IMPROVED IMAGE BASED LOCALIZATION USING SEMANTIC DESCRIPTORS

A Thesis Submitted to
the Graduate School of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
İbrahim ÇINAROĞLU

January 2021
İZMİR

To the my family who have empowered me with their presence:

my father, Muzaffer,

my mother, Nuran,

my sister, Bahar,

and my wife, Sinem.

ACKNOWLEDGMENTS

In life, we meet people in different times that will change and inspire your life. In this section, I would like to express my greatest gratitude to the people who have supported me during this crazy journey. I am very fortunate to met, work and have such people in my life.

First and foremost I would like to express my sincere gratefulness to my Ph.D. supervisor Assoc. Prof. Dr. Yalın BAŞTANLAR for the limitless support during my studies. I feel very lucky to work with him with his patience, motivation and leniency during these past three years. This thesis would not have been completed without his generosity in sharing knowledge.

Furthermore, I would like to thank my thesis monitoring committee members Assoc. Prof. Dr. Mustafa ÖZUYSAL, Assoc. Prof. Dr. Şevket GÜMÜŞTEKİN, also to examining committee members Assoc. Prof. Dr. Hacer YALIM KELEŞ and Assoc. Prof. Dr. Zerrin IŞIK for all the valuable feedback they have provided to me.

I also would like to thank all my friends and colleagues from Computer Engineering Department of İYTE, especially to my office mates for their support and friendship during thesis work.

I wish to express my special thanks to my dad Muzaffer ÇINAROĞLU who passed away during my Ph.D. study. I could never succeed in this difficult journey without his endless life experience that he left behind.

I would like to express my deepest gratitude to my amazing wife Sinem ÇINAROĞLU with her inexhaustible mental and academic support, I feel myself the luckiest man ever to meet and marry with her. I am also indebted to my mom Nuran ÇINAROĞLU who kept me highly motivated, to my sister Bahar GÖRGÜLÜ with her cheerful family, for all the times they were there with their unconditional love when I needed them.

ABSTRACT

IMPROVED IMAGE BASED LOCALIZATION USING SEMANTIC DESCRIPTORS

Place recognition and Visual Localization (VL) for autonomous driving are the topics that keep their popularity in the field of Computer Vision. In this study, semantically improved Hybrid-VL approaches, that use localization aware semantic information in street-level driving images are proposed. Initially, Semantic Descriptor (SD) is extracted from semantically segmented images with a Convolutional Neural Network (CNN) trained for localization task. Then, image retrieval based VL task is performed using the approximate nearest neighbor search (ANNS) in *2D-2D* matching context. This proposed method is named as SD-VL and its success is compared with the success of the state-of-the-art Local Descriptor (LD) based VL method (LD-VL) which is frequently used in the literature. Furthermore, with the aim of alleviating the shortcomings of both two methods, a novel decision-level Hybrid-VL (Hybrid-VL$_{DL}$) method is proposed by combining SD-VL and LD-VL in post-processing stage. Also feature-level Hybrid-VL (Hybrid-VL$_{FL}$) method is proposed in order to produce automatically tuned hybrid result.

These proposed VL methods are examined on two challenging benchmarks; *RobotCar Seasons* and *Malaga Downtown* Data Sets. Moreover, a new VL data set *Malaga Streetview Challenge* is generated by collecting *Google Streetview* images on the same path of *Malaga Downtown* in order to observe impact of environmental and wide-baseline changes. This newly generated test set will be useful for researchers studying in this field. After all, the proposed semantically boosted Hybrid-VL$_{DL}$ method is able to increase localization performance on both *RobotCar Seasons* and *Malaga Streetview Challenge* data sets by 11.6% and 4.5% *Top-1 recall@5*, and 4% and 5.4% *recall@1* scores respectively. Additionally, reliability of our hyper-parameter (W) based Hybrid-VL$_{DL}$ approach is supported by very close performance of the Hybrid-VL$_{FL}$ method.

ÖZET

ANLAMSAL BETİMLEYİCİLER İLE GELİŞMİŞ İMGE TABANLI KONUMLANDIRMA

İnsansız araçlar için yer tespiti ve İmge Tabanlı Konumlandırma (İTK) Bilgisayarlı Görü alanındaki popülerliğini koruyan araştırma konularının başında yer almaktadır. Bu çalışmada, konuma duyarlı anlamsal bilgiye dayalı olarak sürüş senaryosu içeren sokak düzeyindeki imgeler üzerinde çalışan Hibrid-İTK yaklaşımı önerilmiştir. Bu amaç dorğrultusunda ilk aşama olarak, Evrişimli Sinir Ağı (ESA) üzerinde konumlandırma hedefi ile eğitilen Anlamsal Betimleyici (AB) elde edilmiştir. Ardından, iki boyutlu (2B-2B) imge eşleştirmesine dayalı olan konumlandırma yöntemimiz yaklaşık en yakın komşu arama (YEKA) yaklaşımı ile gerçeklenmiştir. AB-İTK olarak isimlendirilen bu yöntemin konumlandırmadaki başarısı, literatürde sıklıkla kullanılan Yerel Betimleyici (YB) tabanlı İTK yöntemi (YB-İTK) ile kıyaslanmıştır. Buna ek olarak, bahsi geçen bu YB-İTK ve AB-İTK yöntemleri birbirlerinin eksikliklerini tamamlayacak şekilde son işlem evresinde bir araya getirilmiş ve önerilen bu yeni yöntem karar-düzeyinde Hibrid-İTK (Hibrid-İTK$_{KD}$) yöntemi olarak adlandırılmıştır. Ayrıca, otomatik olarak en iyi şekilde ayarlanmış hibrid bir sonuç üretmek için öznitelik-düzeyinde Hibrid-İTK (Hibrid-İTK$_{OD}$) yöntemi önerilmiştir.

Önerilen bu İTK yöntemlerinin başarısı, literaturde kriter olarak kabul edilen RobotCar Seasons ve Malaga Downtown veri setleri üzerinde sınanmıştır. Ayrıca Malaga Streetview Challenge veri seti, çeveresel ve referans noktasındaki değişimlerin etkisini gözlemleyebilmek adına özel olarak bu çalışma için, Malaga Downtown ile aynı güzergahdaki Google Streetview imgelerinin bir araya getirilimesi ile oluşturulmuştur. Yeni oluşturulan veri seti bu alanda çalışan araştırmacılar için yararlı olacaktır. Önerilen Hibrid-İTK$_{KD}$ yöntemi ile RobotCar Seasons ve Malaga Streetview Challenge veri setleri üzerindeki konumlandırma başarısı, sırası ile 1.6% - 4.5% *Top-1 recall@5*, ve 4% - 5.4% *recall@1* oranlarında artırılmıştır. Ek olarak, önerilmiş olan hiper-parametre (W) tabanlı Hibrid-İTK$_{KD}$ yaklaşımının güvenilirliği hemen hemen aynı deneysel sonuçların Hibrid-İTK$_{OD}$ tarafından elde edilmesi ile desteklenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

VL	Visual Localization
Hybrid-VL	Hybrid Descriptor Based Visual Localization
SD	Semantic Descriptor
CNN	Convolutional Neural Network
ANNS	Approximate Nearest Neighbor Search
SD-VL	Semantic Descriptor Based Visual Localization
LD	Local (Key-Point) Descriptor
LD-VL	Local Descriptor Based Visual Localization
Hybrid-VL$_{DL}$	decision-level Hybrid-VL
Hybrid-VL$_{FL}$	feature-level Hybrid-VL
IMU	Inertial Measurement Unit
GPS	Global Positioning System
Geo-Tagged	Geographically Labeled
FLANN	Fast Library for Approximate Nearest Neighbors
NN	Neural Network
BOF	Bag Of Features
BOF	Bag Of Features
k-d	*k* Dimensional
TNS	Threshold for Negative Sampling
CL	Convolution Layer
FCL	Fully Connected layer

CHAPTER 1

INTRODUCTION

Information on location of a mobile device (could be a pedestrian or a vehicle) is critical for city-scale navigation and other location-based services. Generally, Inertial Measurement Unit (IMU) or Global Positioning Systems (GPS) are used for navigation and localization tasks. However it is a known fact that GPS may not be activated in some cases because of various environmental factors like dense vegetation and structures, tunnels, military zone etc. One of the well-defined but extremely challenging state of the art solution against this inability is Visual Localization (VL). This problem will shortly be defined with that question: "Given an image taken in a known environment, can an agent geographically localize itself?". From image retrieval point of view, the task is retrieving the best matching images among the geographically labeled (Geo-Tagged) reference images for a given query image. Owing to known localization information of the retrieved best image we will determine the location of the query image. Visual recognition is also a growing research field, as evidenced by dedicated workshops in renowned international conferences of computer vision. One reason to this ongoing interest is its potential use in autonomous vehicles.

1.1. Motivation of the Study

Many articles and studies have mentioned difficulties that changing long-short term conditions have brought into VL. Achieving a stable visual place recognition can be difficult due to different type of challenges; drastic changes will occur on the same scene (Fig. 1.1), different places may look very similar known as perceptual aliasing, kinds of distortions like shining, and places may not always be revisited from the same viewpoint as before. In order to evaluate the algorithms that attack these problems, valuable VL benchmark data sets served in numerous studies (Blanco-Claraco et al., 2014; Carlevaris-Bianco et al., 2016; Maddern et al., 2017; Sattler et al., 2018) that include all these mentioned challenges for a driving scenario. Thus, the proposed (Hybrid-VL) methods are evaluated on most often used Robotcar Seasons and Malaga Downtown benchmark data sets like they are used in many recent studies (Germain et al., 2018; Naseer et al., 2014; Piasco et al., 2019; Seymour et al., 2018; Stenborg et al., 2018). Also, additional test set including street-level driving images is collected from *Google Streetview* on the same path of Malaga Downtown in order to observe impact of long/short term environmental and wide-viewpoint changes. Moreover, this newly generated test set based on Malaga Urban (Blanco-Claraco et al., 2014) data set will be useful to the community of VL area.

Previous studies on VL commonly has expressed an image with local descriptors (LD) that is created from points of interest, such as SIFT (Lowe, 2004), SURF (Bay et al., 2006), FAST (Rosten and Drummond, 2006), Harris (Harris et al., 1988), BRIEF (Calonder et al., 2010) etc.. And also, another frequently used descriptor extraction technique is holistic(general) descriptors, such as GIST (Oliva and Torralba, 2006), HOG (Dalal and Triggs, 2005) etc. in which images are expressed as a whole. Latest examples of effective VL studies gain their success by using improved version of LD's (Disloc (Arandjelović and Zisserman, 2014), VLAD (Jégou et al., 2010), DenseVLAD (Torii et al., 2015)) and their CNN modeled versions which are supported by deep learning. So that in this study, images are represented with localization aware descriptors that use CNN based NetVLAD (Arandjelovic et al., 2016) model and NetVLAD based LD-VL. These approaches are accepted as baseline methods.

While comparing the similarity of images in VL task a question that may initially arise is; we make comparison in which space? From the side of this question, while some VL studies make comparison in two-dimensional image space (*2D-2D* Matching), on the other hand there are also studies that firstly construct the *3D* model of the localization map then make comparison between *3D* points of reconstructed scene and *2D* points of perspective image (*2D-3D* Matching). Also these *2D-3D* Matching and *3D-3D* Matching methods will be named as a structure-less and structure-based localization respectively. There are several studies (Radenović et al., 2016, 2018) using *3D* structure-based localization with many frequently used LD methods (Cao and Snavely, 2014; Irschara et al., 2009; Sattler et al., 2015, 2012; Zeisl et al., 2015).

However *2D-2D* Matching approaches still keep their popularity using several advanced LD (Arandjelovic et al., 2016; Arandjelović and Zisserman, 2014; Perronnin and Dance, 2007; Sattler et al., 2016; Torii et al., 2015). These approaches owe their popularity to be less costly in comparison with *2D-3D* matching based approaches with almost equally successful results. These reasons are widely explained in the following two works. Camposeco et al. (2018) focused on to solve the pose estimation problem of calibrated pinhole and generalized cameras w.r.t. a Structure-from-Motion (SfM) model. They compared both 2D-3D correspondences as well as 2D-2D correspondences. Absolute pose approaches are limited in their performance because of the quality of the 3D point triangulations and the structuring accuracy of the 3D model. On the other side, relative pose approaches are more accurate, also they tend to be far more computationally costly and often return dozens of possible solutions. In order to cope with this trade-off they propose a new RANSAC based approach. By this way they manage automatically choosing the best type of solver to use at each iteration in a data driven way. These RANSAC based solvers can range from pure structure-based or structure-less solvers, to any possible combination of hybrid solvers (i.e. using both types of matches) in between. A number of these new hybrid minimal solvers are also presented in this paper. Consequently both synthetic and real data experiments approve their approach to be as accurate as structure-less approaches, while staying close to the efficiency of structure-based methods. Torii et al. (2019) emphasized the trade-off between structured and structure-less VL methods and its importance in autonomous navigation. In their paper, authors demonstrated experimentally that large-scale 3D models are not strictly necessary for accurate visual localization and they created reference poses for a large and challenging urban data set. They showed that combining image-based methods with local reconstructions results in a pose accuracy similar to recent structured methods. So that, their results suggest that we might want to reconsider the current approach for accurate large-scale localization. Under the light of these examined studies above, we adopted the proposed method in *2D-2D* matching space which is less expensive than a structured one with almost equally successful performance.

Secondly, the following question may come to mind; how we can perform this similarity comparison in an efficient way? After analyzing studies including both approaches (*2D-2D* Matching & *2D-3D* Matching), since the geographic location of the retrieved database image serves as an approximate position of the query image, it is clear that we must apply robust descriptor matching method as a core step of VL. In this context, efficiency of using Approximately Nearest Neighborhood search (ANNS) method for high-dimensional data matching is highlighted in several studies (Muja and Lowe, 2009b, 2014). Therefore in this study Fast Library for Approximate Nearest Neighbors (FLANN (Muja and Lowe, 2009a)) tool is employed in order to retrieve the most similar images.

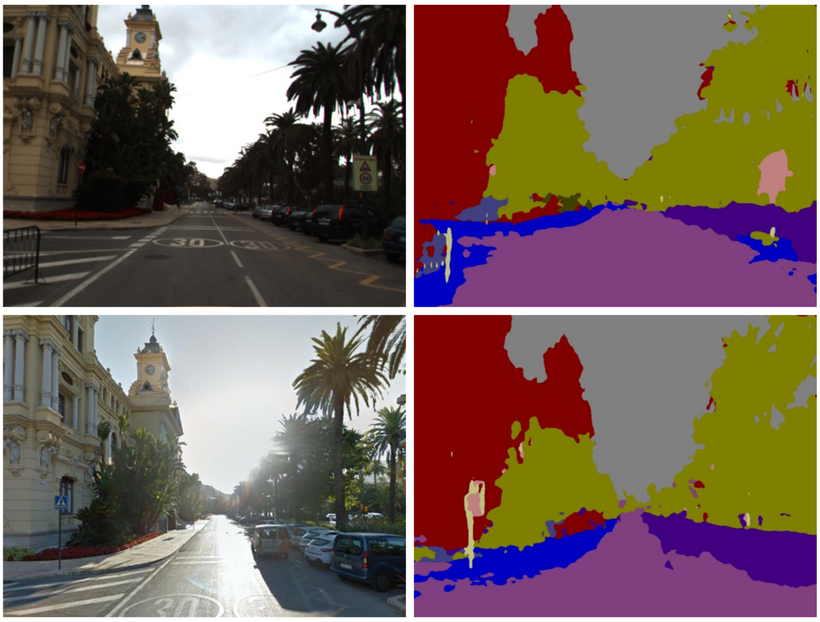


Figure 1.1. Importance of semantic segmentation against appearance changing.

Semantic segmentation of an image will be more stable than standard LD approaches against considerable illumination and seasonal changes as depicted in Figure 1.1. On the left side of this figure, two images of the same scene with considerable illumination and seasonal changes are displayed. On the right side, their semantic segmentation results are illustrated. Standard methods (LD based) have low performance for such cases, where more stable semantic segmentation can help. There are studies in which semantic information in an image is used in order to improve localization performance differently from this thesis. Ondruska et al. (2016) incorporated the semantic segmentation to city-scale tracking task with a different kind of Neural Network (NN). In their work they proposed a novel recurrent NN architecture which filters laser measurements to make inference on location of a object in both visible and occluded areas. Singh and Košecká (2012) produced a hand-crafted descriptor after semantic segmentation of images, and they used this descriptor for grouping the scenes such as on a street, in front of a building or at a crossroads. In another LD based study, Mousavian et al. (2015) eliminated local descriptors of objects (e.g. trees) that do not come from man-made structures by using extracted semantic labels. Thus, they increased the wights of features coming from man-made structures compared to non man-made structures since natural structures have low chance of healthy matching. In his another work (Mousavian and Kosecka, 2016), for each query image researchers identified which buildings are in the image as well as the orientation of building facades by means of semantic segmentation. Then they used the identity of the buildings and orientation of building facades and the map, in order to find the probability distribution for the location of image. There are also studies (Schönberger et al., 2018; Stenborg et al., 2018; Toft et al., 2018) that incorporated structure-based (*2D-3D*) approach with semantic information. In first example of semantic clues based *2D-3D* matching study (Toft et al., 2018), accuracy of image matching was also checked semantically. In another work using *3D* scene reconstruction, Schönberger et al. (2018) prepared a dictionary for semantic content and expressed the scene as bag of features (BOF) for this reconstructed scene. Also Stenborg et al. (2018) considered the problem of visual localization against logn-term changes in the context of *2D-3D* matching space. They managed to label an environment semantically with its all corresponding points by means of semantically segmented images. Then they demonstrated that a vehicle localization without the need for detailed feature descriptors (SIFT, SURF, etc.) will be achieved by efficient usage of labeled 3D point maps. In this way, instead of depending on hand-crafted feature descriptors, they discussed on the training of an image segmenter. In these similar studies, semantic cues were used to improve localization performance, however study is the first to perform localization with direct usage of semantically segmented images.

1.2. Contributions of the Thesis

The contributions of this thesis study are provided below:

- Firstly, semantic information is extracted from equally divided parts of semantically segmented images as a novel hand crafted semantic descriptor (SD) for VL in *2D-2D* matching space which is called as *non-learnt* SD-VL. Differently from the first one, a new SD is trained with a CNN model including NetVLAD (Arandjelovic et al., 2016) layer using semantically segmented images as an input, then this captured semantic representation is used directly for VL that is named as *learnt* SD-VL method. Both query and database images are segmented by applying the up-to-date CNN based semantic segmentation method (DeepLabv3+ (Chen et al., 2018)) invented by *Google Research*. Also, all proposed localization approaches are based on 2D-2D image matching and their semantic segmentation results. It is much cheaper than the approaches that require the semantic 3D reconstruction of the environment (Schönberger et al., 2018; Stenborg et al., 2018; Toft et al., 2018).
- Secondly, fine tuned Hybrid-VL$_{DL}$ is proposed to combine the proposed SD-VL and the baseline LD-VL methods in post-processing stage. Also Hybrid-VL$_{FL}$ that is based on NN trained with triplet loss is proposed in order to produce automatically tuned hybrid result. Improved localization performances are measured with a frequently used evaluation metric; which computes percentage of queries correctly located under changing distance (meter) thresholds for changing top *N* retrieved images.

1.3. Organization of the Thesis

In Chapter 2, different approaches on VL are broadly examined individually within the scope of descriptor types, descriptor matching algorithms, benchmark data sets and on the subject of semantic representation context. Also necessary background and literature review are given in this chapter.

Implementation details of the proposed method is explained in Chapter 3. Firstly, preparation of newly generated *Malaga Streetview Challenge* data set is described. Secondly different semantic segmentation methods are compared. After that in accordance with the purpose of this study, adoption of SD-VL method trained with semantically segmented image and generation of proposed decision/feature-level Hybrid-VL methods are introduced. In Chapter 4, used evaluation metric is described firstly, then experimental results are presented with kinds of case studies. Finally, the conclusion and future work of this thesis is given in Chapter 5.

CHAPTER 2

BACKGROUND AND REVIEW OF LITERATURE

In the literature there are different approaches on VL. Image based localization is the subject of different application areas which may take place in outdoor or indoor environments. These areas will be exemplified such as automotive industry, tourism, health care, safety and visual surveillance. Although there are some works for indoor environment, majority of these researches have concerned with outdoor environment especially for autonomous vehicles. Basically, a typical image retrieval based VL task for outdoor environment will be demonstrated with Figure 2.1. The task is defined as

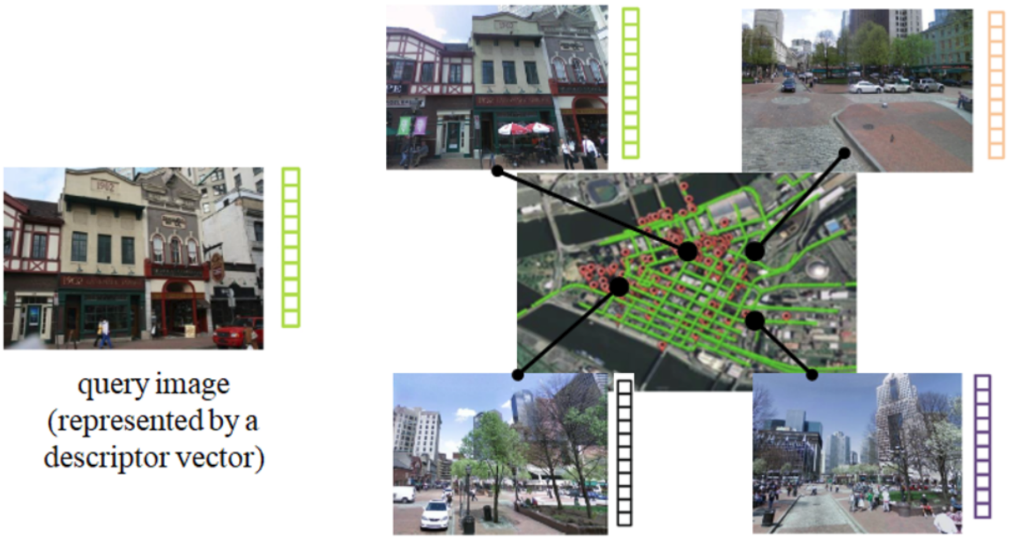


Figure 2.1. On the left, we see a query image. On the right, we see a district with a database of images with known GPS coordinates. Retrieval from the database is based on the similarity of descriptor vectors. The GPS location of the image retrieved from the database serves as the position estimate of the query image. If a correct match is retrieved, then the localization is successful.

retrieving the best matching images among the geographically labeled (Geo-Tagged) database images for a given query image. This challenging job is mainly achieved by combination of these two stages: image representation and image matching. Also note that, in a characteristic VL system the localization path which is illustrated by green lines in Figure 2.1 is already given as a database image collection which is named as prior map. Considerable improvements in VL have been recorded in the last decade especially thanks to the usage of deep learning techniques. Therefore recent works generally use trainable LD based descriptors to represent images and ANNS for image matching.

Description of these techniques are detailed with referenced studies in the following subsections. For a clear understanding, literature review is organized according to these sub-contexts; descriptor types, descriptor matching algorithms, benchmark data sets and semantic knowledge usage respectively.

2.1. Descriptor Types

Description techniques in VL are divided into two main categories: those that determine the interesting points of an image then just concentrates on them; and others that describe the whole scene, without a determining process. One example of the first category is Scale-Invariant Feature Transforms (SIFT (Lowe, 1999, 2004)), and in this study distinctive invariant features were extracted from images which also can be used to carry out reliable matching between changing views of an object or scene. Thanks to the these scale and rotation invariant features, image matching becomes more robust against change in 3D viewpoint, affine distortion and change in illumination. All these valuable properties makes this descriptor highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also showed up a new method to use these features for object recognition. This recognition is performed by robust nearest-neighbor algorithm based matching between single features and features collection comes from known objects. Then they operated Hough transformation to find out clusters belonging to a single object. At the last step, verification by least-squares solution for consistent pose parameters is performed. Therefore, this approach provides us robust object identification among collection with near real-time performance for a recognition task.

Bay et al. (2006) presented the another commonly used scale and rotation invariant local feature descriptor named as SURF (Speeded-Up Robust Features). This feature approximates or even outperforms previously proposed LD descriptors with respect to repeatability, distinctiveness, and robustness, despite it can be computed and matching much faster. This improvement is managed by trusting on integral images for image convolutions. In a more detailed definition, they build this descriptor on successful existing detectors (Hessian matrix-based measure for the detector) and a descriptors (distribution-based descriptor) by simplifying these methods. Finally, superiority of their method - combination of novel detection, description, and matching steps- is demonstrated in the context of a real-life object recognition application.

Additionally there have been many other LD like Harris Corner Detector (Harris et al., 1988), FAST (Rosten and Drummond, 2006), BRIEF (Calonder et al., 2010) and ORB-BRIEF (Rublee et al., 2011) which have been proposed for efficient image comparison. We know that, LD's first need a detection process which finds the interesting points of the image to accept as local features. In contrast, global image descriptors such as Gist (Oliva and Torralba, 2001, 2006) do not have a detection phase but process the whole image regardless of its content. This difference in their processes makes these descriptors have different advantages and disadvantages. LD's can also be combined with metric information to give capability of metric corrections for localization (Andreasson and Duckett, 2004; Davison et al., 2007; Konolige and Agrawal, 2008). On the other side, global descriptors do not have the same usability, and furthermore, whole-image descriptors are more vulnerable against a changing in robot's pose than local descriptor methods, because whole-image descriptor comparison methods tend to assume that the camera viewpoint remains similar. In order to tackle with this problem, Milford and Wyeth (2008) performed circular shifts, also a solution by combining a bag-of-words approach with a Gist descriptor method was proposed in different studies (Murillo and Kosecka, 2009; Murillo et al., 2012). Whole-image descriptors are more pose dependent than LD's, despite that LD perform poorly against changing lighting conditions (Furgale and Barfoot, 2010) and are comprehensively outperformed by global descriptors at application of place recognition in changing conditions (Milford and Wyeth, 2012; Naseer et al., 2014). Starting from incorporating the known shortcomings of both descriptor types, McManus et al. (2014) applied the well known global descriptor HOG (Dalal and Triggs, 2005) on segmented parts of an image to learn condition invariant scene signatures. They firstly divided a image into segments than employing this whole image descriptor on them. By this way they managed to combine conditional in-variant power of global descriptor with pose in-variant power of local features.

Under the light of comparison between local and global descriptors in the previous paragraphs, foremost works have preferred to construct their VL structure on LD's and their developed versions as analyzed in this paragraphs. We should note that, using LD's in BOF (Bag Of Features) approach is the classical way for representing a image in VL task.

The BOF approach groups LD's for representing an image. In this approach, firstly a dictionary which includes *k* number of "visual words" is defined. This definition is usually performed by k-means clustering method. LD's that comes from all images are assigned to the closest centroid, then histogram of the assignment of all image descriptors to visual words provides the BOF representation. As a result, k-dimensional vector is generated that is subsequently normalized and also kinds of histogram normalization methods are introduced. BOF vector is generally normalized using the Manhattan distance, other common choice is Euclidean normalization. Then, components of gained vector are weighted by *idf* (inverse document frequency) calculation with a different type of weighting scheme (Nister and Stewenius, 2006; Sivic and Zisserman, 2003). Moreover, there are developed version of BOF methods (Philbin et al., 2008; Van Gemert et al., 2009) that operates soft quantization technique instead of k-means. Jaakkola and Haussler (1999) introduced a new powerful tool in order to transform an incoming variable-size set of independent samples into a fixed size vector representation. As an advance version of Fisher kernel, Perronnin and Dance (2007) carried out it in the context of image classification. They model the visual words with a Gaussian mixture model, restricted to diagonal variance matrices for each of the *k* components of the mixture.

Also there are hand-crafted LD's based enhanced descriptors which are especially constructed for VL task. Disloc (Arandjelović and Zisserman, 2014; Aubry et al., 2014) is a state-of-the-art method based on the BOF representation and Hamming embedding (Jegou et al., 2008). Disloc uses the density of the Hamming space in order to give less weight to features which is found on repeating structures while keeping the impact of unique features. Some works combined Disloc descriptors with the geometric burstiness weighting scheme (Sattler et al., 2016) for usage in VL. However, Disloc is based on the BOF paradigm and therefore it needs to store an entry for each image feature in an inverted file. As a result of this necessity, representation leads to large memory requirements for large-scale scenes.

In order to struggle with this inefficiency, Jégou et al. (2010) proposed a novel representation: vector of locally aggregated descriptors which is named as VLAD. They addressed the problem of searching the most similar images in a very large image database (ten million images or more). And they underlined the incapability of both BOF and LSH (Locality Sensitive Hashing)(Kulis and Grauman, 2009) approaches against such a large scale image searching scenario. Also it was mentioned in their study, limited small vocabulary sizes of BOF technique will yield lower search accuracy. Differently from these disadvantages, VLAD optimizes these three constraints: the search accuracy, its efficiency and the memory usage of representation and these optimizations were achieved by means of the given three steps below:

1. Compose local image descriptors into a compact vector.

2. Reduce the dimension of these vectors in the best way.

3. Apply an efficient indexing methodology for searching.

On behalf of the first step, they actually combined the BOF and Fisher kernel (Perronnin and Dance, 2007), and managed to obtain a compact representation by aggregating the SIFT descriptors. In other words they simplified the Fisher kernel method with BOF paradigm.

In BOF approach, firstly a dictionary $C = \{c_1, ...c_k\}$ including k visual words is generated via k-means. Then each local descriptor x is assigned to its nearest visual word

$c_i = NN(x)$. The main objective of the VLAD descriptor is collecting the differences $x - c_i$ of the vectors where x assigned to c_i, and this collection is operated for each visual word c_i. By this way distribution of the vectors with respect to the center is built up and dimension D of this representation becomes $D = k$ x d with a d-dimensional local descriptor. Now VLAD descriptor will be represented with $v_{i,j}$, where the indices $i = 1 \dots k$ and $j = 1 \dots d$ respectively index the visual word and the local descriptor component. As a result of these assumption, we will acquire a component of v as a sum over all the image descriptors:

$$v_{i,j} = \sum_{x\, such\, that\, NN(x)=c_i} (\, x_j - c_{i,j}\,)\,, \tag{2.1}$$

where x_j denotes the j_{th} component of the descriptor x and $c_{i,j}$ denotes the same component in its corresponding visual word c_i. Then this v vector is L2 -normalized by $v := v/||v||_2$ as a post-process. The VLAD representations associated with a few images are visualized in Figure 2.2, that aggregates 128-dimensional SIFT descriptors. The components of given descriptors are depicted like a SIFT descriptors, in other words each 16 components in $v_i = 1..k$ corresponds to 4 x 4 spatial grid representation of oriented gradients. In this sample, descriptors are accumulated in 16 of them, one per visual word. Note that, the subtraction operation in Equation 2.1 makes a component may be negative differently from a SIFT descriptor. These negative components are depicted with red on this figure. Next in terms of 2_{nd} and 3_{rd} steps, they solved the dilemma between the dimensionality reduction and the indexing by optimizing these phases jointly. Finally, they also demonstrated the superiority of VLAD against standard BOF for the same size.

The DenseVLAD descriptor (Torii et al., 2015) is an another example for a state-of-the-art VL algorithm which is extended on VLAD. Images are represented by densely sampled VLAD vector (Arandjelovic and Zisserman, 2013; Jégou et al., 2010; Kendall and Cipolla, 2017), resulting in a more compact representation of reference images. The DenseVLAD descriptor is also based on SIFT descriptor which actually aggregates RootSIFT (Arandjelović and Zisserman, 2012) descriptors densely sampled

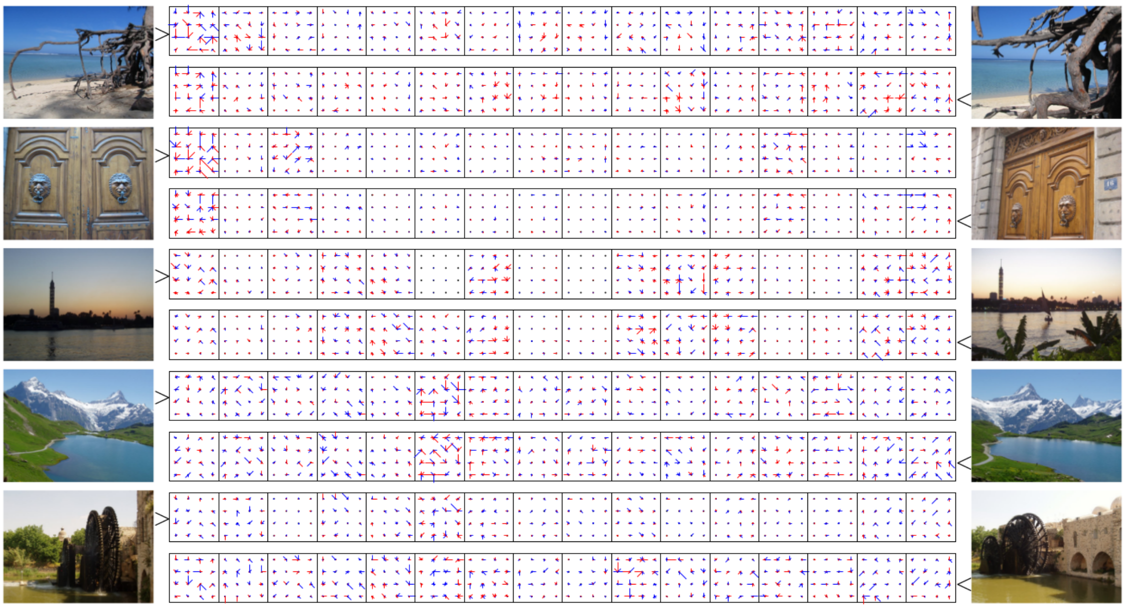


Figure 2.2. Images and corresponding VLAD descriptors, for k=16 centroids result in compact vector D=16x128. The components of the descriptor are represented like SIFT, with negative components in red which caused from the Equation 2.1 (Jégou et al., 2010).

on a regular grid in each image. By the way, contribution of using DenseVLAD in feature detection phase of image retrieval is showed up, especially in the presence of strong illumination changes (Torii et al., 2015).

Recently, we have witnessed the tremendous growing impact of deep learning techniques for several research areas. As a reflection of this fact, several studies proposed types of CNN based trainable descriptors for their VL applications. Trade-off between usage of global and local features are already mentioned in the previous parts, Sünderhauf et al. (2015) overcomes this dilemma by using the Edge Boxes object proposal method (Zitnick and Dollár, 2014) combined with a mid-level convolutional neural network (CNN) feature (Krizhevsky et al., 2012) to identify and extract landmarks. These proposed methods serve us an ability of just looking into valuable parts of image. Furthermore in their latter works, they (Sünderhauf et al., 2015) compared the success of VL with respect to trainable descriptors which were extracted from the different layers of a CNN network. Robustness of different layers in VL was especially considered against the wide viewpoint changes. On the other hand there are studies in which task oriented trainable descriptors are directly described and these are generally trained with *Triplet Loss* (Schroff et al., 2015). A typical *Triplet Loss* minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. Also, many works benefit from same *Triplet Loss* training procedure for VL task into *2D-3D* matching space. Radenović et al. (2016) proposed to fine-tune CNN for image retrieval from a large collection of unordered images in a fully automated manner. They provide state-of-the-art retrieval and Structure-from-Motion (SfM) methods to obtain 3D models, which are used to guide the selection of the training data for CNN fine-tuning. They demonstrated that object retrieval performance will be enhanced with contribution of both hard positive and hard negative examples. In another study (Radenović et al., 2018), a novel trainable Generalized-Mean (GeM) pooling layer was proposed. This layer generalizes max and average pooling and show that it boosts retrieval performance on a well-known benchmarks: Oxford Buildings, Paris, and Holidays data sets. Differently from VL task Radenovic et al. (2018) introduced shape matching as metric learning with *Triplet Loss* based convolutional networks. On the other hand, one and foremost used method into *2D-2D* matching space, that also benefits from same *Triplet Loss* for VL task, is NetVLAD (Arandjelovic et al., 2016). NetVLAD representations uses a CNN to learn the descriptors that are aggregated into a VLAD descriptor. They introduced the following three principal contributions. Initially, a CNN architecture is developed that is trainable in an end-to-end manner directly for the VL task. Second, they developed training procedure in order to learn parameters of the architecture in an end-to-end manner from images which representing the same places in different time. They achieved this learning owing to their new weakly supervised ranking loss. Finally, they have figured out the improving place recognition performance over DenseVLAD and other compact image descriptors on two challenging VL benchmarks. All these reasons make NetVLAD the most preferred LD based descriptor as a baseline VL method, thanks to the its grand success against changing environmental conditions.

2.2. Descriptor Matching Approaches

Performing an efficient image comparison emerges as one of the most important steps of a characteristic VL system. With respect to this importance, different types of ANNS methods (Andoni and Indyk, 2006; Beis and Lowe, 1997; Bentley, 1975) are firstly introduced in order to handle large databases in computer vision applications. These methods look for the approximate nearest neighbors instead of exact nearest neighbors with different types of techniques. One of the frequently used techniques is Euclidean Locality-Sensitive Hashing (Datar et al., 2004), which has been extended in (Kulis and Grauman, 2009) to arbitrary metrics. But, these approaches are memory consuming, as several hash tables are required. In order to cope with this inefficiency, Weiss et al. (2009) tried to satisfy the memory constraint by embedding the vector into a binary space. However from the aspect of accuracy and memory trade-off, their method becomes unsuccessful against product quantization-based approximate search method which is proposed by Jegou et al. (2010). Moreover, superiority of using *k*-dimensional (*k-d*) trees in ANNS approach is highlighted in numerous works and its variants (Jo et al., 2017; Silpa-Anan and Hartley, 2008) have been proposed for different computer vision task. In order to cope with high dimensional feature matching in more efficient way, Muja and Lowe (2009b, 2014) randomized these *k-d* tress and named as *multiple randomized k-d* tress. In addition, they also mapped their new method into a compact tool that is called fast library for ANNS (FLANN (Muja and Lowe, 2009a)).

In this study, all the performed LD and SD based VL methods were constructed on FLANN to retrieve the most similar database image for a given query. Employing FLANN in feature matching needs collection of same dimensional descriptors for both database (prior map) and query images individually, this concatenated collection will be defined in a matrix form like given below:

- $\mathbf{k}$: is the number of nearest neighbors to be returned for a given descriptor.
- $\mathbf{d}$ x $\mathbf{n}$: is a descriptor matrix of database images which contains *n* number of *d*-dimensional descriptors, stored in a column-major order.
- $\mathbf{d}$ x $\mathbf{m}$: is a descriptor matrix of query images which contains *m* number of query

descriptors whose *k*-nearest-neighbors need to be found.

- $\mathbf{k}$ x $\mathbf{m_{ngh}}$: is a result matrix that contains indexes (among *d* x *n*) of the returned *k* number of nearest neighbors for a given *m* number of query descriptors.
- $\mathbf{k}$ x $\mathbf{m_{dist}}$: is a distance matrix that contains euclidean distances (L2 Norm) between returned *k* number of nearest neighbors and their corresponding query descriptors.

First of all, FLANN builds a powerful index on our database descriptors '*d* x *n*' by means of *multiple randomized k-d* tress. Then performs ANNS for each of given query descriptor in '*d* x *m*' by using this already created index. Finally, FLANN returns these two matrices k x m_{ngh} and k x m_{dist} which are in the same size. The first one gives returned *k* nearest candidates for a given *m* number of query descriptors, which was used for performing the whole image matching phases in this study. Second one provides us the euclidean distance (L2 Norm: square root of summed squared difference) between these returned candidates and their corresponding query images. As a result, the k x m_{ngh} matrix is employed for image matching phase of proposed methods, on the other side k x m_{dist} is especially used in generation of the Hybrid-VL$_{DL}$ method as described in Section 3.5.1. Also while practicing the FLANN in this study, a valuable result was deducted: the accuracy of FLANN will change with respect to applied number of randomized trees. This fact is approved with comparing standard euclidean distance computation. Therefore in the implementation of proposed method high numbers of randomized trees parameter for high dimensional SD and LD descriptor is operated.

Also note that, all these previous studies mentioned above were operated on perspective images taken with monocular cameras. As a unusual perspective, different type of omnidirectional cameras are also preferred (Goedemé et al., 2004, 2007; Lhuillier, 2005, 2007; Murillo et al., 2010; Singh and Kosecka, 2010) for image matching in order to take advantage of their wide field of viewpoint. Although, using these type of cameras will increase the effectiveness of comparing phase, conversely it causes a extra computational cost. Because of this extra effort, which is spent for determining the suitable slice in an omnidirectional image, recent efficiency oriented VL approaches (Majdik et al., 2015; Schroth et al., 2011) would prefer using ANNS with perspective images as being in this work.

2.3. Benchmark Data Sets

Although there are some VL works (Choi et al., 2015; Fraundorfer et al., 2007; Furnari et al., 2016; Goedemé et al., 2004, 2007) performed in indoor environment, majority of these researches concentrate on outdoor environments especially for urban or sub-urban areas. In contrast to indoor environment studies, outdoor environment studies are in need of well organized data sets with many requirements. Moreover, a new outdoor VL method must be evaluated against changing lon-short term conditions. In line with this necessities, challenging outdoor street-level-driving data sets (Table 2.1) for VL task are described in the following parts.

Table 2.1. Comparison of benchmark VL Data sets for normal field of view cameras.

Data set	Setting	Image Capture	# Images		Condition Changes		
			Database	Query	Weather	Seasons	Day-Night
RobotCar(Maddern et al., 2017)	Urban	Trajectory/short baseline	20 billion		✓	✓	✓
Malaga Downtown(Blanco-Claraco et al., 2014)	Urban	Trajectory/no baseline	62886				
Aachen Day-Night(Sattler et al., 2018)	Historic City	Free Viewpoint (mobile)	4328	922			✓
CMU Seasons(Sattler et al., 2018)	Suburban	Trajectory/short baseline	7159	75335	✓	✓	
RobotCar Seasons(Sattler et al., 2018)	Urban	Trajectory/short baseline	6954(rear)	3978(rear)	✓	✓	✓
Malaga Streetview Challenge(ours)	Urban	Trajectory/wide baseline	1571(rear)	436(rear-streetview)	✓	✓	

Maddern et al. (2017) proposed a challenging new data set named as Oxford RobotCar Data set which represent a driving scenario. They drove on the same path through central Oxford in different time period from 2014 to 2015. By this way, 20 million images were collected from 6 cameras mounted to the vehicle as well as with LIDAR, GPS and INS ground truth. Images were collected in all weather conditions, containing heavy rain, night, direct sunlight and snow. During the collection period, there are also many long-term changes on the roads and buildings occurred. As a result, they managed to investigating long-term localization and mapping for autonomous vehicles in real-world thanks to their frequent traversing. Also their full data set is available at: *http://robotcar-dataset.robots.ox.ac.uk*.

Carlevaris-Bianco et al. (2016) mentioned that, previous work on VL has generally focused on perspective images, in their paper large scale and long-term autonomy data set for robotic researches was collected on the University of Michigan's North Campus with omnidirectional cameras and named as North Campus Long-Term (NCLT) data set. Additional to omnidirectional imagery, also 3D lidar, planar lidar, GPS information is available thanks to their fully equipped robot. The data set was collected to facilitate research focusing on long-term autonomous operation in changing environments. They captured 27 traversals both for indoors and outdoors which is captured in different conditions (times of a day) among 15 months. Therefor thanks to this configuration, many challenging elements including: moving obstacles, changing lighting, varying viewpoint,seasonal and weather changes, and long-term structural changes could be captured with this data set.

Blanco-Claraco et al. (2014) pointed on the lack of publicly accessible data sets with a reliable ground truth and the difficulty of creating Malaga Urban data set. For this aim, they addressed both the practical and theoretical issues found while building a collection of six outdoor data sets. Among these six collections *Malaga Downtown* is the most suitable one with its nearly 8km route for our work. This sub set was collected with kinds of sensors, including laser scanners and one stereo camera (Bumblebee2) in urban driving scenarios. One distinctive feature of the present data set is the existence of high-resolution stereo images [1024 X 768] grabbed at high rate (20fps) during driving, turning the data set into a suitable benchmark for VL techniques. Both plain text and binary files are provided which include corresponding GPS data.

Sattler et al. (2018) considered the problem of inconsistency between the previously created VL data sets. Under the light of this deficiency, they generated the sub version of frequently used driving data sets in context of changing environmental conditions. Also they underlined that, a practical visual localization approaches need to be robust to a wide variety of viewing condition, including day-night changes, as well as weather and seasonal variations, while providing highly accurate 6 degree-of-freedom (6DOF) camera pose estimates. Under the light of this aim, they introduced the first benchmark data sets specifically designed for analyzing the impact of such factors on visual localization. They carefully created ground truth poses for query images taken under a wide variety of conditions. Next on, the impact of various factors on 6DOF camera pose estimation accuracy through extensive experiments was evaluated with state-of-the-art localization approaches. This data set contains high resolution [1024X1024] images recorded with three synchronized global shutter Point Grey Grasshopper2 cameras mounted to the left, rear, and right (triplet) from an autonomous vehicle platform over 12 months in Oxford, UK. Each traversals were recorded on the same 10km route, among these provided 10 traversals one of them was selected as reference traversal in overcast conditions and rest of them were used as query traversals that cover a wide range of conditions (Table 3.1). For each sub set highly accurate 6 degree-of-freedom (6DOF) camera pose estimation and GPS data are also provided. Also they introduced another two data sets present different challenges. The Aachen Day-Night data set focuses on localizing night-time photos against a 3D model built from day-time imagery and images are taken with hand-held cameras with wide viewpoints changes. CMU Seasons data sets represent automotive scenarios, with images captured from a car. This data set exhibits less variability in viewpoints but a larger variance in viewing conditions like RobotCar Seasons data set. In contrast, the CMU data set is collected in sub-urban areas that contains a significant amount of vegetation. As a result they showed that long-term localization is far from solved, and proposed promising avenues or future work, including sequence-based localization approaches and the need for better local features. Their benchmark is available at *visuallocalization.net* (Sattler et al., 2018).

Based upon the detailed comparison on literature, Table 2.1 demonstrates that *RobotCar Seasons* (Sattler et al., 2018) data set is the most suitable one for this study. Because this data set especially design for VL challenge derived from more dense Oxford RobotCar (Maddern et al., 2017) data set. And differently from the other examined data sets, it becomes pointed with its 'Setting: Urban' and 'Condition Changes: All' properties as given in this table. Also recent works support this preference, state of the art VL approaches (Germain et al., 2018; Piasco et al., 2019; Seymour et al., 2018) prove their success on *RobotCar Seasons* data set. Furthermore the necessity of data set which simultaneously provides us 'Condition Changes: all', 'Image Capture: Trajectory/wide baseline' and 'Setting: Urban' properties is figured out in this same table. Therefore in addition to *RobotCar Seasons* data set, we generate a Google Streetview based *Malaga Streetview Challenge* data set which supplies all these underlined necessity simultaneously. This new data set was derived from Malaga Downtown (Blanco-Claraco et al., 2014) and its generation phase is explained in Section 3.1.1.

Consequently, all proposed methods in this study are examined on this two data set *RobotCar Seasons* and *Malaga Streetview Challenge*. Moreover many other street-level driving data sets are proposed for different tasks like image segmentation. Hereby, Camvid (Cambridge Labeled Objects in Video) data set (Fauqueur et al., 2007) stands out as a very sufficient and frequently used segmentation data set among the others (Cordts et al., 2016; Gaidon et al., 2016) that helps us while training our semantic segmentation model in this study.

2.4. Semantic Representation

Currently, we see that deep learning methods surpass traditional approaches in many tasks of computer vision and natural language processing in terms of accuracy and sometimes even efficiency (Tekir and Bastanlar, 2020). Semantic segmentation is also one of these computer vision tasks. As a short definition, these semantic segmentation methods transformed those existing and up-to-date classification models – AlexNet (Krizhevsky et al., 2012), VGG (16-layer net) (Simonyan and Zisserman, 2014), GoogLeNet (Szegedy et al., 2015),and ResNet (He et al., 2016) – into fully convolutional networks (FCN) by replacing the fully connected layers with convolutional ones in order to output spatial maps instead of classification scores. Then dense per-pixel labeled outputs are produced by upsampling those maps. This upsampling process also named as deconvolutions (Zeiler and Fergus, 2014; Zeiler et al., 2011). A typical semantic segmentation Convolutional Neural Networks (CNN) consists of encoder and decoder phases. SegNet (Badrinarayanan et al., 2017) specializes with its decoder phase. In detail, decoder stage of SegNet is constructed on a set of upsampling and convolution layers. Pixel-wise labeled image is produced using softmax classifier as an output which has the same resolution as the input image. Note that each upsampling layer in the decoder stage corresponds to a max-pooling one in the encoder part. Those layers upsample feature maps using the max-pooling indices from their corresponding feature maps in the encoder phase. After hat, dense feature maps are produced by convolving the upsampled maps with a set of trainable filter banks. Feature maps are fed to the softmax classifier to produce the final segmentation when they have been restored to the original resolution. As a different CNN based segmentation approach, the DeepLab models (Chen et al., 2014, 2017, 2019) uses a fully connected CNN structure which is firstly introduced by Krähenbühl and Koltun (2011, 2013). They refine the segmentation result by using this fully connected structure which is separated in post-processing step. As a first step they models each pixel as a node in the field, then one pairwise term is employed for each pair of pixels no matter how far they lie. In shortly, they renders the system able to recover detailed structures in the segmentation that were lost due to the spatial invariance of the CNN by considering the both short and long-range interactions. In addition, its recent version of *DeepLabv3+* (Chen et al., 2018) is also available.

Besides all these semantic segmentation works, there are studies in which semantic information in an image is benefited in order to improve localization performance differently from this thesis. Naseer et al. (2014) claimed that existing methods generally leverage feature descriptions of whole images or image regions from Deep CNNs. Also they noted that, there are many other studies that benefit from sequential information in order to struggle the problem of spatially inconsistent and non-perfect image matching. Differently from this studies, they achieved to learn a discriminative holistic image representation which uses the image content to create a dense and salient scene description. These salient descriptions are learnt over a variety of data sets under large perceptual changes. Thanks to the their method, segmented regions of an image are able to captured exactly which are also geometrically stable over large time lags. Then, they combined features from these salient regions and an off-the-shelf holistic representation to form a more robust scene descriptor.

Seymour et al. (2018) motivated with that specific scene regions remain stable in the semantic modality even in the presence of vast differences in the appearance modality. In their study, they developed a deep learning based method for fusing appearance and semantic information using visual attention for 2D image- based localization (2D-VL) across extreme changes in viewing conditions. They operated this fusion in descriptor level and their proposed attention-based module learns to focus not only on discriminative visual regions for place recognition.

Mousavian and Kosecka (2016) drew attention to that, density of the database images and the robustness of the image representation will directly influence the success of VL more than the conditional changes. In their work, first a sparse set of geo-tagged reference views are generated for a map, then they determined camera location and orientation using this map and reference views. By this way they achieved to localize a novel view geographically. In this line, they proposed a novel technique for detection and identification of building facades from geo-tagged reference view using the map and geometry of the building facades which is obtained from semantically segmented images. As a last step, they put together the information comes from detected landmark (building) identities from reference views, 2D map of the environment, and geometry of building facades in order to compute the likelihood of camera location and orientation of the query images.

Ondruska et al. (2016) incorporated the semantic segmentation to city-scale tracking task with a different kind of Neural Network. In their work, they presented a recurrent neural network based framework in order to classify and track the hardly observable real-world surrounding of a robot. Their end-to-end trainable model manages to filter an input stream of laser measurements in order to directly determine object locations. In short, they provided a new tracking and semantic classification method owing to their trainable RCNN architecture.

Stenborg et al. (2018) considered the problem of logn-term visual localization in the context of *2D-3D* matching space. They managed to label an environment semantically with its all corresponding point by means of semantically segmented images. Then they demonstrated that a vehicle localization without the need for detailed feature descriptors (SIFT, SURF, etc.) will be achieved by efficiently usage of labeled 3D point maps. In this way, instead of depending on hand-crafted feature descriptors, they discussed on the training of an image segmenter.

Singh and Košecká (2012) produced a hand-crafted descriptor after semantic segmentation of images, and they used these descriptors for grouping the scenes such as on a street, in front of a building or at a crossroads. In another study, Mousavian et al.

(2015) in their LD based study, they eliminated local descriptors of objects (such as tree) that do not come from man-made structures by using extracted semantic labels. Thus, they increased the wights of features coming from man-made structures compared to non man-made structures since natural structures have low chance of healthy matching. Again in an example of semantic clues based *2D-3D* matching study [8], accuracy of image matching was also checked semantically. In an another area in which Semantic knowledge gained popularity is studies that using *3D* scene reconstruction, Schönberger et al. (2018) prepared a dictionary for semantic content and expressed the scene as bag of features (BOF) for this reconstructed scene.

CHAPTER 3

IMPLEMENTATION OF THE PROPOSED METHOD

In this chapter, methodology of operated approaches is explained. The reasons behind putting these methods to use in this study had already been given in Section 2 with a broad comparison. Hereby, after defining working scheme of this proposed method, the prepared data set variants and implementation details of proposed methods are explained in the following sections.

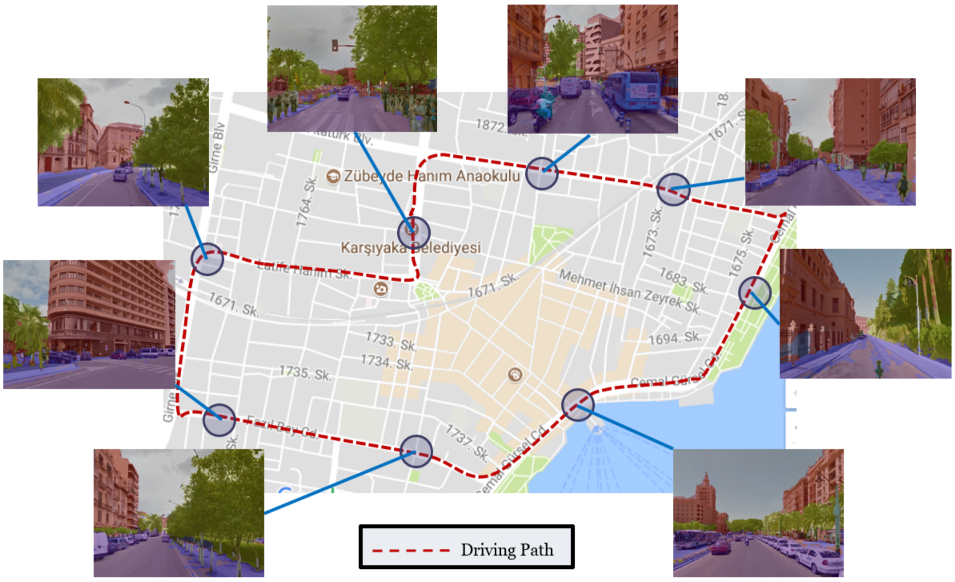
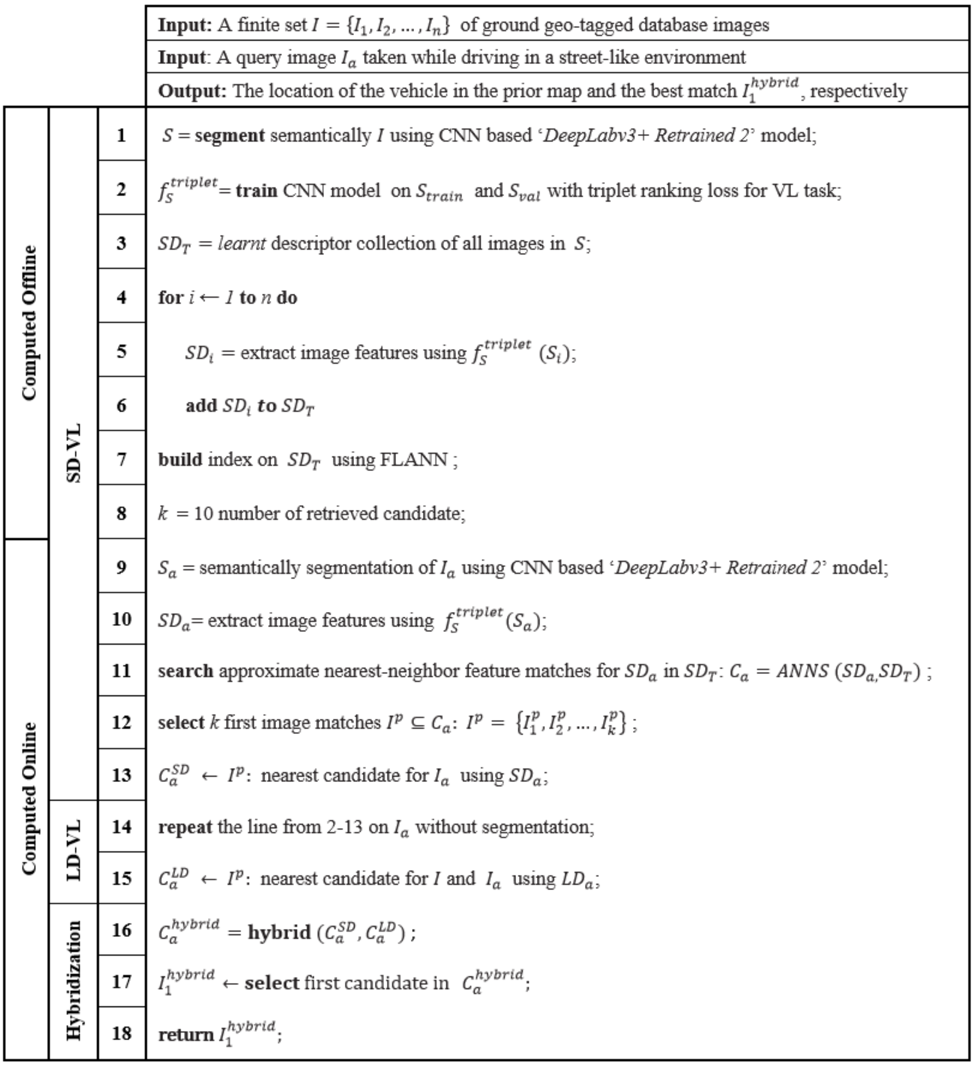


Figure 3.1. Proposed semantic content based VL.

Before giving the details of the proposed methodology, semantically aided VL approach is demonstrated in Figure 3.1 which is also based on image retrieval technique as previously depicted in Figure 2.1. As it is supposed to be in typical VL studies, our prior map (red dotted lines) corresponds to reference traversal of the given data sets, while images of other traversals collected in changing conditions on the same path are accepted as our query images. Differently from the previous studies, we directly used the power of semantically segmented images in order to improve performance of any state-of-the-art VL method as depicted in this figure. It also should be noted that, this study will be defined as an implementation of topometric localization, which combines the robustness of topological localization (roughly localization of the nodes in a graph) with the geometric accuracy of metric. Therefore localization precision of proposed method is evaluated in meter threshold as being in state-of-the art VL approaches, which is also described in Section 4.1.

Proposed Hybrid-VL$_{DL}$ method can be summarized step by step with the given pseudocode of the proposed Hybrid-VL$_{DL}$ method in Figure 3.2 which is constructed on semantic content based image retrieval in 2D-2D matching space. This reductionist representation of proposed VL method also provides us which step corresponds to which key components of a characteristic image retrieval based localization system (image representation, image matching, hybridization). In addition, we able to show not only where the novel parts of this study takes place with their corresponding steps, but also how (offline-online) these parts are operated.

In this paragraph, the proposed algorithm to match input images with geo-tagged ones is introduced. Firstly note that the algorithm from line 1 to 8 can and should be computed offline, regarding to an actual driving mission. In this representation, proposed *learnt* SD-VL takes a query image I_a and return *k* number of candidates C_a^{SD} from database images in lines from 1 to 13. In the first line previously trained semantic segmentation method '*DeepLabV3+ Retrained-2*' is employed on database images I that gives us their segmented versions S. In line 2, a CNN model is trained on S with triplet ranking loss for VL task, then the part from line 3 to 6 corresponds to *learnt* semantic descriptor SD_i extraction process. Robust indexing is built up in line 7 with our ANNS method FLANN for database image descriptors collection SD_T. Next, these same steps are operated for a semantically segmented query image S_a in line 9 and 10. From line 11 to 13, ANNS is conducted and k number of best matching images retrieved. Moreover, the same steps for SD-VL (2-13) is repeated without segmentation in order to obtain $learnt$ LD-VL in line 14 and 15, so that we obtain best matching k candidates C_a^{LD} for our LD-VL method. Finally, effective decision-level hybridization methods is represented from line 16 to 18, that incorporates C_a^{LD} and C_a^{SD} in post-processing level. As a result, among the k number of Hybrid-VL candidates C_a^{hybrid} the top first one



			Input: A finite set $I = \{I_1, I_2, \ldots, I_n\}$ of ground geo-tagged database images
			Input: A query image I_a taken while driving in a street-like environment
			Output: The location of the vehicle in the prior map and the best match I_1^{hybrid}, respectively
Computed Offline	**SD-VL**	**1**	S = **segment** semantically I using CNN based '*DeepLabv3+ Retrained 2*' model;
		2	$f_S^{triplet}$= **train** CNN model on S_{train} and S_{val} with triplet ranking loss for VL task;
		3	SD_T = *learnt* descriptor collection of all images in S;
		4	**for** $i \leftarrow 1$ **to** n **do**
		5	SD_i = extract image features using $f_S^{triplet}$ (S_i);
		6	**add** SD_i **to** SD_T
		7	**build** index on SD_T using FLANN ;
		8	k = 10 number of retrieved candidate;
Computed Online		**9**	S_a = semantically segmentation of I_a using CNN based '*DeepLabv3+ Retrained 2*' model;
		10	SD_a= extract image features using $f_S^{triplet}(S_a)$;
		11	**search** approximate nearest-neighbor feature matches for SD_a in SD_T: $C_a = ANNS\ (SD_a, SD_T)$;
		12	**select** k first image matches $I^p \subseteq C_a$: $I^p = \{I_1^p, I_2^p, \ldots, I_k^p\}$;
		13	$C_a^{SD} \leftarrow I^p$: nearest candidate for I_a using SD_a;
	LD-VL	**14**	**repeat** the line from 2-13 on I_a without segmentation;
		15	$C_a^{LD} \leftarrow I^p$: nearest candidate for I and I_a using LD_a;
	Hybridization	**16**	C_a^{hybrid} = **hybrid** (C_a^{SD}, C_a^{LD}) ;
		17	$I_1^{hybrid} \leftarrow$ **select** first candidate in C_a^{hybrid};
		18	**return** I_1^{hybrid};

Figure 3.2. Proposed algorithm of semantic content based Hybrid-VL$_{DL}$.

I_1^{hybrid} is returned against the given query image I_a.

In the following sections, we give further explanation about the proposed algorithm with detailed implementation of each steps.

3.1. Data sets and Variants

In this section, the preferred *Malaga Downtown* and *RobotCar Seasons* data sets, used in our experiments, are presented. These benchmark VL data sets and their comparison with the others are already described in Chapter 2. Therefore, detailed usage and variants of these data sets are defined in the following paragraphs.

Malaga Urban data set (Blanco-Claraco et al., 2014) provides us number of sub sections individually for the convenience of usage. Among these sub sections, the most suitable one ‘Malaga Downtown’ (Figure 3.3) with nearly 8km trajectory is chosen with respect to the topic of this study. As it is mentioned below there are 62886 stereo images, however we just turned to account one of them (left or right) that both looks at the same front direction. This preference cuts in half with a 31443 number of perspective images. It is clear that, because of its high rate (20fps) image capturing Malaga Downtown needs to be more sparse in order to be suitable for VL task as described in Section 4.2. On behalf of this aim, every sequential 20th images were brought together which provides us nearly 5 meter distance between sequential images. This decreased sub set was named as *Malaga Downtown Base* data set that contains 1571 images.

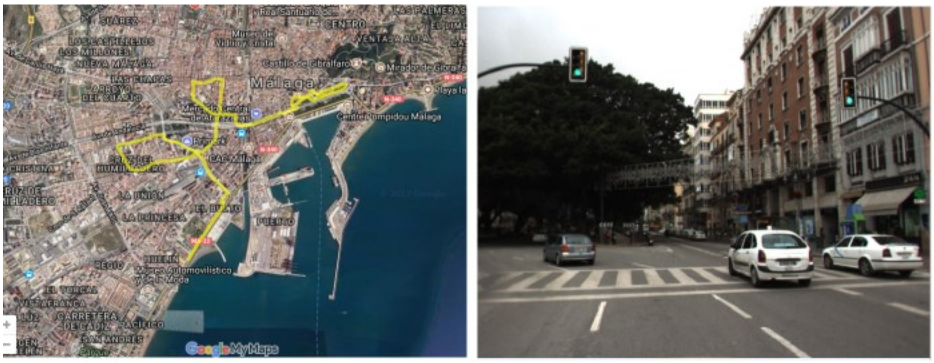


Figure 3.3. *Malaga Downtown Base* data set trajectory (left) and a sample image of this data set(right).

RobotCar Seasons data set was introduced (Sattler et al., 2018) in 2018 as the first benchmark data set specifically designed for analyzing the impact of weather and seasonal changes on VL as underlined in Chapter 2. In other words, in contrast to the Malaga Downtown data set, this data set exhibits less variability in viewpoints (trajectory) but a larger variance in viewing conditions for a city-scale urban driving scenario (Figure 3.4). This data set especially design for VL task with reduction of Oxford RobotCar data set (Maddern et al., 2017). As it is obviously seen in Table 2.1 from Section 2.3, its properties makes this data set mostly matching one with expectations of this study. This data set contains 10 different traversals that each of them were recorded on the same 10km route. For each sub set highly accurate 6 degree-of-freedom (6DOF) camera pose estimation and GPS data are also provided. Among these provided 10 traversals one of them was selected as reference traversal (database image) in overcast conditions and rest of them were used as query traversals that cover a wide range of conditions as given in Table 3.1. Also, number of images for each traversals and their changing conditions are given it this table. In this study, proposed methods were examined on these query traversals (especially concentrated on Overcast Winter) versus to same single overcast reference traversal of this data set. Therefore it can be said, practiced RobotCar Seasons data set contains 6954 database images with changing query images from 400-500 (Table 3.1) according to used query traversal.

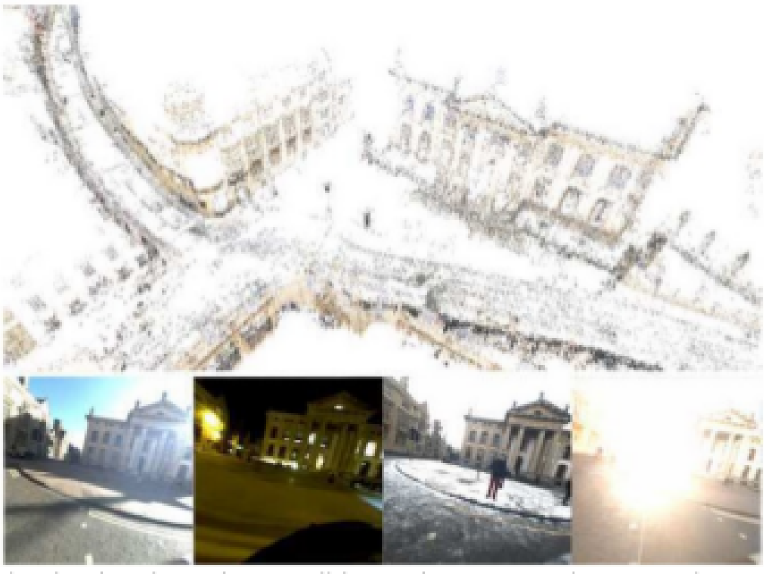


Figure 3.4. Sample RobotCar Seasons query images which represent different conditional changes especially in weather with changing seasons(bottom) (Sattler et al., 2018).

It should be claimed, in this study, despite images with different viewing angle (45° left or right) are also supplied with RobotCar Seasons data sets. Experiments including left and right with the rear ones are also carried out, however these extra images did not improve the success of proposed method. Moreover, thanks to using just rear image preference, proposed VL method becomes less time consuming from the both online and offline computation phases as depicted in the proposed algorithm (Figure 3.2).

Furthermore, adapting the triplet loss for localization aware descriptors requires to divide our driving path (prior map) for both of the data sets into 3 parts (training-validation-test) as it is described in Section 3.4 detailed. And this division will directly affects the performance of SD and LD VL methods. Therefore at least 2 different variants of used Malaga Urban and RobotCar Seasons data sets which were generated by separating the same path (images) into different 3 parts were examined with proposed methods.

3.1.1. Newly Generated Data Set: Malaga Streetview Challenge

Differently from RobotCar Seasons data set, there is no available traversals which is collected in changing conditions for Malaga Downtown data set as it is depicted in Table 2.1. Starting from this deficiency, we generated a new test traversal on the same path of Malaga Downtown (Figure 3.3) by means of *Google Streetviews* (Orlita, 2016). This newly generated set will be called *Malaga Streetview* test set, supplies us another challenging condition: 'wide baseline' as given in Table 3.1. We know that RobotCar Seasons traversal collected with nearly same trajectory with almost unchanging viewpoint in different times, on the other side *Malaga Streetview* test set provides us wide viewpoint changing. This challenging fact is derived from the limited *Google Streetview* images that captures the corresponding scene. If we explain more in detail, sometimes necessary query image is obtained from the other lane (with the opposite direction of the driving with wide changing in viewpoint) of the road forcefully, because there is no corresponding image in this zone of current lane. In this way, 436 street view query images is obtained with changing frequency from 10 meter to 20 meter in the same 8km route. Also corresponding GPS coordinates are acquired from the already given GPS data of Malaga Downtown, in line with this objective GPS coordinates of the two nearest images from the Malaga Downtown are carefully interpolated. Moreover, though *Malaga Streetview* test set is formed with one pass through the same path, it provides all long-short term environmental changes at once. This changing arises from this fact, Malaga Downtown data set is generated during restricted time period of a day in 2014, nevertheless sequential images in *Google Streetview* are taken from different time period of different years (2014-2020) that supplies us natural diversity in short-long terms changes. Both these wide-viewpoint and long-short term environmental changes are demonstrated in Figure 3.5. Moreover, this new set reflects a real life case, like a person who is driving towards same path, this realistic scenario is achieved owing to collecting the images which is mostly coherent with original ones from Malaga Downtown data set and capturing them with fixed pan-tilt-zoom value. Also manually segmented version of this new data set is generated as described in Section 3.2.

This newly generated *Malaga Streetview* test set (436) used as a query images while *Malaga Downtown Base* set (1571) is accepted as database images (prior map). So that, this new integrated data set totally includes 2007 query and database images. Then, experiments related with Malaga Downtown in the next Chapter were mainly carried out on this new challenging data set named as *Malaga Streetview Challenge*. Detailed statistic and comparison with RobotCar Seasons data set are giving in Table 3.1. In addition, this newly generated test data set will be useful to the community of VL area.

Table 3.1. Detailed statistics for the two benchmark data sets used in this study.

Data set	**Image Capture**	**database images conditions (# images)**	**query images conditions (# images)**
RobotCar Seasons(Sattler et al., 2018)	Short baseline	Overcast-November (6954)	dawn (483), dusk (394), night (483), night+rain (440), rain (421), overcast summer /winter (463/390), snow (489), sun (460)
Malaga Streetview Challenge(ours)	Wide baseline	Overcast-September (1571)	Google Streetview (436): all short-long term changes by different time period and years from 2014 to 2020

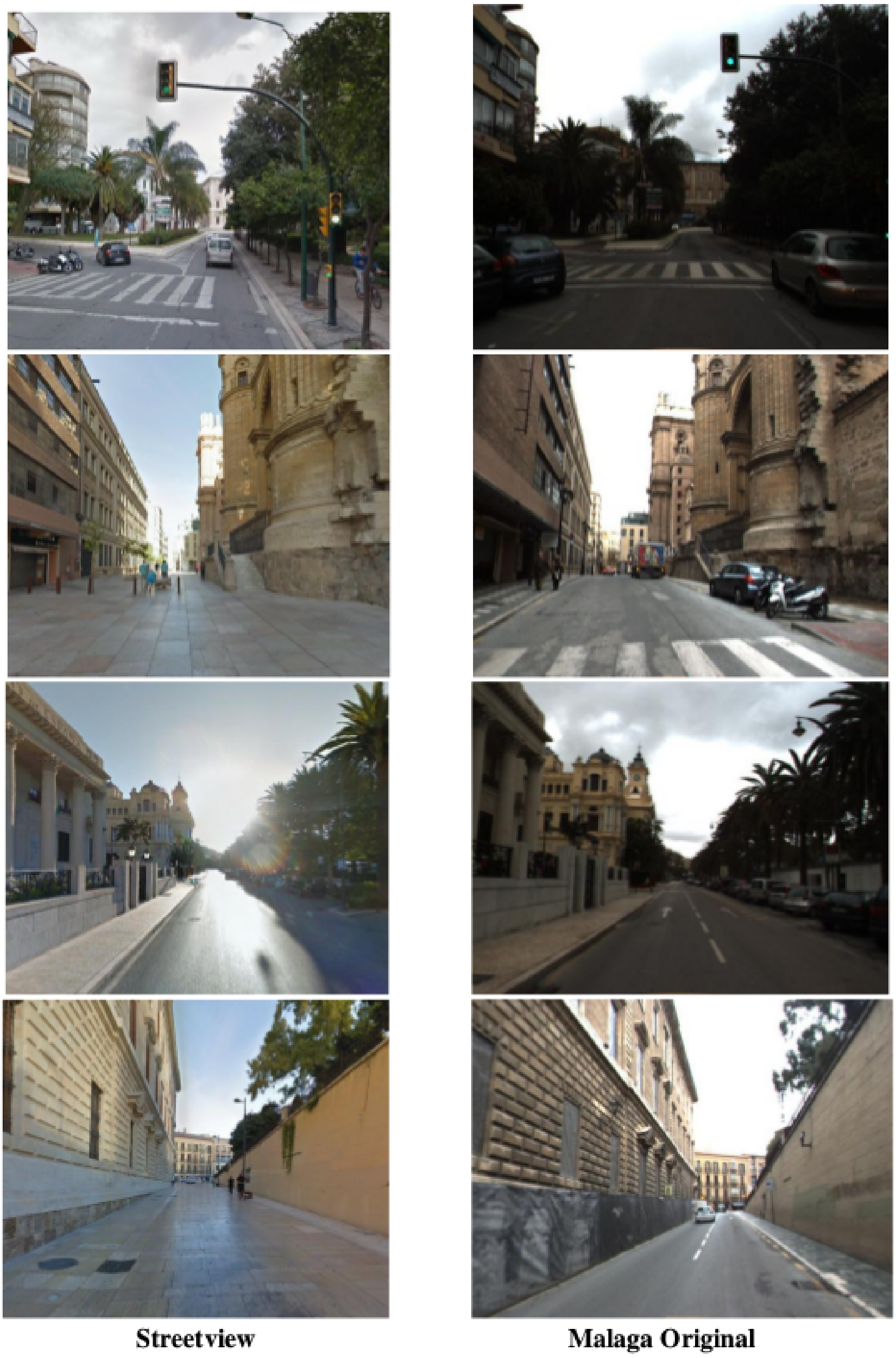


Figure 3.5. Changes in wide viewpoint and short/long term conditions (weather, lighting / new buildings etc.) between *Malaga Streetview* (left) and corresponding *Malaga Downtown Base* (right) images is compared.

3.2. Semantic Segmentation Methodology

As mentioned in previous chapters main contribution of this thesis was achieved by using the semantic information in a scene. This semantic information is acquired from semantically segmented images. Different kinds of semantic segmentation approaches (such as manually annotating and automatically annotating) were examined in this study. Implementation details of these approaches are described in this section.

In the initial stage of this work, our semantic segmentation was carried out with a manually annotating tool. Among the all usable pixel wise labeling tool just LIBLABEL (Geiger et al., 2013) makes polygonal annotation. It is a Matlab tool for annotating images with polygons. We can easily obtain semantic label map of a given image, this tool let us add and remove polygons around objects. By this way we managed to segment our image according to specific semantic classes with their respective colors as illustrated in Figure 3.6. In this figure, an image from *Malaga Downtown Base* is depicted with its annotated and segmented versions.

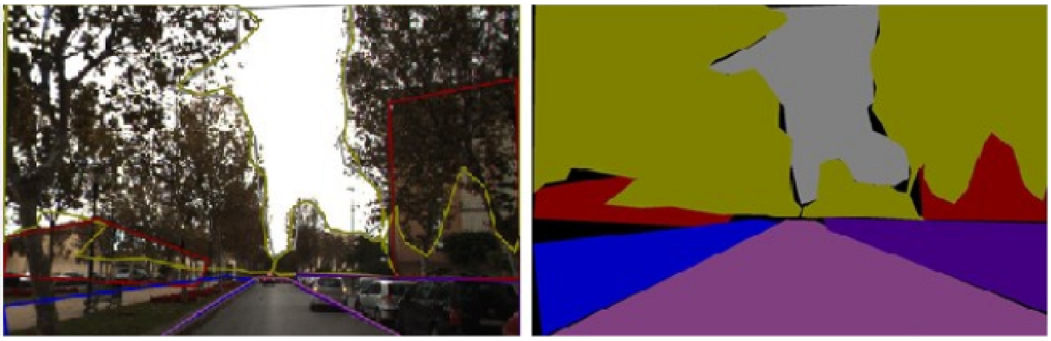


Figure 3.6. An image from *Malaga Streetview Challenge* data set and its semantically segmented version with LIBLABEL.

That is obviously seen in this figure, we just needed to label large area and stationary object because its known tiny objects do not affect semantic information so much, on the other hand labeling moving objects makes the semantic information vulnerable against short term changes in a city-scale driving scenario. Under the light of these labeling rules, I configured LIBLABEL tool and only labeled images in *Malaga Streetview Challenge* data set according to 7 object classes: Building, Car, Road, Sidewalk, Sky, Tree, Wall. On behalf of labeling whole *Malaga Streetview Challenge* data set images (2007), first polygon labels had been prepared for each image in ".mat" format then they were converted to segmented images. Also a matrix form of each image was created which just includes the object class ID instead of labeled color values. Moreover this provided semantically annotated data will be useful to the community of VL area who especially works on semantic segmentation subject.

This manual segmentation process is very time consuming together with its inconsistent annotation performance. Therefore, a CNN based semantic segmentation framework SegNet: a deep convolutional encoder-decoder architecture for robust semantic multi-class pixel-wise segmentation (Badrinarayanan et al., 2017) which is explained in Section 2.4 was employed in order to using it in our SD-VL method. This semantic segmentation network classifies every pixel in an image, resulting in an image that is segmented by classes. Pretrained version of SegNet model '*Pretrained SegNet*' was designed especially in order to segment road for autonomous driving. This model had already trained on Camvid (Cambridge Labeled Objects in Video) data set (Fauqueur et al., 2007) with provided 701 manually segmented images, that contains street-level views obtained while driving. The data set provides pixel-level labels for 32 semantic classes. In this work, these detailed 32 semantic classes compressed into 11 object classes (Building, Car, Road, Sidewalk, Sky, Tree, Pedestrian, Bicycle, Pole, Fence, SignSymbol) that also includes the previously selected 7 semantic classes. Thanks to the SegNet, more consistent semantic segmentation labels were gathered automatically with these enlarged 11 semantic classes as demonstrated in the Figure 3.7. Image pair in this figure clearly shows that, SegNet not only segments image of *Malaga Downtown Base* successfully but also it performs the same success on newly created *Malaga Streetview* test set.

The growing usage of deep learning techniques in last five years also has accelerated the development of CNN based segmentation models. And we know, the more robust semantic segmentation is managed the more successful proposed SD-VL will be performed in this study. This expectation leads us seeking for more powerful

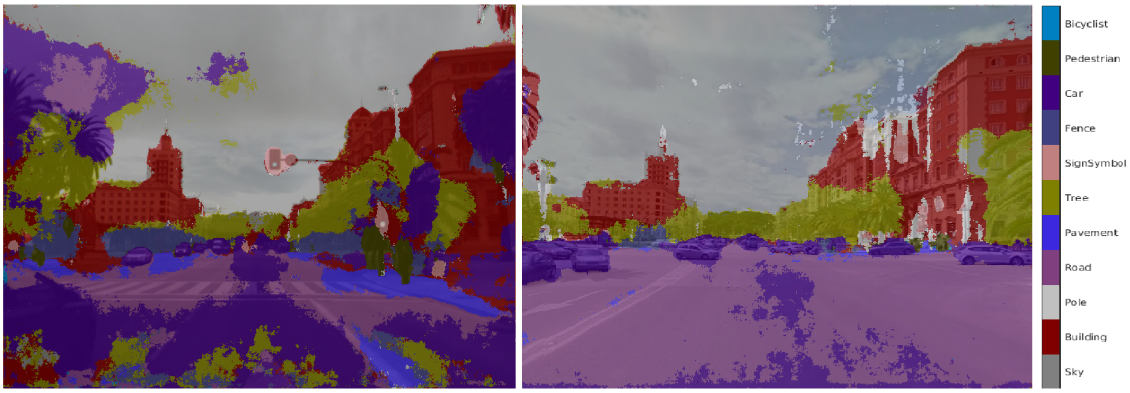


Figure 3.7. Pair of *Malaga Downtown Base* image (left side) and corresponding *Malaga Streetview* (right side) one, both are segmented semantically with ‘*Pretrained SegNet*’.

semantic segmentation model to segment our images more accurately with respect to SegNet. Hence, we came across with DeepLabv3+ (Chen et al., 2018) as a state of the art semantic segmentation method, which was designed by *Google* and meets with our expectation. DeepLab series has come along for versions from DeepLabv1 [8], DeepLabv2 [9], and DeepLabv3 [10]. In this study, its latest version ‘DeepLabv3+’ released in 2018 was employed. Also its superiority of this model against its previous version and SegNet had already shown in Section 3.2 which is mainly results from its characteristic Encoder-Decoder structure with Atrous Separable Convolutions. Whole these reasons canalize us using ‘DeepLabv3+’ model as our final baseline semantic segmentation method. First of all, its pretrained version ‘*Pretrained DeepLabv3+*’ that had already been trained on Camvid data set as like as SegNet was examined. After that, its performance is enhanced with retraining it on our RobotCar Seasons data set.

Using *Pretrained DeepLabv3+* had already increased the semantic segmentation success with respect to *Pretrained SegNet* as depicted with comparison of the first two column in Figure 3.8. Moreover, we were curious about whether the performance of *Pretrained DeepLabv3+* will be increased with retraining on our database instead of using pretrained models which is just trained on manually annotated 701 CamVid images. On the other side, we had already observed the negative impact of differences in horizon level on *Pretrained SegNet* based hand crafted SD-VL methods with respect to different query traversal of RobotCar Season data set in Figure 4.4. This fact will be reasoned with that, a semantic segmentation model trained with viewing tilt angle of CamVid inevitably gives a poor result when it is practiced on a query image of RobotCar Seasons which also has different tilt angle of view. Thus, re-training the model with different data set which have similar horizon level (horizon level of CamVid assimilating to horizon level of RobotCar Seasons) will increase the performance of semantic segmentation. Under the line of these expectation we retrained the 'DeepLabV3+' network on different traversals of RobotCar Seasons data set. In this way, we need to generate our own weakly-supervised segmented version of images as a training and validation set, because of the absence of annotated RobotCar Seasons images. We inspired from the proposed procedure as it is being in the paper named as 'Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy' (Barnes et al., 2017). We present a similar weakly-supervised approach in order to segment selected training images with the goal of autonomous driving in complex urban environments. By this way, our employed methodology generated vast quantities of labeled images especially containing our target object classes without requiring manual annotation, which we then use to train a deep semantic segmentation network.

Then as it is mentioned in previous paragraphs, re-training the model with different data set which have similar horizon level will increase the performance of semantic segmentation. On this purpose, we adjusted the horizon level of CamVid images so that it conforms to the horizon level in RobotCar Seasons. Thus and so with addition of manually labelled CamVid (701) data set to our *RobotCar Seasons Segmented-1* we created our *RobotCar Seasons Segmented-2* data set that consists of 1725 images

After generation of these two *RobotCar Seasons Segmented* sets, we randomly divided %75 of both sets as training sets and used rest of them as a validation sets. Then, 'Deeplab v3+' network whose weights initialized from a pretrained Resnet-18 network was trained with these two segmented data sets. ResNet-18 is a CNN that has 18 deep layers and which had been trained on more than a million images from the ImageNet database (Deng et al., 2009). Best training parameters were determined by means of many trials, and we had carried out a healthy training process with a validation accuracy that nearly reaches to % 92 for both segmented sets. Resulting re-trained models on *RobotCar Seasons Segmented-1* and *RobotCar Seasons Segmented-2* were named as *DeepLabV3+ Retrained-1* and *DeepLabV3+ Retrained-2* respectively. Steps of this weakly-supervised segmentation methodology is given below:

- All images of 7 traversals (Table 3.1) without *night* and *night-rain* in RobotCar Seasons data set were semantically segmented with *Pretrained DeepLabv3+* model.
- After visualizing all these segmented images, nearly 150 images per traversals which reflect our semantic classes in the best way were selected systematically.
- Selected images which composes our segmented label data set are excluded from their traversals (query sets).

Thanks to this weakly-supervised segmentation, our new segmented data set that is named as *RobotCar Seasons Segmented-1* consists of 1024 images. Then as it is mentioned in previous paragraphs, re-training the model with different data set which have similar horizon level will increase the performance of semantic segmentation. On this purpose, we adjusted the horizon level of CamVid images as similar as RobotCar Seasons' horizon level. Thus and so with addition of manually labelled CamVid (701) data set to our *RobotCar Seasons Segmented-1* we created our *RobotCar Seasons Segmented-2* data set that consists of 1725 images.

After generation of these two *RobotCar Seasons Segmented* sets, we randomly divided %75 of both sets as training sets and used rest of them as a validation sets. Then, 'Deeplab v3+' network whose weights initialized from a pretrained Resnet-18 network was trained on these two segmented data sets. ResNet-18 is a CNN that has 18 deep layers and which had been trained on more than a million images from the ImageNet database (Deng et al., 2009). Best training parameters were determined by means of many trials, and we had carried out a healthy training process with a validation accuracy that nearly reaches to % 92 for both segmented sets. Resulting re-trained models on *RobotCar Seasons Segmented-1* and *RobotCar Seasons Segmented-2* were named as *DeepLabV3+ Retrained-1* and *DeepLabV3+ Retrained-2* respectively.

These new ‘DeepLabv3+’ based retrained semantic segmentation models are compared in Figure 3.8 with *Pretrained DeepLabv3+* and ‘*Pretrained SegNet*’ that both are previously trained just on manually annotated CamVid driving set.

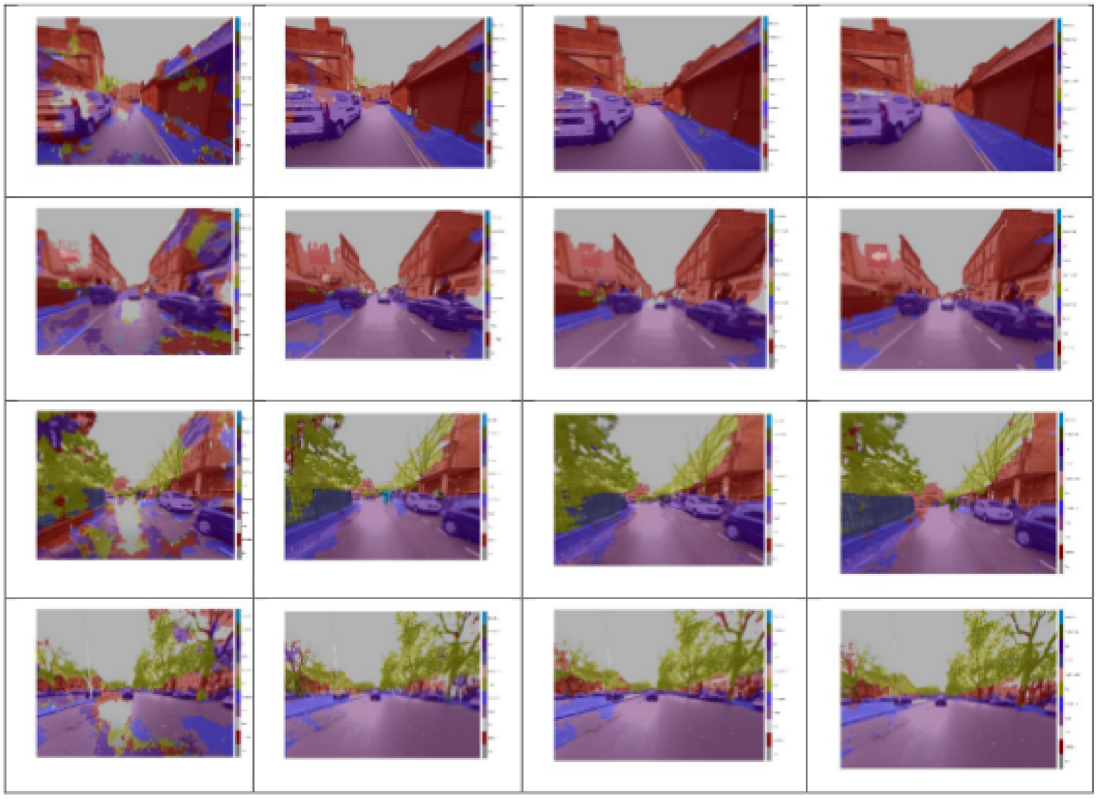


Figure 3.8. Segmentation performance comparison between performed semantic segmentation models, ‘*Pretrained SegNet*’, ‘*Pretrained DeepLabv3+*’, ‘*DeepLabV3+ Retrained-1*’ and ‘*DeepLabV3+ Retrained-2*’ respectively from left to right for those same images (columns).

In Figure 3.8 each column corresponds to compared these 4 models *Pretrained SegNet*, *Pretrained DeepLabv3+*, *DeepLabV3+ Retrained-1* and *DeepLabV3+ Retrained-2* respectively from left to right. And each row corresponds to same image that is selected from the different traversals of *RobotCar Seasons* data set. This figure obviously visualized that, there is a big semantic segmentation difference especially between *Pretrained SegNet* and *Pretrained DeepLabv3+* (from 1_{st} column to 2_{nd}

column). In addition to this, we also could say that retraining a model makes contribution to semantic segmentation performance as we observe the increasing segmentation success from 2_{nd} column to 3_{rd} one. At last column, *DeepLabV3+ Retrained-2* which retrained on the enlarged *RobotCar Seasons Segmented-2* data set is deduced as the superior one.

Furthermore, we also employed the same *DeepLabV3+ Retrained-2* model successfully while obtaining the semantically segmented version of *Malaga Streetview Challenge* data set without retraining on this data set again. Robustness of this model against these unseen images is visualized in Figure 3.9.

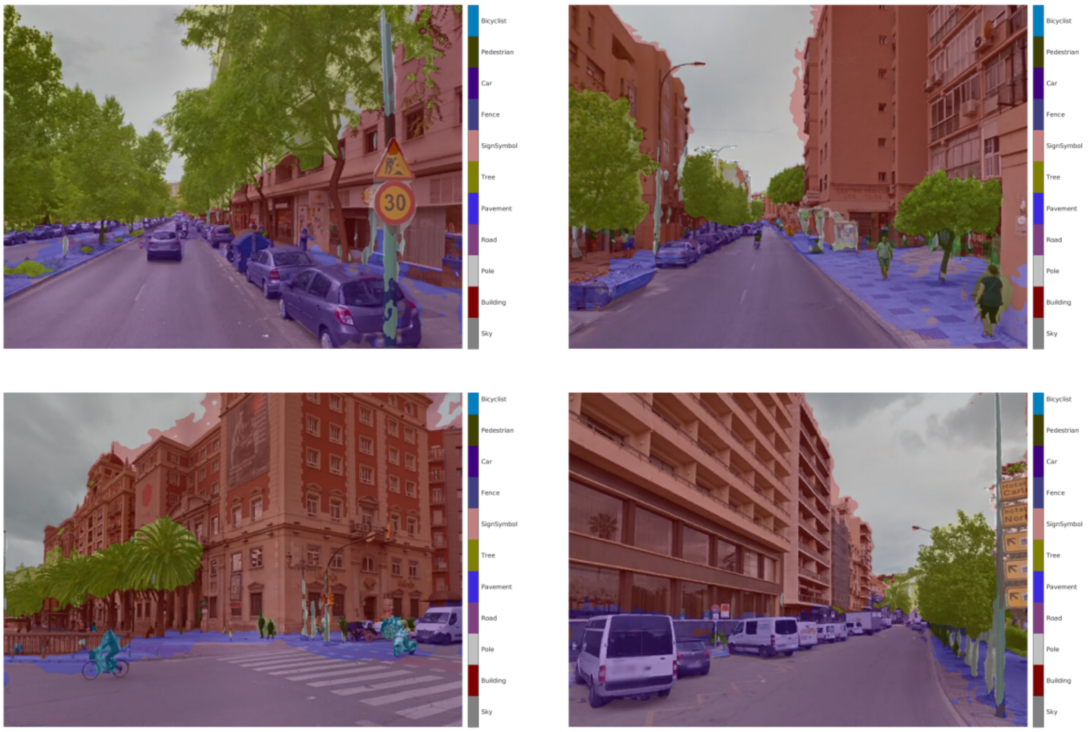


Figure 3.9. Successful semantic segmentation performance of *DeepLabV3+ Retrained-2* on some sample images of *Malaga Streetview Challenge* data set.

As a result of its approved success (Fig. 3.8, Fig. 3.9), in this study we mainly employed the '*DeepLabV3+ Retrained-2*' model as a baseline method while we were segmented our images into 11 semantic classes.

3.3. Non-Learnt Descriptor based VL's

In this section, employed SURF based LD-VL and newly generated SD-VL methods based on segmentation are introduced. These '*non-learnt*' VL approaches which means we did not train them for our localization task were applied with integration of our ANNS methods FLANN, and they were accepted as our base hand crafted approaches in this study. Also, our initial Hybrid-VL methods were instructed on these LD-VL methods. Our first promising results for Hybrid-VL were acquired thanks to these base methods in the early stage of this work.

3.3.1. Hand Crafted LD-VL with SURF

Interest points of related images were extracted by means of SURF descriptor on which our hand crafted LD-VL method is instructed. In the early stage of this study this *non-learnt* LD-VL method was employed especially for the poof of the concept (Section 4.2) in which our initial promising results were obtained.

The important points we need to pay attention in this section is that, if we manage to represent whole image with a single compact descriptor, image matching will be easily performed with directly usage of FLANN. On the other hand, if we use LD like SURF we need to apply special matching procedure which is constructed on repetition number of matching features which are returned as interest (key) points of an image. Because SURF descriptor represents an image with changeable number of binary feature vector $[1X64]$ whose number varies form image to image. And then, FLANN matching algorithm makes their matching on data points, where these data points correspond to our binary feature vectors in an image. In line with solving this problem we converted this descriptor level matching into image level matching by implementing a powerful indexing that provides us the information of which LD's come from which images. The idea is simple: given a certain collection containing images, and a small image to be matched, the FLANN must process each one of them, determining a value which represents how many key points from the query image have been spotted into the cycled collection image. At the end of the loop, the collection image that will have returned the higher values are more likely to be the results we expect (in other word, higher the score, higher are the chances our query is contained into those images).

After making FLANN compatible with SURF based LD's we called this hand crafted method as $LD\text{-}VL_{SURF}$. Also note that, all the other implemented methods that represent whole image with a single compact descriptor do not need to use this compatibility procedure while using our ANNS method.

3.3.2. A Novel Hand Crafted SD-VL with Segmentation

Semantic information were extracted from equally divided parts of our segmented images as a novel SD on which our hand crafted SD-VL method is instructed. In the early stage of this study this *non-learnt* SD-VL method was especially employed (Section 4.2) to obtain our initial promising hybrid results. Moreover, necessity of location-aware SD was came to light owing to several experiments in which this hand crafted descriptor based SD-VL was compared.

Semantically segmented images were acquired with different type of approaches as described in Section 3.2. Unfortunately this semantic information does not mean anything without converting them into a useful SD descriptor which also has a compatible form for usage of FLANN. Images in *Malaga Streetview Challenge* (2007) and *RobotCar Seasons* had already been segmented up to 11 semantic classes (Building, Car, Road, Sidewalk, Sky, Tree, Pedestrian, Bicycle, Pole, Fence, SignSymbol) according to employed semantic segmentation methodology as described in the related part. Actually, these semantic classes were especially preferred in order to create our hand crafted SD descriptor, because contribution of these frequently seen classes must be greater than other rarely seen classes on transmitting the semantic information.

In line with converting semantic information into a compact feature vector, the ratio of amount of the pixels assigned to each semantic class to all pixels of relevant region was embedded into a feature vector $[1 \, \text{x} \, n_{SC}]$, where n_{SC} denotes number of semantic classes. By this way in each cell of this vector, repetition information belonging to each classes are stored as a decimal number which ranges between 0-1. After this process was applied to the whole segmented image, the same process was repeated on the equally divided sub parts of the images, where n_{SP} denotes number of sub-images. So that we generated extra n_{SP} number of $[1 \, \mathrm{x} \, n_{SC}]$ feature vectors. Concatenation of these feature vectors resulted in with our final hand crafted SD with the given size below:

$$SD_{non\_learnt} = [1 \, \mathrm{x} \, n_{SC}] + n_{SP} * [1 \, \mathrm{x} \, n_{SC}] \tag{3.1}$$

For a clear comprehension, Figure 3.10 visualizes the Equation 3.1 on the same image used in Figure 3.6. In this figure, implementation parameters n_{SC}=7 and n_{SP}=4 denote 7 semantic classes and 4 equal sub parts that give us 5 pieces of $[1\mathrm{x}7]$ feature vectors. After concatenation of these 5 vectors this sample hand crafted SD is finally represented with a $[1\mathrm{x}35]$ sized vector.

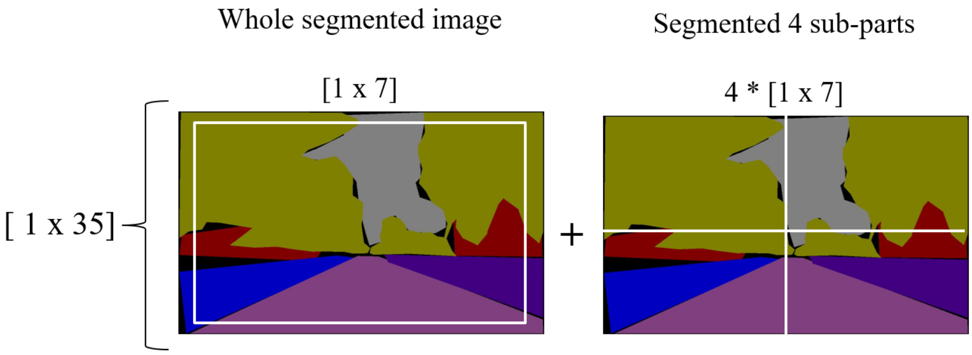


Figure 3.10. Hand crafted SD representation with 4 sub-images: each cell $[1 \, \mathrm{x} \, 7]$ of these vector stores the percentage of the corresponding semantic class pixel.

Furthermore, someone will ask this question "how many number of sub part should a segmented image divided into?". In order to find answer to this question, we applied experiments without and with dividing the whole segmented image into different numbers of sub parts like 4,9,16,64. Dividing our image into 4 equal parts with addition of whole image gave the optimal result for the VL task. This result will be reasoned with that, just using the whole segmented image makes our hand crafted SD behave like a global descriptor, on the other side using segmented image that is equally divided into 64 sub parts makes it behave like a key-point local descriptor. As a result of this fact, proposed *non-learnt* SD in this study were generated with concatenation of whole segmented image and its 4 equally divided parts as it is demonstrated in Figure 3.10.

This novel hand craft SD extraction methodology provides us the distinguishing talent of the relative positions of the objects in an image; in example this proposed SD easily able to differentiate a scene in which trees appear on the left side and building appears on the right side from another scene in which these objects takes part in revers side.

Moreover, this proposed compact feature vector comes from a whole segmented image also makes our SD work directly with FLANN matching algorithm without any extra indexing procedure as being for $LD\text{-}VL_{SURF}$. Because we can directly obtain our SD descriptor SD_{non_learnt} which able to represent a whole segmented image with one feature vector (Equation 3.1).

To sum up, different type of hand crafted SD-VL methods were built up by incorporation of our ANNS method. Regards to using the previously introduced (Section 3.2) semantic segmentation models: '*LIBLABEL*', '*Pretrained SegNet*', '*Pretrained DeepLabv3+*', '*DeepLabV3+ Retrained-1*' and '*DeepLabV3+ Retrained-2*'. These '*non-learnt*' SD-VL methods were named as $SD\text{-}VL_{LIBLABEL}$, $SD\text{-}VL_{PretSegNet}$, $SD\text{-}VL_{PretDeepLabv3+}$, $SD\text{-}VL_{DeepLabV3+\_Retr1}$ and $SD\text{-}VL_{DeepLabV3+\_Retr2}$ respectively.

3.4. Learnt Descriptor based VL's

In this section, employed NetVLAD based LD-VL and newly proposed SD-VL methods based on segmentation with triplet loss are introduced. The state-of-the-art VL performance is achieved by NetVLAD as underlined in Section 2.1, therefore this architecture was employed while training our descriptors for our localization task. Then, these location-aware SD and LD descriptors were integrated with FLANN and they were named as ‘*learnt*’ VL approaches. Furthermore, these learnt VL approaches were used as our baseline approaches in this thesis and comparison with recent work could be possible owing to these approaches. In other words, the final success (Section 4.4) of improved Hybrid-VL methods were obtained thanks to these ‘*learnt*’ descriptor based VL methods.

Pretrained networks have been recently used as off-the-shelf dense descriptor extractors for image retrieval task. Differently from the previous deep learning based VL approaches, NetVLAD (Arandjelovic et al., 2016) trains a CNN in order to optimize and output the embedding itself directly, rather than optimizing an intermediate bottleneck layer generating embedding. NetVLAD architecture was properly designed for VL task inspired by the VLAD representation (Jégou et al., 2010). In other words, *NetVLAD* mimicks the VLAD in a CNN framework by converting it into trainable generalized VLAD layer. This trainable VLAD layer is illustrated in Figure 3.11. In this figure, *K* denotes the number of cluster used while clustering all extracted descriptors, also detailed explanation related with VLAD descriptor will be found in Section 2.1. This networks is mainly constructed on two well known pre-trained architecture VGG16 (Simonyan and Zisserman, 2014) and AlexNet (Krizhevsky et al., 2012) and initialized with their pretrained weights. Both these networks are pretrained for classification task on ImageNet (Deng et al., 2009) and Places205 (Zhou et al., 2014), thus both are cropped at the last convolutional layer (conv5), before ReLU then these base architectures are extended with NetVLAD layers instead of Max pooling. This developed new pooling layer aggregates mid-level (conv5) convolutional features extracted from the entire image into a fixed length vector representation and its parameters are learnable via back-propagation.

Learning phase is implemented with triplet loss which was firstly introduced in FaceNet (Schroff et al., 2015). NetVLAD adopted the same triplet loss training procedure for VL task instead of face recognition and this new loss named as ‘Weakly supervised triplet ranking loss’. Let the desired location-aware descriptor is represented with $f_\theta(q) \in \mathbb{R}^d$ where a query image q is embedded into a d-dimensional euclidean space. From

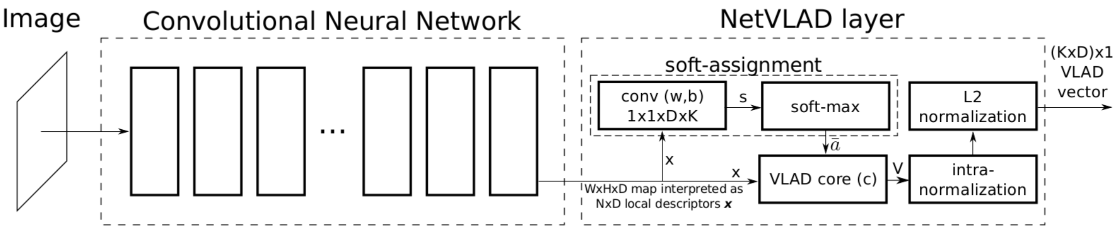


Figure 3.11. NetVLAD architecture (Arandjelovic et al., 2016).

RobotCar Seasons and *Malaga Streetview Challenge* data sets, we obtain a training data set of tuples (q, p_i^q, $\{n_j^q\}$), where for each training query image *q* we have a best matching positive p_i^q and a set of definite negatives $\{n_j^q\}$. Here we want to optimize the training parameters θ that a query image q (anchor) of a specific location is closer to a differently-viewed version p_i^q (positive) of the same place than it is to any image n_j^q (negative) of any other place. This triplet loss based optimization is visualized in Figure 3.12 and its translation into a ranking loss for VL task is summarized in the following paragraph.

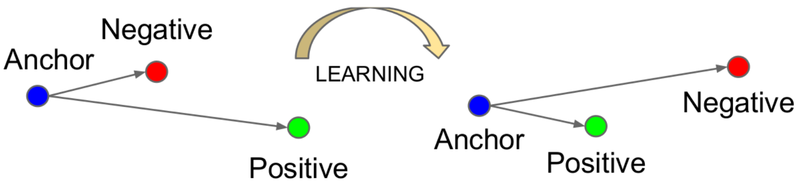


Figure 3.12. Triplet Loss minimizes the distance between the anchor(query) and its positive sample, while maximizes the distance between the same anchor and its negative exemplars.

We assume a query q is given as a test image. Same 'triplet ranking loss' for NetVLAD applied in this study but with a different selection of best matching positive image p_i^q

$$p_i^q = \underset{t^{db}}{\operatorname{argmin}}\ d_{gps}(q, t^{db}) \tag{3.2}$$

among the database images t^{db} of training set for each training tuple (q, p_i^q, $\{n_j^q\}$).

This difference is raised because of the usage of weakly labeled Google Streetview Time Machine panoramas in the original NetVLAD implementation. A panoramic image depicting the same place from different viewpoints over time, therefore they find the best matching slice of this panoramic image by computing the euclidean distance between a query image and this slices (potential positives). However in this study our data sets directly provide us the over time effect with their traversals that contains sequential perspective images. Therefore we directly accepted the best matching positive p_i^q by determining the nearest (meter) database images in accordance with GPS information. The objective then becomes to learn an image representation $f_\theta(q)$ so that euclidean distance $d_\theta(q, p_i^q)$ between the training query q and the best matching positive p_i^q is smaller than the distance $d_\theta(q, n_j^q)$ between the query q and all negative images in $\{n_j^q\}$:

$$d_\theta(q, p_i^q) \; < \; d_\theta(q, n_j^q), \quad \forall j. \tag{3.3}$$

Under the light of this objective, final 'triplet ranking lost' L_θ for a training tuple $(q$, p_i^q, $\{n_j^q\})$ is defined as in NetVLAD (Arandjelovic et al., 2016)

$$L_\theta \; = \sum_j h \left(d_\theta^2(q, p_i^q) \; + \; m \; - \; d_\theta^2(q, n_j^q) \right), \tag{3.4}$$

where h is the hinge loss $h(x) = max(x, 0)$, and m is a margin that is enforced between positive and negative pairs. The Equation 3.4 is a sum of losses for negative images in $\{n_j^q\}$. According to this function, for every negative that has distance between the query and the negative is greater by a margin than the distance between the query and the best matching positive, the loss h is zero. In a reverse condition, if the margin is violated by the distance to the negative image that is exceedingly grater than the distance to the best matching positive, the loss increases in proportional to the amount of violation. This loss function in NetVLAD architecture makes our '*learnt*' descriptors end-to-end trainable on our data sets.

In order to train, triplets of roughly aligned matching / non-matching place tuples generated using a novel online triplet mining method for not only training data set but also for validation and test sets. Both the database and query sets of the used data sets are divided into mutually geographically disjoint 3 parts for training, validation and testing. This division was done geographically to ensure these 3 separated parts contain independent image from each other while ensuring to have same interval of prior map withing themselves for their database and query set. This division will be explained with an example. Consecutive images within a training part must have overlapping visual content for its own query and database sets, while images in this part are geographically independent from the validation and testing parts. This division certainly affects the performance of training and testing of our proposed VL's. Therefore VL's are employed at least 2 different variants (Section 3.1) of these divisions for both data sets. As a result of this divisions *Malaga Steetview Challenge* data set contains around 523 database images and 145 queries; *RobotCar Seasons* data set contains around 2318 database and 150 query images for each of its traversals.

Moreover, in order to ensure fast convergence it is critical to select hard triplets that violate the triplet constraint m in Equation 3.4. This means that, for a given q , we should select an n_j^q (hard negative) such that

$$\operatorname*{argmin}_{n_j^q} d_\theta^2(q, n_j^q). \tag{3.5}$$

The similar hard mining could not be applied for hard positives because we directly selected the best matching positive p_i^q in regarding to nearest GPS location. However, applying this hard negative triplet mining across the whole training database set is infeasible. Therefore efficient online triplet generation can be done by selecting the hard negative exemplars from within a mini-batch. Also to speed up the training, NetVLAD provides us to compute a cache of all the features, to be used for a number of iterations, and then be recomputed again. This number is called 'compute and cache frequency', and it's best value is between 1000 and 500. This triplet ranking loss training procedure and its usage in this study are summarized below:

- Establish NetVLAD framework: the network is constructed on the pretrained base

(VGG16 (Simonyan and Zisserman, 2014) or AlexNet (Krizhevsky et al., 2012)) models that is extended with NetVLAD (Arandjelovic et al., 2016) as depicted in Figure 3.11. Also these networks are initialized with pretrained weights of their base models.

- Take the training and validation sets: after our data sets are geographically divided into training,validation and testing parts for each of the database and query sets as depicted in previous paragraphs, all these two parts are stored in a structure. This structure consists of the path for the query and database images, how many database and query images there are, the coordinates, thresholds in meter for choosing hard negative samples (TNS: default value is 25 meter) and some others information. Then, this carefully created structure is given to the previously defined dataloader as a argument, that additionally computes and stores the suitable closest point according to previously determined triplet threshold.

- Initialize the training: all queries in training set is separated according to batch size for each epochs, then descriptors of the database are extracted for each sequential batches. This extraction is performed on each queries and its corresponding tensor returned by the same dataloader, which simultaneously computed the nearest (Eq. 3.2) positive and the 10 hard negatives (Eq. 3.5) for each query according to previously determined closest points (TNS). Thanks to the caching process, sequential database descriptor are computed and saved for a fixed interval that provides us an efficient computation. After that, the Triplet Loss (Eq. 3.4) is calculated and the backpropagation algorithm is performed for all the subsets of queries set.

- Pick the best model: success of the each epoch is examined on the previously generated test set (database & query images) and results of each epochs compared thanks to stored checkpoints. Comparison is conducted by means of Recall@5 evaluation metric, which give percentage of correctly localized queries with respect to top 5 returned candidates that lies inside the TNS meter radius of the ground truth query position. Finally, after best epoch is determined its corresponding model is selected as the trained *best model*.

- Extracting learnt descriptor: testing data set (database & query) is also loaded with the same dataloader. Then, the best model obtained from the training is employed on both the database and query images of this data set. As a result of this feature extraction, we gained the 16k and 32k dimensional VLAD vectors (Figure 3.11) with $K = 64$ cluster numbers regarding to used base models AlexNet and VGG16 respectively . These image representation was used as a learnt SD & LD descriptors in our proposed VL methods.

The essential code that was used and adjusted in this study will be downloaded from a GitHub repository (Arandjelović, 2015), where the NetVLAD was originally developed in Matlab using MatConvNet (Vedaldi and Lenc, 2015) CNN toolbox. Our trainings on NetVLAD architecture have been done in GPU mode with the given configurations:

– Base model: AlexNET (Krizhevsky et al., 2012) or VGG16 (Simonyan and Zisserman, 2014) pre-trained on ImageNet (Deng et al., 2009) and Places205 (Zhou et al., 2014);

– Training Loss: NetVLAD (Arandjelovic et al., 2016) triplet ranking loss;

– Layer Name: Crop the initial network at last convolutional layer, conv5 for AlexNet, conv5_3 for VGG16;

– Trainable Layers: whole NetVLAD layer + layers strat from last conv layers of base models to changing former layers(conv5_1 for VGG16 & conv2 for AlexNet);

– Optimizer: Stochastic Gradient Descent (SGD) ;

– Number of clusters K: 64, clustering the extracted descriptors in VLAD (Jégou et al., 2010) manner;

– Learning rate: 0.0001;

– Momentum: 0.9;

– Weight decay: 0.001;

- Margin: 0.1
- Number of Epochs: 15;
- Batch size: 4 tuples (each tuple contains the query, the positive and at most 10 negatives);
- TNS: 25 / 70 (w.r.t. data set) meter threshold for choosing negative samples;
- Compute and cache frequency: 1000 features;

Some of these configuration above are given with changing options, these options had already been examined and its impact on our VL's method is depicted in Chapter 4.

3.4.1. LD-VL with NetVLAD

NetVLAD based triplet ranking loss was directly employed on RGB rear images (database - query sets) of *RobotCar Seasons* and we obtained the '*learnt*' image representation with 16K/32K dimensional feature vectors. In other words, instead of using a hand-crafted descriptor (SURF) we used CNN based (learnt) descriptors and integrated them with our ANN image matching method. In this way we instructed our baseline LD-VL method that is called $LD\text{-}VL_{NetVLAD}$. This baseline method was implemented for different traversals (e.g. Overcast-Winter) of *RobotCar Seasons* data set so that their *best models* are gained individually. During the network training phase of this method, we followed the same steps with the same configurations as represented in the previous section. Detailed generation parameters of the gained best models for this method will be found in Section 4.3 with their corresponding experimental results. By this way, similar state-of-the-art performance for this traversals was achieved regarding to recent works which approves the reliability of our baseline methods. After this confirmation on *RobotCar Seasons*, the same learning procedure is repeated with the same parameters (with different TNS) on the *Malaga Streetview Challenge* data set. So that, specific $LD\text{-}VL_{NetVLAD}$ was generated for this data set with its *best model*. Also note that, different appropriate TNS parameters are applied regarding to sparseness of data sets.

3.4.2. A Novel SD-VL trained with Triplet Ranking Loss

Using the proposed hand crafted descriptor based '*non-learnt*' SD-VL methods, which are described in Section 3.3.2 in detailed, will be inadequate for our localization task as it is showed up in Section 4.3. In order to cope with this inability, the same triplet ranking loss was implemented on semantically segmented versions of the same images (database-query sets) belonging to *RobotCar Seasons* and *Malaga Streetview Challenge* data sets.

In order to be fair against the baseline $LD\text{-}VL_{NetVLAD}$ method, the same training steps which are given at the end of Section 3.4 was implemented with the same parameter configuration (*Base model*, *NetVLAD layer* based pooling, *Optimizer*, *Trainable Layers*, *K*, *TNS* etc.). In addition, also the *best model* of $SD\text{-}VL_{Learnt}$ method was generated on the same divided variants of the same traversals which are used for $LD\text{-}VL_{NetVLAD}$ method. By this way, our *learnt* SD was extracted with localization objective as illustrated in Figure 3.13. Note that, CNN part of this new training setting visualized in Figure 3.13 corresponds to the same architecture of NetVLAD (Figure 3.11). In other words, instead of using a hand-crafted semantic descriptor (Section 3.3.2) we used CNN based (learnt) descriptors and integrated them with our ANN image matching method. Then this new VL method was named as $SD\text{-}VL_{Learnt}$.

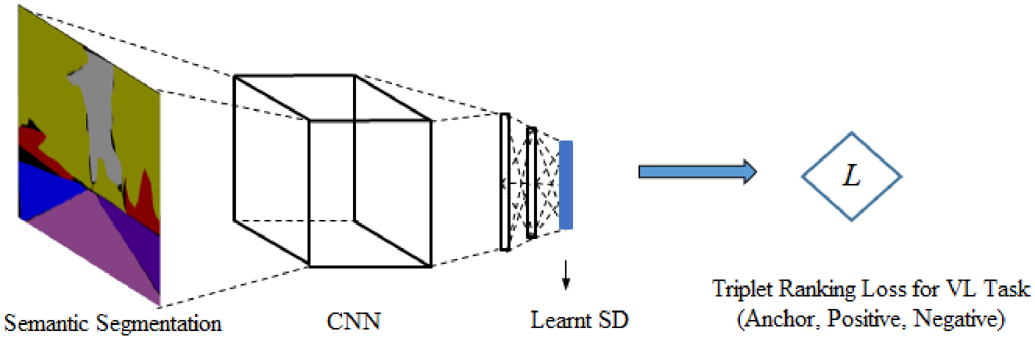

L
Semantic Segmentation
CNN
Learnt SD
Triplet Ranking Loss for VL Task
(Anchor, Positive, Negative)

Figure 3.13. *Learnt* SD trained with triplet ranking loss for VL task on semantically segmented images.

Superiority of our retrained semantic segmentation model '*DeepLabV3+ Retrained-2*' on both data sets had already been demonstrated in Section 3.2 with Figure 3.8 and Fig. 3.9. Thus, as an input of proposed $SD\text{-}VL_{Learnt}$ method, we used images which had been previously segmented into 11 semantic classes by means of this baseline '*DeepLabV3+ Retrained-2*' model. Consequently, thanks to this novel SD-VL method which works on semantically color coded images, we directly incorporated the localization awareness with the distinguishing power of the relative positions of the objects in an scene. Also success of the proposed $SD\text{-}VL_{Learnt}$ method has already been demonstrated in our previous study (Cinaroglu and Bastanlar, 2020). Detailed generation parameters of the gained best models for this method will be found in Section 4.4 with their corresponding experimental results.

3.5. Improved Hybrid-VL Methods

As a main contribution of this study, firstly a novel Hybrid-VL$_{DL}$ method is proposed by combining SD-VL and LD-VL methods with the aim of alleviating the drawbacks of both methods. The success of Hybrid-VL$_{DL}$ method is obtained by means of hyper-parameter *W* as described in Subsection 3.5.1 and its empirical adjustment is given in Section 4.6. Secondly, Hybrid-VL$_{FL}$ is proposed in order to gather a automatically tuned Hybrid-VL result as described in Subsection 3.5.2. Thanks to the Hybrid-VL$_{FL}$ method, optimum hybridization parameters (hidden) are determined by NN trained with triplet loss instead of relying on any hyper-parameter. By this way, reliability of our hyper-parameter (*W*) based Hybrid-VL$_{DL}$ approach is supported by very close performances obtained with Hybrid-VL$_{FL}$ method as given in the next chapter.

3.5.1. Improved Decision Level Hybrid-VL Method

Before giving details it should be noted that, there is no need to use any ANNS while producing our Hybrid-VL$_{DL}$ methods. Because, hybridization occurs in post-process stage and *k* number of matching results had already been obtained as a product of FLANN which came from both methods. Then this results are combined regarding to their retrieving orders of candidate images and their distance values in this proposed hybridization methodology. The same methodology is employed for all Hybrid-VL$_{DL}$ methods examined in the next chapter with different combinations of SD-VL and LD-VL methods.

Our ANNS method had already synchronized for both SD-VL and LD-VL methods with suitable adaptation in which images are directly compared owing to used compact vector representations (16K / 32K). ANNS was employed for *k*=10 nearest neighbors in all experiments as depicted in the representation of proposed algorithm (Figure 3.2). Assume that, 10 nearest candidate images stored in SD_i and LD_i are returned from both two methods with corresponding distance vectors $D_j(SD_i)$ and $D_j(LD_i)$ respectively for a given ith query image. Where *j* refers *k* nearest candidate that changes from 1 to 10. Note that, $D_j(SD_i)$ and $D_j(LD_i)$ are returned in ascending order. Then, assume we have *m* number of query images for our SD-VL method, both of the retrieved candidate vectors SD_i and $D_j(SD_i)$ are concatenated in matrices k x m_{ngh}, k x m_{dist} (Section 2.2) respectively as a collection. Also same matrices are gathered for the LD-VL methods. There is unequal distribution in values, between these distance collections k x m_{dist} returned from each method. In order to achieve a reliable hybridization based on distance values we need to normalize these collections within themselves in $[0 - 1]$ range. Unfortunately, the normalized distance values in each collections still will be unevenly distributed as depicted in the top row of Figure 3.14. Thus, histogram equalization on both kxm_{dist} matrices were carried out in order to obtain a balanced distribution which is depicted in Figure 3.14.

As depicted in Figure 3.14, normalized LD-VL distances are condensed close to zero, whereas normalized SD-VL distances are closer to one. Before explaining the reason behind this imbalance distribution we shouldn't forget that these values in Figure 3.14 (top row ones) are scaled $[0 - 1]$ euclidean distances between descriptor pairs. From the side of LD distances, here there are small number of outliers which are extreme-high distance values (such as 1000) with respect to majority of LD distances. Hence, when we normalize all these values in $[0 - 1]$ range, majority of distribution stick to left side (0) of scale (top-left histogram in Figure 3.14). The reason for these extreme-high outliers is the diversity between descriptors, because there are detailed differences between a mosque and a house, although they are both buildings(round large windows vs small square windows). On the other hand, SD distribution also has small number of outliers, but these are extreme-low distance values (such as 0.001) according to majority of SD distances. Therefore, when we normalize all these values in $[0-1]$ range, majority of distribution stick to right side (1) of scale (top-right histogram in Figure 3.14). The reason of these extreme-low outliers is that, both a mosque and a house are labeled as a building which means there may be segmented images that are identical to each other. Although learnt-SD learns to separate this similarity, this inevitable sameness makes these extreme-low distance values possible between outlier descriptors.

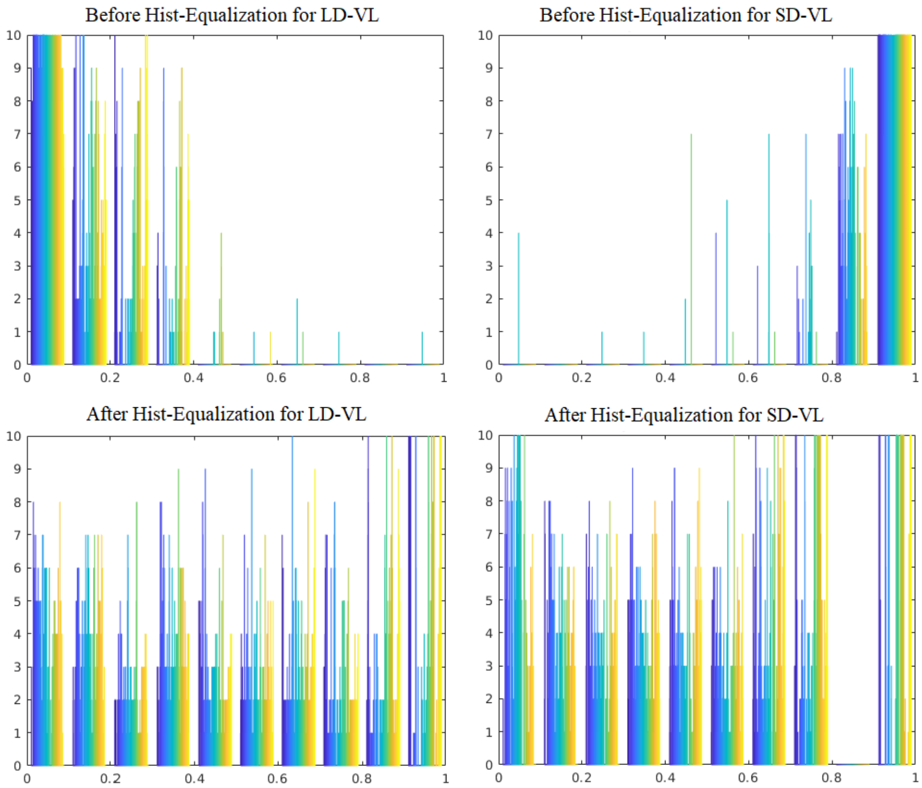


Figure 3.14. Distributions of distance values in k x m_{dist} matrix for LD-VL (1_{st} column) and SD-VL (2_{nd} column) methods before (1_{st} row) and after(2_{nd} row) histogram equalization.

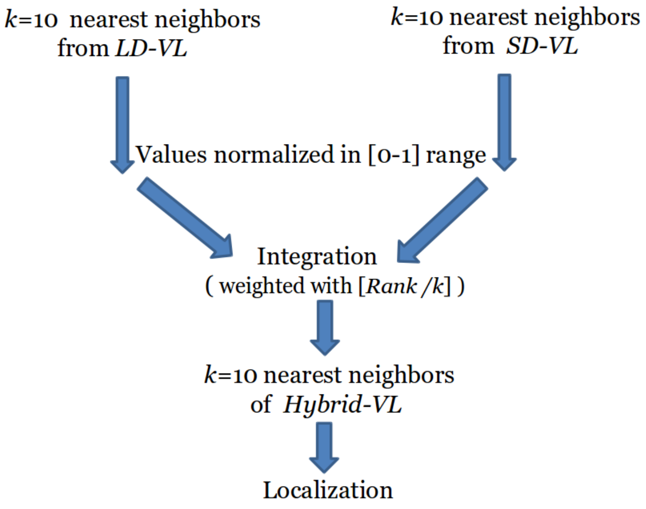
After pre-processing the distance values for a reliable integration, now we able to combine the SD_i and LD_i results came from both VL methods for all query images as displayed in Figure 3.15. In order to achieve this integration strategy, current distance values ($D_j(SD_i)$ and $D_j(LD_i)$) per each query images are updated by weighting each candidates with respect to their ranking and *W* hyper-parameter. This hybrid distance updating equation is given below:

$$D_{new}(i_j) = \begin{cases} D_j(SD_i) * (\frac{rnk_j(SD_i)}{k}) * W + D_j(LD_i) * (\frac{rnk_j(LD_i)}{k}) * (1 - W) & \text{if } j \in (SD_i \cap LD_i) \\ D_j(LD_i) & \text{else if } j \in (LD_i) \\ D_j(SD_i) & \text{else if } j \in (SD_i) \end{cases} \tag{3.6}$$

where $rnk_j(SD_i)$ and $rnk_j(LD_i)$ denote the ranking of *j* th candidate image in SD_i and LD_i lists. We know that $D_j(LD_i)$ and $D_j(SD_i)$ are already normalized in the range $[0-1]$, but summation operation in the first case $j \in (SD_i \cap LD_i)$ of the Equation 3.6 disrupts the normalization situation against the other two cases ($j \in (LD_i)$, $j \in (SD_i)$). In other words, summation $D_j(LD_i) + D_j(SD_i)$ makes first case disadvantageous with total distance value varies in $[0-2]$ range. In order to be fair against all these cases, while updating a hybrid distance $D_{new}(i_j)$ value in case 1, we fit them in range $[0-1]$ again by weighting the both side of summation operator with *W* and '1-*W*' hyper-parameters. At the same time, thanks to the *W* parameters we can also tune the contribution of SD-VL and LD-VL in the first case. The lower *W* value we set in case 1, the more we trust on LD-VL method.

At the same time among these three cases, to increase importance of first case $j \in (SD_i \cap LD_i)$ against others, we also reward these $D_j(LD_i)$ and $D_j(SD_i)$ distances directly proportionate to their rankings $\frac{rnk_j}{k}$ (w.r.t ascending order). I.e. for higher rank candidates distance values are even decreased. As a last step of this hybridization, we reorder these previously returned 20 candidate images (SD_i and LD_i) according to their updated distance values $D_{new}(i_j)$, then we accept the top 10 images in this new list as a final result of Hybrid-VL$_{DL}$ method.

I should note that, fine-tuning the decision level hybridization with *W* hyper-parameter is very important point. We should trust on the better VL method



k=10 nearest neighbors from *LD-VL*
k=10 nearest neighbors from *SD-VL*
Values normalized in [0-1] range
Integration (weighted with [*Rank* /*k*])
k=10 nearest neighbors of *Hybrid-VL*
Localization

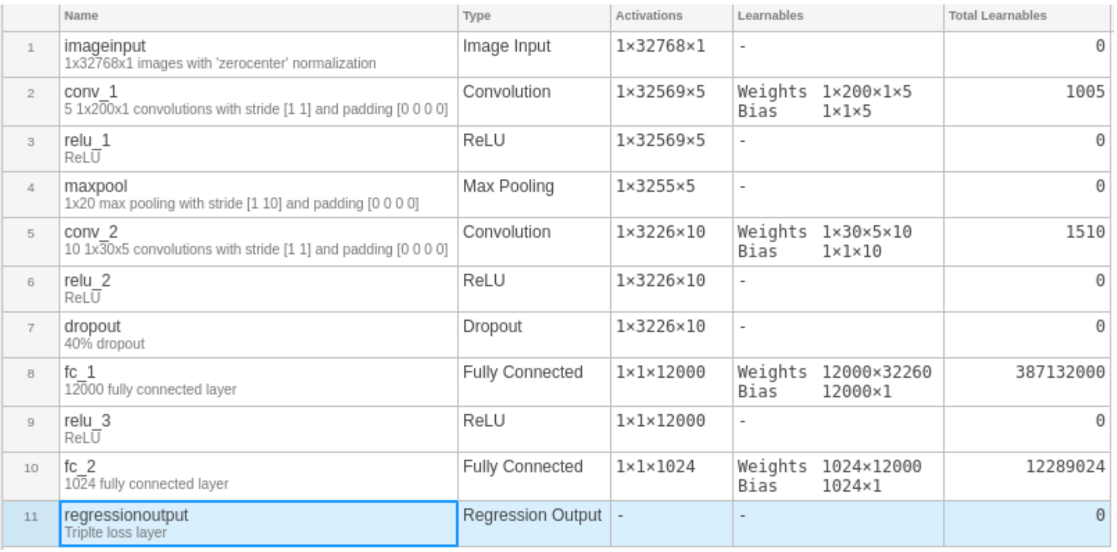
Figure 3.15. Decision-level Hybridization methodology.

among the SD-VL and LD-VL methods. Impact of 'tuning with W parameter' of Hybrid-VL$_{DL}$ is demonstrated in Section 4.6.

3.5.2. Improved Feature Level Hybrid-VL Method

Our proposed decision-level Hybrid-VL approach depends on a externally tuned hyper-parameter *W*. We also investigate if an automatically tuned hybrid method achieve the same or better performance under same conditions. Therefore, a new Hybrid-VL$_{FL}$ method that is based on NN trained with triplet loss is proposed in order to produce automatically tuned hybrid result. In other words, this newly proposed Hybrid-VL fuses the same SD and LD used in Hybrid-VL$_{DL}$ but it exploits a NN based training instead to fuse matching results of these descriptors in post-processing level. In this way, we directly obtain a best trained NN model that produces automatically tuned hybrid descriptor for an image without making any manual parameter tuning.

In accordance with this purpose, we design our own NN that includes Convolution layers (CL) and a Fully Connected layers (FCL) as depicted in Figure 3.16. This NN design was chosen among many examined differently designed NNs such as just including FCLs or CLs or including them in different number, order and combination. While choosing this best NN design we considered their triplet loss training performance on the test sets for both *RobotCar Seasons* and *Malaga Streetview Challenge* data sets.



	Name	Type	Activations	Learnables	Total Learnables
1	imageinput 1x32768x1 images with 'zerocenter' normalization	Image Input	1×32768×1	-	0
2	conv_1 5 1x200x1 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	1×32569×5	Weights 1×200×1×5 Bias 1×1×5	1005
3	relu_1 ReLU	ReLU	1×32569×5	-	0
4	maxpool 1x20 max pooling with stride [1 10] and padding [0 0 0 0]	Max Pooling	1×3255×5	-	0
5	conv_2 10 1x30x5 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	1×3226×10	Weights 1×30×5×10 Bias 1×1×10	1510
6	relu_2 ReLU	ReLU	1×3226×10	-	0
7	dropout 40% dropout	Dropout	1×3226×10	-	0
8	fc_1 12000 fully connected layer	Fully Connected	1×1×12000	Weights 12000×32260 Bias 12000×1	387132000
9	relu_3 ReLU	ReLU	1×1×12000	-	0
10	fc_2 1024 fully connected layer	Fully Connected	1×1×1024	Weights 1024×12000 Bias 1024×1	12289024
11	regressionoutput Triplte loss layer	Regression Output	-	-	0

Figure 3.16. Triplet Loss NN design that contains 11 layers for Hybrid-VL$_{FL}$ method.

Triplet loss training procedure for Hybrid-VL$_{FL}$ and its implementation in this study are demonstrated in Figure 3.17 and summarized below:

- Establish Triplet Loss NN framework: Use the designed Triplet Loss NN (Figure 3.16) that takes 1D (32K) image semantic and rgb descriptors as an input and generates 1D (1024) hybrid location-aware learnt descriptor.

- Concatenate SD & LD representations: Note that here we used the SD (16K) and LD (16K - PCA reduction from 32K) image representations resulted from LD-VL with NetVLAD (Section 3.4.1) and SD-VL previously trained with triplet loss (Section 3.4.2). In order to manage feature-level fusion, we concatenates SD(16K) and LD(16K) image descriptor for each image in Train/Validation/Test sets. In this way we gain the hybrid representation (32K) form of an image in order to use it as an 1D input for our Triplet Loss NN.

- Prepare the triplets for training: The most critical phase of triplet loss training is choosing the right triplets (Anchor, Query, Negative) according the corresponding TNS values. This triplet generation is applied as same as in Learnt descriptor generation (Section 3.4) of Hybrid-VL$_{DL}$ method except for 'online-generation'. Here triplets are generated and stored 'offline' before training. This generation is performed on each queries of Train-Validation-Test sets, which computes the nearest (Eq. 3.2) positive and the 10 hard negatives (Eq. 3.5) for each query according to previously determined TNS threshold. Note that, we work on concatenated hybrid descriptors (32K) instead of images. To give an example, if we have 145 train queries we generate 4350 (145*3*10) training triplets with hard negative sampling (1A 1P 10N); then our final training matrix becomes [4350 x 32K].

- Take the training and validation sets: Because we have limited number of queries in Train sets of *RobotCar Seasons* and *Malaga Streetview Challenge* data sets, we added the hybrid descriptors (32K) of Validation set to the hybrid descriptors (32K) of training set. By the way, we could enrich our training sets for our Hybrid-VL$_{FL}$ method like that: $[((NumberOfTrainQueries * 3 * 10) + (NumberOfValidationQueries * 3 * 10))$ x 32K].

- Initialize the training: All queries in training set (concatenated Train and Validation) are in triplet forms (1A 1P 1N). The Triplet Loss (Eq. 3.4) is calculated and the backpropagation algorithm is performed for all the subsets of queries set.

- Extracting learnt hybrid descriptor: The trained model is employed on both the database and query images of test data set. As a result of this feature extraction, we gain the 1D (1024) hybrid location-aware learnt descriptors. Then, like using a SD or LD descriptor as being in Section 3.4.1 and Section 3.4.2, this newly proposed *learnt* representation is integrated with our ANN image matching method and named as Hybrid-VL$_{FL}$ method. As a result, pre-determined *k* number of nearest candidate images are automatically returned respectively for a given each test query image and accepted as a feature-level hybrid matching results.

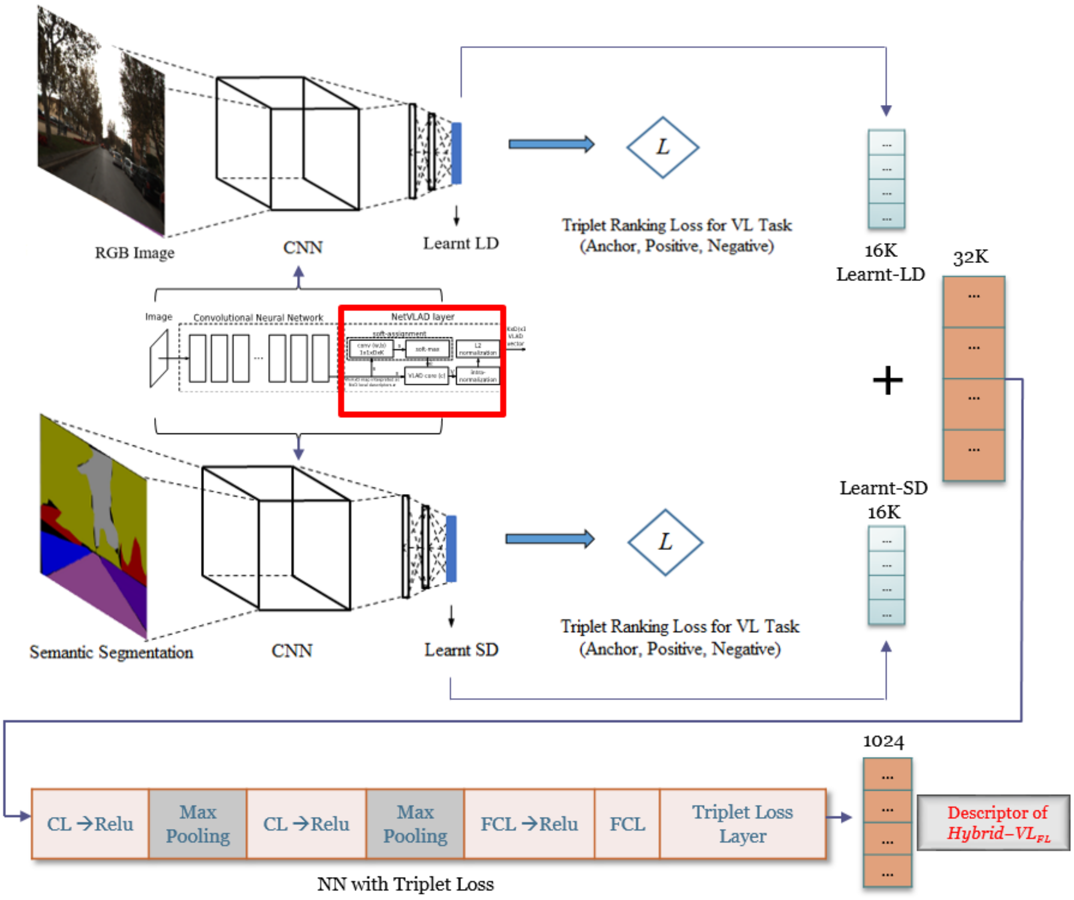


Figure 3.17. Illustration of triplet loss training procedure for Hybrid-VL$_{FL}$ and its implementation detail.

These training steps implemented and adjusted in this study were originally developed in Matlab Deep Learning Toolbox. Our trainings on our Triplet Loss NN have been done in GPU mode with the given configurations:

- Base model: Triplet Loss NN (Figure 3.16) ;
- Training Loss: NetVLAD (Arandjelovic et al., 2016) triplet ranking loss as defined in Equation 3.4 ;
- Layers Name: 2 CLs + 2 FCLs + Triplet Loss Layer;
- Trainable Layers: whole layers with nearly 400 hundred million total learnables;
- Optimizer: Stochastic Gradient Descent (SGD) ;

- Layers Weight initializer: Initialize the input weights with the ‘Glorot’ initializer (Glorot and Bengio, 2010);
- Learning rate: 0.01;
- Momentum: 0.99;
- Weight decay: 0.9;
- Margin: 0.2
- Number of Epochs: 15;
- Batch size: 10 tuples (each tuple contains the query, the positive and at most 10 negatives);
- TNS: 25 / 70 (w.r.t. data set) meter threshold for choosing negative samples;

The parameters above had already been examined to reach the best Triplet Loss training performance for the Hybrid-VL$_{FL}$ method and their best results are given in Section 4.5. Also TNS options were kept up same for each data set as they were used in Learnt descriptor generation (Section 3.4) of Hybrid-VL$_{DL}$ method.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, performance of the proposed Hybrid-VL methods are compared with the corresponding LD and SD based VL methods with several experiments. These case studies were carried out on different traversals of *RobotCar Seasons* and newly created streetview traversal of *Malaga Streetview Challenge* data sets.

First of all, initial promising decision-level hybrid result is demonstrated by means of the proposed hand crafted SD and LD VL methods (*non-learnt*) which are introduced in Section 3.3. After reaching the first promising hybrid results with these early stage experiments, we displayed the necessity of a location-aware descriptor based VL methods with the following experiments. Consequently, decision-level hybridization success of *learnt* descriptor based $SD\text{-}VL_{Learnt}$ and $LD\text{-}VL_{NetVLAD}$ methods (Section 3.4) is examined on the each data sets. Additionally, feature-level hybridization results for *learnt* descriptors also examined and compared under the same conditions with decision-level one. Note that, every individual Hybrid-VL method in this chapter incorporated different type of SD and LD descriptors for k=10 candidates while repeated the different types of hybridization approach proposed in Section 3.5.1 and Section 3.5.2. Configuration details of these Hybrid methods, from feature extraction to used data set, is given where they are performed in the following sections.

Also, before considering on experimental studies, we will give a brief explanation on used software and equipment. In this study, all implementations were conducted with MATLAB R2019b in Ubuntu operating system. In addition for enhancing the running time performance, MATLAB allows us to rewrite the any time consuming part of our MATLAB methods with using C / C++ codes. From the side of used equipment, our CNN based models were trained on NVIDIA™ Titan XP with 12 GB of memory and its higher compute capability which is also supported with 16 GB Random Access Memory.

4.1. Evaluation Metrics

In this study, GPS based metric error computation was applied in order to evaluate the performance of SD-VL, LD-VL and Hybrid-VL methods. It is already described, each database and query image is associated with an accurate GPS position (Latitude Longitude in decimal degree), which is in WGS84 geographic coordinate system. It is known Google Maps also works with WGS84 geodetic datum. The distance between two locations were computed using Haversine formula (Inman, 1849; Veness, 2002) that returns the distance in meter. However, just using GPS coordinates of the query image as ground truth to measure localization accuracy will be unreliable. Because this type of a error computation is *not fairly penalized*; similar mismatching cases will cause the very different meter error. Therefore we followed the standard place recognition evaluation procedure (Arandjelovic et al., 2016; Arandjelović and Zisserman, 2014; Germain et al., 2018; Piasco et al., 2019; Sattler et al., 2012; Torii et al., 2015). The query image is accepted correctly localized if at least one of the top *N* retrieved database images is within a given *D* meters threshold (radius) from the ground truth position of the query. The percentage of correctly localized queries (Recall) is then plotted versus different values of *N* and *D*, these two type of evaluation metrics are explained below:

- ***Recall @N***: Percentage of well localized queries is plotted with respect to the *N* number of returned candidates. A query is considered well localized if one of the top *N* retrieved images lies inside the 25m radius of the ground truth query position.
- ***Top-1 recall @D***: Distance between the top ranked (1_{st}) returned database image position and the query ground truth position is calculated. Then the percentage of queries with distances less than a fix threshold *D* (changing from 5 to 150 meter) is plotted like in the related works.

4.2. Hybrid-VL$_{DL}$ with Non-Learnt Descriptors

Most successful appearance based localization methods typically rely on a large database of views represented with image descriptors and struggle to retrieve the views of the same location. The quality of the results is often affected by the density of the reference views (database images) and the robustness of the image representation with respect to viewpoint variations, clutter and seasonal changes.

First of all, we cut in half the *Malaga Downtown* (31443) (Section 3.1) and used half of them as a query set (15721) and rest of them as a database images. When a basic image retrieval task is employed by $LD\text{-}VL_{SURF}$ method ($k = 1$) on these reduced new sets, we observed that there is no mismatching case. In other words, even this hand crafted VL system finds the best matching images for a given queries. This fact actually results from the trade-off between the density of this data set (high rate with 20fps) and VL performance. Therefore we need more sparse data set to reflect real life cases VL task, because high rate sequential images are not available in normal life. In order to make the right reduction of the density in the *Malaga Downtown* data set, we generated a basic case study.

In this case initially, every sequential 20_{th} images were picked up which provides us nearly 5 meter distance between sequential images. This decreased sub set was named as *Malaga Downtown Base* data set that contains 1571 images. Then two sub sets (database-query) are derived by reducing the image number again. We selected the every sequential 3_{rd} image as a query and named it as Set1 (1048 database images- 523 queries), further reduced the Set1 by discarding one of each two images which is named as Set2 (524 database images- 523 queries). When we implemented $LD\text{-}VL_{SURF}$ method (k=1) on these sets for image retrieval task, our hand crafted method starts to miss best matching images as depicted in Figure 4.1. In this figure, mismatching image samples are directly penalized with their distance to query images in meter, in order to visualized the matching case in descending order. It is clearly seen, more sparser data set (Set2) inevitably gives rise to poor VL performance with two times more mismatching images against the more denser one (Set1).

Under the light of this case study, in order to observe the success of our methods

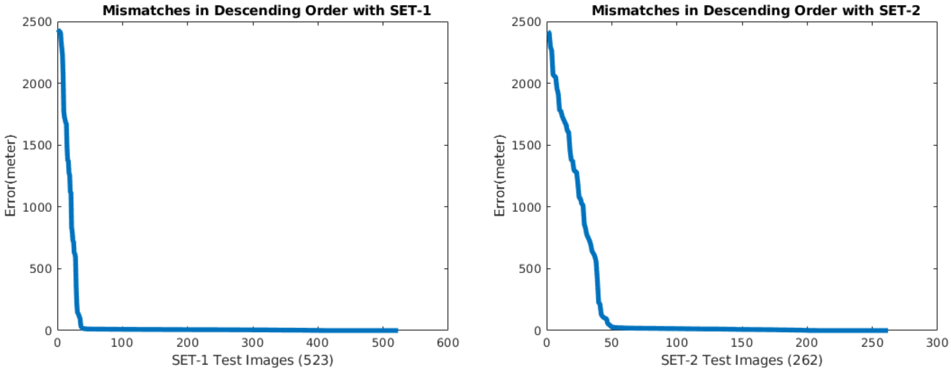


Figure 4.1. Mismatching image samples in descending order according to GPS error with implementation of $LD\text{-}VL_{SURF}$ method ($k = 1$) on Set1 and Set2.

against sparsely provided reference images, we generated our *Malaga Streetview Challenge* data set as described in 3.1. In this data set the *Malaga Downtown Base* set is accepted as database images (1571) while collected *Google Streetview* set is accepted as query images (436). This newly generated challenging set serves us not only a sparser images with nearly 5 meter interval between sequential images by comparison with *RobotCar Seasons*(nearly 1.5 meter intervals), but also short-long term changes (Table 3.1).

In order to examine our *non-learnt* Hybrid-VL$_{DL}$ methods on these data sets we employed the previously described (Section 3.3) hand-crafted descriptor based VL methods. Note that, we had already demonstrated the success of our initial Hybrid-VL$_{DL}$ method (k=10) in our previous study (Çinaroğlu and Baştanlar, 2019) which incorporates the $LD\text{-}VL_{SURF}$ and $SD\text{-}VL_{LIBLABEL}$ methods and examined on *Malaga Streetview Challenge* data set. Next, on the purpose of improving the same hybrid method, differently from our previous study we employed the proposed $SD\text{-}VL_{PretSegNet}$ methods instead of using $SD\text{-}VL_{LIBLABEL}$ and visualized the result in Figure 4.2. Result with *Top-1 recall @D* metric in this figure support our Hybrid-VL$_{DL}$ concept with its superiority against the others, but it is clear we should need to employ more robust VL methods from the side of each SD-VL and LD-VL. Because, they hardly achieved correct localization for few query images (*Top-1 recall@5* performance under 0.05) in every distance threshold (5m-150m) as depicted in this figure (Fig. 4.2).

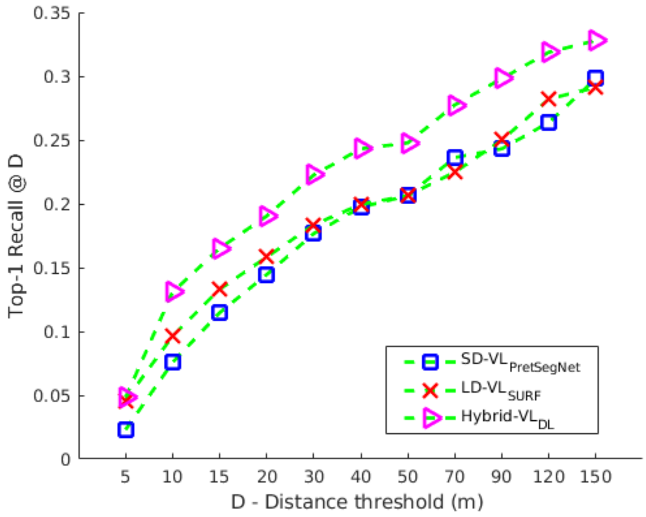


Figure 4.2. Superiority of Hybrid-VL$_{DL}$ methods (*Top-1 recall@D*) against the other methods on Malaga Streetview Challenge data set (436) with $SD\text{-}VL_{PretSegNet}$ and $LD\text{-}VL_{SURF}$ methods.

4.3. Hybrid-VL$_{DL}$ with Learnt LD-VL and Nonlearnt SD-VL

Results in the previous section shows that, we must improve the performance of both SD-VL and LD-VL methods, in addition we also need to evaluate this improvement on an another state-of-the-art data set. Therefore we preferred to used *RobotCar Seasons* data set which is especially designed for VL challenge with its traversals that represent different conditional changes with a low variance in viewpoint. The reason of this preference among the many other VL data sets is already described detailed in Section 2.3.

After this preference, we firstly concentrated on improving the performance of LD-VL method because of the poor performance (Fig. 4.2) of $LD\text{-}VL_{SURF}$ on Malaga Streetview Challenge data set. On behalf of this aim, we preferred to construct our baseline LD-VL method on a *learnt* descriptor which was trained on a CNN with NetVLAD based triplet ranking loss for the purpose of localization task. Then we incorporated this extracted learnt LD with our ANNS method which is named as $LD\text{-}VL_{NetVLAD}$. Advantages of using NetVLAD based triplet ranking loss had already been demonstrated in Section 2.1 with various recent studies, after all its popularity definitely results form its grand success against changing environmental conditions. While training our CNN network with triplet ranking loss for $LD\text{-}VL_{NetVLAD}$ method we followed the same steps with the same configurations as represented in the Section 3.4. Nevertheless, there are also key parameters that we must underline because some of them applied simultaneously to each data set while some of them will change according to used data set.

Whole these learning was examined on separated parts as described in Section 3.1, which were created by dividing the prior map into geographically disjoint distinct 3 parts as training, validation and testing set for both data sets. Inequality of this separation will inevitably affect the performance of $LD\text{-}VL_{NetVLAD}$ method, therefore at least 2 different division is generated for both data sets experiments. Then the average performance of these divisions are accepted as a final result of $LD\text{-}VL_{NetVLAD}$ method. Also note that, even faraway images can visualize the same scene with same objects. For example, the Izmir Clock Tower can be visible from many faraway locations in Izmir. Hence for the purpose of topometric localization task, we considered in this study such image pairs as negative examples because they actually are not captured from the same localization. This distinction is achieved owing to employing suitable TNS threshold which is 25 meter for *RobotCar Seasons* data set and 70 meter for the *Malaga Streetview Challenge* data set during their training processes. In order to be fair while training with both data sets, the higher TNS threshold for the second data set is implemented because of its sparse structure of database images (1571) with respect to first data set (6954). In other words, on roads of the nearly same length (8km versus 10km) different numbers of image collections cause a different sequential image intervals (1.5m versus 5m respectively) for each data sets. As a result of this fact, all the *best models* for each data set is achieved with these determined TNS thresholds. Furthermore, as it is given in Table 2.1 both of the *Malaga Streetview Challenge* and *RobotCar Seasons* data sets have less image collections in comparison with other examined data sets. Because, these data sets especially reduced carefully from their extended versions (Section 3.1) in purpose of making them more suitable for VL task. As a result of these reduced image numbers for both data sets, we observed that selection of different base model (AlexNet & VGG16) for NetVLAD architecture directly influences the success of proposed VL methods in our experiments. AlexNet (16K dimensional feature vector) based training provided a better trained model rather than VGG16 (32K dimensional feature vector) for the each data sets. Because AlexNet is more suitable for training on narrow data sets with its less deeper architecture, this important preference is also underlined in many other touchstone studies (Azizpour et al., 2015; Babenko et al., 2014). Therefore all the *learnt* descriptor based VL's in the following experiments are trained on AlexNet base model regardless of the used data set.

Under the light of these given configuration, we firstly examined our new Hybrid-VL$_{DL}$ method on the '*Sun*' traversal (Table 3.1) of *RobotCar Seasons* data set with incorporation of $LD\text{-}VL_{NetVLAD}$ and $SD\text{-}VL_{PretSegNet}$ methods. We divided it (6954 database images-460 queries) into geographically disjoint distinct 3 parts like that, training set (2318 database images - 180 queries), validation set (2318 database images - 142 queries) and test set (2318 database images - 138 queries) then examined both SD and LD VL methods on the same divisions. Result of this Hybrid-VL$_{DL}$ method (Figure 4.3) displays that, on this new benchmark data set we achieved to increase performance of our method with the *learnt* $LD\text{-}VL_{NetVLAD}$ against to $SD\text{-}VL_{PretSegNet}$. This improvement results in performance gap between these two SD-VL and LD-VL method increases by nearly 0.35 *Top-1 recall@5* value. As a fact of decision-level hybridization, also the poor performance of SD-VL method pulls down the performance of Hybrid-VL$_{DL}$ method.

In order to explore the reason of this poor localization performance of SD-VL method, we employed the same $SD\text{-}VL_{PretSegNet}$ method on different traversals of *RobotCar Seasons* data set as depicted in Figure 4.4. At the same time, we also examined the possible effect of different horizon level on our pre-trained (CamVid) SD-VL method in this figure. In line of this purpose, we generated the adjusted version of each traversals, by resembling their horizon level to CamVid data set images. Figure 4.4 shows that, different traversal with different conditions influences the performance of the SD-VL method because of the environmental changes (e.g. shining distortion). In the line of these results, we decided on using the *Overcast Winter* traversal for our next experiments because of its mid-level performance (magenta square in Figure 4.4). We

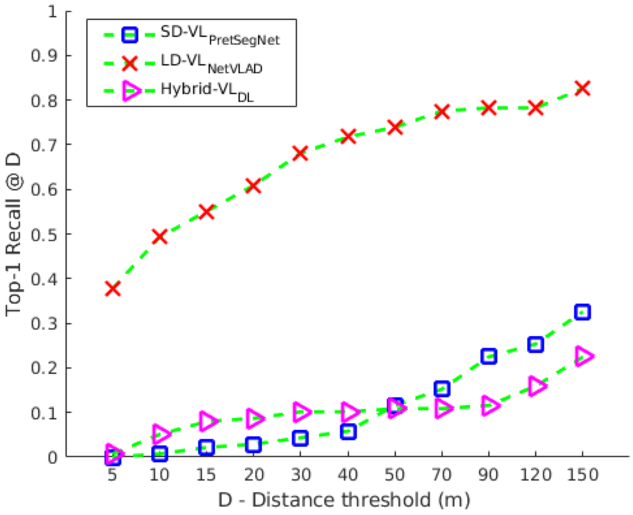


Figure 4.3. Improved *Top-1 recall@D* performance of LD-VL method (LD-$VL_{NetVLAD}$) against the SD-VL method (SD-$VL_{PretSegNet}$) on '*Sun*' (138) traversal of *RobotCar Seasons* data set.

also see that SD-VL method on adjusted version of each traversals (yellow lines in Fig. 4.4) outperforms their not-adjusted versions (black lines) by near 0.2 *Top-1 recall@5*. As a result of this effect, we learned that horizon level differences between the training set of trained model SD-$VL_{PretSegNet}$ and test set causes a poor localization performance. Therefore we used horizon-level-adjusted images in our future experiments for each SD and LD VL methods.

Obtaining a improved LD-VL method is not enough for our Hybrid-VL$_{DL}$ method as it is shown in Figure 4.3, in other words SD-$VL_{PretSegNet}$ method becomes insufficient against learnt-LD methods. Hence, we were in hopes of increasing the performance of SD-VL method by contribution of a more powerful semantic segmentation method. This expectation led us employing more powerful semantic segmentation model DeepLabv3+ as a state of the art semantic segmentation method, which was invented by *Google* and meets with our expectation. Thus, instead of using its pretrained version ('*Pretrained DeepLabv3+*') that had already been trained on Camvid data set, we re-trained this model on our RobotCar Seasons data set. On the other side, we had already observe the negative influence of difference in horizon level between the

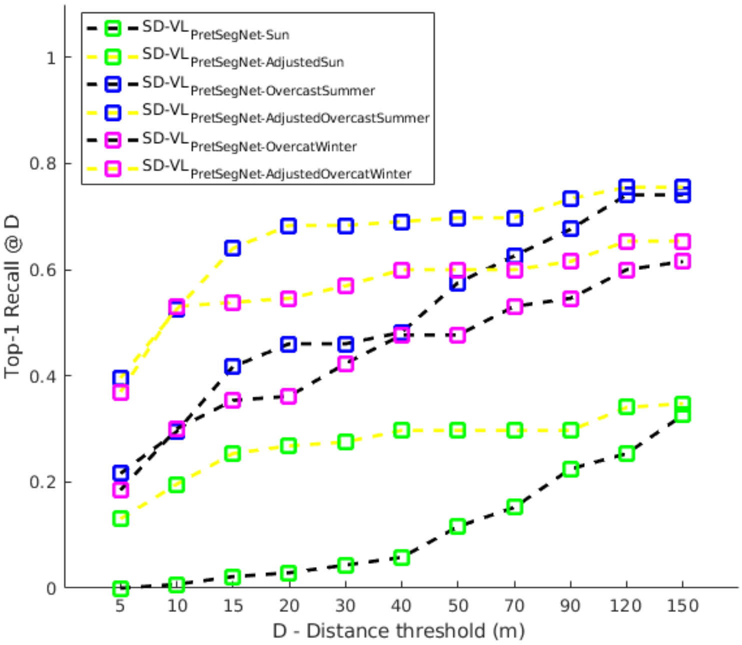


Figure 4.4. *Top-1 recall@D* performance comparison of different horizon level impact on $SD\text{-}VL_{PretSegNet}$ method according to different traversals (Sun,Overcast-Summer, Overcast-Winter) of RobotCar Seasons. Yellow lines denote the horizon level '*adjusted*' version of each traversals.

database and query images as depicted in Figure 4.4 on a different query traversal of RobotCar Seasons data set. Therefore we re-trained DeepLabv3+ model on concatenated collection of CamVid (horizon level of CamVid resembling to horizon level of RobotCar Seasons) and RobotCar Seasons data set. In Section 3.2, detailed re-training steps of DeepLabv3+ is described, and superior performance of '*DeepLabV3+ Retrained-2*' in semantic segmentation is depicted in Figure 3.8 and Figure 3.9. And this superior segmentation model based *non-learnt* VL method is named as $SD\text{-}VL_{DeepLabV3+\_Retr2}$.

After all, we evaluated our new Hybrid-VL$_{DL}$ method on the *Overcast-Winter* traversal (Table 3.1) of *RobotCar Seasons* data set with incorporation of $LD\text{-}VL_{NetVLAD}$ and $SD\text{-}VL_{DeepLabV3+\_Retr2}$ methods. Note that, we applied the same division criteria on *Overcast-Winter* (6954 database images-460 queries) data set as implemented on *Sun* traversal then trained the $LD\text{-}VL_{NetVLAD}$ method on training and validation sets. Result of the Hybrid-VL$_{DL}$ method on *Overcast-Winter* test set (130 images) is displayed in Figure 4.5 (lower one). In order to visualize the impact of our powerful '*DeepLabV3+ Retrained-2*' semantic segmentation model on VL, we generated another Hybrid-VL$_{DL}$ method (left side one in Figure 4.5) that combines the same $LD\text{-}VL_{NetVLAD}$ with $SD\text{-}VL_{PretSegNet}$ again examined on the same division of *Overcast-Winter* traversal.

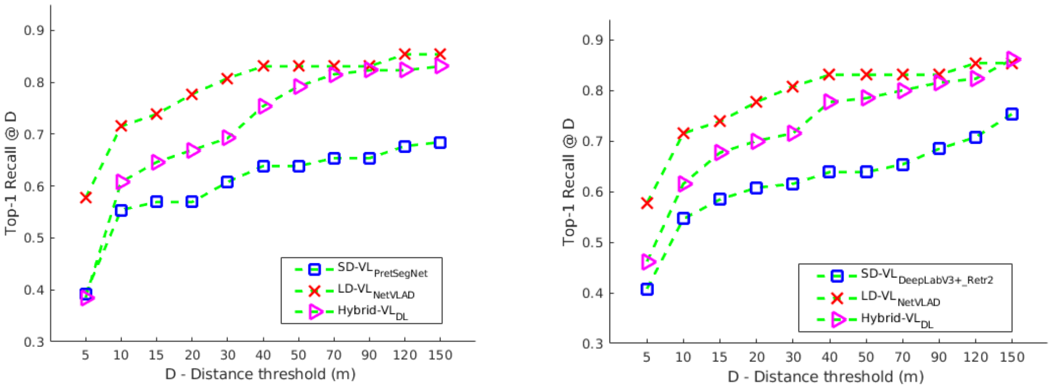


Figure 4.5. Limited contribution of the robust segmentation model based $SD\text{-}VL_{DeepLabV3+\_Retr2}$ method (right one) on Hybrid-VL$_{DL}$ regarding to $SD\text{-}VL_{PretSegNet}$ method. Both hybrid methods are obtained with the same $LD\text{-}VL_{NetVLAD}$ method on *Overcast-Winter* traversal of *RobotCar Seasons* data set.

It is clear that in Figure 4.5, there is a little development in SD-VL performance (0.02 *Top-1 recall@5*) between these two sub-figures which correspond to the $SD\text{-}VL_{PretSegNet}$ (blue squares on left image) and $SD\text{-}VL_{DeepLabV3+\_Retr2}$ (blue squares on right image) methods. This comparative result between the left and right sub figures shows that, segmentation success of '*DeepLabV3+ Retrained-2*' is not reflected to the VL success as much as we expected. In other words, expected improvement in *non-learnt* SD-VL was not achieved by just changing our segmentation model with a robust one. This ineffectiveness will be explained with this reason, both of the segmentation model repeats the same poor segmentation performance simultaneously for the same references and query images.

4.4. Hybrid-VL$_{DL}$ with Learnt SD-VL

Inadequacy of using a '*non-learnt*' SD-VL methods against the *learnt* $LD\text{-}VL_{NetVLAD}$ method is demonstrated in previous section. In order to cope with this inability, the same triplet ranking loss was implemented on semantically segmented versions of images (database-query sets) belonging to *RobotCar Seasons* and *Malaga Streetview Challenge* data sets as described in Section 3.4.2 detailed. By this way we obtained our novel SD-VL called $SD\text{-}VL_{Learnt}$ that is based on the images semantically segmented by '*DeepLabV3+ Retrained-2*' model. During the training phase of $SD\text{-}VL_{Learnt}$ method, the same configuration of $LD\text{-}VL_{NetVLAD}$ method are implemented to be fair against this method. In a word, best trained models were achieved again with the same base model (AlexNet with 16K dimensional feature vectors), TNS threshold (25 meter for *RobotCar Seasons* and 70 meter for the *Malaga Streetview Challenge*) and other configurations described as in previous section. Note that, we used the same geographically disjoint divisions of both data sets for each of the VL methods.

After all, we examined our baseline Hybrid-VL$_{DL}$ method on the *Overcast-Winter* traversal of RobotCar Seasons (145 Train, 113 Validation, 130 Test queries) and Malaga Streetview Challenge (249 Train, 78 Validation, 111 Test queries) data sets with incorporation of $LD\text{-}VL_{NetVLAD}$ and $SD\text{-}VL_{Learnt}$ methods. And superiority of proposed Hybrid-VL$_{DL}$ method is evaluated with previously defined evaluation metrics (*Top-1 recall@D*, *Recall @N*) in Figure 4.6 and Figure 4.7 respectively.

From the side of *Top-1 recall@D* evaluation metric, proposed decision-level hybrid method is able to increase *Top-1 recall@5* localization performance against the $LD\text{-}VL_{NetVLAD}$ method by 11.6% and 4.5% on the both Overcast-Winter traversal of RobotCar Seasons (left one) and Malaga Streetview Challenge (right one) data sets respectively.

In addition, performance of the same proposed Hybrid-VL$_{DL}$ is evaluated with *Recall @N* metric in Figure 4.7. Again, our approach achieved the best performances for D=25m from the side of both data sets. Proposed hybrid methods is able to increase *Recall @1* localization performance against the $LD\text{-}VL_{NetVLAD}$ method by 4% and

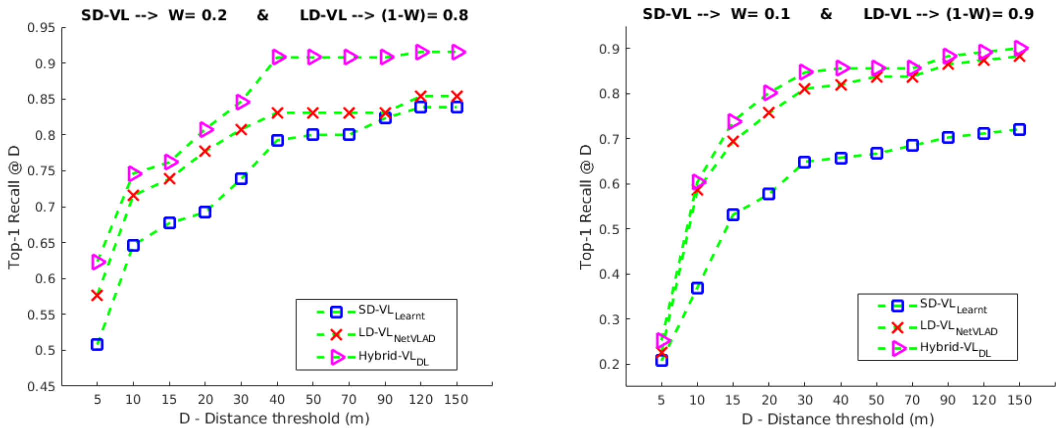


Figure 4.6. Superiority of proposed Hybrid-VL$_{DL}$ method that incorporates LD-$VL_{NetVLAD}$ and SD-VL_{Learnt} methods. Results represented with *Top-1 recall@D* evaluation metric on the *Overcast-Winter* traversal (left one) and *Malaga Streetview Challenge* (right one).

5.4% on the both Overcast-Winter traversal of RobotCar Seasons (left one) and Malaga Streetview Challenge (right one) data sets respectively.

Also I should underline that, fine-tuning the decision level hybridization by *W* hyper-parameter is very important point. We should trust on the better VL method among the SD-VL and LD-VL methods. The best results in Figure 4.6 and Figure 4.7 support this inference with lower *W* values; *W* = 0.2 for *RobotCar Seasons* and *W* = 0.1 for *Malaga Streetview Challenge* data set. We know that lower *W* value increases the importance of LD-VL method in our decision-level hybridization. Under the line of these outcomes we can conclude that, if we trust our LD-VL method more than SD-VL we can reach the best Hybrid-VL$_{DL}$ results for both data sets. Results for varying values of *W* are demonstrated in Section 4.6.

To sum up, experimental results indicate that the performance of the proposed Hybrid-VL$_{DL}$ method is superior against the state-of-the-art baseline LD-VL method on both examined data sets with respect to each evaluation metric. This performance improvement is achieved owing to incorporating the distinguishing power of the relative positions of the objects in a semantically segmented image with power of location-aware

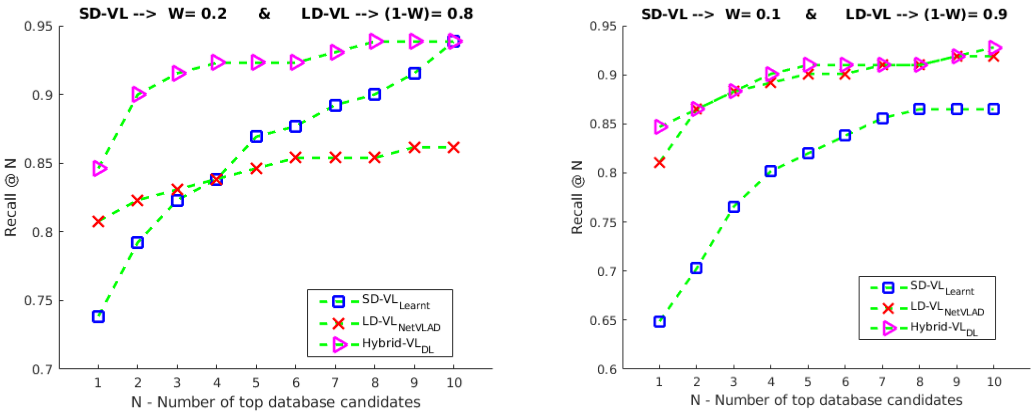


Figure 4.7. Superiority of proposed Hybrid-VL$_{DL}$ method that incorporates LD-$VL_{NetVLAD}$ and SD-VL_{Learnt} methods. Results represented with *Recall @N* evaluation metric on the *Overcast-Winter* traversal (left one) and *Malaga Streetview Challenge* (right one).

triplet ranking loss training. And remember that, fine-tuned-W parameter contributes this achievement directly.

Furthermore, contribution of the SD-VL_{Learnt} method to the Hybrid-VL$_{DL}$ is also demonstrated in Figure 4.8 with sample cases in which LD-$VL_{NetVLAD}$ method can not retrieve the correct images. Also sample cases where the Hybrid-VL$_{DL}$ fails but the LD-$VL_{NetVLAD}$ does correctly localize are illustrated in Figure 4.9. These image-pair based results in Figure 4.8 make us think that the proposed Hybrid-VL$_{DL}$ method is more successful while localizing in contrast (database vs query images) weather scenarios. To investigate the impact of localizing in contrast weather conditions, same *Malaga Streetview Challenge* 111 test queries were separated into two subsets according to lighting condition in images. 24 relatively sunny images are collected and named as 'Sunny Subset' and rest of the relatively darker images are named as 'Overcast Subset'. Also we should remember that, *Malaga Streetview Challenge* database images (nearly 523 images) had already been captured under overcast weather condition. After all, we examined our baseline Hybrid-VL$_{DL}$ method on the *Sunny* (24 test queries) and *Overcast* (87 test queries) subsets of *Malaga Streetview Challenge* against its *Overcast* database images with incorporation of LD-$VL_{NetVLAD}$ and SD-VL_{Learnt} methods.

And superiority of proposed Hybrid-VL$_{DL}$ method is depicted with the *Top-1 recall@D* evaluation metric in Figure 4.10. But more important than that for this experiment, results in this Figure 4.10 support our initial opinion on localizing in contrast weather scenarios. To explain further, although there are limited number of images in the *Sunny* subset (right one in Figure 4.10) the $SD\text{-}VL_{Learnt}$ method (blue square) has given better result than the $LD\text{-}VL_{NetVLAD}$ method (red cross). The point to be noted here, while the Hybrid-VL$_{DL}$ method on the *Overcast* subset (left one in Figure 4.10) owes its success to the $LD\text{-}VL_{NetVLAD}$ method (W=0.1), on the other hand Hybrid-VL$_{DL}$ method on the *Sunny* subset (right one in Figure 4.10) owes its success to the $SD\text{-}VL_{Learnt}$ method (W=0.9).

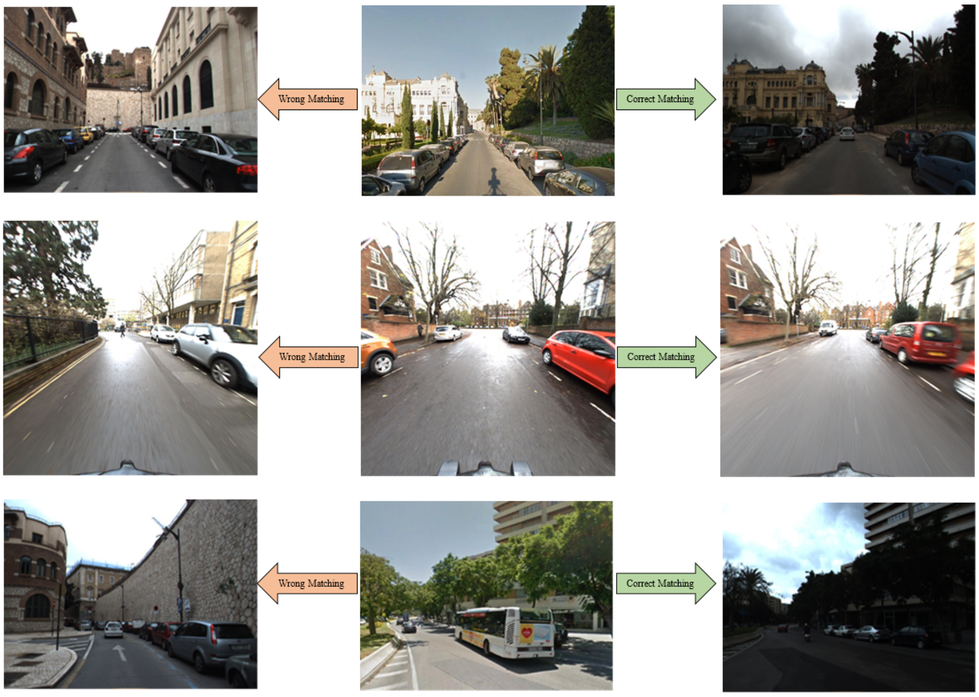


Figure 4.8. Superiority of proposed Hybrid-VL$_{DL}$ method with three sample localization cases from both data sets. RGB image based method $LD\text{-}VL_{NetVLAD}$ (left) fails but Hybrid-VL$_{DL}$ (right) accomplishes for a given query (middle)

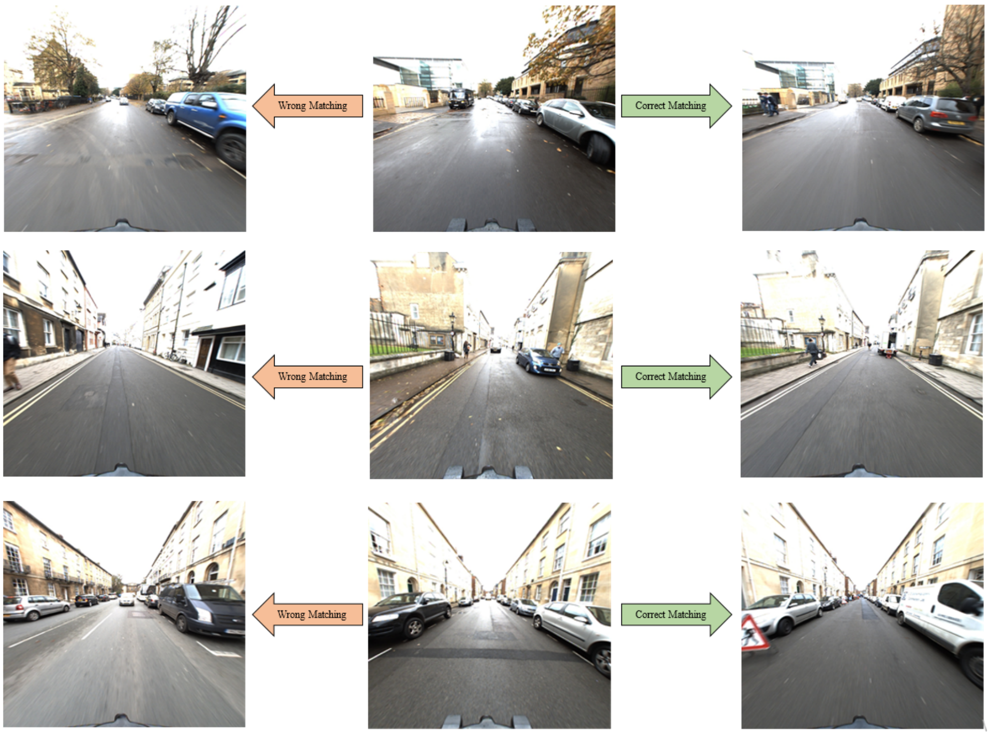


Figure 4.9. Sample three cases where the Hybrid-VL$_{DL}$ fails but the $LD\text{-}VL_{NetVLAD}$ does correctly localize. Hybrid-VL$_{DL}$ (left) fails but $LD\text{-}VL_{NetVLAD}$ (right) accomplishes for a given query (middle)

4.5. Hybrid-VL$_{FL}$ with Learnt SD-VL

All the experimental hybrid results demonstrated until this section are implemented by Hybrid-VL$_{DL}$ method which depend on W hyper-parameter. In this section, results of the proposed Hybrid-VL$_{FL}$ method (described in Section 3.5.2) that produces automatically tuned hybrid results are given for the both data sets.

I should note that, all these experiments were carried out with the same SD (16K) and LD (16K - PCA reduction from 32K) learnt-descriptors which were also combined in the Hybrid-VL$_{DL}$ method in the previous section experiments. Also the same TNS thresholds (25 meter for *RobotCar Seasons* and 70 meter for the *Malaga Streetview Challenge*) were applied while training our Triplet Loss NN, and experiments were conducted on the same partition of both data sets with previous section. We provided the same conditions with Hybrid-VL$_{DL}$ method, because we were looking for that whether

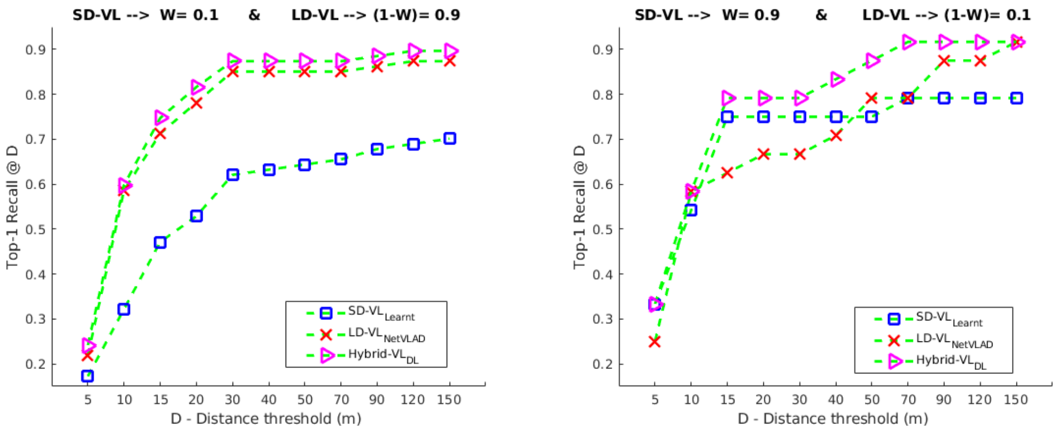


Figure 4.10. Impact of localizing in contrast weather conditions for the Hybrid-VL$_{DL}$ method on the subsets of *Malaga Streetview Challenge*. Results are represented with *Top-1 recall@D* evaluation metric on the *Overcast* (87 test queries) subset (left one) and the *Sunny* (24 test queries) subset(right one).

Hybrid-VL$_{FL}$ method achieves the same or better performance without any manual hyper-parameter tuning. Furthermore, we stuck to all implementation steps and settings described in Section 3.5.2.

After all, we examined our baseline Hybrid-VL$_{FL}$ method on the Overcast-Winter traversal of RobotCar Seasons (145 Train, 113 Validation, 130 Test queries) and Malaga Streetview Challenge (249 Train, 78 Validation, 111 Test queries) data sets with incorporation of $LD\text{-}VL_{NetVLAD}$ and $SD\text{-}VL_{Learnt}$ methods. Hence as it is described in Section 3.5.2, best models were trained on these training triplets generated with hard negative sampling (1A, 1P, 10); [7740 x 32K] ([$((145 \ast 3 \ast 10) + (113 \ast 3 \ast 10))$ x 32K]) and [9810 x 32K] ([$((249 \ast 3 \ast 10) + (78 \ast 3 \ast 10))$ x 32K]) for our RobotCar and Malaga data sets respectively. And comparison of proposed Hybrid-VL$_{FL}$ method with Hybrid-VL$_{DL}$ is carried out with previously defined evaluation metrics (*Top-1 recall@D*, *Recall @N*) in Figure 4.11 and Figure 4.12 respectively.

With *Top-1 recall@D* and *Recall @N* (D=25m) evaluation metrics, Hybrid-VL$_{FL}$ (black circle) method displays slightly poorer performance in comparison with Hybrid-

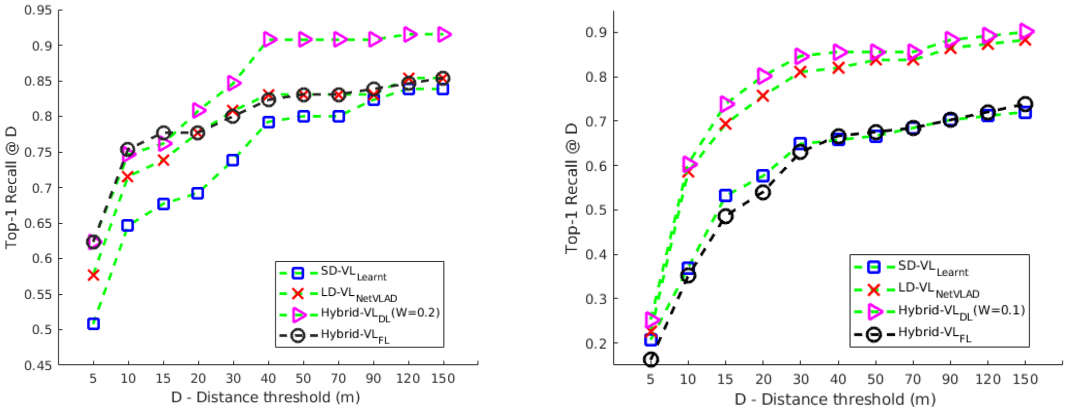


Figure 4.11. Performance comparison of proposed Hybrid-VL$_{FL}$ method against the tuned-with-W-parameter Hybrid-VL$_{DL}$ one. $LD\text{-}VL_{NetVLAD}$ and $SD\text{-}VL_{Learnt}$ methods are incorporated and results are represented with *Top-1 recall@D* evaluation metric on the *Overcast-Winter* traversal (left one) and Malaga Streetview Challenge (right one).

VL$_{DL}$ (magenta triangle) on Overcast-Winter traversal of RobotCar Seasons (left sides of each figures). However we can see that Hybrid-VL$_{FL}$ method displays significantly worse performance on Malaga Streetview Challenge (right sides of each figures) when compared to Hybrid-VL$_{DL}$. Poor performance of Hybrid-VL$_{FL}$ on Malaga Streetview Challenge can be explained with its sparse database images (1571 images for 8km trajectory) with respect to RobotCar Seasons data set(6954 images for 10km trajectory). Collecting a denser database images on the same path may increase the performance of Hybrid-VL$_{FL}$ on Malaga Streetview Challenge.

To sum up, very close performance of automatically tuned Hybrid-VL$_{FL}$ method (especially on Overcast-Winter traversal of RobotCar Seasons) compared to the Hybrid-VL$_{DL}$ method supports the reliability of the manually tuned-with-W-parameter Hybrid-VL$_{DL}$ methodology.

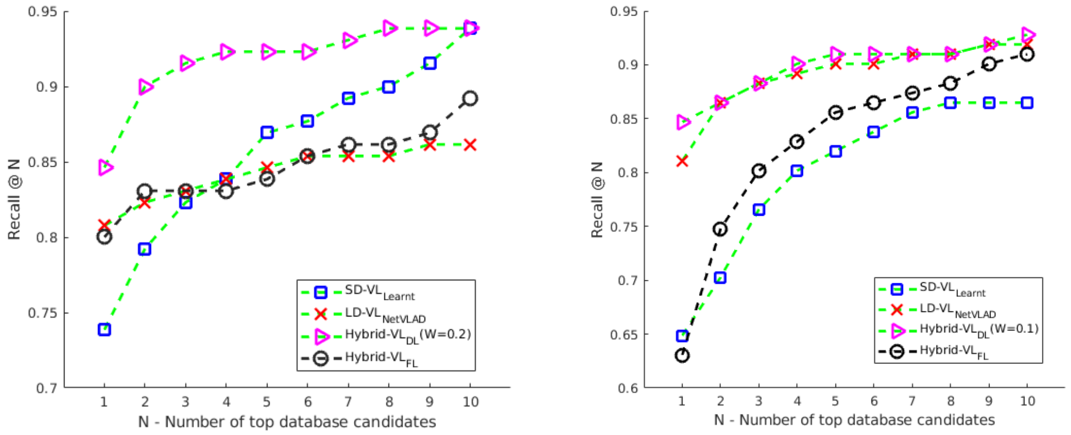


Figure 4.12. Performance comparison of proposed Hybrid-VL$_{FL}$ method against the tuned-with-W-parameter Hybrid-VL$_{DL}$ one. LD-$VL_{NetVLAD}$ and SD-VL_{Learnt} methods are incorporated and results are represented with *Recall @N* evaluation metric on the *Overcast-Winter* traversal (left one) and Malaga Streetview Challenge (right one).

4.6. Ablation Study

The results given in the previous subsections are the best ones obtained among the numerous trials with different experimental settings. In this section some important empirical results are demonstrated in order to convey the sense of what and why we choose these settings. At the same time, inferences are presented that may be useful for those who will repeat similar experiments.

We know that Hybrid-VL$_{DL}$ method is based on *W* hyper parameter and impact of this parameter is examined in this paragraph. Implementation detail of Hybrid-VL$_{DL}$ and its tuned-with-W-parameter results are given in Section 3.5.1 and Section 4.4 respectively. Thanks to the ’W’ parameters we are able to tune the contributions of SD-VL and LD-VL in the decision-level hybridization. And importance of fine-tuning with ’*W*’ hyper- parameter had already been underlined in previous sections. Logically, we should trust on the better VL method among the SD-VL and LD-VL methods. This inference is approved in Figure 4.13 and Figure 4.14 with varying ‘*W*’ values for each data set. The best Hybrid-VL$_{DL}$ results are acquired by increasing the importance of LD-VL method for both *Overcast-Winter traversal of RobotCar Seasons* (W=0.2) and *Malaga Streetview Challenge* (W=0.1) data sets as we expected. When we trust more on SD-VL with higher *W* value, performance of Hybrid-VL$_{DL}$ method decreases.

In Section 3.5.2 we proposed the Hybrid-VL$_{FL}$ method in order to obtain better or same result with the Hybrid-VL$_{DL}$ method. However, results in Section 4.5 show up the poor performance of this feature-level fusion method on both of the data sets (very close performance for *Overcast-Winter traversal of RobotCar Seasons* and worse performance for *Malaga Streetview Challenge*) with both evaluation metrics. Therefore, we wondered if we could increase the success of this hybrid method. In accordance with this purpose, we redesigned our Triplet Loss NN layers which takes 4K (1D) sized descriptor instead 32K and again outputs 1024 (1D) sized representation. So that before training our Hybrid-VL$_{FL}$ method, feature dimension of training and test sets are decreased with Principal Component Analysis (PCA). Impact of the dimensional reduction on Hybrid-VL$_{FL}$ is demonstrated for both data sets with respect to *Top-1 recall@D* and *Recall @N* evaluation metrics in Figure 4.15 and Figure 4.16 respectively. These results clearly figure out that dimensional reduction (from 32K to 4K) decreases the success of Hybrid-VL$_{FL}$ from the side of both metrics.

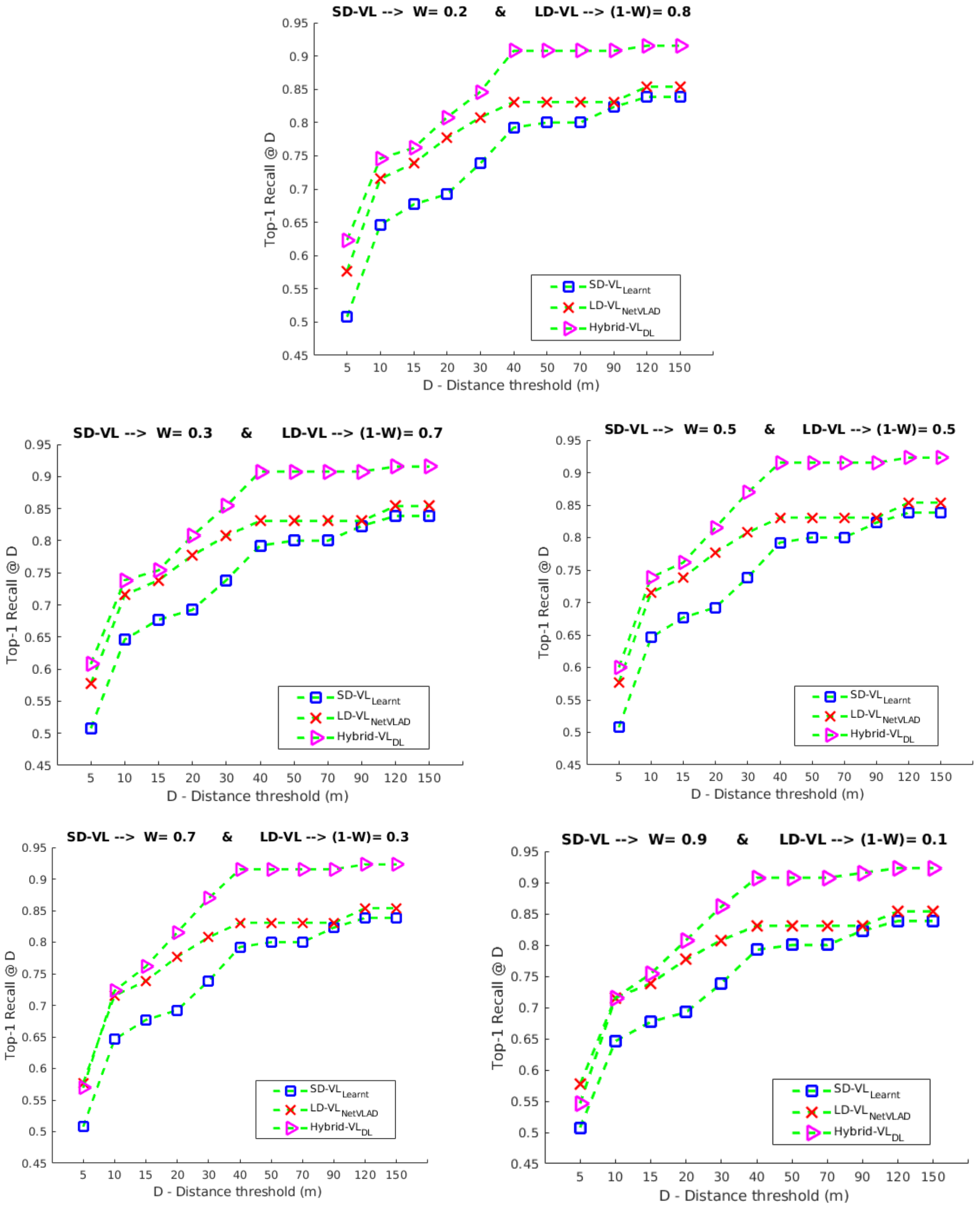


Figure 4.13. Importance of fine-tuning in Hybrid-VL$_{DL}$ method on *Overcast-Winter traversal of RobotCar Seasons* data set with varying W hyper-parameters. Best hybridization result (top row one) is gained with higher contribution of LD-VL (W=0.2).

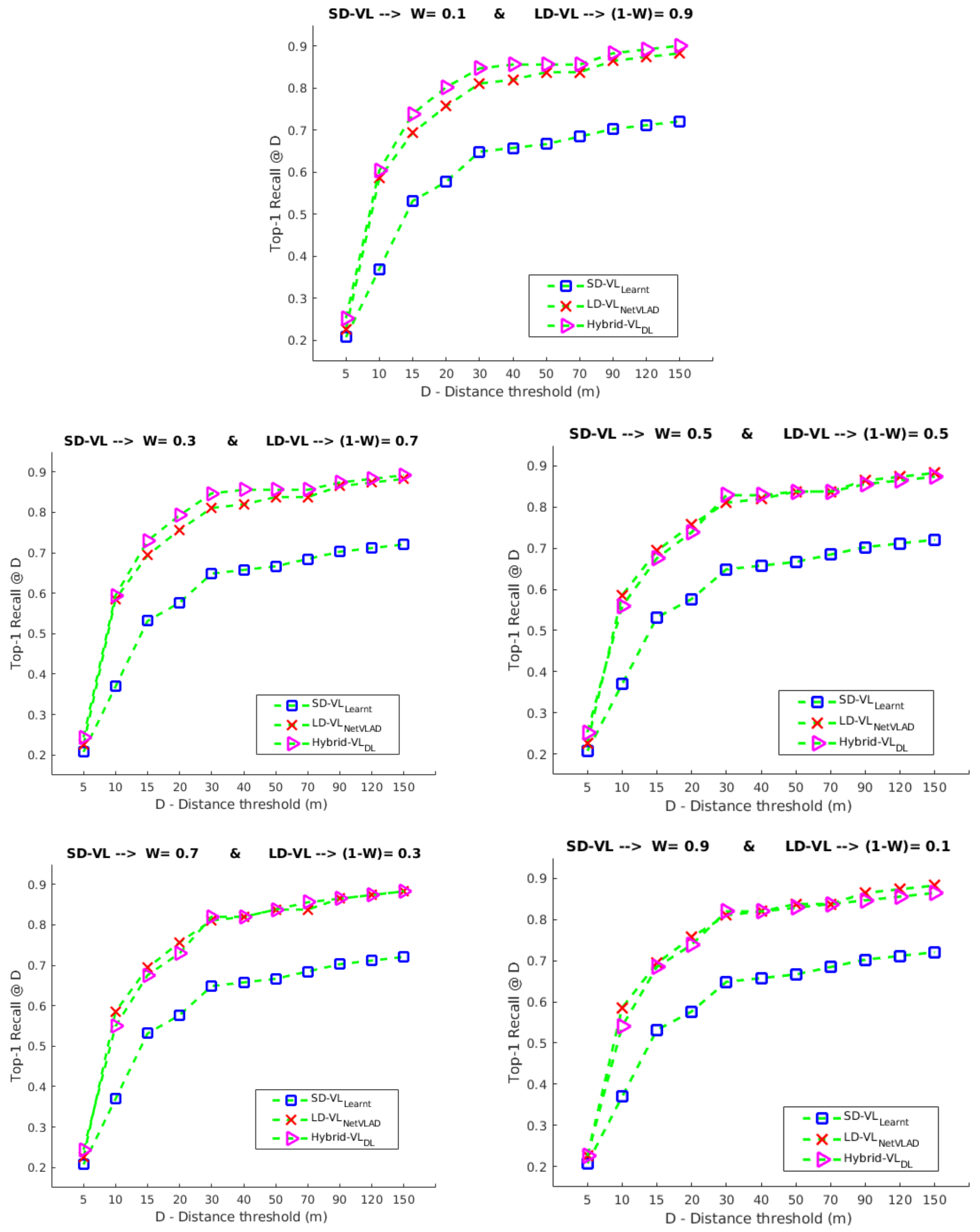


Figure 4.14. Importance of fine-tuning in Hybrid-VL$_{DL}$ method on *Malaga Streetview Challenge* data set with varying *W* hyper-parameters. Best hybridization result (top row one) is gained with higher contribution of LD-VL (W=0.1).

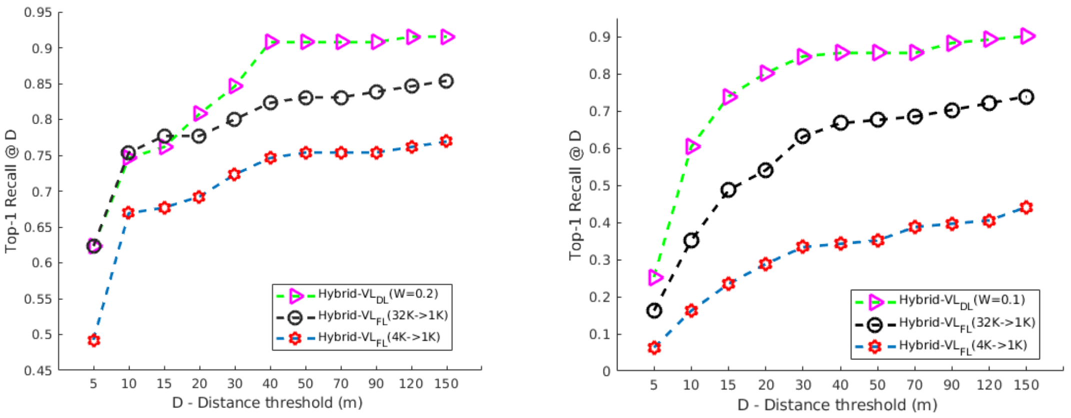


Figure 4.15. Impact of the dimensional reduction (from 32K to 4K) on Hybrid-VL$_{FL}$ method is demonstrated for both data sets with respect to *Top-1 recall@D* metric. Dimensional reduction decreases the success of Hybrid-VL$_{FL}$ method.

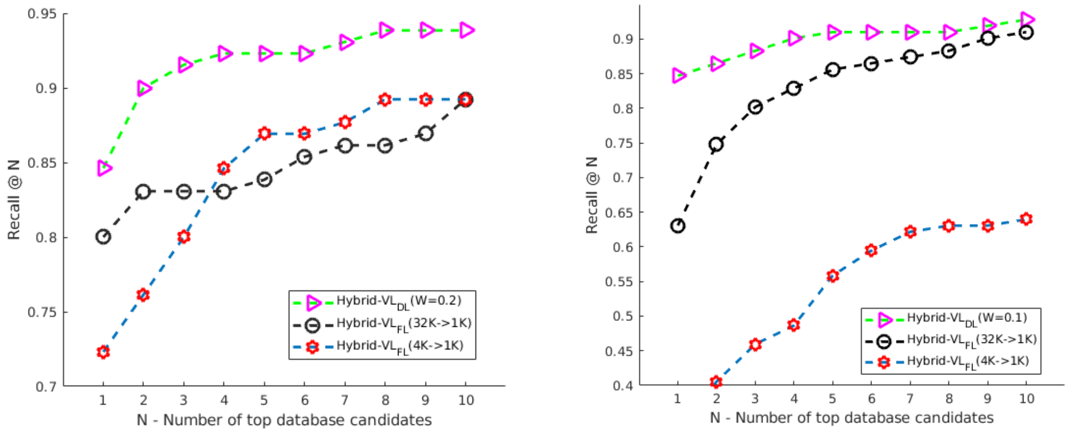


Figure 4.16. Impact of the dimensional reduction (from 32K to 4K) on Hybrid-VL$_{FL}$ method is demonstrated for both data sets with respect to *Recall @N* metric. Dimensional reduction decreases the success of Hybrid-VL$_{FL}$ method.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis we proposed Hybrid-VL methods based on semantic segmentation to improve localization performance. Firstly, semantic information is extracted from equally divided parts of semantically segmented images as a novel hand crafted semantic descriptor (SD) for VL in *2D-2D* matching space which is called as *non-learnt* SD-VL. Differently from the first one, a new SD is trained with a triplet ranking loss based CNN model using semantically segmented images, then this captured semantic representation is used directly for VL that is named as *learnt* SD-VL method. Also, both query and database images are segmented by applying the up-to-date CNN based semantic segmentation method DeepLabv3+.

Secondly, manually tuned-with-W-parameter Hybrid-VL$_{DL}$ method is proposed with combining the proposed *learnt* SD-VL and the baseline LD-VL methods in post-processing stage. Additionally, Hybrid-VL$_{FL}$ method that is based on newly designed NN trained with triplet loss is proposed in order to produce automatically tuned hybrid result. Then improved localization performances is measured with frequently used evaluation metrics on the benchmark *RobotCar Seasons* data set and newly generated *Malaga Streetview Challenge* data set which will be useful to the community of VL area. Also note that, the proposed localization approach is based on 2D-2D image matching and their semantic segmentation results which is much cheaper than the approaches that require the semantic 3D reconstruction of the environment.

Experimental results indicate that the performance of the proposed Hybrid-VL$_{DL}$ method is superior against the state-of-the-art baseline LD-VL method on both examined data sets. Proposed method is able to increase *Top-1 recall@5* localization performance by 11.6% and 4.5% on the RobotCar Seasons and Malaga Streetview Challenge data sets respectively. Also our approach outperforms the baseline method for D=25m by a 4% and 5.4% *Recall @1* performances on both data sets again respectively. Furthermore, reliability of our hyper-parameter (W) based Hybrid-VL$_{DL}$ approach is supported with the fact that very close performance is achieved with automatically tuned Hybrid-VL$_{FL}$ method. Finally we can conclude that, proposed Hybrid-VL$_{DL}$ method achieved to alleviate the shortcoming of the baseline method in such cases when it retrieves a wrong image as a result.

As for the future work, employing different kind of descriptors (e.g. using depth information instead of segmentation) would contribute to the success of this work. At the same time, performing the proposed method on omnidirectional cameras will increase the localization performance owing to its wide field of viewing angle. Moreover, success of the Hybrid-VL$_{FL}$ method could be increased via finding more suitable NN to train the feature-level hybrid descriptor. Furthermore, applying another non-linear machine learning technique (such as SVM, Random Forest Learning etc.) instead performing a CNN based one will also increase the performance of Hybrid-VL$_{FL}$ method.

Beyond all these technical evaluations I would like to share the experiences I have gained during this study. We have automated the ground truth generation for semantic segmentation via weakly-supervised segmentation method using a powerful NN based segmentation model. This implementation saves more time than labeling images manually. Also another important factor while building a segmentation-based application is to consider horizon level difference. Because, horizon level difference between training images used for pre-training a segmentation model and test images limits the performance of segmentation in a bad way. Further showing up the success of the proposed Hybrid-VL$_{DL}$ method which had already been within our expectations, interestingly we discovered that semantic segmentation contributes more to the Hybrid-VL$_{DL}$ method in cases where there is a contrast weather scenarios between database and query images. This result may be valuable for other studies that will use semantic segmentation. From start of this study to the end, I have witnessed how the insane advancement in deep learning has shaped issues in the field of computer vision. Despite all these changes in topics, we see that image-based localization studies maintain their popularity as being on the first day and studies in this area continue gaining more importance with the developments in autonomous vehicle driving.

REFERENCES

Andoni, A. and P. Indyk (2006). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pp. 459–468. IEEE.

Andreasson, H. and T. Duckett (2004). Topological localization for mobile robots using omni-directional vision and local features. *IFAC Proceedings Volumes 37*(8), 36–41.

Arandjelovic, R., P. Gronat, A. Torii, T. Pajdla, and J. Sivic (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307.

Arandjelović, R. and A. Zisserman (2012). Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2911–2918. IEEE.

Arandjelovic, R. and A. Zisserman (2013). All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1578–1585.

Arandjelović, R. and A. Zisserman (2014). Dislocation: Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, pp. 188–204. Springer.

Arandjelović, R. (2015). NetVLAD: CNN architecture for weakly supervised place recognition. `https://github.com/Relja/netvlad`. Online; accessed 30 June 2020.

Aubry, M., B. C. Russell, and J. Sivic (2014). Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics (ToG) 33*(2), 1–14.

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

Babenko, A., A. Slesarev, A. Chigorin, and V. Lempitsky (2014). Neural codes for image retrieval. In *European conference on computer vision*, pp. 584–599. Springer.

Badrinarayanan, V., A. Kendall, and R. Cipolla (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence 39*(12), 2481–2495.

Barnes, D., W. Maddern, and I. Posner (2017). Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 203–210. IEEE.

Bay, H., T. Tuytelaars, and L. Van Gool (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pp. 404–417. Springer.

Beis, J. S. and D. G. Lowe (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pp. 1000–1006. IEEE.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM 18*(9), 509–517.

Blanco-Claraco, J.-L., F.-Á. Moreno-Dueñas, and J. González-Jiménez (2014). The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *The International Journal of Robotics Research 33*(2), 207–214.

Calonder, M., V. Lepetit, C. Strecha, and P. Fua (2010). Brief: Binary robust independent elementary features. In *European conference on computer vision*, pp. 778–792. Springer.

Camposeco, F., A. Cohen, M. Pollefeys, and T. Sattler (2018). Hybrid camera pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 136–144.

Cao, S. and N. Snavely (2014). Minimal scene descriptions from structure from motion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 461–468.

Carlevaris-Bianco, N., A. K. Ushani, and R. M. Eustice (2016). University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research 35*(9), 1023–1035.

Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence 40*(4), 834–848.

Chen, L.-C., G. Papandreou, F. Schroff, and H. Adam (2019). Rethinking atrous convolution for semantic image segmentation. arxiv 2017. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818.

Choi, W., Y.-W. Chao, C. Pantofaru, and S. Savarese (2015). Indoor scene understanding with geometric and semantic contexts. *International Journal of Computer Vision 112*(2), 204–220.

Çinaroğlu, İ. and Y. Baştanlar (2019). Image based localization using semantic segmentation for autonomous driving. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4. IEEE.

Cinaroglu, I. and Y. Bastanlar (23 August 2020). Training semantic descriptors for image-based localization. In *ECCV 2020 Workshop on Perception for Autonomous Driving(PAD)*. ECCV.

Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Volume 1, pp. 886–893. IEEE.

Datar, M., N. Immorlica, P. Indyk, and V. S. Mirrokni (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pp. 253–262.

Davison, A. J., I. D. Reid, N. D. Molton, and O. Stasse (2007). Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence 29*(6), 1052–1067.

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

Fauqueur, J., G. Brostow, and R. Cipolla (2007). Assisted video object labeling by joint tracking of regions and keypoints. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–7. IEEE.

Fraundorfer, F., C. Engels, and D. Nistér (2007). Topological mapping, localization and navigation using image collections. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3872–3877. IEEE.

Furgale, P. and T. D. Barfoot (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics 27*(5), 534–560.

Furnari, A., G. M. Farinella, and S. Battiato (2016). Recognizing personal locations from egocentric videos. *IEEE Transactions on Human-Machine Systems 47*(1), 6–18.

Gaidon, A., Q. Wang, Y. Cabon, and E. Vig (2016). Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349.

Geiger, A., M. Lauer, C. Wojek, C. Stiller, and R. Urtasun (2013). 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence 36*(5), 1012–1025.

Germain, H., G. Bourmaud, and V. Lepetit (2018). Efficient condition-based representations for long-term visual localization. *arXiv preprint arXiv:1812.03707*.

Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Goedemé, T., M. Nuttin, T. Tuytelaars, and L. Van Gool (2004). Markerless computer vision based localization using automatically generated topological maps. In *Proceedings of the European Navigation Conference*, Volume 1, pp. 235–243.

Goedemé, T., M. Nuttin, T. Tuytelaars, and L. Van Gool (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision 74*(3), 219–236.

Harris, C. G., M. Stephens, et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, Volume 15, pp. 10–5244. Citeseer.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Inman, J. (1849). *Navigation and Nautical Astronomy, for the Use of British Seamen*. F. & J. Rivington.

Irschara, A., C. Zach, J.-M. Frahm, and H. Bischof (2009). From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2599–2606. IEEE.

Jaakkola, T. and D. Haussler (1999). Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems*, pp. 487–493.

Jegou, H., M. Douze, and C. Schmid (2008). Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pp. 304–317. Springer.

Jegou, H., M. Douze, and C. Schmid (2010). Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence 33*(1), 117–128.

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

Jo, J., J. Seo, and J.-D. Fekete (2017). A progressive kd tree for approximate k-nearest neighbors. In *2017 IEEE Workshop on Data Systems for Interactive Analysis (DSIA)*, pp. 1–5. IEEE.

Kendall, A. and R. Cipolla (2017). Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983.

Konolige, K. and M. Agrawal (2008). Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics 24*(5), 1066–1077.

Krähenbühl, P. and V. Koltun (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pp. 109–117.

Krähenbühl, P. and V. Koltun (2013). Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pp. 513–521.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105.

Kulis, B. and K. Grauman (2009). Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th international conference on computer vision*, pp. 2130–2137. IEEE.

Lhuillier, M. (2005). Automatic structure and motion using a catadioptric camera.

Lhuillier, M. (2007). Toward flexible 3d modeling using a catadioptric camera. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, Volume 2, pp. 1150–1157. Ieee.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*(2), 91–110.

Maddern, W., G. Pascoe, C. Linegar, and P. Newman (2017). 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research 36*(1), 3–15.

Majdik, A. L., D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza (2015). Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles. *Journal of Field Robotics 32*(7), 1015–1039.

McManus, C., B. Upcroft, and P. Newmann (2014). Scene signatures: Localised and point-less features for localisation.

Milford, M. J. and G. F. Wyeth (2008). Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics 24*(5), 1038–1053.

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

Mousavian, A. and J. Kosecka (2016). Semantic image based geolocation given a map. *arXiv preprint arXiv:1609.00278*.

Mousavian, A., J. Košecká, and J.-M. Lien (2015). Semantically guided location recognition for outdoors scenes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4882–4889. IEEE.

Muja, M. and D. Lowe (2009a). Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*.

Muja, M. and D. G. Lowe (2009b). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1) 2*(331-340), 2.

Muja, M. and D. G. Lowe (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine* *intelligence 36*(11), 2227–2240.

Murillo, A., P. Campos, J. Kosecka, and J. Guerrero (2010). Gist vocabularies in omnidirectional images for appearance based mapping and localization. *10th OMNIVIS, held with Robotics: Science and Systems (RSS) 3*.

Murillo, A. C. and J. Kosecka (2009). Experiments in place recognition using gist panoramas. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 2196–2203. IEEE.

Murillo, A. C., G. Singh, J. Kosecká, and J. J. Guerrero (2012). Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics 29*(1), 146–160.

Naseer, T., L. Spinello, W. Burgard, and C. Stachniss (2014). Robust visual robot localization across seasons using network flows. In *Twenty-eighth AAAI conference on artificial intelligence*.

Nister, D. and H. Stewenius (2006). Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Volume 2, pp. 2161–2168. Ieee.

Oliva, A. and A. Torralba (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision 42*(3), 145–175.

Oliva, A. and A. Torralba (2006). Building the gist of a scene: The role of global image features in recognition. *Progress in brain research 155*, 23–36.

Ondruska, P., J. Dequaire, D. Z. Wang, and I. Posner (2016). End-to-end tracking and semantic segmentation using recurrent neural networks. *arXiv preprint arXiv:1604.05091*.

Orlita, T. (2016). iStreetView.com is an online viewer for Street View™. `https://thomasorlita.com/projects/istreetview.com/`. Online; accessed 30 June 2020.

Perronnin, F. and C. Dance (2007). Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8. IEEE.

Philbin, J., O. Chum, M. Isard, J. Sivic, and A. Zisserman (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8. IEEE.

Piasco, N., D. Sidibé, V. Gouet-Brunet, and C. Demonceaux (2019). Learning scene geometry for visual localization in challenging conditions. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9094–9100. IEEE.

Radenović, F., G. Tolias, and O. Chum (2016). Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pp. 3–20. Springer.

Radenovic, F., G. Tolias, and O. Chum (2018). Deep shape matching. In *Proceedings of the european conference on computer vision (eccv)*, pp. 751–767.

Radenović, F., G. Tolias, and O. Chum (2018). Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence 41*(7), 1655–1668.

Rosten, E. and T. Drummond (2006). Machine learning for high-speed corner detection. In *European conference on computer vision*, pp. 430–443. Springer.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski (2011). Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pp. 2564–2571.

Ieee.

Sattler, T., M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys (2015). Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2102–2110.

Sattler, T., M. Havlena, K. Schindler, and M. Pollefeys (2016). Large-scale location recognition and the geometric burstiness problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1582–1590.

Sattler, T., W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, F. Kahl, M. Pollefeys, J. Sivic, and T. Pajdla (2018). The Visual Localization Benchmark. `https://www.visuallocalization.net/`. Online; accessed 30 June 2020.

Sattler, T., W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, et al. (2018). Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8601–8610.

Sattler, T., T. Weyand, B. Leibe, and L. Kobbelt (2012). Image retrieval for image-based localization revisited. In *BMVC*, Volume 1, pp. 4.

Schönberger, J. L., M. Pollefeys, A. Geiger, and T. Sattler (2018). Semantic visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6896–6906.

Schroff, F., D. Kalenichenko, and J. Philbin (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823.

Schroth, G., R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach (2011). Mobile visual location recognition. *IEEE Signal Processing Magazine 28*(4), 77–89.

Seymour, Z., K. Sikka, H.-P. Chiu, S. Samarasekera, and R. Kumar (2018). Semantically-aware attentive neural embeddings for image-based visual localization. *arXiv preprint arXiv:1812.03402*.

Silpa-Anan, C. and R. Hartley (2008). Optimised kd-trees for fast image descriptor matching. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE.

Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, G. and J. Kosecka (2010). Visual loop closing using gist descriptors in manhattan world. In *ICRA omnidirectional vision workshop*, pp. 4042–4047.

Singh, G. and J. Košecká (2012). Acquiring semantics induced topology in urban environments. In *2012 IEEE International Conference on Robotics and Automation*, pp. 3509–3514. IEEE.

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

Stenborg, E., C. Toft, and L. Hammarstrand (2018). Long-term visual localization using semantically segmented images. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6484–6490. IEEE.

Sünderhauf, N., S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford (2015). On the performance of convnet features for place recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4297–4304. IEEE.

Sünderhauf, N., S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and

M. Milford (2015). Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tekir, S. and Y. Bastanlar (2020). Deep learning: Exemplar studies in natural language processing and computer vision. In *Data Mining-Methods, Applications and Systems*. IntechOpen.

Toft, C., E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl (2018). Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 383–399.

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

Torii, A., H. Taira, J. Sivic, M. Pollefeys, M. Okutomi, T. Pajdla, and T. Sattler (2019). Are large-scale 3d models really necessary for accurate visual localization? *IEEE transactions on pattern analysis and machine intelligence*.

Van Gemert, J. C., C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek (2009). Visual word ambiguity. *IEEE transactions on pattern analysis and machine intelligence 32*(7), 1271–1283.

Vedaldi, A. and K. Lenc (2015). Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*.

Veness, C. (2002). Calculate distance, bearing and more between Latitude/Longitude points. `https://www.movable-type.co.uk/scripts/latlong.html`.

Online; accessed 30 June 2020.

Weiss, Y., A. Torralba, and R. Fergus (2009). Spectral hashing. In *Advances in neural information processing systems*, pp. 1753–1760.

Zeiler, M. D. and R. Fergus (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer.

Zeiler, M. D., G. W. Taylor, and R. Fergus (2011). Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pp. 2018–2025. IEEE.

Zeisl, B., T. Sattler, and M. Pollefeys (2015). Camera pose voting for large-scale image-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2704–2712.

Zhou, B., A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pp. 487–495.

Zitnick, C. L. and P. Dollár (2014). Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pp. 391–405. Springer.

VITA

İbrahim ÇINAROĞLU

Academic Experience

2013–Present **Research Assistant**, Department of Computer Engineering, Izmir Institute of Technology

Education

2011–2014 **MSc in Computer Engineering**, İzmir Institute of Technology, İzmir, Turkey
Title: A Direct Approach for Object Detection with Omnidirectional Cameras
Advisor: Asst. Prof. Dr. Yalın Baştanlar

2006–2010 **BSc in Computer Engineering**, Süleyman Demirel University, Isparta, Turkey

Publications

JOURNAL ARTICLES

2016 Cinaroglu, I. and Y. Bastanlar. A direct approach for object detection with catadioptric omnidirectional cameras. Signal, Image and Video Processing, 10(2), 413-420, Springer.

CONFERENCE PRESENTATIONS

2020 Cinaroglu, I. and Y. Bastanlar. Training semantic descriptors for image-based localization. In ECCV 2020 Workshop on Perception for Autonomous Driving(PAD). ECCV.

2019 Çinaroğlu, İ. and Y. Baştanlar. Image based localization using semantic segmentation for autonomous driving. In 2019 27th Signal Processing and Communications Applications Conference (SIU), pp. 1–4. IEEE.

2015 Karaimer, H. C., I. Cinaroglu, and Y. Bastanlar. Combining shape-based and gradient-based classifiers for vehicle classification. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems (pp. 800-805). IEEE.

2014 Cinaroglu, I., and Y. Bastanlar. A direct approach for human detection with catadioptric omnidirectional cameras. In 2014 22nd Signal Processing and Communications Applications Conference (SIU) (pp. 2275-2279). IEEE.