SEMANTIC SEGMENTATION OF PANORAMIC IMAGES AND PANORAMIC IMAGE BASED OUTDOOR VISUAL LOCALIZATION

A Dissertation Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
Semih ORHAN

October 2022
İZMİR

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere thanks to my advisor, Assoc. Prof. Dr. Yalın Baştanlar, for his patience, guidance, support, and feedback throughout my PhD studies.

I would like to thank my thesis monitoring committee Asst. Prof. Dr. Nesli Erdoğmuş and Assoc. Prof. Dr. Devrim Ünay, and thesis examining committee members Assoc. Prof. Dr. Zerrin Işık and Assoc. Prof. Dr. Mustafa Özuysal for their valuable feedback.

I would like to thank my friends and colleagues. Finally, I would like to thank my family for their love and support.

This thesis is supported by Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 120E500 and also under 2214-A International Researcher Fellowship Programme.

ABSTRACT

SEMANTIC SEGMENTATION OF PANORAMIC IMAGES AND PANORAMIC IMAGE BASED OUTDOOR VISUAL LOCALIZATION

360-degree views are captured by full omnidirectional cameras and generally represented with panoramic images. Unfortunately, these images heavily suffer from the spherical distortion at the poles of the sphere. In previous studies of Convolutional Neural Networks (CNNs), several methods have been proposed (e.g. equirectangular convolution) to alleviate spherical distortion. Getting inspired from these previous efforts, we developed an equirectangular version of the UNet model. We evaluated the semantic segmentation performance of the UNet model and its equirectangular version on an outdoor panoramic dataset. Experimental results showed that the equirectangular version of UNet performed better than UNet. In addition, we released the pixel-level annotated dataset, which is one of the first semantic segmentation datasets of outdoor panoramic images.

In visual localization, localizing perspective query images in a panoramic image dataset can alleviate the non-overlapping view problem between cameras. Generally, perspective query images are localized in a panoramic image database with generating its virtual 4 or 8 gnomonic views, which is deforming sphere into cube faces. Doing so can simplify the searching problem to perspective to perspective search, but still there might be a non-overlapping view problem between query and gnomonic database images. Therefore we propose directly localizing perspective query images in panoramic images by applying sliding windows on the last convolution layer of CNNs. Features are extracted with R-MAC, GeM, and SFRS. Experimental results showed that the sliding window approach outperformed 4-gnomonic views, and we get competitive results compared with 8 and 12 gnomonic views.

Any city-scale visual localization system has to be robust against long-term changes. Semantic information is more robust to such changes (e.g. surface of the building), and the depth maps provide geometric clues. In our work, we utilized semantic and depth information while pose verification, that is checking semantic and depth similarity to verify the poses (retrievals) obtained with the approach that use only RGB image features. Semantic and depth information are represented with a self-supervised contrastive learning approach (SimCLR). Experimental results showed that pose verification with semantic and depth features improved the visual localization performance of the RGB-only model.

ÖZET

PANORAMİK İMGELERDE ANLAMSAL BÖLÜTLEME VE PANORAMİK İMGE TABANLI DIŞ MEKAN GÖRSEL KONUMLANDIRMA

360-derece görüntüler tümyönlü kameralar ile çekilir ve genellikle panoramik imgeler ile temsil edilir. Ne yazık ki, panoramik imgeler kürenin kutup noktalarında aşırı küresel bozunuma maruz kalır. Evrişimli Yapay Sinir Ağları (EYSA) literatüründe, küresel bozunumun etkisini azaltmak için birçok yöntem önerilmiştir (örn. eşdikdörtgensel evrişim). Önceki çalışmalardan esinlenerek, UNet modelinin eşdikdörtgensel evrişim versiyonunu geliştirdik. UNet modeli ve onun eşdikdörtgensel evrişim versiyonunun anlamsal bölütleme performansını dış mekan panoramik veri kümesi üzerinde ölçtük. Deney sonuçları, UNet'in eşdikdörtgensel evrişim versiyonunun, UNet'den daha iyi performans gösterdiğini göstermiştir. Ek olarak, piksel seviyesinde etiketlenmiş anlamsal bölütleme için ilk dış mekan panoramik imge veri kümelerinden birini yayınladık.

Görsel konumlandırma yaparken, perspektif sorgu imgelerini panoramik veri kümesinde aramak kameralar arasındaki örtüşmeyen görüntü problemini hafifletebilir. Genellikle, perspektif sorgu imgeleri panoramik veri kümesi içinde panoramik imgelerin 4 veya 8 gnomonik görüntüleri (kürenin küp ile temsili) üretilerek konumlandırılır. Bunu yapmak, konumlandırma problemini perspektiften perspektif aramaya indirgeyebilir, fakat sorgu ve gnomonik veri kümesi imgeleri arasında hala örtüşmeyen görüş açısı problemi olabilir. Bu nedenle perspektif sorgu imgelerini doğrudan panoramik imgeler içerisinde aramayı önerdik. Bunu yapmak için, kayan pencere yaklaşımını EYSA'nın son evrişim katmanına uyguladık. Öznitelikleri R-MAC, GeM ve SFRS ile çıkardık. Deney sonuçlarında, kayan pencere yöntemi 4 gnomonik görüşe göre çok daha iyi sonuçlar üretti, ve kayan pencere yöntemi ile 8 ve 12 gnomonik görüşe göre rekabetçi sonuçlar aldık.

Herhangi bir görsel konumlandırma sistemi uzun vadeli değişikliklere karşı gürbüz olmalıdır. Anlamsal bilgi bu değişikliklere karşı daha gürbüzdür (örn: binanın yüzeyi), ve derinlik haritaları geometrik bilgi sağlar. Çalışmamızda, anlamsal ve derinlik bilgisini poz doğrulama aşamasında kullandık. Poz doğrulama RGB model ile getirilen pozların (sonuçların) anlamsal ve derinlik benzerlikleri ile doğrulanmasıdır. Anlamsal ve derinlik bilgisini özdenetimli karşılaştırmalı öğrenme yaklaşımı (SimCLR) ile temsil ettik. Deney sonuçları anlamsal ve derinlik öznitelikleri ile poz doğrulamanın sadece RGB öznitelik kullanan modelin görsel konumlandırma performansını arttırdığını gösterdi.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure **Page**

Figure **Page**

Figure **Page**

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

FOV . . . . . Field-of-view
CNN . . . . . Convolutional Neural Network
GPS . . . . . Global Positioning System
Equiconv . . . . . Equirectangular convolution
Stdconv . . . . . Standard convolution
FCN . . . . . Fully Convolutional Networks
OPP . . . . . Overlapping Pyramid Pooling
VLAD . . . . . Vector of locally aggregated descriptors
MAC . . . . . Maximum activations of convolutions
R-MAC . . . . . Regional maximum activation of convolutions
GeM . . . . . Generalized-Mean
SFRS . . . . . Self-supervising fine-grained region similarities
PnP . . . . . Perspective-n-Learned-Points
SimCLR . . . . . Simple framework for contrastive learning
MoCo . . . . . Momentum contrast
ReLU . . . . . Rectified linear units
mIoU . . . . . Mean intersection over union
UNet . . . . . U-shaped neural network
SGD . . . . . Stochastic gradient descent
PCA . . . . . Principal component analysis

CHAPTER 1

INTRODUCTION

Omnidirectional cameras can capture surrounding area (360° of view) within a single shot. They provide more information than perspective cameras. Recently, they have gained popularity because of having wider field-of-view (FOV). Many computer vision applications can benefit from it (e.g. autonomous driving and visual localization). Full omnidirectional cameras cover 180° vertical (left to right), and 360° horizontal (bottom to up) views. Generally, 360° imagery is represented with equirectangular projection. Coordinates are proportional to latitude and longitude of the sphere, i.e., unit distance in horizontal or vertical direction in the image corresponds to a fixed amount of angular coverage. Unfortunately, equirectangular projection heavily suffers from spherical distortion moving towards to poles due to not having enough pixel space. Objects which are close to the poles look different than they would appear in the perspective images. This effect is shown in Figure 1.1.

Figure 1.1. An equirectangular panoramic image.

Spherical distortion is a challenge for conventional computer vision approaches, since most of approaches are developed considering perspective images, and adopting

already existing approaches for omnidirectional 360° view is not a trivial task. In the past, numerous methods were proposed to handle this distortion (e.g. (Lourenço et al. (2012); Cinaroglu and Bastanlar (2016); Demiröz et al. (2019))). Not surprisingly, recent efforts on handling the distortion have been focused on CNNs (Fernandez-Labrador et al. (2020); Tateno et al. (2018); Guerrero-Viu et al. (2020); Coors et al. (2018)). In previous works of CNNs, spherical distortion is explicitly modeled, and the offsets of the grids are calculated beforehand. Unlike the standard convolution layer, spherical convolution is done regarding spherical coordinates. Previous works (Fernandez-Labrador et al. (2020); Tateno et al. (2018); Guerrero-Viu et al. (2020); Coors et al. (2018)) were limited to object detection, depth map estimation, semantic segmentation on synthetic outdoor images and real indoor images. We developed a CNN model (called UNet-equiconv) for semantic segmentation on outdoor panoramic images in the third chapter of the thesis. In our CNN model, we replaced each standard convolution layer with equirectangular convolution (Fernandez-Labrador et al. (2020)) to eliminate spherical distortion of panoramic images at convolution time.

In the fourth chapter of the thesis, we focused on the visual localization of outdoor panoramic images. In visual localization, approximate location of query material is estimated within a visual map. GPS-based systems stumble in such case where environment is cluttered, or if there is harsh weather conditions. Due to the fact that, interest in visual localization systems has increased in recent years (Piasco et al. (2018)), which can be used as supporting or alternative localization system. In our work, we used image retrieval (Tolias et al. (2016); Radenović et al. (2018)) and metric localization based (Ge et al. (2020)) approaches. In our settings, perspective (narrow FOV) query images is searched in an equirectangular outdoor panoramic image database. In the dataset, both query and database have GPS information. GPS location (latitude and longitude) of retrieved database images serve as an approximate location of query images. In the last decade, many computer vision approaches (Torii et al. (2015); Babenko and Lempitsky (2015); Arandjelovic et al. (2016); Ge et al. (2020)) have been proposed for visual localization. There are several challenges for these systems, and one of them is long-term visual localization (Toft et al. (2018a); Stenborg et al. (2018); Naiming et al. (2018)). Query and database images can be collected in different seasons or under different illumination and visual localization systems should handle these long-term effects.

In our dataset, database images are collected from different locations of Pittsburgh, PA., and query images are randomly taken from UCF dataset (Zamir and Shah (2014)).

There is at least one database image within a five meter distance to each query image. Our dataset fits more the topological localization problem rather than a metric localization (Lu et al. (2013); Goedemé et al. (2007); Chen et al. (2017); Iscen et al. (2017)). A common way to search perspective query images in an equirectangular panoramic image database is to generate 4 or 8 virtual perspective gnomonic views, which are generated moving alongside the equator of spheres, and each gnomonic images have a 90° FOV. Each gnomonic view in the 8-gnomonic database overlaps 45-degree FOV with the next one. But there might exist non-overlapping FOV views between two perspective images, thus query images could not be correctly localized. 360° imagery (panoramic images) helps us to solve this problem. In our work (Orhan and Bastanlar (2021)), we used 360° vision, full equirectangular panoramic images directly.

Our main contribution in the fourth chapter is that, unlike previous works which generate virtual perspective images using gnomonic projection, we directly localize perspective query images in a database that is composed of equirectangular outdoor panoramic images by applying a sliding window to the last convolutional layer of the CNN. We used three different feature extraction methods (Tolias et al. (2016); Radenović et al. (2018); Ge et al. (2020)), and provided experimental results with topological localization. Experimental results show that sliding windows outperform 4 gnomonic projections (90-degree field-of-view non overlapping images), and we get competitive results compared to 8 and 12 gnomonic projections (45-degree overlapping and 60-degree overlapping, 90-degree FOV images, respectively).

In the fifth chapter of the thesis, we exploit semantic and depth information for visual localization (Orhan et al. (2022)) to alleviate the long-term appearance changes such as query and database images could be collected in different years which might cause the seasonal difference and structures changes. Semantic information of the scene is more robust to long-term changes (e.g, surface of the building), and depth maps provides geometric clues. In our scenario, database images consist of 8 and 12 gnomonic views generated from panoramic images, and query set consisting of perspective images which are captured within a five meter of database image in different years. In (Bastanlar and Orhan (2022)), we observed that self-supervised contrastive learning approach can be used for semantic instance discrimination. Thus, we represent semantic, and depth information with a self-supervised contrastive learning (SimCLR) approach that is proposed by Chen et al. (2020). Unlike supervised training, self-supervised learning helps us to train our model without requiring a decent amount of labeled data. We can obtain a vast amount of

pseudo labels for a pretext task with a well-performing CNN model. We represented each location in our database with its semantic and depth masks using self-supervised learning. Embedding space representation is learned by comparing the similarity between positive samples and dissimilarity between negative ones in contrastive learning. The main objective is to learn an embedding space where positive pairs (anchor and its augmented version or other samples with the same label) stay close, and negative pairs stay far away. In self-supervised settings, positive pairs are generated by applying data augmentation on the same instance (e.g. random crop, rotation, and color jitters), and negative ones are other instances than input samples. In (Orhan et al. (2022)), we improved visual localization performance of the RGB-only model (Ge et al. (2020)) more than %1 utilizing semantic features at pose verification steps.

We extended our work (Orhan et al. (2022)) utilizing depth information at the pose verification step in addition to RGB and semantic features. To localize perspective query images in a panoramic image database, we employed the sliding window and gnomonic views (details are in Chapter 5). We illustrated multi-modal visual localization in Figure 1.2.

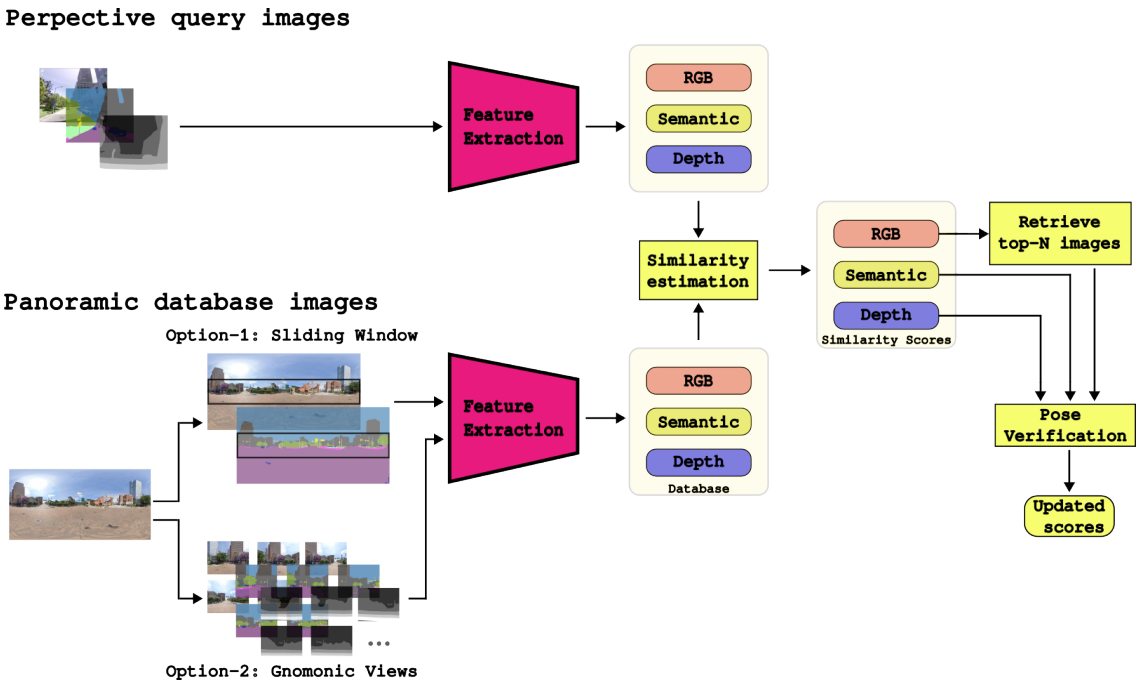


Figure 1.2. Illustration of multi-modal visual localization.

I summarize main contributions of the thesis below:

- To tackle the spherical distortion of panoramic images during semantic segmentation, we developed an equirectangular version of UNet (called UNet-equiconv). The only difference between standard and equirectangular version of UNet is their convolution types. We conducted several experiments and observed that the equirectangular version of UNet performed better than its standard convolution version (UNet-stdconv). We released one of the first equirectangular outdoor panoramic image dataset for semantic segmentation task.

- In previous works, a common way to localize perspective query images in a panoramic image database is to generate their virtual gnomonic views. Instead of generating virtual perspective gnomonic images and matching query images with them, we directly localize query images in an equirectangular panoramic image database. To do it, we applied sliding windows to the last feature map of the CNN, and we visited more locations in less amount of time. Experimental results showed that the sliding window outperformed the 4-gnomonic projection, and we get competitive results compared to 8 and 12 gnomonic projections.

- In the thesis' fifth chapter, we researched long-term visual localization on panoramic images. We utilized semantic and depth information at the pose verification step. We represented semantic and depth information with a self-supervised contrastive learning approach (SimCLR). Experimental results showed that utilizing semantic and depth information improved the visual localization performance of the RGB-only model (SFRS).

CHAPTER 2

LITERATURE REVIEW

2.1. Semantic Segmentation

Computer vision composes of many research areas (e.g. object detection, scene understanding, visual localization). Semantic segmentation is one of them. It is assigning labels to each pixel in images. Many computer vision applications benefit of semantic segmentation such as pedestrian detection (Mao et al. (2017); Costea and Nedevschi (2016)); autonomous vehicles (Siam et al. (2018); Teichmann et al. (2018)); remote sensing (Kampffmeyer et al. (2016); Sun and Wang (2018)); and pose estimation (Peng et al. (2019); Wong et al. (2017)).

In recent years, deep learning-based approaches outperformed previous works on semantic segmentation. Here, we adhere literature review only on deep learning-based approaches. We divided semantic segmentation literature into two. First, we summarize studies that are proposed for perspective (narrow FOV) images. Second, we explain previous works on panoramic images which tackle or not the spherical distortion of panoramic images.

2.1.1. Semantic segmentation of narrow FOV Images

Long et al. (2015) proposed Fully Convolutional Networks (FCN). They removed the classification layer of CNN models. With this modification, the proposed CNN model could work with variable size of inputs. Features are encoded with convolutional layers, and up-sampled with skip connection and bilinear interpolation. Features are taken from the different depths of the model. Noh et al. (2015) proposed DeconvNet. The proposed model composes of two parts, encoder and decoder. In the encoder part, features are encoded with convolution and pooling layers. In the decoder part, features are up-sampled using unpooling and transposed convolution (deconvolution) layers. In a follow-

up study, SegNet is proposed by (Badrinarayanan et al. (2017)). It is an encoder-decoder CNN model. Features are encoded with convolution and max-pooling layers. They are up-sampled with deconvolution and un-pooling layers in the decoder part of the model. Apart from previous encoder-decoder models, pooling indexes are used while decoding features. It helps to reduce parameters of the CNN model. Another encoder-decoder CNN model is proposed by (Ronneberger et al. (2015)), called UNet. Encoded features are decoded with bilinear interpolation and concatenated using skip connections. There are better-performing CNN models (Chen et al. (2022); Yuan et al. (2020); Yan et al. (2022)) on semantic segmentation, but we preferred to use UNet (Ronneberger et al. (2015)) in our work because of easy implementation of equirectangular convolution to the CNN architecture.

2.1.2. Semantic Segmentation on wide FOV Images

Distortion aware convolution methods on panoramic images exist in the previous works (Fernandez-Labrador et al. (2020); Coors et al. (2018); Tateno et al. (2018)). The main idea is that convolution operating is done with offsets of the grids that are calculated regarding the spherical distortion beforehand.

Spherical convolution was proposed by Coors et al. (2018). Unlike standard convolution layers, which use regular grid coordinates, spherical convolution is done with offsets of the grids. They conducted experiments on object detection and image classification tasks. Another distortion-aware convolution approach was proposed by Tateno et al. (2018). They tested the proposed approach on depth estimation and semantic segmentation. They trained proposed CNN model on perspective images and tested on panoramic images.

SPHCONV is proposed by Su and Grauman (2017). An advantage of the proposed CNN model is that it can learn spherical representation training on narrow FOV images and does not need any labeled panoramic image dataset. Perspective cameras have narrow field-of-view and objects near to the cameras can not be fully captured. Thus training on perspective images, where objects close to the cameras degrades the performance. Another limitation of SPHCONV is that number of parameter of the model linearly increase with the height of the panoramic images.

OOP-net is proposed by Deng et al. (2017). Apart from the previous works shown

above, OOP-net consists of Overlapping Pyramid Pooling (OPP) method that utilizes not only local but also global information simultaneously. Proposed OOP-net model is trained on a fisheye dataset where FOV is $180°$, without distortion handling.

Equirectangular convolution is proposed by Fernandez-Labrador et al. (2020), to estimate the 3D layout of the rooms. It is a special form of deformable convolution (Dai et al. (2017)) layer, where offsets of the convolution layers are calculated regarding the spherical distortion. In a later work, equirectangular version of BliztNet (Dvornik et al. (2017)) is introduced by Guerrero-Viu et al. (2020). They presented results on a panoramic indoor dataset.

A synthetic panoramic image dataset is released by Xu et al. (2019) for semantic segmentation task. They generated synthetic panoramic images from SYNTHIA sequence dataset (Ros et al. (2016)). There is a style difference between synthetic and real images, which degrades the performance of CNNs. In Orhan and Bastanlar (2022), we released pixel-level annotated one of the first real panoramic image datasets for semantic segmentation.

We used equirectangular convolution (Fernandez-Labrador et al. (2020)) in our work (Orhan and Bastanlar (2022)) and introduced UNet-equiconv. Apart from previous works, we trained and presented our results on a dataset that consists of real outdoor panoramic images. We showed the results with different pre-training weights and evaluated the distortion elimination effect on the outdoor panoramic image dataset.

2.2. Visual Localization and Image Retrieval

We summarize visual localization and image retrieval literature into two groups. First, we will explain visual localization studies before and after the era of CNNs. Afterward, we will summarize previous studies which are proposed for panoramic images.

Before the era of CNN-based methods, image retrieval, and visual localization systems used hand-crafted based approaches such as Bag-of-Features (Philbin et al. (2007)). Descriptors were extracted with SIFT (Lowe (1999)) and SURF (Bay et al. (2006)) like approaches, and databases are clustered into sets of visual words. Jégou et al. (2011) introduced Vector of locally aggregated descriptors (VLAD). The proposed approach extract features in a more compact representation. Later on, approaches which are robust to viewpoint and illumination changes (Torii et al. (2015)) and repetitive structures (Torii

et al. (2015)) were proposed.

In recent years, deep learning-based approaches performed well in visual localization and image retrieval tasks. Sünderhauf et al. (2015) showed viewpoint-invariant property of features extracted from CNN for the place recognition task. Chen et al. (2017) compared CNN and non-CNN-based methods and demonstrated results with different feature extraction techniques. Arandjelovic et al. (2016) proposed a CNN model which consists of convolution and learn-able VLAD (Jégou et al. (2011)) layers. Ge et al. (2020) proposed SFRS, which alleviates the noisy labels of geo-tagged images and outperformed previous works on visual localization task.

Another body of research conducted on image retrieval. It is a system, where the most similar N database images are retrieved regarding the query. These systems use topological localization. R-MAC pooling layer is proposed by Tolias et al. (2016). The max-pooling operation is applied to different location of feature maps with varying resolutions. Radenović et al. (2018) proposed one of the first trainable generalized mean (GeM) pooling layers. Instead of taking the maximum or an average of the receptive field, GeM pooling learns the pooling characteristic after training. In a follow-up study, Radenović et al. (2018) demonstrated GeM pooling performance on well-known image retrieval datasets.

The previous works we mentioned about did not use 360-degree FOV. They localized perspective narrow FOV query images in perspective image databases. While searching perspective query images in a perspective image database, there might be a non-overlapping view problem between query and database images. This problem is illustrated in Figure 2.1. This problem frequently happens in well-known benchmark datasets (Sattler et al. (2018)).

We can group previous works of image retrieval and visual localization systems with 360-degree imagery into two. In the first group, query and database are panoramic (Karkus et al. (2020); Hansen and Browning (2015); Cheng et al. (2019); Murillo et al. (2012); Lu et al. (2013); Wang et al. (2018); Iscen et al. (2017); Goedemé et al. (2007)). They work directly on omnidirectional images (dough-nut images obtained with an omnidirectional sensor), others convert omnidirectional images to panoramic images. The problem can be called panorama to panorama matching in the first group. Features can be extracted with SIFT-like (Lowe (1999); Bay et al. (2006)), CNN-based methods (Karkus et al. (2020); Cheng et al. (2019); Wang et al. (2018); Iscen et al. (2017)).

In the second group, database consists of panoramic images, whereas query set

(a) (b)

Figure 2.1. An example scenario where query image is collected when the car is moving toward to street, and database image is collected from opposite direction. An example database is shown in (a), and an example query image taken from the same location with the opposite viewing angle is shown in (b). The scene depicted in the query and database is quite different even though they are collected from the same location.

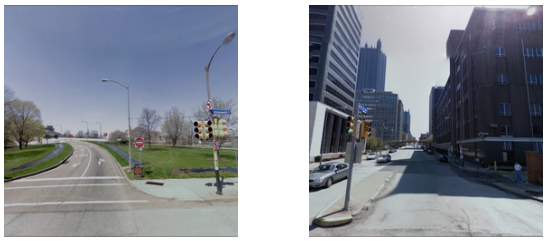
consists of perspective images (Zamir and Shah (2010); Schroth et al. (2011); Huang et al. (2016)). This scenario is more realistic because panoramic images can be collected offline while query images can be captured with any standard FOV cameras. Previous studies generally represented panoramic images with 4 or 8 gnomonic projections. 4 non-overlapping gnomonic projection (each 90° FOV) corresponds to cubemap representation (Zamir and Shah (2010)). An example virtual perspective images generated with 4-gnomonic projection are shown in Figure 2.2b. 8 gnomonic projection were used in (Huang et al. (2016)).

We can reduce perspective to panoramic searching problem to perspective to perspective searching by using gnomonic projection, but it comes with the cost of increasing the total number of images in the database. Generating virtual perspective images with gnomonic projection can not guarantee good matching between query (Fig. 2.2c) and database (Fig 2.2d) images. To tackle this problem, we can generate virtual perspective images with higher overlapping field-of-view, but it increases the number of images in the database. Apart from previous works, in (Orhan and Bastanlar (2021)), we directly searched perspective query images in an equirectangular panoramic image database by applying sliding to the last convolution layer (feature map) of CNN models (Ge et al.

(2020); Tolias et al. (2016); Radenović et al. (2018)). With this method, we do not need to generate virtual gnomonic view database to localize perspective query images.

(a)

(b)



(c) (d)

Figure 2.2. a) An example panoramic image. b) Virtual perspective images (each has 90° FOV) generated from the panoramic image. c) An example query image having 45° orientation. d) Another query image having 225° orientation. Query images shown (c) and (d) do not overlap with the database images (b). There is not only a non-overlapping problem between query and database images but also illumination changes. This will result in poor matching performance.

Utilizing semantic information for visual localization. Semantic information more robust to seasonal and structural changes, which is one of the main challenges for city-scale visual localization. We categorize semantic visual localization literature into two. First, we summarize literature works of 3D structure-based methods, and after that we explain related works of 2D-based image retrieval approaches. In (Stenborg et al. (2018)), 3D structure of the scene are built together with its semantic labels. Query images are localized in a 3D environment. In another work, 2D-3D point matches are checked if their semantic labels are the same (Toft et al. (2018b)). We used 2D-based approaches in our work which relies on retrieving the most similar images from the database. Main advantage of 2D-based approaches is that generally they requires less computation power than 3D based methods (Sattler et al. (2017)).

Among 2D approaches, street intersections are detected with training a classifier on semantic descriptors in (Singh and Košecká (2012)). Semantic edge-based localization method is proposed in (Yu et al. (2018)). Semantic edges (e.g. tree-sky, building-sky) are extracted with CASENet (Yu et al. (2017)). Cinaroglu and Bastanlar (2020) used only semantic features for visual localization. An attention-based visual localization approach is proposed by Seymour et al. (2019). The proposed CNN model utilizes appearance and semantic scene information. To detect the man-made landmark structures (e.g. buildings) of the scene, semantic information is incorporated by Mousavian et al. (2015). Features belonging to other than landmarks are considered unreliable and discarded. Some semantic objects are more robust to long-term changes (e.g. buildings). Based-on this idea, semantic weighting approach is proposed in (Naseer et al. (2017)). Regarding robustness of semantic classes to long-term changes, semantic weights of the robust classes (e.g building) are increased. Cinaroglu and Bastanlar (2022) proposed a hybrid method which utilize RGB and semantic information for visual localization. They trained NetVLAD on semantic maps with triplet loss.

Previous work mentioned above used semantic information to eliminate unreliable regions to guide where to focus, or visual localization is done with semantic features. Apart from previous works, we utilized semantic information at the pose verification step to validate initially retrieved images and the semantic similarity is learned via a self-supervised approach (detailed in Section 2.3).

Utilizing Depth Information for visual localization. Depth maps provide geometric clues of the scenes. Before the era of CNNs, camera re-localization was done on RGB-D image datasets with random forest-based approaches (Valentin et al. (2015);

Shotton et al. (2013)). In recent years, CNN-based approaches have been proposed for the pose refinement of queries images on the depth maps (Piasco et al. (2019a)). In (Piasco et al. (2019a)), they trained the encoder-decoder CNN model (which is similar to UNet) on depth maps. Initial query retrieval results are obtained with MAC (Razavian et al. (2016)) and NetVLAD (Arandjelovic et al. (2016)). Initial point clouds are generated with depth maps, and they are used to refine the pose of query images with the Iterative Closes Points (ICP) algorithm. In the follow-up study of Piasco et al. (2019b), the most similar candidate is retrieved with a 2D-based feature extraction approach (NetVLAD, Arandjelovic et al. (2016)), and the initial camera pose of query images is refined on dense depth maps with Perspective-n-Learned-Points (PnP) algorithm. Dense maps were used 2D points matching in the 3D environment for query pose estimation. The most similar work to ours is proposed by Piasco et al. (2021). They improved RGB-based visual localization performance with geometric information of the scene provided by the depth maps for the long-term visual localization. They proposed a CNN architecture that learns to deconstruct depth maps of input image. This idea is similar to Hallucination CNN model (Hoffman et al. (2016)). The proposed model simultaneously learns to extract depth and RGB information of a given input. They trained the model with triplet margin loss (Arandjelovic et al. (2016)) on depth and RGB images, at test time, they only used RGB images.

2.3. Contrastive Learning

History of contrastive learning date back to the 1990s, but it has recently gained popularity because of enormous success in computer vision (Le-Khac et al. (2020)). To train the model in supervised manner, considerable amount of annotated data is needed, which is not easy to obtain since it is a labor-intensive task. With the help of self-supervised learning, we can train models for the pre-task on unlabeled data. In self-supervised learning, the model is trained on pseudo-labels that are generated from the part of the input data. We use supervised training loss for the pretext task since positive pairs are part of the anchors. This training helps us to learn embedding space representation. Performance on the pretext task is not that important because, for most cases, we train our model with self-supervised manner on a big dataset and fine-tune part of it with a small amount of labeled dataset. There are several pretext tasks (e.g instance dis-

crimination, image colorization, image in-painting) that can be contribute to downstream tasks.

Dosovitskiy et al. (2014) used unsupervised learning for exempler-based classification. Positive pairs are created with data augmentation methods, such as color jitter, rotation, random crop, rotation. Applying data augmentation to whole dataset, and learning similarity between positive pairs and dissimilarity between negative ones is called instance discrimination Wu et al. (2018). In (Gidaris et al. (2018)), each image is rotated 90° and CNN model is trained on those images with self-supervised learning to estimate degree of rotation. This problem can be seen as four class classification. This learned representation can be used for object detection as a final task. Zhang et al. (2016) trained CNN on image pairs that consists of gray-scale and colorized version of the same image. By doing so, proposed model learns to colorize gray-scale images. This latent space can be useful for various downstream tasks.

Contrastive learning approaches are able to learn good enough embedding space representation with self-supervised training. Data augmented version of two sample are fed to the Siamese CNN, and after the training, the goal is to learn such embedding space where the positive sample stays close, and the negative ones are far away. SimCLR (Chen et al. (2020)) and MoCo (He et al. (2020)) use negative sample together with positives ones during the training. Unlike using positive and negative samples during the training, there are some methods (Chen and He (2021); Grill et al. (2020)) only use positive pairs (data augmented version of the same sample) for the training. Experimental results in previous studies showed that many computer vision research areas (e.g. semantic segmentation, image classification) benefit of contrastive learning.

In (Orhan et al. (2022)), we represented semantic information with self-supervised learning approach (SimCLR, Chen et al. (2020)). Unlike a labeled dataset, we can easily obtain an unlabeled dataset to train SimCLR on semantic masks or depth maps. Hence, we collected database images in different location of Pittsburgh, PA. and estimated their semantic masks with well-performing CNN model (Sun et al. (2019)). We used the self-supervised contrastive learning approach (SimCLR). Positive pairs are generated with random crop and rotation, and negative ones are collected from different parts of the city. An example of positive and negative pairs are illustrated in Figure 2.3. We extended our previous (Orhan et al. (2022)) with depth and multi-modal features in Section 5.

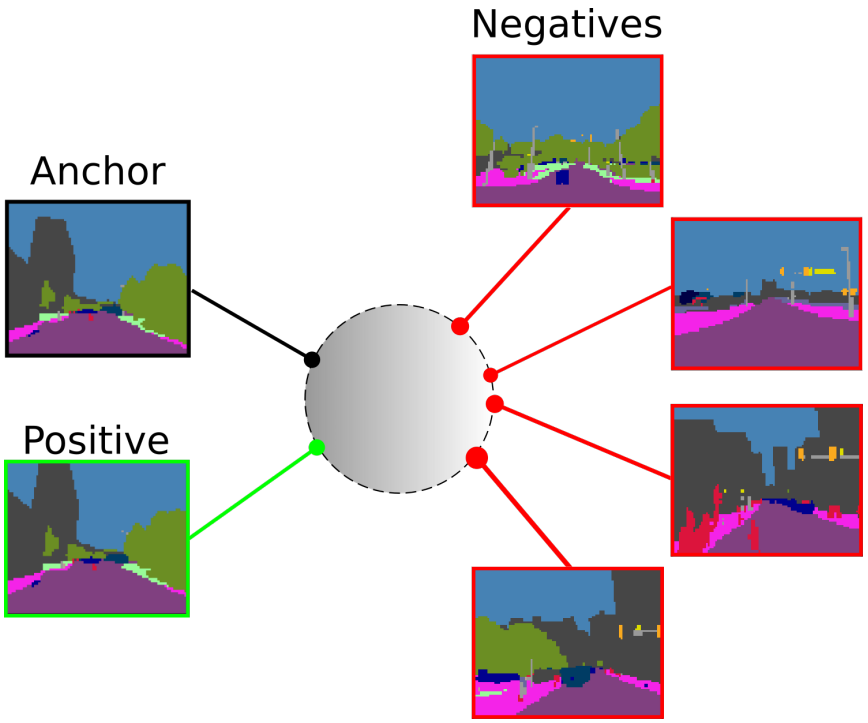


Figure 2.3. An example scenario is to learn semantic representations with the self-supervised learning approach (SimCLR). The positive pairs are generated with random crops and random rotations, and negative ones are collected from different parts of the city.

CHAPTER 3

SEMANTIC SEGMENTATION OF OUTDOOR PANORAMIC IMAGES

Full omnidirectional cameras can capture 360-degree of view with a single pose. Omnidirectional views are generally represented using equirectangular projection. Spatial coordinates are proportional to the sphere's longitude and latitude (same distance on the image corresponds to equal amount of viewing angle, e.g. 50 pixels = 32deg FOV). Unfortunately, it heavily suffers from distortion moving towers to the poles of the sphere. Spherical distortion degrades performance of computer vision approaches because most of them are optimized for perspective views. To tackle the spherical distortion, Coors et al. (2018) and Fernandez-Labrador et al. (2020) proposed spherical and equirectangular convolution respectively. The main idea in ((Coors et al. (2018); Fernandez-Labrador et al. (2020)) is that offset of the grids is calculated beforehand regarding the spherical distortion, and convolution operation is done with these offsets. We utilized equirectangular convolution (Fernandez-Labrador et al. (2020)) in our work and proposed an equirectangular version of UNet model. Although semantic segmentation was done on indoor panoramic and synthetic images, we conducted experiments on outdoor real panoramic images for the first time. Our main contribution is as follows: we developed UNet-equiconv, which is an equirectangular version of UNet. We replaced convolution layers with equirectangular convolution to tackle with spherical distortion, and we released publicly available one of the first outdoor panoramic image dataset for semantic segmentation.

3.1. Method

We introduced UNet-equiconv to alleviate the spherical distortion effect. It is an equirectangular version of UNet (Ronneberger et al. (2015)). We replaced each convolution layer with its equirectangular version. We illustrated the architecture of UNet-equiconv in Figure 3.1. After each convolutional layer, we applied batch normalization and ReLU (rectified linear units). For the simplicity, we did not illustrate these operations in the figure. We showed the repeated process in the figure with the 'x' symbol.

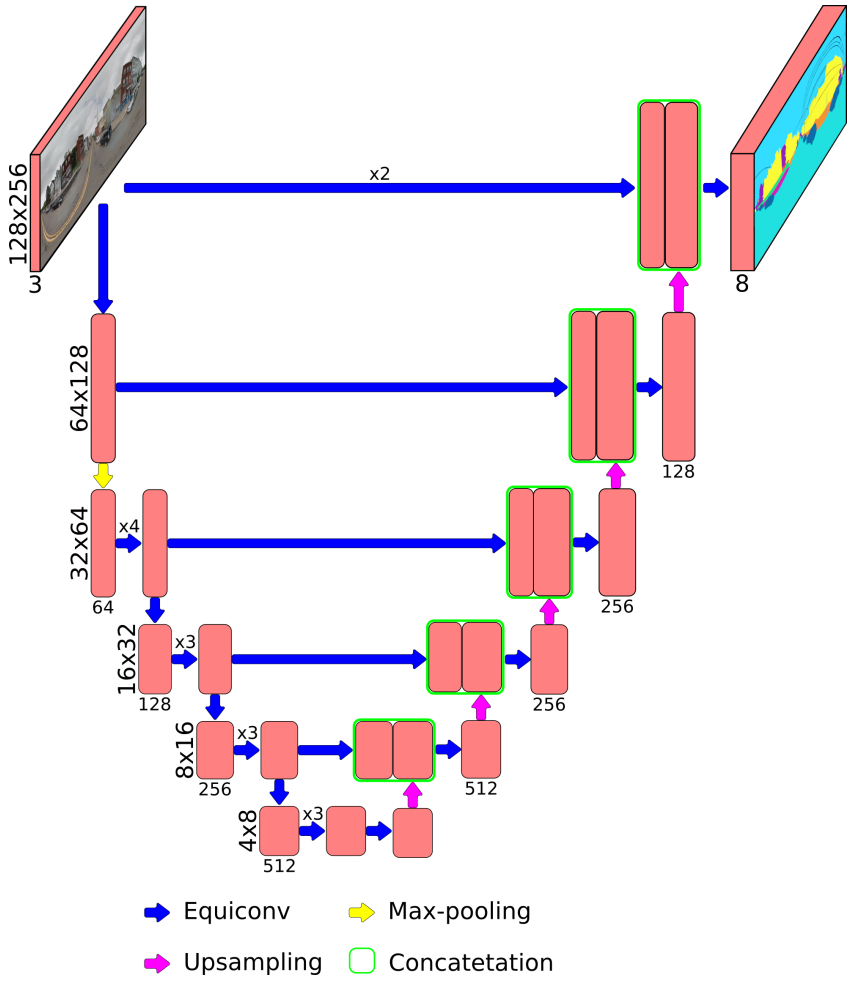


Figure 3.1. Architecture of UNet-equiconv.

3.1.1. Equirectangular Convolution

Previous works (Coors et al. (2018); Su and Grauman (2017); Tateno et al. (2018); Guerrero-Viu et al. (2020)) showed that explicitly modeling spherical distortion at the kernel level increased the semantic segmentation and object detection performance. The main idea in equirectangular convolution is that convolution kernels are moved onto the sphere rather than planar imagery. The convolution kernel is moved onto the sphere, and its location is calculated by spherical coordinates (θ and ϕ).

We generally use a square shape convolution kernel in CNN. In this section, we explain calculating spherical location from the convolution kernel at location p on the unit sphere. All transformation operations are illustrated in Fig. 3.2.

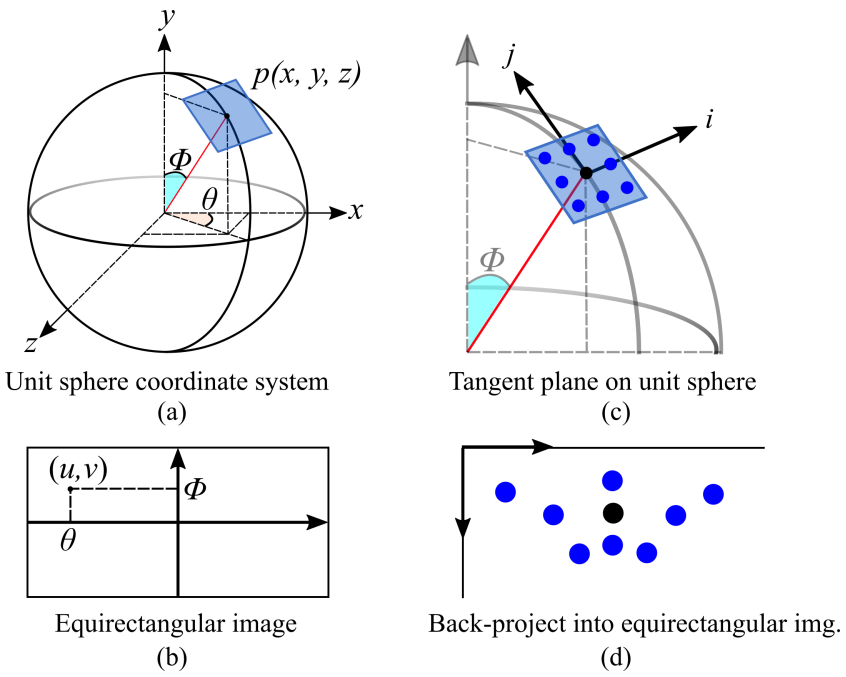


Figure 3.2. Distortion-aware convolution. Each pixel p in the equirectangular image is transformed into unit sphere coordinates, then the sampling grid is computed on the tangent plane in unit sphere coordinates, finally the sampling grid is back-projected into equirectangular image to determine the location of the distorted sampling grid.

We followed the instructions explained in by Fernandez-Labrador et al. (2020). As a first step, we defined $(u_{0,0}, v_{0,0})$ as the corresponding pixel location on the equirectangular image where we apply the convolution operation (i.e. the image coordinate where the center of the kernel is located). Then, these coordinates are transformed to longitude and latitude in the spherical coordinate system (Fig. 3.2b).

$$\theta_{0,0} = \left(u_{0,0} - \frac{W}{2}\right)\frac{360}{W}; \quad \phi_{0,0} = -\left(v_{0,0} - \frac{H}{2}\right)\frac{180}{H} \tag{3.1}$$

where θ and ϕ are in degrees and W and H are, respectively, the width and height of the equirectangular image in pixels.

Subsequently, the 3D coordinates for every element in the kernel (the tangent plane) is computed (Fig. 3.2c). When we consider a 3x3 kernel on the equator, kernel

element 3D coordinates are:

$$\hat{p}_{ij} = \begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \\ \hat{z}_{ij} \end{bmatrix} \tag{3.2}$$

where i and j are the horizontal and vertical indexes of a kernel element. 3D coordinates change as follows:

$$\hat{p}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},\ \hat{p}_{\pm 1,0} = \begin{bmatrix} \pm \tan \Delta_\theta \\ 0 \\ 1 \end{bmatrix},\ \hat{p}_{0,\pm 1} = \begin{bmatrix} 0 \\ \pm \tan \Delta_\phi \\ 1 \end{bmatrix} \tag{3.3}$$

where Δ_θ and Δ_ϕ are $360/W$ and $180/H$ in degrees respectively. These correspond to the angles covered by one pixel in the equator of the sphere. When the filter size is larger, angular coverage of kernel also decreases. Although we do not employ, lower resolution kernels can also be defined for wide angles. Readers can find detailed formulation on various kernel resolutions in Fernandez-Labrador et al. (2020).

We keep the kernel shape on the tangent plane fixed. When applying the filter at a different location (θ,ϕ), we rotate the points to the corresponding point of the sphere. We also project each point onto the sphere surface by normalizing the vectors:

$$p_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix} = R_y(\phi_{0,0}) R_x(\theta_{0,0}) \frac{\hat{p}_{ij}}{|\hat{p}_{ij}|} \tag{3.4}$$

where $R_a(\beta)$ stands for a rotation matrix of an angle β around a axis.

Finally, the rest of elements are back-projected to the equirectangular image domain (Fig. 3.2d). First, 3D kernel coordinates are transferred to latitude and longitude angles, which is called as the inverse gnomonic projection:

$$\theta_{ij} = \arctan\left(\frac{x_{ij}}{z_{ij}}\right); \quad \phi_{ij} = \arcsin\left(y_{ij}\right) \tag{3.5}$$

Then, converted to the original 2D equirectangular image domain:

$$u_{ij} = \left(\frac{\theta_{ij}}{360} + \frac{1}{2}\right) W; \quad v_{ij} = \left(-\frac{\phi_{ij}}{180} + \frac{1}{2}\right) H \tag{3.6}$$

Equirectangular convolution is a special form of deformable convolution (Dai et al. (2017)) layer. Unlike applying convolution operation to grid-like plane, offsets of the convolution layers are calculated regarding the spherical distortion.

Spherical distortion has a pattern in panoramic images. Hence, the offsets of the convolution layers are not learned with training but calculated regarding the equirectangular projection geometry. The offsets of the convolutional layers do not change moving horizontally, but they increase as the kernel moves toward the poles. It is illustrated in Figure 3.3.

Figure 3.3. The offsets of spherical kernel are visualized in three different positions. Kernel offset behaves as a regular grid on the equator. As the kernel is moved towards the poles, offset of the grid far apart. When the borders are exceeded, offsets move to the other side of the panoramic image.

3.2. Dataset

Most of the publicly available semantic segmentation datasets are collected with narrow FOV cameras (Lin et al. (2014); Cordts et al. (2016); Brostow et al. (2009); Geiger et al. (2012)). Thus, releasing a 360-degree view semantic segmentation dataset can contribute to several computer vision applications. In (Orhan and Bastanlar (2022)), we released one of the first outdoor panoramic image dataset for semantic segmentation task. We hope that it will be helpful to the various research on computer vision.

3.2.1. Equirectangular Outdoor Panoramic Image Dataset for Semantic Segmentation

We released one of the first equirectangular panoramic datasets of outdoor images for semantic segmentation, called CVRG-Pano. Panoramic images were collected from Pittsburgh, PA. with Google Street View application[1]. CVRG-Pano comprises 600 equirectangular panoramic images with 20 semantic labels. Semantic classes are grouped into seven categories regarding their semantic associations. Semantic labels and categories are shown in Table 3.1. Pixel distribution of each category is shown in Figure 3.4. We divided the dataset into three sets: training, validation, and test. Training set consists of 446, validation set consists of 48, and test set consists of 76 images. CVRG-Pano can be downloaded from the following link: https://github.com/semihorhan/semseg-outdoor-pano. In Figure 3.5, we show an equirectangular panoramic image from the dataset with its semantic labels.

Table 3.1. Semantic classes and their categorical groups.

construction	building, wall, fence, bridge
sky	sky
object	traffic light, pole, traffic sign
person	person
nature	terrain, vegetation
vehicle	bus, motorcycle, truck, bicycle, car
flat	road, sidewalk, parking, ground

3.2.2. Semantic Mask Generation with well-performing CNN

Manual label annotation takes a lot of time and effort. As an alternative approach, we can automatically generated semantic mask of outdoor panoramic images with well-performing CNN models.

[1] iStreetView.com

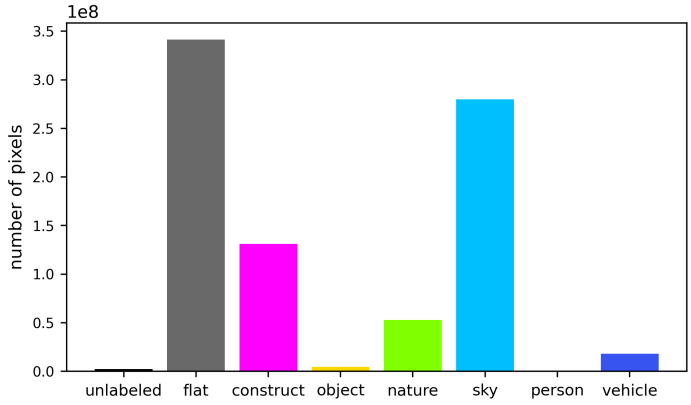


Figure 3.4. The total number of annotated pixels is shown on the y-axis, and their semantic labels on the x-axis.

Figure 3.5. An example image from the equirectangular outdoor panoramic image dataset with its semantic mask.

Cubemaps of panoramic images are generated as a first step. Later, we estimated the semantic labels of each cubemaps with a well-performing CNN model (Yuan et al. (2020)). The CNN model used to estimate the semantic mask of the cubemaps was trained on Cityscapes (Cordts et al. (2016)). As the last step, we generated a semantic mask for each panorama from the semantic mask of the cubemaps. The whole process is illustrated in Figure 3.6. In this way, we generated 504 panoramic semantic labels and grouped them into seven categories regarding Table 3.1.

Figure 3.6. The whole step of semantic mask generation for panoramic images. First, we generate cubemaps from panoramic images and estimate their semantic labels with a well-performing CNN model. Afterward, we project the semantics of masks of cubemaps to a panoramic image.

3.3. Experiments

We trained and fine-tuned UNet-stdconv and UNet-equiconv on the released dataset. We demonstrated with experiments effect of alleviating spherical distortion at the kernel level and weight initialization on semantic segmentation. As a deep learning framework, we used Pytorch 1.7.1 (Paszke et al. (2019)). We trained all CNN models on a computer with following specifications: Nvidia GeForce GTX 1080 GPU, Intel i7-8700K processor, and 16 GB memory.

3.3.1. Evaluation Metric

We used mean intersection over union (*mIoU*) in our experiments. It is a well-known evaluation metric for semantic segmentation (e.g. Cordts et al. (2016), Guerrero-Viu et al. (2020)). In Eq. 3.7, M represents the total number of classes, A_i is total number of ground truth pixels of class i, and $\hat{A}_i$ is predicted number of pixels for class i.

$$mIoU = \frac{1}{M} \sum_{i=1}^{M} \frac{A_i \cap \hat{A}_i}{A_i \cup \hat{A}_i} \tag{3.7}$$

3.3.2. Weight Initialization

We have a relatively small size of dataset to train our models. To evaluate contribution of the weight initialization, we evaluated performance of the model with several experiments using different pre-trained weights. We demonstrated weight initialization effect in Table 3.2. We fine-tuned all models on the CVRG-Pano (the released dataset). The test set consists of 76 equirectangular panoramic images with their semantic labels. Experimental results of different weight initialization are shown in Table 3.2. Experimental results showed that we get the best results with pre-trained Cityscape weights. We can also conclude that using pre-trained ImageNet increases performance since the low level of details (e.g., edges, curves) are shared between CVRG-Pano and ImageNet.

Table 3.2. Pre-training weight effect.

	Test *mIoU* scores
Training from scratch	0.610
ImageNet	0.634
Cityscapes	0.649

3.3.3. Standard vs. Equirectangular Convolution

In this section, to show the performance difference between standard and equirectangular convolution layers, we evaluated both models on outdoor panoramic image dataset. As a first step, we trained both models on Cityscapes and fine on the released panoramic image dataset. Experimental results in Table 3.3 showed that, UNet-equiconv perform better than UNet-stdconv. These results showed that handling spherical distortion at the kernel level helps to improve semantic segmentation of CNN model on outdoor panoramic images (Orhan and Bastanlar (2022)). Qualitative comparison are shown in Figure 3.7.

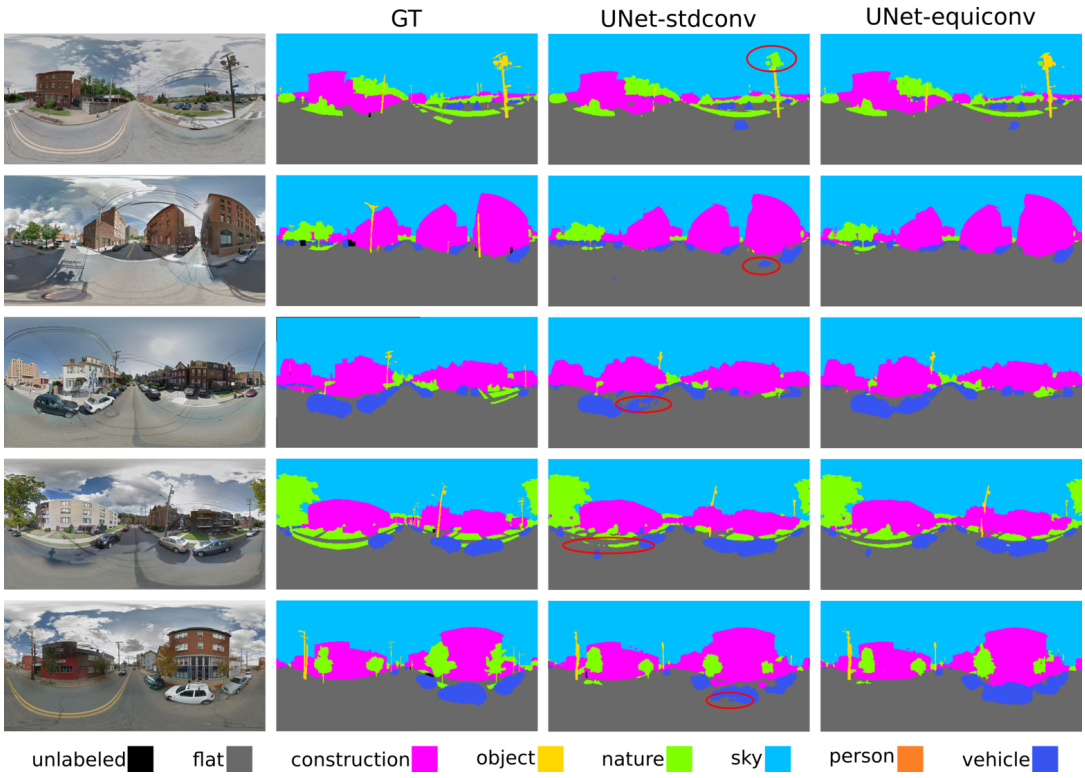


Figure 3.7. Example qualitative samples of UNet-stdconv and UNet-equiconv. Some semantic segmentation errors are highlighted with red circles.

Table 3.3. UNet-stdconv and UNet-equiconv performance on CVGR-Pano.

Model name	mean IoU	sky	construction	vehicle	nature	person	flat	objects
UNet-stdconv	0.65	0.98	0.83	0.69	0.79	0.13	0.96	0.17
UNet-equiconv	0.68	0.98	0.84	0.73	0.80	0.17	0.96	0.23

CHAPTER 4

SEARCHING PERSPECTIVE QUERY IMAGES IN A PANORAMIC IMAGE DATABASE WITHOUT GENERATING PERSPECTIVE VIEWS

In visual localization, an approximate location of query material is estimated within a visual map. Any city-scale visual localization system has to be robust against long-term changes (e.g illumination, seasonal, and structural). In our settings, we localize perspective (narrow FOV) query images in an equirectangular outdoor panoramic image database. A common way to search perspective query images in a panoramic image database is to generate virtual perspective gnomonic views. Even though we generate virtual perspective views from panoramic images, there might be a non-overlapping views problem between query and gnomic images. We can alleviate this problem by directly searching query images in a panoramic image database. Our main contribution in this chapter is that, while localizing perspective query images in the panoramic image database, instead of generating virtual perspective images using gnomic projection, we apply sliding window on the last convolution layer of CNN and directly localize query images in outdoor panoramic database. More detail of sliding window is explained in Section 4.2.1.

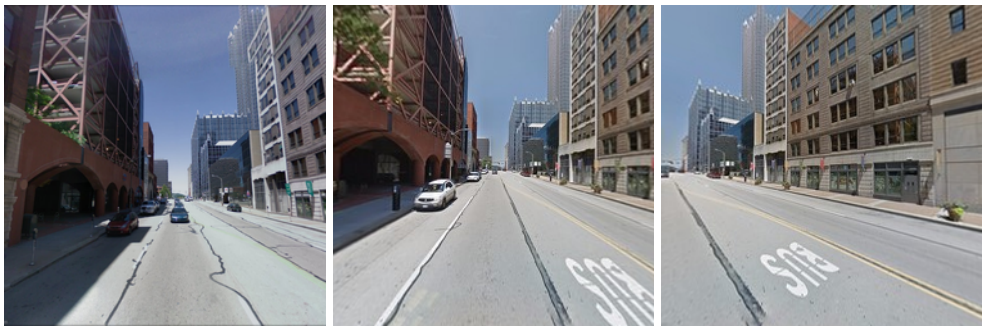
4.1. Dataset for Visual Localization

To localize perspective query images, we formed visual localization database consisting of panoramic outdoor images. In our dataset, query images are part of the UCF dataset (Zamir and Shah (2014)) and collected from Pittsburgh, PA, in 2014. We downloaded our panoramic image database from the same location as query images with Street View Download 360 application[1], collected in 2019. Images collected from different years cause long-term changes (Toft et al. (2018b); Naiming et al. (2018); Sattler et al. (2018)) such as seasonal, structural, and illumination changes.

[1] iStreetView.com

In our dataset, query images are collected from 123 locations, and panoramic database images are collected from 222 locations. There is at least one database image for each query within a five-meter distance. We generated 4, 8, and 12 gnomonic versions of our panoramic database. Our query set consist of 123x4=492 images; our 4, 8 and 12 gnomonic database consist of 222x4=888, 222x8=1176, and 222x12=2664 images respectively. In the 8-gnomonic database, each gnomonic view has 90-degree FOV and overlaps a 45-degree FOV with the next gnomonic image. In the 12-gnomonic image database, each gnomonic image overlaps 60-degree FOV with the next one.

The reason for generating 8 and 12 gnomonic datasets is that generating more gnomonic views with higher overlapping FOV alleviates the non-overlapping view problem. The best and the worst case scenarios for 4-gnomonic database are shown in Figure 4.1 and good overlap between query and 12 gnomonic database images are shown in Figure 4.2.



(a) (b) (c)

Figure 4.1. Query image appears in (a), best case scenario, 90° overlap between query and database images are in (b). Worst-case scenario in 4-gnomonic database, 45° overlapping in (c).

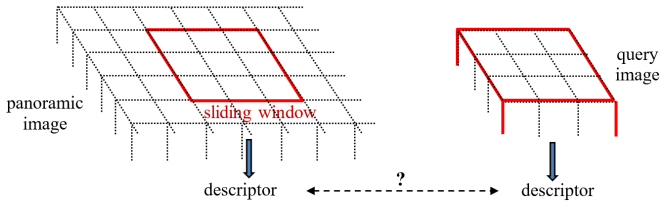
Figure 4.2. An example query and database pairs were collected from the same location. The panoramic database image is shown at the top left, and perspective images collected from the same location are shown at the top right. Each query image has 90° FOV and does not overlap with to next one. A generated 12-gnomonic database images to localize perspective query images are shown in the bottom two rows.

4.2. Methodology

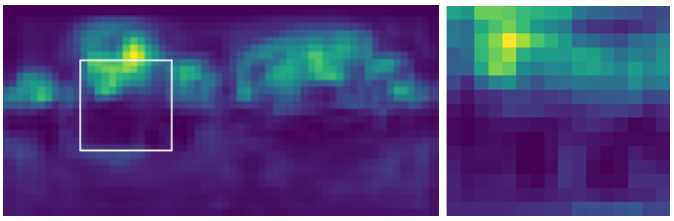
4.2.1. Searching perspective query image in an equirectangular panoramic image database

In our settings, perspective query images are directly localized in an equirectangular panoramic image database. To extract the features, we apply sliding windows to the last feature maps of CNNs. The images in our panoramic database are fully equirectangular, which means it covers 180° vertical views (bottom to top) and 360° horizontal views (left to right). Query images of our dataset are localized around the horizon. Therefore, we only search query images around the equator in panoramic images. The sliding window process is illustrated in Fig 4.3a.

(a) (b)



(c)



(d) (e)

	D. Blue	Red	Green	L. Blue	Yellow
Matching score	0.753	**0.850**	0.709	0.665	0.715

(f)

Figure 4.3. Equirectangular panoramic image and query images are shown in (a) and (b), respectively. Sliding windows applied on the panoramic image are highlighted with a different color in (a). Red sliding window correspond to the actual location of the query image. Feature maps extracted from panoramic and query images are illustrated in (c). Activation maps of panoramic and query images are visualized in (d) and (e). We get a similar activation pattern from the exact location of the query in the panoramic activation map. Feature similarity scores extracted with GeM pooling are shown in (f). We get the highest score from the exact location (red window) of the query image.

There are several studies based on pooling methods (Radenović et al. (2018); Razavian et al. (2016); Tolias et al. (2016)) to represent CNN features as compact and distinctive way. Let X be a WxHxK dimensional matrix corresponding to the last feature map of the convolution layer of CNN. X_i represents a single 2D activation plane in the feature map, where $i = 1, ..., K$ and $X_i(p)$ is the response at position p. If we select the maximum value in X_i (Eq.4.1), this results in a K-size feature vector for the image (MAC, Razavian et al. (2016)).

$$\mathbf{f} = [f_1, ...f_i, ...f_K]^T, \quad f_i = \max_{p \in X_i} X_i(p) \tag{4.1}$$

MAC pooling was improved in a follow-up study (Tolias et al. (2016)). R-MAC pooling is proposed by Tolias et al. (2016). It is based on applying the max-pooling method to different regions of feature maps with varying resolutions. Generalized-Mean (GeM) pooling layer is proposed by Radenović et al. (2018). It is one of the first trainable (differentiable) pooling layers. It learns pooling value after training rather than taking the maximum or average of the receptive field.

$$\mathbf{f} = [f_1, ...f_i, ...f_K]^T, \quad f_i = \left(\frac{1}{|X_i|} \sum_{p \in X_i} X_i(p)^{c_i} \right)^{\frac{1}{c_i}} \tag{4.2}$$

GeM pooling (Radenović et al. (2018)) behaves as a max-pooling (Razavian et al. (2016)) when $c_i \to \infty$. They used the Flickr dataset to train the GeM pooling layer. The Flickr dataset is composed of 7.4 million images. Flickr images consist of landmarks, top attraction of the cities, which fits well for the image retrieval task. We did not train the GeM pooling on our dataset because it is already trained on millions of images.

Another body of research focuses on visual localization in a city-scale map rather than top attractions or landmarks. Geo-tagged datasets, which are used for visual localization, consist of images collected from different times of the year with close-by locations. NetVLAD is proposed by Arandjelovic et al. (2016). The architecture of the NetVLAD includes several convolutions and trainable VLAD layers. It is trained triplets loss. Geo-tagged images have noisy labels. It is because images collected from the same location might not depict the same scene due to having different viewing angles of the cameras. To tackle noisy labels of geo-tagged images, SFRS is proposed by Ge et al. (2020). In (Ge et al. (2020)), an alternative training regimen is proposed. During the training, database images are divided into small parts, and similarity scores are estimated with these parts. By doing so, the positioning error of geo-tagged images is alleviated and CNN is trained

with more robust features. It outperformed the previous studies on the well-known visual localization benchmark datasets. In this chapter, we presented our results with R-MAC (Tolias et al. (2016)) pooling, SFRS (Ge et al. (2020)), and GeM (Radenović et al. (2018)) pooling.

We can give varying resolution of inputs to the same CNN model, which enables us to extract features from panoramic (1664x832 resolution) and query (500x400 resolution) images. Before conducting extensive experiments, we checked our hypothesis and visualized activation feature maps of panoramic image in Fig. 4.3d, and activation feature maps of query images in Fig. 4.3e. Fig. 4.3 shows that, we get the similar activation map patterns from exact location of query image in panoramic image database.

Some previous work on image classification and object detection; (Coors et al. (2018)), depth estimation and semantic segmentation (Tateno et al. (2018)); and 3D layout estimation (Fernandez-Labrador et al. (2020)) used distortion handling methods for panoramic images. However, it is not mandatory for us to use distortion handling methods since query images are around the equator of the sphere where they appear as perspective images. In addition, training a CNN model with a small dataset might not perform well compared to a CNN trained with a large dataset.

4.3. Experimental Results

In previous works (Sec. 2.2), a perspective query is localized in panoramic image database by generating 4 or 8 gnomonic virtual perspective views. Thus, we compared our method with gnomonic views. As a first step, we compare our method with 4-gnomonic projections (cubemaps, each having non-overlapping 90° FOV, cf. Fig. 2.2b). We assume the query is correctly localized if the query and the panoramic image and its gnomonic versions are collected from the same location (within 5 meter distance). We do not check the overlapping ratio between query and database images.

While searching perspective query images in the 4-gnomonic image database, accuracy mostly depends on overlapping FOV between query and database images. In the best case, there is a 100% overlap between FOV views; in the worst case scenario, there is only 45° FOV overlap between query and database images. We randomly chose starting point of the cubemaps (4-gnomonic images).

We conducted several experiments on the 4-gnomonic, 8-gnomonic, 12-gnomonic,

and panoramic datasets (details are explained in Section 4.1) with several feature extractors: R-MAC, GeM, SFRS. At first, we extracted features with R-MAC pooling (Tolias et al. (2016)). Table 4.1 shows the experimental results obtained with R-MAC pooling. The best result is obtained with the 12-gnomonic projection, which is 70.3%. We get competitive results with the proposed sliding window method (67.2%). 12-gnomonic projection and the proposed sliding window approach significantly outperform the 4-gnomonic projection (50.8%).

Experimental results obtained with GeM pooling (Tolias et al. (2016)) are shown in Table 4.2 and experimental results obtained with SFRS (Ge et al. (2020)) are shown in Table 4.3. The sliding window method outperformed the 4-gnomonic projection in all experiments, and we obtained competitive results compared to 8 and 12 gnomonic projections. With R-MAC and GeM poolings, the sliding window requires much less feature extraction time than 8 and 12 gnomonic projections (Table 4.4), but we do not observe a similar outcome with SFRS. This is due to the fact that Principal component analysis (PCA) is applied each feature map (window) to reduce dimension of VLAD features, which drastically increase the computation time. Nevertheless, the main advantage of sliding is that we can directly localize perspective images in a panoramic image database without generating its gnomonic views. We observe significant performance improvement in visual localization performance with SFRS. It is partly because SFRS is trained on Pittsburgh perspective images. All visual localization results are visualized in Figure 4.4 and qualitative results are shown in Figure 4.5.

4.3.1. Computation Cost

Descriptor extraction time of 4, 8, 12 gnomonic projections and sliding window are shown in Table 4.4. Table 4.4 shows that, with R-MAC and GeM pooling, the sliding window approach takes much less feature extraction time than 8 and 12-gnomonic projections while getting competitive results (see Table 4.1), but we did not observed the similar outcome with SFRS. The sliding window takes more time than 8 and 12 gnomonic projection with SFRS. This is due to the fact that Principal component analysis (PCA) is applied each feature map (window) to reduce dimension of VLAD features, which drastically increase the computation time. Nevertheless, the main advantage of sliding is that we can directly localize perspective images in a panoramic image database without generating its

Table 4.1. Visual Localization Results with R-MAC pooling (Tolias et al. (2016)).

Methods	**Recall@N**					
	N=1	**N=2**	**N=3**	**N=4**	**N=5**	**Avg=1-3**
4 gnomonic views (cubemaps)	0.508	0.589	0.630	0.674	0.699	0.576
8 gnomonic views	0.650	0.743	0.778	0.802	0.819	**0.724**
12 gnomonic views	0.703	0.784	0.813	0.833	0.851	**0.767**
Sliding Window (20x20 stride=3)	0.648	0.754	0.789	0.795	0.809	0.730
Sliding Window (20x20 stride=5)	0.652	0.732	0.778	0.799	0.811	0.721
Sliding Window (20x20 stride=7)	0.634	0.756	0.772	0.797	0.825	0.721
Sliding Window (24x24 stride=3)	0.672	0.740	0.787	0.807	0.821	0.732
Sliding Window (24x24 stride=5)	0.673	0.740	0.778	0.809	0.827	0.730
Sliding Window (24x24 stride=7)	0.663	0.740	0.780	0.799	0.807	0.728
Sliding Window (28x28 stride=3)	0.683	0.748	0.791	0.807	0.817	**0.740**
Sliding Window (28x28 stride=5)	0.675	0.750	0.791	0.801	0.821	**0.738**
Sliding Window (28x28 stride=7)	0.661	0.750	0.789	0.805	0.823	0.733
Sliding Window (32x32 stride=3)	0.669	0.736	0.776	0.803	0.815	0.727
Sliding Window (32x32 stride=5)	0.665	0.733	0.778	0.801	0.817	0.725
Sliding Window (32x32 stride=7)	0.659	0.740	0.785	0.797	0.811	0.728

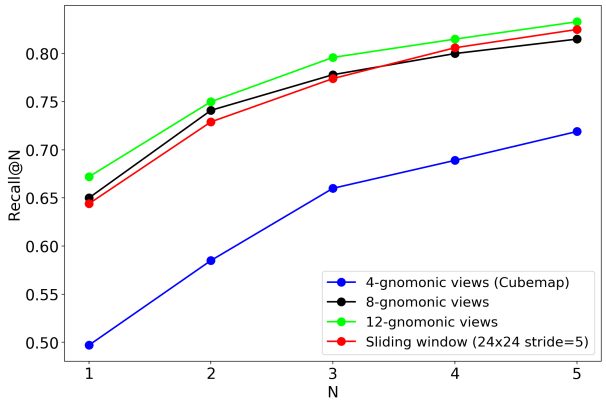
gnomonic views. All experiments are run on a computer with following specifications: NVIDIA GeForce GTX 1080 GPU, Intel i7-8700K processor, and 16 GB memory.

Table 4.2. Visual Localization Results with GeM pooling (Radenović et al. (2018)).

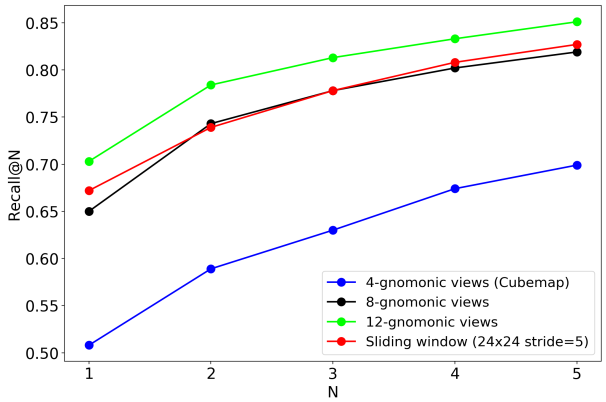
Methods	**Recall@N**					
	N=1	**N=2**	**N=3**	**N=4**	**N=5**	**Avg=1-3**
4 gnomonic views (cubemaps)	0.497	0.585	0.660	0.689	0.719	0.580
8 gnomonic views	0.650	0.741	0.778	0.800	0.815	**0.723**
12 gnomonic views	0.672	0.750	0.796	0.815	0.833	**0.739**
Sliding Window (20x20 stride=3)	0.642	0.729	0.770	0.799	0.819	0.714
Sliding Window (20x20 stride=5)	0.646	0.728	0.764	0.793	0.813	0.713
Sliding Window (20x20 stride=7)	0.628	0.713	0.766	0.805	0.813	0.703
Sliding Window (24x24 stride=3)	0.646	0.735	0.774	0.797	0.819	**0.719**
Sliding Window (24x24 stride=5)	0.644	0.730	0.774	0.807	0.825	**0.716**
Sliding Window (24x24 stride=7)	0.612	0.707	0.762	0.792	0.815	0.694
Sliding Window (28x28 stride=3)	0.630	0.711	0.750	0.799	0.819	0.697
Sliding Window (28x28 stride=5)	0.636	0.717	0.762	0.799	0.813	0.705
Sliding Window (28x28 stride=7)	0.618	0.722	0.754	0.789	0.805	0.698
Sliding Window (32x32 stride=3)	0.596	0.703	0.734	0.776	0.801	0.677
Sliding Window (32x32 stride=5)	0.581	0.697	0.746	0.774	0.795	0.675
Sliding Window (32x32 stride=7)	0.579	0.673	0.728	0.760	0.789	0.660

Table 4.3. Visual Localization Results with SFRS (Ge et al. (2020)).

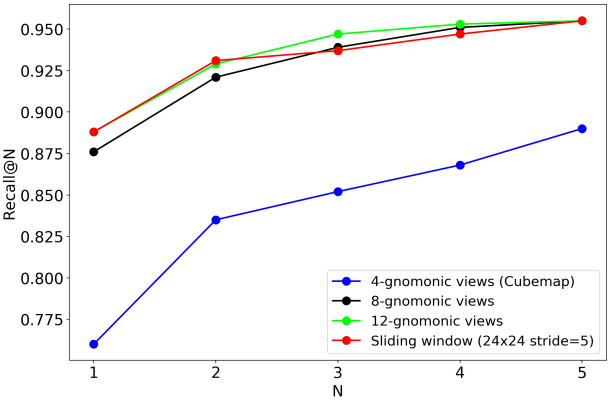
Methods	**Recall@N**					
	N=1	**N=2**	**N=3**	**N=4**	**N=5**	**Avg=1-3**
4 gnomonic views (cubemaps)	0.760	0.835	0.852	0.868	0.890	0.816
8 gnomonic views	0.876	0.921	0.939	0.951	0.955	**0.912**
12 gnomonic views	0.888	0.929	0.947	0.953	0.955	**0.921**
Sliding Window (20x20 stride=3)	0.886	0.927	0.935	0.943	0.951	0.916
Sliding Window (20x20 stride=5)	0.872	0.921	0.929	0.941	0.947	0.907
Sliding Window (20x20 stride=7)	0.866	0.917	0.929	0.941	0.949	0.904
Sliding Window (25x25 stride=3)	0.884	0.929	0.941	0.949	0.955	**0.918**
Sliding Window (25x25 stride=5)	0.884	0.929	0.935	0.947	0.949	**0.916**
Sliding Window (25x25 stride=7)	0.868	0.919	0.935	0.941	0.949	0.907
Sliding Window (28x28 stride=3)	0.870	0.911	0.927	0.935	0.939	0.902
Sliding Window (28x28 stride=5)	0.872	0.915	0.927	0.937	0.941	0.904
Sliding Window (28x28 stride=7)	0.860	0.917	0.925	0.939	0.943	0.900
Sliding Window (32x32 stride=3)	0.831	0.900	0.921	0.929	0.937	0.884
Sliding Window (32x32 stride=5)	0.839	0.894	0.915	0.927	0.931	0.883
Sliding Window (32x32 stride=7)	0.846	0.900	0.917	0.925	0.927	0.888



(a)



(b)



(c)

Figure 4.4. Visual localization result obtained with GeM pooling (a), R-MAC (b) and SFRS.

(a)

(b)

Figure 4.5. Two samples of query database pairs when 4 and 8 gnomonic projections fail, but the sliding window correctly localizes the query images. Query images are shown in the upper-right corner of (a) and (b). 8-gnomonic database images are shown in the bottom rows. In sample (a), there is a non-overlapping problem between query and database images and also an illumination difference. In sample (b), although the FOV of query and database images overlap almost perfectly, there are long-term changes (e.g. illumination and vegetation difference).

Table 4.4. Feature extraction time of gnomonic projections and sliding window.

Approach	**Database**	**Feature Extraction Time in sec.**
R-MAC	4-gnomonic view	29.34
	8-gnomonic view	51.38
	12-gnomonic view	77.22
	Panoramic (sliding window, stride=3)	44.74
	Panoramic (sliding window, stride=5)	34.55
	Panoramic (sliding window, stride=7)	29.89
GeM	4-gnomonic view	26.61
	8-gnomonic view	44.03
	12-gnomonic view	65.93
	Panoramic (sliding window, stride=3)	19.23
	Panoramic (sliding window, stride=5)	18.35
	Panoramic (sliding window, stride=7)	17.97
SFRS	4-gnomonic view	40.20
	8-gnomonic view	70.62
	12-gnomonic view	105.51
	Panoramic (sliding window, stride=3)	131.03
	Panoramic (sliding window, stride=5)	85.57
	Panoramic (sliding window, stride=7)	65.75

CHAPTER 5

MULTI-MODAL POSE VERIFICATION FOR LONG-TERM OUTDOOR VISUAL LOCALIZATION WITH SELF-SUPERVISED CONTRASTIVE LEARNING

Any city-scale visual localization system should overcome long-term appearance changes. Semantic information robust to seasonal, structural, an illumination changes and depth maps provide geometric clues. In this chapter, we utilized semantic and depth information for visual localization at the pose verification step. Semantic masks and depth maps of each image are automatically generated with well-performing CNN models and represented with a self-supervised contrastive learning approach (SimCLR, Chen et al. (2020)). To evaluate the semantic and depth contribution, we evaluated pose verification with experiments on the dataset explained in Section 4.1. Query images in our dataset consist of perspective images, and our database consists of panoramic images and their gnomonic versions. Our main contribution in this chapter is that we represented semantic and depth information with the self-supervised contrastive learning approached (SimCLR) and improved the state-of-the-art RGB-only model (SFRS) performance more than 1%.

5.1. Methodology

5.1.1. Visual localization of perspective query images in a panoramic image database

To measure the contribution of using semantic and depth information at the pose verification step, we conducted several experiments on our dataset that consists of perspective query and panoramic database images (details are in Section 4.1). We employed gnomonic projection and sliding windows approaches in our settings.

In long-term visual localization, images captured from close locations might not

depict the same scene due to seasonal and illumination differences. Several approaches (Tolias et al. (2016); Babenko and Lempitsky (2015); Ge et al. (2020); Arandjelovic et al. (2016)) have been proposed to tackle long-term changes (e.g seasonal difference, and structural changes). In addition the long-term changes, there might be a non-overlapping FOV problem between cameras. We can alleviate the overlapping problem by generating 12 gnomonic views from a panoramic image, but there might still exists non-overlapping view between query and gnomonic database images. In the worst-case scenario, view-point difference of query and 12-gnomonic database is $15°$; on average, it is $7.5°$. Recently SFRS has been proposed by Ge et al. (2020). They used a training regimen based on image to region similarity to alleviate noisy labels of geo-tagged images. By doing so, the proposed model is trained with more robust features. SFRS outperformed previous approaches on well-known visual localization datasets (Torii et al. (2013, 2015)).

5.1.2. Feature extraction on semantic masks and depth maps

As a first step, we estimate the semantic masks and depth maps of query and database images with CNN models (Sun et al. (2019); Ranftl et al. (2020)). The CNN model proposed for semantic segmentation is trained on the Cityscapes dataset (Cordts et al. (2016)), which consists of 30 semantic classes (e.g. sky, building, road). The depth model is trained on various depth datasets (Ranftl et al. (2020)).

Estimating the semantic similarity of the given two masks is a non-trivial task. SIFT and SURF like descriptor extractors do not exist to match semantic features. Images collected from the same location in different years might depict quite different scenes due to FOV different between cameras, or long-term changes. We tried to utilize CNN-based geometric alignment method (Rocco et al. (2017)) in our work to fit one semantic mask to another, but it did not work. Therefore, we proposed to represent the semantic and depth information with a trainable feature extractor that intrinsically learns small shifts and view point difference between masks. We utilized trainable feature extraction at the pose verification step of the RGB-only model (SFRS).

As a first step to test our hypothesis, we estimated the semantic similarity between query and database masks with pixel-wise similarity, which can be seen as a hand-crafted feature extractor. The drawback of this approach is that it is sensitive to small shifts and view point difference.

Pixel-wise Similarity. We compute pixel similarity between query and database masks as follows:

$$\textit{pixel-wise similarity} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} sim(Q_{(i,j)}, D_{(i,j)})}{m \cdot n} \tag{5.1}$$

where $sim(a, b)$ is equal to 1 if $a = b$, 0 otherwise. Q represents the query image's mask and D represents the database image's mask, both having size $m \times n$, $(i, j) \in \{1, ..., m\} \times \{1, ..., n\}$. A pixel is considered as a matching pixel if $Q_{(i,j)} = D_{(i,j)}$ and it increases similarity.

Self-supervised contrastive learning (trainable feature extractor). We represented the semantic and depth content of the scene with a self-supervised learning approach (SimCLR, Chen et al. (2020)) since semantic masks and depth maps can be easily generated by well-performing CNN models. In our work, we estimated semantic masks and depth maps of images with (Sun et al. (2019)) and (Ranftl et al. (2020)). We trained SimCLR on 3484 semantic masks and depth maps randomly collected from UCF (Zamir and Shah (2014)) dataset. Since the quality of automatically generated masks is enough to represent semantic and depth information, we did not need ground-truth labels.

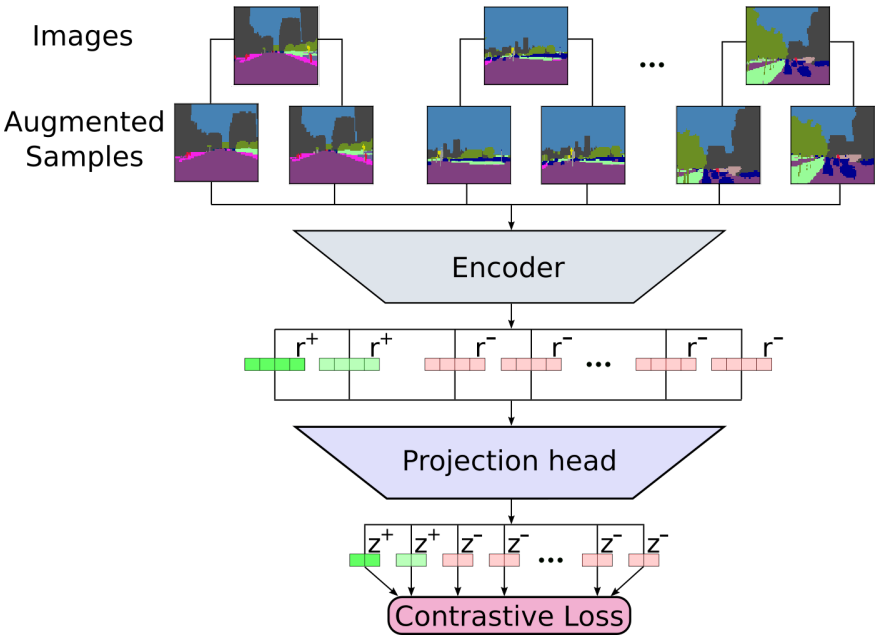


Figure 5.1. An example scenario where the CNN model is trained on semantically segmented masks with self-supervised contrastive loss.

We represented semantic masks and depth maps of the scenes with self-supervised contrastive learning approach (SimCLR, Chen et al. (2020)). ResNet-18 is used as an encoder of SimCLR. In our configuration, the encoder model produces $r = Enc(x) \in R^{512}$ size of features, projection model produces $z = Proj(r) \in R^{2048}$ size of features. This process is illustrated in Fig. 5.1. We used random resized crop and random rotation augmentation methods during the training, and resized semantic masks and depth maps to 64x80 pixels due to memory limitations. We set the random rotation degree as 3, and the random resized crop ratio as 0.6, which means that cropped area covers equal to or more than 60% of input masks. The data augmentation process of a semantic mask is illustrated in Fig. 5.1. We trained our model with contrastive loss shown in Eq. 5.2 (Khosla et al. (2020); Chen et al. (2020)). It is a categorical cross-entropy loss to learn the dissimilarity of the positive inputs amongst negative ones (inspired from InfoNCE, van den Oord et al. (2018)).

$$L^{self} = \sum_{i \in I} L_i^{self} = -\sum_{i \in I} \log \frac{\exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_{a(i)}/\tau)} \tag{5.2}$$

We randomly take N images from the semantic or depth mask dataset and generate augmented versions of the positive samples during the data loading step. Let $i \in I \equiv \{1...2N\}$ be the index of an arbitrary augmented sample, then $j(i)$ is the index of the other augmentation of the same original image. $\tau \in R^+$ is a scalar temperature parameter, $\cdot$ represents the dot product, and $A(i) \equiv I - \{i\}$. We call index i the anchor, index $j(i)$ is the positive, and the other $2(N-1)$ indices as negatives. The denominator has a total of $2N-1$ terms (one positive and $2N-2$ negatives).

Now the CNN model (SimCLR) trained on semantic masks or depth maps are ready to estimate the similarity between query and database images. We updated RGB-only methods scores (SimCLR) with semantic and depth similarity at the pose verification step (detail are in Section 5.1.3).

After self-supervised contrastive learning, we can fine-tune our self-supervised trained CNN model on a dataset that consists of query and database pairs. By doing so, the CNN model can learn the small shift and view point difference between query and database images. Therefore, we prepared a query-database pair dataset. It consists of 227 query-database pairs. Fine-tuning dataset is very small compared to the self-supervised training dataset, which consists of 3484 images. The standard procedure in contrastive learning literature is to replace the projection head, and fine-tune it on a small dataset

for object detection, image classification, and semantic segmentation tasks (final tasks). Since our pretext task and the final task are the same (instance discrimination which means estimating similarity between two masks), we partially or fully fine-tune our model and add a new projection head (explained in Section.5.2.2).

5.1.3. Updating RGB-only scores with semantic and depth similarity

At first, we normalize RGB-only, semantic, and depth similarity scores to $[-1, +1]$, and then update RGB-only scores with Eq. 5.3 for pose verification with semantic features, and we update RGB-only scores with Eq. 5.4 for pose verification with depth features. As last, Eq. 5.5 is used to update RGB-only scores with multi-modal features.

$$\textit{updated-score}_i = \textit{rgb-score}_i + W_s \cdot \textit{semantic-score}_i \tag{5.3}$$

$$\textit{updated-score}_i = \textit{rgb-score}_i + W_d \cdot \textit{depth-score}_i \tag{5.4}$$

$$\textit{updated-score}_i = \textit{rgb-score}_i + W_s \cdot \textit{semantic-score}_i + W_d \cdot \textit{depth-score}_i \tag{5.5}$$

where i is the index within the top K candidates for each query image, W_s is a weight coefficient of semantic similarity, and W_d is weight coefficient of depth similarity scores. We only update similarity scores of top K candidate database images retrieved by the RGB-only model for each query image. We set K as 10 in all experiments, and the example scenario is illustrated in Fig. 5.2. We update similar scores of neighbors of the top K candidate panoramic image (all gnomonic views) since neighbors of the most similar view can provide supportive information.

While updating similarity of RGB-only model, we select weight coefficient value (W) with regards to highest localization performance separately for depth and semantic features. Semantic, depth, and multi-modal weight coefficient effects are visualized in Figure 5.5, Figure 5.9, and Figure 5.11 respectively.

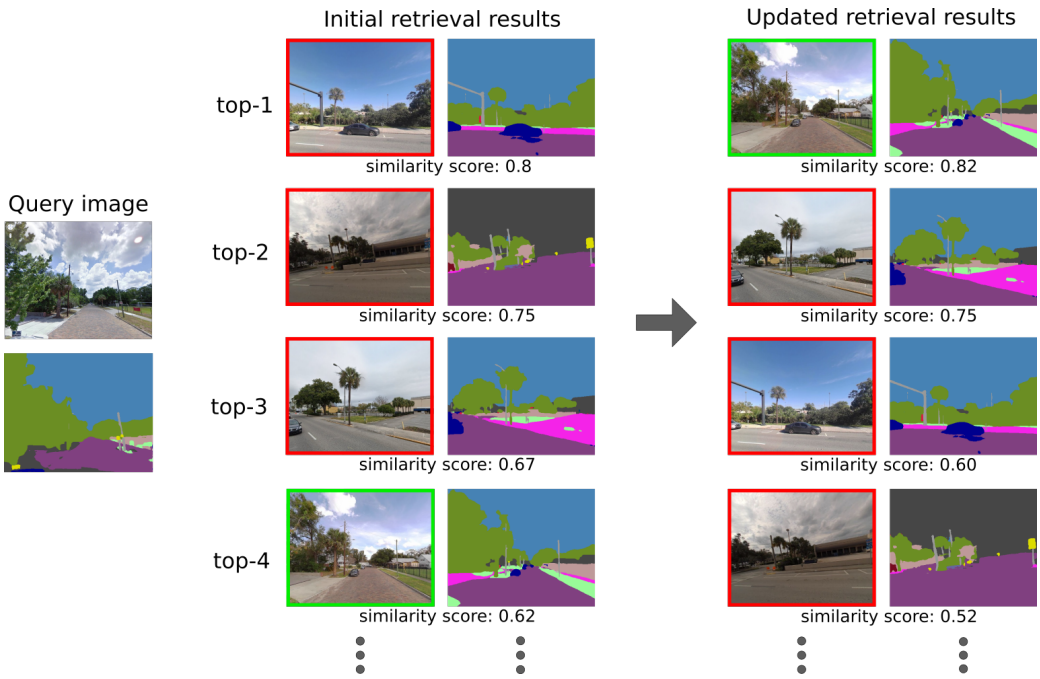


Figure 5.2. An example of visual localization results of the RGB-only model. RGB-only model fails to localize query image in a 12-gnomonic image database (middle column). A model that utilizes RGB and semantic information at the pose verification step correctly localizes the query image (right column).

5.2. Experimental Results

We trained our self-supervised contrastive learning model on 3484 images which is a subset of the UCF (Zamir and Shah (2014)) dataset, and we checked in case of shared images in the test set. Stochastic gradient descent (SGD) was used as optimizer, and initial learning rate (l_r) was set as 0.05. Batch size was set as 174 images ($2N$), and the temperature parameter (τ) was 0.07.

5.2.1. Pose Verification with Semantic Features

To show the visual localization performance of RGB-only model and utilizing semantic information at the pose verification step, we performed many experiments on

our dataset (details are in Section 4.1). We localized perspective query images in 8-gnomonic, 12-gnomonic, and panoramic image databases. Our query test consists of 492 images; 8 and 12 gnomonic databases consist of 1776 (222x8) and 2664 (222x12) images, respectively. The panoramic image database consists of 222 images. We evaluated visual localization performance of four approaches. In the first approach, visual localization is done with SimCLR (Chen et al. (2020)) on only RGB images. We called this approach as RGB-only model. In the second approach, we updated the similarity scores of the RGB-only model with pixel-wise similarity. In the third approach, we updated RGB-only scores with a trainable feature extractor (SimCLR) on 8 or 12 gnomonic databases, and the last approach is updating RGB-only scores with the sliding window. We demonstrated performance of the RGB-only and semantic pose verification methods with an ablation study on the labeled dataset, which consists of query-database pairs for each location. We provided these results in Section 5.2.2.

We compare the performance of RGB-only model and semantic pose verification with Recall@N metric. A query is considered correctly localized if the distance between any of the top N candidate database images is smaller than the distance threshold. We set the distance threshold as 5 meters in all experiments. We provided semantic pose verification results in Table 5.1. Experimental results show that pose verification improves the visual localization accuracy of RGB-only model for all methods. Semantic pose verification contributed to most (more than 1%) when visual localization is done on 12-gnomonic and panoramic image databases. It is because there is a small shift between queries and the most similar database images. The Sliding window approach (25x25 window, stride=3) performed similar to the 12-gnomonic projection. We visualized all the experimental results in Fig. 5.3. In addition to experimental results in Table 5.1, we conducted experiments with varying distance threshold. Experimental results showed that pose verification with SimCLR still outperformed the RGB-only and pixel-wise similarity scores, that is shown Fig. 5.3. We provide qualitative visual localization results in Fig. 5.4.

5.2.2. Additional Experiments with Semantic Features

So far, we have demonstrated experimental results training on the unlabeled dataset with self-supervised contrastive learning. This section provides additional experimental

Table 5.1. Visual localization results. Results are obtained with RGB-only and pose verification with semantic features. W_s corresponds to semantic weight coefficient.

Approaches	**Recall@N**					
	N=1	**N=2**	**N=3**	**N=4**	**N=5**	**Avg=1-3**
8-gnomonic (RGB-only)	0.876	0.921	0.939	0.951	0.955	0.909
8-gnomonic with SimCLR (W_s: 0.20)	0.888	0.921	0.941	0.945	0.949	0.917
12-gnomonic (RGB-only)	0.888	0.929	0.947	0.953	0.955	0.921
12-gnomonic with Pixel-wise (W_s: 0.20)	0.892	0.931	0.945	0.951	0.957	0.923
12-gnomonic with SimCLR (W_s: 0.20)	0.902	0.941	0.951	0.957	0.957	0.931
Sliding Window (RGB-only)	0.884	0.929	0.941	0.949	0.955	0.918
Sliding Window with SimCLR (W_s: 0.20)	0.9	0.933	0.949	0.953	0.959	0.927

results by fine-tuning our model on a labeled dataset consisting of query-database pairs for each location. In this section, we investigate the sensitivity of crop ratio and semantic weight coefficient (W) parameters.

We show fine-tuning and self-supervised training results in Table. 5.2. We fine-tuned our model on 227 query-database pairs. In the fine-tuning dataset, each location is represented with a mask of query-database pairs. We fine-tuned the last two dense layers, added an additional two dense layers, and we fine-tuned all layers of the CNN. Even though we followed various fine-tuning regimens, we still got the best results with self-supervised learning. Our fine-tuning dataset is quite small than the training dataset ($227 \ll 3484$). The reason we are not getting better results after fine-tuning is that our labeled dataset might be too small to learn viewpoint shifts between query database pairs, or our pretext task and the main task are the same (instance discrimination). In previous works where fine-tuning improved the results, projection heads are generally replaced for image classification or objection detection tasks. We also conducted several experiments with SimSiam (Chen and He (2021)), but it did not perform well with our configurations.

We conducted several experiments with different crop ratio parameters. Although recall scores fluctuate for where N=$\{2,..,5\}$, we get the highest scores when the crop ratio parameter is 0.6. Table 5.3 shows that visual localization performance decreases smaller and larger crop ratio parameter than 0.6. This is coherent with the reverse-U shape finding

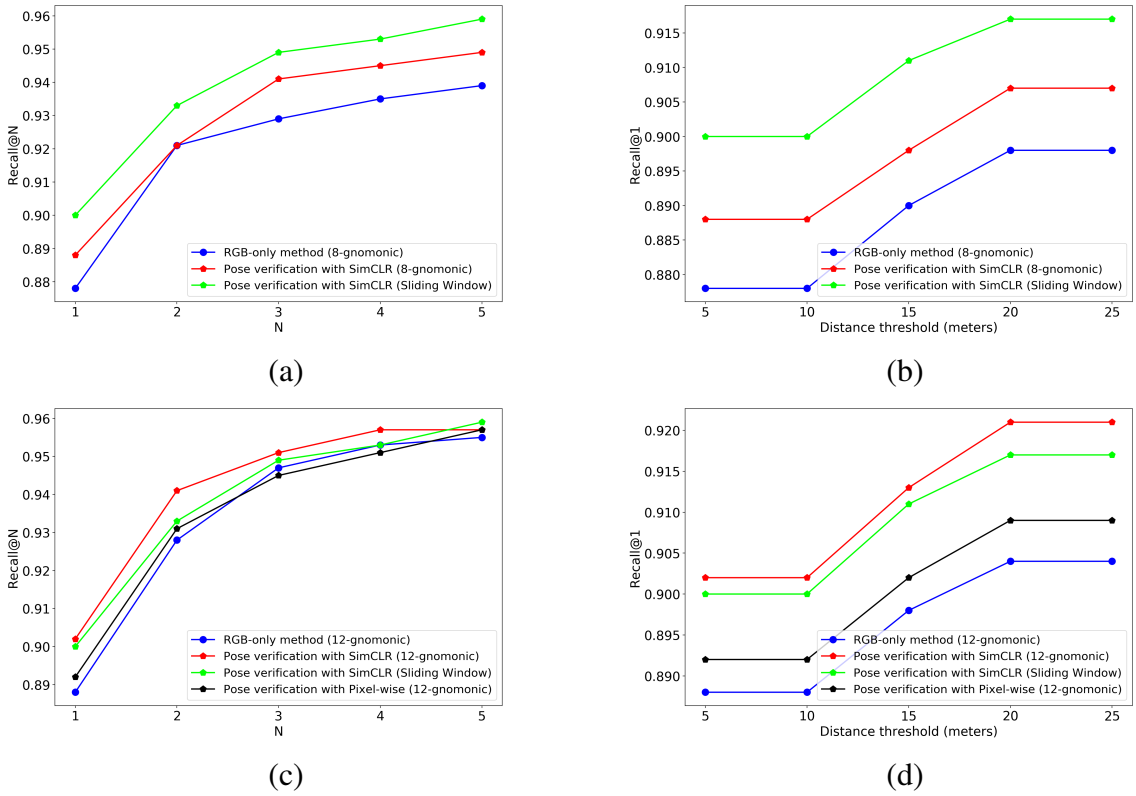


Figure 5.3. Recall@N scores of RGB-only and pose verification with semantic features for 8-gnomonic experiments (a), and 12-gnomonic and sliding window experiments (c). Recall@1 with different distance thresholds for 8-gnomonic experiments (b), and 12-gnomonic and sliding window experiments (d).

Table 5.2. Fine-tuning and self-supervised learning visual localization results. We set semantic weight coefficient (W) as 0.20 in all experiments.

Training Methods	**Recall@N**				
	N=1	**N=2**	**N=3**	**N=4**	**N=5**
Self-supervised training	**0.902**	0.941	0.951	0.957	0.957
Fine-tuning projection head	0.892	0.937	0.949	0.953	0.961
Adding two new dense layers	0.898	0.931	0.949	0.953	0.957
Fine-tuning all layers	**0.902**	0.939	0.949	0.955	0.959

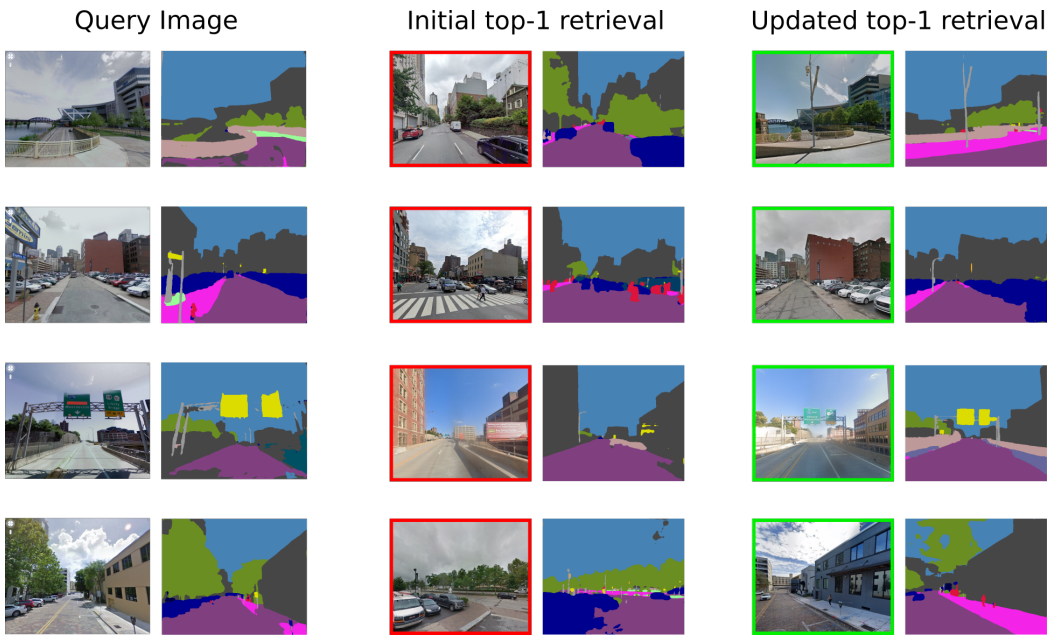


Figure 5.4. Example visual localization results when semantic pose verification improves the RGB-only scores. Query images are shown in the first column, and initial retrieval results are in the second column. Updated results with SimCLR appear in the third column. Pose verification with semantic features moved up the correct candidate when semantic information of the query and database images are similar (first two columns). Distinctive semantic classes in query and database masks (e.g., traffic signs) helped to improve the visual localization (third row). In some cases, pose verification on semantic masks where partial labeling error exists improved the visual localization(last row).

explained in (Tian et al. (2020)). High mutual information, such as when crop ratio is 0.9, does not provide enough information to the model because anchor and the positive pairs look almost identical. On the other hand, if there is low mutual information (e.g. crop ratio: 0.1), anchor and positive pairs depict almost different scenes. In both cases, the performance is lower than the peak value where it resided between them. The semantic weight coefficient is another parameter that influences performance. We showed semantic coefficient effect with different weights in Fig. 5.5. It is shown in Fig. 5.5 that we get the best performance when W are between 0.20 and 0.30.

Table 5.3. Visual localization results with different crop ratio parameters.

Crop Ratio	Recall@N				
	N=1	N=2	N=3	N=4	N=5
0.9	0.898	0.935	0.941	0.947	0.951
0.8	0.900	0.939	0.949	0.953	0.957
0.7	0.896	0.941	0.949	0.955	0.957
0.6	**0.902**	0.941	0.951	0.957	0.957
0.5	0.900	0.939	0.945	0.949	0.951
0.4	0.896	0.939	0.949	0.957	0.959
0.3	0.896	0.935	0.947	0.957	0.959

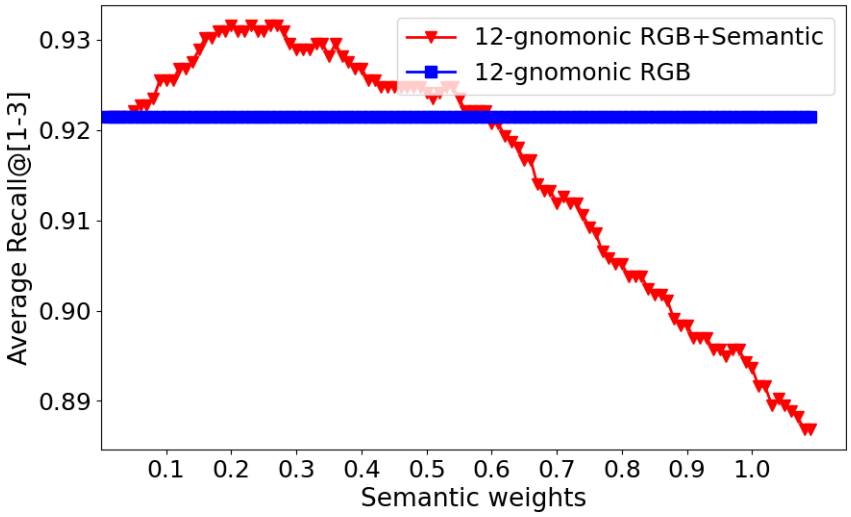
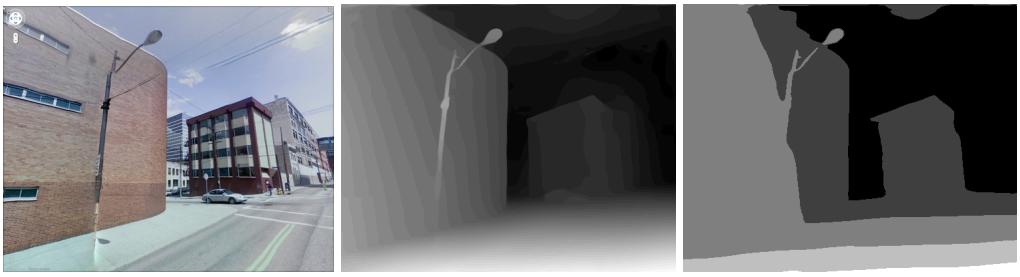


Figure 5.5. Visual localization results with average of Recall where N={1,...,3}.

5.2.3. Pose Verification with Depth Features

In addition to semantic information, we utilized depth information at the pose verification step. At first, we trained SimCLR on a dataset composed of depth maps estimated by (Ranftl et al. (2020)), but training loss did not decrease enough at the end of the training and the trained model did not perform well for the instance discrimination task. To simplify the learning process of instance discrimination from depth maps, we quantized depth maps into non-linear bins. An example estimated depth maps and its quantized version are shown in Figure 2.2.



(a) (b) (c)

Figure 5.6. An example RGB image is shown in (a), its estimated depth map is in (b) and quantized version of depth map is shown in (c).

We conducted several experiments on 8 and 12 gnomonic databases. Experiment results in Table 5.4 shows that utilizing depth information improved performance of the RGB-only model more than 1%, regarding Recall@1 metric. We plot experimental results in Figure 5.8. Qualitative examples where the pose verification with depth features improved the RGB-only model are provided in Figure 5.7. We evaluated the depth weight coefficient effect on visual localization. Fig. 5.5 shows the visual localization results with different depth weight coefficients (W). Depth features contribute the most when the depth weight coefficient (W) is between 0.30 and 0.40.

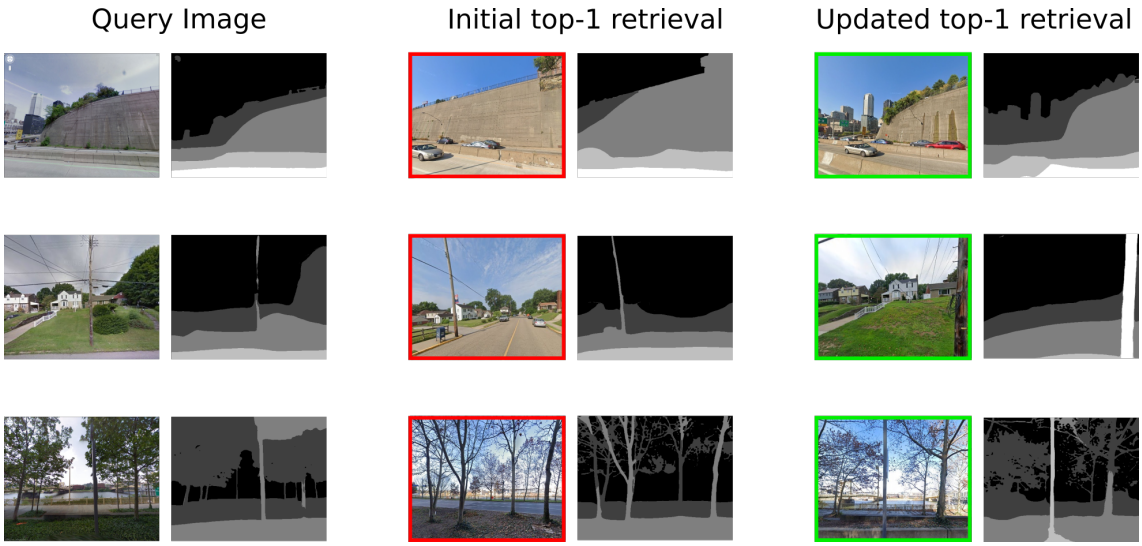


Figure 5.7. Example query images and their initial localization results with RGB-modal (middle column). Their updated results with depth features are shown in the last column. The red rectangle indicates the false localization of the query, and the green rectangle indicates correct localization.

Table 5.4. Visual localization results. Results are obtained with RGB-only and pose verification with depth features. W_d corresponds to depth weight coefficient.

Approaches	**Recall@N**					
	N=1	**N=2**	**N=3**	**N=4**	**N=5**	**Avg=1-3**
8-gnomonic (RGB-only)	0.878	0.921	0.929	0.935	0.939	0.909
8-gnomonic with SimCLR (W_d:0.30)	0.888	0.917	0.923	0.935	0.943	0.909
12-gnomonic (RGB-only)	0.888	0.928	0.947	0.953	0.955	0.921
12-gnomonic with SimCLR (W_d:0.30)	**0.9**	0.935	0.945	0.951	0.959	**0.927**

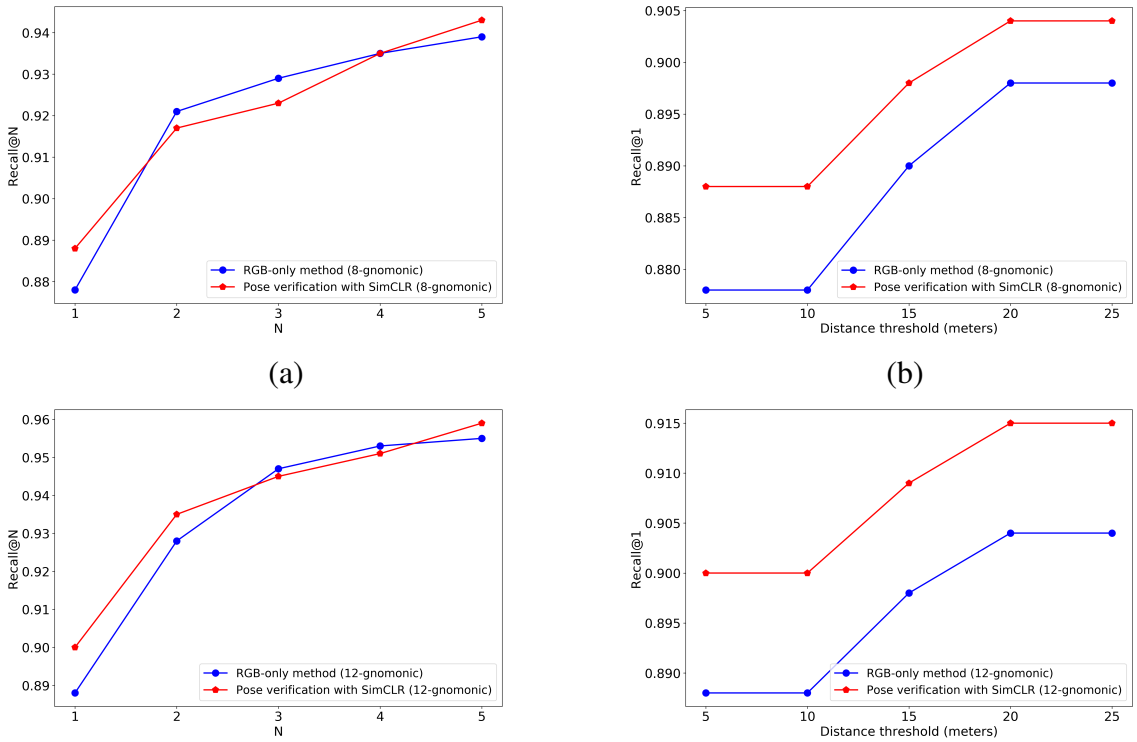


Figure 5.8. Recall@N scores of RGB-only and pose verification with depth features for 8-gnomonic experiments (a), and 12-gnomonic and sliding window experiments (c). Recall@1 with different distance thresholds for 8-gnomonic experiments (b), and 12-gnomonic and sliding window experiments (d).

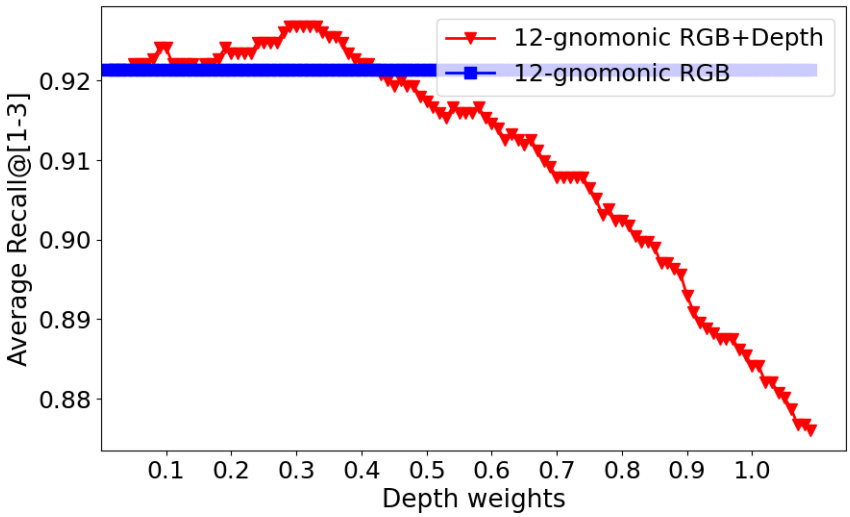


Figure 5.9. Visual localization results with average of Recall where N={1,...,3}.

5.2.4. Pose Verification with Multi-modal Features

RGB, semantic, and depth features can provide distinctive and complementary information about the same scene. Based on this idea, we combined RGB, semantic, and depth features to see if multi-modal visual localization with self-supervised learning is meaningful. Experimental results in Table 5.5 shows that multi-modal pose verification improved the RGB-only model performance more than 1%. Even though multi-modal visual pose verification improves the visual localization scores of RGB-only model, multi-modal visual localization is almost on par with the pose verification with semantic features (average Recall N={1,...3}=0.931).

Semantic and depth weight coefficient effect are visualized in Figure 5.10. Figure 5.11 shows that multi-modal visual localization perform the best when semantic weight is between 0.2 to 0.3 and depth weight is between 0.2 to 0.3.

Table 5.5. Visual localization results obtained RGB, depth, semantic and multi-modal features. W_s refers to semantic, W_d refers to depth weight coefficients.

Approaches	**Recall@N**					
	N=1	**N=2**	**N=3**	**N=4**	**N=5**	**Avg=1-3**
8-gnm. (RGB-only)	0.878	0.921	0.929	0.935	0.939	0.909
8-gnm. with SimCLR (W_s:0.20)	0.888	0.921	0.941	0.945	0.949	0.917
8-gnm. with SimCLR (W_d:0.30)	0.888	0.917	0.923	0.935	0.943	0.909
8-gnm. with SimCLR (W_s:0.30,W_d:0.20)	0.89	0.925	0.941	0.947	0.953	0.919
12-gnm. (RGB-only)	0.888	0.928	0.947	0.953	0.955	0.921
12-gnm. with SimCLR (W_s:0.20)	0.902	0.941	0.951	0.957	0.957	0.931
12-gnm. with SimCLR (W_d:0.30)	0.9	0.935	0.945	0.951	0.959	0.927
12-gnm. with SimCLR (W_s:0.30,W_d:0.20)	0.9	0.943	0.955	0.955	0.957	0.933

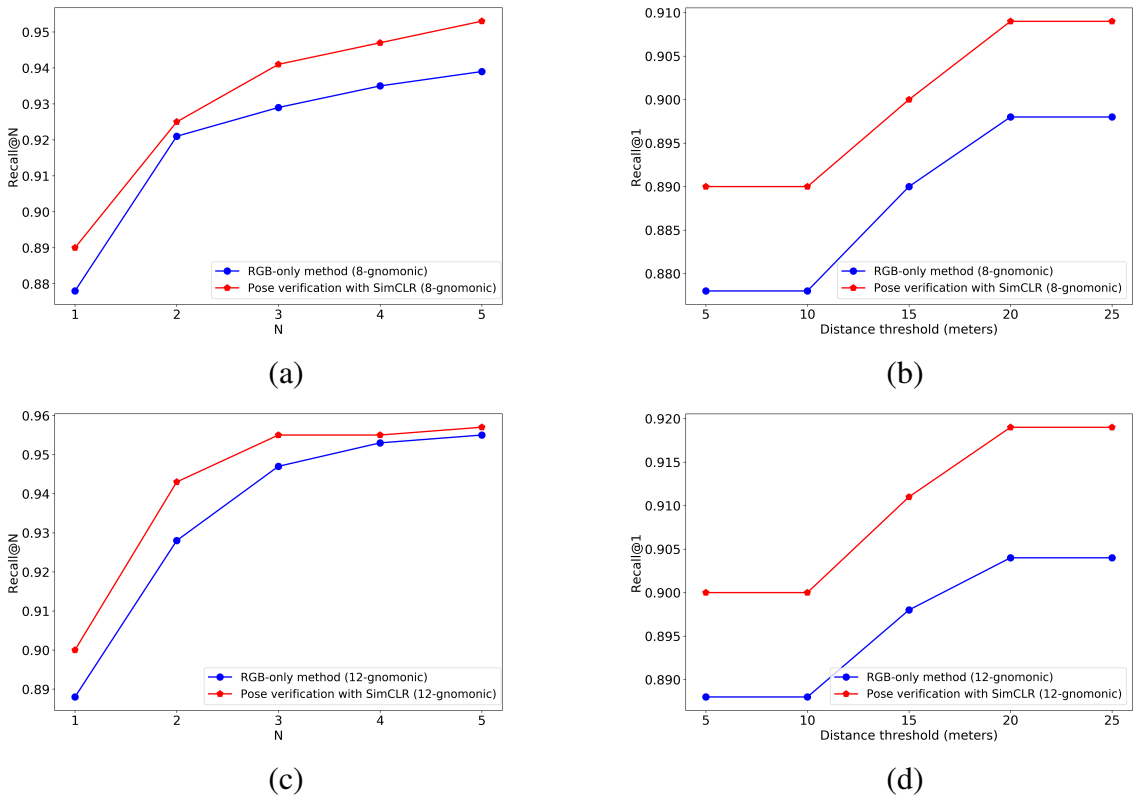


Figure 5.10. Recall@N scores of RGB-only and pose verification with multi-modal (semantic and depth together) features for 8-gnomonic experiments (a), and 12-gnomonic and sliding window experiments (c). Recall@1 with different distance thresholds for 8-gnomonic experiments (b), and 12-gnomonic and sliding window experiments (d).

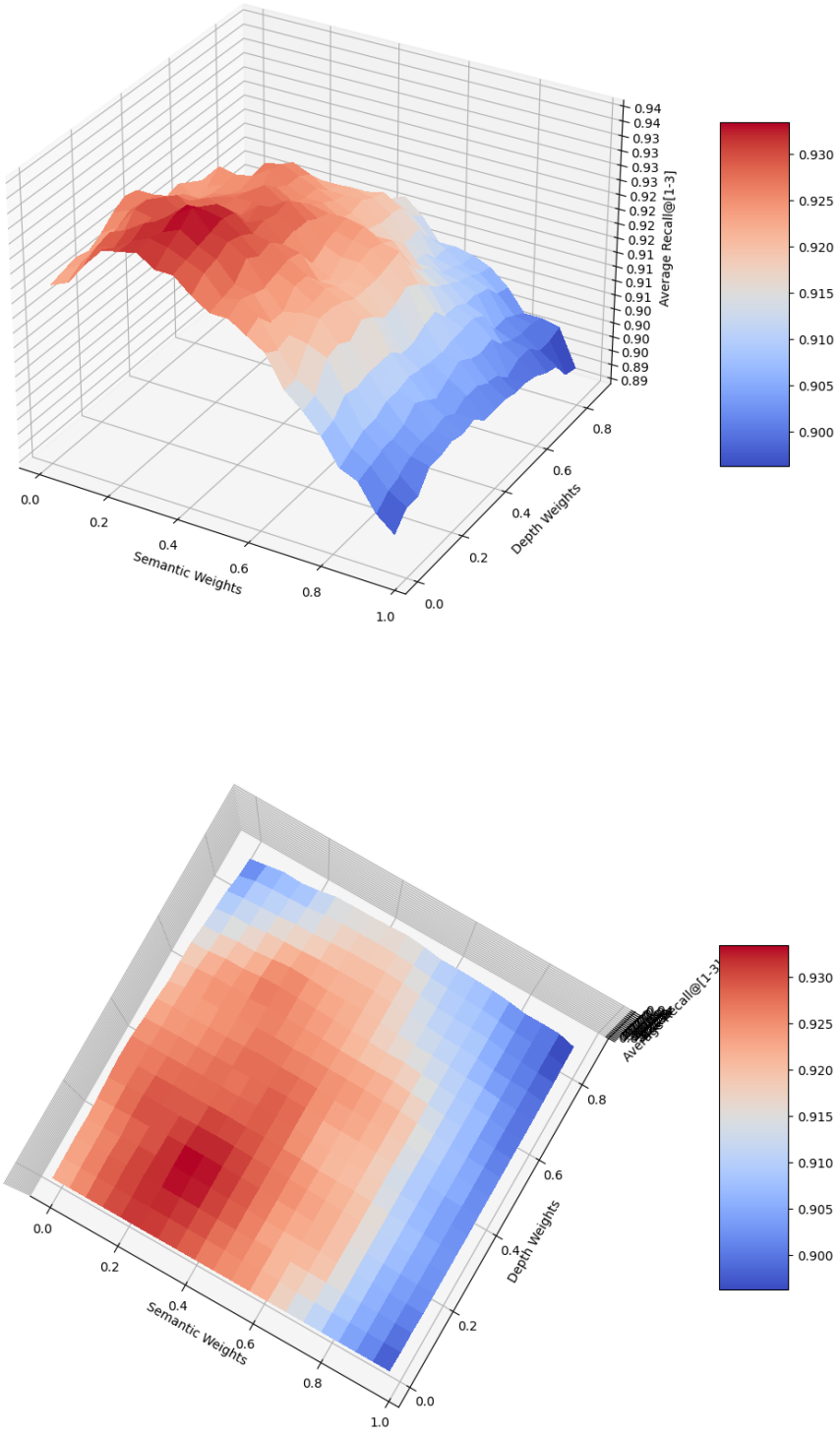


Figure 5.11. Pose verification with multi-modal features (semantic and depth together). Visual localization results are provided with average of Recall where N={1,...,3}.

CHAPTER 6

CONCLUSION

In this thesis, we localized perspective query images in an outdoor panoramic image database and its gnomonic views. In addition to RGB information, we utilized semantic and depth information for visual localization at the pose verification step.

Full omnidirectional (360-degree) views are generally represented in the planar space using equirectangular projection. In omnidirectional views, we capture 360-degree views with a single shot, but it comes with the cost of spherical distortion at the poles of the sphere. Objects which are close to the poles look quite different than they would appear in perspective views. In previous studies of CNNs, several methods (e.g. equirectangular convolution, spherical convolution) were proposed to tackle the spherical distortion. In the third chapter of the thesis, we developed an equirectangular version of UNet for semantic segmentation of outdoor panoramic images. Their convolution type is the only difference between the standard UNet and its equirectangular version (UNet-equiconv). Experimental results showed that UNet-equiconv performs better than its standard version (UNet-stdconv). We released one of the first outdoor panoramic image datasets for semantic segmentation. This dataset can be used for various computer vision research areas from autonomous driving to visual localization.

The distortion appears not only in panoramic images but also in any large FOV images, such as the fish eye. By explicitly modeling distortion patterns at convolution time, we expect to get a similar performance improvement also for other types of images.

In the fourth chapter of the thesis, we focused on localizing perspective query images in an outdoor panoramic image database. In previous works, a common way to search perspective query images in a panoramic image database is to generate virtual perspective views of panoramic images with gnomonic projection. Yet, a non-overlapping view problem might still exist between query and database images. We can alleviate this problem by generating more gnomonic views with a higher overlapping ratio with the next gnomonic view, but it increases the database size. To directly localize perspective query images within a panoramic image database, we proposed an alternative searching (visual localization) method. Instead of generating virtual perspective views from panoramic im-

ages, we apply a sliding window on the last convolution layer of CNNs. We prepared a dataset that consists of perspective queries and panoramic database images. We also generated 4, 8, and 12 gnomonic views of panoramic images. We compared the visual localization performance of the sliding window approach with gnomonic views. We used R-MAC, GeM, and SFRS as feature extractors. Experimental results show that the sliding window approach outperformed 4-gnomonic views, and we get competitive results compared to 8 and 12-gnomonic projections. With R-MAC and GeM pooling, feature extraction takes much less time with the sliding window compared to 8 and 12 gnomonic projection, but we did not observe a similar outcome with SFRS. With SFRS, the feature extraction time of the sliding window takes longer than 8-gnomonic projection. It is because PCA is applied to each extracted feature (windows), which drastically increases the computation time. The main advantage of the sliding window is that it can directly localize perspective query images in a panoramic images dataset (e.g. Google Street View) without generating its gnomonic views.

In visual localization systems, search time is more important than feature extraction time because searching is done online, unlike feature extraction. As future work, the search time of the sliding window and gnomonic view approaches can be evaluated on a big dataset. Thus, we can evaluate search time and performance gain with respect to the number of windows extracted from each panoramic image or the number of gnomonic views generated from each panoramic image.

In the fifth and last chapter of the thesis, we researched long-term visual localization on outdoor panoramic images. Every long-term visual localization system has to handle long-term changes (e.g. illumination, seasonal, and structural). Semantic information about the scene is more robust to changes (e.g. surface of the building), and depth maps provide geometric information, which can be used as a complementary modality, in addition to RGB features. Based on these ideas, we utilized semantic and depth information for long-term visual localization at the pose verification step. We represented semantic masks and depth maps with self-supervised contrastive learning (SimCLR). We conducted several experiments to evaluate the performance difference between the RGB-only model and the pose verification with semantic and depth features. Experimental results showed that pose verification with semantic, depth, and multi-modal (semantic and depth together) features improved the RGB-only model performance. Depth features contributed the least, and we obtained similar visual localization performance with semantic and multi-modal features.

Other modalities (e.g. surface normal) can also be represented with self-supervised learning. In the last chapter, we tried to utilize normal surface normal in addition to the semantic features, but the visual localization performance of the RGB-only model did not improve. We did get similar semantic information in urban areas where buildings are generally surrounded by roads. In such areas, semantic information is not distinctive, and the contribution of semantic information was little to no.

As future work, utilizing semantic and depth information at the pose verification step can be evaluated on more challenging and bigger datasets, such as the ones that consist of day and night images, where drastic illumination changes occur, or images that are taken under foggy or snowy weather.

REFERENCES

Arandjelovic, R., P. Gronat, A. Torii, T. Pajdla, and J. Sivic (2016). Net VLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307.

Babenko, A. and V. Lempitsky (2015). Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*.

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.

Bastanlar, Y. and S. Orhan (2022). Self-supervised contrastive representation learning in computer vision. In P. C. M. Travieso-Gonzalez (Ed.), *Applied Intelligence - Annual Volume 2022*, Chapter 14. IntechOpen.

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

Brostow, G. J., J. Fauqueur, and R. Cipolla (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters 30*(2), 88–97.

Chen, T., S. Kornblith, M. Norouzi, and G. Hinton (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607.

Chen, X. and K. He (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758.

Chen, Z., Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao (2022). Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*.

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

Cheng, R., K. Wang, S. Lin, W. Hu, K. Yang, X. Huang, H. Li, D. Sun, and J. Bai (2019). Panoramic annular localizer: Tackling the variation challenges of outdoor localization using panoramic annular images and active deep descriptors. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 920–925.

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

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

Cinaroglu, I. and Y. Bastanlar (2022). Long-term image-based vehicle localization improved with learnt semantic descriptors. *Engineering Science and Technology, an International Journal (JESTECH) 35*.

Coors, B., A. P. Condurache, and A. Geiger (2018). Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 518–533.

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.

Costea, A. D. and S. Nedevschi (2016). Semantic channels for fast pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2360–2368.

Dai, J., H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer*

vision, pp. 764–773.

Demiröz, B. E., A. A. Salah, Y. Bastanlar, and L. Akarun (2019). Affordable person detection in omnidirectional cameras using radial integral channel features. *Machine Vision and Applications 30*(4), 645–655.

Deng, L., M. Yang, Y. Qian, C. Wang, and B. Wang (2017). Cnn based semantic segmentation for urban traffic scenes using fisheye camera. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 231–236. IEEE.

Dosovitskiy, A., J. T. Springenberg, M. Riedmiller, and T. Brox (2014). Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems 27*.

Dvornik, N., K. Shmelkov, J. Mairal, and C. Schmid (2017). Blitznet: A real-time deep network for scene understanding. In *Proceedings of the IEEE international conference on computer vision*, pp. 4154–4162.

Fernandez-Labrador, C., J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. J. Guerrero (2020). Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters 5*(2), 1255–1262.

Ge, Y., H. Wang, F. Zhu, R. Zhao, and H. Li (2020). Self-supervising fine-grained region similarities for large-scale image localization. In *European Conference on Computer Vision*, pp. 369–386.

Geiger, A., P. Lenz, and R. Urtasun (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE.

Gidaris, S., P. Singh, and N. Komodakis (2018). Unsupervised representation learning by predicting image rotations. In *ICLR 2018*.

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.

Grill, J.-B., F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko (2020). Bootstrap your own latent: A new approach to self-supervised learning.

Guerrero-Viu, J., C. Fernandez-Labrador, C. Demonceaux, and J. J. Guerrero (2020). What's in my room? object recognition on indoor panoramic images. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 567–573. IEEE.

Hansen, P. and B. Browning (2015). Omnidirectional visual place recognition using rotation invariant sequence matching. *Technical Report*.

He, K., H. Fan, Y. Wu, S. Xie, and R. Girshick (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hoffman, J., S. Gupta, and T. Darrell (2016). Learning with side information through modality hallucination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 826–834.

Huang, J.-Y., S.-H. Lee, and C.-H. Tsai (2016). A fast image matching technique for the panoramic-based localization. In *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*.

Iscen, A., G. Tolias, Y. Avrithis, T. Furon, and O. Chum (2017). Panorama to panorama matching for location recognition. In *ACM International Conference on Multimedia Retrieval (ICMR)*.

Jégou, H., F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid (2011). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*(9), 1704–1716.

Kampffmeyer, M., A.-B. Salberg, and R. Jenssen (2016). Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep con-

volutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1–9.

Karkus, P., A. Angelova, V. Vanhoucke, and R. Jonschkowski (2020). Differentiable mapping networks: Learning structured map representations for sparse visual localization. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Khosla, P., P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan (2020). Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Le-Khac, P. H., G. Healy, and A. F. Smeaton (2020). Contrastive representation learning: A framework and review. *IEEE Access*.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer.

Long, J., E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Lourenço, M., J. P. Barreto, and F. Vasconcelos (2012). srd-sift: keypoint detection and matching in images with radial distortion. *IEEE Transactions on Robotics 28*(3), 752–760.

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.

Lu, H., X. Li, H. Zhang, and Z. Zheng (2013). Robust place recognition based on omnidirectional vision and real-time local visual features for mobile robots. *Advanced Robotics 27*(18), 1439–1453.

Mao, J., T. Xiao, Y. Jiang, and Z. Cao (2017). What can help pedestrian detection? In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3127–3136.

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

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

Naiming, Y., T. Kanji, F. Yichu, F. Xiaoxiao, I. Kazunori, and I. Yuuki (2018). Long-term vehicle localization using compressed visual experiences. In *21st International Conference on Intelligent Transportation Systems (ITSC)*.

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

Noh, H., S. Hong, and B. Han (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528.

Orhan, S. and Y. Bastanlar (2021). Efficient search in a panoramic image database for long-term visual localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

Orhan, S. and Y. Bastanlar (2022, Apr). Semantic segmentation of outdoor panoramic images. *Signal, Image and Video Processing 16*(3), 643–650.

Orhan, S., J. J. Guerrero, and Y. Bastanlar (2022). Semantic pose verification for outdoor visual localization with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, pp. 3989–3998.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Peng, S., Y. Liu, Q. Huang, X. Zhou, and H. Bao (2019). Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570.

Philbin, J., O. Chum, M. Isard, J. Sivic, and A. Zisserman (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Piasco, N., D. Sidibé, C. Demonceaux, and V. Gouet-Brunet (2019a). Geometric camera pose refinement with learned depth maps. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2561–2565. IEEE.

Piasco, N., D. Sidibé, C. Demonceaux, and V. Gouet-Brunet (2019b). Perspective-n-learned-point: Pose estimation from relative depth. In *In British Machine Vision Conference (BMVC)*.

Piasco, N., D. Sidibe, C. Demonceaux, and G.-B. Valerie (2018). A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition 74*, 90–109.

Piasco, N., D. Sidibé, V. Gouet-Brunet, and C. Demonceaux (2021). Improving image description with auxiliary modality for visual localization in challenging conditions. *International Journal of Computer Vision 129*(1), 185–202.

Radenović, F., A. Iscen, G. Tolias, Y. Avrithis, and O. Chum (2018). Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5706–5715.

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.

Ranftl, R., K. Lasinger, D. Hafner, K. Schindler, and V. Koltun (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Razavian, A., J. Sullivan, S. Carlsson, and A. Maki (2016). Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*.

Rocco, I., R. Arandjelovic, and J. Sivic (2017). Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Ronneberger, O., P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer.

Ros, G., L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243.

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., A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla (2017). Are large-scale 3d models really necessary for accurate visual localization? In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6175–6184.

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 (2019). Semantically-aware attentive neural embeddings for long-term 2d visual localization. In *British Machine Vision Conference*.

Shotton, J., B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon (2013). Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2930–2937.

Siam, M., M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang (2018). A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 587–597.

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.

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.

Su, Y.-C. and K. Grauman (2017). Learning spherical convolution for fast features from 360° imagery. In *NIPS*.

Sun, K., Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang (2019). High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*.

Sun, W. and R. Wang (2018). Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm. *IEEE Geoscience and Remote Sensing Letters 15*(3), 474–478.

Sünderhauf, N., S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford (2015). On the performance of convnet features for place recognition. In *IEEE/RSJ International Confer-*

ence on Intelligent Robots and Systems (IROS), pp. 4297–4304.

Tateno, K., N. Navab, and F. Tombari (2018). Distortion-aware convolutional filters for dense prediction in panoramic images. In *European Conference on Computer Vision (ECCV)*, pp. 707–722.

Teichmann, M., M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun (2018). Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1013–1020.

Tian, Y., C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola (2020). What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems (NeurIPS)*.

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

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

Tolias, G., R. Sicre, and H. Jégou (2016). Particular object retrieval with integral max-pooling of cnn activations. In *ICLR 2016-International Conference on Learning Representations*, pp. 1–12.

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

Torii, A., J. Sivic, T. Pajdla, and M. Okutomi (2013). Visual place recognition with repetitive structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 883–890.

Torii, A., J. Sivic, T. Pajdla, and M. Okutomi (2015). Visual place recognition with repet-

itive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 2346–2359.

Valentin, J., M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. H. Torr (2015). Exploiting uncertainty in regression forests for accurate camera relocalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4400–4408.

van den Oord, A., Y. Li, and O. Vinyals (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Wang, T., H. Huang, J. Lin, C. Hu, K. Zeng, and M. Sun (2018). Omnidirectional CNN for visual place recognition and navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Wong, J. M., V. Kee, T. Le, S. Wagner, G.-L. Mariottini, A. Schneider, L. Hamilton, R. Chipalkatty, M. Hebert, D. M. Johnson, et al. (2017). Segicp: Integrated deep semantic segmentation and pose estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5784–5789. IEEE.

Wu, Z., Y. Xiong, S. X. Yu, and D. Lin (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742.

Xu, Y., K. Wang, K. Yang, D. Sun, and J. Fu (2019). Semantic segmentation of panoramic images using a synthetic dataset. In *Artificial Intelligence and Machine Learning in Defense Applications*, Volume 11169, pp. 111690B. International Society for Optics and Photonics.

Yan, H., C. Zhang, and M. Wu (2022). Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention. *arXiv preprint arXiv:2201.01615*.

Yu, X., S. Chaturvedi, C. Feng, Y. Taguchi, T.-Y. Lee, C. Fernandes, and S. Ramalingam (2018). Vlase: Vehicle localization by aggregating semantic edges. In *2018 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pp. 3196–3203. IEEE.

Yu, Z., C. Feng, M.-Y. Liu, and S. Ramalingam (2017). Casenet: Deep category-aware semantic edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5964–5973.

Yuan, Y., X. Chen, and J. Wang (2020). Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pp. 173–190. Springer.

Zamir, A. R. and M. Shah (2010). Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, pp. 255–268. Springer.

Zamir, A. R. and M. Shah (2014). Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*(8), 1546–1558.

Zhang, R., P. Isola, and A. A. Efros (2016). Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer.

Semih Orhan

Vita

Academic Experience

2021–2022 Visiting Research Fellow, *University of Zaragoza*

2018–2021 Research/Teaching Assistant, *Department of Computer Engineering, İzmir Institute of Technology*

Education

2022 PhD in Computer Engineering, *İzmir Institute of Technology*

2018 MSc in Computer Engineering, *İzmir Institute of Technology*

Publications

Journal Publications

2022 Orhan, Semih, and Yalin Bastanlar. "Semantic segmentation of outdoor panoramic images." Signal, Image and Video Processing 16, no. 3 (2022): 643-650.

2018 Orhan, Semih, and Yalin Bastanlar. "Training CNNs with image patches for object localisation." Electronics Letters 54, no. 7 (2018): 424-426.

Book Chapters

2022 Bastanlar, Yalin, and Semih Orhan. "Self-Supervised Contrastive Representation Learning in Computer Vision" In Pattern Recognition - New Insights. London: IntechOpen, 2022. 10.5772/intechopen.104785

Conference/Workshop Publications

2022 Orhan, Semih, Jose J. Guerrero, and Yalin Bastanlar. "Semantic pose verification for outdoor visual localization with self-supervised contrastive learning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3989-3998. 2022.

2021 Orhan, Semih, and Yalin Bastanlar. "Efficient search in a panoramic image database for long-term visual localization." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1727-1734. 2021.

2017 Orhan, Semih, and Yalin Bastanlar. "Effect of patch based training on object localization with convolutional neural networks." In 2017 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4. IEEE, 2017.