M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Lifelong Place Recognition

Curation and Validation of the Mapillary Street-Level Dataset

Frederik Warburg (s153847)

Kongens Lyngby 2020

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Abstract

Lifelong place recognition is an essential and challenging task in computer vision with vast applications in robust localization and efficient large-scale 3D reconstruction. Progress is currently hindered by a lack of large, diverse, publicly available datasets. We contribute with Mapillary Street-Level Sequences (SLS), a large dataset for urban and suburban place recognition from image sequences. It contains more than 1.6 million images curated from the Mapillary[1] collaborative mapping platform. The dataset is orders of magnitude larger than current data sources, and is designed to reflect the diversities of true lifelong learning. It features images from 30 major cities across six continents, hundreds of distinct cameras, and substantially different viewpoints and capture times, spanning all seasons over a nine year period. All images are geo-located with GPS and compass, and feature high-level attributes such as road type.

We propose a set of benchmark tasks designed to push state-of-the-art performance and provide baseline studies. We show that current state-of-the-art methods still have a long way to go, and that the lack of diversity in existing datasets have prevented generalization to new environments. We show that training on the diverse SLS dataset can improve upon state-of-the-art performance across several datasets. In order to achieve these results we present a novel subcaching methodology with hard positive and negative mining. Furthermore, we extend several single-view place recognition models to multi-view models that are able to incorporate the temporal and spatial structure found in image sequences, and show that this can further improve model performance. Lastly, we propose two new research tasks; sequence to image and image to sequence place recognition, and present techniques to solve both these tasks.

[1] https://www.mapillary.com

Resumé

Livslang stedgenkendelse er en essentiel og udfordrende opgave i computer vision, som har mange applikationer i robust lokalisering og efficient 3D rekonstruktion i stor skala. Fremgangen er begrænset af mangel på store, diverse, offentligt tilgængelige datasæt. Vi bidrager med Mapillary Street-Level Sequences (SLS), som er et stort datasæt for by- og forstadsstedgenkendelse. Datasættet indeholder mere end 1,6 millioner billeder fra den kollaborative kortlæggelsesplatform Mapillary. Datasættet er flere ordner større end nuværende datakilder og er designet til at reflektere de mange diversiteter i livslang stedgenkendelse. Det indeholder billeder fra 30 store byer fordelt over seks kontinenter, optaget med hundredvis af forskellige kameraer af scener med stor forskel i perspektiv og optagelsestidspunkt. Optagelsestidspunkterne spænder alle sæsoner over en ni-årig periode. Alle billeder er geo-lokaliseret med GPS, kompass og abstrakte beskrivelser, som vejtype.

Vi foreslår flere benchmark opgaver, som er designet til at udvikle modellers ydeevne, og bidrager med evaluering af flere modeller. Vi viser, at de nuværende bedste metoder stadigt er langt fra færdigudviklede og at den manglende diversitet i nuværende dataset har forhindret modellernes evne til at generalisere til nye miljøer. Vi viser, at træning på det diverse SLS datasæt kan forbedre de bedste modellers præstationer over flere datasæt. For at opnå disse resultater, præsenterer vi et ny sub-caching metode, som vælger svære positive og negative eksempler. Derudover, udbygger vi flere enkelt-billede stedgenkendelsesmodeller til mange-billeder modeller, som er i stand til at inkorporere den tids- og stedsafhænging struktur fundet i billedsekvenser. Vi viser, at dette kan yderligere forbedre modellernes ydeevne. Til sidst, foreslår vi to nye forskningsopgaver; nemlig sekvens til billede og billede til sekvens stedgenkendelse. Vi udvikler flere metoder til at løse disse opgaver og rapporterer benchmarkresultater for begge disse opgaver.

Preface

This thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfillment of the requirements for acquiring a M.Sc. degree in Engineering (Mathematical Modelling and Computation).

I would like to thank my supervisor, Søren Hauberg, for helping me through out the entire process of developing the final work. I would also like to thank Javier Civera for valuable insights and counseling. Lastly, I would like to thank Mapillary for allowing me access to their large data corpus and especially thanks to Yubin Kaung, Manuel Antequera and Pau Gargallo for always providing excellent and fast assistance with the experimental work.

Kongens Lyngby, January 16, 2020

Frederik Warburg (s153847)

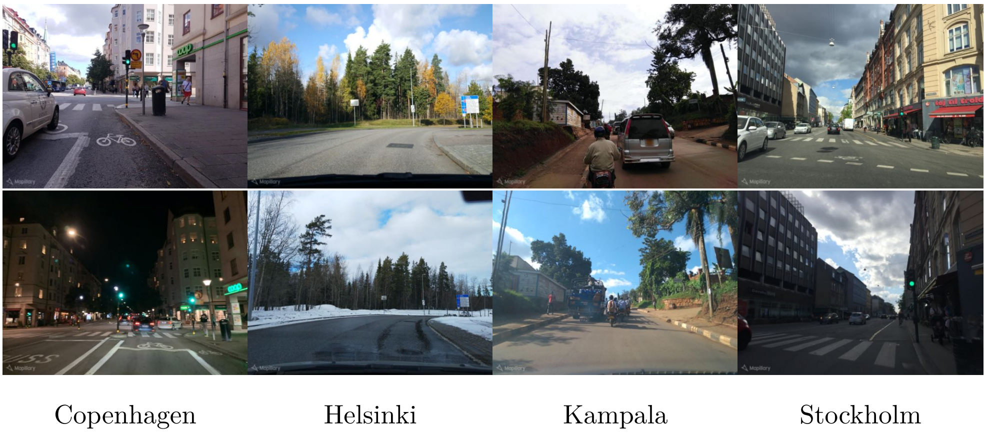
Contents

CHAPTER 1

Introduction

Visual place recognition is essential for the long-term operation of augmented reality and robotic systems (Lowry et al., 2016). The task consists of retrieving images within a database that are taken from the same geographical location as a query image (Lowry et al., 2016; Zaffar et al., 2019) where "the same geographical location" is typically defined as images within a radius of 25 meters. However, despite its relevance and vast research efforts, it remains challenging in practical settings due to the wide array of appearance variations in outdoor scenes, as seen on the examples extracted from the dataset propose as part of this thesis in Figure 1.5.

Recent research on place recognition has shown that features learned by deep neural networks outperform traditional hand-crafted features, particularly for drastic appearance changes (Lowry et al., 2016; Arandjelovic et al., 2016; Zaffar et al., 2019). This has motivated the release of several datasets for training, evaluating and comparing deep learning models. However, these datasets are limited, in at least three aspects. *First*, none of them covers the many appearance variations encountered in real-world applications. *Second*, many of them have insufficient size for training large networks.



Copenhagen Helsinki Kampala Stockholm

Figure 1.5: Show four samples of places from Copenhagen, Helsinki, Kampala and Stockholm with challenging appearance changes due to viewpoint, structural, seasonal, dynamic, and illumination differences.

Finally, most datasets are collected in small areas, lacking the geographical diversity needed for generalization.

This thesis contributes to the progress of lifelong place recognition by creating a dataset addressing all the challenges described above. We present **Mapillary Street-Level Sequences (SLS)**, the largest dataset for place recognition to date and the one that contains the widest variety of perceptual changes and the broadest geographical spread.[1] Mapillary SLS covers the following causes of appearance change: different seasons, changing weather conditions, varying illumination at different times of the day, dynamic objects such as moving pedestrians or cars, structural modifications such as roadworks or architectural work, camera intrinsics and viewpoints. Our data spans six continents, including diverse cities like Kampala, Zurich, Amman and Bangkok.

In addition to the dataset, we make several contributions related to its experimental validation. We formulate a novel subset caching methodology for hard triplet mining that does not scale in time or memory with the dataset size; enabling training of place recognition models on very large datasets (as ours). We tackle a wider set of problems not limited to image-to-image localization by proposing six variations of MultiViewNet (Fácil et al., 2019) to model sequence-to-sequence place recognition. Moreover, we formulate two new research tasks: sequence-to-image and image-to-sequence recognition, and propose several feature descriptors that extend pretrained image-to-image models to these two new tasks. Finally, we show that training on our dataset reduces models' geographical bias and improves their generalization capabilities.

1.1 Overview

The development and evaluation of place recognition methods has been limited by the size, diversity and geographical coverage of available datasets. In Chapter 2, we summarize the development within place recognition methods. We highlight the characteristics of existing datasets and propose a more fair test/train division for both the Pittsburgh250k and Tokyo24/7 datasets. Based on this literature review and analysis of existing datasets, we discovered a need for a large, diverse and sequentially structured dataset with a large geographical coverage. Constructing such a dataset would require hundreds of people to record thousands of places for many years around the world. Fortunately, this has already been done through the collaborative mapping platform, Mapillary. In Chapter 3 we explain how we curated a large place recognition dataset from this collaborative platform. In the following chapter, Chapter 4, we explain the models, experimental setup and model training used to validate our proposed dataset. We introduce a novel sub-caching method with hard positive and negative mining and suggest several variations of MultiViewNet (Fácil et al., 2019). Furthermore, we propose two new multi-view place recognition applications and tech-

[1] See the video accompanying the thesis for an overview and sample images.

niques to solve them. We report and discuss the results in Chapter 5, and round the thesis of with a conclusion in Chapter 6.
Note that substantial parts of the text is taken from a Computer Vision and Pattern Recognition (CVPR) paper submission, which was curated while writing the thesis. The CVPR submission is included in Appendix A.1.

CHAPTER 2

Related Work

Visual place recognition consists of recognizing a geographical location of a place based only on image data. The place recognition task is often cast as a retrieval problem. Given a query image, we wish to find the images from the same geographical location as the query image. We will refer to the images from the same location as positives and images from other locations as negatives. Note that both the positive and negative images are in the database.
The retrieval problem is solve by building a low dimensional, place-descriptor for both query and database images and for each query image retrieve the closest n database images (Lowry et al., 2016; Zaffar et al., 2019). The goal is to build a robust place descriptor that efficiently represents an exact geographical location based only on image information.

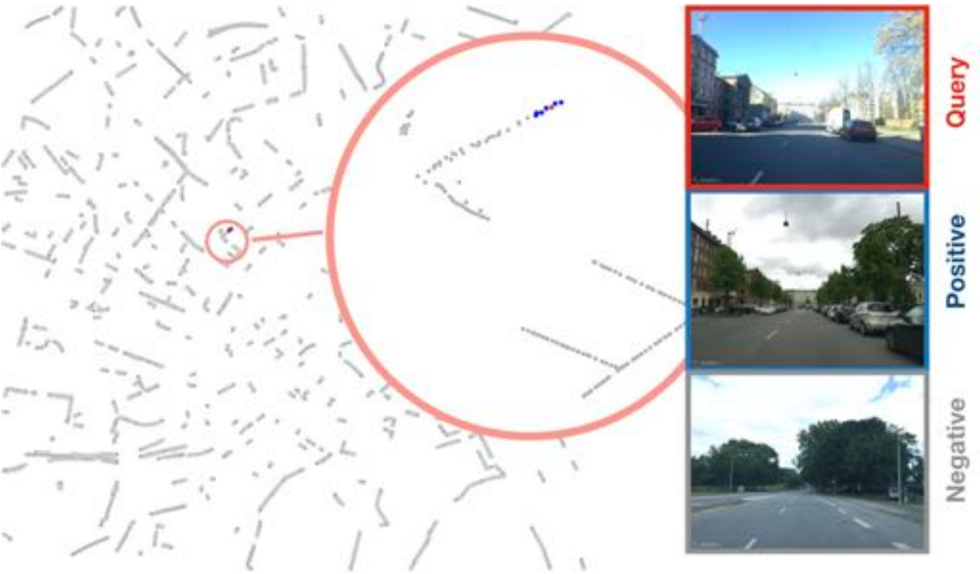


Figure 2.1: Shows a query image that has 6 positive images from the Copenhagen database. In this examples we use 25 meter as threshold to define positive images. All images are extracted from the Mapillary SLS dataset.

2.1 Related Methods

Traditional visual place descriptors are based on aggregating local features using bag-of-words (Sivic and Zisserman, 2003), Fischer vectors (Perronnin et al., 2010) or VLAD (Jégou et al., 2010). These features especially are prone to viewpoint, dynamic and seasonal changes between the query and positive images. To create place-descriptors that are more robust towards viewpoint and dynamic changes, other hand-crafted approaches exploit geometric and/or temporal consistencies in image sequences (Cummins and Newman, 2011; Gálvez-López and Tardos, 2012; Milford and Wyeth, 2012). An examples is DenseVLAD (Torii et al., 2015) that synthesizes viewpoint changes from panorama images with associated depth. These synthetic images make the DenseVLAD more robust to viewpoint changes.

As in other computer vision tasks, deep features have demonstrated better performance than hand-crafted ones (Lowry et al., 2016; Zaffar et al., 2019). Initially, features from existing pre-trained networks were used for single-view place recognition (Sünderhauf et al., 2015b,a; Chen et al., 2014; Sünderhauf et al., 2015; Yandex and Lempitsky, 2015). Later works demonstrated that the performance improves if the networks are trained for the specific task of place recognition (Arandjelovic et al., 2016; Lopez-Antequera et al., 2017; Gomez-Ojeda et al., 2015). These networks consist of a base architecture followed by a trainable layer that aggregates the features (see Figure 2.2). The most commonly used base architectures are AlexNet (Krizhevsky et al., 2012), VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2015), DenseNet (Huang et al., 2016) pre-trained on the ImageNet Challenge (ILSVRC) dataset (Russakovsky et al., 2015). The first aggregation layers proposed for place recognition were different pooling layers such as max-pooling (Razavian et al., 2014), average-pooling (Babenko and Lempitsky, 2015a) and sum-pooling (Babenko and Lempitsky, 2015b) with more (Tolias et al., 2015; Kalantidis et al., 2015). Radenovic et al. (2017) extended on these pooling layers by proposing a Generalized-Mean pooling layer. This layer is differentiable and have a parameter for each feature map - enabling the layer to learn the optimal type of pooling for each feature map (see section 4.1.2 for more details). Another recent success is the

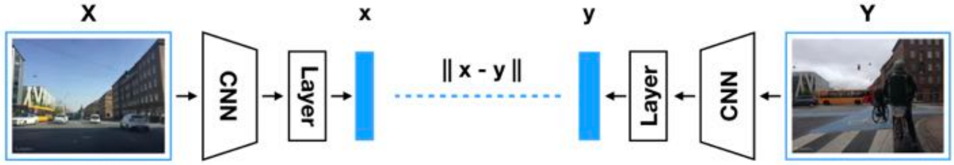


Figure 2.2: Recent place recognition networks consist of a deep CNN base architecture followed by layer specialized for place recognition. The goal is to learn an embedding where images that are geographically close will also be close in the learnt embedding space.

generalized VLAD layer (NetVLAD) (Arandjelovic et al., 2016; Zaffar et al., 2019). This layer softly assigns residual features that are extracted from the feature maps to "visual words" to form a robust place descriptor that imitates the VLAD layer (see section 4.1.1 for more details). Other work operates directly on the feature maps, thus keeping the spatial information of the features. These includes R-MAC (Tolias et al., 2016) and work by Chen et al. (2017) that extract regions from the feature map surrounding high activation to form place descriptors.

Recent deep learning-based methods exploit the temporal, spatial, and semantic information in images or image sequences. Garg et al. (2019a) uses single-view depth predictions to recognize places revisited from opposite directions. Also, addressing extreme viewpoint changes, Garg et al. (2019b) suggests semantically aggregating salient visual information. The 3D geometry of a place is also used by PointNetVLAD (Angelina Uy and Hee Lee, 2018) that combines PointNet and NetVLAD to form a global place descriptor from LiDAR data. MultiViewNet (Fácil et al., 2019) investigates different pooling strategies, descriptor fusion and LSTMs to model temporal information in image sequences. In this thesis, we extend upon MultiViewNet by incorporating the NetVLAD and GeM layers, which have both been successful for single-view place recognition.

All these models have in common that they strive to learn representations of places that are close if the places are geographically close (see Figure 2.2). The models are evaluated by their top-n recall. This is a global evaluation metric that requires a descriptor for all the images in the database, making the recall impractical to naively use as a loss function during training. Therefore, several local approximations of the recall have been proposed for training. These ranking losses consider either image pairs (Radenovic et al., 2017), triplets (Gordo et al., 2016), quadruplets (Chen et al., 2017a), or n-tuples (Sohn, 2016) to optimize a small local region around a query image (see section 4.4 for more details). As these local approximations does not necessary optimize the recall, it requires significant engineering efforts to learn a robust embedding. Arandjelovic et al. (2016) show that presenting the model for positive and negative embeddings that are already close to the query embedding significantly improves the model performance. This process is usually referred to as hard positive and negative mining. In this thesis, we further extent upon this idea and shows that carefully deciding which local regions to optimize also significantly improves the performance (see section 4.4.2 for more details). Other approaches, include estimating and using the 3D reconstructed scene to mine hard positive and negatives (Radenovic et al., 2017). However, this typically requires a large amount of images of the same place to create an accurate 3D reconstruction, which is not the case in our proposed Mapillary SLS dataset. Recently, Revaud et al. (2019) proposed several clever tricks to directly use the recall as the loss function in end-to-end training. This is enabled by using a histogram binning approximation to make the recall differentiable and multistage backpropagation to fit the optimization of deep networks on large images into memory.

These methods are limited by the availability of a large and diverse dataset for training and evaluation that captures the many appearances changes of a place. In the next section, we will highlight currently available datasets.

2.2 Related Datasets

In this section, we will describe the currently available place recognition datasets. Table 2.1 summarizes a set of relevant place recognition datasets. Below we highlight more details and compare our contributions against existing datasets.

						Type of appearance changes						
Name	Environment	Total length	Geographical coverage	Temporal coverage	Frames	Seasonal	Weather	Viewpoint	Dynamic	Day/night	Intrinsics	Structural
Nordland 41; 42	Natural + urban	728 km	182 km	1 year	~ 115K	✓	✗	✗	✗	✗	✗	✗
SPED 12	Urban	-	-	1 year	~ **2.5M**	✓	✓	✗	✓	✓	✗	✗
KITTI 20	Urban + suburban	39.2 km	1.7 km	3 days	~ 13K	✗	✗	✓	✓	✗	✗	✗
Eynsham 14	Urban + suburban	70 km	35 km	1 day	~ 10K	✗	✗	✗	✓	✗	✗	✗
St. Lucia 21	Suburban	47.5 km	9.5 km	1 day	~ 33K	✗	✗	✗	✓	✗	✗	✗
NCLT 9	Campus	148.5 km	5.5 km	15 mon.	~ 300K	✓	✗	✓	✓	✗	✗	✗
Oxford RobotCar 37	Urban + suburban	1.000 km	10 km	1 year	~ 27K	✓	✓	✓	✓	✓	✗	✓
VL-CMU 7	Urban + suburban	128 km	8 km	1 year	~ 1.4K	✗	✗	✓	✓	✗	✗	✗
FAS 39	Urban + suburban	120 km	70 km	3 years	~ 43K	✓	✓	✓	✓	✗	✗	✓
Garden Point 46	Urban + campus	< 12 km	4 km	1 week	~ 600	✗	✗	✓	✗	✓	✗	✗
SYNTHIA 51	Urban	6 km	1.5 km	-	~ 200K	✓	✓	✓	✓	✓	✗	✗
GSV 68	Urban	-	-	-	~ 60K	✗	✗	✗	✗	✗	✗	✗
Pittsburgh 250k 62	Urban	-	-	-	~ 254K	✗	✗	✓	✓	✗	✗	✗
TokyoTM/247 61	Urban	-	-	-	~ 174K	✓	✗	✓	✓	✓	✗	✓
TB-places 32	Gardens	< 100m	< 100m	1 year	~ 60K	✗	✗	✓	✓	✗	✗	✗
Mapillary SLS (Ours)	Urban + suburban	**11, 560 km**	**4, 228 km**	**7 years**	~ 1.68M	✓	✓	✓	✓	✓	✓	✓

Table 2.1: **Summary of place recognition datasets**. Geographical coverage is the length of unique traversed routes. Total length is the geographical coverage multiplied by the number of times each route was traversed. Temporal coverage is the time span from the first recording of a route to the last recording. "–" stands for "not applicable".

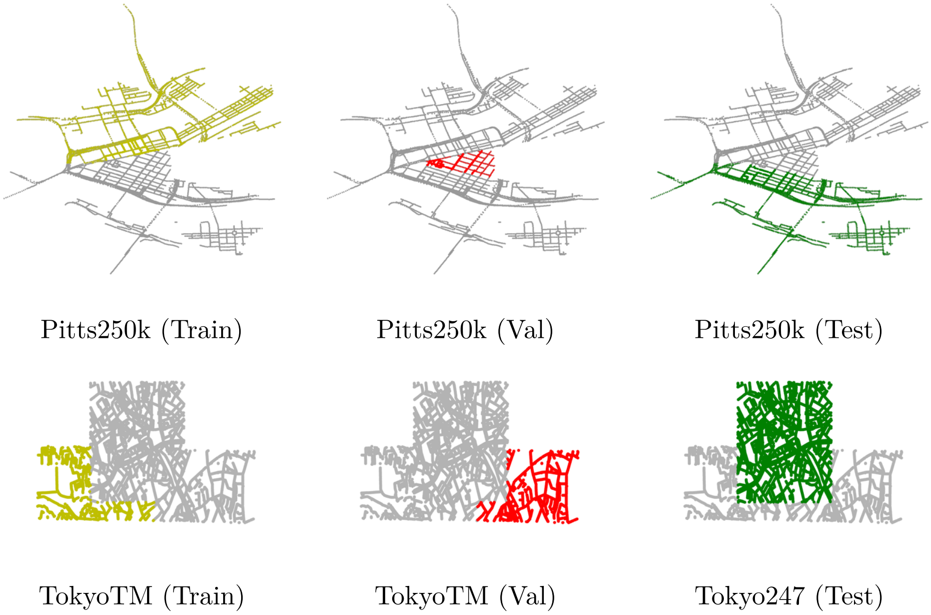
Nordland (NRK, 2013; Olid et al., 2018) contains 4 sequences of a 182km-long train journey, traversed once per season. It captures seasonal changes but contains small variations in viewpoint, camera intrinsics, time of day or structural changes.
SPED (Chen et al., 2017b) was curated from images taken by 2.5K static surveillance cameras over 1 year. It contains dynamic, illumination, weather and seasonal changes. However, it does not include viewpoint changes nor ego-motion.
KITTI (Geiger et al., 2013), **Eynsham** (Cummins, 2009) and **St. Lucia** (Glover et al., 2010) were all recorded by car-mounted cameras. In all three cases the cars drove in urban environments within a few-days, capturing dynamic elements and slight viewpoint and weather changes, but no long-term variations. There are several other datasets oriented to autonomous driving collected over longer periods: **NCLT** (Carlevaris-Bianco et al., 2015) (recorded over a period of 15 months in a campus environment), **Oxford RobotCar** (Maddern et al., 2017) (recorded from a car traversing the same 10 km route twice every week for a year), **VL-CMU** (Badino et al., 2011) (composed by 16 × 8 km street-view videos captured over one year) and **Freiburg Across Seasons (FAS)** (Naseer et al., 2018) (composed of 2 × 60 km summer

videos and 1 × 10 km winter video over a period of three years). None of them has geographical diversity nor variations in the camera intrinsics, and their viewpoint, structural and weather changes are minor.
Gardens Point (Queensland University of Technology, 2014) was recorded with a hand-held iPhone. It contains day/night and significant viewpoint changes, but a small representation of other appearance changes and has a small size. **SYNTHIA** (Ros et al., 2016) contains 4 synthetic image sequences along the same route. It includes varying viewpoints, seasonal, weather, dynamic and day/night changes.
GSV (Zamir and Shah, 2014) compiled a street-level image dataset from Google Street View. However, it is relatively small at $60,000$ images. It is limited to a few US cities with no temporal changes and it is composed of still images instead of sequences. **Pittsburgh250k** (Torii et al., 2015) was also extracted from Google Street View panoramas in Pittsburgh ($10,586$ of them specifically, using two yaw directions and 12 pitch directions). The limited geographical span of these datasets results in a low number of unique places compared to ours.
Tokyo (Torii et al., 2015) comes in two versions: The Tokyo Time Machine dataset (~ 98K images) and Tokyo 24/7 (~ 75K images). Tokyo 24/7 has significant day/night changes. However, Arandjelovic et al. (2016) comments that the trained model with the Tokyo datasets shows signs of overfitting, probably caused by their limited geographical coverage and size.
Notice that **GSV**, **Pittsburgh250k** and **Tokyo** have significant viewpoint variation but do not include information of viewing direction for the images, and hence positive mining of images with overlaps in view point is not straightforward. In our **Mapillary SLS**, we include viewing direction information for each image. (see details in Chapter 3).
Image retrieval is a similar task to place recognition, aiming to find an image in a database that is the most similar one to a query image. There exist several image retrieval datasets (typically created from Flickr images) and established benchmarks, *e.g.*, Holidays (Jegou et al., 2008), Oxford5k, Paris6k (Philbin et al., 2007), Revisited Oxford5k and Paris6k (Radenovic et al., 2018), and Google Landmarks (Noh et al., 2016; Ozaki and Yokoo, 2019). They usually focus on single-image retrieval and have a very large set of images from the same landmark or object, which limits their application in benchmarking lifelong place recognition.

2.3 Revisiting Pittsburgh and Tokyo

The two largest and most diverse dataset currently available for place recognition are the Pittsburgh250k (Torii et al., 2015) and the Tokyo (Torii et al., 2015) datasets. These datasets are collected with the same procedure from Google Street View panorama images. In this section, we highlight the undesirable visual overlap between the train/-validation and test set for both of these dataset. We propose to separate the train/-validation and test set by 100 meters to ensure a more fair evaluation on the test set.

In Figure 2.9, we show the train/validation/test division of the Pittsburgh250k and Tokyo datasets. We see that both the Pittsburgh250k and the Tokyo test set share borders with their respective training and validation sets. It is common practice to train on both the training and the validation set once the hyper-parameters are chosen. Therefore, to obtain a fair evaluation on the test sets, we do not wish that the test sets share visual content with neither the training nor the validation set.



Pitts250k (Train) Pitts250k (Val) Pitts250k (Test)

TokyoTM (Train) TokyoTM (Val) Tokyo247 (Test)

Figure 2.9: Shows the train/validation/test division of the Pittsburgh250K and Tokyo datasets. Note that both the training and validation set share geographical borders with the test set for both datasets.

In Figure 2.10, we show the original division of the Pittsburgh250k test and train/validation set. We highlight examples along the border of the test and train/validation set that share substantial visual information. We found the closest points to be less than 1 meter apart. Note that the Pittsburgh30K dataset is a subset of the Pittsburgh250K dataset, so the findings also apply for this dataset division. Based on these findings, we propose to discard all the images in the test set that are closer than 100 m to the border. This proposed train/validation/test division is showed in Figure 2.11 for the Pittsburgh dataset. Note that the images at the border do not share visual information with the proposed data division. For fair visualizations, these images are chosen as the closest images that have the same view-direction similar to in Figure 2.10. By discarding images close to the border, we reduce the Pittsburgh250k test set by 6.792 (7.4%) images and the Pittsburgh30k test set by 2.760 (16.4%) images, however, we ensure a fair evaluation of models trained on the Pittsburgh dataset.

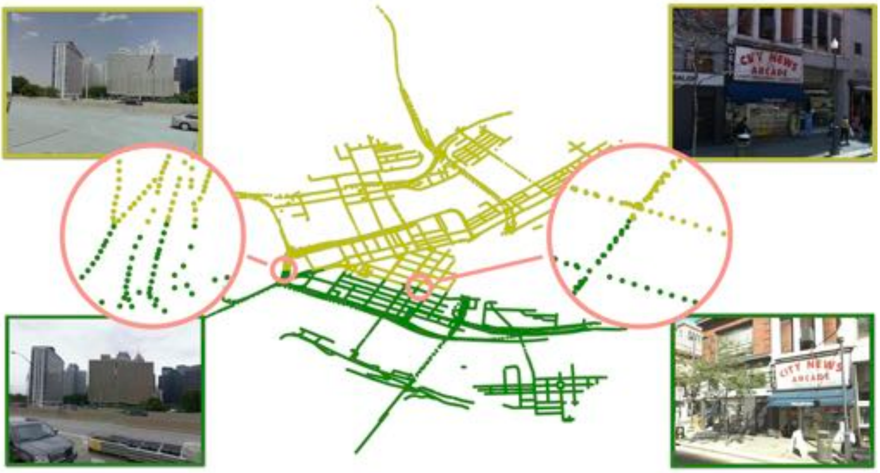


Figure 2.10: Shows a clear visual overlap between Pittsburgh test and train/validation dataset. The two zooms highlight that there is not a geographical overlap, but a clear visual overlap between the test and train/validation sets.

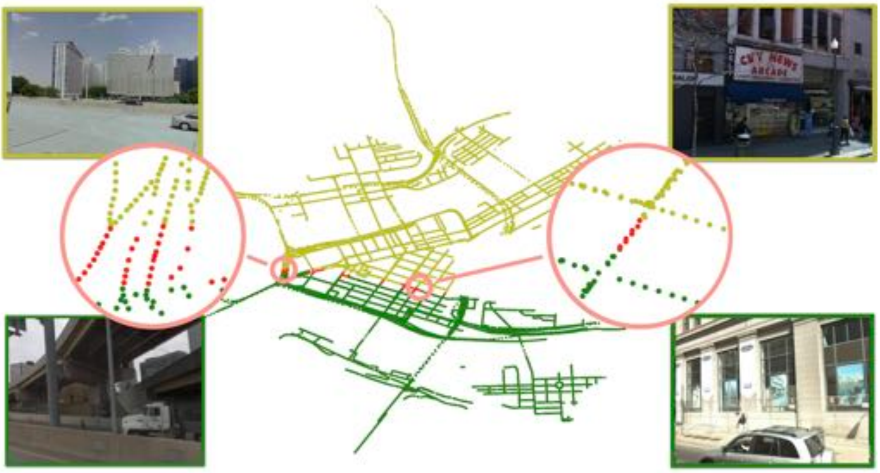


Figure 2.11: Shows the revisited Pittsburgh dataset. The yellow points are the original train/validation set, the red points are the images discarded from the test set, and the green points are the remaining images in the test set. The two zooms highlight that there is not a visual overlap between the revisited test and train/validation sets.

We implemented the same procedure for the Tokyo dataset. Figure 2.12 highlights the new train/validation/test division for this dataset as well as images for before the proposed split. The revisited Tokyo247 test is reduced by 9.342 (12.2%) images.

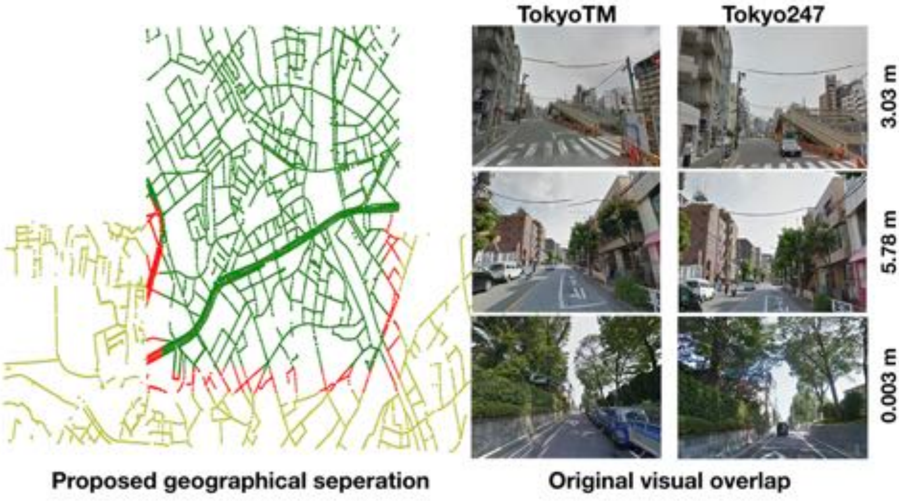


Figure 2.12: Shows the proposed train/validation test division of the Tokyo dataset and three image pairs that shows the significant visual overlap between the train/-validation set (TokyoTM) and the test set (Tokyo247) in the original Tokyo dataset. Note that the highlighted places are all less than 6 meters apart and that the same place is usually defined within a 25 meters radius. Thus, the same places are part of both train/validation and test set.

The proposed separation between the test and train/validation sets result in a more fair evaluation on the test set. However, as these datasets are collected from a rather small geographical region it is still unsure how models generalize to other regions, cities, suburban environments or more rural areas. Furthermore, these datasets are curated with image-to-image methods in mind, making it difficult to exploit the temporal information in image sequences.
This motivate us to create a new dataset with a very large geographical coverage, such that we can evaluate how models generalize to new cities, urban environments and rural areas. Furthermore, we want to make a dataset that is curated for sequential methods, such that we can enable the development of new place recognition methods that exploit the temporal information. In the next chapter, we explain how we achieve both these requirements through a close collaboration with mapping company, Mapillary.[1]

[1] www.mapillary.com

CHAPTER 3
Mapillary SLS dataset

To push the state-of-the-art in lifelong place recognition, there is a need for a larger and more diverse dataset that is curated for sequential methods. We have, with this in mind, created a new dataset comprised of 1.6 million images from Mapillary[1]. Mapillary is a collaborative platform that crowdsources imagery from users who take street-level pictures with their smartphones. The images are hosted on a website similar to Google Street View. Mapillary's infrastructure builds on top of another collaborative platform, OpenStreetMap[2], which enables users to edit a shared map of the world.

In this chapter, we present how we collected data from both these collaborative platforms to create a very large dataset for place recognition with detailed meta data about the curated imagery. We show important characteristics and statistics of the dataset. With the available sequential information of the dataset, we additionally propose two new research benchmark tasks.

Similar to previous works, we cast place recognition as an image retrieval problem and use the top-5 recall as evaluation metric. Our dataset is applicable for both single-view and multi-view models. Therefore, we will in the following use the query/database **example** to describe either a query/database image or a query/database sequence.

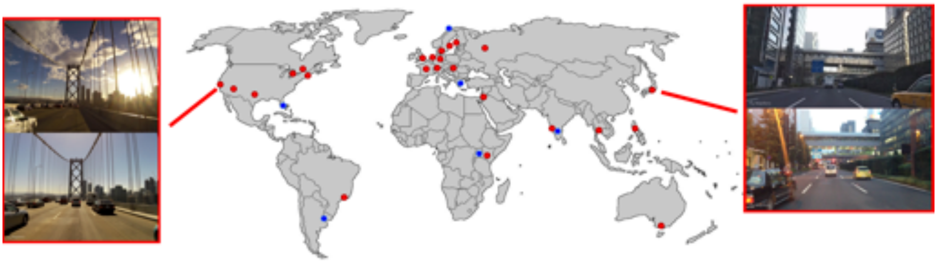


Figure 3.1: Mapillary SLS contains imagery from 30 major cities around the world; red stands for training cities and blue for test cities. See two samples from San Francisco and Tokyo with challenging appearance changes due to dynamic and illumination differences.

[1] www.mapillary.com

[2] www.openstreetmap.org

3.1 Data Curation

Our goal is to create a dataset for place recognition with images that (1) has wide geographical reach, reducing bias towards highly populated cities in developed countries (2) is visually diverse, capturing scenarios under varying weather, lighting, and time (3) tagged with reliable geometric and sequential information, enabling new research and practical applications.

Acquiring such a dataset would require hundreds of people and several years of recording. Fortunately, Mapillary have already collected imagery from a large part of the world over the last decade. We will in the following, explain how we have recollected data from their platform to create a large, visual diverse, sequentially structured dataset for place recognition.

3.1.1 Image Selection

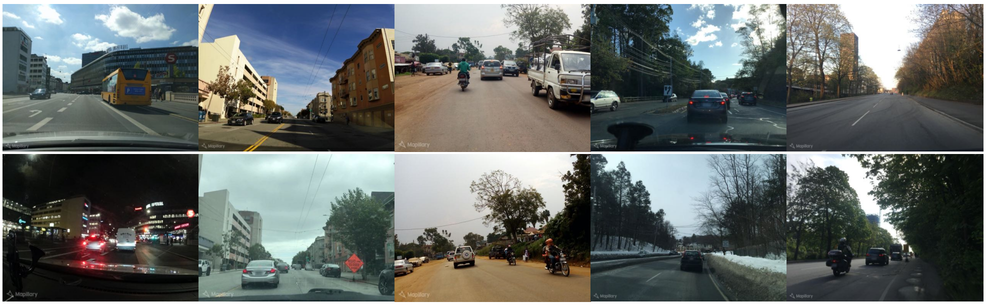


Figure 3.2: Mapillary SLS pairs showing day/night, weather, seasonal, structural, viewpoint and domain changes. Each column shows images from the same place that are recorded at different dates.

Geographical Diversity. To ensure geographical diversity, we start with a set of candidate cities for image selection. For each candidate city, we create a regular grid of 500m^2 cell size and process each of the cells independently. For each cell we extract a series of image sequences recorded within this cell. Each sequence contains the image keys and their associated GPS coordinates and raw compass angles (indicating viewing direction).

Mapillary SLS contains data from 30 cities spread over 6 continents. See Figure 3.1 and Table 3.1 for details. It covers diverse urban and suburban environments as indicated by the distribution of corresponding OpenStreetMap (OSM)[3] road attributes (Figure 3.3).

[3] www.openstreetmap.org

Continent	# Frames	# Night Frames	Geo. Coverage [km]	Total Coverage [km]	#Clusters
Europe	516 K	1,098	1,052	2,985	8,654
Asia	468 K	9,820	965	2,729	5,483
North America	431 K	3,968	171	4,616	6,504
South America	61 K	1,177	214	599	1,065
Australia	200 K	0	259	568	1,493
Africa	5 K	0	28	63	108
Total	**1,681 K**	**16,063**	**4,228**	**11,560**	**23,307**

Table 3.1: Continental coverage in Mapillary SLS.

Unique User and Capture Time. To ensure variation in the scene structure, time of day, camera intrinsics and view points within each geographical cell, we only keep one sequence per photographer and pick sequences from different days.

Consistent Viewing Direction. To ensure that viewing direction measurement is reliable for selecting matching images, we enforce consistency between raw compass angles (measured by the capturing device) and the estimated viewing direction computed with Structure from Motion (SfM)[4]. We select only sequences in which at least 80% of the images' computed angles agree ($\leq 30^\circ$ difference) with the raw compass angle.

3.1.2 Sequence Clustering

Given this initial set of sequences from the image selection process, we generate clusters of sequences that are candidates for place recognition. To avoid sequences where the distance between consecutive images is large, we first split each raw sequence into subsequences if there are more than 30 m between two consecutive frames. Then, we pairwise-match these sub-sequences based on their distance, viewing direction, and motion direction.[5] This is done by searching among all the sub-sequences and forming candidate clusters (sub-sequence pairs) based on their distances to all other neighboring sub-sequences.

To form a candidate cluster we use the following criteria: Frames from sub-sequences A and B are clustered together if: **1)** Their distance is less than 30 m. **2)** The difference between their viewing directions is less than 40°. **3)** The difference between their moving directions is less than 40°. In practice, we use a k-d tree to efficiently discover these pairwise correspondences. The above criteria sometimes skips intermediate images in a sequence, *e.g.*, a subsequence might have the images $\{1, 2, 4, 5\}$, thus missing image 3. To avoid this effect, we added all such skipped images back into the sequence.

[4] The estimated viewing directions were computed based on the relative camera poses estimated using the default OpenSfM (Ope) pipeline with the camera positions aligned with the GPS measurements

[5] The motion direction for each image is calculated using the GPS measurement and the capture times of consecutive images in a sequence.

After matching sub-sequence pairs into potential clusters, we prune them to obtain the frames where both subsequences overlap and hence can be used for sequence-to-sequence place recognition. Since there might be more sequences matching, we merge all pairwise clusters (e.g., we merge clusters A, B and C if there are images that belong to clusters AB, AC and BC.)

We end up with clusters of sequences that have the same geographical coverage and the same moving- and viewing-direction. The sequences in the clusters are relatively short (5-300 frames) providing a very diverse set of sequential examples for training and development of multi-view place descriptors.

Finally, we filter the resulting clusters enforcing: **1)** that each subsequence has 5 or more frames for proper evaluation of multi-view place recognition models; and **2)** that each cluster has at least two subsequences, in order to have a sufficient number of positive training and test samples (See appendix A.5 for more details on the curation of the MSLS dataset).

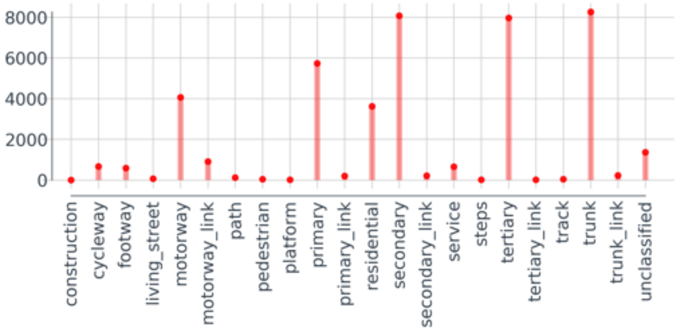


Figure 3.3: Distribution of OSM road attributes for Mapillary SLS.

3.2 Image Attributes

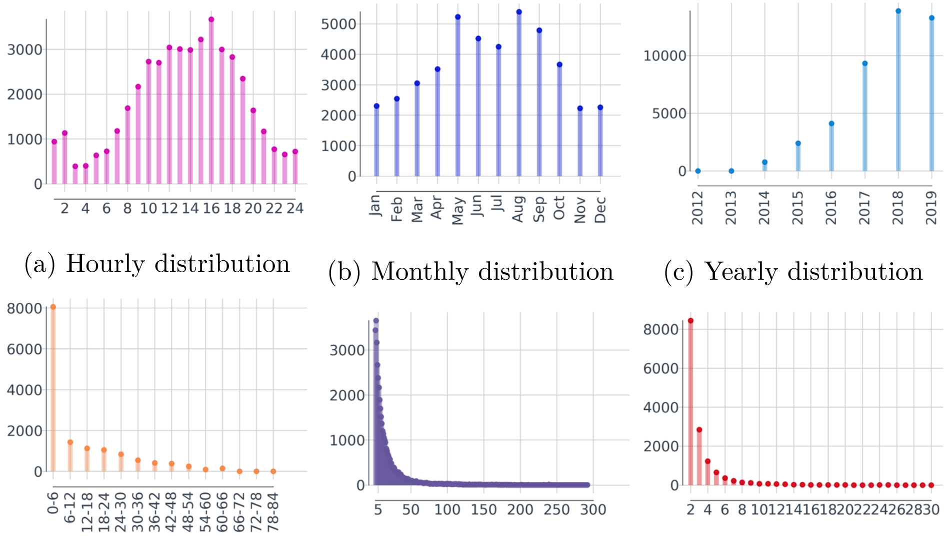
For each image, we additionally provide several image attributes that are relevant for further research.

Day and Night. We provide an attribute indicating if a sequence is captured during day or night time. We verified that the day/night attribute could not be robustly estimated from the capture time of the images. Therefore, we implemented a day/night classifier based on the hue distribution of the entire image and of the sky region identified using semantic segmentation. Given the prediction of each image, we then performed a majority voting across the entire sequence to provide consistent day/night tags. To obtain the sky region, we use a semantic segmentation mask provided by Mapillary's API. By manual inspection, we found that such a classifier is sufficient.

Facing Direction. We additionally include the facing direction of the camera: forward, backward or sideways, which is calculated using the estimated moving direction and viewing direction for each image.
Road Attributes. Based on the GPS locations of the images, we have also tagged each sequence with road attributes (*e.g.,* residential, motorway, path or others), which are obtained from OpenStreetMap[6] (OSM).

3.3 Data Overview

In this section, we provide an overview of the Mapillary SLS dataset in terms of its diversity. In Figures 3.10 (a) and (b), we show that the dataset covers all times of the day and months of the year. Figures 3.10 (c) and (d) show that it spans nine years and that the same places have been revisited with up to seven years time difference, making Mapillary SLS the dataset with largest time span for lifelong place recognition. Figures 3.10 (e) and (f) show the large variety in sequence length and the number of recordings for the same places.



(a) Hourly distribution (b) Monthly distribution (c) Yearly distribution

(d) Time diff [months] (e) #of frames/sequence (f) # of sequences/cluster

Figure 3.10: Distribution of image sequences in Mapillary SLS on a daily, monthly, yearly scale, time variation, and sequence-related characteristics.

To highlight the broad variety and challenge, we show in Figure 3.2 image samples from our dataset, where each column contains a query and a database image at a

[6] https://wiki.openstreetmap.org

nearby location. In the first column, the query images are taken during the day whereas the database image is taken at night. The second column shows an example of drastic weather changes as well as a new roadwork traffic sign. The third column shows images from Kampala; a drastic change in environment compared to the images in the first two columns from Copenhagen and San Francisco. Seasonal and structural changes are visible in the two last column, as the sky scraper on the left side of the road is under construction in the bottom image and stands finished in the top one. More visual examples of vast variety of changes between query and database images are available in appendix A.2.

3.4 Data Partition and Evaluation

We divide the dataset into a training set (roughly 90%) and a test set (the remaining 10%) containing disjoint sets of cities. Specifically, the test set consists of images collected from Miami, Athens, Buenos Aires, Stockholm, Bengaluru and Kampala. We phrase four place recognition tasks combining single images and sequences in the query and database, that will be referred from here on as **im-2-im**, **seq-2-seq**, **im-2-seq**, and **seq-2-im** (**x-2-y** stands for query **x** and database **y**), respectively.

Similar to previous works, we cast place recognition as an image retrieval problem and use the top-5 recall as evaluation metric. For each cluster, we randomly choose one query sequence and add the rest of the sequences to the database. The query examples are then chosen as the center frame(s) in the chosen query sequence. Only one query example is chosen per query sequence, ensuring an equal weight of every place in the evaluation independently of its number of frames. We define the ground truth matches as images within a radius of 25 m of the query image that have a viewing angle difference smaller than 40°.

In addition to evaluating on the whole test set, we suggest the following three research challenges and provide a separate evaluation for each: **Day/Night** (how well the model recognizes places from day and night and vice versa), **Seasonal** (how well the model recognizes places between seasons, Summer/Winter and vice versa being the most challenging) and **New/Old** (how well does the model recognizes places after several years).

We have in this chapter described the curation of the very large Mapillary SLS place recognition dataset. We will in the next chapter explain the methods, experimental setup and training procedure used to validates its contribution.

CHAPTER 4

Theory and Experimental Setup

We wish to validate the contribution of the Mapillary SLS dataset. We expect that the larger size, geographical coverage and increased diversity of the dataset improve the performance of deep models. In this chapter, we describe the methods, experimental setup and training procedure used to validate the dataset. First, we describe the single-view methods for im-2-im evaluation, then the sequence based models used for seq-2-seq, im-2-seq and seq-2-im evaluation, and round the chapter of by describing the conducted training procedure.

4.1 Single-View Methods

Most research on place recognition focuses on image to image retrieval. Two of the most successful models are NetVLAD (Arandjelovic et al., 2016) and GeM (Radenovic et al., 2017). Both these models consists of a deep convolutional neural network (CNN) as base network, e.g. VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2015) or AlexNet (Krizhevsky et al., 2012),[1] followed by either the NetVLAD layer or the Generalized Mean (GeM) pooling layer. These specially designed layers aggregates the local features extracted from the base network to form a D dimensional place descriptor. We will in the following describe the NetVLAD and GeM layers in more detail.

4.1.1 NetVLAD

The NetVLAD layer is a differential generalization of the Vector of Locally Aggregated Descriptors (VLAD) - allowing for end-to-end training and learnable ”visual words” (cluster centres). VLAD has traditionally been a popular descriptor for both instance-level retrieval and image classification as it captures information about the statistics of local descriptors aggregated over the image. VLAD stores the sum of residuals

[1] These models are designed for image classification. Therefore, the last fully connected classification layers are removed when the base architectures are used as encoders for place recognition.

(the difference vector between the descriptor and its corresponding cluster centre) for each visual word.
More formally, given N D-dimensional local descriptors $\{x_i\}$ as input, and K cluster centres $\{c_k\}$, the output place descriptor V is a $K \times D$ matrix. The differentiability is achieved by replacing the hard cluster assignment, which is used in VLAD, by a soft assignment $\overline{a}$.

$$V(j,k) = \sum_{i=1}^{N} \overline{a}_k(x_i)(x_i(j) - c_k(j)) \tag{4.1}$$

where

$$\overline{a}_k(x_i) = \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} \tag{4.2}$$

where $\{c_k\}$, $\{w_k\}$ and $\{b_k\}$ are trainable parameters and $\overline{a}_k(x_i)$ decides how likely a local descriptor $\{x_i\}$ is to belong to cluster k. Note that the soft-assignment, $\overline{a}_k(x)$, can efficiently be calculated with a standard convolution followed by a softmax operations. The soft-assignment allows end-to-end training.
The columns in the matrix V are then normalized individually, stacked and normalized again to form a $K \cdot D$ place descriptor.

4.1.2 Generalized Mean

Both global max and average pooling across the feature map have been used successfully for place recognition (Razavian et al., 2014; Babenko and Lempitsky, 2015a). Radenovic et al. (2017) suggest a trainable Generalized-Mean (GeM) pooling layer which is a generalization of max and average pooling. Given D (H, W)-dimensional feature maps $\{\chi_k\}$ as input, GeM pooling will output a D-dimensional place descriptor V.

$$V(k) = \left(\frac{1}{|\chi_k|} \sum_{x \in \chi_k} x^{p_k} \right)^{\frac{1}{p_k}} \tag{4.3}$$

where the pooling operation reduces each feature map $\{\chi_k\}$ to just one element in the place descriptor V. The GeM layer has D trainable parameters $\{p_k\}$ - one for each feature map. Note that $p_k \rightarrow \infty$ is max-pooling and $p_k = 1$ is average pooling.
The GeM feature representation is a very compact place descriptor. Its place descriptor is $1/K$ times the size of NetVLAD. This means that the NetVLAD descriptor has potential for a more fine grained modelling of a place, but it will be slower to find the nearest place in the database. On the other hand, GeM has D trainable parameters compared to NetVLAD that has $3K$.[2] For many popular base networks the dimen-

[2] NetVLAD has 1 trainable parameter for each K cluster centre $\{c_k\}$ and $((1 \cdot 1) + 1) \cdot K = 2K$ convolutional parameters as the kernel size is $(1, 1)$ and the output dimension is K.

sion of the last feature map is 256 (AlexNet (Krizhevsky et al., 2012)), 512 (VGG (Simonyan and Zisserman, 2015)), 2048 (ResNet (He et al., 2015))[3] and typically K is set to 64 (Arandjelovic et al., 2016). Thus, NetVLAD has significantly less trainable parameters. This means that the NetVLAD layer is less prone to overfitting and requires less data for training. Based on these considerations, and its longer history within place recognition, and established benchmarks, we have chosen to primarily focus on the NetVLAD layer in our validation of the Mapillary SLS dataset.

4.2 Multi-View Methods

Less work have focused on utilizing multiple views for place recognition. We propose several variants of MultiViewNet (Fácil et al., 2019) that allows us to solve both sequence to sequence (seq-2-seq), sequence to image (seq-2-im) and image to sequence (im-2-seq) retrieval. To the best of our knowledge, we are the first to propose methods for the two latter research tasks. We will in the following describe our suggested multi-view methodology.

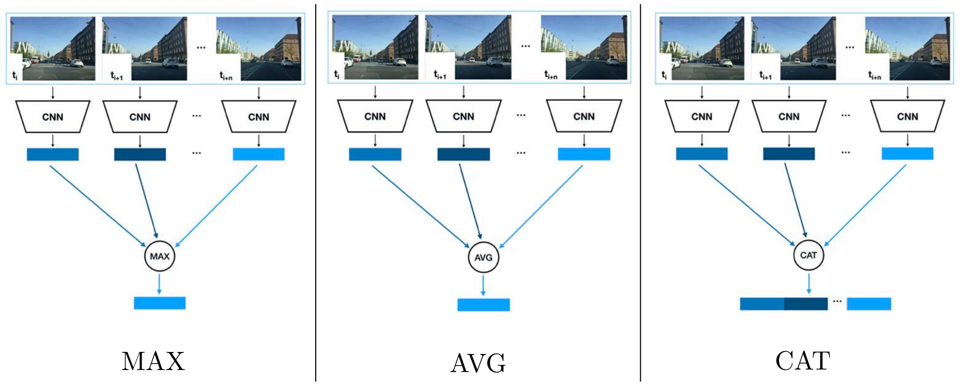
4.2.1 Seq-2-Seq Methods

We propose six variations of a MutiViewNet (Fácil et al., 2019), specifically, three pooling techniques for NetVLAD and three for GeM. The motivation is to adapt embeddings that are known to work well for single-view place recognition. The first techniques, NetVLAD/GeM-MAX/AVG, performs max or average pooling across the embeddings of each image in the sequence. And the third variation, NetVLAD/GeM-CAT concatenates the embeddings, building a embedding that has dimension $(n + 1) \cdot D$, where D is the dimension of the single-frame descriptors and $(n + 1)$ is the number of frames in the sequence.

4.2.2 Seq-2-Im and Im-2-Seq Methods

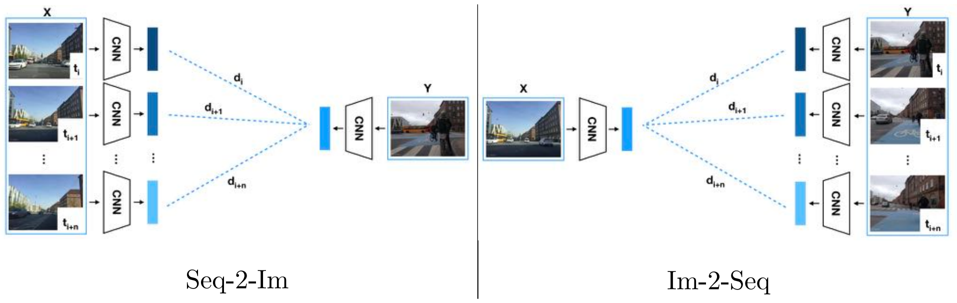
In the sequence to image case, we propose to make a majority voting across the sequence, i.e. select the image in the database that most images are nearest to in the query sequence. Given a query sequence of N frames, we calculate the distance from each frame to each database image. We then look at the closest k distances for each of our N frames in our query sequence. This gives a total of $k \times N$ closest database images. We then select the most frequently occurring. The intuition is that if all the frames in a sequence are close to a database image, then we are more confident that this database image is indeed close to the query sequence. We also test the selection of the closest image in the database among all the images in the query sequence. Again, we test these methods using both the VGG16 + GeM and VGG16 + NetVLAD embeddings

[3] Note that the smallest ResNet (ResNet18) has a 512 dimensional encoding.



MAX AVG CAT

Figure 4.4: Shows the proposed variant of MultiViewNet (Fácil et al., 2019). Frame $t_i \dots t_{i+n}$ in the sequence are forwarded through copies of a CNN to form $n+1$ descriptors. These descriptors are either pooled (MAX/AVG) or concatenated (CAT) into one descriptor. We propose to use NetVLAD/GeM as the frame-encoder (CNN).



Seq-2-Im Im-2-Seq

Figure 4.7: Shows the proposed sequence to image (Seq-2-Im) and image to sequence (Im-2-Seq) methods. The central idea is to treat each frame in the sequence independently in the model and exploit the sequential structure when retrieving the closest places. In the Seq-2-Im task, for each query sequence, distances are $d_i \dots d_{i+n}$ are calculated for each single image in the database. We investigate both retrieving the image from the database with the minimal distance or the shortest distance to most frames in the sequence. In the Im-2-Seq task, we extract the sequence that contains the image with the shortest distance to the query image.

In the image to sequence case, we select the sequence that contain the frame that is closest to query image. In practice, we calculate the distances from the query image to all the frames in the database sequences, and select the sequence that contains the nearest frame. This is visualized in Figure 4.7.

In this section, we explained two recent place recognition networks; NetVLAD and GeM. We introduced several variations of the MultiViewNet, which have potential to solve the seq-2-seq place recognition task. We further proposed techniques to evaluate seq-2-im and im-2-seq tasks using pretrained single-view models. In the following, we will explain the evaluation and training procedure.

4.3 Evaluation

The place recognition problem is often cast as a retrieval problem, where models are evaluated by their top-n recall. In place recognition, the top-n recall is defined as the proportion of correctly retrieved places. A correct retrieved place means that at least one of the n closest predictions must match one of the query place's positives, where these positives are typically defined as all the images within a 25 m radius of the query image. In the rest of thesis, we will assume $n = 5$ and a 25 m radius if not otherwise stated.

4.4 Training

Training place descriptors is not a trivial task as there is a very large number of different places and only weakly labelled data available.[4] In this thesis, we used the triplet loss to train our models. However, this loss function and the weak supervision requires substantial engineering effort in order to work well. In the following, we explain the conducted training procedure.

4.4.1 Triplet Loss

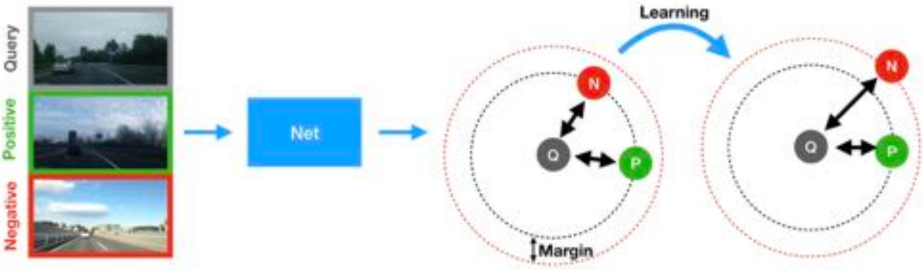


Figure 4.8: Illustrates the triplet loss. A model is fed a triplet that consists of query, positive and negative images. The triplet loss forces the distance between the query and the negative embedding to be larger than the distance between the query and the positive embedding plus a margin.

[4] Usually only GPS coordinates are available. GPS coordinates are usually associated with around 10 m error.

As our data is only weakly labelled and since we have a very high number of unique places (classes) we apply the triplet loss (Vassileios Balntas and Mikolajczyk, 2016) to train our networks. The intuition behind the triplet loss is that images of places from the same location are forced together in descriptor space and images from different places are pushed away from each other in the descriptor space. More formally, the loss function is given as

$$L(q, p, n, margin) = \max(0, ||q - p||^2 - ||q - n||^2 + margin) \tag{4.4}$$

where q is the embedding from the query, p is the positive embedding and n is the negative embedding (see Figure 4.8). The margin is a hyper-parameter that states how far from each other the descriptors of different places shall be. This is often fixed to 0.1. In all our experiments we use $||\cdot||^2 = ||\cdot||_2^2$, however other distance functions could be used.

Equation 4.4 shows that if the distance between the query and the negative descriptor is closer than the distance between the query and positive descriptor (plus a margin), then the loss will be zero, i.e.

$$||q - n|| < ||q - p|| + margin \Rightarrow L(q, p, n, margin) = 0 \tag{4.5}$$

We will refer to as a trivial example. As the networks improves this constraint becomes harder to violate, which means that the network does not improve. These trivial examples are the motivation for hard positive and negative mining.

4.4.2 Positive and Negative Mining

There is an extended literature on positive and negative mining (Arandjelovic et al., 2016; Radenovic et al., 2017; Li et al., 2019). In the following, the methodology adapted from Arandjelovic et al. (2016) is explained.

In this procedure, the descriptors of the entire dataset are saved in a cache, which is updated regularly throughout the training. For each query image, its cached descriptor is retrieved from the cache as well as the descriptors of all the putative positive images (images that are < 10 m from the query image). As the images from the same place can have imprecise view direction and GPS coordinates, we are not guaranteed that the query image have a visual overlap with all the images from the same place. Therefore, the distances between the cached query descriptor and the cached putative positive descriptor are calculated. In order to find a true positive descriptor, the cached putative positive with the minimum distance to the cached query descriptor is chosen as the positive example.

$$p_{i*}^q = argmin_{p_i^q} ||q - p_i^q|| \tag{4.6}$$

where p_{i*}^q is the closest positive and $||q - p_i^q||$ is the distance between the query and the embedding of the i^{th} putative positive that belongs to the query example.

The goal is to select negative images where their descriptors will violate the triplet constraint (Equation 4.4). Arandjelovic et al. (2016) suggest to select 1000 random indices in the cache that do not belong to any putative images (we ensure this by selecting images that are more than 25 m from the query image). We then insert in the triplet constraint (Equation 4.4) and calculate how much each negative descriptor violates the constraint. We select the k negative indices, which descriptors violate the constraint the most and use these as our hard negatives. This ensures that the network keeps learning. Furthermore, Arandjelovic et al. (2016) state that including the hardest negatives from the previous epoch stabilizes the convergence.

It is important to note that the cache will consist of descriptors from an outdated network. We can express this as

$$d = f_{t-T}(I) \tag{4.7}$$

where d is the cached descriptor of image I calculated by the network, f, at time $t - T$, where t is the current time and T is the time when the cache was last updated. If the time since the descriptor was last updated $t - T$ becomes too large, the model will begin overfitting to the cache. These outdated features will result in a trivial loss and the model will fail to improve. Therefore, it is required to update the cache regularly throughout the training.

Ideally, we would prefer to update the cache only once per epoch such that we could simply cache the descriptors calculated in the forward passes in training. This way the cache would only result in a memory overhead. However, in practice, using one epoch old descriptors causes the model to overfit to the cache and provides a trivial loss. Therefore, Arandjelovic et al. (2016) suggest to update the cache regularly every 1000 iteration. Updating the cache regularly, and independently of the training, obviously creates a substantial overhead in training time. In practice, this means that the time spent on updating the cache scales linearly with the size of the dataset, making it infeasible to train place recognition models on very large datasets, such as Mapillary SLS. Therefore, we propose a novel sub-caching method.

4.4.3 Sub-Caching Method

We propose a simple, yet effective sub-caching method with constant time and space use. We propose to divide the entire dataset into equally sized and randomly sampled subsets. For each query image in a subset, we add all its positive images to the given subset. We then update the cache for this subset, making the cache-time scale with respect to the subset size rather than the size of the entire dataset. Both the query and positive images can be sampled from the cache as well as negatives. It is important to keep the subset size large enough to find adequately hard triplets. In our experiments, we use $10,000$ query images and refresh the cache every $1,000$ iterations. We use 5 negative examples per triplet instead of 10 (Arandjelovic et al., 2016) as this allow us to fit a batch size of 4 into memory on a single GPU.

Furthermore, it is important to update all regions of the embedding space. As the triplet loss is a local approximation of the recall, we are not guaranteed to update the embedding space everywhere. We found that in the MSLS dataset, the images taken at night and images that are sideways facing are underrepresented compared to forward looking images taken during daytime. This meant that our model performed poorly on these underrepresented image classes. Through empirical studies, we found that using a weighted sampling of underrepresented classes during training significantly improves the convergence speed and generalization capabilities across different datasets. We use the curated image tags to weight these underrepresented classes by the inverse of their occurrence when sampling examples to the sub-cache. This results in a sub-cache with equally amount of sideways/front-facing and day/night images.

In this chapter, we have presented the methods, experimental setup and training procedure use to validate and benchmark the Mapillary SLS dataset. In the next chapter, we report and discuss results of these methods.

CHAPTER 5

Results and Discussions

In this chapter, we show quantitatively and qualitatively that training models on the Mapillary SLS dataset improves their generalization capabilities and reduces their geographical bias. We show that exploiting sequential information can further improve the model performance. Lastly, we conduct a qualitative evaluation of our trained models. We visualize the triplets mined during training, investigate failure cases, explore the network attention and the learned embedding space. The chapter is divided into two sections; first quantitative results followed by qualitative results.

5.1 Quantitative Results

	Model	Training set	Base	Input Size	Dim	Su/Wi	Wi/Su	Da/Ni	Ni/Da	Ol/Ne	Ne/Ol	All (@1/5/10)
im-2-im	Amos	SPED	CaffeNet	227x227	2543	0.17	0.09	0.20	0.09	0.17	0.14	0.06/0.11/0.14
	Hybrid	SPED	CaffeNet	227x227	2543	0.13	0.11	0.14	0.11	0.18	0.17	0.08/0.13/0.15
	NetVLAD	Pitts250k	VGG16	640x480	32768	0.43	0.44	0.37	0.09	0.49	0.50	0.28/0.35/0.39
	GeM	SfM-120k	VGG16	640x480	2048	0.51	0.48	0.37	0.20	0.55	0.56	0.30/0.40/0.44
	NetVLAD	SLS	VGG16	640x480	32768	**0.76**	**0.74**	**0.49**	**0.23**	**0.71**	**0.75**	**0.48/0.58/0.64**
seq-2-seq	NetVLAD + MAX	Pitts250k	VGG16	640x480	32768	0.40	0.51	0.37	0.09	0.55	0.57	0.23/0.32/0.36
	NetVLAD + AVG	Pitts250k	VGG16	640x480	32768	0.41	0.39	0.37	0.09	0.54	0.54	0.20/0.31/0.34
	NetVLAD + CAT	Pitts250k	VGG16	640x480	98304	0.44	0.47	0.37	0.14	0.57	0.56	0.23/0.33/0.37
	GeM + MAX	SfM-120k	VGG16	640x480	2048	0.53	0.54	0.43	**0.26**	0.67	0.57	0.29/0.43/0.48
	GeM + AVG	SfM-120k	VGG16	640x480	2048	0.60	0.52	0.40	0.14	0.66	0.57	0.29/0.42/0.46
	GeM + CAT	SfM-120k	VGG16	640x480	6144	0.55	0.46	0.46	**0.26**	0.65	0.53	0.28/0.42/0.46
	NetVLAD + MAX	SLS	VGG16	640x480	32768	0.75	**0.79**	0.51	0.14	**0.80**	**0.76**	0.42/0.58/0.63
	NetVLAD + AVG	SLS	VGG16	640x480	32768	0.75	0.78	0.51	0.06	0.78	0.73	0.37/0.56/0.60
	NetVLAD + CAT	SLS	VGG16	640x480	98304	**0.84**	0.76	**0.57**	0.20	**0.80**	0.72	**0.41/0.60/0.65**
seq-2-im	NetVLAD + MIN	Pitts250k	VGG16	640x480	32768	0.53	0.53	0.37	0.03	0.60	0.62	0.30/0.37/0.40
	NetVLAD + MODE	Pitts250k	VGG16	640x480	32768	0.53	0.51	0.46	0.06	0.61	0.59	0.28/0.37/0.41
	GeM + MIN	SfM-120k	VGG16	640x480	2048	0.62	0.62	0.37	0.23	0.71	0.67	0.38/0.47/0.50
	GeM + MODE	SfM-120k	VGG16	640x480	2048	0.59	0.52	0.46	**0.26**	0.67	0.66	0.32/0.45/0.51
	NetVLAD + MIN	SLS	VGG16	640x480	32768	**0.86**	**0.86**	**0.54**	0.20	**0.83**	**0.81**	**0.56/0.68/0.71**
	NetVLAD + MODE	SLS	VGG16	640x480	32768	0.53	0.51	0.46	0.06	0.61	0.59	0.28/0.37/0.41
im-2-seq	NetVLAD + MIN	Pitts250k	VGG16	640x480	32768	0.20	0.30	0.29	0.14	0.33	0.28	0.12/0.20/0.26
	GeM + MIN	SfM-120k	VGG16	640x480	2048	0.24	0.22	0.26	**0.31**	0.37	0.29	0.13/0.22/0.31
	NetVLAD + MIN	SLS	VGG16	640x480	32768	**0.45**	**0.39**	**0.31**	0.23	**0.48**	**0.37**	**0.23/0.34/0.48**

Table 5.1: Evaluation of im-2-im, seq-2-seq, seq-2-im and im-2-seq models on Mapillary SLS test set. We report the models recall@5 on several challenging recognition cases as well as their overall recall@1/5/10. For a fair comparison, we compare models with similar backbone architecture.

Table 5.1 shows a comparison of models evaluated on the SLS dataset. For a fair comparison, we present models with a similar base network. We show both single-view and multi-view models in the same table for easier comparison. Particularly, we benchmark models for im-2-im, seq-2-seq, im-2-seq and seq-2-im recognition, reporting their top-5 recall on several challenging recognition cases as well as their top-1/5/10 recall on the entire test set. These challenging cases include summer to winter (Su/Wi), day to night (Da/Ni), old to new (Ol/Ne) and vice versa. We define old images as those taken between 2011–2016 and new images as those taken since 2018. The goal is to separately evaluate the performance of each method when exposed to seasonal, day/night and structural changes.

Single-View Place Recognition: Table 5.1 shows that the NetVLAD and GeM models significantly outperform AmosNet and HybridNet. It is worth noting that AmosNet and HybridNet operates on smaller input images and relies on a smaller base architecture. We see that the pre-trained GeM model (trained on SfM-120k) performs slightly better than the pre-trained NetVLAD model (trained on Pittsburgh250k). The GeM model seems to better capture the day/night variations, which might be caused by a better embedding of geometric information stemming from the training on 3D reconstructions. Furthermore, we see that training the NetVLAD model on the diverse SLS improves the overall performance (NetVLAD top-5 recall improves 13% on our test dataset), making it the best performing single-view model on our test set. The performance boost is mainly caused by improved capabilities to recognize places that have undergone seasonal and temporal changes. We see that all the models are especially challenged by the night to day changes, which intuitively makes sense as there are very large visual changes between night and day as highlighted with examples in Appendix A.2.

Multi-View Place Recognition: Table 5.1 also shows that simple pooling strategies such as max and average pooling do not improve performance compared to single-view models. We see that concatenation only improves the model performance slightly. The reason is that SLS sequences are captured at different frame-rates and user velocities, requiring the model to learn a time-independent relation between the frames. This complex relations cannot be captured by simple pooling strategies. We see that using a strong single-view model before pooling the last layer improves the performance. This motivates the development and further research in multi-view methods, which is accommodated by release of the Mapillary SLS.

In Table 5.1, we further benchmark the performance of the propose solution to both seq-2-im and im-2-seq challenges. For seq-2-im, we found that matching based on the minimum distance between frames in the given sequence gave superior performance to looking at the mode. We found this method to be extremely effective, beating the best performing im-2-im and seq-2-seq models. This method encapsulates more information of a scene than im-2-im methods, as it is given a sequence, and operates on frame-level, such that it does not try to model the non-linearity between the frames in a sequence, which makes this model perform surprisingly well.

Lastly, Table 5.1 shows the performance of im-2-seq models. This problem seems to be more challenging to model. However, we see that using a good performing single-view model as base improves the performance.

Detailed Comparison: In Table 5.1, we fixed the distance threshold for positives to 25 m and looked at the top-1/5/10 recall. In Figure 5.1, we investigate the performance of the single-view models when varying these thresholds. The figure shows a detailed comparison of the best performing im-2-im models with varying distance and number of image candidates. Again, we see that training NetVLAD on our dataset improves the model performance. Furthermore, we see that the ordering of the models is the same for all distance thresholds and that after the 25 m threshold, the curves are approximately parallel. These observations justify the choice of a distance threshold of 25 m for selecting positive matches.

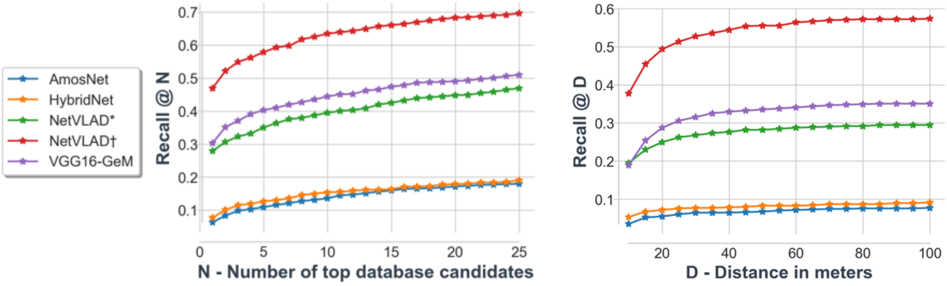


Figure 5.1: Recall of best performing im-2-im methods on Mapillary SLS as a function of number of nearest neighbors (left) and distance threshold (right).

In this subsection, we have shown that day/night changes are especially challenging as the scene undergoes significant visual changes. We found that training on the SLS dataset can improve model performance and that utilizing sequential information can further boost model performance. We saw that simple pooling techniques do not capture the non-linear dependencies between the frames in a sequence, but concatenating the embedding ensure more robustness. Finally, we found that our proposed seq-2-im minimization method to work surprisingly well. This method did not seek to model the inter-frame structures, but exploited the temporal information from the sequence data to retrieve place with high a recall. Next, we show that training on the more diverse MSLS dataset improves models generalization capabilities. As existing dataset are curated for single-view models, we will perform this comparison only for single-view models.

5.1.1 Improved Generalization Capabilities

We show that the increased size and diversity of our dataset can improve models' generalization capabilities. We compare four NetVLAD models: A version trained on the Pittsburgh250k dataset (NetVLAD (Pitts250k)) and a version trained on the SLS dataset (NetVLAD (SLS)). These are trained with the same training procedure. We further include results from a model trained on SLS with our proposed positive mining strategy (NetVLAD† (SLS)), and lastly a model trained on SLS with our mining strategy followed by fine-tuning on Pittsburgh250k (NetVLAD† (SLS + Pitts250k)). We justify the improved generalization by evaluating on several popular datasets (see Table 5.2). Observe the smoother cross-dataset performance of the models trained on Mapillary SLS.

From Table 5.2, we see that the model trained on Pittsburgh seems to overfit to the Pittsburgh environment as it achieves a high recall for the Pittsburgh datasets but relatively low recall for the SLS, RobotCar and Tokyo24/7 datasets. As shown in Section 2.3, there is a visual overlap between the training and test set of the Pittsburgh and Tokyo datasets. We presented the Revisited-Pittsburgh250k test set, where the images with visual overlap in the test set are excluded, however, evaluation on these revisited datasets result in the same ordering of the models. This might suggest that the model overfit to the image collection procedure, image artifacts or very specific features from the Pittsburgh environment rather than the specific places near the border of the original test set.

Table 5.2 also shows that all the models trained on the SLS dataset perform better on the SLS, Tokyo24/7 and RobotCar datasets than the model trained on Pittsburgh250k. From Table 5.2, we show that fine-tuning the model on the Pittsburgh250k dataset after training on the SLS dataset results in the most general features. This model has the highest or second highest recall across all datasets. This shows that the increased size and diversity of the Mapillary SLS dataset helps the model learn a better feature representation for place recognition and improve its generalization capabilities. We note that our proposed positive mining strategy seems to improve model performance slightly across all datasets. We found experimentally that the positive mining strategy was important for good fine-tuning performance.

Lastly, note that top-5 recall is significantly lower on the SLS test set, which is a measurement of its higher degree of difficulty. This shows that current state-of-the-art place recognition models still have a long way to go, and that the diversity and challenge of existing datasets does not sufficiently reflect real-world use-cases.

	NetVLAD (Pitts250k)	NetVLAD (SLS)	NetVLAD† (SLS)	NetVLAD† (SLS + Pitts250k)
TokyoTM	**0.98**	**0.98**	**0.98**	**0.98**
Tokyo24/7	0.72	0.75	0.76	**0.81**
Pitts250k	**0.91**	0.87	0.87	0.90
Pitts30k	**0.91**	0.89	0.90	**0.91**
R-Tokyo24/7	0.71	0.74	0.75	**0.81**
R-Pitts250k	**0.91**	0.87	0.88	0.90
R-Pitts30k	**0.93**	0.91	0.92	**0.93**
RobotCar	0.74	**0.81**	**0.81**	**0.81**
SLS	0.35	0.44	**0.47**	**0.47**

Table 5.2: Evaluation on test sets of popular datasets. We compare the top-5 recall of NetVLAD trained on Pittsburgh250k, NetVLAD trained on SLS, and NetVLAD trained on SLS and fine-tuned on Pittsburgh250k. † indicates that we apply our proposed positive mining strategy during training to sample more day/night and sideways-facing triplets.

We achieved state-of-the-art performance by training our model on the SLS dataset with our proposed positive mining strategy and fine-tuning on the Pittsburgh dataset. We found that this model had the best performance across several popular datasets for place recognition. Note that all presented models so far have relied on the VGG16 base architecture. In the next subsection, we investigate whether an alternative backbone architecture can further improve the model.

5.1.2 Comparison of Base Network

We compare the NetVLAD layer with different base networks. We train all models with the same training procedure using our proposed positive mining method. Models are trained with early stopping. Depending on the base architecture training took between one and three days. We evaluate the models on Revisited Tokyo, Revisted Pittsburgh, RobotCar and SLS to understand the cross dataset performance.
It is important to note that the final feature map size of ResNetX is half the size of the one of AlexNet and VGG16. We found experimentally that training a ResNetX base followed by NetVLAD layer on 640x480 input images did not improve performance during training. The reason is that the last feature map size of the ResNetX was too small to be a good feature to describe the appearance of a place. Therefore, we upsampled the input images of the ResNetX before feeding the images to the network as this would also double the final feature map size. However, for a more fair comparison, we decided not to train on higher resolution image. This increased feature map size meant a larger spatial coverage of the image, which seem to be very important for the NetVLAD layer to learn a good embedding for place recognition.

Base	Dim	R-Pitts30k	R-Pittsb250k	TokyoTM	R-Tokyo247	SLS	Robotcar
AlexNet	256	0.69	0.57	0.94	0.47	0.32	0.71
VGG16	512	0.86	0.81	**0.97**	0.65	0.42	0.78
ResNet18	2048	**0.89**	**0.85**	0.95	0.62	0.53	0.78
ResNet50	2048	0.88	0.84	**0.97**	**0.68**	**0.58**	**0.81**
ResNet101	2048	0.83	0.78	0.94	0.64	0.54	0.77

Table 5.3: Evaluation on different base architectures. We compare the top-5 recall. Dim refers to the number of channels fed to the NetVLAD layer. The final output dimension is the number of channels times 64, which is the number of clusters in the NetVLAD layer.

From Table 5.3, we see that AlexNet has the worst performance. This is the smallest and fastest network, so it is not surprising that this base architecture is a poor choice, when our evaluation criterion is recall. In applications where memory footprint or speed is an requirement, this base architecture might be more relevant. We found that ResNet50 is the best performing model across most datasets. This is surprising as we would expect the deeper ResNet101 to encapsulate more information about a place. We see that in general, the ResNet base architecture is better than the VGG16.

This correspond to what is usually observed for image classification (Canziani et al., 2016).

In this section, we quantitatively evaluated several deep models for place recognition. We beat state-of-the-art performance on several datasets by training on the larger and more diverse MSLS dataset. Furthermore, we showed that utilizing sequential information can further improve model performance. Lastly, we found that using a ResNet base architecture can also further improve model performance. We will in the next section qualitatively analyse the training procedure and investigate the trained models.

5.2 Qualitative Results

In this section, to better understand the strength and the diversity of the SLS dataset and gain a deeper insight into the performance of our trained networks, we present a qualitative analysis of the recognition models.

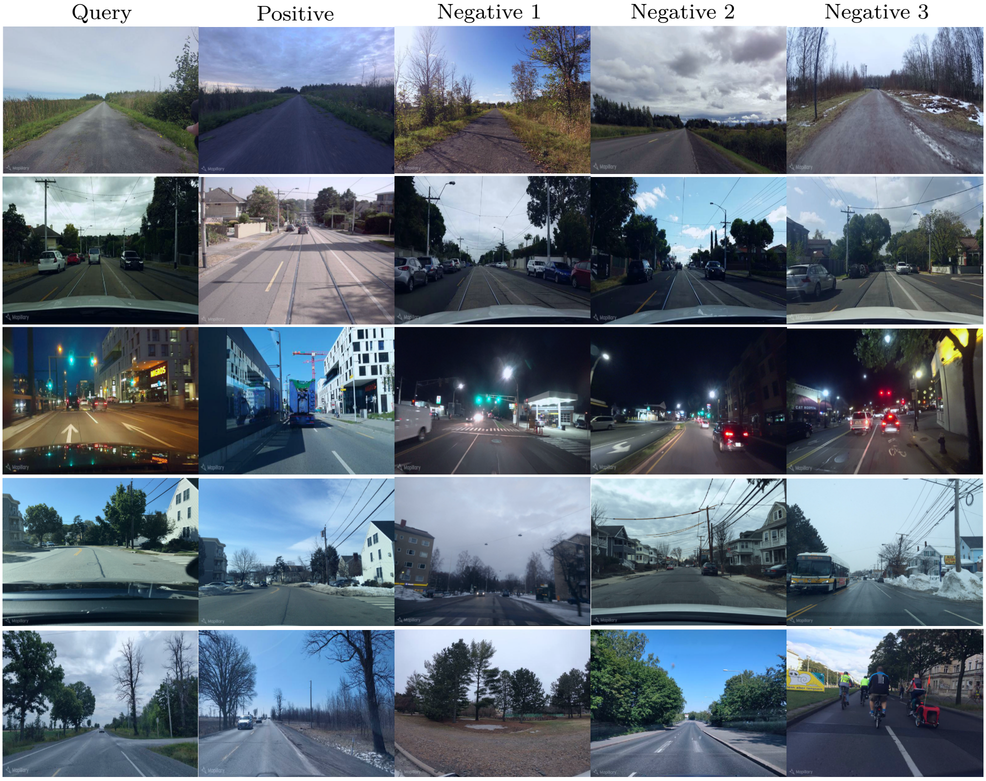


Figure 5.2: Triplets with multiple negatives. Hard-negatives are mined during training using our proposed sub-caching methodology. For each query-positive pair, the negatives that violate the triplet constraint most, $||q-p||_2^2 + m < ||q-n||_2^2$, are chosen. Here q, p, n refer to the cached embeddings for the query, positive and negative images. m is the margin.

5.2.1 Triplets

The models are trained using the triplet loss combined with hard negative and positive mining. The triplet constraint has to be violated to get a non-trivial loss. Therefore, it is important to mine negatives that are closer to the query image than its positives. In other words, the triplet loss is a local approximation of the recall. Therefore, we

wish to mine negative images that appear very similar to the query image. In Figure 5.2 we show triplets that violates the triplet constraint during training. Note that the negative images are very similar to the query image and that the positive images, in terms of seasonal, weather and illumination, are not necessary similar to the query image. For example we see that if the query image is captured at night, then its hard negatives are often also captured at night.
In Figure 5.3 we show an examples of triplets mined for sequence to sequence training. These triplets are found with the same procedure as for single-view. Also, for the sequence triplets, we see that challenging triplets are mined during training. See Appendix A.3 for more triplet sequence examples.

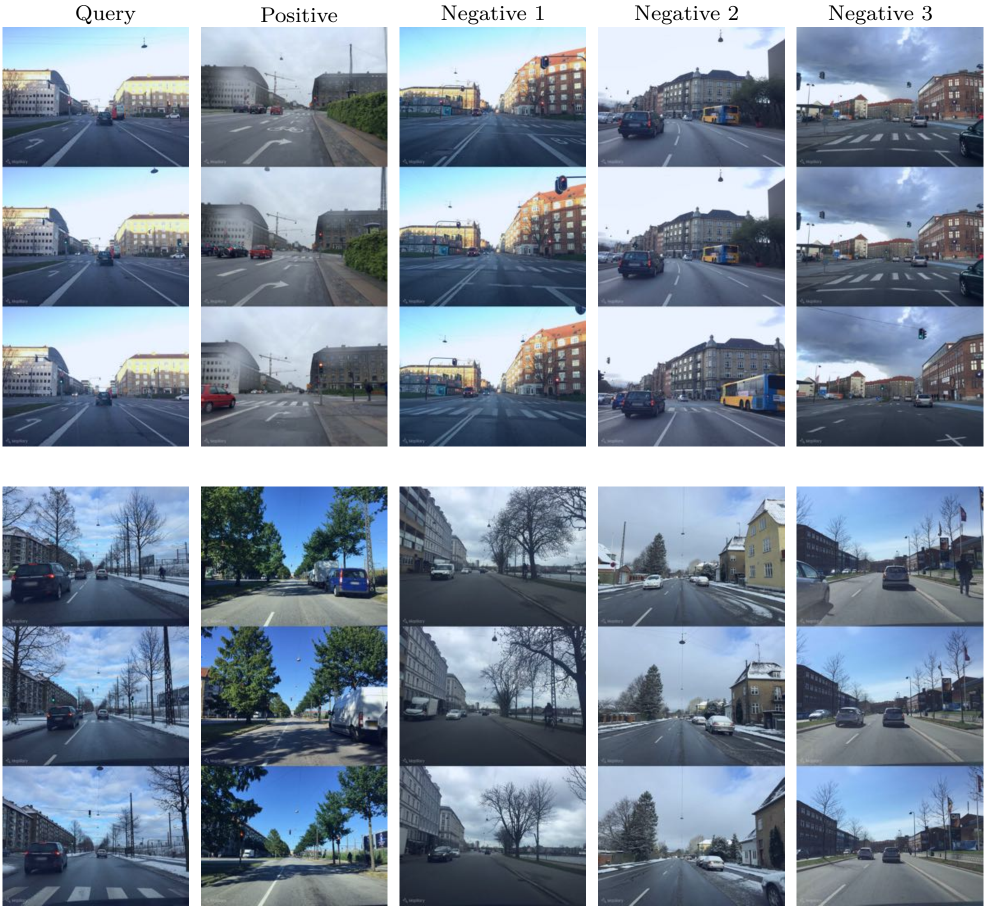


Figure 5.3: Sequence triplets with multiple negatives mined with the same training procedure as the in the image to image case. Note the drastic seasonal and weather changes between the query and positive images.

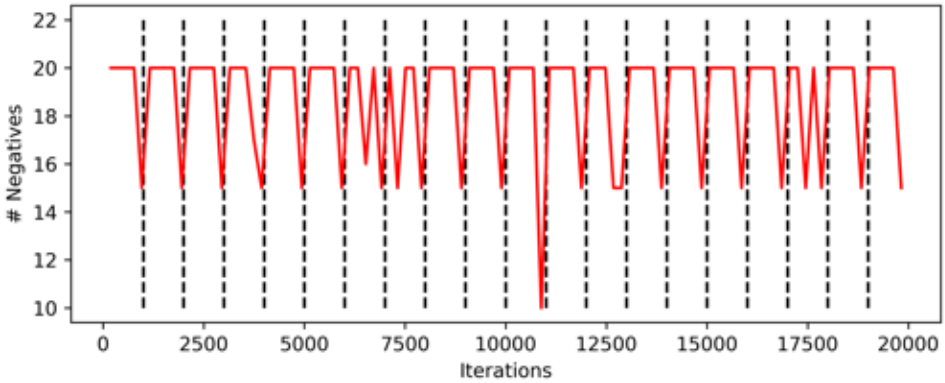


Figure 5.4: The red curve shows the number of non-trivial negatives found during training. We use a batch size of four triplets and five negatives per query image. Thus, the maximum number of non-trivial negatives is 20. The vertical black lines indicate when the cache was updated. Note that this figure only shows the first $20k$ iterations to highlight the function's oscillating behavior.

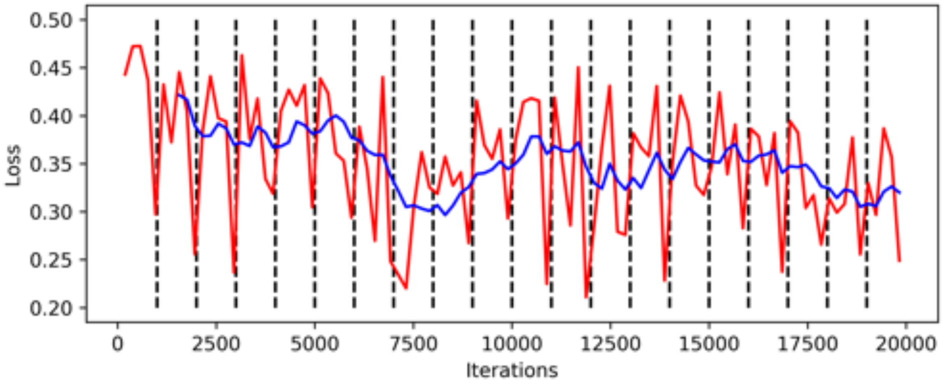


Figure 5.5: The red curve shows the triplet loss during training. The blue line is a moving average of the loss, and the vertical black lines indicate when the cache was updated.

It is important to update the cache frequently. In Figure 5.4, we show the number of non-trivial negatives (Equation 4.5) found during training. In this experiment, we used a batch size of four triplets and five negatives per query example. Thus, the maximum number of non-trivial negatives was 20. From the figure, we see that the number of non-trivial negative oscillates. Each time the cache is updated (indicated by the black vertical lines), the number of non-trivial negatives increases. As the cache gets outdated, the model begins to overfit to the cache, and the number of non-trivial negatives decreases. We found that updating the cache every 1000 iteration was a good compromise between finding a large number of non-trivial negatives and reducing the overhead of updating the cache.

Figure 5.5 shows the triplet loss during training. We note the same oscillation behavior as in Figure 5.2 because the model begins to overfit slightly to the cache. Furthermore, we see that the decrease in average triplet loss (blue line) is rather low compared to what is usually observed for global loss functions and standard training procedures. The reason is that the difficulty of the cached examples will increase in order to violate the triplet constraint as the model improves. This leads to an increasingly difficult optimization landscape, which is why the average loss decreases very slowly.

After this qualitative analysis of the conducted training procedure, we will in the next subsection visually inspect the models' ability to retrieve imagery from the same geographical location.

5.2.2 Model Comparison

In Table 5.1, we showed quantitatively that NetVLAD trained on the SLS dataset beat state-of-the-art on the SLS dataset. In this subsection, we visually inspect the predictions of all the single-view models from this table. In Figure 5.6, we qualitatively evaluate AmosNet, HybridNet, GeM and NetVLAD trained on respectively Pittsburgh250k and Mapillary SLS. From this figure, we see that both AmosNet and HybridNet have many erroneous predictions (highlighted by the red bounding box). We see that both these models' predictions have rather similar weather conditions to the query images. This suggests that neither of the models have learned to be invariant towards weather changes, which is very important for recognizing places. This also seems to be challenging scenarios for NetVLAD model trained on Pittsburgh250k. In contrast, both GeM (trained on SfM-120k) and NetVLAD (trained on SLS) are more invariant towards weather changes, e.g. in row $3-5$ in Figure 5.6, NetVLAD (trained on SLS) retrieves the correct place even though there are significant weather changes. Furthermore, we notice how the diversity of the Mapillary SLS data makes NetVLAD more robust to viewpoint changes and exotic vegetation, such as the palm trees, compared to models that are trained on other datasets that do not encapsulate as much diversity as our dataset.

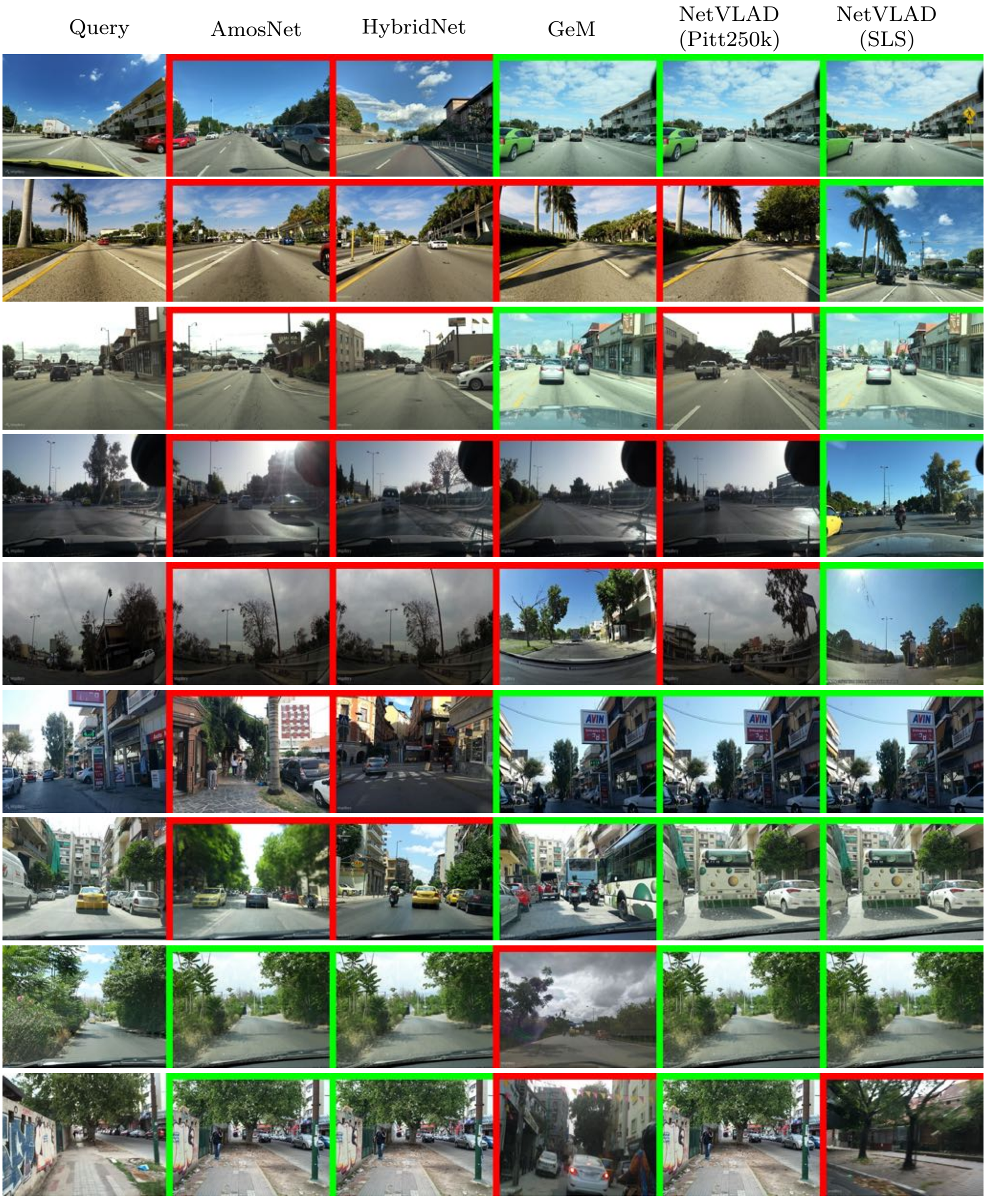


Figure 5.6: Qualitative comparison of different pre-trained networks as well as our NetVLAD model trained on Pittsburgh250k and Mapillary SLS. Training on SLS improves robustness towards weather changes and diverse vegetation such as palm trees. Green: true positive; Red: false positive.

5.2.3 Network Attention

To gain a better understanding of what the networks have learned, we visualize the layer activations at different depth of the NetVLAD model with VGG16 base. We visualize the activation across different input images to explain the networks' attention. We expect the model to have higher activation around stationary objects in the scene such as buildings and lower activation around changing objects such as vehicles, vegetation and roadwork. In Figure 5.7, we show the activation for several filters at different depths in the model.

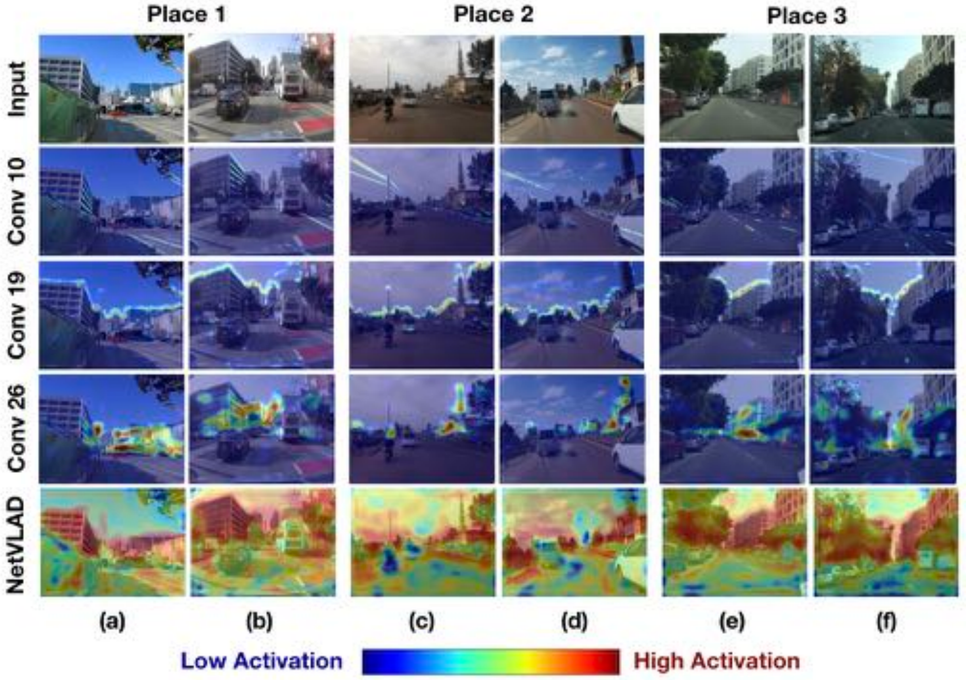


Figure 5.7: The top row shows the input images. The remaining rows shows the feature activation for filters from convolutional layers 10, 19, 26 and the NetVLAD layer. These are visualized as heatmaps overlaid on top of the input image (see Appendix A.4 for original feature maps). Blue indicates low activation and red indicates high feature activation.

From Figure 5.7 we see that the filter from convolutional layer 10 fires at longer diagonal lines, such as those found at the power cables in (c, d, f), the building facades in (a, b) and on the road paint in (a, b, e, f). These line features are simple, low-level features, which coincides with the fact that low level features are typically found in the early convolutional layers (Zeiler and Fergus, 2013). We see that the filter from convolutional layer 19 fires around the horizontal contour of the buildings. This more general feature is important for place recognition as the contour of the building is typically a non-changing feature across seasons and weather. Note that when there are small viewpoint changes such as between (e) and (f) this feature is effective for recognizing a place. Even more general features are found in convolutional layer 26. This filter activation fires on buildings. Notice that this activation seem to be invariant to viewpoint and scale changes as it fires in the same image regions for each of the image pairs. Examples are the church tower in (c, d) and the building facades in (a, b) and (e, f). Surprisingly, this filter also fires at the scooter driver's head in (c). This is a undesirable feature as we would expect the model to ignore dynamic objects. This might also suggest that using a segmentation model to filter out dynamic object might improve performance. Finally, we find the most general feature activation in the NetVLAD layer. This layer also fires on buildings and has lower activation around seasonal objects (such as trees) and dynamic objects (such as scooters, cars and buses), which is inline with the behavior that we expect the model to have learned.

5.2.4 Visualization of Learned Embedding

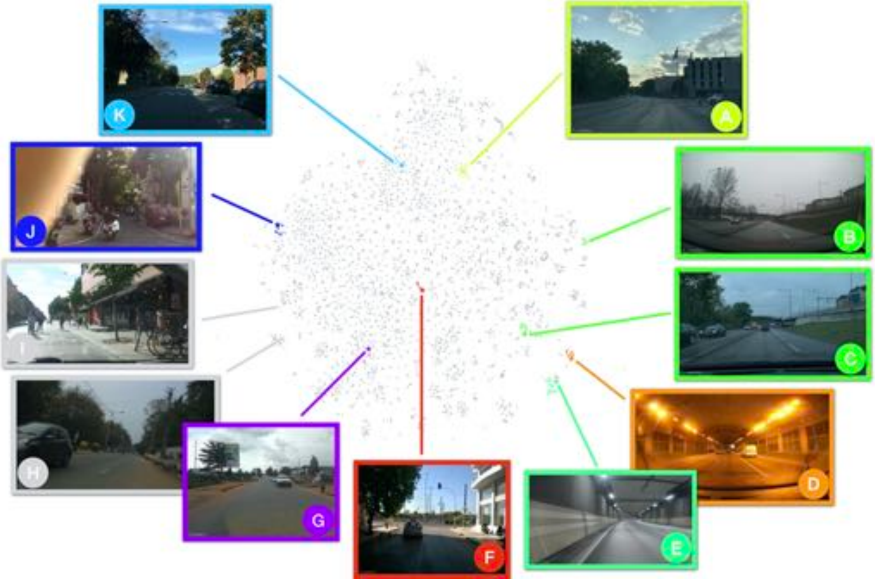


Figure 5.8: Visualization of the learned embedding space. 5000 images from the test set are projected into a 2-D space using T-SNE. 8 randomly selected query images are highlighted with a star and their positives are colored in the same color as them. We highlight 11 image examples $A \dots K$ from the embedding space.

We investigate the learned embedding space to get a better understanding of what the model has learned. We randomly select 5000 images from the test set, calculate their position in the 32.768 dimensional embedding space, and then use T-SNE to project these points into a two dimensional plane. In Figure 5.8 we show that the model has learned that the query examples are close to their positives. The only exception is the green cluster - highlighted with the two images B and C. We see that this scene has significant seasonal, viewpoint and weather changes - making it a challenging scenario. Furthermore, the reflection in the front window makes the example even more challenging. Below these images, we highlight two query images, D and E, captured inside a tunnel. These query images are close to their respective positives, but also close to each other in the embedding space. This intuitively makes sense as we hope that images that appear similar are close in the embedding space. Images G, H and J are captured from Bengaluru and Kampala. It seems that the model cluster these cities in the lower left corner. This is further confirmed in Figure 5.9. We see that image I, which is taken in Stockholm, is also found in the lower left corner of the embedding space. This image is very blurry - suggesting that the model might have picked up the artifact that the imagery from Bengaluru and Kampala are more often in lower resolution and more blurry than the imagery from more developed cities. Also, this image contains many bicycles in the middle of the road, which the model might confuse with the many scooters often spotted in Bengaluru and Kampala.

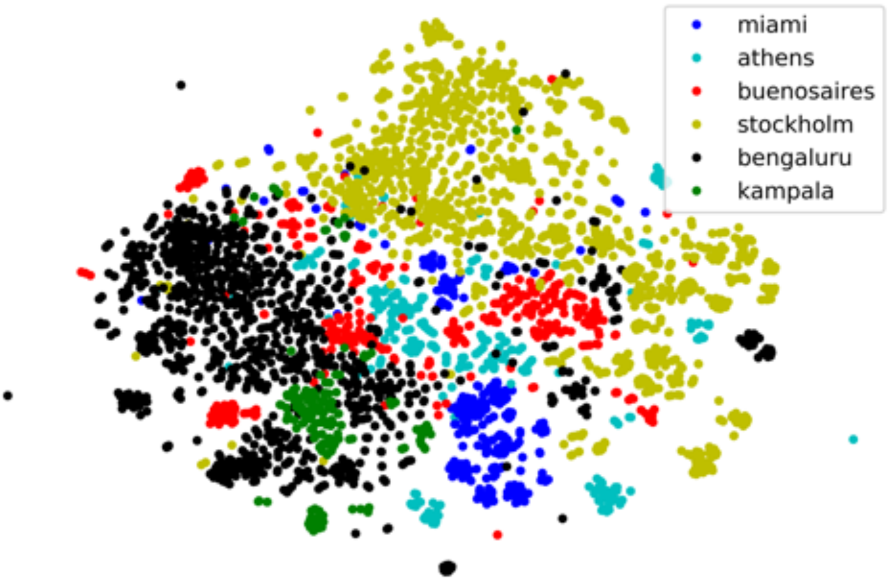


Figure 5.9: Visualization of the learned embedding space color-coded by city. We see that the cities tend to cluster together and cities we would expect to be more similar, such as Kampala and Bengaluru, are also close in the embedding space.

In Figure 5.9, we highlight the models ability to cluster cities. The figure shows the same embedding space as visualized in Figure 5.8, but color-coded by city. We see that cities we would expect to be more similar, such as Kampala and Bengaluru, are also closer in the embedding space. This clustering by cities hints that a hierarchical approach, where a classifier is trained to classify city or region followed by a place descriptor might enable a better place descriptor and a reduction of the number of model parameters. We will in the following subsection, further investigate the model performance across different parts of the world to map out the models' geographical biases.

5.2.5 Reduce Geographical Bias

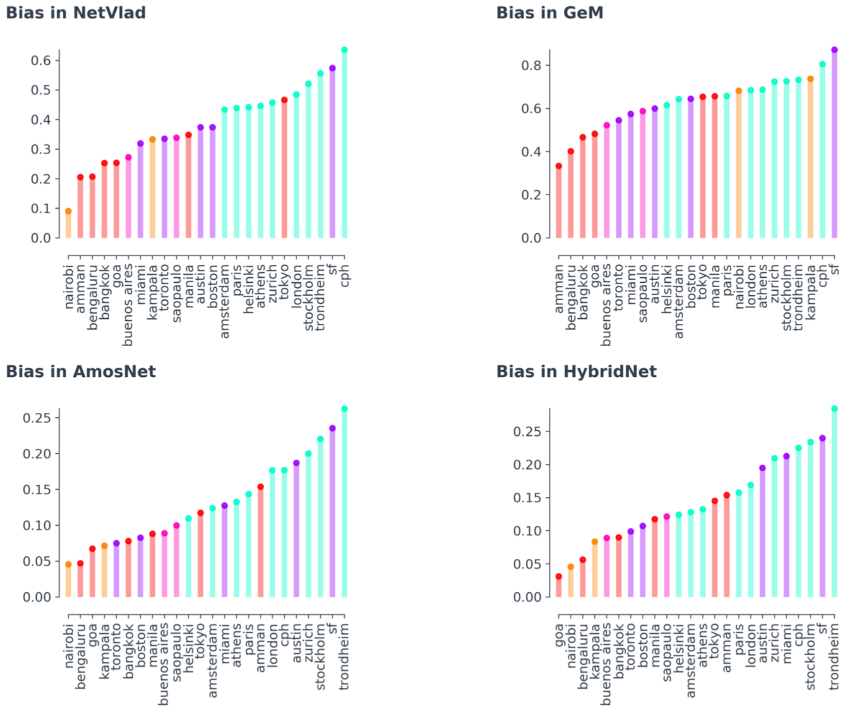
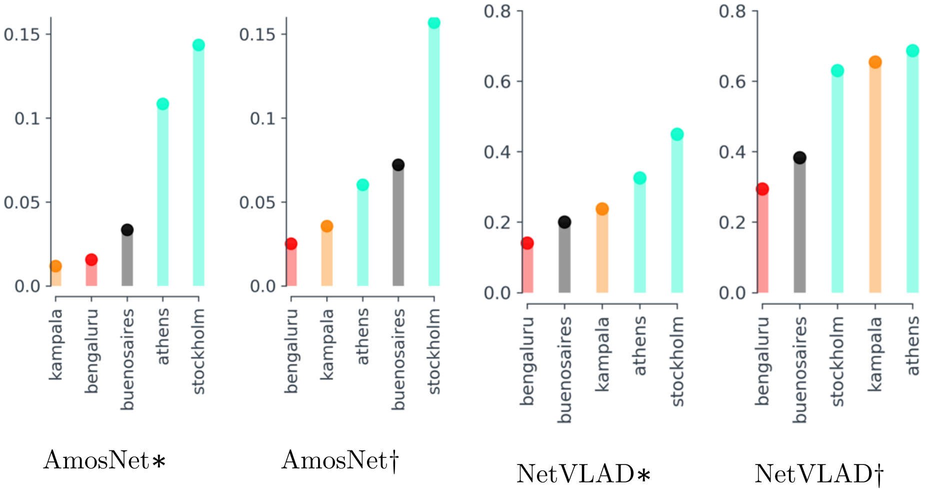


Figure 5.10: Geographical bias in four place recognition models. y-axis shows the top-5 recall. Cities are colored depending on their location: Africa (orange), Asia (red), South America (pink), North America (purple), Europe (turquoise).

State-of-the-art place recognition networks are trained on images from developed countries. This means that there is an obvious geographical bias inherent in the data. The bias in the data leads to a bias in the models trained on the data. A simple method to check for this bias is to evaluate the model on different subsets of the dataset. Thus, in order to investigate geographical bias, we can evaluate the model on individual cities. By this measure, an unbiased model would perform equally well across different cities and continents. Figure 5.10 shows the performance of several models evaluated on individual cities from Mapillary SLS. From the figure, we see a clear geographical bias as the models perform significantly better on European and North American cities compared to Asian, South American and African cities. The only exception is Tokyo, however, Tokyo is a develop city, similar to many American cities. Further notice that the GeM model has slightly less geographical bias, since the performance drop on Asian cities is relatively low. This could be related to the fact that this model is trained on Flickr images, which are more diverse, in terms of geographical coverage, than other datasets.



AmosNet∗ AmosNet† NetVLAD∗ NetVLAD†

Figure 5.15: Bias reduction for both AmosNet and NetVLAD when trained on SLS and evaluated on the SLS test set. ∗/† indicates pretrained/trained models, respectively.

Place Recognition models with geographical bias can have catastrophic consequences in real world applications, e.g. for a global fleet of autonomous vehicles, we must ensure that the software works equally when across countries and continents. In Figure 5.15, we show that the geographical bias in both AmosNet and NetVLAD can be reduced by training on the SLS dataset. We show that the large geographical coverage of the Mapillary SLS dataset enables evaluation of model biases. However, the Mapillary SLS dataset is still largely over-representing European and North American cities (see Table 3.1), thus in order to mitigate geographical bias completely practitioners and researchers must be considerate, e.g we see from Figure 5.15 that the models still have a large geographical bias. These models are trained on the entire SLS dataset with uniform sampling across cities and continents. We expect that weighting African, South American and Asian cities higher, using our proposed positive mining methodology, could further reduce the model bias.

In this section, we investigated the trained models qualitatively. We analyzed the training procedure by visually inspected the mined triplets and highlighted the importance of updating the cache frequently to avoid overfitting to the cache. We found that updating the cache every 1000 iteration to be a good compromise between having many non-trivial negatives and a low computational overhead. We made a qualitatively model comparison of several single-view models and found that training on the SLS dataset improved models' invariance towards viewpoint and weather changes as well as improving their abilities to recognize places based on exotic vegetation such as palm trees. By visualizing a trained model's feature maps, we found that it focuses on stationary objects in the scene, confirming our expectation of what important objects for place recognition would look like. Via a T-SNE projection, we were able to visualize the learned embedding space in 2D. We found that imagery from the same places appear close in the embedding space, and that places that share visual content (such as tunnels) also appear close in the learned embedding space. We further found that the cities were clustered together in the embedding space. Lastly, we have shown that existing place recognition models inherits the geographical bias from their training data. We propose a simple, qualitative method to evaluate the model bias and showed that training on a more diverse dataset with larger geographical coverage can reduce the model bias.

CHAPTER 6

Conclusion

Through a comprehensive literature review and analysis of existing datasets, we discovered that current datasets for place recognition are limited in terms of size, geographical coverage and diversity. Many of these datasets are curated for single-view models, making it difficult to train multi-view models. Furthermore, we found that the largest and most diverse of these datasets, the TokyoTM/247 and Pittsburgh250k, have a substantial visual overlap between their training and test set, making the evaluation on these dataset unreliable for models trained on them. We proposed a new division with Revisited-Tokyo24/7 and Revisited-Pittsburgh30k/250k, but also identified a need for a larger, more diverse and sequential structured dataset for place recognition with a world-wide geographical span.

We have presented Mapillary SLS, a large collection of image sequences for training and evaluating place recognition algorithms. The data has been collected from Mapillary and contains over 1.6 million frames from 30 different cities over six continents. The gathered sequences span a period of seven years and places experience large perceptual changes due to seasons, constructions, dynamic objects, cameras, weather and lighting.

Mapillary SLS contains the largest geographical and temporal coverage of all publicly available datasets; and it is among the ones with the widest range of appearance variations and the largest number of images. All these features makes our dataset a valuable addition to the available data corpus for training place recognition algorithms. The many variation modes and the considerable size of realistic urban data make it particularly appealing for deep learning approaches and autonomous car applications.

We have run extensive benchmarks on our dataset with existing state-of-the-art methods to illustrate the difficulty and contribution of our dataset. By training on the very diverse MSLS dataset, we improve state-of-the-art models across numerous existing datasets including RobotCar and Tokyo247 where our best performing model performed respectively 9% and 13% better with a comparable model architecture. These results were achieved through a simple, yet effective sub-caching method that enable training on the very large MSLS dataset. Furthermore, we proposed a hard positive mining scheme that sampled sideways facing and night images more frequently, resulting in faster cross dataset generalization. The reason is that the MSLS dataset have a rather low percentage of night and sideways facing images (still a very large amount) so an uniformly sampling would rarely present the model for triplets containing either sideways facing nor night imagery. We found this sampling method to be effective and believe it to be easily extendable to include other meta data such as road type, city or country.

We experimented with using different base architectures, and found empirical that the ResNet architecture gave better performance than both the VGG16 and AlexNet. Particularly, we found the ResNet50 to perform well across multiple datasets.

In this thesis, we also presented several variations of MultiViewNet and showed that exploiting the sequential information between frames can improve place retrieval. We found that simple pooling strategies, such as max and average pooling, did not encapsulate the non-linearity between the frames, and could actually hurt the performance. Our presented concatenation method better models the sequence, however, results in a very large place descriptor. The MSLS dataset is designed for sequential dataset, however more advanced modelling of the inter-sequential relations is needed to properly exploit this temporal information between the frames. This could be achieved by using a fully connected layer to fuse the individual frame-descriptors together. However, in order to avoid a very large number of trainable parameters, the dimensionality of the frame descriptors must be reduced substantially as well as the dimensionality of the final sequence descriptor. We also believe that a more explicit modelling of the temporal information might yet yield better results. This could be achieved by inserting a Long Short-Term Memory (LSTM) network or a Gated Recurrent Unit (GRU) network to replace the pooling/fusion models. However, these methods also require a reduction of the dimensionality of the place descriptors.

We introduced two new tasks: seq-2-im and im-2-seq and compared several new techniques to solve these tasks using models trained for im-2-im place recognition. We found that in the seq-2-im case, using the minimum distance between the query frame and database frame across the sequence to be highly effective and outperforming all im-2-im and seq-2-seq models. The reason that our proposed minimazation technique performs well is that it exploits the sequential information in the input sequences, however does not rely on a too simple modelling of the relation between the frames in the sequence.

We were the first to show that current state-of-the-art place recognition models are heavily biased towards developed cities in Europe and North America. This bias stems from their training data, which does not include imagery from undeveloped cities and countries. We showed that MSLS can be useful for evaluating model bias, because of its large geographical coverage. Furthermore, we showed that training on the MSLS dataset can reduce models' geographical bias, however, we expect that extending of our proposed positive mining strategy to include a more frequent sampling of imagery from Asian, South American and African cities, could further reduce the geographical bias for place recognition models.

We additionally conducted a qualitatively examination of the models trained on our dataset. As we expected, the model's feature maps had high activation at stationary objects such as buildings and lower activation around dynamic objects such as car, bikes and vegetation. This further supports that our trained model has pick up intuitively important parts of a scene for place recognition.

Analysing the learned embedding space, we found that geographical similar places were located close in the embedding space. Furthermore, places that shared visual content such as tunnels were also close in the embedding space. Even more general attributes, such as in which city an image was captured, were easily visible from a low dimensional visualization of the embedding space. These qualitative results lead to a better intuition and explainability on the inner mechanisms of the model. We believe that a hierarchical approach, where a classifier is trained to predict city or continent followed by a place descriptor per individual city, might yield better results, reduction in the number of required model parameters and enable a lower dimensional place descriptor.

Although, the focus of this thesis is place recognition, Mapillary SLS is also useful for other computer vision tasks such as pose regression Kendall et al. (2015); Walch et al. (2017), image synthesis (e.g., night-to-day translation Anoosheh et al. (2019)), image-to-gps Vo et al. (2017); Zemene et al. (2018)), change detection, feature learning, scene classification using the OSM road tags and unsupervised depth learning Li and Snavely (2018); Gordon et al. (2019).

Bibliography

https://github.com/mapillary/OpenSfM. 15

M. Angelina Uy and G. Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018. 7

A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964. IEEE, 2019. 47

R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 1, 6, 7, 9, 19, 21, 24, 25

A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. *ICCV*, 10 2015a. 6, 20

A. Babenko and V. S. Lempitsky. Aggregating deep convolutional features for image retrieval. *CoRR*, abs/1510.07493, 2015b. URL http://arxiv.org/abs/1510.07493. 6

H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 794–799, June 2011. doi: 10.1109/IVS.2011.5940504. 8

A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678, 2016. URL http://arxiv.org/abs/1605.07678. 32

N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of Michigan North Campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9):1023–1035, 2015. 8

W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. *CoRR*, abs/1704.01719, 2017a. URL http://arxiv.org/abs/1704.01719. 7

Z. Chen, O. Lam, A. Jacobson, and M. Milford. Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509, 2014. URL http://arxiv.org/abs/1411.1509. 6

Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford. Deep learning features at scale for visual place recognition. *CoRR*, abs/1701.05105, 2017b. URL http://arxiv.org/abs/1701.05105. 8

Z. Chen, F. Maffra, I. Sa, and M. Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9–16, Sep. 2017. doi: 10.1109/IROS.2017.8202131. 7

M. Cummins. Highly scalable appearance-only SLAM - FAB-MAP 2.0. *Proc. Robotics: Sciences and Systems (RSS), 2009*, 2009. 8

M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011. 6

J. M. Fácil, D. Olid, L. Montesano, and J. Civera. Condition-invariant multi-view place recognition. *CoRR*, abs/1902.09516, 2019. URL http://arxiv.org/abs/1902.09516. 2, 7, 21, 22

D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. 6

S. Garg, V. M. Babu, T. Dharmasiri, S. Hausler, N. Sünderhauf, S. Kumar, T. Drummond, and M. Milford. Look no deeper: Recognizing places from opposing viewpoints under varying scene appearance using single-view depth estimation. *CoRR*, abs/1902.07381, 2019a. URL http://arxiv.org/abs/1902.07381. 7

S. Garg, N. Sünderhauf, and M. Milford. Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, page 027836491983976, 04 2019b. doi: 10.1177/0278364919839761. 7

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 8

A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth. FAB-MAP+ RatSLAM: Appearance-based SLAM for multiple times of day. In *2010 IEEE international conference on robotics and automation*, pages 3507–3512. IEEE, 2010. 8

R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. G. Jiménez. Training a convolutional neural network for appearance-invariant place recognition. *CoRR*, abs/1505.07428, 2015. URL http://arxiv.org/abs/1505.07428. 6

A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *CoRR*, abs/1610.07940, 2016. URL http://arxiv.org/abs/1610.07940. 7

A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 47

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385. 6, 19, 21

G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993. 6

H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 304–317, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_24. URL http://dx.doi.org/10.1007/978-3-540-88682-2_24. 9

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311. IEEE Computer Society, 2010. 6

Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. *CoRR*, abs/1512.04065, 2015. URL http://arxiv.org/abs/1512.04065. 6

A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 47

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. 6, 19, 21

M. Leyva-Vallina, N. Strisciuglio, M. L. Antequera, R. Tylecek, M. Blaich, and N. Petkov. Tb-places: A data set for visual place recognition in garden environments. *IEEE Access*, 7:52277–52287, 2019. 8

M. Li, L. Wu, A. Wiliem, K. Zhao, T. Zhang, and B. C. Lovell. Deep instance-level hard negative mining model for histopathology images. *CoRR*, abs/1906.09681, 2019. URL http://arxiv.org/abs/1906.09681. 24

Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 47

M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92:89–95, 2017. 6

S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32 (1):1–19, 2016. 1, 5, 6

W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36 (1):3–15, 2017. doi: 10.1177/0278364916679498. URL http://dx.doi.org/10.1177/0278364916679498. 8

M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation*, pages 1643–1649. IEEE, 2012. 6

T. Naseer, W. Burgard, and C. Stachniss. Robust visual localization across seasons. *IEEE Transactions on Robotics*, 34(2):289–302, April 2018. ISSN 1552-3098. doi: 10.1109/TRO.2017.2788045. 8

H. Noh, A. Araujo, J. Sim, and B. Han. Image retrieval with deep local features and attention-based keypoints. *CoRR*, abs/1612.06321, 2016. URL http://arxiv.org/abs/1612.06321. 9

NRK. Nordlandsbanen: minute by minute, season by season, 2013. URL https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/. 8

D. Olid, J. M. Fácil, and J. Civera. Single-view place recognition under seasonal changes. *CoRR*, abs/1808.06516, 2018. URL http://arxiv.org/abs/1808.06516. 8

K. Ozaki and S. Yokoo. Large-scale landmark retrieval/recognition under a noisy and diverse dataset. *CoRR*, abs/1906.04087, 2019. URL http://arxiv.org/abs/1906.04087. 9

F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, June 2010. doi: 10.1109/CVPR.2010.5540009. 6

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383172. 9

A. Queensland University of Technology, Brisbane. Day and night with lateral pose change datasets, 2014. URL https://wiki.qut.edu.au/display/cyphy/Day+and+Night+with+Lateral+Pose+Change+Datasets. 8, 9

F. Radenovic, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *CoRR*, abs/1711.02512, 2017. URL http://arxiv.org/abs/1711.02512. 6, 7, 19, 20, 24

F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. *CoRR*, abs/1803.11285, 2018. URL http://arxiv.org/abs/1803.11285. 9

A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Visual instance retrieval with deep convolutional networks. *CoRR*, abs/1412.6574, 2014. 6, 20

J. Revaud, J. Almazán, R. S. de Rezende, and C. R. de Souza. Learning with average precision: Training image retrieval with a listwise loss. *CoRR*, abs/1906.07589, 2019. URL http://arxiv.org/abs/1906.07589. 7

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 8, 9

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015. doi: 10.1007/s11263-015-0816-y. 6

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6, 19, 21

J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003. 6

K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6200-improved-deep-metric-learning-with-multi-class-n-pair-loss-objective.pdf. 7

N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. *CoRR*, abs/1501.04158, 2015. URL http://arxiv.org/abs/1501.04158. 6

N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4297–4304. IEEE, 2015a. 6

N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015b. 6

G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. 11 2015. 6

G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. 2016. 7

A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015. 6, 8, 9

A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (11):2346–2359, Nov 2015. doi: 10.1109/TPAMI.2015.2409868. 8, 9

D. P. Vassileios Balntas, Edgar Riba and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.11. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.119. URL https://dx.doi.org/10.5244/C.30.119. 24

N. Vo, N. Jacobs, and J. Hays. Revisiting im2gps in the deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2630, 2017. 47

F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017. 47

A. B. Yandex and V. Lempitsky. Aggregating local deep features for image retrieval. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, Dec 2015. doi: 10.1109/ICCV.2015.150. 6

M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, and K. D. McDonald-Maier. Levelling the playing field: A comprehensive comparison of visual place recognition approaches under changing conditions. *CoRR*, abs/1903.09107, 2019. URL http://arxiv.org/abs/1903.09107. 1, 5, 6, 7

A. R. Zamir and M. Shah. Image geo-localization based on multiplenearest neighbor feature matching using generalized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014. 8, 9

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901. 39

E. Zemene, Y. T. Tesfaye, H. Idrees, A. Prati, M. Pelillo, and M. Shah. Large-scale image geo-localization using dominant sets. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):148–161, 2018. 47

APPENDIX A

Appendix

A.1 CVPR paper submission

A paper for the conference of Computer Vision and Pattern Recognition (CVPR) was curated and submitted as part of this thesis. We have included the submitted paper on the following pages.

Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition

Anonymous CVPR submission

Paper ID 9217

Abstract

Lifelong place recognition is an essential and challenging task in computer vision with vast applications in robust localization and efficient large-scale 3D reconstruction. Progress is currently hindered by a lack of large, diverse, publicly available datasets. We contribute with Mapillary Street-Level Sequences (SLS), a large dataset for urban and suburban place recognition from image sequences. It contains more than 1.6 million images curated from the Mapillary[1] collaborative mapping platform. The dataset is orders of magnitude larger than current data sources, and is designed to reflect the diversities of true lifelong learning. It features images from 30 major cities across six continents, hundreds of distinct cameras, and substantially different viewpoints and capture times, spanning all seasons over a nine year period. All images are geo-located with GPS and compass, and feature high-level attributes such as road type.

We propose a set of benchmark tasks designed to push state-of-the-art performance and provide baseline studies. We show that current state-of-the-art methods still have a long way to go, and that the lack of diversity in existing datasets have prevented generalization to new environments. The dataset and benchmarks will be made available for academic research.

1. Introduction

Visual place recognition is essential for the long-term operation of Augmented Reality and robotic systems [28]. However, despite its relevance and vast research efforts, it remains challenging in practical settings due to the wide array of appearance variations in outdoor scenes, as seen on the examples extracted from our dataset in Figure 1.

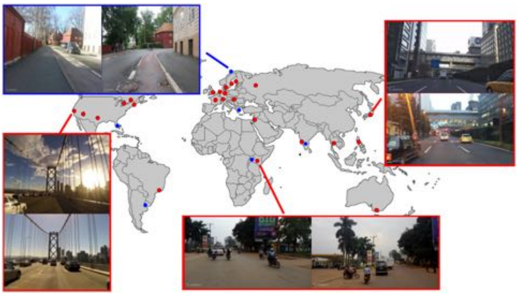


Figure 1: Mapillary SLS contains imagery from 30 major cities around the world; red stands for training cities and blue for test cities. See four samples from San Francisco, Trondheim, Kampala and Tokyo with challenging appearance changes due to viewpoint, structural, seasonal, dynamic, and illumination.

Recent research on place recognition has shown that features learned by deep neural networks outperform traditional hand-crafted features, particularly for drastic appearance changes [4, 28, 54]. This has motivated the release of several datasets for training, evaluating and comparing deep learning models. However, such datasets are limited, in at least three aspects. First, none of them covers the many appearance variations encountered in real-world applications. Second, many of them have insufficient size for training large networks. Finally, most datasets are collected in small areas, lacking the geographical diversity needed for generalization.

This paper contributes to the progress of lifelong place recognition by creating a dataset addressing all the challenges described above. We present **Mapillary Street-Level Sequences (SLS)**, the largest dataset for place recognition to date and the one that contains the widest variety of perceptual changes and the broadest geographical spread[2]. Mapillary SLS covers the following causes of appearance change: different seasons, changing weather conditions, varying illumination at different times of the day, dynamic objects such as moving pedestrians or cars, structural modifications such as roadworks or architectural work, camera intrinsics and viewpoints. Our data spans six continents, including diverse cities like Kampala, Zurich, Amman and Bangkok.

In addition to the dataset, we make several contributions related to its experimental validation. We formulate a novel subset caching methodology for hard triplet mining that does

[1] https://www.mapillary.com

[2] See the video accompanying the paper for an overview and sample images.

						Type of appearance changes						
Name	Environment	Total length	Geographical coverage	Temporal coverage	Frames	Seasonal	Weather	Viewpoint	Dynamic	Day/night	Intrinsics	Structural
Nordland [33, 34]	Natural + urban	728 km	182 km	1 year	∼115K	✓	✗	✗	✗	✗	✗	✗
SPED [8]	Urban	-	-	1 year	**∼2.5M**	✓	✓	✗	✓	✓	✗	✗
KITTI [16]	Urban + suburban	39.2 km	1.7 km	3 days	∼13K	✗	✗	✓	✓	✗	✗	✗
Eynsham [10]	Urban + suburban	70 km	35 km	1 day	∼10K	✗	✗	✗	✓	✗	✗	✗
St. Lucia [17]	Suburban	47.5 km	9.5 km	1 day	∼33K	✗	✗	✗	✓	✗	✗	✗
NCLT [6]	Campus	148.5 km	5.5 km	15 mon.	∼300K	✓	✗	✓	✓	✗	✗	✗
Oxford RobotCar [29]	Urban + suburban	1.000 km	10 km	1 year	∼27K	✓	✓	✓	✓	✓	✗	✓
VL-CMU [5]	Urban + suburban	128 km	8 km	1 year	∼1.4K	✗	✗	✓	✓	✗	✗	✗
FAS [31]	Urban + suburban	120 km	70 km	3 years	∼43K	✓	✓	✓	✓	✗	✗	✓
Garden Point [38]	Urban + campus	<12 km	4 km	1 week	∼600	✗	✗	✓	✗	✓	✗	✗
SYNTHIA [41]	Urban	6 km	1.5 km	-	∼200K	✓	✓	✓	✓	✓	✗	✗
GSV [55]	Urban	-	-	-	∼60K	✗	✗	✗	✗	✗	✗	✗
Pittsburgh 250k [49]	Urban	-	-	-	∼254K	✗	✗	✓	✓	✗	✗	✗
TokyoTM/247 [48]	Urban	-	-	-	∼174K	✓	✗	✓	✓	✓	✗	✓
TB-places [25]	Gardens	<100m	<100m	1 year	∼60K	✗	✗	✓	✓	✗	✗	✗
Mapillary SLS (Ours)	Urban + suburban	**11,560 km**	**4,228 km**	**7 years**	∼1.68M	✓	✓	✓	✓	✓	✓	✓

Table 1: **Summary of place recognition datasets**. Geographical coverage is the length of unique traversed routes. Total length is the geographical coverage multiplied by the number of times each route was traversed. Temporal coverage is the time span from the first recording of a route to the last recording. "–" stands for "not applicable".

not scale in time or memory with the dataset size; enabling training of place recognition models on very large datasets (as ours). We tackle a wider set of problems not limited to image-to-image localization by proposing six variations of MultiViewNet [12] to model sequence-to-sequence place recognition. Moreover, we formulate two new research tasks: sequence-to-image and image-to-sequence recognition, and propose several feature descriptors that extend pretrained image-to-image models to these two new tasks.

2. Related Work

Place Recognition. Place recognition consists of finding the most similar place of a query image within a database of registered images [28, 54]. Traditional visual place descriptors are based on aggregating local features using bag-of-words [42], Fischer vectors [36] or VLAD [21]. Other hand-crafted approaches exploit geometric and/or temporal consistency [11, 13, 30] in image sequences. DenseVLAD [48] synthesizes viewpoint changes from panorama images with associated depth. These synthetic images make the DenseVLAD more robust to viewpoint and day/night changes.

As in other computer vision tasks, deep features have demonstrated better performance than hand-crafted ones [54]. Initially, features from existing pre-trained networks were used for single-view place recognition [7, 43–45, 53]. Later works demonstrated that the performance improves if the networks are trained for the specific task of place recogntion [4, 18, 27]. One of the recent successes is NetVLAD [4, 54], which use a base network (e.g. VGG16) followed by a generalized VLAD layer (NetVLAD) as an image descriptor. Other works, such as R-MAC [47] and Chen et al. [9], extract regions surrounding high feature map activations to form place descriptors.

Recent deep learning-based methods exploit the temporal, spatial, and semantic information in images or image sequences. Radenovic et al. [39] propose using the geometry of a 3D reconstructed scene to mine hard positives and negatives for training a Generalized Mean (GeM) layer. Garg et al. [14], on the other hand, uses single-view depth predictions to recognize places revisited from opposite directions. Also, addressing extreme viewpoint changes, Garg et al. [15] suggests semantically aggregating salient visual information. The 3D geometry of a place is also used by PointNetVLAD [2] that combines PointNet and NetVLAD to form a global place descriptor from LiDAR data. MultiViewNet [12] investigates different pooling strategies, descriptor fusion and LSTMs to model temporal information in image sequences. This research is, however, hindered by the lack of appropriate datasets.

Another line of research addresses compute- and storage-efficient place recognition for mobile robotics. Region-VLAD [23] extracts regions of interest from a lightweight CNN that are propagated through a VLAD layer, resulting in a fast and memory efficient method. Other examples in this area are [24, 46].

Place Recognition Datasets. Table 1 summarizes a set of relevant place recognition datasets. Below we highlight more details and compare our contributions against existing datasets.

Nordland [33, 34] contains 4 sequences of a 182km-long train journey, traversed once per season. It captures seasonal changes but contains small variations in viewpoint, camera intrinsics, time of day or structural changes.

SPED [8] was curated from images taken by 2.5K static surveillance cameras over 1 year. It contains dynamic, illumination, weather and seasonal changes. However, it does not include viewpoint changes nor ego-motion.

KITTI [16], **Eynsham** [10] and **St. Lucia** [17] were all recorded by car-mounted cameras. In all three cases the cars drove in urban environments within a few-days, capturing dynamic elements and slight viewpoint and weather changes, but no long-term variations. There are several other datasets oriented to autonomous driving collected over longer periods: **NCLT** [6] (recorded over a period of 15 months in a campus environment), **Oxford RobotCar** [29] (recorded from a car traversing the same 10 km route twice every week for a year), **VL-CMU** [5] (composed by 16×8 km street-view videos captured over one year) and **Freiburg Across Seasons (FAS)** [31] (composed of 2×60 km summer videos and 1×10 km winter video over a period of three years). None of them has geographical diversity nor variations in the camera intrinsics, and their viewpoint, structural and weather changes are minor.

Gardens Point [38] was recorded with a hand-held iPhone. It contains day/night and significant viewpoint changes, but a small representation of other appearance changes and has a small size. **SYNTHIA** [41] contains 4 synthetic image sequences along the same route. It includes varying viewpoints, seasonal, weather, dynamic and day/night changes.

GSV [55] compiled a street-level image dataset from Google Street View. However, it is relatively small at 60,000 images. It is limited to a few US cities with no temporal changes and it is composed of still images instead of sequences. **Pittsburgh250k** [49] was also extracted from Google Street View panoramas in Pittsburgh ($10,586$ of them specifically, using two yaw directions and 12 pitch directions). The limited geographical span of these datasets results in a low number of unique places compared to ours.

Tokyo [48] comes in two versions: The Tokyo Time Machine dataset ($\sim$ 98K images) and Tokyo 24/7 ($\sim$ 75K images). Tokyo 24/7 has significant day/night changes. However, [4] comments that the trained model with the Tokyo datasets shows signs of overfitting, probably caused by their limited geographical coverage and size.

Notice that **GSV**, **Pittsburgh250k** and **Tokyo** have significant viewpoint variation but do not include information of viewing direction for the images, and hence positive mining of images with overlaps in view point is not straightforward. In our **Mapillary SLS**, we include viewing direction information for each image. (see details in section 3).

Image retrieval is a similar task to place recognition, aiming to find an image in a database that is the most similar one to a query image. There exist several image retrieval datasets (typically created from Flickr images) and established benchmarks, *e.g.*, Holidays [20], Oxford5k, Paris6k [37], Revisited Oxford5k and Paris6k [40], and Google Landmarks [32, 35]. They usually focus on single-image retrieval and have a very large set of images from the same place, which limits their application in benchmarking lifelong place recognition.

3. The Mapillary SLS Dataset

To push the state-of-the-art in lifelong place recognition, there is a need for a larger and more diverse dataset. We have, with this in mind, created a new dataset comprised of 1.6 million images from Mapillary[3]. In this section, we present an overview of the curation process, characteristics, and statistics of the dataset. With the available sequential information of the dataset, we additionally propose two new research benchmark tasks.

3.1. Data Curation

Our goal is to create a dataset for place recognition with images that (1) has wide geographical reach, reducing bias towards highly populated cities in developed countries (2) is visually diverse, capturing scenarios under varying weather, lighting, and time (3) tagged with reliable geometric and sequential information, enabling new research and practical applications.

3.1.1 Image Selection

Geographical Diversity. To ensure geographical diversity, we start with a set of candidate cities for image selection. For each candidate city, we create a regular grid of 500m^2 cell size and process each of the cells independently. For each cell we extract a series of image sequences recorded within this cell. Each sequence contains the image keys and their associated GPS coordinates and raw compass angles (indicating viewing direction).

Mapillary SLS contains data from 30 cities spread over 6 continents. See Figure 1 and Table 2 for details. It covers diverse urban and suburban environments as indicated by the distribution of corresponding OpenStreetMap (OSM)[4] road attributes (Figure 3).

Unique User and Capture Time. To ensure variation in the scene structure, time of day, camera intrinsics and view points within each geographical cell, we only keep one sequence per photographer and pick sequences from different days.

Consistent Viewing Direction. To ensure that viewing direction measurement is reliable for selecting matching images, we enforce consistency between raw compass angles (measured by the capturing device) and the estimated viewing direction computed with Structure from Motion (SfM)[5]. We select only sequences in which at least 80% of the images' computed angles agree ($\leq 30°$ difference) with the raw compass angle.

3.1.2 Sequence Clustering

Given this initial set of sequences from the image selection process, we generate clusters of sequences that are candidates

[3] www.mapillary.com

[4] www.openstreetmap.org

[5] The estimated viewing directions were computed based on the relative camera poses estimated using the default OpenSfM [1] pipeline with the camera positions aligned with the GPS measurements

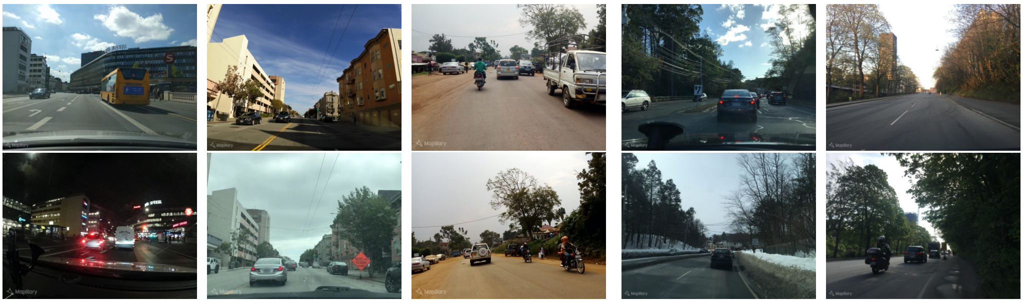


Figure 2: Mapillary SLS pairs showing day/night, weather, seasonal, structural, viewpoint and domain changes.

Continent	# Frames	# Night Frames	Geo. Coverage [km]	Total Coverage [km]	#Clusters
Europe	516 K	1,098	1,052	2,985	8,654
Asia	468 K	9,820	965	2,729	5,483
North America	431 K	3,968	171	4,616	6,504
South America	61 K	1,177	214	599	1,065
Australia	200 K	0	259	568	1,493
Africa	5 K	0	28	63	108
Total	**1,681 K**	**16,063**	**4,228**	**11,560**	**23,307**

Table 2: Continental coverage in Mapillary SLS.

for place recognition. To avoid sequences where the distance between consecutive images is large, we first split each raw sequence into subsequences if there are more than 30 m between two consecutive frames. Then, we pairwise-match these sub-sequences based on their distance, viewing direction, and motion direction [6]. This is done by searching among all the sub-sequences and forming candidate clusters (sub-sequence pairs) based on their distances to all other neighboring sub-sequences.

To form a candidate cluster we use the following criteria: Frames from sub-sequences A and B are clustered together if: **1)** Their distance is less than 30 m. **2)** The difference between their viewing directions is less than $40°$. **3)** The difference between their moving directions is less than $40°$. In practice, we use a k-d tree to efficiently discover these pairwise correspondences. The above criteria sometimes skips intermediate images in a sequence, *e.g.*, a subsequence might have the images $\{1,2,4,5\}$, thus missing image 3. To avoid this effect, we added all such skipped images back into the sequence.

After matching sub-sequence pairs into potential clusters, we prune them to obtain the frames where both subsequences overlap and hence can be used for sequence-to-sequence place recognition. Since there might be more sequences matching, we merge all pairwise clusters (e.g., we merge clusters A, B and C if there are images that belong to clusters AB, AC and BC.)

We end up with clusters of sequences that have the same geographical coverage and the same moving- and viewing-direction. The sequences in the clusters are relatively short (5-300 frames) providing a very diverse set of sequential examples for training and development of multi-view place descriptors.

Finally, we filter the resulting clusters enforcing: **1)** that each subsequence has 5 or more frames for proper evaluation of multi-view place recognition models; and **2)** that each cluster has at least two subsequences, in order to have a sufficient number of positive training and test samples.

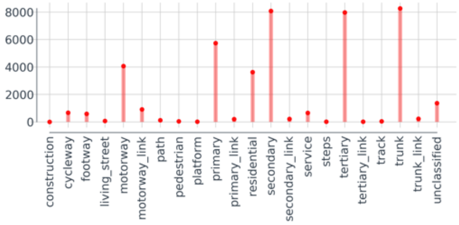


Figure 3: Distribution of OSM road attributes for Mapillary SLS.

3.2. Image Attributes

For each image, we additionally provide several image attributes that are relevant for further research.

Day and Night. We provide an attribute indicating if a sequence is captured during day or night time. We verified that the day/night attribute could not be robustly estimated from the capture time of the images. Therefore, we implemented a day/night classifier based on the hue distribution of the entire

[6] The motion direction for each image is calculated using the GPS measurement and the capture times of consecutive images in a sequence.

image and of the sky region identified using semantic segmentation. Given the prediction of each image, we then performed a majority voting across the entire sequence to provide consistent day/night tags. To obtain the sky region, we use a semantic segmentation mask provided by Mapillary's API. By manual inspection, we found that such a classifier is sufficient.
Facing Direction. We additionally include the facing direction of the camera: forward, backward or sideways, which is calculated using the estimated moving direction and viewing direction for each image.
Road Attributes. Based on the GPS locations of the images, we have also tagged each sequence with road attributes (*e.g.,* residential, motorway, path or others), which are obtained from OpenStreetMap[7] (OSM).

3.3. Data Overview

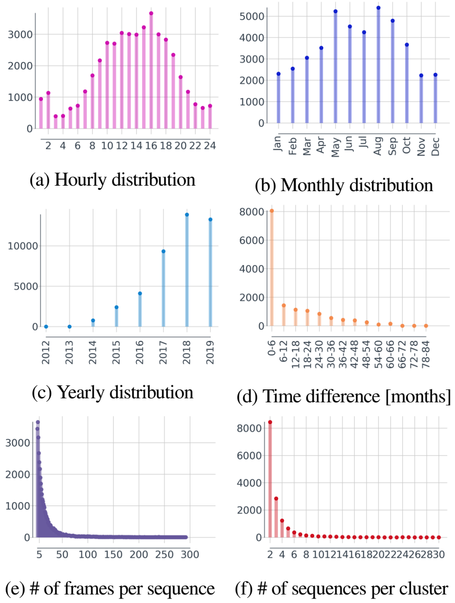
In this section, we provide an overview of the Mapillary SLS dataset in terms of its diversity. In Figures 4a and 4b, we show that the dataset covers all times of the day and months of the year. Figures 4c and 4d show that it spans nine years and that the same places have been revisited with up to seven years time difference, making Mapillary SLS the dataset with largest time span for lifelong place recognition. Figures 4e and 4f show large variety in sequence length and number of recordings for the same places.

To highlight the board variety and challenge, we show in Figure 2 image samples from our dataset, where each column contains an query and a database images at a nearby location. In the first column, the query images are taken during the day whereas the database image is taken at night. The second column shows an example of drastic weather changes as well as a new roadwork traffic sign. The third column shows images from Kampala; a drastic change in environment compared to the images in the first two columns from Copenhagen and San Francisco. Seasonal and structural changes are visible in the two last column, as the sky scraper on the left side of the road is under construction in the bottom image and stands finished in the top one. More visual examples of vast variety of changes between query and database images are available in the supplementary material.

3.4. Data Partition and Evaluation

We divide the dataset into a training set (roughly 90%) and a test set (the remaining 10%) containing disjoint sets of cities. Specifically, the test set consists of images collected from Miami, Athens, Buenos Aires, Stockholm, Bengaluru and Kampala. We phrase four place recognition tasks combining single images and sequences in the query and database, that will be referred from here on as **im2im**, **seq2seq**, **im2seq**, and **seq2im** (**x2y** stands for query **x** and database **y**), respectively.

In addition to evaluating on the whole test set, we suggest the following three research challenges and provide a separate scoreboard for each: **Day/Night** (how well the model recognizes places from day and night and vice versa), **Seasonal** (how well the model recognizes places between seasons, Summer/Winter and vice versa being the most challenging) and **New/Old** (how well does the model recognizes places after several years).



(a) Hourly distribution (b) Monthly distribution

(c) Yearly distribution (d) Time difference [months]

(e) # of frames per sequence (f) # of sequences per cluster

Figure 4: Distribution of image sequences in Mapillary SLS on a daily, monthly, yearly scale, time variation, and sequence-related characteristics.

Similar to previous works, we cast place recognition as an image retrieval problem and use the top-5 recall as evaluation metric. For each cluster, we choose one sequence to be the query and the remaining ones to be the database. In the following we will use the **query example** to describe either a query image or a query sequence. The query example is chosen as the center frame(s) in the chosen query sequence. Only one query example is chosen per query sequence, ensuring an equal weight of every place in the evaluation independently of its number of frames. We define the ground truth matches as those images within a radius of 25m of the query image with a viewing angle difference to it smaller than 40°

4. Experiments

In this section, we present first our training procedure for the baseline methods. We show experimental results on the Mapillary SLS dataset in both single-view and multi-view settings.

4.1. Training

For the baseline method, we have used NetVLAD [4] and followed a similar training procedure and hyper-parameter se-

[7] https://wiki.openstreetmap.org

lection scheme. The model is trained with the triplet loss [50], for which presenting hard triplets is critical to learn a good embedding. The usual procedure to extract hard triplets is to cache the entire dataset at regular intervals and look for examples that violate the triplet constraint. In practice, this means that the time spent on updating the cache scales linearly with the size of the dataset, making it infeasible to train place recognition models on very large datasets, such as Mapillary SLS. We propose a simple, yet effective sub-caching method with constant time and space use. We propose to divide the entire dataset into equally sized and randomly sampled subsets. For each query image in a subset, we add all its positive images to the given subset. We then update the cache for this subset, making the cache-time scale with respect to the subset size rather than the size of the entire dataset. Both the query and positive images can be sampled from the cache as well as negatives. It is important to keep the subset size large enough to find adequately hard triplets. In our experiments, we use $10{,}000$ query images and refresh the cache every $1{,}000$ iterations. We use 5 negative examples per triplet instead of 10 [4] as this allow us to fit a batch size of 4 into memory.

4.2. Single-View Place Recognition

In Table 3, we benchmark the most common deep models for **im2im** recognition, reporting their top-5 recall on several challenging recognition cases as well as their top-1/5/10 recall on the entire test set. These challenging cases include summer to winter (Su/Wi), day to night (Da/Ni), old to new (Ol/Ne) and vice versa. We define old images as those taken between 2011–2016 and new images as those taken since 2018. The goal is to separately evaluate the performance of each method when exposed to seasonal, day/night and structural changes.

We evaluate two early models: Amosnet and Hybridnet [8] and two more recent ones: NetVLAD [4] and GeM (Generalized Mean) [39]. Amosnet and Hybridnet have a Caffe-net backbone followed by two fully connected layers. NetVLAD [4] consists of a VGG16 core with a trainable VLAD layer, and is the state-of-the-art on several place recognition datasets. We evaluated the variant of GeM with VGG16 backbone architecture that is trained on 3d reconstructions of 120k images from Flikr (SfM-120k).

Figure 3 shows that training on the diverse SLS improves the overall performance. The performance boost is mainly caused by improved capabilities to recognize places that have undergone seasonal and temporal changes. All models are especially challenged by the night to day changes. Observe also that top-5 recall is in general lower on the SLS test set, which is a measurement of its higher degree of difficulty. Figure 5 shows a detailed comparison of models trained on other datasets to the NetVLAD on our dataset with varying distance and number of image candidates.

4.3. Multi-View Place Recognition

We propose to reformulate MutiViewNet [12] to address **seq2seq**, **seq2im**, and **seq2im** place recognition. To the best of our knowledge, no previous work has addressed these two last cases. We propose two novel architectures based on NetVLAD and show the results in Table 3.

seq2seq. We propose six variations of a MutiViewNet [12], specifically, three pooling techniques for NetVLAD and three for GeM. The motivation is to adapt embeddings that are known to work well for single-view place recognition. The first technique, NetVLAD/GeM-MAX, performs max pooling across the embeddings of each image in the sequence. The second variation, NetVLAD/GeM-AVG, does average pooling. The last one, NetVLAD/GeM-CAT, concatenates the embeddings. Results are reported in Table 3.

seq2im. In the sequence-to-image case, we propose to make a majority voting across the sequence, ie select the image in the database that most images are nearest to in the query sequence. Given a query sequence of N frames, we calculate the distance from each frame to each database image. We then look at the closest k distances for each of our N frames in our query sequence. This gives a total of $k \times N$ closest database images. We then select the most frequently occurring. The intuition is that if all the frames in a sequence are close to a database image, then we are more confident that this database image is indeed close to the query sequence. We also test the selection of the closest image in the database among all the images in the query sequence. Again, we test these methods using both the VGG16 + GeM and VGG + NetVLAD embeddings (See Table 3).

im2seq. In the image-to-sequence case, we test the selection of the sequence containing the image with the nearest query. In practice, we calculate the distances from the query image to all the frames in the database sequences, and select the sequence that contains the nearest frame.

Table 3 shows that these simple pooling strategies do not improve model performance much compared to single-view models. The reason is that SLS sequences are captured at different frame-rates and user velocities, requiring the model to learn a time-independent relation between the frames. This complex relations cannot be captured by simple pooling strategies. This motivates the development and further research in multi-view methods, which is accommodated by release of the Mapillary SLS.

4.4. Further Analysis

In this section, to understand better the strength of the diversity of the dataset, we present qualitative and geographical analysis on the recognition results.

Qualitative Model Comparison: In Figure 7, we qualitatively evaluate AmosNet, HybridNet, VGG16-GeM and NetVLAD trained on Pittsburgh250k and SfM120K, and NetVLAD trained on Mapillary SLS. Notice how the diversity of the Mapillary SLS data makes NetVLAD more robust to viewpoint and weather changes, comparing to models that

	Model	Training set	Base	Input Size	Dim	Su/Wi	Wi/Su	Da/Ni	Ni/Da	Ol/Ne	Ne/Ol	All (@1/5/10)
im2im	Amos	SPED	CaffeNet	227x227	2543	0.17	0.09	0.20	0.09	0.17	0.14	0.06/0.11/0.14
	Hybrid	SPED	CaffeNet	227x227	2543	0.13	0.11	0.14	0.11	0.18	0.17	0.08/0.13/0.15
	NetVLAD	Pitts250k	VGG16	480x640	512	0.43	0.44	0.37	0.09	0.49	0.50	0.28/0.35/0.39
	GeM	SfM-120k	VGG16	480x640	2048	0.51	0.48	0.37	0.20	0.55	0.56	0.30/0.40/0.44
	NetVLAD	SLS	VGG16	480x640	512	**0.76**	**0.74**	**0.49**	**0.23**	**0.71**	**0.75**	**0.48/0.58/0.64**
seq2seq	NetVLAD + MAX	Pitts250k	VGG16	480x640	512	0.40	0.51	0.37	0.09	0.55	0.57	0.23/0.32/0.36
	NetVLAD + AVG	Pitts250k	VGG16	480x640	512	0.41	0.39	0.37	0.09	0.54	0.54	0.20/0.31/0.34
	NetVLAD + CAT	Pitts250k	VGG16	480x640	512	0.44	0.47	0.37	0.14	0.57	0.56	0.23/0.33/0.37
	GeM + MAX	SfM-120k	VGG16	480x640	2048	0.53	0.54	0.43	**0.26**	0.67	0.57	0.29/0.43/0.48
	GeM + AVG	SfM-120k	VGG16	480x640	2048	0.60	0.52	0.40	0.14	0.66	0.57	0.29/0.42/0.46
	GeM + CAT	SfM-120k	VGG16	480x640	2048	0.55	0.46	0.46	**0.26**	0.65	0.53	0.28/0.42/0.46
	NetVLAD + MAX	SLS	VGG16	480x640	512	0.75	**0.79**	0.51	0.14	**0.80**	**0.76**	0.42/0.58/0.63
	NetVLAD + AVG	SLS	VGG16	480x640	512	0.75	0.78	0.51	0.06	0.78	0.73	0.37/0.56/0.60
	NetVLAD + CAT	SLS	VGG16	480x640	512	**0.84**	0.76	**0.57**	0.20	**0.80**	0.72	**0.41/0.60/0.65**
seq2im	NetVLAD + MIN	Pitts250k	VGG16	480x640	512	0.53	0.53	0.37	0.03	0.60	0.62	0.30/0.37/0.40
	NetVLAD + MODE	Pitts250k	VGG16	480x640	512	0.53	0.51	0.46	0.06	0.61	0.59	0.28/0.37/0.41
	GeM + MIN	SfM-120k	VGG16	480x640	2048	0.62	0.62	0.37	0.23	0.71	0.67	0.38/0.47/0.50
	GeM + MODE	SfM-120k	VGG16	480x640	2048	0.59	0.52	0.46	**0.26**	0.67	0.66	0.32/0.45/0.51
	NetVLAD + MIN	SLS	VGG16	480x640	512	**0.86**	**0.86**	**0.54**	0.20	**0.83**	**0.81**	**0.56/0.68/0.71**
	NetVLAD + MODE	SLS	VGG16	480x640	512	0.53	0.51	0.46	0.06	0.61	0.59	0.28/0.37/0.41
im2seq	NetVLAD + MIN	Pitts250k	VGG16	480x640	512	0.20	0.30	0.29	0.14	0.33	0.28	0.12/0.20/0.26
	GeM + MIN	SfM-120k	VGG16	480x640	2048	0.24	0.22	0.26	**0.31**	0.37	0.29	0.13/0.22/0.31
	NetVLAD + MIN	SLS	VGG16	480x640	512	**0.45**	**0.39**	**0.31**	0.23	**0.48**	**0.37**	**0.23/0.34/0.48**

Table 3: Evaluation of different im2im, seq2seq, seq2im and im2seq models on Mapillary SLS test set. We report the models recall@5 on several challenging recognition cases as well as their overall recall@1/5/10. For a fair comparison, we compare models with similar backbone architecture.

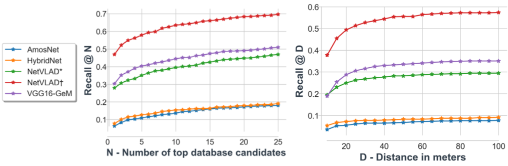


Figure 5: Recall of different methods on Mapillary SLS as a function of number of nearest neighbors (left) and distance threshold (right).

trained on other datasets that do not encapsulate as much diversity as our dataset.

Geographical Bias: State-of-the-art place recognition networks are trained on images from developed countries. Figure 8 shows the performance of several models on individual cities from Mapillary SLS, confirming their geographical bias. Notice that the GeM model has slightly less bias, the performance drop on Asian cities being relatively low. This could be related to the fact that this model is trained on Flickr images, which are more diverse than other datasets. Figure 9 shows that this geographical bias is reduced by training on SLS for both AmosNet and NetVLAD.

5. Conclusions and Future Work

We have presented Mapillary SLS, a large collection of image sequences for training and evaluating place recognition algorithms. The data has been collected from Mapillary and contains over 1.6 million frames from 30 different cities over six continents. The gathered sequences span a period of seven years and places experience large perceptual changes due to seasons, constructions, dynamic objects, cameras, weather and lighting.

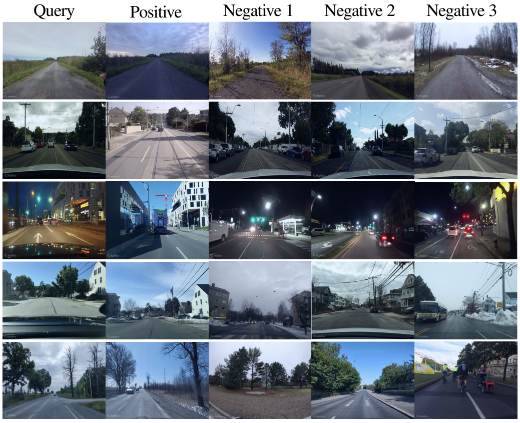


Figure 6: Triplets with multiple negatives. Hard-negatives are mined during training using our proposed sub-caching methodology. For each query-positive pair, the negatives that most violate the triplet constraint, $||q-p||_2^2+m<||q-n||_2^2$, are chosen. Here q, p, n refer to the cached embeddings for the query, positive and negative images. m is the margin.

Mapillary SLS contains the largest geographical and

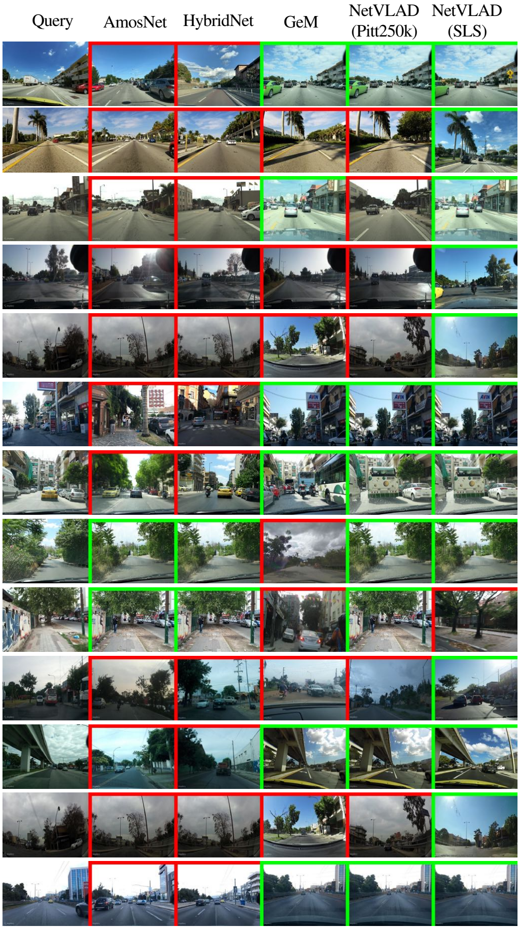


Figure 7: Qualitative comparison of different pre-trained networks as well as our NetVLAD model trained on Pittsburgh250k and Mapillary SLS. Training on SLS improves robustness towards weather changes and diverse vegetation such as palm trees. Green: true positive; Red: false positive.

temporal coverage of all publicly available datasets; and it is among the ones with the widest range of appearance variations and the largest number of images. All these features makes our dataset a valuable addition to the available data corpus for training place recognition algorithms. The many variation modes and the considerable size of realistic urban data make it particularly appealing for deep learning approaches and autonomous car applications.

We have also run extensive benchmarks on our dataset with previous state of the art methods to illustrate the difficulty of our dataset. We also introduce two new tasks: seq2im and im2seq. We propose new techniques to solve these tasks using models trained for im2im place recognition and evaluate several pre-trained models as well as models trained on SLS.

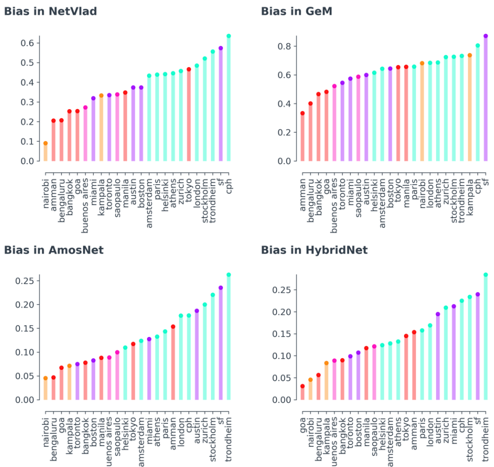
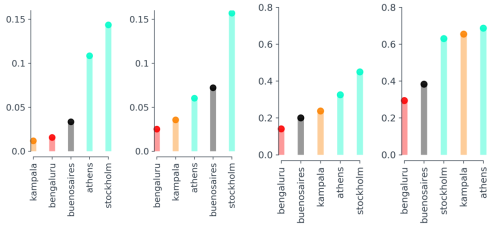


Figure 8: Geographical bias in 4 place recognition models. y-axis shows the top-5 recall. Cities are colored depending on their location: Africa (orange), Asia (red), South America (pink), North America (purple), Europe (turquoise).



(a) AmosNet∗ (b) AmosNet† (c) NetVLAD∗ (d) NetVLAD†

Figure 9: Bias reduction for both AmosNet and NetVLAD when trained on SLS and evaluated on the SLS test set. ∗/† indicates pretrained/trained models, respectively.

Although, the focus of the present paper is place recognition, Mapillary SLS is also useful for other computer vision tasks such as pose regression [22, 52], image synthesis (e.g., night-to-day translation [3]), image-to-gps [51, 56]), change detection, feature learning, scene classification using the OSM road tags and unsupervised depth learning [19, 26].

References

[1] OpenSfM. https://github.com/mapillary/OpenSfM. 3

[2] M. Angelina Uy and G. Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018. 2

[3] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964. IEEE, 2019. 8

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 1, 2, 3, 5, 6

[5] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 794–799, June 2011. doi: 10.1109/IVS.2011.5940504. 2, 3

[6] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of Michigan North Campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9): 1023–1035, 2015. 2, 3

[7] Z. Chen, O. Lam, A. Jacobson, and M. Milford. Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509, 2014. URL http://arxiv.org/abs/1411.1509. 2

[8] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. D. Reid, and M. Milford. Deep learning features at scale for visual place recognition. *CoRR*, abs/1701.05105, 2017. URL http://arxiv.org/abs/1701.05105. 2, 6

[9] Z. Chen, F. Maffra, I. Sa, and M. Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9–16, Sep. 2017. doi: 10.1109/IROS.2017.8202131. 2

[10] M. Cummins. Highly scalable appearance-only SLAM - FAB-MAP 2.0. *Proc. Robotics: Sciences and Systems (RSS), 2009*, 2009. 2, 3

[11] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011. 2

[12] J. M. Fácil, D. Olid, L. Montesano, and J. Civera. Condition-invariant multi-view place recognition. *CoRR*, abs/1902.09516, 2019. URL http://arxiv.org/abs/1902.09516. 2, 6

[13] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. 2

[14] S. Garg, V. M. Babu, T. Dharmasiri, S. Hausler, N. Sünderhauf, S. Kumar, T. Drummond, and M. Milford. Look no deeper: Recognizing places from opposing viewpoints under varying scene appearance using single-view depth estimation. *CoRR*, abs/1902.07381, 2019. URL http://arxiv.org/abs/1902.07381. 2

[15] S. Garg, N. Sünderhauf, and M. Milford. Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, page 027836491983976, 04 2019. doi: 10.1177/0278364919839761. 2

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 3

[17] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth. FAB-MAP+ RatSLAM: Appearance-based SLAM for multiple times of day. In *2010 IEEE international conference on robotics and automation*, pages 3507–3512. IEEE, 2010. 2, 3

[18] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. G. Jiménez. Training a convolutional neural network for appearance-invariant place recognition. *CoRR*, abs/1505.07428, 2015. URL http://arxiv.org/abs/1505.07428. 2

[19] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 8

[20] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 304–317, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_24. URL http://dx.doi.org/10.1007/978-3-540-88682-2_24. 3

[21] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311. IEEE Computer Society, 2010. 2

[22] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 8

[23] A. Khaliq, S. Ehsan, M. Milford, and K. D. McDonald-Maier. A holistic visual place recognition approach using lightweight cnns for severe viewpoint and appearance changes. *CoRR*, abs/1811.03032, 2018. URL http://arxiv.org/abs/1811.03032. 2

[24] H. Le, T. Hoang, and M. Milford. BTEL: A binary tree encoding approach for visual localization. *CoRR*, abs/1906.11992, 2019. URL http://arxiv.org/abs/1906.11992. 2

[25] M. Leyva-Vallina, N. Strisciuglio, M. L. Antequera, R. Tylecek, M. Blaich, and N. Petkov. Tb-places: A data set for visual place recognition in garden environments. *IEEE Access*, 7: 52277–52287, 2019. 2

[26] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 8

[27] M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92:89–95, 2017. 2

[28] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. 1, 2

[29] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. doi: 10.1177/0278364916679498. URL http://dx.doi.org/10.1177/0278364916679498. 2, 3

[30] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation*, pages 1643–1649. IEEE, 2012. 2

[31] T. Naseer, W. Burgard, and C. Stachniss. Robust visual localization across seasons. *IEEE Transactions on Robotics*, 34(2):289–302, April 2018. ISSN 1552-3098. doi: 10.1109/TRO.2017.2788045. 2, 3

[32] H. Noh, A. Araujo, J. Sim, and B. Han. Image retrieval with deep local features and attention-based keypoints. *CoRR*, abs/1612.06321, 2016. URL http://arxiv.org/abs/1612.06321. 3

[33] NRK. Nordlandsbanen: minute by minute, season by season, 2013. URL https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/. 2

[34] D. Olid, J. M. Fácil, and J. Civera. Single-view place recognition under seasonal changes. *CoRR*, abs/1808.06516, 2018. URL http://arxiv.org/abs/1808.06516. 2

[35] K. Ozaki and S. Yokoo. Large-scale landmark retrieval/recognition under a noisy and diverse dataset. *CoRR*, abs/1906.04087, 2019. URL http://arxiv.org/abs/1906.04087. 3

[36] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, June 2010. doi: 10.1109/CVPR.2010.5540009. 2

[37] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383172. 3

[38] A. Queensland University of Technology, Brisbane. Day and night with lateral pose change datasets, 2014. URL https://wiki.qut.edu.au/display/cyphy/Day+and+Night+with+Lateral+Pose+Change+Datasets. 2, 3

[39] F. Radenovic, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *CoRR*, abs/1711.02512, 2017. URL http://arxiv.org/abs/1711.02512. 2, 6

[40] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. *CoRR*, abs/1803.11285, 2018. URL http://arxiv.org/abs/1803.11285. 3

[41] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3

[42] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003. 2

[43] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. *CoRR*, abs/1501.04158, 2015. URL http://arxiv.org/abs/1501.04158. 2

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

[45] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015. 2

[46] M. Teichmann, A. Araujo, M. Zhu, and J. Sim. Detect-to-retrieve: Efficient regional aggregation for image search. *CoRR*, abs/1812.01584, 2018. URL http://arxiv.org/abs/1812.01584. 2

[47] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. 2016. 2

[48] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015. 2, 3

[49] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2346–2359, Nov 2015. doi: 10.1109/TPAMI.2015.2409868. 2, 3

[50] D. P. Vassileios Balntas, Edgar Riba and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.11. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.119. URL https://dx.doi.org/10.5244/C.30.119. 6

[51] N. Vo, N. Jacobs, and J. Hays. Revisiting im2gps in the deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2630, 2017. 8

[52] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017. 8

[53] A. B. Yandex and V. Lempitsky. Aggregating local deep features for image retrieval. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, Dec 2015. doi: 10.1109/ICCV.2015.150. 2

[54] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, and K. D. McDonald-Maier. Levelling the playing field: A comprehensive comparison of visual place recognition approaches under changing conditions. *CoRR*, abs/1903.09107, 2019. URL http://arxiv.org/abs/1903.09107. 1, 2

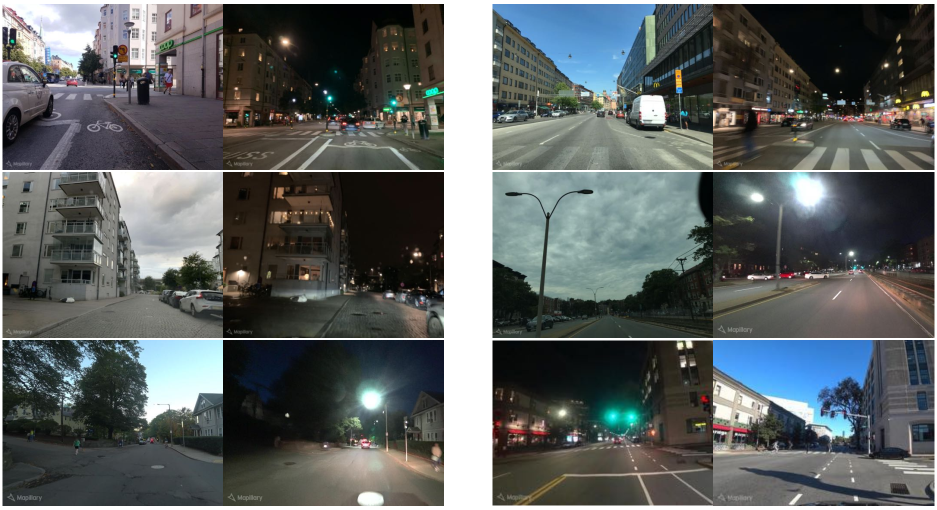
[55] A. R. Zamir and M. Shah. Image geo-localization based on multiplenearest neighbor feature matching using generalized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014. 2, 3

[56] E. Zemene, Y. T. Tesfaye, H. Idrees, A. Prati, M. Pelillo, and M. Shah. Large-scale image geo-localization using dominant sets. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):148–161, 2018. 8

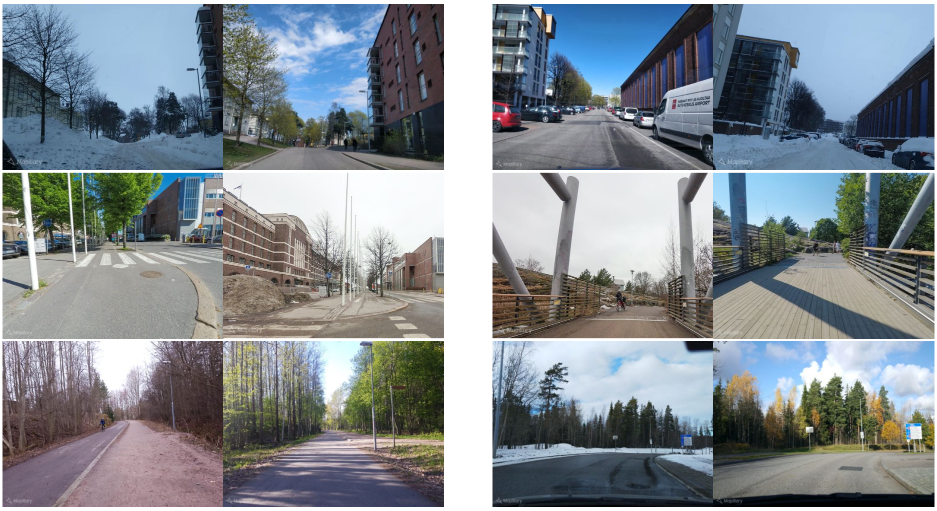
A.2 More Images

Place Recognition is a challenging problem due to a place's appearance change over time. In this appendix, we illustrate some of the many appearance changes that are covered in the Mapillary Street-Level Sequence dataset.

Day/Night Changes



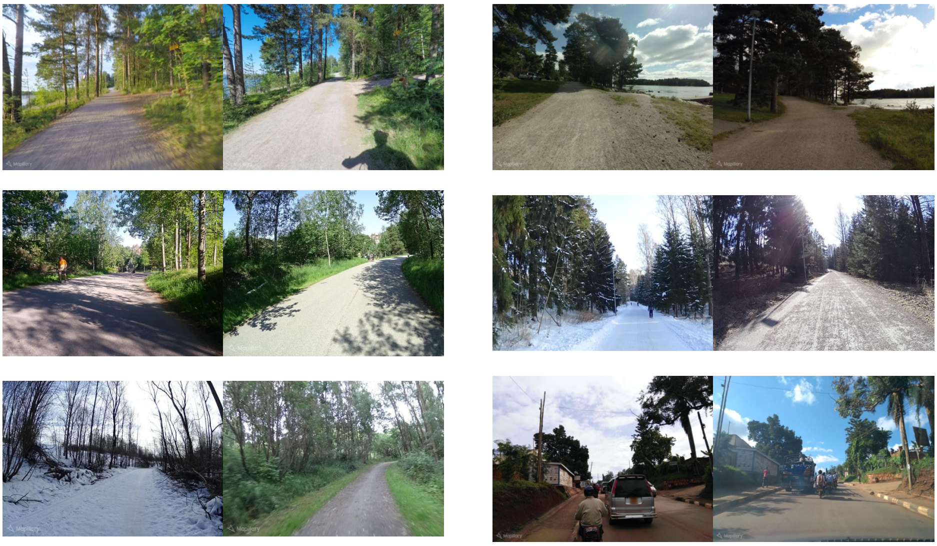
Seasonal Changes & Changing Weather



Urban Environment & Dynamic Objects

Suburban Environment & Varying Viewpoints

Rural Environment



A.3 More Seq-2-Seq Triplets

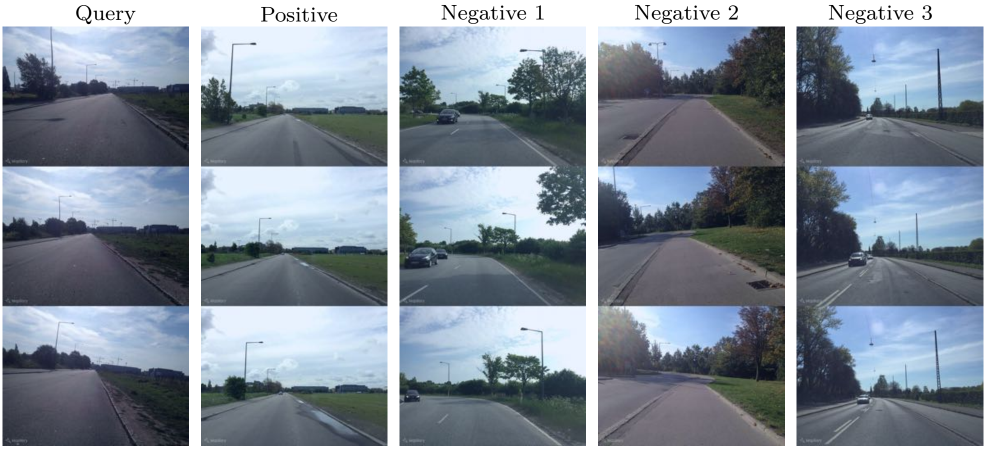


Figure A.1: Sequence triplets with multiple negatives mined with the same training procedure as the in the image to image case.

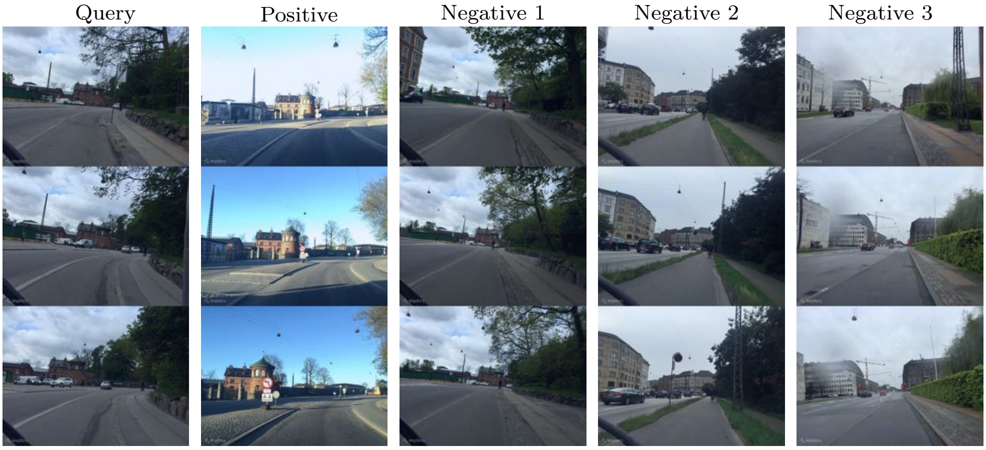


Figure A.2: Sequence triplets with multiple negatives mined with the same training procedure as the in the image to image case.

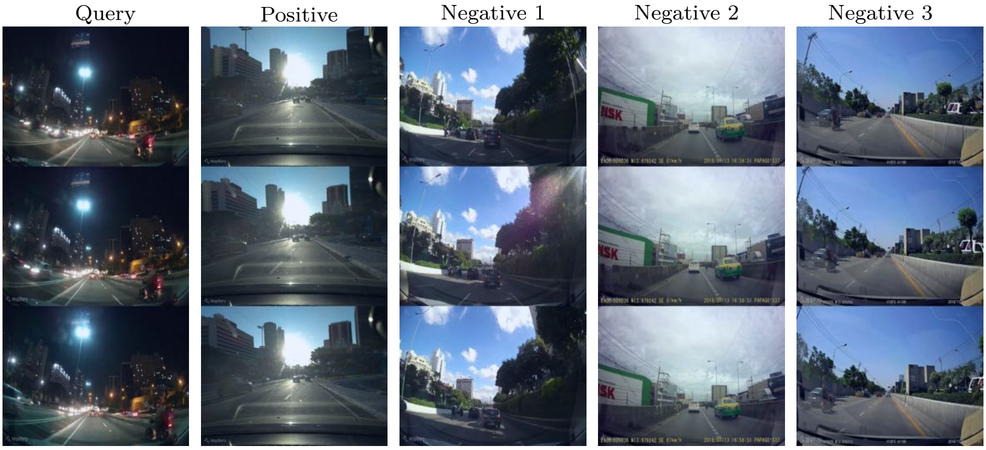


Figure A.3: Sequence triplets with multiple negatives mined with the same training procedure as the in the image to image case.

A.4 Model Attention

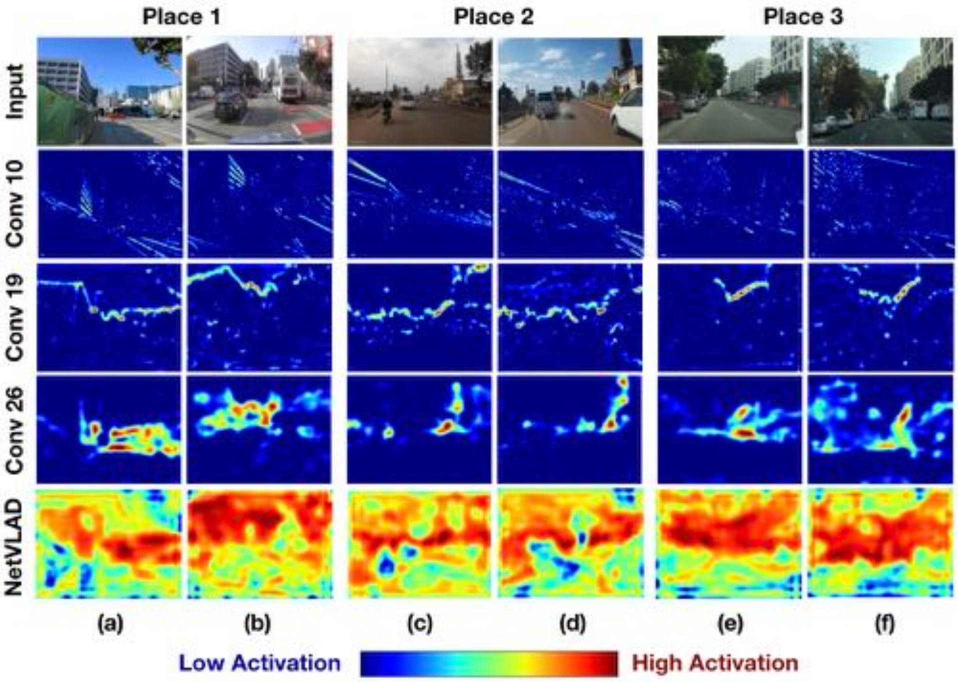


Figure A.4: The top row shows the input images. The remaining rows shows the feature activation for filters from convolutional layers 10, 19, 26 and the NetVLAD layer. These a visualized as heatmaps for original feature maps. Blue indicates low activation and red indicates high feature activation.

A.5 Curation of the Mapillary SLS dataset

In this section, we describe in more detail and with more examples the pipeline for curating the Mapillary Street-Level dataset. This section is more code specific and useful for replication of the dataset. The pipeline consists of 8 steps and exploits the sequence structure of which the Mapillary data is already stored.

A.5.1 Region of Interest

A region of interest is specified as a bounding box. This region is divided into n windows of size $500m^2$. Each window is analysed independently. The public Mapillary

API is called used to obtain all the sequences within the window.

A.5.2 Preprocessing

The data is then formatted, preprocessed and some preliminary calculations are performed. The preproccesing includes the following criteria:

- The sequences must be recorded by a combination of an unique user and timestamp. Where an unique timestamp is defined in intervals of 5 min.
- The sequence must have 80% consensus between the compass angle (cas) and the computed compass angle (ccas). Where consensus is defined as less than 30 degrees difference.
- The sequence must have all cas values less than 360 degrees.
- The sequence must be more than 5 frames long.
- Only the frames inside the $500m^2$ window are kept.

Furthermore, the moving orientation is calculated for each point. This is done first calculating the moving direction for each point:

$$[\Delta lat, \Delta lon]^T = [lat_{i-1}, lon_{i-1}]^T - [lat_i, lon_i]^T$$

And then calculating the signed angle between two vectors: moving direction and north $[0, 1]^T$ via.

$$\theta = tan^{-1}(\frac{\Delta lon}{\Delta lat}) - tan^{-1}(\frac{1}{0})$$

This angle is later used for filtering.

A.5.3 Split sequences

Each sequence is split into subsequences if there are more than 50 m between to consecutive frames. Again, it is required that the subsequences are at least 5 frames long.

A.5.4 Potential Clusters

The next step is to cluster the sequences according to their position and view direction. This is done by looking through all the subsequences and form potential clusters based on their distances to all other subsequences. This function uses a KD tree to efficiently find points with close distances. We convert latitude and longitude to x,y,z on a sphere to measure distances in meters.The following criteria is used.

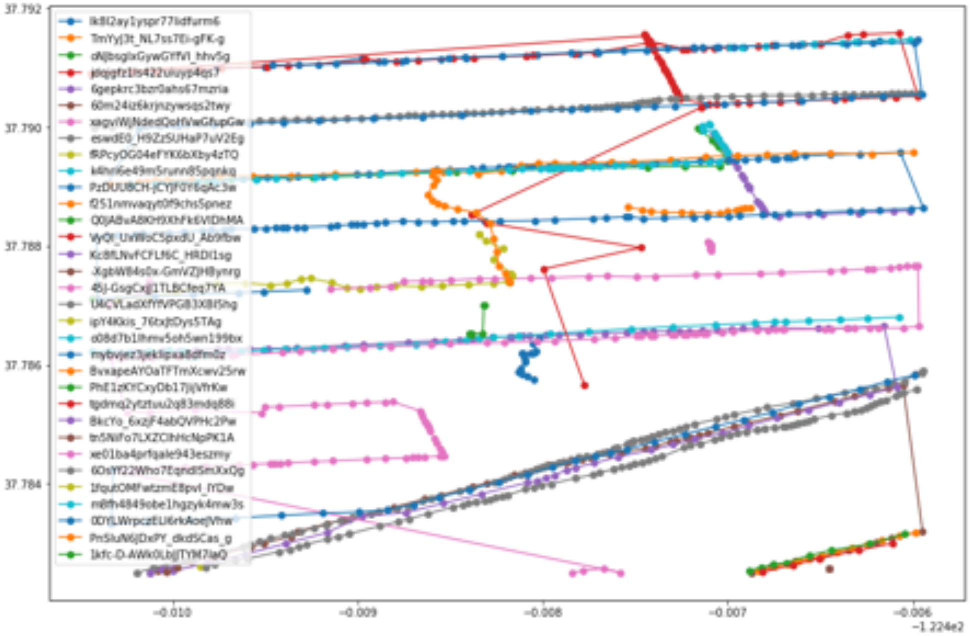


Figure A.5: Shows the results after the preprocessing step. Each sequence has an unique color. Each frame is a represented by a dot.

- Points from subsequence A and subsequence B are clustered together if their distance is less than 50 m.
- Points from subsequence A and subsequence B are clustered together if their difference between cas is less than 40 degrees.
- Points from subsequence A and subsequence B are clustered together if their difference between moving direction orientation is less than 40 degrees.

A.5.5 Add skipped images

We found that the above described requirements sometimes skipped some consecutive images, e.g. a subsequence could have the images [1,2,4,5], thus image 3 was missing. To compensate for this, we add all ”skipped” images in the sequence to get more smooth sequences.

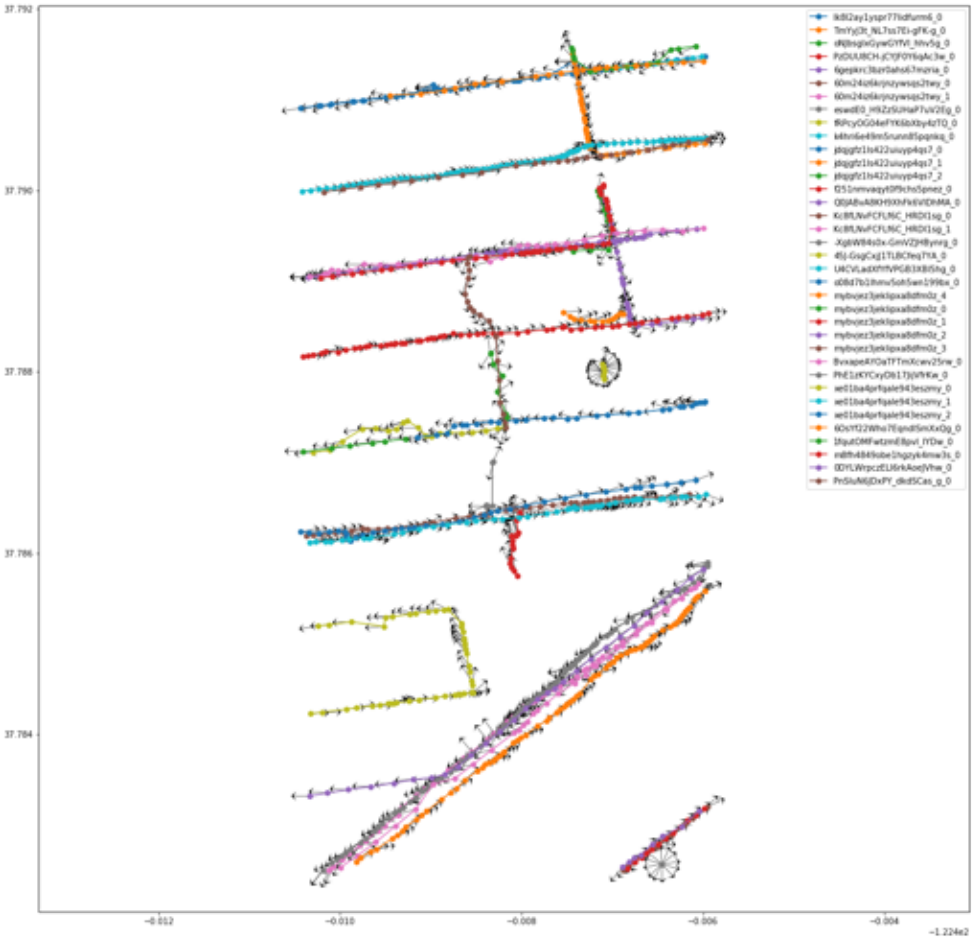


Figure A.6: Shows the subsequences. Each color is a an unique subsequence. The arrows indicate the view direction (cas) of each frame.

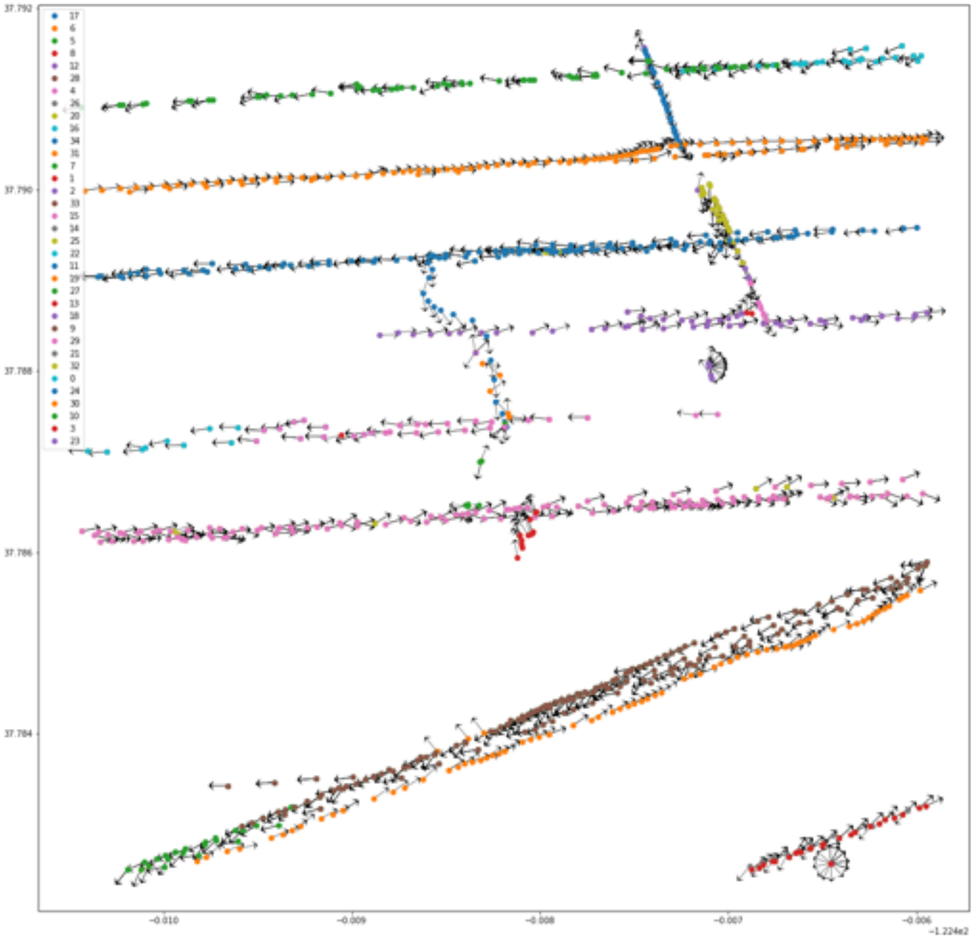


Figure A.7: Shows the potential clusters. Each cluster has a unique color. The arrows indicate the view direction (cas) of each point.

A.5.6 Prune Clusters

As many of the potential clusters are overlapping, we need to prune them in order to find the places where all subsequence are overlapping. In other words the potential clusters finds where A is close to B and C, where B is close to A and C, and where C is close to B and A. Now we want to find where A and B and C are close. This is done using the intersection of the sequences' unique image keys. This works with arbitrary many subsequences.

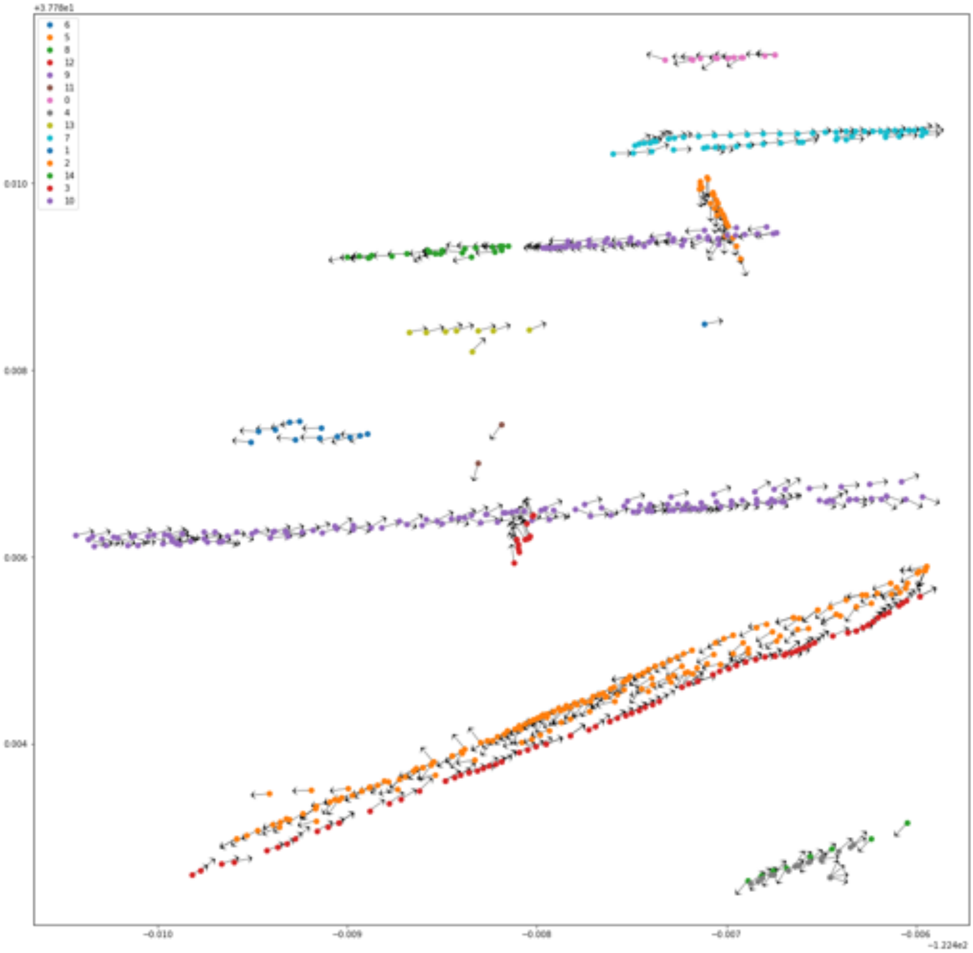


Figure A.8: Shows the pruned clusters. Each cluster has a unique color. The arrows indicate the cas of each point.

A.5.7 Filter Clusters

We then filter the clusters. Here we enforce the following requirements:

- Each subsequence in a cluster must have 2 or more unique dates
- Each subsequence in a cluster must have 5 or more frames.

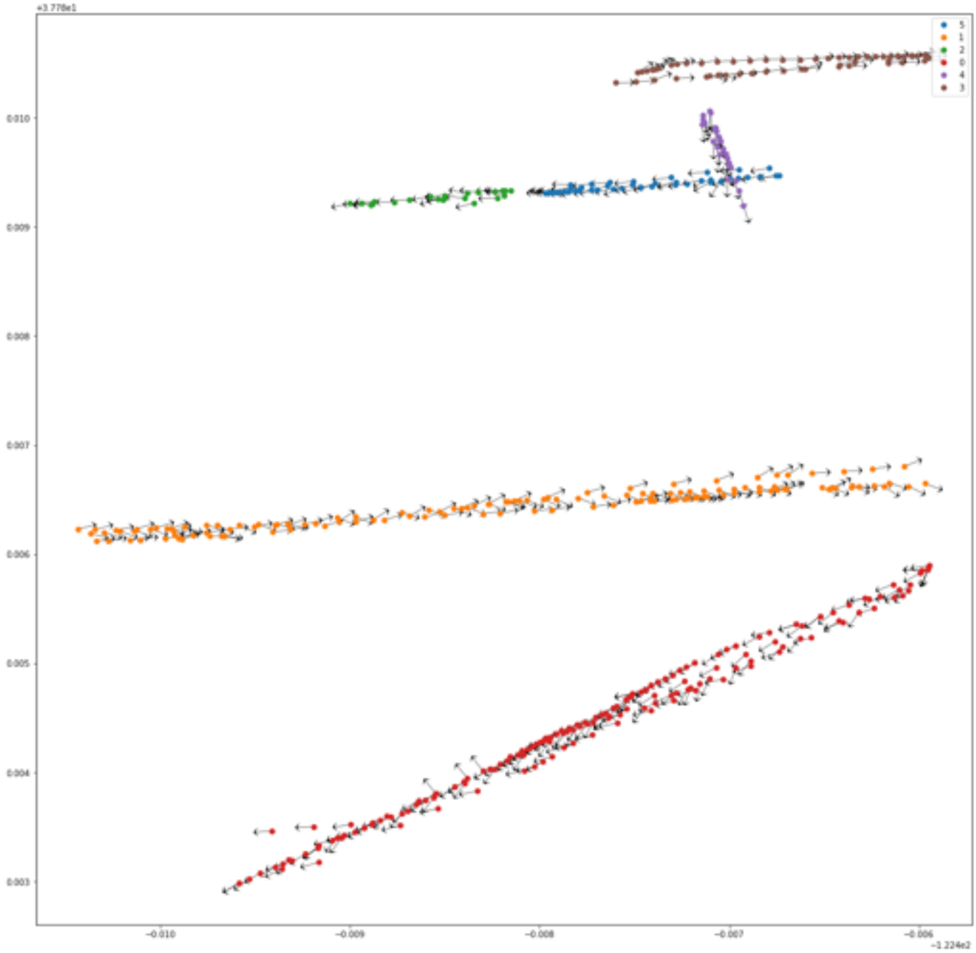
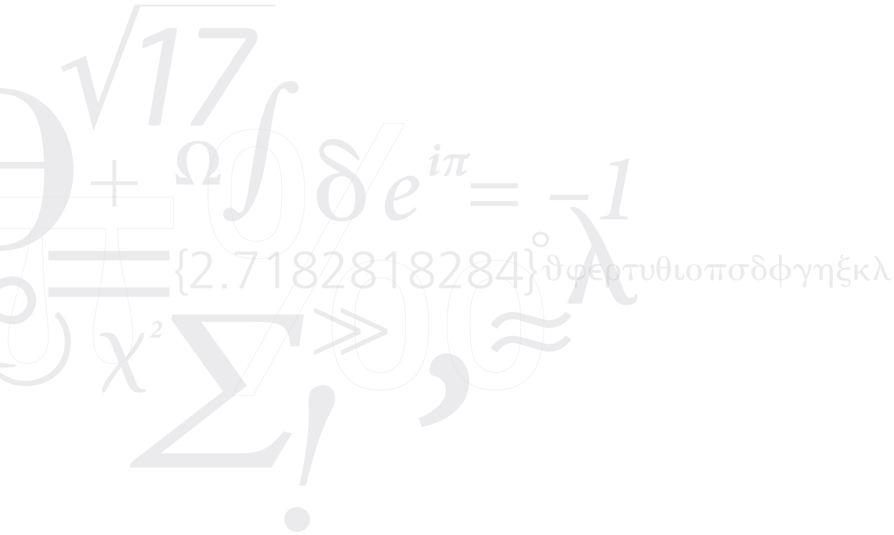


Figure A.9: Shows the filtered clusters. Each cluster has a unique color. The arrows indicate the view direction (cas) of each point.

Finally, the images are downloaded using the Mapillary public API


√17
Ω
e^iπ = −1
{2.7182818284}
ϑφερτυθιοπσδφγηξκλ
χ²
Σ!