NORTHWESTERN UNIVERSITY

TOWARDS MORE ROBUST LOCAL AND GLOBAL VISUAL LOCALIZATION

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the Degree

DOCTOR OF PHILOSOPHY

Field of Electrical and Computer Engineering

By

Pengbo Zhao

EVANSTON, ILLINOIS

June 2023

ABSTRACT

Visual localization is a critical capability for autonomous systems, enabling them to accurately estimate their position and orientation within an environment using visual data. This thesis focuses on achieving a robust and reliable visual localization on both local and global level to enhance localization performance in a wide range of environments.

For global localization, we consider the task of both Visual Place Recognition (VPR) and cross-view geo-localization (CVGL). Visual place recognition enables the system to recognize previously visited locations and refine its position estimates. Additionally, it can provide a preliminary position estimate when the GPS signal is weak and cannot offer real-time GPS positioning. In this thesis, we leverage high-level semantic information generated from scene-graph generators with traditional VPR pipeline to achieve a more robust global visual localization under extremely diverse and challenging environmental conditions.

Cross-view geo-localization is an essential aspect of visual localization, allowing systems to estimate their position and orientation by matching ground-level images to aerial or satellite imagery. CVGL remains extremely challenging due to the drastic appearance differences across aerial–ground views. In existing methods, the interactive benefits global representations of different views are seldom taken into account. In this thesis, we present a novel approach using cross-view knowledge generative techniques in combination with transformers, namely mutual generative transformer learning (MGTL), for CVGL.

For local localization, we focus on Visual odometry (VO). Visual odometry is a key component in visual localization, as it estimates the relative motion of the autonomous system using consecutive visual frames. Existing VO techniques are prone to accumulated error stemming from steering angle deviation, resulting in sub-optimal precision. In this thesis, we propose a novel VO

framework comprising steering angle-weighted learning and triple-frame hybrid constraint learning, alleviating the aforementioned problem and achieving a more robust local localization.

ACKNOWLEDGEMENTS

My heartfelt appreciation goes to my advisor, Prof. Ying Wu, without whom the completion of this thesis would not have been possible. His passion, enthusiasm, and dedication towards computer vision have been passed on to me, and I cannot express enough gratitude for his support, encouragement, and inspiration. His guidance, insights, and expertise have been invaluable throughout my research journey.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Thrasyvoulos N. Pappas and Prof. Qi Zhu for their encouragement, insightful comments, and challenging questions.

I am very thankful I had the chance to work in the Northwestern Computational Vision Lab. I also wish to express my sincere gratitude to all of my colleagues and friends in the CV lab: Xu Zou, Yin Xia, Pei Yu, Wei Tang, Jiahuan Zhou, Xiangyun Zhao, Wei Wei, Peixi Xiong, Yunxuan Li, Lei Fan, Mingfu Liang, Jianxiong Zhou, Xiaoying Xing, and Chen Jiang, for their numerous long discussions, for sharing their valuable comments and ideas, and for helping me through my PhD studies.

Most importantly, I want to thank my family and friends for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Visual localization is a critical capability for autonomous systems, enabling them to accurately estimate their position and orientation within an environment using visual data. With the rise of more and more autonomous navigation systems including autonomous robots and autonomous driving, *etc.* creating an accurate and robust visual localization model has become a vital task in computer vision [1]. Visual Localization refers to the process of identifying the location or pose (position + orientation) of a visual query material within a known spatial representation, depending on the application. For instance, a camera's pose that captured a particular photograph based on a set of geo-localized images or a 3D model is an example of a localization system. In the last decade, Visual localization has gained significant attention due to the availability of large geo-localized image databases, the proliferation of embedded visual acquisition systems (e.g., cameras on smartphones), and the limitations of conventional localization systems in urban environments (e.g., weak GPS signal in cluttered areas) [2]. This localization problem has practical applications in GPS-like localization systems, indoor or outdoor navigation, 3D reconstruction, models and databases update, consumer photography, augmented reality, and robotics, where visual localization is used to solve SLAM loop-closure problems or kidnapped robot scenarios.

In thesis, we focus on achieving a robust and reliable visual localization on both local and global level to enhance localization performance in a wide range of environments.

Visual Place Recognition (VPR) plays a crucial role in various robotics and autonomous system applications, serving as both a standalone positioning capability when using a pre-existing map and an essential component of comprehensive Simultaneous Localization and Mapping (SLAM)

systems. Due to significant variations in appearance, illumination, and viewpoint, accomplishing this task can be challenging, making it a subject of ongoing research in the fields of computer vision and robotics.

VPR is commonly approached as an image retrieval task where the objective is to retrieve the most similar database image (along with associated metadata such as camera pose) when given a query image. Two prevalent ways of representing query and reference images are using global descriptors that describe the entire image, or using local descriptors that describe specific regions of interest. Global descriptor matching typically employs nearest neighbor search between the query and reference images. These descriptors are generally more robust to changes in appearance and illumination, as they are optimized specifically for place recognition. On the other hand, local descriptors are often cross-matched, followed by geometric verification, prioritizing spatial accuracy, primarily on a pixel-level, by using a fixed-size spatial neighborhood to facilitate highly precise 6-DoF pose estimation. Recently, Patch-NetVLAD [3] is proposed to combine the mutual strengths of both local and global approaches while minimizing their weaknesses.

There are two major types of challenges in the task of VPR that separates it from pure image retrieval, namely:

- Query and database images from the same place can have drastic different appearances due to day-night changes, seasonal variations, or a huge time gap. For instance, a query image can be taken in summer/daytime while the dataset consist of images taken from the same place but in winter/night only. Another instance will be the query image is from the current time while the database was built years ago with images from the past, and there is a huge difference in the appearance of the place. The latter scenario is extremely common for urban area VPR because constructions are going on all the time and updating the database means we need to rebuild the whole thing and is extremely costly to do on regular bases

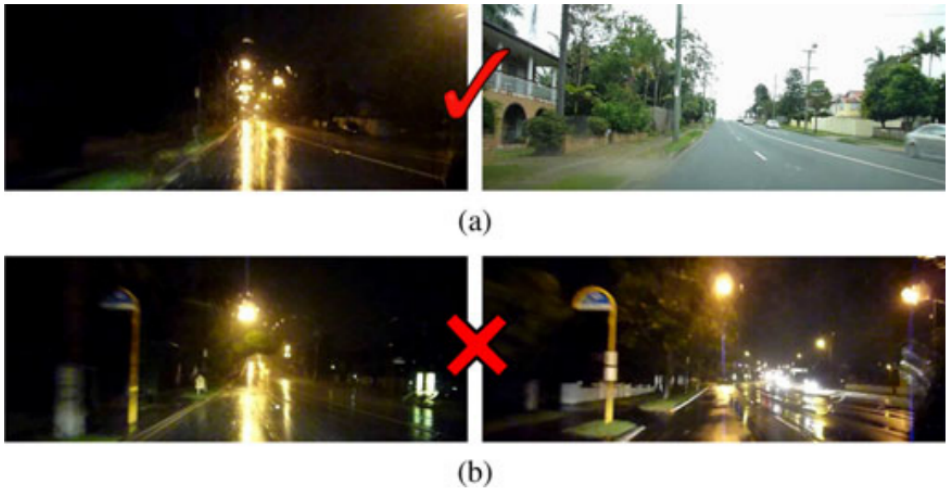


Figure 1.1: An example demonstrating the goal of visual place recognition(VPR), which often needs to retrieve images of the same place with drastic appearance changes. Figure is from [4]

as most urban VPR databases have large scales; thus, navigation systems that deployed in urban areas often need to work with current time input query images and compare them to database images from the past, which is very challenging. In addition, there might be dynamic objects that are essentially not a part of the place such as cars and humans that enlarge the appearance gap between the query and ground truth image pairs.

- Images from different places in an environment may have similar appearances, this problem is also called perceptual aliasing [4]. In addition, places may not always be revisited from the same viewpoint and position as before. Perceptual aliasing may confuse the navigation system and undermines its robustness. This problem happens especially often in urban settings since many buildings or streets share a lot of things in common visually.

Existing methods tackled these challenges and made significant progresses [3, 5, 6, 7, 8]. However, the problem is far from solved. The performance of these methods are still limited when there is significant variance in appearances between query the database images. Intuitively, many

researchers think of utilizing semantic informant ion to increase the discriminativity of images towards more robust VPR systems [9, 10]. Nevertheless, despite the fact that VPR methods that leverages semantic information [9, 10] exist, the semantic information they use are often at the level of semantic segmentation from off-the-shelf semantic segmentation methods such as UNet [11] or SegNet [12]. The semantic segmentation from these architectures only considers pixel level semantic labels without label weighting,*i.e.*, every class is treated equally. Though such information is enough for some downstream tasks, it is insufficient for VPR. For instance, treating all semantic labels equally means that similarity search will be highly affected by dynamic objects: two different places from a environment may have a very high similarity scores because there are similar cars or human beings in both places, limiting the performance of the model. Thus, we need a semantic information generator that provides more than just pixel level semantic labels. From literature survey, we noticed that an approach that leverages high-level semantic information such as the output from a scene-graph generator is unprecedented, and we believe that the characteristics of scene graphs satisfies the needs of extra discriminative information needed for VPR. A scene graph is a structural representation that explicitly models objects, attributes of objects, and relations between paired objects; additional attributes and relations provide semantic labels with much more discriminativity, which is we need. For example, for existing approaches, two different cars from two images are treated as the same semantic label even when the surroundings are different and the cars themselves are different. Nonetheless, if we take the extra attributes and relations from scene graphs into account, we can separate one car from another and produce a more discriminative similarity score. Nevertheless, even for scene graphs, semantic information alone are pretty noisy so such information must be used as a complementary for visual-based matching.

In this thesis, we propose a novel VPR frame work called NetVLAD with Scene Graph Generation (SGG-NetVLAD) that combines the visual representation of NetVLAD variant methods and

semantic representation from a scene graph generator. The two modalities are then fused together to produce a more robust VPR estimation to leverage the strength of both methods.

Next, we tackle the task of **cross-view geo-localization (CVGL)**. Geo-location identification of automobiles has been a topic of growing interest in recent years due to its potential applications in navigation and route planning for intelligent vehicles [13, 14, 15, 16, 17, 18, 19]. Conventionally, obtaining the geographic location of a vehicle through Global Navigation Satellite Systems (GNSS) has been a convenient and cost-effective method. However, GNSS signals are prone to being unreliable or unavailable due to the presence of dense high-rise obstacles, network failures, etc. For example, scenarios such as dense primordial forests and crowded buildings are shown in Figure 1.2. Fortunately, current satellite images can cover most outdoor scenarios where automobiles are involved and are easily collected offline in advance through open services like Google Maps. To overcome this limitation, the use of registered ground–satellite image retrieval for geographic location estimation has gained increasing attention [5, 20, 21, 22, 23, 24, 25]. This method involves the comparison of visual data obtained from the vehicle with geo-tagged references stored in a database, resulting in the estimation of the geographic location that is aligned with the closest reference. This pipeline is schematically illustrated in Figure 1.2.

Geolocation involves comparing perspectives from previously visited sites with similar scene content to detect loop closures. In the absence of GNSS signals, contextual scene analysis is used to determine the vehicle's position. This requires a nuanced understanding of the scene's geometric and structural information, such as edge and corner features, shapes, and relative positions. Overfeat [26] was an early deep learning-based study in the field, inspiring improvements such as ground-to-ground matching for localization by gathering views at diverse locations and times. However, these methods are labor-intensive and cannot locate places outside the reference dataset. To improve the location model's generalization performance, researchers aim to establish inter-

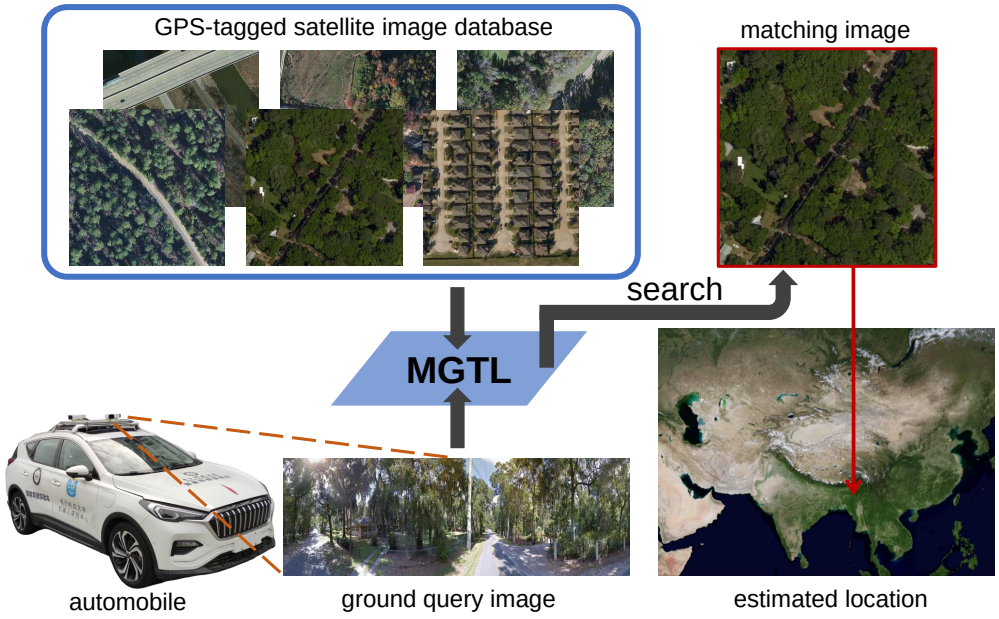


Figure 1.2: Schematic diagram of image matching-based cross-view geo-localization.

connectivity between satellite and ground views through cross-view geo-localization (CVGL) [5, 27, 28, 29, 30]. While Siamese-like networks have been successful in encoding cross-view views independently, several challenges remain. First, semantic consistency between views is not fully leveraged. Second, co-visual relationships between views are not explicitly accounted for. Third, deep contextual semantic mining is not yet sufficient.

To address the above deficiencies, we present a novel mutual generative transformer learning (MGTL) for the CVGL task.We first revisit the attention learning strategy and propose a novel cascaded attention masking algorithm to create network reasoning for the co-visual patterns between ground and satellite views. Then, two symmetrical generative sub-modules, i.e., Ground-to-Satellite (G2S) and Satellite-to-Ground (S2G), are thoughtfully designed to generate the simulated cross-view knowledge and to capitalize on the mutual benefits across views. Specifically, S2G takes the aerial semantics and skillfully simulates the ground-aware knowledge, and vice versa. Subsequently, the view-specific simulated knowledge is applied to strengthen the current view fea-

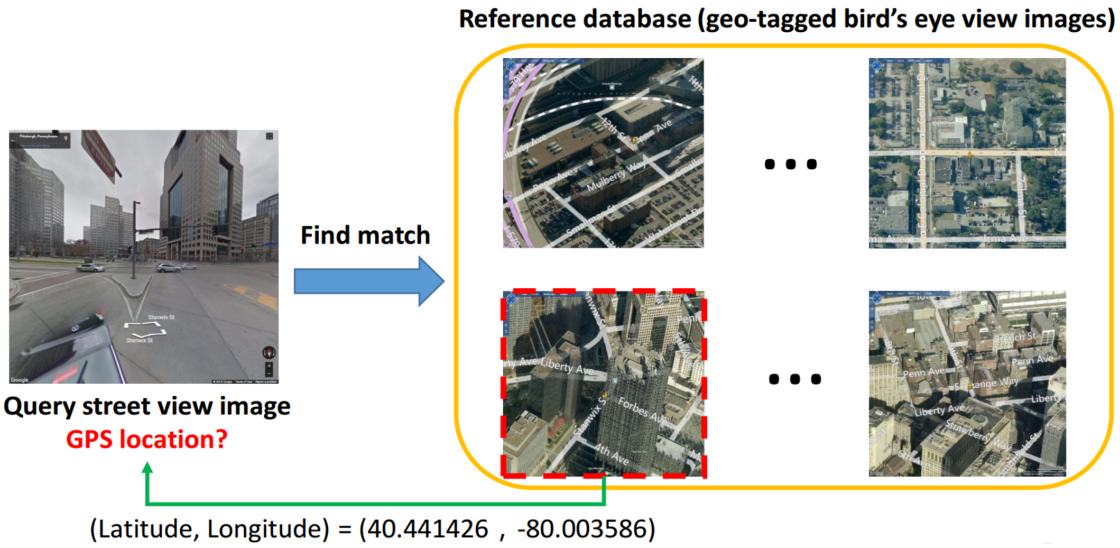


Figure 1.3: An example demonstrating the task of cross-view geo-localization (CVGL), figure from [31]

tures via attention learning, and all the sub-components work in concert within a transformer-based framework to accomplish the CVGL task. The experimental results for several challenging public benchmarks unequivocally establish the superiority of our proposal.

The monocular **visual odometry (VO)** framework estimates camera motion trajectory using only a monocular camera without GPS or other localization devices, making it essential in robotics localization and autonomous driving. Since the task of VO focuses on estimating the location of the system given consecutive frames from the camera, it can be treated as a local visual localization task. VO has received plenty of attention in recent years and researchers have made significant progresses [32, 33, 34, 35]. However, complex perceptual scenarios and uncorrected strategies result in VO remaining challenging.

Despite the promising results achieved by current deep learning methods [36, 37, 38], there still exist several shortcomings that need to be addressed, namely:**i)** these methods fail to incorporate error correction during the learning process, resulting in the propagation of estimation errors

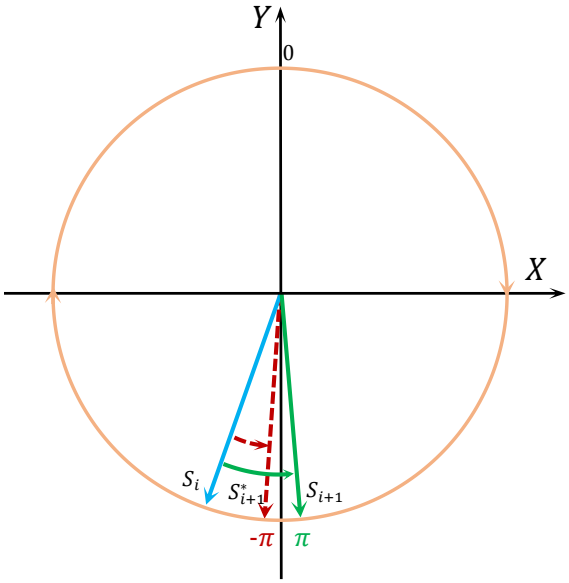
Table 1.1: Straight and turning examples statistics in KITTI training benchmark.

Seq.	00	02	08	09
straight	3382	3322	3039	959
turning	1158	1338	1031	631

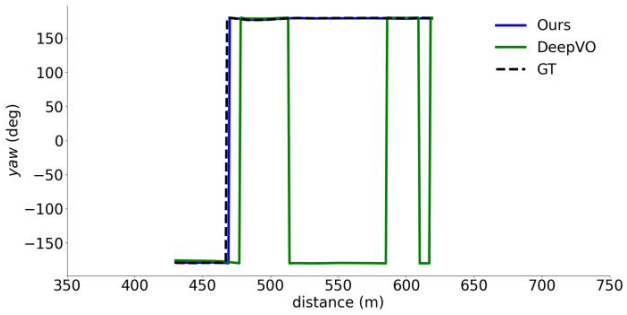
and persistent trajectory deviations. **ii)** the inherent limitations of steering angle modeling are often neglected, leading to sudden and erratic changes in trajectory during turning maneuvers. As illustrated in Figure 1.4a and Figure 1.4b, when the estimated steering angle at time i to $i+1$ is at the boundary of the model, the absolute trajectory is susceptible to substantial discrepancies. **iii)** the issue of imbalanced data distribution in training datasets results in poor performance when generalized to real-world scenarios. As shown in Table 1.1, straight driving often dominates the datasets, leading to insufficient training on turning maneuvers and other challenging scenarios. **iv)** the existing techniques exhibit insufficient exploitation of both temporal and contextual information, which restricts the ability to fully comprehend and accurately reconstruct the environment, resulting in sub-optimal results.

To cope with these deficiencies, in this thesis, we propose a novel VO framework, called Steering Angle Correction network (SACNet), which incorporates the steering angle as a weighted constraint during the learning process, leverages cross-frame information to mitigate steering angle discontinuities and integrates LSTM and attention mechanisms to obtain dependable contextural features. Through extensive experiments on the challenging KITTI VO benchmark [39], we compare our proposed SACNet against strong baselines and state-of-the-art methods, demonstrating its effectiveness in retrieving steering angle guidance information for visual odometry.

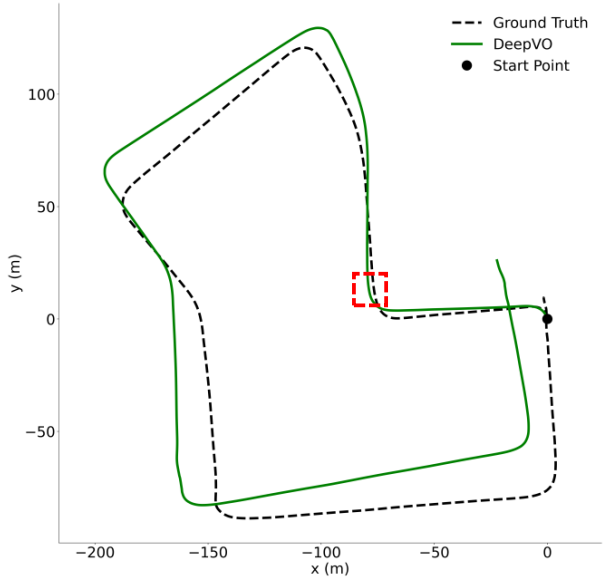
The rest of the thesis is organized as following: in chapter 2, we review the related works of the three tasks we tackle in this thesis. For chapter 3, we discuss in details the method and experimental results for our novel VPR framework SGG-NetVLAD, and the same goes for MGTL for CVGL in chapter 4 and SACNet in chapter 5. Finally, in chapter 6, we summarize our findings



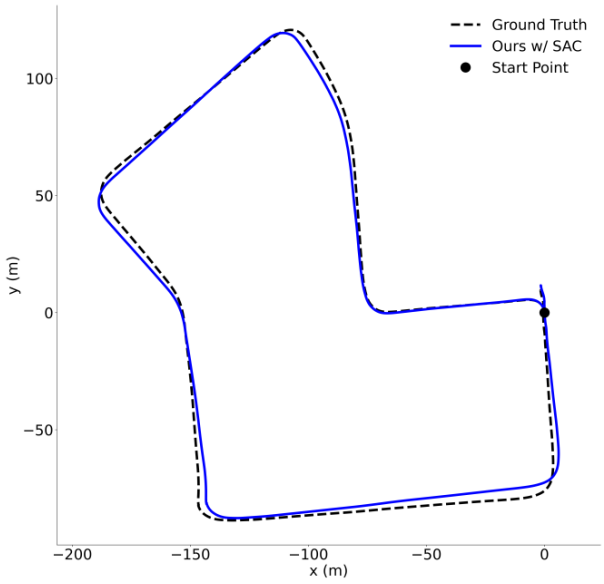
(a) Cause analysis of steering angle hopping.



(b) Example of steering angle hopping.



(c) The camera trajectory reconstructed by DeepVO [36].



(d) The camera trajectory reconstructed by SACNet.

Figure 1.4: The schematic illustration of the research motivation of our approach ((a-b)) and the experimental comparison results ((c-d)). The red dashed rectangle in (c) indicates that the turning angle deviates from the ground-truth trajectory.

and contributions and conclude our works.

CHAPTER 2
RELATED WORKS

2.1 Robust Visual Place Recognition with Scene-Graphs

2.1.1 Visual Place Recognition

Visual place recognition (VPR) is a well-defined but extremely challenging problem to solve in the general sense; given an image of a place, can a human, animal, or robot decide whether or not this image is of a place it has already seen? [4]. VPR is a crucial aspect of vision-based navigation and localization, particularly in the field of autonomous driving. Generally speaking, VPR can take a variety of inputs including 2D images[5, 29, 28], 3D point clouds [40, 41], *etc.* In this thesis, we consider the task of pure 2D image-based VPR. The problem tackled by VPR is to determine the current camera's location in the existing image database, and the task becomes challenging due to factors such as seasonal variations and dynamic visual object changes. A primary approach to overcome this challenge involves extracting high-level feature descriptors from input images and comparing them based on distance. To improve VPR performance, Arandjelovic et al. [5] modified the traditional non-differentiable operation in vector of locally-aggregated descriptors (VLAD) and incorporated it into CNN-based networks to develop an end-to-end trainable VLAD descriptor named NetVLAD. Following the success of NetVLAD, several variants [29, 28] have been proposed. To exploit the multi-scale information, spatial pyramid-enhanced NetVLAD (SPE-NetVLAD) [29] integrated multi-scale features in the training phase by cascading encoding features with varying scales in the final convolutional layer of NetVLAD to improve the performance of VPR. Multi-resolution NetVLAD (MultiRes-NetVLAD) [28] utilized low-resolution

image pyramid coding and presented a multi-resolution residual aggregation scheme to enhance the NetVLAD learning feature representation ability. In addition, to address the issue of seasonal and time-of-day variations, Latif et al. [30] approached the VPR problem as a region translation task. A pair of coupled generative adversarial networks (GANs) was utilized to generate the appearance of one domain from another without requiring image-to-image correspondences across the domains. Most recently, Patch-NetVLAD [3] has used an integral feature space to derive patch descriptors from the global image feature and has achieved state-of-the-art performance in several benchmarks. However, the performance of these existing methods are limited under drastic appearance changes such as seasonal variations or huge change in lighting conditions; a query image from nighttime versus a database built with daytime images only, for instance.

2.1.2 Content-Based Image Retrieval

Content-Based Image Retrieval (CBIR) is closely related to pure image-based VPR since both tasks seek to retrieve the most similar images of a given query from an image database that is built in an off-line stage. For over two decades, content-based image search or retrieval has remained a significant challenge in the multimedia field [42]. A general flowchart of a CBIR framework is depicted in 2.1. A typical visual search framework comprises an off-line and an on-line stage. In the off-line stage, a database is constructed by crawling images, representing each image as vectors, and indexing them. The on-line stage involves several modules, such as user intention analysis, query formation, image representation, image scoring, search re-ranking, and retrieval browsing. The image representation module is employed in both stages.

In content based image retrieval, the key problem is how to efficiently measure the similarity between images. Due to the possible variations or transformations of visual objects or scenes, comparing images at the pixel level is impractical. Typically, visual features are extracted from

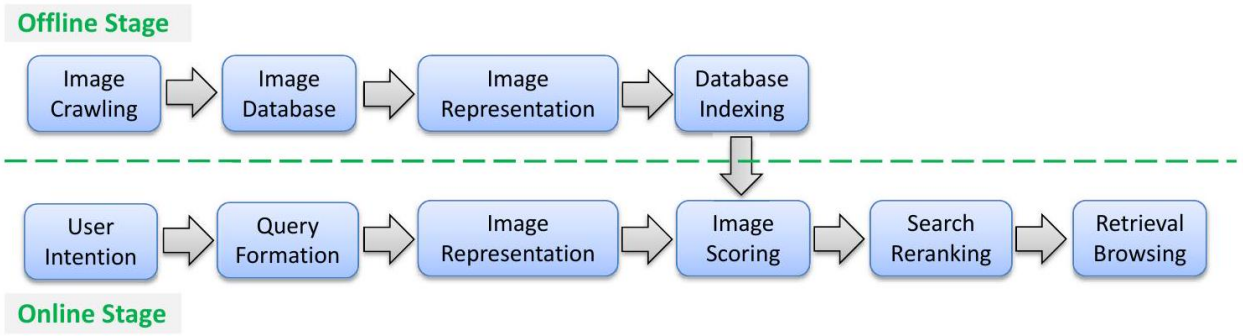


Figure 2.1: The general framework of content-based image retrieval. The modules above and below the green dashed line are in the off-line stage and on-line stage, respectively [42].

images and transformed into a fixed-size vector for image representation. To handle the trade-off between the large-scale image database and the need for efficient query response, it is essential to ”pack” the visual features to facilitate indexing and image comparison. To this end, quantization, along with visual codebook training, is often used for feature aggregation/pooling as a standard encoding procedure. Moreover, spatial context, as an essential characteristic of visual data, plays a vital role in improving the distinctiveness of visual representation.

Traditionally, visual features are created using heuristic methods and can be classified into local and global features. However, in recent years, there has been a growing trend towards the use of learning-based features. Here, we discuss both approaches in the following.

Hand Crafted Features: In early CBIR algorithms and systems, global features are commonly used to describe image content by color, shape, texture, and structure into a single holistic representation [42]. As one of the representative global feature, GIST feature [43] is known for its low computational complexity and has been widely applied to evaluate approximate nearest neighbor search algorithms [44]. With compact representation and efficient implementation, global visual feature are very suitable for coarse similarity search in large-scale image database to produce a list of similar images to the query, but may not have a very high precision, *i.e.*, the target image is in the list but not at the top of the list. Typically, global features are used as a first stage of coarse

search and then the resulting list is then re-ranked by local features.

Local features represents a pattern or distinct structure found in an image, such as a point, edge, or small image patch. Typically, these features are related to an image patch that exhibits variations in texture, color, or intensity compared to its neighboring regions. After introduction of the SIFT [45], local features have become a widely used image representation method in numerous studies related to content-based image retrieval. Generally speaking, local feature extraction comprises two essential steps: interest point detection and local region description. During interest point detection, certain key points or regions with a distinct scale are identified with high repeatability, indicating that these interest points can be recognized despite undergoing various transformations or changes.

Following the identification of interest points, one or more descriptors are extracted to depict the visual appearance of the local region centered at the interest point. Typically, these descriptors are designed to be invariant to rotation changes, robust to affine distortion, addition of noise, illumination changes, and other factors. Moreover, they should be distinctive enough to match a single feature against a vast collection of features from multiple images with high accuracy, which is crucial for large-scale visual applications. SIFT feature [45] is the most popular choice for this purpose, given its desirable properties. Alternatively, SURF [46] is also demonstrated to perform comparably but with better efficiency.

Besides using floating point features such as SIFT, binary features have gained popularity and can be directly extracted from the local region of interest. Recently, BRIEF [47], a binary feature, and its variants, including ORB [48], FREAK [49], and BRISK [50], have been proposed and attracted significant interest in visual matching applications. These binary features are computed using simple intensity difference tests, which are highly computationally efficient. Additionally, due to the advantages of using Hamming distance computation, binary features based on the FAST

detector [51] have the potential to be useful in large-scale image search applications.

Learning Based Features: In addition to the aforementioned handcrafted visual features, it is feasible to acquire features for image retrieval through a data-driven approach.

In recent years, the success of learning-based features in multiple areas has been demonstrated by the extensive research on deep neural networks (DNN) [52, 53]. DNN's deep architecture enables the learning of high-level abstractions, similar to the human cognition process [54]. Consequently, it is possible to utilize DNN to extract semantic-aware features by analyzing the activations of various layers in the network. For instance, in [55], features are obtained in local patches using a deep restricted Boltzmann machine (DBN), which is refined using back-propagation. Deep convolutional neural network (CNN) [56], as a typical structure of the DNN family, has demonstrated state-of-the-art performance in various tasks related to image recognition and retrieval [57]. In [58], comprehensive studies were conducted on the potential of learned visual features with deep CNN for various applications, including content-based image retrieval. In [59], the activations of the sixth layer of the Alex-Net [56] were extracted as a DNN feature for each image, which was fused at the image similarity score level with traditional visual features such as SIFT-based Bag-of-Words feature, HSV histogram, and GIST.

2.1.3 Scene Graph Generator

A scene graph is a structural representation that explicitly models objects (*e.g.*, "man," "fire hydrant," "shorts"), attributes of objects (*e.g.*, "fire hydrant is yellow"), and relations between paired objects (*e.g.*, "man jumping over fire hydrant"), as illustrated in Fig 2.2 [60]. The fundamental components of a scene graph are objects, attributes, and relations. Objects/subjects, the core building blocks of an image, can be located using bounding boxes. Each object can have zero or more attributes, such as color (*e.g.*, yellow), state (*e.g.*, standing), material (*e.g.*, wooden), etc. Relations

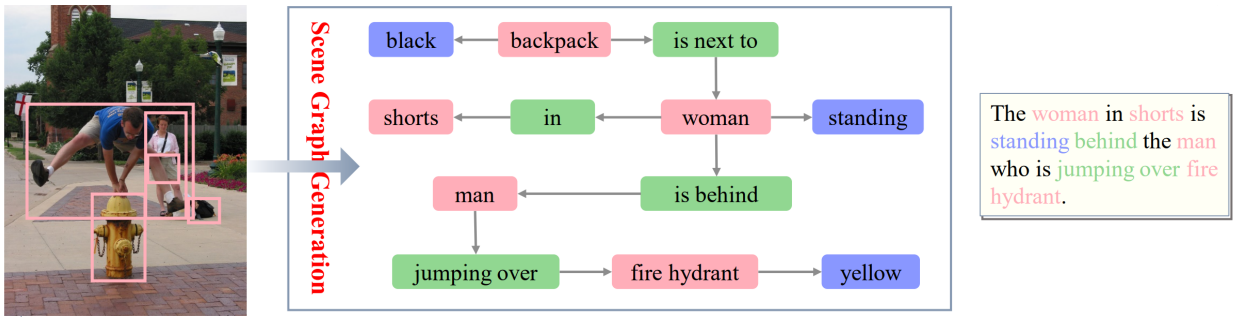


Figure 2.2: A visual illustration of a scene graph structure. Scene graph generation models take an image as an input and generate a visually-grounded scene graph [60]

can take various forms, including actions (*e.g.*, "jump over"), spatial (*e.g.*, "is behind"), descriptive verbs (*e.g.*, wear), prepositions (*e.g.*, with), comparatives (*e.g.*, taller than), prepositional phrases (*e.g.*, drive on), *etc.* [61, 62]. In summary, a scene graph is a collection of visual relationship triplets represented as $\langle subject, relation, object \rangle$ or $\langle object, is, attribute \rangle$. The latter is also treated as a relationship triplet.

The primary objective of scene graph generation is to analyze an image or a sequence of images to produce a structured representation, thereby bridging the gap between visual and semantic perception and ultimately attaining a comprehensive understanding of visual scenes. Thanks to recent advancements in scene graph generation [63, 64], numerous high-level visual semantic tasks have emerged, including VQA [65], image captioning [62], and expression comprehension [66]. However, despite the fact that VPR methods that leverages semantic information [9, 10] exist, an approach that leverages high-level semantic information such as the output from a scene-graph generator is unprecedented.

2.2 Co-Visual Pattern-Augmented Generative Transformer Learning

2.2.1 Cross-View Geo-Localization

Current cross-view geo-localization (CVGL) pipelines utilize a Siamese-like neural network to extract feature representations from each view, followed by the definition of a metric that places the embedding features of cross-view images in close proximity based on their GPS coordinates. The primary obstacle in CVGL tasks is the significant appearance gap between ground and aerial views caused by changes in viewpoint [67]. Satellite-view images are typically composed of satellite images captured by specialized panchromatic and multispectral cameras on board satellites, whereas ground-view images consist of panoramic images taken using handheld or vehicular optical cameras. These two images have different imaging principles and shooting angles, leading to stark differences in image appearance, such as the representation of visual objects and their spatial layout. This problem is further exacerbated by the large time intervals between acquisition of images. Prior work has mainly addressed this issue by focusing on extracting viewpoint-invariant features [68, 69, 70] or applying viewpoint transformation [71, 72, 73]. The former involves designing effective network architectures that can extract invariant features across views. Workman et al. [20] proposed a convolutional neural network (CNN) to learn a joint semantic feature representation for aerial and ground-level imagery, while Lin et al. [74] introduced a Siamese-like network followed by Euclidean distance calculation to measure cross-view feature representation similarity. More recently, Hu et al. [22] utilized NetVLAD to encode global descriptors and a Siamese-like CNN-based network to extract local feature descriptors for more robust representation learning. Sun et al. [75] further presented a pure convolutional network equipped with capsule layers to model the spatial feature hierarchies. In contrast, to address the imagery geometric gap caused by viewpoint differences, Shi et al. [76, 77] used polar transform and attention mechanisms

to pre-process satellite imagery, which has been shown to be highly effective. Recently, Yang et al. [25] and Zhu et al. [24] proposed transformer-based methods, leveraging self-attention mechanisms to model global dependencies. Zhu et al. [24] introduced a novel attention-based masking mechanism to remove redundant areas in satellite images, reducing interference in matching performance. The latter approach involves exploring ways to synthesize realistic cross-domain imagery using viewpoint transformation. Ren et al. [78] proposed a cascaded cross MLP-mixer GAN (CrossMLP) module to extract latent mapping cues between cross-view imagery, while Toker et al. [79] developed a GAN-based multi-task architecture to synthesize realistic street views from satellite images. However, existing methods lack mutual learning across views and fail to consider the inter-dependencies between latent features in different network branches. In this paper, we propose a novel approach that integrates cross-view knowledge generative tactics into the transformer architecture, referred to as mutual generative transformer learning. This approach leverages mutual learning across different views to improve feature representation ability and retrieval performance.

2.2.2 Vision Transformer

The transformer [80] has gained widespread use in the field of natural language processing (NLP) due to its excellent global modeling ability and self-attention mechanism, as demonstrated by its superior properties [80]. The self-attention mechanism is based on the calculation of dot product similarity by query and key, which are then multiplied with value, where query, key, and value represent different embedding spaces computed by the input feature sequence. Dosovitskiy et al. [81] introduced the Vision Transformer (ViT), which is a modified version of the standard transformer that takes the embedding sequences of image patches with $k \times k$ resolution as input [81]. Unlike the standard transformer in NLP, ViT discards the locality assumption and requires less vision-specific sensing bias, dominating in classification [82, 83, 84], semantic segmentation [85,

86, 87], object detection results [88, 89, 90], super-resolution restoration [91, 92], depth estimation [93, 94], etc. Chen et al. [82] proposed a multi-headed and multi-tailed shared backbone structure to cope with different vision tasks. Lanchantin et al. [84] proposed the classification transformer (C-Trans) network to complete a generic multi-label image classification task. Segmentation transformer (Segmenter) [85] defined the semantic segmentation task as a sequence-to-sequence problem and employed the transformer architecture. Zheng et al. [87] incorporated different decoders into ViT to tackle segmentation tasks. Detection transformer (DETR) [88] employed a transformer-based approach and treated object detection as a set prediction problem. Misra et al. [89] added non-parametric queries and Fourier positional embeddings to the traditional transformer to suit the 3D object detection task. Zamir et al. [92] modified several key designs in a multi-head attention and feed-forward network so that they can capture long-range pixel interactions while still being suitable for high-resolution images. Liang et al. [91] introduced the ViT into light field image super-resolution restore tasks. Li et al. [93] utilized dense pixel matching with location information and attention mechanisms in the transformer to take the place of customer construction widely used for depth estimation. Ding et al. [94] designed a novel end-to-end deep neural network based on a feature matching transformer (FMT). In addition to RGB image fields, current works have also scrutinized the application of transformers in hyperspectral images (HSI) [95, 96, 97, 98] and achieved superior results. He et al. [95] introduced a new spatial–spectral transformer (SST) classification framework comprising an improved dense transformer layer for HSI classification. Sun et al. [97] improved a spectral—spatial feature tokenization transformer (SSFTT) method to capture spectral-–spatial features and high-level semantic features. Multispectral fusion transformer network (MFTNet) [98] was designed as a novel feature fusion tactic to generate robust cross-spectral fusion features. Researchers have proposed a series of variants to improve the general ability of ViT. These variants contain substantial skillful tactics such as enhanced locality,

improved self-attention algorithms, and structural redesign [99, 100, 101, 102, 103]. To introduce the locality principle in the transformer, Chu et al. [99] proposed the conditional positional vision transformer (CPVT), which uses a conditional positional encoding scheme consisting of a 2D CNN to realize translation invariance. Positional embeddings are generated based on the local relationship of the restricted tokens, which encode the relative location information of tokens implicitly [99]. Locality vision transformer (LocalViT) [100] is inspired by the comparison between feed-forward networks (FFN) and reverse residual blocks, and depth-wise convolutional is applied to FFN to add locality to the vision transformer [100]. Cross-scale attention transformer (CrossFormer) [104] presented multi-scale feature representation learning tactics in combination with a vision transformer. Cross-attention multi-scale vision transformer (CrossViT) [101] proposed a two-branch transformer to process tokens generated by patches of different sizes and then fused these tokens multiple times to achieve mutual complementation of semantic information by applying cross-attention interaction [101]. Liu et al. [102] proposed a hierarchical vision transformer using shift windows (swin-transformer), using a shift-window-based module to replace the traditional multi-head self-attention. The framework allows for cross-window connections and promotes the flexibility of modeling at different scales. Considering the transformer's powerful global modeling ability and successful application in visual works, we designed a transformer-based network to further explore its potential in the cross-view geo-localization task.

2.3 Visual Odometry

Odometry is the process of estimating an agent's change in position and orientation over time. Visual odometry (VO) is the designation given when relying on the input of a single or multiple cameras attached to the agent. VO methodologies consist of reckoning the pose of the sensor (or system where it is mounted, e.g., autonomous vehicle) by extracting ego-motion parameters from

correspondences between sequential image frames.

Given the agent's pose in timestep $k-1$, X_{k-1}, in a fixed frame, the goal of visual odometry is to compute the transformation T_k^{k-1} (Equation 1), such that $X_k = T_k^{k-1} X_{k-1}$. This operation allows to retrieve an estimate of the pose in timestep k, X_k, by relating the different camera perspectives of successive frames [105].

$$T_k^{k-1} = \begin{pmatrix} R_k^{k-1} & t_k^{k-1} \\ 0 & 1 \end{pmatrix} \tag{2.1}$$

$R_k^{k-1} \in \mathrm{SO}(3)$ and $t_k^{k-1} \in \mathbb{R}^3$ are the rotation and translation, respectively, between poses in time-steps $k-1$ and k. The vehicle's trajectory up to a timestep k, can thus be reconstructed by integration from the initial pose X_0, following Equation 2 [105].

$$T_k^0 = T_1^0 T_2^1 \dots T_k^{k-1} \tag{2.2}$$

The set of existing VO methods can be divided into two distinct groups: knowledge-based and learning-based approaches. The first exploits camera geometrical relations to assess the motion, whereas the other is based on Machine Learning techniques, which rely on considerable amounts of data to acquire pose prediction capabilities. As illustrated in Figure 2.3, knowledge-based methods can be categorized into three sub-groups: appearance-based, feature-based, and hybrid, according to how visual components are used to generate odometry estimates.

Classical knowledge-based monocular VO systems have developed a fixed and intricate pipeline, wherein each module necessitates meticulous design to achieve reliable pose estimation. With deep learning techniques achieve dominating performance in various vision tasks [81, 106, 103, 107], classification, segmentation and detection, *etc*. Researchers have commenced scrutinizing the possibilities of neural networks in VO. The primary objective of this study is to develop an end-to-end learning-based VO system.

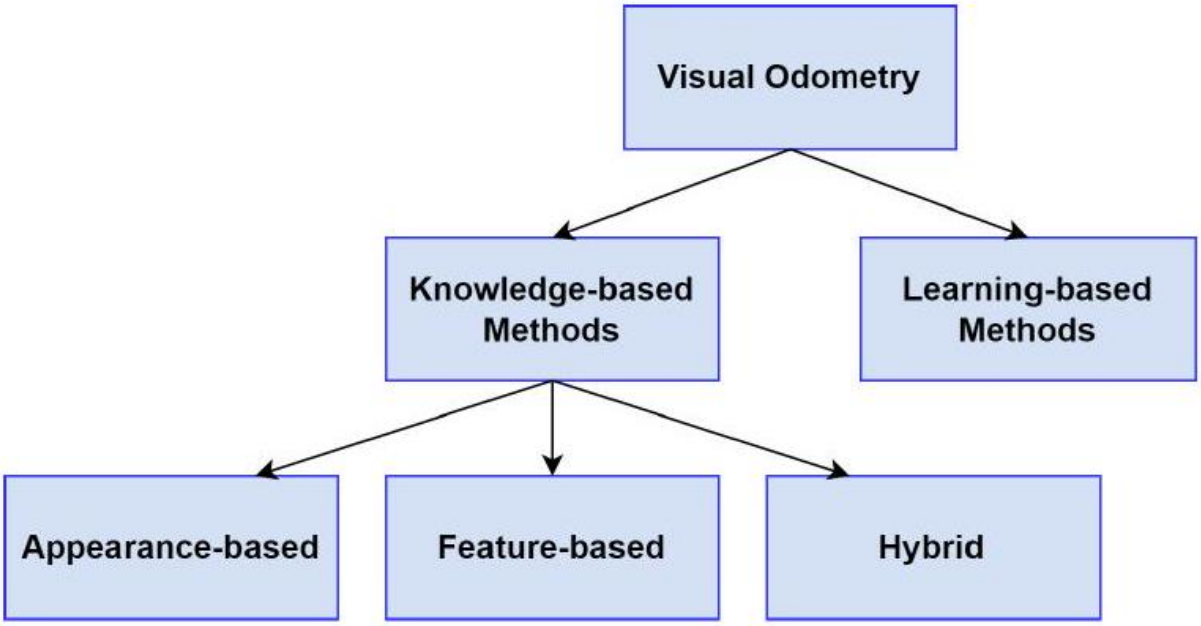


Figure 2.3: Visual Odometry categorization: the distinction is made between approaches based on Machine Learning techniques and knowledge-based ones

2.3.1 Knowledge-based VO Solutions

The primary obstacles of knowledge-based solutions for monocular visual odometry reside in robust feature descriptor extraction, outlier rejection, absolute scale estimation, and estimation result optimization. The term visual odometry was first introduced by Nister *et al.* [108] who conducted real-time monocular camera trajectory estimation of large scenes. MonoSLAM [109] was the first monocular camera-based Simultaneous Localization and Mapping (SLAM) technique that employing the Kalman filter to compute camera pose but incurred high computational complexity. To overcome this issue, Klein *et al.* [110] proposed the PTAM algorithm that utilizing nonlinear optimization as the SLAM backend and introduced the key-frame processing algorithm, bundle adjustment [111]. Based on PTAM, Mur-Artal *et al.* [48] proposed ORB-SLAM that leveraging Oriented FAST and Rotated BRIEF (ORB) feature descriptors and creatively introduced the concept of

loop closure to significantly promote the pose estimation accuracy. ORB-SLAM2 [112] further improved the previous methods by optimizing the detection process and introduced a re-localization strategy, thereby enhancing pose estimation, localization accuracy and efficiency. Conventional monocular knowledge-based VO solutions often struggle to extract robust feature descriptors in scenarios with limited or textureless features, resulting in lacking generalization capability to varying extent.

2.3.2 Learning-based VO Solutions

The key challenge of learning-based solutions is to obtain high-order understandings of the scenarios to enable pose and scale estimation from a massive dataset without explicit camera modeling. Since the emergence of DeepVO [36], numerous variants have been proposed. Building on DeepVO [36], ESP-VO [113] computed the uncertainty of pose estimations to reduce random errors while utilizing the Euclidean group SE(3) and Lie algebra $se(3)$ as constraint functions. PoseCon-vGRU [114] replaces LSTM with a gated recurrent unit [115], achieving similar performance as DeepVO [36] with lower computational complexity. GFS-VO [116] reinforced different motion patterns with context-based attention mechanisms and introduced ConvLSTM [117] to capture robust temporal dependencies. TartanVO [38] incorporated a scale-consistency loss to constrain translation pose estimation. DeepAVO [37] adopted a two-stage scheme by utilizing FlowNet [118] to extract the optical flow representations first, followed by incorporating CBAM [119] to learn motion-sensitive regions while suppressing background regions. DAVO [120] proposed a dynamic attention mechanism that processes weighted semantic segmentation results generated by the semantic segmentation network to obtain the attention map and highlight regions with higher motion involvement. Beyond Tracking [121] argued that VO needs more than just relative pose tracking and introduced a global information memory module

to achieve absolute pose estimation constraints, significantly improving pose estimation accuracy, but with higher computational complexity.

CHAPTER 3
ROBUST VISUAL PLACE RECOGNITION WITH SCENE-GRAPHS

3.1 Background

Visual Place Recognition (VPR) plays a crucial role in various robotics and autonomous system applications, serving as both a standalone positioning capability when using a pre-existing map and an essential component of comprehensive Simultaneous Localization and Mapping (SLAM) systems. Due to significant variations in appearance, illumination, and viewpoint, accomplishing this task can be challenging, making it a subject of ongoing research in the fields of computer vision and robotics.

First, we need to make the definition of a place clear. What is a place? According to [4], the concept of a place varies depending on the navigation context and can be defined either as a precise point or a larger area. As [4] describes, a place represents a zero-dimensional point that describes part of the environment,*i.e.*, a precise location, while others consider it to be an abstraction of a two-dimensional or three-dimensional region. For instance, a room in a building may qualify as a single place in some cases, while in others, it may contain multiple places. Notably, a place does not have an orientation, unlike a robot pose. Therefore, ensuring recognition irrespective of the robot's orientation within the place is a significant challenge in place recognition.

VPR is commonly approached as an image retrieval task where the objective is to retrieve the most similar database image (along with associated metadata such as camera pose) when given a query image. Two prevalent ways of representing query and reference images are using global descriptors that describe the entire image, or using local descriptors that describe specific regions of interest. Global descriptor matching typically employs nearest neighbor search between the

query and reference images. These descriptors are generally more robust to changes in appearance and illumination, as they are optimized specifically for place recognition. On the other hand, local descriptors are often cross-matched, followed by geometric verification, prioritizing spatial accuracy, primarily on a pixel-level, by using a fixed-size spatial neighborhood to facilitate highly precise 6-DoF pose estimation. Despite their complementary strengths, there has been limited research aimed at integrating global and local approaches. Recently, Patch-NetVLAD [3] is proposed to combine the mutual strengths of both local and global approaches while minimizing their weaknesses.

Despite significant progress in existing methods for visual place recognition (VPR)[3, 5, 6, 7, 8], the problem is still challenging, especially when query images have significant appearance variations from database images. To address this limitation, researchers have explored leveraging semantic information to increase the discriminativity of images in VPR systems[9, 10]. However, the semantic information used in these methods is often limited to pixel-level semantic labels from off-the-shelf semantic segmentation models, such as UNet [11] or SegNet [12], where all classes are treated equally without label weighting. While such information is sufficient for some downstream tasks, it is insufficient for VPR, as it can be highly affected by dynamic objects, limiting the performance of the model. Thus, a semantic information generator that provides more than just pixel-level semantic labels is needed. From literature survey, we noticed that using a high-level semantic representation, such as the output from a scene-graph generator, is unprecedented and can provide the extra discriminative information needed for VPR. Scene graphs explicitly model objects, attributes of objects, and relations between paired objects, providing semantic labels with much more discriminativity than pixel-level semantic labels. For example, existing approaches treat two different cars from two images as the same semantic label, even when the surroundings and the cars themselves are different. However, using extra attributes and relations from

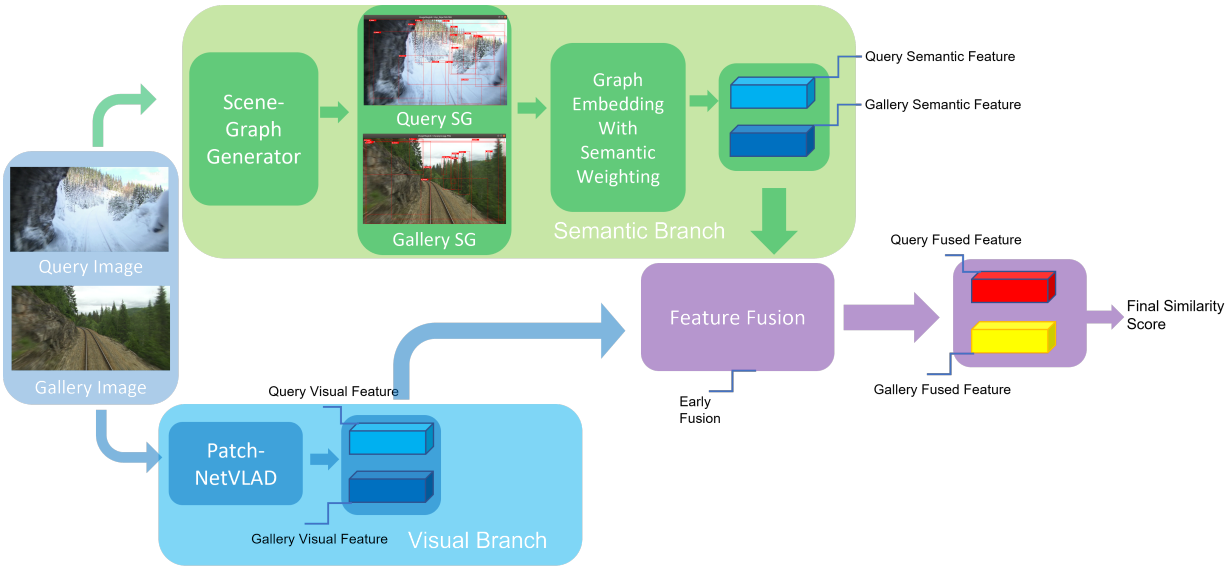


Figure 3.1: The overview of the framework of SGG-NetVLAD. There are two branches, namely, the visual branch and semantic branch. The input image pairs are first feed into both branches, a visual feature and a semantic feature is extracted for each image. Finally, an early feature fusion is conducted to produce the final fused features for final similarity score computation.

scene graphs can separate one car from another, producing a more discriminative similarity score. Nonetheless, even with scene graphs, semantic information alone can be noisy, and it must be used in complement with visual-based matching.

In this study, we propose a novel VPR frame work called NetVLAD with Scene Graph Generation (SGG-NetVLAD) that combines the visual representation of NetVLAD variant methods and semantic representation from a scene graph generator. The two modalities are then fused together to produce a more robust VPR estimation to leverage the strength of both methods. An overview of the framework of SGG-NetVLAD is shown in Fig 3.1

3.2 Method

Overview: The goal of SGG-NetVLAD is to produce a more robust similarity score for visual place recognition. We do so by leverage the State-of-the-art VPR method, namely, Patch-

NetVLAD [3] as a visual feature extractor and the semantic output of a scene graph generator from [122].

SGG-NetVLAD consists of two separate branches for feature extraction, namely, the visual branch and semantic branch. Given an input image I, we first utilize the global feature search with NetVLAD and get a coarse matching of the top *k* (in experiment, *k*=100) most similar images to the query from the database. This step is to efficiently get a ranking list reasonable in size from a large-scale database that can be used for later re-ranking. Next, we take the image and extract its visual feature f_{vis} using Patch-NetVLAD, and generate a scene graph S_I. S_I with scene graph generator (SGG). The scene graph is then embedded to a semantic feature f_{sem} with label weighting. Finally, f_{vis} and f_{sem} are then feed into a feature fusion network to produce the fused feature f_{fused}, *i.e.*,early feature fusion, for final similarity score computation. The same is done for both images of the query-database image pair and a similarity score is computed for re-ranking. In section 3.2.1 and 3.2.2, we review the backbones architectures we used for the visual branch and semantic branch, respectively. Then, in 3.2.3, we discuss our proposed feature fusing module with semantic label weighting.

3.2.1 The Visual Branch

We first explain here in details why Patch-NetVLAD is such a powerful backbone and is suitable for the visual branch of SGG-NetVLAD.

In order to achieve their objective, [3] present several contributions illustrated in Figure 1. Firstly, [3] propose a novel place recognition system that calculates a similarity score between a pair of images through a spatial score, obtained via exhaustive matching of locally-global descriptors. These descriptors are generated for densely-sampled local patches within the feature space using a VPR-optimized aggregation technique, specifically NetVLAD [5]. Secondly, [3] intro-

duce a multi-scale fusion method that creates and merges these hybrid descriptors of various sizes to achieve enhanced performance compared to a single scale approach. To minimize the impact of moving to a multi-scale approach on computational requirements, we develop an integral feature space (akin to integral images) to derive the local features for different patch sizes. Together, these contributions offer users flexibility based on their specific task needs. Our final contribution is the demonstration of a range of readily implementable system configurations that attain various performance and computational balances, including a performance-focused setup that achieves state-of-the-art recall performance when tight error thresholds are necessary, a balanced configuration that performs nearly as well as the state-of-the-art while being three times faster than SuperGlue [123] and 28 times faster than DELG[124], and a speed-focused configuration that is at least ten times faster than the state-of-the-art.

Patch-NetVLAD evaluated the effectiveness of their proposed system on various datasets commonly used in VPR research. They compared Patch-NetVLAD with state-of-the-art global feature descriptor methods, as well as with the recent local descriptor method DELG [124] and new SuperPoint [125] and SuperGlue [123]-enabled VPR pipelines as competitive baselines. The results showed that Patch-NetVLAD outperformed global feature descriptor methods by large margins (up to 330% relative increase) across all datasets and achieved superior performance (up to a relative increase of 54%) compared to SuperGlue. While Patch-NetVLAD performed worse than DELG in some datasets, its order-of-magnitude faster computation speed makes it more practical in real-world scenarios. Patch-NetVLAD was also the winner of the Facebook Mapillary Long-term Localization Challenge as part of the ECCV 2020 Workshop on Long-Term Visual Localization. To gain a detailed understanding of the system's properties, the researchers conducted numerous ablation studies to analyze the role of individual components of Patch-NetVLAD and showed the system's robustness to changes in various parameters.

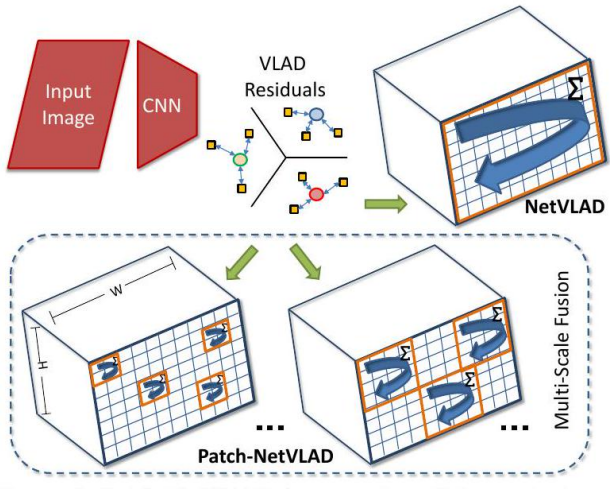


Figure 3.2: An overview of the framework of Patch-NetVLAD [3]. Patch-NetVLAD utilizes local matching of locally-global descriptors extracted from patches in each image's feature space to produce a similarity score. For our study, we only use Patch-NetVLAD as a visual feature extractor.

Patch-NetVLAD ultimately produces a similarity score between a pair of images, measuring the spatial and appearance consistency between these images. Our hierarchical approach first uses the original NetVLAD descriptors to retrieve the top-k (we use k = 100 in our experiments) most likely matches given a query image. We then compute a new type of patch descriptor using an alternative to the VLAD layer used in NetVLAD [5], and perform local matching of patch-level descriptors to reorder the initial match list andrefine the final image retrievals. This combined approach minimizes the additional overall computation cost incurred by cross matching patch features without sacrificing recall performance at the final image retrieval stage. An overview of the complete pipeline of [3] can be found in Fig 3.2.

The original NetVLAD network [5] architecture uses the Vector-of-Locally-Aggregated-Descriptors (VLAD) approach to generate a condition and viewpoint invariant embedding of an image by aggregating the intermediate feature maps extracted from a pre-trained Convolutional Neural Network (CNN) used for image classification [56]. Specifically, let $f_\theta : I \rightarrow \mathbb{R}^{H \times W \times D}$ be the base

architecture which given an image I, outputs a $H \times W \times D$ dimensional feature map F (e.g. the conv5 layer for VGG). The original NetVLAD architecture aggregates these D-dimensional features into a $K \times D$-dimensional matrix by summing the residuals between each feature $\mathbf{x}_i \in \mathbb{R}^D$ and K learned cluster centers weighted by soft-assignment.Formally, for $N \times D$-dimensional features, let the VLAD aggregation layer $f_{\text{VLAD}} : \mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{K \times D}$ be given by

$$f_{VLAD}(F)(j,k) = \sum_{i=1}^{N} \bar{a}_k \left(\mathbf{x}_i\right) \left(x_i(j) - c_k(j)\right) \tag{3.1}$$

where $x_i(j)$ is the j^{th} element of the i^{th} descriptor, $\bar{a}_k$ is the soft-assignment function and $\mathbf{c}_k$ denotes the k^{th} cluster center. After VLAD aggregation, the resultant matrix is then projected down into a dimensionality reduced vector using a projection layer $f_{\text{proj}} : \mathbb{R}^{K \times D} \rightarrow \mathbb{R}^{D_{\text{proj}}}$ by first applying intra(column)-wise normalization, unrolling into a single vector, L2-normalizing in its entirety and finally applying PCA (learned on a training set) with whitening and L2-normalization [5].

The main contribution of [3] is extracting Patch-level Global Features, [3] extract a set of $d_x \times d_y$ patches $\{P_i, x_i, y_i\}_{i=1}^{n_p}$ with stride s_p from the feature map $F \in \mathbb{R}^{H \times W \times D}$, where the total number of patches is given by

$$n_p = \left\lfloor \frac{H - d_y}{s_p} + 1 \right\rfloor * \left\lfloor \frac{W - d_x}{s_p} + 1 \right\rfloor, d_y, d_x \leq H, W \tag{3.2}$$

and $P_i \in \mathbb{R}^{(d_x \times d_y) \times D}$ and x_i, y_i are the set of patch features and the coordinate of the center of the patch within the feature map, respectively.

For each patch, a descriptor is subsequently extracted yielding the patch descriptor set $\{\mathbf{f}_i\}_{i=1}^{n_p}$ where $\mathbf{f}_i = f_{\text{proj}}\left(f_{\text{VLAD}}\left(P_i\right)\right) \in \mathbb{R}^{D_{\text{proj}}}$ uses the NetVLAD aggregation and projection layer on the relevant set of patch features.

Even though there are further parts to the Patch-NetVLAD methods such as finding mutual

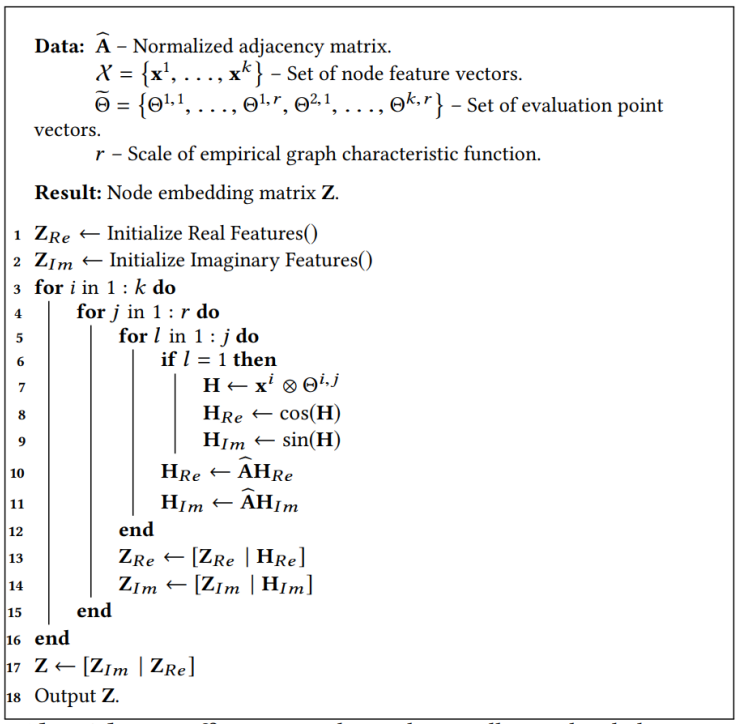
nearest neighbors and spacial scoring for similarity score calculation, we will not review it here since we are only interested in the extracted visual feature.

Patch-NetVLAD takes global descriptor techniques and employed them to enhance the robustness of local descriptors to appearance changes, hitting SOTA performances on various benchmarks, and hense suitable for our need as the visual feature extractor for the visual branch of SGG-NetVLAD.

3.2.2 Semantic Branch

For scene graph generator, we choose the publicly available and powerful performance-wise scene graph generator proposed in [122], which focuses on solving the problem of generating unbiased scene graphs to alleviate the severe training bias, *e.g.*, collapsing diverse human walk on/ sit on/lay on beach into human on beach, providing a more reliable scene graph. Once the scene graph S_I is generated, we then need to embed it into a semantic feature that represents the semantic structure of the image, namely, f_{sem}. This is done by utilizing the FEATHER graph embedding algorithm [126], whose algorithm is briefly reviewed here, and I quote:

"FEATHER [126] evaluates the characteristic functions for a graph for each feature $\mathbf{x} \in \mathcal{X}$ at all scales up to r. The connectivity of the graph is described by the normalized adjacency matrix $\widehat{\mathbf{A}}$. For each feature vector $\mathbf{x}^i, i \in 1, \ldots, k$ at scale r we have a corresponding characteristic function evaluation vector $\Theta^{i,r} \in \mathbb{R}^d$. For simplicity, assume that we evaluate the characteristic functions at the same number of points. Let us look at the mechanics of Algorithm 1. First, we initialize the real and imaginary parts of the embeddings denoted by $\mathbf{Z}_{Re}$ and $\mathbf{Z}_{\text{Im}}$ respectively (lines 1 and 2). Iterate over the k different node features (line 3) and the scales up to r (line 4). When we consider the first scale (line 6) we calculate the outer product of the feature being considered and the corresponding evaluation point vector this results in $\mathbf{H}$ (line 7). Then, elementwise take the sine



Data: $\widehat{\mathbf{A}}$ – Normalized adjacency matrix.
$\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$ – Set of node feature vectors.
$\widetilde{\Theta} = \{\Theta^{1,1}, \ldots, \Theta^{1,r}, \Theta^{2,1}, \ldots, \Theta^{k,r}\}$ – Set of evaluation point vectors.
r – Scale of empirical graph characteristic function.

Result: Node embedding matrix $\mathbf{Z}$.

1 $\mathbf{Z}_{Re} \leftarrow$ Initialize Real Features()
2 $\mathbf{Z}_{Im} \leftarrow$ Initialize Imaginary Features()
3 **for** i in $1 : k$ **do**
4 **for** j in $1 : r$ **do**
5 **for** l in $1 : j$ **do**
6 **if** $l = 1$ **then**
7 $\mathbf{H} \leftarrow \mathbf{x}^i \otimes \Theta^{i,j}$
8 $\mathbf{H}_{Re} \leftarrow \cos(\mathbf{H})$
9 $\mathbf{H}_{Im} \leftarrow \sin(\mathbf{H})$
10 $\mathbf{H}_{Re} \leftarrow \widehat{\mathbf{A}}\mathbf{H}_{Re}$
11 $\mathbf{H}_{Im} \leftarrow \widehat{\mathbf{A}}\mathbf{H}_{Im}$
12 **end**
13 $\mathbf{Z}_{Re} \leftarrow [\mathbf{Z}_{Re} \mid \mathbf{H}_{Re}]$
14 $\mathbf{Z}_{Im} \leftarrow [\mathbf{Z}_{Im} \mid \mathbf{H}_{Im}]$
15 **end**
16 **end**
17 $\mathbf{Z} \leftarrow [\mathbf{Z}_{Im} \mid \mathbf{Z}_{Re}]$
18 Output $\mathbf{Z}$.

Figure 3.3: The overview of the FEATHER algorithm for graph embedding[126]

and cosine of this matrix (lines 8 and 9). For each scale we calculate the real and imaginary parts of the graph characteristic function evaluations ($\mathbf{H}_{Re}$ and $\mathbf{H}_{Im}$) - we use the normalized adjacency matrix to define the probability weights (lines 10 and 11). Next, append these matrices to the real and imaginary part of the embeddings (lines 13 and 14). When the characteristic function of each feature is evaluated at every scale we concatenate the real and imaginary part of the embeddings (line 17) and we return this embedding (line 18)."

The output semantic feature from scene graph embedding has a dimension of $H \times K$ where K is the number of semantic entities/objects we choose to embed, usually the top-k most reliable ones given by the scene graph generator, and H is the dimension of the embedding for each semantic entity/object. Thanks to this structure, it is easy for us to implement a learnable weight map w_{label} that contains the weight for all possible semantic labels. w_{label} is trained by back propagating the

loss from the final fused feature similarity score calculation. To put simply, we want to adjust the weight for all the possible semantic labels so that it fits the setting of the database. The goal is to not only learn lower weights for dynamic objects, *e.g.* cars, humans, so that we can focus on the scene itself when calculating similarity scores, but also to learn the weight so that we can focus on semantic objects that are more discriminative to the specific setting of the database. For example, for a database built with railway images, semantic objects such as ”mountains” or ”trees” are everywhere and exist in most database images, but ones like ”buildings” or ”houses” might be very discriminative because they appear much less often than the former ones, so we want to learn a larger weight on the latter semantic objects. However, in an urban area setting, ”buildings” or ”houses” are now the confusing ones, and we want to lower they weights in the semantic embedding feature f_{sem}. As we can see, having a learnable weight for semantic labels helps us to generate more discriminative semantic features comparing to pixel-level semantic label only approaches [127].

3.2.3 Fusing Visual and Semantic Features

Once we extracted both f_{vis} and f_{sem} for both images, we feed them into a siamese network to learn their joint embeddings as their fused features. A Siamese Neural Network is a class of neural network architectures that contain two or more identical subnetworks. That is, they have the same configuration with the same parameters and weights, and parameter updating is mirrored across both sub-networks. The process is briefly illustrated in 3.4.

Our loss function is a contrastive loss that aims to learn a smaller distance when two images belongs to the same place, and a larger distance otherwise:

$$L = \frac{1}{2} Y d(W, w_{label})^2 + \frac{1}{2}(1 - Y) \max(m - d(W, w_{label}), 0)^2 \tag{3.3}$$

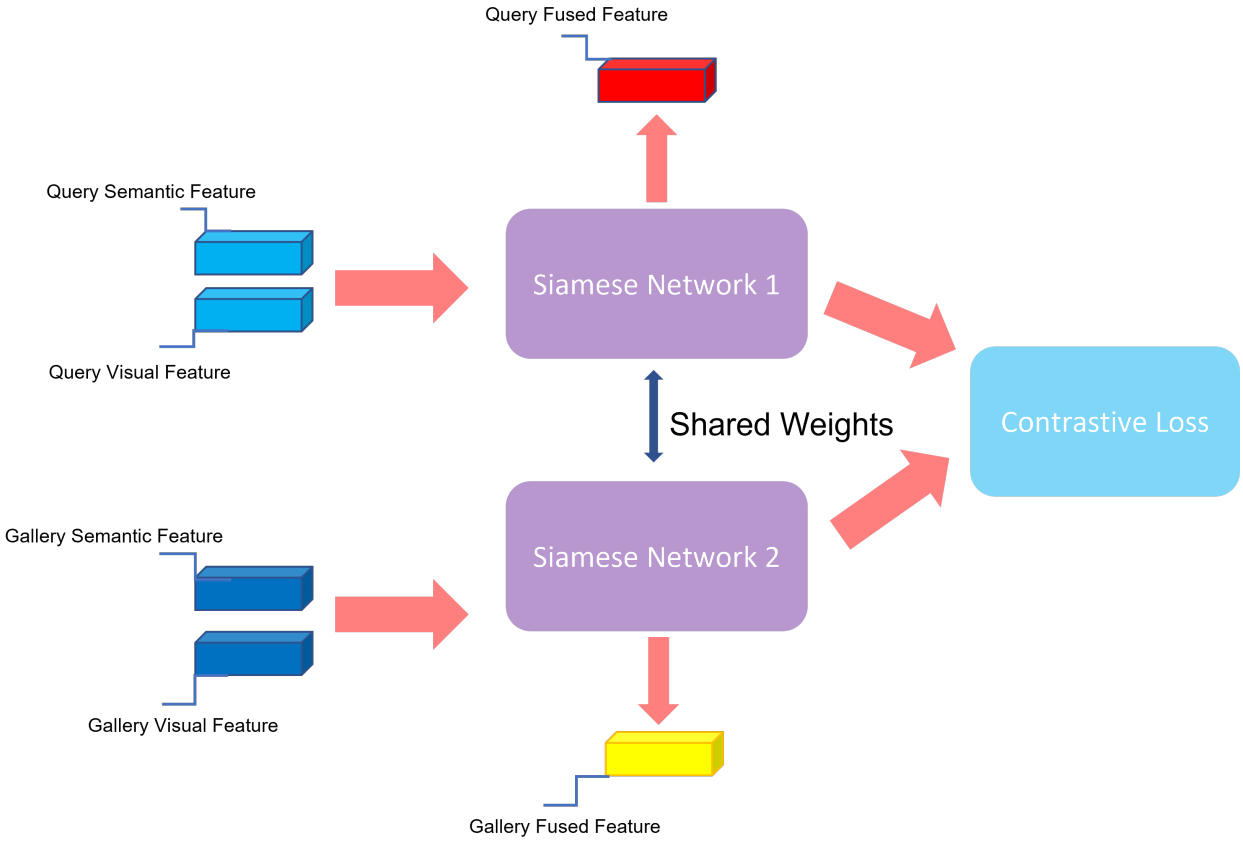


Figure 3.4: An overview of the feature fusion process, f_{vis} and f_{sem} are feed into the a siamese network and trained by a contrastive loss as shown in equation 3.3

where the distance d is calculated with cosine distance S_c as:

$$d(W, w_{label}) = S_c(f_{q_{fused}}(W, w_{label}), f_{d_{fused}}(W, w_{label})) \tag{3.4}$$

here, $f_{q_{fused}}$ and $f_{d_{fused}}$ are the fused feature vector for the query image and the database image, respectively; to be more specific, let F_W be the siamese network mapping function, then we have $f_{q_{fused}} = F_W(f_{q_{vis}}\ ,\ w_{label} \cdot f_{q_{sem}})$ and same goes for $f_{d_{fused}}$. m is a hyper parameter of margin for the contrastive loss. w_{label} is the label weight map and W is the weight shared on both branches of the siamese network. Y is the indicator of whether the query image and the database image belong to the same place, since we have geo-tags during training, we can calculate the distance between the two geo-tags and if their distance is less than d (d is pre-defined distance threshold provided by datasets), we consider them to be the same location, then $Y = 1$ if two images belong to the same location and $Y = 0$ otherwise. Through training, W and w_{label} are updated through back propagation.

3.3 Experimental Results

3.3.1 Implementation

We follow the configuration of [3] and implement SGG-NetVLAD in PyTorch and resize all images to 640 by 480 pixels before extracting visual features. For semantic feature extraction, we did not resize the image, but the locations of the bounding boxes for semantic objects are resized accordingly to so that they are consistent with the visual features. We train on the training set of Pittsburgh 30k [128], same as Patch-NetVLAD for urban datasets, and the spring and autumn subsets of the Nordland dataset for railway setting. The scene graph generator is pre-trained on the Visual Gnome dataset [129] and we use it off-the-shelf.

3.3.2 Datasets and Evaluation

SGG-NetVLAD is evaluated on three key benchmarks in VPR, namely, Pittsburg30k [128], Tokyo 24/7 [130], and Nordland [131]. Datasets were used in their recommended configuration for benchmarking (*e.g.* removal of pitch black tunnels and times when the train is stopped for the Nordland dataset). Pittsburgh30k is a urban setting dataset where the images are taken in the urban area of Pittsburgh, all in daytime. The dataset is divided into training set and testing set, each containing 13k and 17k images, respectively. For testing set, there are 7k query images and 10k database images. The Tokyo 24/7 dataset contains 300 query images and 70k database images taken from the urban area of Tokyo with both daytime and nighttime images, making it a challenging dataset. Nordland is the most challenging dataset among the three, is a railway-view dataset taken from the front of a train. Nordland has 30k images for each season. For experiment, we follow the settings from [3] and use winter images for the queries and summer images for the database.

3.3.3 Comparison to State-of-the-art Methods

We compare against several benchmark localization solutions based on retrieval:APGEM [132], DenseVLAD [130], NetVLAD [5], and Patch-NetVLAD [3].

Table 3.1 contains the quantitative comparisons of SGG-NetVLAD against the baseline methods. As we can observe from the results, SGG-NetVLAD is extremely effective in the Nordland dataset, which is a dataset that contains seasonal variations, resulting a stunning **26.5%** improvement in top-1 recall comparing to the SOTA VPR method Patch-NetVLAD, demonstrating the capability of the SGG-NetVLAD model to produce discriminative similarity score in settings where dramatic visual appearance changes are present. Note that for Nordland we used a different distance threshold, where Patch-NetVLAD used 10 frames, we used 1. Also, we notice that our per-

Table 3.1: **Quantitative results** on Nordland [131], Pittsburgh 30k [128] and Tokyo 24/7 [130]

Methods	**Nordland**			**Pittsburgh 30k**			**Tokyo 24/7**		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
AP-GEM [132]	11.1	13.2	16.1	80.7	91.4	94.0	58.4	69.5	74.3
DenseVLAD [130]	11.9	20.8	26.2	78.2	88.8	92.3	59.4	67.0	71.8
NetVLAD [5]	10.4	16.3	19.7	85.1	92.2	94.4	64.4	78.4	81.6
Patch-NetVLAD [3]	25.6	37.6	42.2	**88.7**	94.5	95.9	**86.0**	88.6	90.5
Ours	**32.4**	**42.6**	**45.2**	88.6	**94.6**	**96.1**	85.8	**88.9**	**90.5**

formance is slightly worse than Patch-NetVLAD at top-1 recall for both Pittsburgh 30k and Tokyo 24/7 dataset, although we slightly beat Patch-NetVLAD at top-5 and top-10 recall. This is because for these two datasets, although appearance changes exist, the changes are not as significant as the ones in Nordland. Also, for urban image datasets such as Pittsburgh 30k and Tokyo 24/7, there are more visual diversity and makes it easier for visual feature-only approaches to achieve better performances; in other words, visual feature can handle such settings rather well already. Our contribution here is that we achieve a remarkable improvement in Nordland while maintaining the performance on Pittsburgh 30k and Tokyo 24/7, thanks to the feature fusing network and semantic label weighting. We also conducted an ablation study to show that it is our feature fusing network and label weighting that keeps the semantic branch from undermining the visual branch when visual features are working fine on their own.

For ablation studies, we evaluate the performance of SGG-NetVLAD when we remove the proposed components. As we can observe from table 3.2, the semantic branch cannot achieve a reasonable performance on its own, and this is exactly why it has to be used as a complimentary to the visual branch. Removing the label weighting component means we treat all the semantic labels equally, and that also results in a drop in performance. We demonstrate that our proposed feature fusing component and semantic label weighting is the key for SGG-NetVLAD to outperform SOTA methods on difficult dataset such as Nordland while maintaining a comparable performance for

Table 3.2: **Ablation Studies** on Semantic Branch and Label Weighting

Methods	Nordland			Pittsburgh 30k			Tokyo 24/7		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Semantic Only	15.2	21.6	30.2	25.6	37.6	42.2	32.3	41.6	53.2
SGG-VLAD(w/o label weighting)	30.2	38.5	44.6	75.4	88.2	90.5	80.7	84.8	88.2
SGG-VLAD	**32.4**	**42.6**	**45.2**	**88.6**	**94.6**	**96.1**	**85.8**	**88.9**	**90.5**

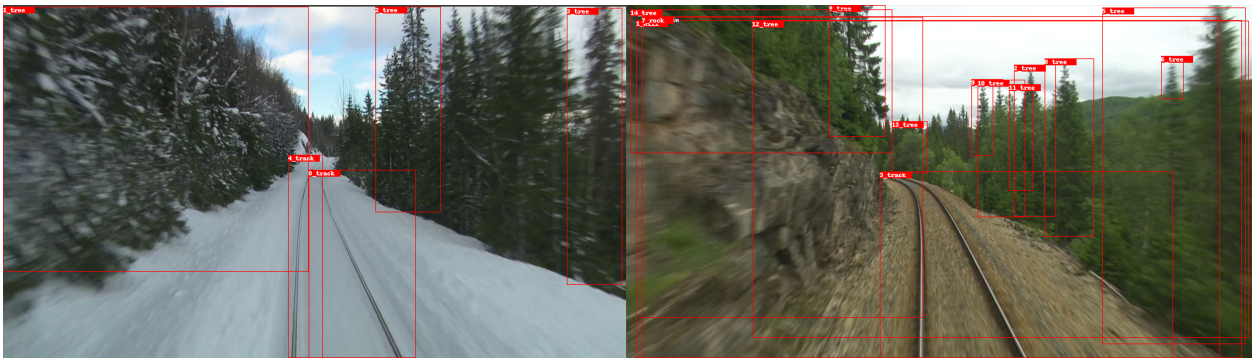


Figure 3.5: Examples of generated scene graphs from Nordland

datasets where visual features already work fine on their own.

Finally, we provide some qualitative results including the visualization of the generated scene graphs from Nordland and Pittsburgh 30k, shown in Figure 3.5 and Figure 3.6, as well as some examples where SGG-NetVLAD retrieves the correct matching result while Patch-NetVLAD fails to do so. These results are shown in Figure 3.7 and Figure 3.8. We also provide some examples shown in Figure 3.9 and Figure 3.10 with results comparison given the generated scene graphs. From the comparison, we can see that the existence of semantic objects such as ”sign” helps the model to distinguish the confusing incorrect match retrieved by Patch-NetVLAD.

3.4 Concluding Remarks

In this study, we proposed a novel framework called SGG-NetVLAD that combines the powerful visual feature from Patch-NetVLAD [3] and semantic feature from embedding scene graph gen-

Figure 3.6: Examples of generated scene graphs from Pittsburgh 30k

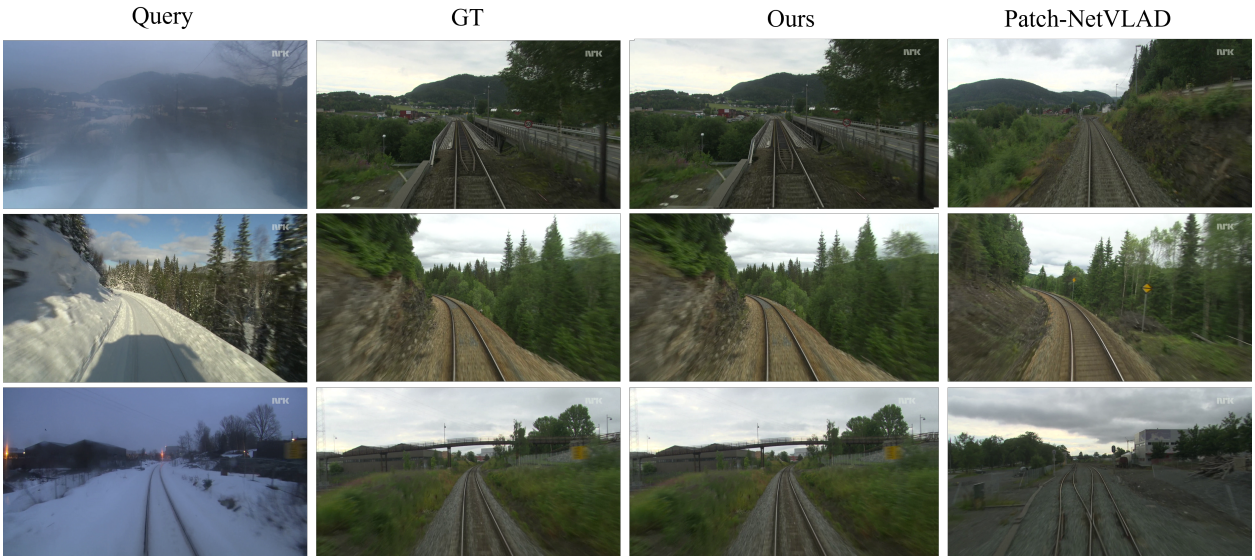


Figure 3.7: Qualitative results from Nordland. Examples where SGG-NetVLAD retrieves the correct matching result while Patch-NetVLAD fails to do so. As we can observe from the figures, the incorrectly retrieved results from Patch-NetVLAD are highly similar to the ground-truth match visually.

Figure 3.8: Some more examples of qualitative results from Pittsburgh 30k

Figure 3.9: Qualitative results. Examples from Pittsburgh 30k with SGG visualized, as we can observe, the existence of semantic objects such as ”sign” helps the model to distinguish the confusing incorrect match.

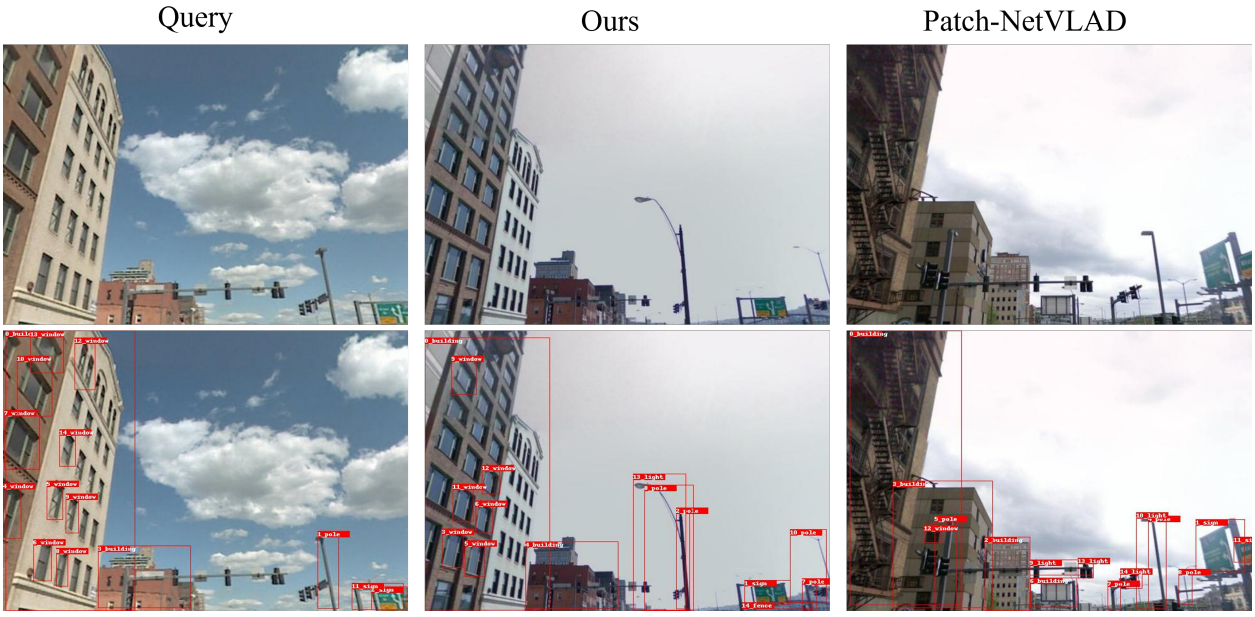


Figure 3.10: More qualitative results from Pittsburgh 30k with scene graph visualized

erated from [122] with semantic label weighting. The features from two modalities are then fused with a siamese network to compute the final similarity score for re-ranking. Through experiments, SGG-NetVLAD proves to be exceptionally effective on dataset that includes seasonal changes such as Nordland [131] while maintaining comparable result for other urban setting dataset, indicating that SGG-VLAD is able to produce more discriminative similarity scores for queries with huge appearance changes while not undermining the effectiveness of the visual features, thanks to the semantic label weighting and feature fusion components. Some future step that could be considered include generating attention aware scene graphs and VLAD features for better discriminativities.

CHAPTER 4
CO-VISUAL PATTERN-AUGMENTED GENERATIVE TRANSFORMER LEARNING FOR AUTOMOBILE GEO-LOCALIZATION

4.1 Background

In recent years, geolocation identification of automobiles has become an increasingly popular topic due to its potential applications in navigation and route planning for intelligent vehicles [13, 14, 15, 16, 17, 18, 19]. Traditionally, obtaining the geographic location of a vehicle has been achieved through Global Navigation Satellite Systems (GNSS), which is a convenient and cost-effective method. However, GNSS signals are susceptible to being unreliable or unavailable in scenarios with dense high-rise obstacles, network failures, and other factors. Examples of such scenarios, such as dense primordial forests and crowded buildings, are depicted in Figure 1.2. Fortunately, current satellite images can cover most outdoor scenarios where automobiles are present and can be easily collected offline in advance through services like Google Maps. To overcome the limitations of GNSS, the use of registered ground-satellite image retrieval for geographic location estimation has gained increasing attention [5, 20, 21, 22, 23, 24, 25]. This method involves comparing visual data obtained from the vehicle with geo-tagged references stored in a database to estimate the geographic location that aligns with the closest reference. The schematic illustration of this pipeline is presented in Figure 1.2.

Typically, geolocation involves the collection of perspectives from sites previously visited by vehicles. Upon subsequent revisits, these views can be compared with similar scene content, constituting a loop closure detection process. In the event of satellite signal failure, the agent is required to determine its position by analyzing the contextual scene. Such methodologies are

designed to mitigate ambiguity by exploring and encoding contextual information and deep semantics. The images are encoded by identical or Siamese-like backbone networks, followed by nearest-neighbor matching. Thus, the geolocation task is akin to image retrieval, albeit with a primary focus on capturing and leveraging geometric and structural information of the environmental features that constitute the scene. Such information may include, but is not limited to, edge and corner features, shapes, and their relative positions, all of which are fundamental to effective geolocation. Consequently, geolocation requires a more nuanced understanding of the scene content than traditional image retrieval, as it must incorporate this rich geometric and structural information into its matching algorithm to achieve accurate results. Overfeat [26] was a pioneering deep learning-based study in the field and inspired a series of improvements [5, 27, 28, 29, 30]. To construct a reference data set with GPS information, these approaches examine the ground-to-ground matching procedure for localization by gathering views at diverse locations at different times, seasons, and weather conditions, as exemplified by Google Street View, a widely-used application. During the localization phase, views with unknown locations are matched with reference sets to estimate their locations. Despite their effectiveness, these methods are labor-intensive and cannot locate places that are not in the reference dataset. Therefore, researchers are striving to establish interconnectivity between satellite views and ground views by extracting the intrinsic similarities between the two view types, namely cross-view geo-localization (CVGL), which increases the generalization performance of the location model. Owing to the dissimilar imaging perspective between satellite and ground views, the appearance of content varies significantly, posing a substantial challenge in achieving cross-view localization. Nonetheless, researchers have made remarkable strides in devising Siamese-like networks that contain two distinct branches responsible for encoding each view independently [133, 127, 25, 134, 76, 135, 136, 137, 138, 69, 70]. While the relationship between different views provides a significant impetus for cross-view localiza-

tion, several challenges persist. First, semantic consistency between views is not fully leveraged. Current methods typically utilize Siamese-like networks for independent encoding of cross-view views but often neglect the high-order consistency semantics of view content, which is essential for matching ground and satellite images. Second, co-visual relationships between views are not explicitly accounted for. The perspective disparities between ground and satellite views limit co-visual relationships exploring, with the latter typically encompassing a more extensive scope; thus, using the whole image for coding would yield suboptimal accuracy. Third, deep contextual semantic mining is not yet sufficient. As the interaction between views remains unconsidered, the existing methods fail to fully explore contextual semantics.

We propose a novel approach, called mutual generative transformer learning (MGTL), to address the deficiencies of current methods in cross-view geo-localization (CVGL). Our method includes a cascaded attention masking algorithm to create network reasoning for co-visual patterns between ground and satellite views, as well as two symmetrical generative sub-modules: Ground-to-Satellite (G2S) and Satellite-to-Ground (S2G). These sub-modules generate simulated cross-view knowledge to capitalize on the mutual benefits across views. S2G simulates ground-aware knowledge using aerial semantics, and vice versa. The view-specific simulated knowledge is then used to enhance current view features through attention learning. All sub-components work together within a transformer-based framework to accomplish the CVGL task. Our approach outperforms existing methods on several challenging public benchmarks, demonstrating its effectiveness.

4.2 Contributions

The contributions of the proposed MGTL can be summarized as follows:

- A novel cross-view knowledge-guided learning approach for CVGL. To the best of our

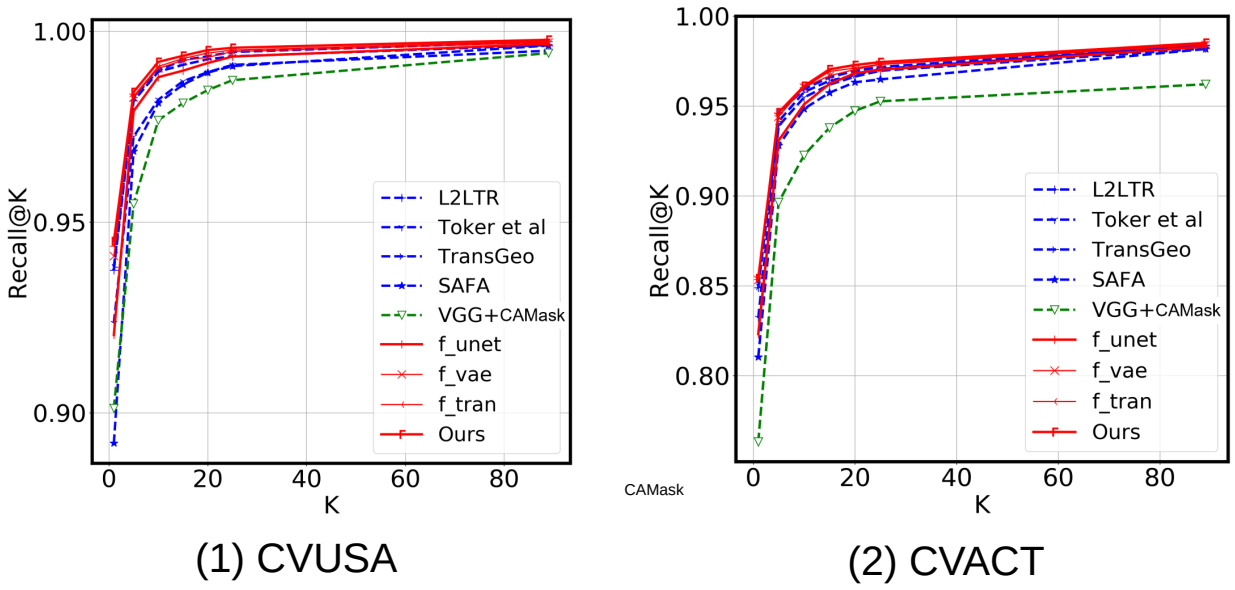


Figure 4.1: The proposed MGTL outperforms existing approaches.

knowledge, the MGTL is the first attempt to build mutual interactions between ground-level and aerial-level patterns in the CVGL community. Unlike existing transformer-based CVGL models that only perform self-attentive reasoning in the respective view, our proposed MGTL produces cross-knowledge information to achieve more representative high-order features.

- Cascaded attention-guided masking to exploit the co-visual patterns. Instead of treating patterns in aerial and ground views equally, we developed an attention-guided exploration algorithm to create network reasoning based on the co-visual patterns, which further improves performance.

- State-of-the-art localization accuracy on widely-used benchmarks. The proposed MGTL outperforms existing deep models on various datasets, i.e., *CVUSA* [139] and *CVACT* [140], as shown in Figure 4.1.

4.3 Method

4.3.1 Problem Formulation

Let the cross-view geo-localization (CVGL) model be indicated as the function $\mathcal{F}_\Theta$ parameterized by weights Θ, which takes an image pair consisting of a ground-view image $\mathbf{I}_G$ and a satellite-view image $\mathbf{I}_S$ as input and produces their corresponding representations $\mathbf{F}_G$ and $\mathbf{F}_S$. Our goal is to learn Θ from the labeled training triplets $\{\mathbf{I}_G^i, \mathbf{I}_{SP}^i, \mathbf{I}_{SN}^i\}_{i=1}^N$ to make $\mathbf{F}_G$ and $\mathbf{F}_S$ closer while their corresponding cross-view images are matching, where $\mathbf{I}_G^i$ is the ground-view image and $\mathbf{I}_{SP}^i$ and $\mathbf{I}_{SN}^i$ are the positive and negative samples relative to $\mathbf{I}_G^i$, respectively. The process can be formulated as follows:

$$\begin{aligned} &\mathbf{F}_G, \mathbf{F}_S = \mathcal{F}_\Theta(\mathbf{I}_G; \mathbf{I}_S), \\ &||\mathbf{I}_G^i - \mathbf{I}_{SP}^i||^2 + \alpha < ||\mathbf{I}_G^i - \mathbf{I}_{SN}^i||^2 \end{aligned} \tag{4.1}$$

where α is the margin in the triplet loss.

4.3.2 View-Independent Feature Extractor ($f_{\texttt{VIFE}}$)

Overview: We retained the initial 13 convolutional layers in VGG16 [52] and split them into 5 stages according to spatial resolutions in order to extract high-order features from input images. Then, we designed a cascaded attention-masking (CAMask) algorithm for learning fine-grained co-visual relationships by cascading multi-branch convolutional modules. Figure 4.2 illustrates the overview of our proposed mutual generative transformer learning (MGTL). As mentioned above, $f_{\texttt{VIFE}}$ takes an image pair $<\mathbf{I}_G, \mathbf{I}_S>$ as input and produces two view-specified semantic representations $<\mathbf{F}_G^{'}, \mathbf{F}_S^{'}>$ and corresponding spatial attention masks $<\mathbf{M}_G, \mathbf{M}_S>$, following Equations (4.2) and (4.3). Additionally, we have listed the main abbreviations in Table 4.1 for ease

Table 4.1: List of Abbreviations.

Abbreviation	Explanation
CVGL	Cross-view geo-localization
MGTL	Mutual generative transformer learning
CAMask	Cascaded attention masking
CVI	Cross-view interaction
G2S	Ground-to-satellite
VIFE	View-independent feature extractor
S2G	Satellite-to-ground
SA	Spatial attention
SCE	Spatial context enhancement
MSFA	Multi-scale feature aggregation
GKST	Generative knowledge-supported transformer

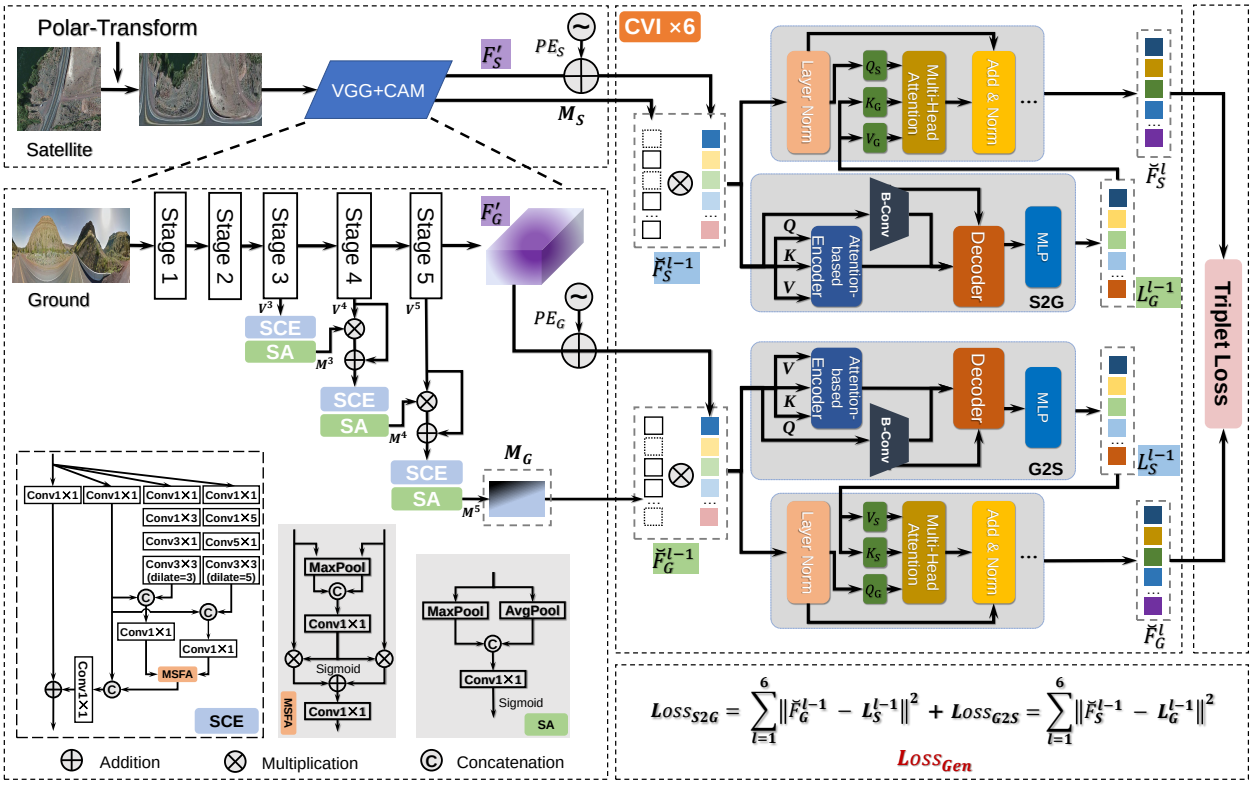
of reference.

Feature Extractor: Formally, given an image pair $\mathbf{I}_G \in \mathbb{R}^{H_1 \times W_1 \times 3}$ and $\mathbf{I}_S \in \mathbb{R}^{H_2 \times W_2 \times 3}$, a multi-branch backbone (i.e., a Siamese-like VGG-based convolutional network with parameters $\Theta_{\texttt{VIFE}}$) is used to extract features and generate spatial attention masks for each view simultaneously:

$$\mathbf{F}_G^{'} = f_{\texttt{VIFE}}(\mathbf{I}_G; \Theta_{\texttt{VIFE}}); \mathbf{F}_S^{'} = f_{\texttt{VIFE}}(\mathbf{I}_S; \Theta_{\texttt{VIFE}}) \tag{4.2}$$

where $\mathbf{F}_G^{'} \in \mathbb{R}^{c \times h \times w}$ and $\mathbf{F}_S^{'} \in \mathbb{R}^{c \times h \times w}$ are semantic representations with c channels and $h \times w$ spatial resolutions for ground-view and satellite-view, respectively.

Cascaded Attention Masking: Viewpoint changes result in drastic appearance differences, which means much redundant information exists in $\mathbf{F}_G^{'}$ and $\mathbf{F}_S^{'}$ while matching. To encourage the



$$Loss_{S2G} = \sum_{l=1}^{6} \|\breve{F}_G^{l-1} - L_S^{l-1}\|^2 + Loss_{G2S} = \sum_{l=1}^{6} \|\breve{F}_S^{l-1} - L_G^{l-1}\|^2$$

$$Loss_{Gen}$$

Figure 4.2: Overview of the proposed MGTL.

network to focus on the co-visual regions, we designed a cascaded attention-masking (CAMask) algorithm and integrated it into the VGG16 backbone [52], seeking to learn the spatial attention masks that inhibit the non-co-visual areas adaptively. Figure 4.2 (left) illustrates the basic structure of the CAMask. Generally, the CAMask takes the side-output features $\{\mathbf{V}^i\}_{i=3}^5$ generated by the backbone as input and produces spatial attention masks $\mathbf{M}$ to enhance the inter-view co-visual information. Specifically, the fine-grained feature map captured by spatial context enhancement (SCE) is fed into two parallel pooling layers (i.e. maxpooling and avgpooling) along the channel dimension to generate two single-channel feature maps, respectively. Subsequently, these feature maps are concatenated along the channel dimension, and a convolutional layer is employed to adaptively generate masks with $h \times w$ resolutions. Spatial attention (SA) is illustrated in Figure 4.2.

Note that, for the sake of brevity, $\mathbf{M}$ can refer to $\mathbf{M}_G$ or $\mathbf{M}_S$. The cascaded process can be formulated as follows:

$$\begin{aligned}\mathbf{M}^3 &= \mathbf{SA}(\mathbf{SCE}(\mathbf{V}^3)),\\ \mathbf{M}^4 &= \mathbf{SA}(\mathbf{SCE}(\mathbf{V}^4 \otimes \mathbf{M}^3 + \mathbf{V}^4)),\\ \mathbf{M}^5 &= \mathbf{SA}(\mathbf{SCE}(\mathbf{V}^5 \otimes \mathbf{M}^4 + \mathbf{V}^5)),\end{aligned} \tag{4.3}$$

where $\mathbf{M}^i$ represents the spatial attention mask of the i-th stage, $\mathbf{V}^i$ represents the feature map produced by the i-th stage in the backbone, and $\mathbf{M}^5 \in \mathbb{R}^{h \times w}$ is the final spatial attention mask $\mathbf{M}$. A better understanding of CAMask can be gained by focusing on its two components: spatial context enhancement (SCE) and spatial attention (SA).

Spatial Context Enhancement (SCE): To capture nuanced co-visual relationships, we meticulously devised a novel multi-branch convolutional module that effectively extracts fine-grained spatial representations from each view by utilizing diverse receptive fields. Fan [141] proposed a texture-enhanced module (TEM) consisting of multiple convolutional branches with different receptive fields. There is evidence that it facilitates the sensitive capture of small spatial shifts. There are, however, certain limitations to coarse direct concatenation in TEM when the convolutional branches are independent. Motivated by this, we designed the SCE equipped with the multi-scale feature aggregation (MSFA) module to integrate branches with the guidance of spatial attention mechanism. As shown in Figure 4.2 (left), the SCE includes a shortcut branch and three parallel residual branches $\{b_i\}_{i=1}^3$ with different dilation rates $d \in \{1,3,5\}$, respectively. The shortcut branch utilizes a 1×1 convolutional layer to generate h_0 with channel size C. The branch b_1 only contains a 1×1 convolutional layer to halve the channel, while the remaining two branches $\{b_i\}_{i=2}^3$ adopt a 1×1 convolutional layer to reduce the channel and consist of three convolutional layers, i.e., a $1 \times (2i-1)$ convolutional layer, a $(2i-1) \times 1$ convolutional layer, and a 3×3 convolutional layer with dilation rate $(2i-1)$, to fully explore the spatial context information with

rich receptive fields. Let $\{h_i\}_{i=1}^{3}$ represent the feature maps produced by the residual branches $\{b_i\}_{i=1}^{3}$, respectively. To fully explore the multi-scale information from the features $\{h_i\}_{i=1}^{3}$ generated by different convolutional layers, we carefully designed a **multi-scale feature aggregation (MSFA)** module by taking into account the specificities of spatial regions rather than concatenating them directly. Specifically, we concatenated the features $\{h_i\}_{i=2}^{3}$ with h_1 and fed the concatenated feature maps into a 1×1 convolutional layer to produce features $\{h_i^{'}\}_{i=2}^{3}$ with unified channel C. MSFA takes $\{h_i^{'}\}_{i=2}^{3}$ as input and produces attention-aware feature maps h_{msfa} that are then concatenated with h_1 followed by a 1×1 convolutional layer with a GeLU activation, then added up with h_0 to produce the final enhanced contextual feature representation.

Spatial Attention (SA): Inspired by [119], we learned the spatial attention masks according to the enhanced contextual representations adaptively. In detail, SA takes the enhanced feature produced by SCE and eliminates the channel dimension by adopting the maximum and average pooling layers. In order to generate the spatial attention masks $\mathbf{M}_G(\mathbf{M}_S) \in \mathbb{R}^{h \times w}$, we concatenate the compact features obtained from the pooling layers and then apply a 1×1 convolutional layer with sigmoid activation.

To alleviate the limitation of feature location on the receptive learning field, we re-encode the features $\mathbf{F}_G^{'}$ and $\mathbf{F}_S^{'}$ with position information and enrich the co-visual areas by multiplying the spatial attention masks $\mathbf{M}_G$ and $\mathbf{M}_S$ generated by the cascaded attention-masking (CAMask) algorithm:

$$\hat{\mathbf{F}}_G = (\mathbf{F}_G^{'} + \mathtt{PE_G})\mathbf{M_G}; \hat{\mathbf{F}}_\mathtt{S} = (\mathbf{F}_\mathtt{S}^{'} + \mathtt{PE_S})\mathbf{M_S} \tag{4.4}$$

where $\hat{\mathbf{F}}_G$, $\hat{\mathbf{F}}_S \in \mathbb{R}^{l \times c}$ are compact and position-aware feature representations and $l = h \times w$. Following [81], $\mathtt{PE_G}$ and $\mathtt{PE_S}$ are the positional encoding of feature maps $\mathbf{F}_G^{'}$ and $\mathbf{F}_S^{'}$, respectively.

4.3.3 Cross-View Synthesis

A key principle of our proposed mutual generative transformer learning (MGTL) is cross-view interaction (CVI), which is achieved by generating mutual simulated knowledge through cross-view generative modules $f_{\texttt{G2S}}$ and $f_{\texttt{S2G}}$ with the supervision of generative loss in Equation (4.10). We emphasize that the ground view cannot obtain the matched satellite view in advance during the evaluation/localization period, which makes it impossible to directly take one view as input and produce the features of another view in the training phase. Therefore, each generative module takes only the view feature from the self-branch as input to produce cross-view knowledge by using another view feature as supervision, which means that two sub-branches are completely decoupled while evaluating unlabeled image pairs. Generative modules are embedded in transformer layers, and the generative knowledge is utilized to calculate the *Key* and *Value* while performing the attention mechanism. Further, the generative module is trained with all the transformer parts to fully mine semantic consistency across views using the generative knowledge-supported transformer, which we called generative transformer learning. Figure 4.2 (right) illustrates the overview of the proposed cross-view interaction (CVI), whereby one view's information is taken as input to generate knowledge that is aware of another view. The co-visual enhanced and position-aware representations $\hat{\mathbf{F}}_G$ and $\hat{\mathbf{F}}_S$ are further normalized as $\breve{\mathbf{F}}_G = \texttt{LN}(\hat{\mathbf{F}}_G)$ and $\breve{\mathbf{F}}_S = \texttt{LN}(\hat{\mathbf{F}}_S)$ to maintain representational capacity, respectively, where $\texttt{LN}$ indicates the linear encoding operation following layer normalization. As shown in Figure 4.2 (right), the cross-view interaction module $f_{\texttt{CVI}}$ is constructed by coupling two generative sub-modules $f_{\texttt{G2S}}$ and $f_{\texttt{S2G}}$ with an encoder–decoder structure as follows:

$$\mathbf{L}_S = f_{\texttt{G2S}}(\breve{\mathbf{F}}_G), \mathbf{L}_G = f_{\texttt{S2G}}(\breve{\mathbf{F}}_S). \tag{4.5}$$

Cross-View Generative Module $f_{\texttt{G2S}}$ and $f_{\texttt{S2G}}$: Unet-like [142] architecture comprising an

encoder and a decoder has been widely used in generative tasks recently. Existing sduties [80, 136] demonstrate that the attention mechanism in a transformer is excellent at modeling global contextual information, and CNN excels at encoding local semantic information. With these properties in mind, we propose a novel generative module that owns Unet-like [142] architecture and combines multi-head self-attention and convolutional layers in parallel for mutual benefit. Taking $f_{\texttt{G2S}}$ as an example, the hidden feature representation $\breve{\mathbf{F}}_G$ is fed into the generative module to generate the simulated satellite-view feature representation $\mathbf{L}_G$, and the normalized satellite-view representation $\breve{\mathbf{F}}_S$ is used for supervision, and vice versa. It is worth noting that both generative modules $f_{\texttt{G2S}}$ and $f_{\texttt{S2G}}$ own the same architecture but do not share weights due to the difference in input and generative content.

Encoder: Figure 4.3 illustrates the encoder–decoder architecture in detail. The encoder in the generative module is designed as a hybrid architecture that combines multi-head attention and convolutional layers. Taking $f_{\texttt{G2S}}$ as an illustration, the feature $\breve{\mathbf{F}}_G$ is encoded independently by the attention layers and the convolutional layers, resulting in producing the compact features $\dot{\mathbf{F}}_G^T$ and $\dot{\mathbf{F}}_G^C$, respectively, and these two features are concatenated along the channel to form the encoded feature $\dot{\mathbf{L}}_S$, which contains both global and local contextual information.

Decoder: Following the acquisition of $\dot{\mathbf{L}}_S$, the decoding process begins with a two-layer multi-head attention operation followed by multi-layer perceptions to generate the simulated cross-view feature $\mathbf{L}_S$. The encoder and decoder are combined via skip connections to form a Unet-like [142] architecture, which enables aggregate features at different semantic levels.

4.3.4 Generative Knowledge Supported Transformer (GKST) $f_{\texttt{GKST}}$

So far, we have acquired the inter-view representation $\breve{\mathbf{F}}_G(\breve{\mathbf{F}}_S)$ and the generative cross-view representation $\mathbf{L}_S(\mathbf{L}_G)$. To learn the final representation, $\mathbf{F}_G$ and $\mathbf{F}_S$, we designed a generative

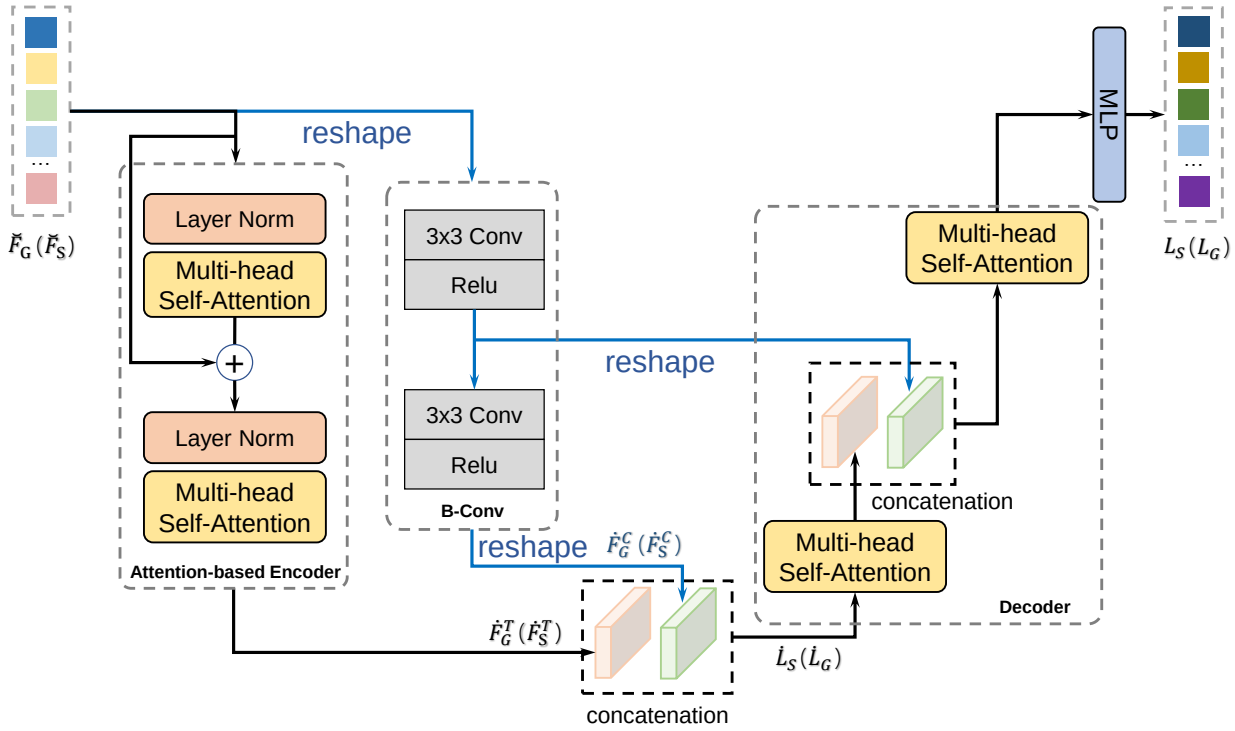


Figure 4.3: Details of the cross-view generative module. The generative module is designed as a Unet-like [142] architecture, taking advantage of transformer and CNN features to extract contextual information.

knowledge-supported transformer (GKST) to fully utilize all information. Formally, $f_{\texttt{GKST}}$ takes $\breve{\mathbf{F}}_G \in \mathbb{R}^{l\times c}$($\breve{\mathbf{F}}_S \in \mathbb{R}^{l\times c}$) and $\mathbf{L}_S \in \mathbb{R}^{l\times c}$($\mathbf{L}_G \in \mathbb{R}^{l\times c}$) as inputs and produces the final high-order representations $\mathbf{F}_G$($\mathbf{F}_S$). Taking the ground view as an illustration, we feed the inter-view representation $\breve{\mathbf{F}}_G$ and cross-view knowledge $\mathbf{L}_S$ into a multi-head cross-attention layer to learn the cross-view enhanced features. The cross-attention process is formulated as follows:

$$\begin{aligned} &Q_G^i = \breve{\mathbf{F}}_G\mathbb{W}_Q^i, K_S^i = \mathbf{L}_S\mathbb{W}_K^i, V_S^i = \mathbf{L}_S\mathbb{W}_V^i, \\ &\texttt{Head}_i = \texttt{Attention}(Q_G^i, K_S^i, V_S^i), \\ &\texttt{MH}(Q,K,V) = \texttt{Concat}(\texttt{Head}_1, ..., \texttt{Head}_n)\mathbb{W}, \end{aligned} \tag{4.6}$$

where $\mathbb{W}_Q^i$, $\mathbb{W}_K^i$, $\mathbb{W}_V^i$, and $\mathbb{W}$ are learnable parameters. The updated representations $\hat{\mathbf{F}}_G$ can be achieved by two residual connections, which are formulated as follows:

$$\begin{aligned} &\mathbf{F}_G^* = \texttt{MH}(Q,K,V) + \breve{\mathbf{F}}_G, \\ &\hat{\mathbf{F}}_G = \mathbf{F}_G^* + \texttt{LN}(\mathbf{F}_G^*). \end{aligned} \tag{4.7}$$

We can easily obtain the final satellite-view feature maps $\hat{\mathbf{F}}_S$ in a similar way.

Recurrent Learning Process: To fully mine the benefits of the cross-view knowledge, we can further formulate the learning process recurrently as follows:

$$\begin{cases} \hat{\mathbf{F}}_G^l = f_{\texttt{GKST}}(\mathbf{L}_S^{l-1}, \breve{\mathbf{F}}_G^{l-1}), \mathbf{L}_S^{l-1} = f_{\texttt{G2S}}(\breve{\mathbf{F}}_G^{l-1}), \\ \hat{\mathbf{F}}_S^l = f_{\texttt{GKST}}(\mathbf{L}_G^{l-1}, \breve{\mathbf{F}}_S^{l-1}), \mathbf{L}_G^{l-1} = f_{\texttt{S2G}}(\breve{\mathbf{F}}_S^{l-1}), \end{cases} \tag{4.8}$$

where $\breve{\mathbf{F}}_G^{l-1} = \texttt{LN}(\hat{\mathbf{F}}_G^{l-1})$, $\breve{\mathbf{F}}_S^{l-1} = \texttt{LN}(\hat{\mathbf{F}}_S^{l-1})$. Note that, at the beginning, (l = 1), $\hat{\mathbf{F}}_G^0$ and $\hat{\mathbf{F}}_S^0$ are produced by Equation (4.4), and the final representations $\mathbf{F}_G$ and $\mathbf{F}_S$ are produced by the last layer.

4.3.5 Loss Function

In order to make the final representations $\mathbf{F}_G$ and $\mathbf{F}_S$ more consistent between matching pairs but more discriminating among unmatching pairs, following [76], we employ a margin triplet loss $\mathbf{L}oss_{\texttt{Triplet}}$ for final representation supervision:

$$\mathbf{L}oss_{\texttt{Triplet}} = \texttt{log}(1 + e^{\gamma(\texttt{d}_{pos} - \texttt{d}_{neg})}), \tag{4.9}$$

where γ indicates the function hyperparameter and $\texttt{d}_{pos}$ and $\texttt{d}_{neg}$ indicate the Euclidean distance between the positive and the negative pairs, respectively. To guarantee the quality of simulated cross-view knowledge, the cross-view knowledge generation module is supervised by mean squared errors (MSE) $\mathbf{L}oss_{\texttt{Gen}}$:

$$\mathbf{L}oss_{\texttt{Gen}} = \sum_{l=1}^{L} (||\breve{\mathbf{F}}_G^{l-1} - \mathbf{L}_S^{l-1}||^2 + ||\breve{\mathbf{F}}_S^{l-1} - \mathbf{L}_G^{l-1}||^2), \tag{4.10}$$

where $\breve{\mathbf{F}}_G^{l-1}(\breve{\mathbf{F}}_S^{l-1})$ and $\mathbf{L}_S^{l-1}(\mathbf{L}_G^{l-1})$ denote inter-view representations and the generative cross-view knowledge at the l-th recurrent step, respectively. L is the total recurrent step. Finally, to learn the optimal parameters Θ for $\mathcal{F}_\Theta$, MGTL is jointly optimized through the overall learning $\mathbf{L}oss$, which is computed as:

$$\mathbf{L}oss = \mathbf{L}oss_{\texttt{Triplet}} + \lambda \mathbf{L}oss_{\texttt{Gen}}, \tag{4.11}$$

where λ is the balancing factor.

4.4 Experiment Results

4.4.1 Experimental Setting

Dataset: Following [134, 76, 25], we evaluated the performance of mutual generative transformer learning (MGTL) on two widely-used challenging benchmarks, *CVUSA* [139] and *CVACT* [140]. *CVUSA* was constructed by Workman et al. [20], containing 1.7 million training pairs collected from San Francisco. However, the relatively limited acquisition locations result in poor generalization capability of the extracted features when images from other positions are taken as input. To address this issue, the researchers reconstructed a new extensive *CVUSA* dataset [139], which contains 1.5 million geo-tagged pairs of ground-view and satellite-view images covering the continental United States, with resolutions of 1232×224 and 750×750, respectively. Ground-view images were collected using the Google Street View app and Flickr with different pre-processing methods. Specifically, the researchers randomly sampled images from the continental United States using the former but divided the entire area into a 100×100 grid and sampled up to 150 images in each cell while using the latter. Further, based on the original *CVUSA* [139], Zhai et al. [143] selected ground-view panoramas from *CVUSA* [139] and satellite-view images from Bing Maps at the same location as the matching pairs. In particular, the panoramas were wrapped to align with the satellite images using camera parameters. Finally, they released a subset of the CVUSA [139] containing 44,416 ground-satellite image pairs collected at the same location as well as 35,532 training pairs and 8884 evaluation pairs. This subset has become a widely-used benchmark because of its high resolution and simple format. To better investigate the possibility of matching geolocation in urban scenarios, Liu et al. [140] created a city-scale cross-view dataset *CVACT* [140] densely covering Canberra, Australia. Similar to *CVUSA*, ground-view panoramas were collected from the Google Street View app at zoom 2 with 1664×832 image reso-

lution, while satellite-view images were collected from the Google Maps app at zoom 20 at the same location with 1200 $\times$ 1200 resolution. In order to fully evaluate the generalization of the CVGL methods, *CVACT* [140] released *CVACT*_test containing an extra 92,802 challenging pairs for testing only. Figure 4.4 displays several ground-satellite image pairs from *CVUSA* [139] and *CVACT* [140].

(a) CVUSA (b) CVACT

Figure 4.4: Example image pairs from *CVUSA* [139] and *CVACT* [67].

Evaluation Metric: Following existing works [76, 25], recall accuracy at top K(r@K) was performed to evaluate the proposed MGTL. Staying in step with these existing methods, $K = 1, 5, 10, 1\%$ were selected.

Training Setting: During the training phase, MGTL adopted VGG16 [52] pre-trained on ImageNet [144] as the backbone. All training images were resized to 112×616 resolution augmented by random cropping, flipping, rotation, etc. We employed the Adam optimizer to optimize the whole network with the initial learning rate of 10^{-5}. We set the recurrent learning step of the generative knowledge supported transformer (GKST) to 6 and equipped 6 attention heads for each step. We set the batch size to 16 and trained the network for up to 150 epochs until complete convergence. The balancing factor λ in Equation (4.11) was carefully set to 0.05, and following [76], the regular item γ in Equation (4.9) was set to 10.0.

Reproducibility: We implemented the MGTL based on TensorFlow and trained the whole network on an NVIDIA GTX Titan X GPU with 12G CUDA memory.

4.4.2 Main Results

Baselines: Cross-view geo-localization (CVGL) has garnered significant research interest, resulting in several impressive works emerging in the field. To demonstrate the superiority of our proposed method, we selected 17 strong baselines and state-of-the-art methods in total, i.e., Workman et al. [20], Vo et al. [21], Zhai et al. [143], Cross-View Matching Network (CVM-Net) [22], Liu et al. [140], Regmi et al. [23], Spatial-Aware Feature Aggregation network (SAFA) [76], Cross-View Feature Transport technique (CVFT) [135], Dynamic Similarity Matching network (DSM) [145], Toker et al. [79], Layer-to-Layer Transformer (L2LTR) [25], Local Pattern Network (LPN) [137], Unit SAFA + Subtraction Attention Module (USAM) [146], LPN + USAM [146], pure transformer-based geo-localization (TransGeo) [24], Transformer-Guided Convolutional Neural Network (TransGCNN) [136], and LPN + Dynamic Weighted Decorrelation Regularization (DWDR) [138]. In particular, for omnidirectional comparison, we use their recommended settings for training. Our MGTL outperformed existing methods across most top K(r@K) metrics on both benchmarks, showcasing the effectiveness of our proposed cascaded attention-masking (CAMask) algorithm and cross-view interaction (CVI) tactic. In this section, we provide a detailed introduction to our experiment setup and experiment results.

Performance on *CVUSA*: The test set of *CVUSA* [139] has 8884 challenging ground-satellite image pairs. The results with 17 SOTAs presented in Table 4.2 (left) show that our approach achieves state-of-the-art performance compared to all baselines in terms of almost all top K(r@K) metrics on *CVUSA* [139]. Our approach achieves the best top 1(r@1) retrieval accuracy, as well as significant increases of 4.34% (90.16% $\rightarrow$ 94.50%) and 3.28% (91.22% $\rightarrow$ 94.50%) over

SAFA+USAM [146] and LPN+USAM [146]. Notably, although the top 1% (r@1%) retrieval accuracy is almost 100%, MGTL still achieves 0.11% growth. Our approach outperforms L2LTR [25] by 0.45% (94.05% $\rightarrow$ 94.50%) in the top 1(r@1) retrieval accuracy while haaving less computation complexity and model capacity. TransGeo [24] utilizes a three-branch vision transformer [81] with a novel attention-based masking scheme. Our results outperform it by 0.42% (94.08% $\rightarrow$ 94.50%) in the top 1(r@1) retrieval accuracy. Nevertheless, methods other than MGTL ignore the semantic consistency revealed by cross-view interaction, making mutual generative learning the more convincing method. Figure 4.5 shows the partial hard image pair retrieval result from Toker et al. [79], L2LTR [25], SAFA [76], and CVFT [135]. The similarity between the ground truth and the selected unmatched satellite images heavily interferes with other models. In contrast, this illustrates that the co-visual enhanced features learned by CAMask and CVI own finer-grained understandings of scenarios and are highly discriminative.

Performance on *CVACT*_val: The evaluation set of *CVACT* [140] contains 8884 ground–satellite image pairs, consistent with *CVUSA* [139]. Table 4.2 (middle) presents the results with 13 STOAs on *CVACT*_val [140]. Our approach achieves the best performance across all top K(r@K) metrics (r@1, r@5, r@10, r@1%) on *CVACT*_val, i.e., 85.42%, 94.64%, 96.11%, and 98.51%, respectively. MGTL achieves significant improvements over LPN + USAM [146], SAFA + USAM [146], and LPN + DWDR [138], increasing the top 1(r@1) retrieval accuracy by 3.40% (82.02% $\rightarrow$ 85.42%), 3.02% (82.40% $\rightarrow$ 85.42%), and 1.69% (83.73% $\rightarrow$ 85.42%), respectively. In addition, our results are significantly better across all metrics compared to classical Siamese-like VGG-based convolutional methods, e.g., SAFA [76], CVFT [135], and DSM [145]. The experimental results mentioned above demonstrate the effectiveness of our CAMask and CVI introduced by MGTL. In comparison with traditional transformer-based methods TransGeo [24] and L2LTR [25], MGTL increased the top 1(r@1) retrieval accuracy significantly by 0.47% (84.95%

$\rightarrow$ 85.42%) and 0.53% (84.89% $\rightarrow$ 85.42%), respectively. This strongly proves the superiority of our generative knowledge-supported transformer framework. Toker et al. [79] proposed a GAN-based method to synthesize realistic ground-view images from satellite images, which explores the benefits of generative learning for cross-view matching. MGTL outperformed it by 2.14% in the top 1(r@1) retrieval accuracy, showing that our mutual generative learning strategy is more effective and has extreme generalizability in urban scenarios.

Figure 4.5: Comparison results of some hard pairs.

Table 4.2: **Quantitative results** on the *CVUSA* [139] and *CVACT* [140] dataset.

Model	CVUSA				CVACT_val				CVACT_test			
	r@1	**r@5**	**r@10**	**r@1%**	**r@1**	**r@5**	**r@10**	**r@1%**	**r@1**	**r@5**	**r@10**	**r@1%**
2015 Workman et al. [20]	-	-	-	34.30	-	-	-	-	-	-	-	-
2016 Vo et al. [21]	-	-	-	63.70	-	-	-	-	-	-	-	-
2017 Zhai et al. [143]	-	-	-	43.20	-	-	-	-	-	-	-	-
2018 CVM-Net [22]	22.47	49.98	63.18	93.62	20.15	45.00	56.87	87.57	5.41	14.79	25.63	54.53
2019 Liu et al. [140]	40.79	66.82	76.36	96.12	46.96	68.28	75.48	92.04	19.9	34.82	41.23	63.79
2019 Regmi et al. [23]	48.75	-	81.27	95.98	-	-	-	-	-	-	-	-
2019 SAFA [76]	89.84	96.93	98.14	99.64	81.03	92.80	94.84	98.17	55.50	79.94	85.08	94.49
2020 CVFT [135]	61.43	84.69	90.49	99.02	61.05	81.33	86.52	95.93	34.39	58.83	66.78	95.99
2020 DSM [145]	91.96	97.50	98.54	99.67	82.49	92.44	93.99	97.32	35.55	60.17	67.95	86.71
2021 Toker et al. [79]	92.56	97.55	98.33	99.57	83.28	93.57	95.42	98.22	61.29	85.13	89.14	98.32
2021 L2LTR [25]	94.05	98.27	98.99	99.67	84.89	94.59	95.96	98.37	60.72	85.85	89.88	96.12
2021 LPN [137]	93.78	98.50	99.03	99.72	82.87	92.26	94.09	97.77	-	-	-	-
2022 SAFA+USAM [146]	90.16	-	-	99.67	82.40	-	-	98.00	56.16	-	-	95.22
2022 LPN+USAM [146]	91.22	-	-	99.67	82.02	-	-	98.18	37.71	-	-	87.04
2022 TransGeo [24]	94.08	98.36	99.04	99.77	84.95	94.14	95.78	98.37	-	-	-	-
2022 TransGCNN [136]	94.15	98.21	98.94	99.79	84.92	94.46	95.88	98.36	-	-	-	-
2022 LPN+DWDR [138]	94.33	**98.54**	99.09	**99.80**	83.73	92.78	94.53	97.78	-	-	-	-
Ours	**94.50**	98.41	**99.20**	99.78	**85.42**	**94.64**	**96.11**	**98.51**	**61.55**	**86.61**	**90.74**	**98.46** [1]

[1] Results are cited directly, and the best results are highlighted in bold.

Performance on *CVACT*_test: *CVACT*_test is massive and extremely challenging, consisting of 92,802 ground–satellite image pairs in urban scenarios for testing only. For the challenging *CVACT*_test, we compared our approach with 9 SOTAs. As shown in Table 4.2 (Right), our MGTL sets new retrieval accuracy records across all metrics compared to existing SOTAs. MGTL increases the top 1(r@1) retrieval accuracy significantly by 0.83% (60.72% $\rightarrow$ 61.55%) and 5.39% (56.16% $\rightarrow$ 61.55%) compared to L2LTR [25] and SAFA+USAM [146], respectively. Furthermore, our results not only outperform others in top 1(r@1) retrieval accuracy, but also gain a remarkable increase of 1.48% (85.13% $\rightarrow$ 86.61%) in the top 5(r@5) retrieval accuracy over Toker et al. [79] and 2.34% (96.12% $\rightarrow$ 98.46%) in the top 1%(r@1%) recall accuracy over L2LTR [25]. These superior experiment results showcase that MGTL is capable of capturing high-order under-

standings of cross-view scenarios essential for CVGL in unfamiliar environments in the absence of prior knowledge.

4.4.3 Ablation Study

As the mutual generative transformer learning (MGTL) incorporates the cascaded attention-masking (CAMask) algorithm and cross-view interaction (CVI) tactic into the cross-view geo-localization (CVGL) task, we conduct substantial ablation studies to carefully scrutinize how each component affects the learning ability of the model.

Effectiveness of CAMask: To qualitatively study the effectiveness of our proposed CAMask algorithm, we inspect the performance of the VGG16 backbone [52] with fully-connected layers removed. As shown in Table 4.3, all metrics degrade significantly when removing CAMask. The top 1(r@1) retrieval accuracy suffers a drastic decrease of 10.18%, from 90.12% to 79.94%, supporting the notion that co-visual information explicitly learned by CAMask is extremely critical for CVGL. In addition, we removed the CAMask from the fully equipped model. Observing the last two lines in Table 4.3, the top 1(r@1) retrieval accuracy still suffers a heavy decrease of 4.06%, from 94.50% to 90.44%, suggesting once again that the above notion is strongly supported. To explore the necessity of SCE and SA, Table 4.4 displays the comparison results while removing each of them, respectively. Replacing the SCE with fully convolutional blocks, the top 1(r@1) retrieval accuracy decreased by 2.21% (94.50% $\rightarrow$ 92.29%) and 3.05% (85.42% $\rightarrow$ 82.37%) on *CVUSA* and *CVACT_val*, respectively. Similarly, when we replaced the SA with global average pooling (GAP), the top 1(r@1) retrieval accuracy degraded by 0.88% (94.50% $\rightarrow$ 93.62%) and 1.09% (85.42% $\rightarrow$ 84.33%) on both datasets. MGTL suffers a decrease in precision for varying content, suggesting co-visual enhanced feature representations learned by CAMask lead to more reliable results. To show the superiority of CAMask qualitatively, we meticulously visualized the cascaded attention

masks in Figure 4.6 to support our claim. The first row indicates the generative attention masks as well as corresponding attention scores. To showcase the co-visual regions intuitively, we binarize the attention masks, as shown in the second row. Subsequently, the original images are cropped with the guidance of binary masks, as shown in the third row. Observing the third row, only the co-visual regions (e.g., road, building) remain, and redundant non-co-visual regions (e.g., 'sky' in ground imagery but absent in satellite imagery) useless for matching were masked. CAMask eradicates these disturbances in a simple but effective manner. Finally, to showcase the correctness of the co-visual relationships, the same regions captured across views are marked with rectangles of the same color, as shown in the fourth row.

Table 4.3: **Ablation study** of the proposed cascaded attention-masking (CAMask) algorithm.

Candidate			Complexity		CVUSA				CVACT_val			
VGG16	**CAMask**	**CVI**	**GFLOPs** ↓	**Param.** ↓	**r@1**	**r@5**	**r@10**	**r@1%**	**r@1**	**r@5**	**r@10**	**r@1%**
✔			28.22	29.42 M	79.94	93.66	96.25	99.31	70.67	87.73	91.13	95.78
✔	✔		59.02	91.18 M	85.15	95.13	96.89	99.43	76.32	89.64	92.26	96.21
✔		✔	28.90	137.21 M	90.44	96.83	97.41	99.45	81.25	92.12	94.38	97.69
✔	✔	✔	59.71	171.97 M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.11**	**98.51** [1]

[1] The best results are highlighted in bold. ↓ means lower is better.

Table 4.4: **Ablation study** of the proposed spatial attention (SA) and spatial context enhancement (SCE) in cascaded attention masking (CAMask).

Candidate			Complexity		CVUSA				CVACT_val			
VGG16 + CVI	**w/SA**	**w/SCE**	**GFLOPs** ↓	**Param.** ↓	**r@1**	**r@5**	**r@10**	**r@1%**	**r@1**	**r@5**	**r@10**	**r@1%**
✔			28.90	137.21 M	90.44	96.83	97.41	99.45	81.25	92.12	94.38	97.69
✔	✔		28.90	137.21 M	92.29	97.65	98.67	99.72	82.37	93.35	95.17	98.23
✔		✔	59.71	171.97 M	93.62	98.41	99.07	99.73	84.33	94.31	95.67	98.43
✔	✔	✔	59.71	171.97 M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.11**	**98.51** [1]

[1] 'w/' meansthe proposed MGTL is equipped with SA or SCE, respectively. The best results are highlighted in bold. ↓ means lower is better.

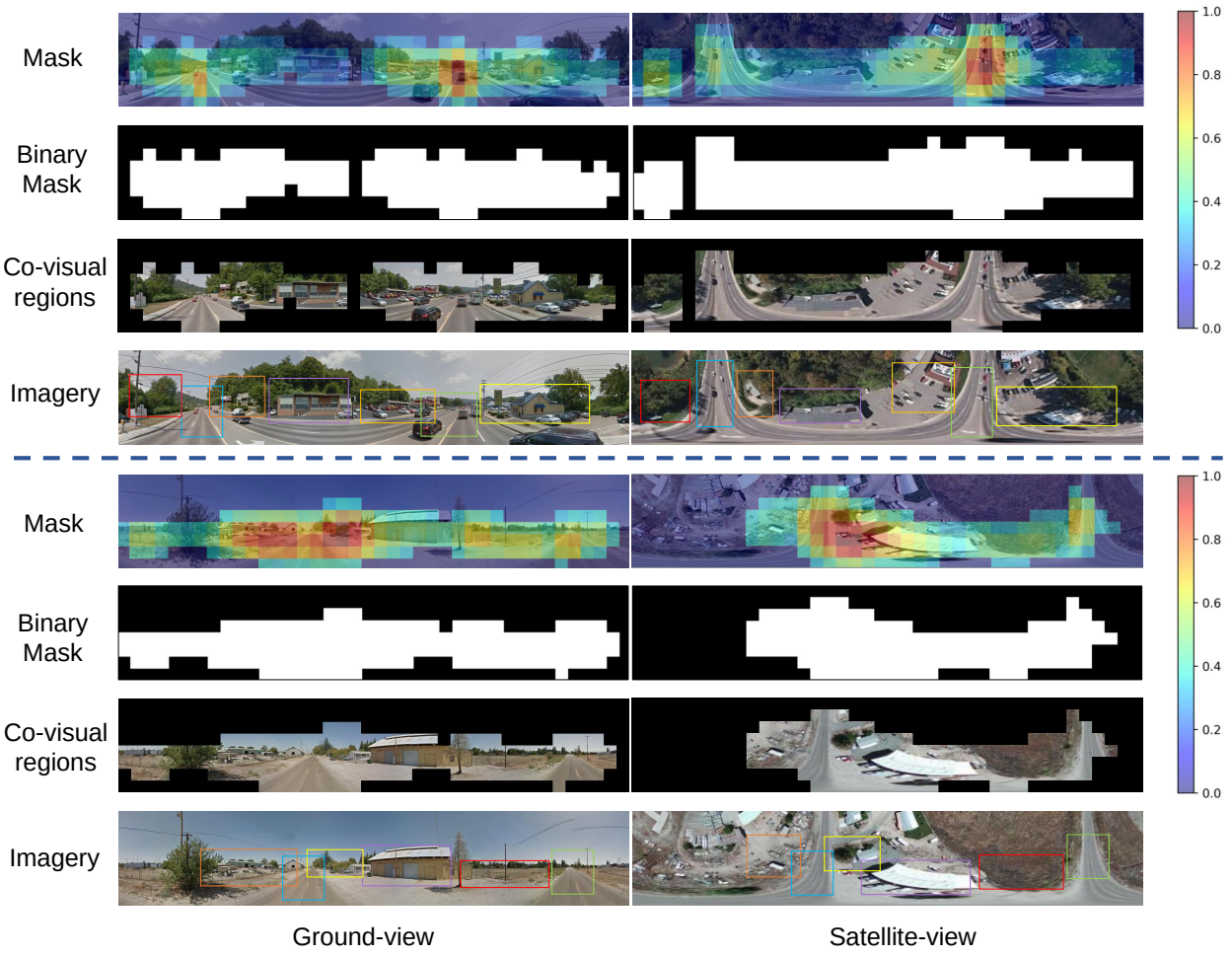


Figure 4.6: Visualization results of cascaded attention masks.

Effectiveness of CVI: We introduce the CVI tactic to implicitly explore co-visual information and empirically investigate the superiority of CVI in MGTL. Table 4.5 shows that when incorporating transformer [80] blocks, the top 1(r@1) retrieval accuracy increases by 6.27% (85.15% $\rightarrow$ 91.42%). However, this is still a sub-optimal performance compared to existing SOTAs. The designation 'w/o' in Table 4.5 refers to the pure transformer without CVI; all metrics suffer drastic degradations, and the top 1(r@1) retrieval accuracy decreases by 3.08% (94.50% $\rightarrow$ 91.42%), showcasing that CVI further boosts the pure transformer learning capability and enhances the sim-

ilarity of feature representations between matching pairs.

Table 4.5: **Ablation study** of cross-view interaction (CVI).

Candidate			Complexity		CVUSA				CVACT_val			
VGG16 + CA-Mask	**w/o**	**w/CVI**	**GFLOPs ↓**	**Param. ↓**	**r@1**	**r@5**	**r@10**	**r@1%**	**r@1**	**r@5**	**r@10**	**r@1%**
✔			59.02	91.98 M	85.15	95.13	96.89	99.43	76.32	89.64	92.26	96.21
✔	✔		59.32	113.48 M	91.42	96.21	98.04	99.62	81.99	93.16	95.04	98.23
✔		✔	59.71	171.97 M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.11**	**98.51** [1]

[1] 'w/' and 'w/o' means the transformer learning is or is not equipped with CVI, respectively. The best results are highlighted in bold. ↓ means lower is better.

[1] 'w/' and 'w/o' means the transformer learning is or is not equipped with CVI, respectively. The best results are highlighted in bold. ↓ means lower is better.

Composition of Generative Module: To determine the most effective interaction mode, we empirically explored the variational autoencoder (VAE) [147], CNN-based Unet [142], and pure transformer [80] block. The results shown in Table 4.6 suggest the superiority of our hybrid generative module. Specifically, VAE [147] is considered to be one of the classical generative models, so we utilized 2 two-layer fully-connected networks as encoder and decoder, respectively. Following [142], we exploited 2 two-layer convolutional blocks as both encoder and decoder to form a simplified Unet-like [142] architecture. However, limited by the locality assumptions, there is significant deterioration in precision. Similarly, we reconstructed a simplified transformer-based generative module referring to [85], whose encoder and decoder both consisted of two transformer layers. In this case, the performance across all metrics was worse than ours, but the complexity is higher. These results demonstrate that our generative module is suitable for plugging into a transformer to generate simulated cross-view knowledge.

Study of Recurrent Learning Steps: To determine the best recurrent learning step that balances quality and complexity, we report the results trained with different recurrent steps in Ta-

Table 4.6: **Detailed ablation study** of the composition of the generative module.

Method	Complexity		CVUSA				CVACT_val			
	GFLOPs ↓	Param. ↓	r@1	r@5	r@10	r@1%	r@1	r@5	r@10	r@1%
VAE [147]	59.89	141.84M	94.11	98.27	99.03	99.71	85.32	94.42	96.04	98.41
Unet [142]	59.53	134.74M	92.04	97.91	98.80	99.67	82.31	93.08	95.09	98.30
Transformer [80]	61.20	276.49M	94.37	98.30	99.08	99.74	85.35	94.45	96.03	98.44
Ours	59.71	171.97M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.11**	**98.51** [1]

[1] The best results are highlighted in bold. ↓ means lower is better.

ble 4.7. This demonstrates that increasing the recurrent learning step leads to improved performance, which proves that recurrent learning can fully mine the representational ability of generative cross-view knowledge. As the learning step increases from 3 to 6, the top 1(r@1) retrieval accuracy improves significantly. However, the gains are negligible and even degrade as the recurrent step rises to 9. We note that the quantity and quality of the new generative knowledge becomes more difficult as the recurrent step rises. Therefore, the recurrent learning step is set to 6 to achieve a trade-off between accuracy and time cost.

Table 4.7: **Detailed ablation study** of different parameter settings.

CVI	CVUSA				CVACT_val			
	r@1	r@5	r@10	r@1%	r@1	r@5	r@10	r@1%
L = 1	87.07	96.48	97.72	99.65	77.92	90.85	93.27	97.20
L = 3	89.67	97.15	98.26	99.70	80.40	91.63	94.18	98.17
L = 6	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	96.11	**98.51**
L = 9	94.25	98.39	99.18	99.76	85.20	94.61	**96.12**	98.49 [1]

[1] The best results are highlighted in bold.

4.4.4 Supplementary Experiment

To further explore the rationality of each module, i.e., multi-scale feature aggregation (MSFA), spatial context enhancement (SCE), and spatial attention (SA) in the cascaded attention-masking (CAMask) algorithm, we conduct intensive experiments with different settings on both *CVUSA* [139]

and *CVACT_val* [140]. All results are reported in Table 4.8.

Exploration of MSFA: We redesign a parallel multi-branch convolutional module named SCE. Unlike existing works, we introduce a novel feature aggregation tactic named MSFA to further integrate branches with different scales. To prove the necessity of MSFA, we replace the MSFA with direct addition operations and convolutional layers, respectively. We observe that all the metrics decreased and that the top 1(r@1) retrieval accuracy decreased drastically by more than 1%. To further illustrate the advancements of our proposed MSFA, we select five typical feature aggregation tactics, including squeeze-and-excitation networks (SENet) [148], convolutional block attention module (Cbam) [119], self-calibrated convolution (SCNet) [149], Non_local [150], and selective kernel convolution (SKC) [151], and then plug them into SCE. As shown in Table 4.8 (top), SCE with MSFA achieves the best retrieval accuracy, with a parameter increase of less than 10 M.

Exploration of SCE: To illustrate the superiority of our proposed SCE, we select two typical multi-branch convolutional modules, i.e., Texture-Enhanced Module (TEM) [141] and Receptive Field Block (RFB) [152]. TEM [141] aims to capture fine-grained texture and context features, and it was initially employed in the concealed object detection (COD) task. Inspired by the human visual system, RFB [152] introduced a multi-branch dilated convolution to enhance the feature extraction ability of the network. Building on this purpose, we introduce SCE with MSFA. As shown in Table 4.8 (middle), SCE achieves the optimal performance with fewer parameters, suggesting the SCE equipped with an attention-based feature aggregation tactic is more suitable for the CVGL task.

Exploration of SA: Spatial attention is thought to adaptively learn discriminative regions in the feature map to generate the spatial masks. We employ SA to connect cascaded structures and used spatial masks generated from cross-level semantic information to compensate for the loss

Table 4.8: **Detailed ablation study** of the rationality of MSFA and MSCM.

Module	Complexity		CVUSA				CVACT_val			
	GFLOPs ↓	Param. ↓	r@1	r@5	r@10	r@1%	r@1	r@5	r@10	r@1%
Why MSFA in SCE?—Comparison with other aggregation methods.										
add	54.62	161.30 M	93.11	98.22	98.93	99.75	83.55	93.66	95.56	98.26
concat + conv	55.75	163.71 M	92.62	97.96	98.85	99.72	83.07	93.84	95.57	98.21
SENet [148]	54.63	161.95 M	93.48	98.31	98.99	99.74	85.01	94.40	96.02	98.48
Cbam [119]	54.63	161.96 M	93.61	98.39	99.02	99.73	84.89	94.27	96.03	98.43
SCNet [149]	56.68	166.12 M	93.82	98.38	99.08	99.74	84.67	94.28	95.97	98.35
Non_Local [150]	65.36	163.72 M	92.84	98.01	98.86	99.67	83.48	93.99	95.68	98.29
SKC [151]	73.84	201.91 M	93.92	98.36	99.03	99.78	84.95	94.36	95.81	98.44
MSFA	59.71	171.97 M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.03**	**98.51**
Why SCE in CAMask?—Comparison with other parallel multi-branch convolutional modules.										
Conv	76.38	208.51 M	92.97	98.16	98.87	99.72	83.51	94.02	95.78	98.37
TEM [141]	67.05	187.31 M	94.21	98.37	99.05	99.74	85.10	94.56	95.99	98.43
RFB [152]	67.61	188.47 M	94.13	98.33	99.07	99.72	85.08	94.48	96.01	98.40
SCE	59.71	171.97 M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.03**	**98.51**
Why SA in CAMask?—Comparison with other attention mask generation methods.										
GAP	59.71	171.97 M	93.62	98.41	99.07	99.73	84.33	94.31	95.67	98.43
GMP	59.71	171.97 M	93.43	98.33	99.05	99.75	84.20	94.21	95.88	98.45
SA	59.71	171.97 M	**94.50**	**98.41**	**99.20**	**99.78**	**85.42**	**94.64**	**96.03**	**98.51** [1]

[1] The best results are highlighted in bold. ↓ means lower is better.

of spatial information due to reduced spatial resolutions via multiplying with high-level semantic features. To study the effectiveness of SA, we replace SA with global average/max pooling (GAP/GMP) layers and observe an overall decrease across all metrics in which the top 1(r@1) retrieval accuracy suffers drastic decreases by 0.88% (94.50% → 93.62%) and 1.07% (94.50% → 93.43%), showcasing the necessity and effectiveness of the SA mechanism in CAMask.

4.5 Discussion

As described in Section 4.4, mutual generative transformer learning (MGTL) outperforms recent outstanding cross-view geo-localization (CVGL) works significantly across almost all metrics on widely-used benchmarks *CVUSA* [139] and *CVACT* [140], owing to our cascaded attention-masking (CAMask) algorithm and cross-view interaction (CVI) tactic. CAMask is integrated into

feature extractor VGG16 [52] to encourage co-visual regions for reasoning during generative transformer learning, which eradicates the interference of viewpoint-sensitive regions. CVI is implemented by a cross-view generative module and by generative knowledge-supported transformer learning. Cross-view mutual generative learning aims to simulate feature representations across views, subsequently exploiting generative knowledge to mine the semantic consistency through the attention mechanism in recurrent transformer learning. Our findings perform excellently in cross-view image matching essential for CVGL. In addition, our MGTL enhances the generalizability of CVGL, driving vision-based geo-localization solutions applicable in autonomous driving fields without GPS support.

By exploiting the inter-view semantic consistency, mutual learning can alleviate ambiguity in cross-view matching. The study of view matching in UAV localization has also been conducted in a similar area for the purpose of completing UAV geographic localization. A benchmark called University-1652 [153] aims to establish correspondence between a UAV view and a satellite view. In our future work, we will explore how mutual learning techniques can play a role in this similar field, including: (1) The slight difference in view perspective between the satellite view and UAV view makes the cross-view semantic consistency easier to obtain, allowing mutual learning to go further in enhancing semantics; and (2) Shared parameter learning, which can make the network more efficient, should be explored in the context of mutual learning.

4.6 Concluding Remarks

In this study, we proposed a novel mutual generative transformer learning network, denoted as MGTL, for addressing the cross-view geo-localization problem. Existing methods commonly rely on a CNN-based Siamese-like backbone to extract high-order feature representations and treat each region equally. Viewpoint-sensitive regions with drastic appearance differences, however,

hinder image matching significantly. Using a cascaded attention-masking algorithm, we introduced a spatial context enhancement module and a spatial attention module in the VGG16 to capture co-visual information. In terms of semantic consistency learning, it is rarely examined in recent works, but incorporating consistency constraints by cross-view interaction during the recurrent learning process will benefit similarity computing. To facilitate high-order information mining within each view, we constructed cross-view generative modules and injected their generative cross-view knowledge into a transformer-based framework. Extensive qualitative and quantitative experiments demonstrated that mutual generative transformer learning significantly alleviated the impact of spatial information mismatch caused by drastic viewpoint changes. By examining cross-view interactions, we highlighted the potential of this perspective to advance automobile geo-location identification research in GPS-denied conditions.

CHAPTER 5
STEERING ANGLE CORRECTION LEARNING FOR VISUAL ODOMETRY

5.1 Background

The monocular visual odometry (VO) framework is crucial for robotics localization and autonomous driving, as it can estimate camera motion trajectory using only a monocular camera, without relying on GPS or other localization devices. This framework focuses on estimating the system's location by analyzing consecutive frames from the camera, which makes it a local visual localization task. Although VO has made significant progress in recent years and has gained attention from researchers, it remains a challenging task due to complex perceptual scenarios and uncorrected strategies. Some recent advancements in this area include [32, 33, 34, 35].

Prior to the popularity of deep learning, conventional manual VO solutions had come a long way and formed a relatively stable methodological foundation of feature extraction, matching, motion estimation, bundle adjustment optimization, *etc*. During this period, SIFT [154], SURF [155], ORB [156], BRISK [50], RANSAC[157], and ORB-SLAM [48] have been developed and received considerable attention, while also being applied to various computer vision tasks [158, 159, 160, 161]. However, traditional methods exhibit certain limitations: **i)** they require practitioners to possess a high degree of expertise and entail a substantial amount of manual labor, such that any error at each step could negatively impact trajectory accuracy. **ii)** traditional feature extraction methods encode the content of the neighborhood of salient feature points, thereby exhibiting limited discriminability in regions with low texture, which impedes the extraction of higher-order contextual features.

Deep learning-based approaches have revolutionized VO by enabling end-to-end learning, pro-

viding significant advantages in the extraction of contextual features with enhanced representational efficacy, and have demonstrated superior performance over traditional techniques. Wang *et al.* [36] proposed the early pioneer in the end-to-end VO framework field based on recurrent convolutional neural networks. The framework leverages the power of convolutional neural networks to extract feature representations from concatenated frames and encodes temporal information of short input frame sequences using long short-term memory (LSTM) networks [162]. Following this framework [36], numerous variants have been proposed, including but not limited to ESP-VO [113], CL-VO [163], DAVO [120].

Although deep learning methods have shown promising results in VO, there are still several limitations that need to be addressed. First, these methods do not incorporate error correction during the learning process, leading to the propagation of estimation errors and persistent trajectory deviations. Second, the limitations of steering angle modeling are often overlooked, resulting in sudden changes in trajectory during turning maneuvers. As shown in Figure 1.4a and Figure 1.4b, when the estimated steering angle is at the boundary of the model, the absolute trajectory is prone to significant discrepancies. Third, imbalanced data distribution in training datasets can result in poor performance when generalized to real-world scenarios. Straight driving dominates the datasets, leading to insufficient training on turning maneuvers and other challenging scenarios. As shown in Table 1.1, straight driving often dominates the datasets, leading to insufficient training on turning maneuvers and other challenging scenarios. Finally, the existing techniques do not fully exploit both temporal and contextual information, limiting their ability to accurately reconstruct the environment and resulting in sub-optimal results.

5.2 Method

To address these limitations, we introduce a new VO framework called the Steering Angle Correction network (SACNet), which incorporates the steering angle as a weighted constraint during learning, utilizes cross-frame information to reduce steering angle discontinuities, and integrates LSTM and attention mechanisms to extract reliable contextual features. We evaluate the proposed SACNet using the KITTI VO benchmark [39], a challenging dataset, and compare it against strong baselines and state-of-the-art methods. The experimental results demonstrate the effectiveness of SACNet in retrieving steering angle guidance information for visual odometry.

5.2.1 Problem Formulation

The designated image sequence $\mathbf{I} = \{I_t\}_{t=1}^{N}$, where t indicates recording time and N is the sequence length, can be divided into K video clips $\mathbf{V} = \{I_t\}_{t=1}^{k} \in \mathbb{R}^{k\times h\times w\times 3}$ each consisting of *k* frames in chronological order. Let the VO model denote as the function $\mathcal{F}_\Theta$ parameterized by weights Θ, which takes a video clip $\mathbf{V}$ as input and produces the 6-DoF pose $r_{i,i+1} = (e_x, e_y, e_z, t_x, t_y, t_z)$ that represents ego-motion between every two adjacent frames I_t and I_{t+1} in $\mathbf{V}$ to form relative pose estimation sequence $\mathbf{R} = \{r_{i,i+1}\}_{i=1}^{k-1}$. Referencing the first frame to recover the absolute pose of every frame incrementally to reconstruct the complete motion trajectory. Our goal is to learn the optimal Θ from K labelled training video clips $\{(\mathbf{V}^j, \mathbf{G}^j)\}_{i=1}^{K}$ with ground-truth $\mathbf{G}^j = \{g_{i,i+1}\}_{i=1}^{k-1}$ to estimate $r_{i,i+1}$ precisely.

5.2.2 Overview

Optical flow has been extensively investigated by numerous learning-based VO works and has been demonstrated to be effective [36, 37, 121]. For an input video clip $\mathbf{V}$, we concatenate every two adjacent frames in $\mathbf{V}$ as input $X_F \in \mathbb{R}^{(k-1)\times h\times w\times 6}$ for FlowNet [118] to extract inter-frame

motion information. Next, we involve the PoseNet [164] and concatenate all k frames in $\mathbf{V}$ as input $X_P \in \mathbb{R}^{h\times w\times 3k}$ to learn long-term context knowledge.

Here we retain the initial 9 convolutional layers in FlowNet [118] and partition them into 5 stages based on different spatial resolutions, and extract the feature maps from the final three stages, which can be described as:

$$\{X^s\}_{s=4}^{6} = \mathbf{E}_{\mathcal{F}}(X_F) \tag{5.1}$$

where $X^{s=4} \in \mathbb{R}^{(k-1)\times h1\times w1\times 512}$, $X^{s=5} \in \mathbb{R}^{(k-1)\times h2\times w2\times 512}$ and $X^{s=6} \in \mathbb{R}^{(k-1)\times h3\times w3\times 1024}$ indicate the feature maps processed by stage s in FlowNet [118].

As mentioned above, we incorporate PoseNet [164] to address the issue of scale estimation inaccuracy. Specifically, we retain the first 7 convolutional layers and modify them slightly to satisfy FlowNet [118], which can be described as:

$$\{Y^s\}_{s=4}^{6} = \mathbf{E}_{\mathcal{P}}(X_P) \tag{5.2}$$

where $Y^{s=4} \in \mathbb{R}^{h1\times w1\times 512}$, $Y^{s=5} \in \mathbb{R}^{h2\times w2\times 512}$ and $Y^{s=6} \in \mathbb{R}^{h3\times w3\times 1024}$ indicate the feature maps processed by stage s in PoseNet [164].

We propose the global and local feature aggregation module (GLA) to augment short-term features with long-term context knowledge learned by PoseNet [164]. To address the spatial information loss due to resolution reduction, we adopt a hierarchical structure comprising multiple GLAs to preserve multi-level contextual information crucial for scale estimation. Subsequently, we introduce three sub-branches, which mainly comprise B-Conv and fully connected layers (FCNs), to accomplish the tasks of pose estimation, triple-frame hybrid constraint learning, and steering angle-weighted learning, respectively. The overview of the proposed SACNet is illustrated in Fig-

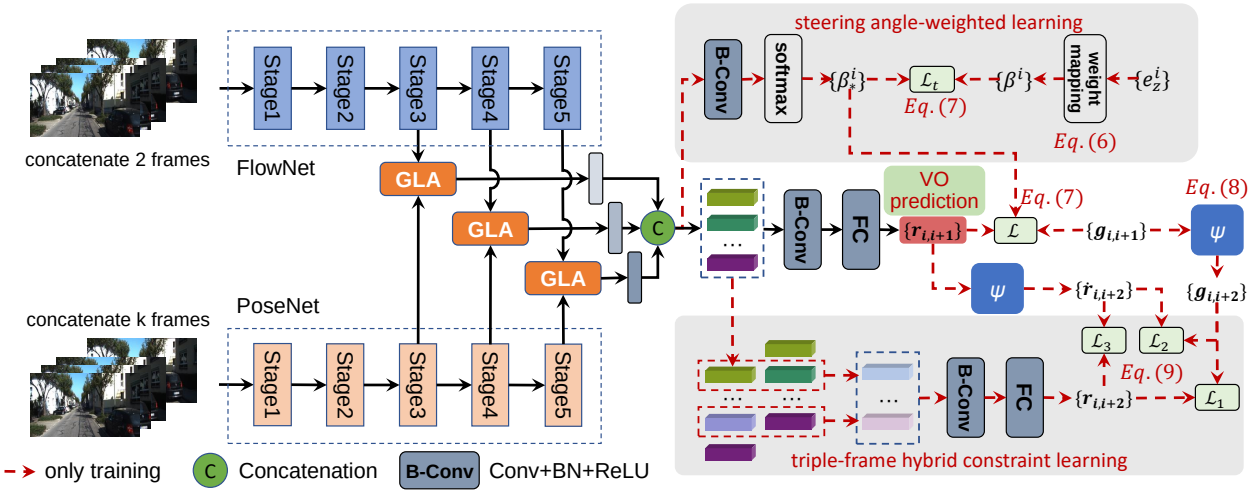


Figure 5.1: The overview of the proposed SACNet. $\{r_{i,i+1}\}$ indicates the estimated pose.

ure 5.1.

5.2.3 Details

Global and local feature aggregation: As shown in Figure 5.2, to aggregate X^s and Y^s, we design GLA consisting of two-layer LSTM and multi-head attention [81] mechanism. Taking s=4 for instance, applying full convolution layers followed by flattening to serialize $X^{s=4}$ and $Y^{s=4}$. Considering that $Y^{s=4}$ lacks the time dimension, $\mathbf{R}_{\times l}$ represents "repeat $\times l$ times" operation along with time dimension. Details are as follows:

$$\begin{aligned} \widehat{X}^{s=4} &= Conv(X^{s=4}), \\ \widehat{Y}^{s=4} &= \mathbf{R}_{\times l}\{Conv(Y^{s=4})\} \end{aligned} \tag{5.3}$$

where $\widehat{X}^{s=4} = \{\widehat{X}_t^{s=4}\}_{t=1}^{k-1}$, $\widehat{Y}^{s=4} \in \mathbb{R}^{l \times C}$ denote the serialized features and $l = k - 1$.

Aiming to incorporate temporal constraints to reduce pose estimation uncertainty, we introduce a two-layer LSTM to model the temporal dependencies of $\widehat{X}_t^{s=4} \in \mathbb{R}^C$ at each time step t. The

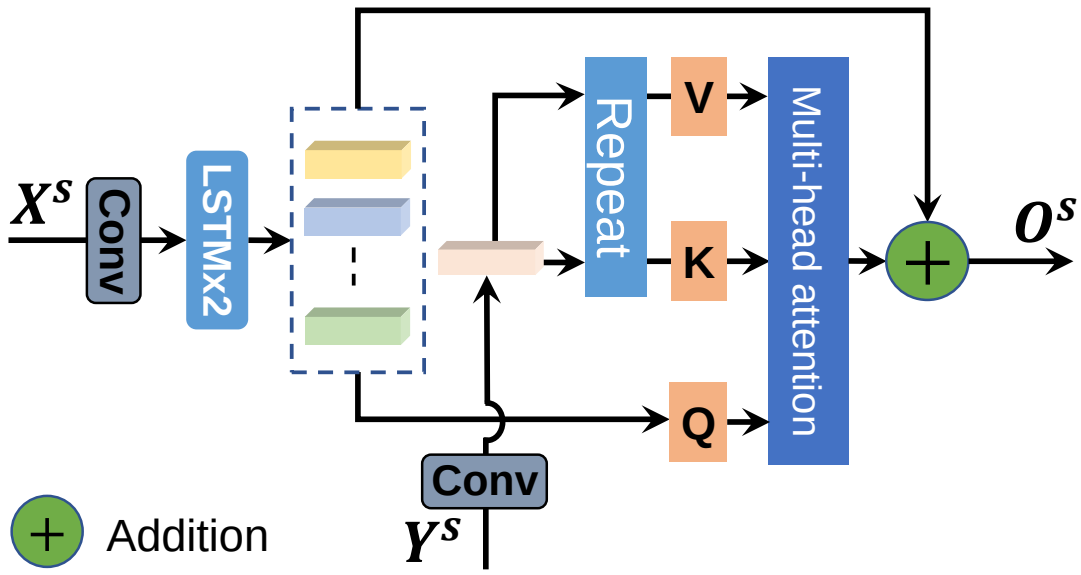


Figure 5.2: The details of GLA.

recurrent process can be defined as follows, h_t and h_{t-1} are the hidden states at current and the last time step:

$$\widetilde{X}_t^{s=4}, h_t = \textit{LSTMs}(\widehat{X}_t^{s=4}, h_{t-1}) \tag{5.4}$$

We employ multi-head attention mechanism to compute the correlation coefficients between local and global context features. Specifically, we compute the query, key, and value taking $\widetilde{X}^{s=4} = \{\widetilde{X}_t^{s=4}\}_{t=1}^{k-1}$ and $\widehat{Y}^{s=4}$ as inputs. Especially, we only keep the diagonal elements of the attention matrix and set the remaining to 0, considering the fact that temporal dimension of $\widehat{Y}^{s=4}$ is consistent. Details are as follows:

$$\begin{aligned} Q_i &= \widetilde{X}^{s=4}\mathbb{W}_Q^i, K_i = \widehat{Y}^{s=4}\mathbb{W}_K^i, V_i = \widehat{Y}^{s=4}\mathbb{W}_V^i \\ H_i^{s=4} &= (\mathbf{Softmax}(Q_i K_i^T) \circ I)V_i \\ O^{s=4} &= \widetilde{X}^{s=4} + Concat(H_1^{s=4}, \cdots, H_n^{s=4})\mathbb{W} \end{aligned} \tag{5.5}$$

where $\mathbb{W}_Q, \mathbb{W}_K, \mathbb{W}_V$ and $\mathbb{W}$ are learnable parameters. "$\circ$" indicates Hadamard product and I is identity matrix. $H_i^{s=4} \in \mathbb{R}^{l \times C}$ indicates the i-th attention head output and $O^{s=4} \in \mathbb{R}^{l \times C}$ denotes the fusion output at stage 4. In a similar manner, we can easily acquire the $O^{s=5} \in \mathbb{R}^{l \times C}$, $O^{s=6} \in \mathbb{R}^{l \times 2C}$ at stage 5, 6 and concatenate them along with channel dimension to obtain $O \in \mathbb{R}^{l \times 4C}$.

Steering angle-weighted loss function: As shown in Table 1.1, turning examples constitute significantly less than straight in training trajectories. The issue of dataset imbalance hinders the learning capacity of the model and causes inaccurate estimation during turning. To mitigate this problem, we introduce a steering angle prediction branch which takes O as input and predicts $A \in \mathbb{R}^{l \times 2}$, whose two columns represents the weights of turning and straight between every two adjacent frames, respectively. Specifically, the branch consists of three-layer FCNs. We employ the e_z (z-axis) in euler angles to generate $E = \{e_z^i\}_{i=1}^{k-1}$. The weighted mapping function Φ as described in Eq. (5.6) maps E to steering angle weights ground truth $B = \{\beta^i\}_{i=1}^{k-1}$.

$$B = (\frac{1}{1 + e^{-\mu E^2}})^{\lambda} \tag{5.6}$$

where λ, μ are parameters to adjust the weighted mapping function. To ensure the weight coefficients distribute in $[0.25, 1]$, λ is set to 2.

Furthermore, this branch is optimized using cross entropy loss $\mathcal{L}_t$. Finally, we modify origin mean square errors (MSE) by multiplying steering angle weights to calculate steering angle-weighted MSE loss $\mathcal{L}$. Details are as follows:

$$\begin{aligned} \mathcal{L}_t &= -\sum_{i=1}^{k-1} \beta_*^i \log \beta^i \\ \mathcal{L} &= \sum_{i=1}^{k-1} \beta_*^i ||r_{i,i+1} - g_{i,i+1}||^2 \end{aligned} \tag{5.7}$$

where $\{\beta_*^i\}_{i=1}^{k-1}$ is the first column of A that representing steering angle weights.

Triple-frame hybrid constraint learning: To impose self-constraint among multiple consecutive frames and mitigate large deviations at the boundary of $-\pi$ and π, a triple-frame hybrid constraint loss consisting of $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_3$ is employed. For a given input video clip V, the relative pose ground truth of the interval frames $\mathbf{G}_{interval} = \{g_{i,i+2}\}_{i=1}^{k-2}$ is computed according to $\mathbf{G}^j$ through ψ as described in Eq. (5.8). Utilizing $g_{i,i+2}$ to optimize $r_{i,i+2}$ calculated from $T_{i,i+2}$ which recoverd through ψ by taking into account $r_{i,i+1}$ and $r_{i+1,i+2}$, with the constraint of MSE loss $\mathcal{L}_1$.

$$\begin{aligned} R_{i,i+1} &= R_z(e_z^i)R_y(e_y^i)R_x(e_x^i) \\ T_{i,i+2} &= T_{i,i+1}T_{i+1,i+2} \end{aligned} \tag{5.8}$$

where R_x, R_y, R_z are rotation matrixs along x, y, z-axis and rigid transformation $T_{i,i+1} = \begin{bmatrix} R_{i,i+1} & t_i \\ 0 & 1 \end{bmatrix}$.

Besides, we design an aggregation module which takes O as input dedicating to estimate egomotion $\dot{r}_{i,i+2}$ between two interval frames I_i and I_{i+2} to compute MSE loss $\mathcal{L}_2$. In addition, we also introduce a self-supervised loss $\mathcal{L}_3$ between $r_{i,i+2}$ and $\dot{r}_{i,i+2}$ that leveraging the pose consistency to accelerate network convergence. The aforementioned losses computing details are as follows:

$$\begin{aligned} \mathcal{L}_1 &= \sum_{i=1}^{k-2} ||r_{i,i+2} - g_{i,i+2}||^2 \\ \mathcal{L}_2 &= \sum_{i=1}^{k-2} ||\dot{r}_{i,i+2} - g_{i,i+2}||^2 \\ \mathcal{L}_3 &= \sum_{i=1}^{k-2} ||r_{i,i+2} - \dot{r}_{i,i+2}||^2 \end{aligned} \tag{5.9}$$

Loss Function: To learn the optimal parameters Θ for $\mathcal{F}_\Theta$, SACNet is jointly trained through the whole loss $\mathcal{L}_{all}$ as follow. Adam optimizer is adopted to iteratively update the parameters Θ to minimize $\mathcal{L}_{all}$:

$$\mathcal{L}_{all} = \mathcal{L} + \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \mathcal{L}_t \tag{5.10}$$

5.3 Experimental Results

5.3.1 Experimental Setup

Following the existing arts [36, 113, 116, 121, 163, 120, 38, 37], we evaluate our SACNet on the KITTI VO benchmark [39] and adopt the root mean square errors (RMSE) from 100, 200, and 800 meters as the measurement metric.

Dataset: The KITTI VO benchmark [39] has emerged as a popular dataset to evaluate the performance of VO/SLAM methods. It consists of 22 raw video sequences captured in various outdoor scenarios such as urban and highway, with speeds of up to 90km/h. Out of these sequences, only 11 (00-10) provide raw video frames and ground-truth labels for ego-motion. The remaining sequences, 11-21, solely provide raw video frames, making it an extremely challenging benchmark to evaluate the generalization capabilities of research works. To achieve a fair comparison with state-of-the-art approaches [36, 37] in omnidirectional evaluation, we adopt the same dataset split as them, using sequences 00, 02, 08, and 09 for training, and sequences 03, 04, 05, 06, 07, and 10 for testing.

Training setting: The proposed method is implemented in PyTorch and trained using an NVIDIA GTX Titan X GPU with 12 CUDA memory. To accommodate the hardware, all input video frames are resized to 640 $\times$ 192 pixels. We set 1000 hidden states in LSTM and utilize the Adam optimizer with parameters β_1 = 0.9 and β_2 = 0.999. To ensure complete network convergence, the model is trained up to 300 epochs, with initial learning rate 0.001 and decayed by 0.5 every 80 epochs. The video clip length, k, is set to 11, and the value of μ is carefully studied and set to 1500, as we discussed in Table 5.4. To enhance the robustness of learning, random masking, and central cropping are employed for data augmentation. It is worth noting that we pre-train FlowNet [118] on the FlyingChairs dataset [118].

Table 5.1: Comparison with Learning-based methods

Method	03		04		05		06		07		10		**Avg.**	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
2017 DeepVO [36]	8.49	6.89	7.19	6.97	2.62	3.61	5.42	5.82	3.91	4.60	8.11	8.83	5.95	6.12
2018 ESP-VO [113]	6.72	6.46	6.33	6.08	3.35	4.93	7.24	7.29	3.52	5.02	9.77	10.20	6.15	6.66
2018 GFS-VO-RNN [116]	6.36	3.62	5.95	2.36	5.85	2.55	14.58	4.98	5.88	2.64	7.44	3.19	7.67	3.22
2018 GFS-VO [116]	5.44	3.32	2.91	1.30	3.27	1.62	8.50	2.74	3.37	2.25	6.32	2.33	4.96	2.26
2019 Beyond Tracking [121]	3.32	2.10	2.96	1.76	2.59	1.25	4.93	1.90	**3.07**	1.76	3.94	1.72	3.47	1.75
2019 CL-VO [163]	8.12	3.47	7.57	2.61	5.77	2.00	7.66	1.66	6.79	3.00	8.29	2.94	7.37	2.66
2019 DeepVO+GA-CL [163]	8.36	3.53	8.66	3.08	5.81	2.10	7.39	1.83	9.79	4.13	8.30	3.03	7.47	2.95
2020 DAVO [120]	5.50	2.71	6.03	2.37	**2.28**	1.14	**4.19**	1.69	4.11	2.61	4.26	1.70	4.40	2.04
2021 TartanVO [38]	-	-	-	-	-	-	4.72	2.95	4.32	3.41	6.89	2.73	5.31	3.03
2022 DeepAVO [37]	3.64	1.89	3.88	0.60	2.57	1.16	4.96	1.34	3.36	2.15	5.49	2.49	3.98	1.61
SACNet	**3.03**	**1.76**	**2.22**	**0.53**	2.78	**1.08**	4.23	**1.14**	3.30	**1.62**	**3.56**	**1.48**	**3.19**	**1.27**

[1] $t_{rel}(\%)$: average translational RMSE drift $(\%)$ on length from 100, 200, 800 meters.
[2] $r_{rel}(^\circ)$: average rotational RMSE drift $(^\circ/100m)$ on length from 100, 200, 800 meters.
[3] The best results are highlighted.

5.3.2 Main results

Comparision with state-of-the-art: We extensively evaluate our proposed SACNet against 10 state-of-the-art (SOTAs), which are trained under their recommended settings using the same training split as our method. The results presented in Table 5.1 demonstrate the superior performance of our approach over the majority of the SOTAs. In comparison to DeepAVO [37], SACNet achieves an average diminution of 19.85% in translational RMSE drift and 21.12% in rotational RMSE drift. Moreover, SACNet achieves a significant improvement over DeepVO [36], with a reduction of translational RMSE drift from 5.95 to 3.19 and rotational RMSE drift from 6.12 to 1.27, exhibiting the superior performance of our proposed framework. One may infer from our experimental results that the proposed SACNet outperforms existing methods in terms of correcting steering angle estimation errors. To further substantiate the qualitative efficacy of our proposed approach, we illustrate the reconstructed trajectory comparison results in Figure 5.3. In addition, to assess the generalization capability of SACNet, we also present the trajectory comparison on sequences 11, 12, 15, 16, and 17 in Figure 5.4. Given the absence of ground truth information

for these sequences, we make use of the results of stereo ORB-SLAM2 [112] as a reference. The qualitative analysis showcases that our SACNet yields more accurate and reliable trajectories. This indicates that the incorporation of steering angle-weighted learning and triple-frame hybrid constraint learning places greater emphasis on the correction of corner errors and facilitates the attainment of smoother trajectories.

Table 5.2: Effectiveness of steering angle-weighted loss and triple-frame hybrid constraint learning

Variants		03		05	
Weighted loss.	Hybrid constriant.	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
✗	✗	3.42	2.48	3.18	2.10
✔	✗	3.18	1.97	3.06	1.34
✗	✔	3.21	2.06	2.98	1.56
✔	✔	**3.03**	**1.76**	**2.78**	**1.08**

5.3.3 Ablation Studies

Effectiveness of steering angle-weighted learning and triple-frame hybrid constraint learning: To verify the effectiveness of the steering angle-weighted loss and triple-frame hybrid constraint learning, we conduct the following experiments as shown in Table 5.2. If any of the components is removed, the estimated translation error and rotation error of the pose will increase. The fully equipped network demonstrates optimal accuracy, implying that these two proposed learning strategies enable the network to prioritize turning moments and achieve higher localization accuracy.

Study of GLA: We conduct the study experiments by removing the LSTM layers and attention operation in GLA and Table 5.3 shows the results. We can observe that while keeping the LSTM layers, the rotational error $r_{rel}(^\circ)$ decreased significantly compared to keeping only the attention operation, which indicates that the LSTM layers are effective in modeling temporal dependencies

for rotation prediction. Conversely, when keeping only the attention operation, the translational error $t_{rel}(\%)$ decreased significantly, suggesting that the long-term context captured by the multi-head attention mechanism effectively supplements the lost spatial information, leading to superior translation results.

Learning hyper-parameters in Φ: We initially investigate a range of functions Φ to be used for weighted mapping. Results are summarized in Table 5.4 (above), where it can be observed that the asymmetric function $(1 + \frac{1}{e^{\mu E}})^{-1}$ is incompatible with this task. Following that, we assess the performance of quadratic functions and symmetric primary functions, and our proposed function clearly outperforms them both. To gain a more profound comprehension of the weighted mapping function Φ, we systematically investigate the effect of varying values of parameter μ. The experimental setup is described in detail in Table 5.4 (bellow). Empirical observation suggests that steering angle falls within [-0.1, 0.1] mostly, and we intend to map the interval angle to the weight between 0 and 1. We initiate the investigation by setting μ to 500 and find that performance improves incrementally when μ is increased. However, we observed limited gain is received when μ exceeded 1500 and we attribute this to a substantial concentration of weights around 1.0 when $\mu > 1500$, which undermines the steering angle discrimination capability. Based on the above experimental analysis, We set $\mu = 1500$ as the official configuration.

5.4 Concluding Remarks

In this study, we present a novel end-to-end framework for steering angle correction in monocular visual odometry (VO). Our framework incorporates two key components: steering angle-weighted learning and triple-frame hybrid constraint learning. The former uses a dedicated branch to predict steering angles, which are then mapped to weights by a weighted mapping function. The latter achieves steering angle correction by imposing constraints between every two adja-

Table 5.3: Abalation study for GLA.

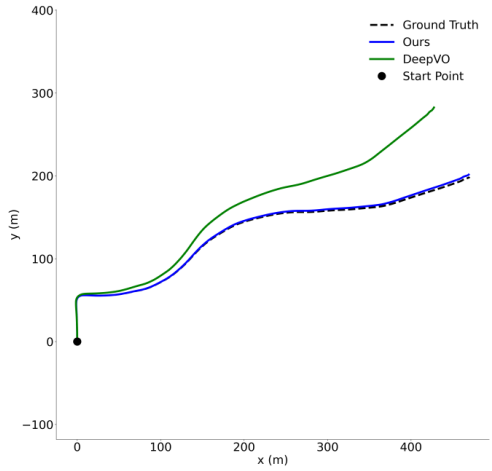
Variants		03		05	
LSTMs	Attention.	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
✗	✗	4.35	2.62	3.90	2.24
✔	✗	3.78	2.17	3.42	1.53
✗	✔	3.27	2.32	2.96	1.89
✔	✔	**3.03**	**1.76**	**2.78**	**1.08**

Table 5.4: Different mapping functions learning.

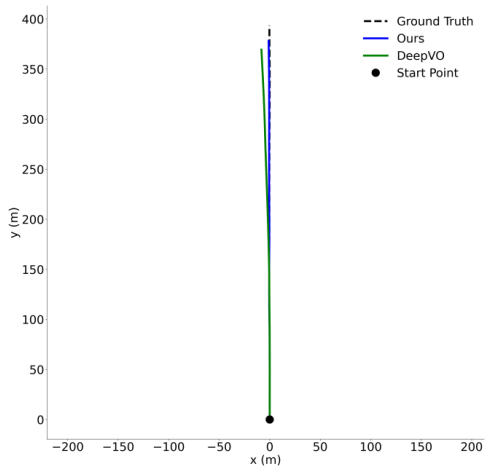
Φ	03		05	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
$(1+\frac{1}{e^{\mu E}})^{-1}$	6.44	3.51	5.69	2.57
$a\left\|E\right\|+b$	4.06	2.62	4.16	1.82
aE^2+b	3.55	2.08	3.35	1.65
Eq. (5.6)	**3.03**	**1.76**	**2.78**	**1.08**
μ	Analysis for the value of μ in Eq. (5.6).			
500	3.11	1.93	2.91	1.46
1000	3.10	1.88	2.84	1.25
1500	**3.03**	1.76	**2.78**	**1.08**
2000	3.07	**1.73**	2.82	**1.08**
2500	3.12	1.77	2.81	1.20

cent frames and interval frames contained in three consecutive frames. Our extensive experiments demonstrate that our approach outperforms previous learning-based monocular VO methods in terms of rotation and translation accuracy, setting new benchmarks in the field. Moreover, we be-

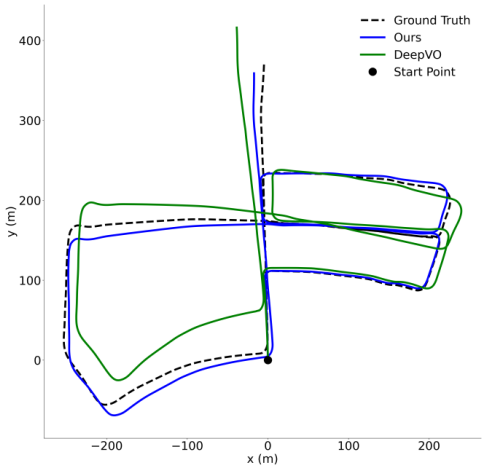
lieve that our framework can be easily integrated into conventional VO systems, making it highly applicable to real-world applications.



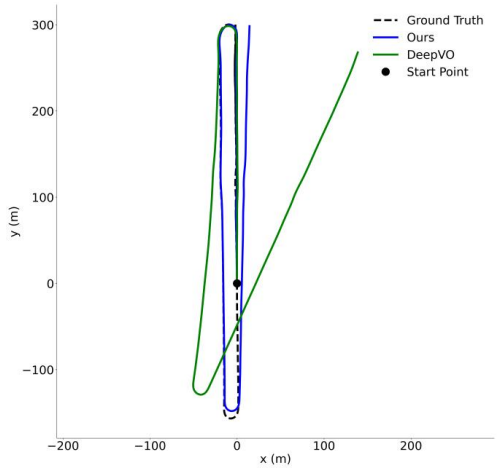
(a) Seq 03



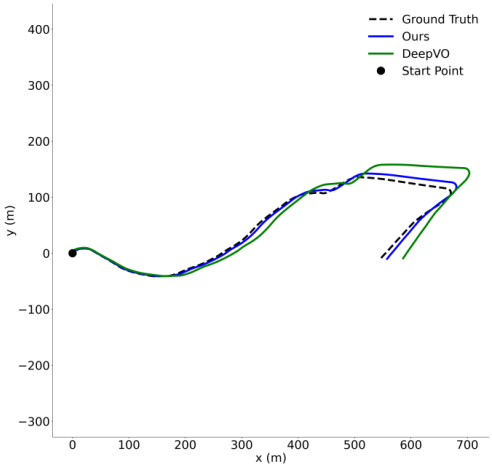
(b) Seq 04



(c) Seq 05

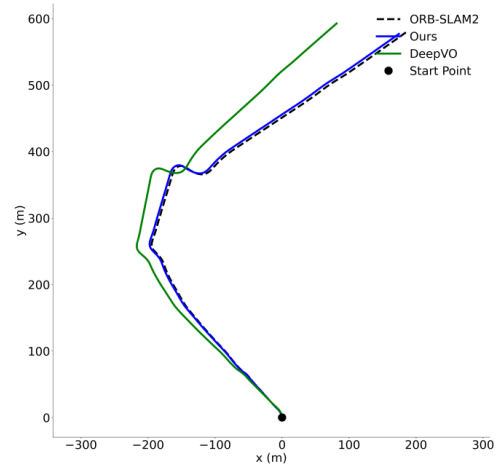


(d) Seq 06

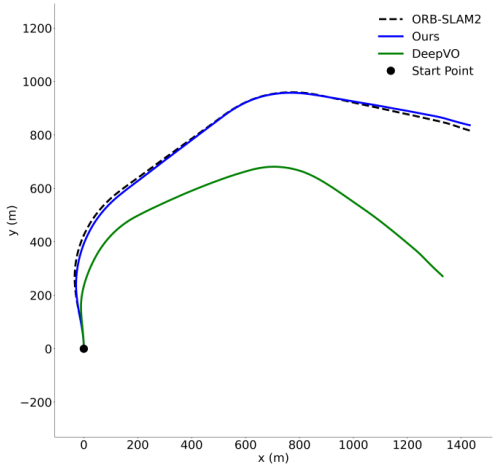


(e) Seq 10

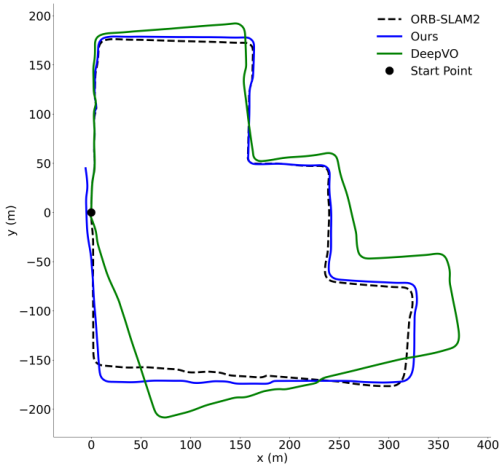
Figure 5.3: The trajectories of ground-truth, DeepVO [36] and Ours on Seq 03, 04, 05, 06, 10



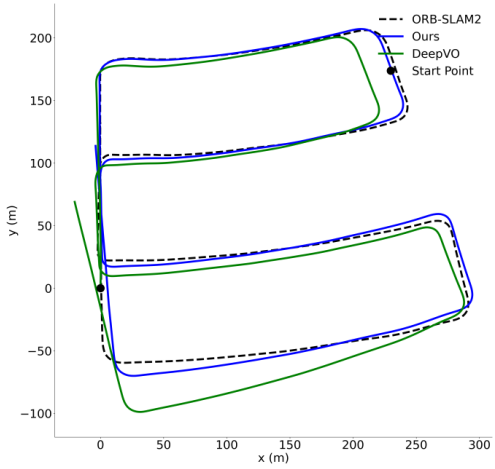
(a) Seq 11



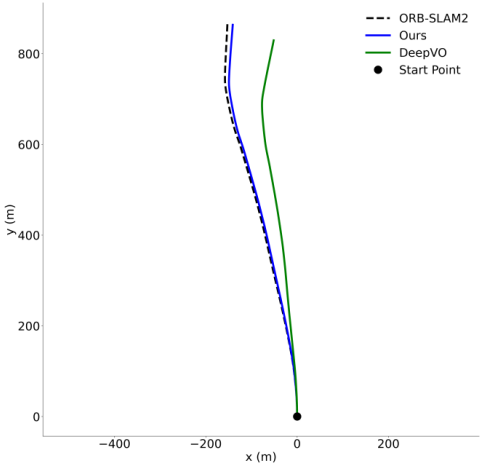
(b) Seq 12



(c) Seq 15



(d) Seq 16



(e) Seq 17

Figure 5.4: The trajectories of ORB-SLAM2 [112], DeepVO [36] and Ours on Seq 11, 12, 15, 16, 17 without ground-truth labels.

CHAPTER 6
CONCLUSIONS

Visual Localization is a vital task in computer vision and has a variety of applications including navigation, autonomous driving, loop closure in SLAM map building, *etc*. In this thesis, we study three major problems in the area of visual localization, namely, Video odometry for local visual local-ization, Visual Place Recognition (VPR) and cross-view geo-localization (CVGL) for global localization aiming to achieve a more robust visual localization in different scales.

In chapter 3, we proposed a novel visual place recognition framework called SGG-NetVLAD that combines the powerful visual feature extractor from SOTA backbone and semantic feature from embedding scene graph generated from a scene graph generator with semantic label weighting. The features from two modalities are then fused with a siamese network to compute the final similarity score for re-ranking. Through experiments, SGG-NetVLAD proves to be exceptionally effective on dataset that includes seasonal changes such as Nordland while maintaining comparable result for other urban setting dataset, indicating that SGG-VLAD is able to produce more discriminative similarity scores for queries with huge appearance changes while not undermining the effectiveness of the visual features, thanks to the semantic label weighting and feature fusion components. For future works, there are two main approaches: (1) include attention for visual features. Currently, methods such as Patch-NetVLAD [3] does not implement patch-level attention, and treats all patches equally when calculating the similarity score; however, it is obvious that this is sub-optimal since not all patches are discriminative and the confusing patches should have a lower weight when computing similarity score. For example, the dynamic objects in the scene such as human beings and cars in an urban dataset should have lower weights, similar to what we

did with our semantic label weighting. An intuitive way to improve on this is to provide patch level attention to patch-level features with the help of vision transformers. (2) We want to generate scene graphs that better fits the need of visual place recognition. Currently, our scene graph generator [122] is trained on Visual Gnome [129] and is not optimal for VPR for obvious reasons such as the semantic label do not always overlap with the semantic objects that appear in a VPR dataset. However, it is not practical to get the same level of annotation for VPR datasets as Visual Gnome due to the huge labor effort. One potential approach we can consider is utilizing transfer learning and adapt the scene graph generator onto VPR datasets with a comparatively small labeled VPR dataset. Also, we currently implement label weighting based on the generated scene graph, where there is a potential risk that the semantic objects we are interested in do not exist at the top confidence ranking list; so, we think it would be better if we can implement such attention into the generation process of scene graphs.

In chapter 4 , we proposed a novel mutual generative transformer learning network, denoted as MGTL, for addressing the cross-view geo-localization problem. Existing methods commonly rely on a CNN-based Siamese-like backbone to extract high-order feature representations and treat each region equally. Viewpoint-sensitive regions with drastic appearance differences, however, hinder image matching significantly. Using a cascaded attention-masking algorithm, we introduced a spatial context enhancement module and a spatial attention module in the VGG16 to capture co-visual information. In terms of semantic consistency learning, it is rarely examined in recent works, but incorporating consistency constraints by cross-view interaction during the recurrent learning process will benefit similarity computing. To facilitate high-order information mining within each view, we constructed cross-view generative modules and injected their generative cross-view knowledge into a transformer-based framework. Extensive qualitative and quantitative experiments demonstrated that mutual generative transformer learning significantly alleviated

the impact of spatial information mismatch caused by drastic viewpoint changes. By examining cross-view interactions, we highlighted the potential of this perspective to advance automobile geo-location identification research in GPS-denied conditions. In our future work, we will explore how mutual learning techniques can play a role in this similar field, including: (1) The slight difference in view perspective between the satellite view and UAV view makes the cross-view semantic consistency easier to obtain, allowing mutual learning to go further in enhancing semantics; and (2) Shared parameter learning, which can make the network more efficient, should be explored in the context of mutual learning.

Finally, in chapter 5, we present a novel end-to-end framework for steering angle correction in monocular visual odometry (VO). Our framework incorporates two key components: steering angle-weighted learning and triple-frame hybrid constraint learning. The former uses a dedicated branch to predict steering angles, which are then mapped to weights by a weighted mapping function. The latter achieves steering angle correction by imposing constraints between every two adjacent frames and interval frames contained in three consecutive frames. Our extensive experiments demonstrate that our approach outperforms previous learning-based monocular VO methods in terms of rotation and translation accuracy, setting new benchmarks in the field. Moreover, we believe that our framework can be easily integrated into conventional VO systems, making it highly applicable to real-world applications.

To sum up, in this thesis, we studied three problems in visual localization of different scales and we aimed to create more robust visual localization systems in a variety of environments. In future works, our goal will be to continue improve our current visual localization methods for better performance and robustness under dynamic environments, and explore the possibility of utilizing our techniques in similar fields.

REFERENCES

[1] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognition*, vol. 74, pp. 90–109, 2018.

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

[4] S. Lowry *et al.*, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[6] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 883–890.

[7] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier, "A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes," *IEEE transactions on robotics*, vol. 36, no. 2, pp. 561–569, 2019.

[8] Y. Tian, J. Miao, X. Wu, H. Yue, Z. Liu, and W. Chen, "Discriminative and semantic feature selection for place recognition towards dynamic environments," *Pattern Recognition Letters*, vol. 153, pp. 75–82, 2022.

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

[10] G. Peng, Y. Yue, J. Zhang, Z. Wu, X. Tang, and D. Wang, "Semantic reinforced attention learning for visual place recognition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13 415–13 422.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[13] O. Saurer, G. Baatz, K. Köser, M. Pollefeys, *et al.*, "Image based geo-localization in the alps," *International Journal of Computer Vision*, vol. 116, pp. 213–225, 2016.

[14] T. Senlet and A. Elgammal, "Satellite image-based precise robot localization on sidewalks," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 2647–2653.

[15] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2020, doi: 10.1109/TITS.2020.3013234.

[16] S. Wang, Y. Zhang, and H. Li, "Satellite image based cross-view localization for autonomous vehicle," *arXiv preprint arXiv:2207.13506*, 2022.

[17] J. Thoma, D. P. Paudel, A. Chhatkuli, T. Probst, and L. V. Gool, "Mapping, localization and path planning for image-based navigation using visual features and map," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7383–7391.

[18] N. Roy and S. Debarshi, "Uav-based person re-identification and dynamic image routing using wireless mesh networking," in *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2020, pp. 914–917.

[19] S. Hu and G. H. Lee, "Image-based geo-localization using satellite imagery," *IJCV*, vol. 128, no. 5, pp. 1205–1219, 2020.

[20] S. Workman and N. Jacobs, "On the location dependence of convolutional neural network features," in *IEEE/CVF Winter Conference on Computer Vision and Pattern Recognition*, 2015, pp. 70–78.

[21] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *European Conference on Computer Vision*, Springer, 2016, pp. 494–509.

[22] S. Hu, M. Feng, R. M. Nguyen, and G. H. Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7258–7267.

[23] K. Regmi and M. Shah, "Bridging the domain gap for ground-to-aerial image matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 470–479.

[24] S. Zhu, M. Shah, and C. Chen, "Transgeo: Transformer is all you need for cross-view image geo-localization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1162–1171.

[25] H. Yang, X. Lu, and Y. Zhu, "Cross-view geo-localization with layer-to-layer transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 009–29 020, 2021.

[26] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," 2014.

[27] Z. Xin *et al.*, "Localizing discriminative visual landmarks for place recognition," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5979–5985.

[28] A. Khaliq, M. Milford, and S. Garg, "Multires-netvlad: Augmenting place recognition training with low-resolution imagery," *IEEE Robotics and Automation Letters*, pp. 3882–3889,

[29] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, "Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition," *IEEE Transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 661–674, 2019, doi: 10.1109/TNNLS.2019.2908982.

[30] Y. Latif, R. Garg, M. Milford, and I. Reid, "Addressing challenging place recognition tasks using generative adversarial networks," in *IEEE European Conference on Computer Vision*, IEEE, 2018, pp. 2349–2355.

[31] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1998–2006.

[32] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, "A light-weight semantic map for visual localization towards autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 11 248–11 254.

[33] Y. Liu, H. Wang, J. Wang, and X. Wang, "Unsupervised monocular visual odometry based on confidence evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5387–5396, 2021.

[34] U.-H. Kim, S.-H. Kim, and J.-H. Kim, "Simvodis++: Neural semantic visual odometry in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4244–4251, 2022.

[35] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. Le Besnerais, "Ov2slam: A fully online and versatile visual slam for real-time applications," *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 1399–1406, 2021.

[36] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 2043–2050.

[37] R. Zhu, M. Yang, W. Liu, R. Song, B. Yan, and Z. Xiao, "Deepavo: Efficient pose refining with feature distilling for deep visual odometry," *Neurocomputing*, vol. 467, pp. 22–35, 2022.

[38] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," in *Conference on Robot Learning*, PMLR, 2021, pp. 1761–1772.

[39] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 3354–3361.

[40] B. Steder, G. Grisetti, and W. Burgard, "Robust place recognition for 3d range data based on point features," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 1400–1405.

[41] X. Kong *et al.*, "Semantic graph based place recognition for 3d point clouds," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 8216–8223.

[42] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *arXiv preprint arXiv:1706.06064*, 2017.

[43] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, pp. 145–175, 2001.

[44] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, pp. 2130–2137.

[45] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[46] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture notes in computer science*, vol. 3951, pp. 404–417, 2006.

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

[48] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[49] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *2012 IEEE conference on computer vision and pattern recognition*, Ieee, 2012, pp. 510–517.

[50] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2548–2555.

[51] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2008.

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

[53] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[54] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[55] E. Hörster and R. Lienhart, "Deep networks for image retrieval on large-scale databases," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 643–646.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

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

[58] J. Wan *et al.*, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 157–166.

[59] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1741–1750.

[60] G. Zhu *et al.*, "Scene graph generation: A comprehensive survey," *arXiv preprint arXiv:2201.00443*, 2022.

[61] J. Zhang, K. Shih, A. Tao, B. Catanzaro, and A. Elgammal, "An interpretable model for scene graph generation," *arXiv preprint arXiv:1811.09543*, 2018.

[62] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 685–10 694.

[63] Y.-S. Wang, C. Liu, X. Zeng, and A. Yuille, "Scene graph parsing as dependency parsing," *arXiv preprint arXiv:1803.09189*, 2018.

[64] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5410–5419.

[65] D. Teney, L. Liu, and A. van Den Hengel, "Graph-structured representations for visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1–9.

[66] P. Wang, Q. Wu, J. Cao, C. Shen, L. Gao, and A. v. d. Hengel, "Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1960–1968.

[67] Y. Zhu, B. Sun, X. Lu, and S. Jia, "Geographic semantic network for cross-view image geo-localization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021, doi: 10.1109/TGRS.2021.3121337.

[68] B. Zhu, C. Yang, J. Dai, J. Fan, and Y. Ye, "R2fd2: Fast and robust matching of multimodal remote sensing image via repeatable feature detector and rotation-invariant feature descriptor," *IEEE Transactions on Geoscience and Remote Sensing*, 2023, doi: 10.1109/TGRS.2023.3264610.

[69] Y. Zhu, H. Yang, Y. Lu, and Q. Huang, "Simple, effective and general: A new backbone for cross-view image geo-localization," *arXiv preprint arXiv:2302.01572*, 2023.

[70] X. Zhang, X. Li, W. Sultani, Y. Zhou, and S. Wshah, "Cross-view geo-localization via learning disentangled geometric layout correspondence," *arXiv preprint arXiv:2212.04074*, 2022.

[71] K. Regmi and A. Borji, "Cross-view image synthesis using conditional gans," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3501–3510.

[72] X. Lu, Z. Li, Z. Cui, M. R. Oswald, M. Pollefeys, and R. Qin, "Geometry-aware satellite-to-ground image synthesis for urban areas," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 859–867.

[73] H. Ding, S. Wu, H. Tang, F. Wu, G. Gao, and X.-Y. Jing, "Cross-view image synthesis with deformable convolution and attention mechanism," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, Springer, 2020, pp. 386–397.

[74] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5007–5015.

[75] B. Sun, C. Chen, Y. Zhu, and J. Jiang, "Geocapsnet: Aerial to ground view image geo-localization using capsule network," *arXiv preprint arXiv:1904.06281*, 2019.

[76] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[77] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, "Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8391–8400.

[78] B. Ren, H. Tang, and N. Sebe, "Cascaded cross mlp-mixer gans for cross-view image translation," *arXiv preprint arXiv:2110.10183*, 2021.

[79] A. Toker, Q. Zhou, M. Maximov, and L. Leal-Taixé, "Coming down to earth: Satellite-to-street view synthesis for geo-localization," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6488–6497.

[80] A. Vaswani *et al.*, "Attention is all you need," vol. 30, 2017.

[81] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[82] H. Chen *et al.*, "Pre-trained image processing transformer," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[83] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 231–10 241.

[84] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 478–16 488.

[85] R. Strudel, R. G. Pinel, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7262–7272.

[86] Y. Jin, D. Han, and H. Ko, "Trseg: Transformer for semantic segmentation," *Pattern Recognition Letters*, vol. 148, pp. 29–35, 2021.

[87] S. Zheng *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[88] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*, Springer, 2020, pp. 213–229.

[89] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2906–2917.

[90] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[91] Z. Liang, Y. Wang, L. Wang, J. Yang, and S. Zhou, "Light field image super-resolution with transformers," *IEEE Signal Processing Letters*, vol. 29, pp. 563–567, 2022.

[92] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5728–5739.

[93] Z. Li *et al.*, "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6197–6206.

[94] Y. Ding *et al.*, "Transmvsnet: Global context-aware multi-view stereo network with transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8585–8594.

[95] X. He, Y. Chen, and Z. Lin, "Spatial-spectral transformer for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 3, p. 498, 2021.

[96] Y. Qing, W. Liu, L. Feng, and W. Gao, "Improved transformer net for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 11, p. 2216, 2021.

[97] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022, doi: 10.1109/TGRS.2022.3144158.

[98] H. Zhou, C. Tian, Z. Zhang, Q. Huo, Y. Xie, and Z. Li, "Multispectral fusion transformer network for rgb-thermal urban scene semantic segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[99] X. Chu *et al.*, "Conditional positional encodings for vision transformers," *arXiv preprint arXiv:2102.10882*, 2021.

[100] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," *arXiv preprint arXiv:2104.05707*, 2021.

[101] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 357–366.

[102] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[103] F. Yang *et al.*, "Uncertainty-guided transformer reasoning for camouflaged object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4146–4155.

[104] W. Wang, L. Yao, L. Chen, D. Cai, X. He, and W. Liu, "Crossformer: A versatile vision transformer based on cross-scale attention," *CoRR abs/2108.00154*,

[105] L. R. Agostinho, N. M. Ricardo, M. I. Pereira, A. Hiolle, and A. M. Pinto, "A practical survey on visual odometry for autonomous driving in challenging scenarios and conditions," *IEEE Access*, vol. 10, pp. 72 182–72 205, 2022.

[106] Q. Zhai *et al.*, "Mgl: Mutual graph learning for camouflaged object detection," *IEEE Transactions on Image Processing*, pp. 1–1, 2022.

[107] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, "Mutual graph learning for camouflaged object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 997–13 007.

[108] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Ieee, vol. 1, 2004, pp. I–I.

[109] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[110] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, IEEE, 2007, pp. 225–234.

[111] C. Engels, H. Stewénius, and D. Nistér, "Bundle adjustment rules," *Photogrammetric computer vision*, vol. 2, no. 32, 2006.

[112] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[113] Wang, Sen and Clark, Ronald and Wen, Hongkai and Trigoni, Niki, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[114] G. Zhai, L. Liu, L. Zhang, Y. Liu, and Y. Jiang, "Poseconvgru: A monocular approach for visual ego-motion estimation by learning," *Pattern Recognition*, vol. 102, p. 107 187, 2020.

[115] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, IEEE, 2017, pp. 1597–1600.

[116] F. Xue, Q. Wang, X. Wang, W. Dong, J. Wang, and H. Zha, "Guided feature selection for deep visual odometry," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI*, Springer, 2019, pp. 293–308.

[117] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[118] A. Dosovitskiy *et al.*, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[119] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[120] X.-Y. Kuo, C. Liu, K.-C. Lin, and C.-Y. Lee, "Dynamic attention-based visual odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 36–37.

[121] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond tracking: Selecting memory and refining poses for deep visual odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8575–8583.

[122] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3716–3725.

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

[124] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, Springer, 2020, pp. 726–743.

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

[126] B. Rozemberczki and R. Sarkar, "Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1325–1334.

[127] A Mousavian and J Kosecka, *Semantic image based geolocation given a map*, 2016.

[128] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 883–890.

[129] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 10 313–10 322.

[130] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *CVPR*, 2015.

[131] D. Olid, J. M. Fácil, and J. Civera, "Single-view place recognition under seasonal changes," *arXiv preprint arXiv:1808.06516*, 2018.

[132] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5107–5116.

[133] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese, "Semantic cross-view matching," in *IEEE/CVF International Conference on Computer Vision Workshops*, 2015, pp. 9–17.

[134] S. Zhu, T. Yang, and C. Chen, "Vigor: Cross-view image geo-localization beyond one-to-one retrieval," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3640–3649.

[135] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," vol. 34, no. 07, pp. 11 990–11 997, 2020.

[136] T. Wang, S. Fan, D. Liu, and C. Sun, "Transformer-guided convolutional neural network for cross-view geolocalization," *arXiv preprint arXiv:2204.09967*, 2022.

[137] T. Wang *et al.*, "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 867–879, 2021, doi: 10.1109/TCSVT.2021.3061265.

[138] T. Wang, Z. Zheng, Z. Zhu, Y. Gao, Y. Yang, and C. Yan, "Learning cross-view geo-localization embeddings via dynamic weighted decorrelation regularization," *arXiv preprint arXiv:2211.05296*, 2022.

[139] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3961–3969.

[140] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5624–5633.

[141] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6024–6042, 2021, doi: 10.1109/TPAMI.2021.3085766.

[142] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.

[143] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 867–875.

[144] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[145] Y. Shi, X. Yu, D. Campbell, and H. Li, "Where am i looking at? joint location and orientation estimation by cross-view matching," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4064–4072.

[146] J. Lin *et al.*, "Joint representation learning and keypoint detection for cross-view geo-localization," *IEEE Transactions on Image Processing*, vol. 31, pp. 3780–3792, 2022, doi: 10.1109/TIP.2022.3175601.

[147] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[148] H. Jie, S. Li, and S. Gang, "Squeeze-and-excitation networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[149] J. J. Liu, Q. Hou, M. M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 096–10 105.

[150] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[151] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510–519.

[152] S. Liu, D. Huang, *et al.*, "Receptive field block net for accurate and fast object detection," in *European Conference on Computer Vision*, 2018, pp. 385–400.

[153] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," in *28th ACM international conference on Multimedia*, 2020, pp. 1395–1403.

[154] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucleic acids research*, vol. 31, no. 13, pp. 3812–3814, 2003.

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

[156] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[157] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25*, Springer, 2003, pp. 236–243.

[158] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.

[159] L. Zhuo, Z. Geng, J. Zhang, and X. guang Li, "Orb feature based web pornographic image recognition," *Neurocomputing*, vol. 173, pp. 511–517, 2016.

[160] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, and N. Y.-C. Chang, "Fast sift design for real-time visual feature extraction," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3158–3167, 2013.

[161] Q. Zhai, F. Yang, X. Li, G.-S. Xie, H. Cheng, and Z. Liu, "Co-communication graph convolutional network for multi-view crowd counting," *IEEE Transactions on Multimedia*, 2022.

[162] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[163] M. R. U. Saputra, P. P. De Gusmao, S. Wang, A. Markham, and N. Trigoni, "Learning monocular visual odometry through geometry-aware curriculum learning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3549–3555.

[164] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.