. In: *Journal of Statistical Software* 21.7 (2007), pp. 1–16. DOI: 10.18637/jss.v021.i07. URL: https://www.jstatsoft.org/index.php/jss/article/view/v021i07.

[47] Arsalane Chouaib Guidoum. “Kernel estimator and bandwidth selection for density and its derivatives”. In: *Department of Probabilities and Statistics, University of Science and Technology, Houari Boumediene, Algeria* (2015).

[48] M.C. Jones. “On kernel density derivative estimation”. In: *Communications in Statistics - Theory and Methods* 23.8 (1994), pp. 2133–2139. DOI: 10.1080/03610929408831377. eprint: https://doi.org/10.1080/03610929408831377. URL: https://doi.org/10.1080/03610929408831377.

[49] Frederik Warburg et al. “Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[50] David M. Chen et al. “City-scale landmark identification on mobile devices”. In: *CVPR 2011*. 2011, pp. 737–744. DOI: 10.1109/CVPR.2011.5995610.

[51] Michael Warren et al. “Unaided stereo vision based pose estimation”. In: *Australasian Conference on Robotics and Automation*. Ed. by Gordon Wyeth and Ben Upcroft. Brisbane: Australian Robotics and Automation Association, 2010. URL: http://eprints.qut.edu.au/39881/.

[52] Mark Cummins. “Highly scalable appearance-only SLAM-FAB-MAP 2.0”. In: *Proc. Robotics: Sciences and Systems (RSS), 2009* (2009).

[53] Emanuel Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065 –1076. DOI: 10.1214/aoms/1177704472. URL: https://doi.org/10.1214/aoms/1177704472.

[54] Murray Rosenblatt. “Remarks on Some Nonparametric Estimates of a Density Function”. In: *The Annals of Mathematical Statistics* 27.3 (1956), pp. 832 –837. DOI: 10.1214/aoms/1177728190. URL: https://doi.org/10.1214/aoms/1177728190.

[55] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

[56] Filip Radenović, Giorgos Tolias, and Ondřej Chum. *Fine-tuning CNN Image Retrieval with No Human Annotation*. 2017. DOI: 10.48550/ARXIV.1711.02512. URL: https://arxiv.org/abs/1711.02512.

[57] *A Quick Guide to Using UTM Coordinates*. https://www.maptools.com/tutorials/utm/quick_guide. Accessed: 2023-01-03.

[58] Chris Fraley and Adrian E Raftery. "Model-Based Clustering, Discriminant Analysis, and Density Estimation". In: *Journal of the American Statistical Association* 97.458 (2002), pp. 611–631. DOI: 10.1198/016214502760047131.

[59] Yang Zhao, Abhishek K. Shrivastava, and Kwok Leung Tsui. "Regularized Gaussian Mixture Model for High-Dimensional Clustering". In: *IEEE Transactions on Cybernetics* 49.10 (2019), pp. 3677–3688. DOI: 10.1109/TCYB.2018.2846404.

[60] Charles Bouveyron and Camille Brunet. "Model-Based Clustering of High-Dimensional Data: A review". In: *Computational Statistics and Data Analysis* 71 (2013), pp. 52–78. DOI: 10.1016/j.csda.2012.12.008. URL: https://hal.archives-ouvertes.fr/hal-00750909.