POLITECNICO DI TORINO

Master degree in Data science and Engineering

Master Thesis

Deep Learning for Visual Geo-localization

Re-ranking methods for visual geo-localization with domain shift

Relatore
Prof. Barbara Caputo

Correlatore:
PhD. Carlo Masone
Dott. Gabriele Berton
Dott. Gabriele Trivigno

Laureando
Mohamad Mostafa
matricola: 291385

Academic year 2022 – 2023

Deep Learning for Visual Geo-localization
Master thesis. Politecnico di Torino, Turin.

February 2023.

Acknowledgements

Finishing this thesis has been a very challenging milestone to accomplish in my academic career. It has been a significant experience that taught me a lot in the field of deep learning.

I would like to thank my family who has always been supportive of me, helped me push through my problems, and worked hard to get me to where I am today.

To my friends who cheered me up and helped get my mind off of all the stress I went through.

To Professor Barbara Caputo who presented me with a great opportunity in the deep learning domain, Ph.D. Carlo Mansone who helped guide and organize my work, and Doctors Gabriele Berton and Gabriele Trivigno who answered all my questions at all times and were very helpful and made my work possible, I am very grateful.

To my colleagues Giovanni who helped me achieve the results of my thesis by assisting me with the code and experiments, and Hajali who participated in this research.

To everyone who participated in making this happen, I am thankful for your effort and support.

Abstract

Visual Geo-localization is the task of estimating the geographical coordinates where a given photo has been taken. The problem is well known in computer vision literature and the common approach relies on image retrieval technique. Recent works achieved high performances leveraging deep convolutional neural network to embed an image in a fixed low dimensional sized vector. However, we observe how domain shift is still a big challenge and the accuracy of these methods can drop when facing such a challenge, for example, a dataset of query images taken at night. In this work, we explore how re-ranking methods based on spatial verification and deep learning can handle this problem by providing new benchmark with state-of-the-art models on datasets with night queries. Moreover, we introduced a new labeled dataset that contains night query images taken in San Francisco.

Contents

Chapter 1

Introduction

1.1 Thesis's objectives

The goal of this thesis is to evaluate the performance of different re-ranking methods extending the main Visual Geo-localization (VG) methods which regards estimating the position of an image taken which we call a query image, based on a set of database images that have been previously collected. VG has received a significant amount of attention in the past years in applications like outdoor navigation systems for autonomous driving where GPS signal may not be reliable depending on the environment, augmented reality, and 3D reconstruction. Although some applications require the 6 degrees-of-freedom position of an object, the focus in this thesis is estimating the location of a photo with a large-scale database covering big areas like cities with a tolerance of a few meters. The use of this task in real applications poses some constraints to be followed:

- **Scalable**: it should be able to match an image against a very large database representing big areas like cities or regions;
- **Performance**: it should predict accurate locations of street level with few meters of error;
- **Efficiency**: it should perform fast so that it can be applied for real-time applications;
- **Robust**: the model should be domain invariant with little effects of domain shift, and it should be able to overcome appearance and viewpoint changes.

1.2 Related works and main problems

VG has been getting more attention and a lot of studies and research is being done on this topic focusing on different aspects of the way to approach like classification or image retrieval. Moreover, many research done for image retrieval and image matching can be used and implemented into the VG model. The purposes of those researches varied in their focus on some major problems that make VG a challenging task:

- **Domain shift**: Domain shift is one of the major problems that computer vision tasks usually have to handle, and in the case of VG, taking a photo at night for example for any real-time application, and matching it with database with day images is difficult;
- **Different viewpoints**: matching the same scene from different viewpoints is not a trivial task for a model;
- **Scenery changes**: The image taken for a scene might change due to some environmental changes or some construction work, also there are vehicles and pedestrians which are dynamic in images which the model may take into consideration as interesting objects affecting its performance;

1.3 Our contribution: research & development

We used as a baseline state-of-the-art VG model that performs image retrieval where given a query image for which the model should estimate coordinates, this image is compared to an extensive database of images with known coordinates. The location of the closest matching image is the predicted position for the query image. Our experiments' main focus is the re-ranking path of the VG pipeline where given a number of candidates retrieved from the database given by the image retrieval model, a more dense model focusing more on local context is to be used for the feature extraction of these candidates, followed by a matching and scoring algorithm for these features which results in the new re-ranked candidates. Although a decrease in efficiency results from such an approach, it can still be managed to a certain point by configuring the number of candidates to re-rank depending on the application, meaning how much we are willing to sacrifice the performance, and on

the other hand, we are getting a significant performance increase, and it is very robust against domain shift. Finally, we propose a labeled query dataset that covers night images in San Francisco which will help further evaluate the robustness of the re-ranking approach. This work has been done with the help of a group of colleagues, which resulted in a submission under review for the CVPR 2023 Workshop.

Chapter 2

Related Works

2.1 VG Approaches

VG has been handled with different approaches in the literature, with image retrieval being developed recently achieving very good results, and this work uses the re-ranking approach which is another version depending on image retrieval.

2.1.1 Image Retrieval

It is a very common approach used by several recent works where a given query image has to be located by matching it to a set of labeled images called a database, where the label from the retrieved image matched from the database is to be the predicted location for the query. The way images are matched requires an image representation that has been handled in different ways, either by global descriptors or local descriptors for which a similarity search like a k-nearest neighbor or a more efficient alternative like approximate nearest neighbor that is supported with FAISS library [1] which benefit from the GPU for faster computation which will lead to the top-k closest matches. For local features there exist other methods like spatial verification which can be used for refining the retrieved candidates by confirming the reliability of matched features using algorithms like RANSAC [2], it can still be used before producing the retrieved candidates.

2.1.1.1 Global Descriptors

Global Features: They are represented with a compact vector that contains the high-level context of an image, which is an advantage considering global information to match images, in addition to its efficiency compared to other approaches, but this also has its disadvantages when facing high viewpoint shifts. To extract these features, GIST and HOG [3, 4] are hand-crafted feature representations are used which process the whole image to extract a single global vector, and there are other methods depending on a Convolutional Neural Network (CNN) to learn to extract features which have been improving lately like CosPlace [5] which has been used in this work as the retrieval method, and [6] that is built upon using VLAD embedding and aggregation. Generalized mean pooling (GeM) [7] is another recent method that generalizes max and average pooling which is differentiable.

2.1.1.2 Local Descriptors

They represent an image at the local level which may be a pixel or a patch, it sacrifices efficiency to get higher performance. Some methods following this approach extract sparse features where they try to find relevant regions by using keypoint detectors [8, 9]and extract the features of those patches or neighbor of the center pixel of the detected regions [10, 11]. Hand-crafted feature extractors have been used like SIFT [12], SURF [13], RootSIFT [14], whereas, recent methods using convolutional representations have been frequently used due to being more robust to challenges that might be faced for image retrieval like R2D2 [15], D2-Net [16], and SuperPoint [17] which are used in this work.

2.1.2 Classification Approach

This approach has been adopted by some methods considering the visual geo-localization (VG) task as classification where the goal is to predict the location of the image. Planet [18] is the first study formulating the VG problem as a classification where either the whole earth's surface or the region we are interested in is to be divided into disjoint cells where each cell is a class. Planet further applies subdivision to have a balanced number of images among classes, but this affects the accuracy of cells with smaller sizes. CPlanet [19] has tried to overcome this issue by applying different coarse divisions and using a classifier backbone with fully connected layers corresponding to each coarse division where cumulative scores are used from

the multiple classifiers overlapping over a region included in multiple coarse divisions.

2.1.3 Re-ranking Approach

It is an extension of the image retrieval approach where a further step of refinement is applied to the candidates retrieved from the image retrieval pipeline, and it is the approach we are using in this work. This approach is a compromise for the performance and efficiency trade-off since the retrieval method is based on global descriptors and then the local descriptors are only used on these candidates which overcome the computational problem and the memory problem of saving descriptors. Methods like DELG [20], CVNet [21], and TransVPR [22] have been used in this work.

2.2 Domain Shift for VG

The focus in this work is on the domain shift challenge that the VG task encounters, where in some applications like autonomous driving cars, the query images captured by the car may be taken at night whereas the geo-tagged database is collected in day conditions, which will be an issue for the retrieval model to perform matching with high accuracy. AdAGeo [23] was recently introduced addressing the task of cross-domain for visual geo-localization based on a generative approach that produces domain-invariant features resulting in higher performance on different domains. Moreover, another work that tackled night-day domain shift for visual localization was done by ToDayGAN [24] which is based on ComboGAN [25] which applies image-to-image translation transforming the night images to daytime representation and it was associated with DenseVLAD [26]. [27] is another method that was used to transform images across domains like day to night for visual geo-localization by training a CNN architecture that produces synthetic images with the ability to apply feature mapping with real images.

Part I

Chapter 3

Data collection

3.1 Building Night Domain Dataset

Introducing a night-domain dataset helps in further evaluating the robustness and performance of our proposed method against images with low light conditions where extracting image features is extremely challenging to match against those extracted from images with normal light conditions, and this is particularly important in applications like navigation systems and urban planning.

To build the new dataset with night queries named SF-XL test night dataset, we use the same database of the test split introduced by [5], whereas, for the query set, we collected images located in San Francisco area from Flickr using a script that downloads large batches of images from a given area given the coordinates of the borders. The download images are labeled but might be inaccurate which will be dealt with. Since the collected images are taken everywhere including 'indoor' and 'outdoor', we used a pre-trained classifier based on EfficientNet [28] that divides images into categories depending on the level of 'indoorness', producing 10 categories plus an extra panoramic category, from which we chose the 6 most 'outdoor' categories and checked manually for any mislabeled image that belongs to indoor in the chosen categories.

The next step is to handle selecting night images from the outdoor images previously chosen which is the main goal for this dataset, this required using a day-night classifier based on MobileNetV2 [29] pre-trained on ImageNet [30] available on [31] to separate night images from day images. Furthermore, we relabeled the night images manually using automated software that compares them to street view images with the same location and scene from `www.`

`instantstreetview.com` and when the correct coordinates and heading are found on the website we can press on 'Next' and the software is gonna change the label for that image to the new one (Fig. 3.1). Then we reduced the images taken for the same scene, since Flickr images are mainly taken by smartphones where people take pictures of touristic places or famous scenery, we can see how the images are distributed over the San Francisco area in 3.2. Finally, we compared manually positive reference images with a threshold under 25 meters with respect to each query image and removed the latter in case no positive reference image matched the same scene as the query.

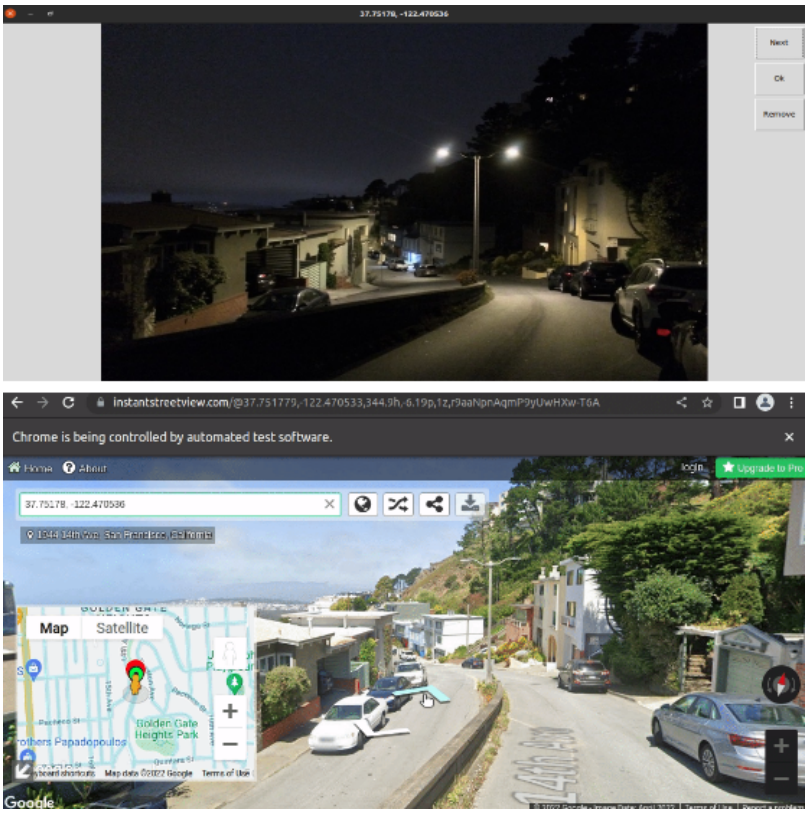

Figure 3.1: **Screenshot of the software that helps relabel the images**. In case the image is labeled correctly press 'next'. If it is mislabeled move to the correct location on `www.instantstreetview.com` and press 'next' to relabel, otherwise press 'ok' to skip image.

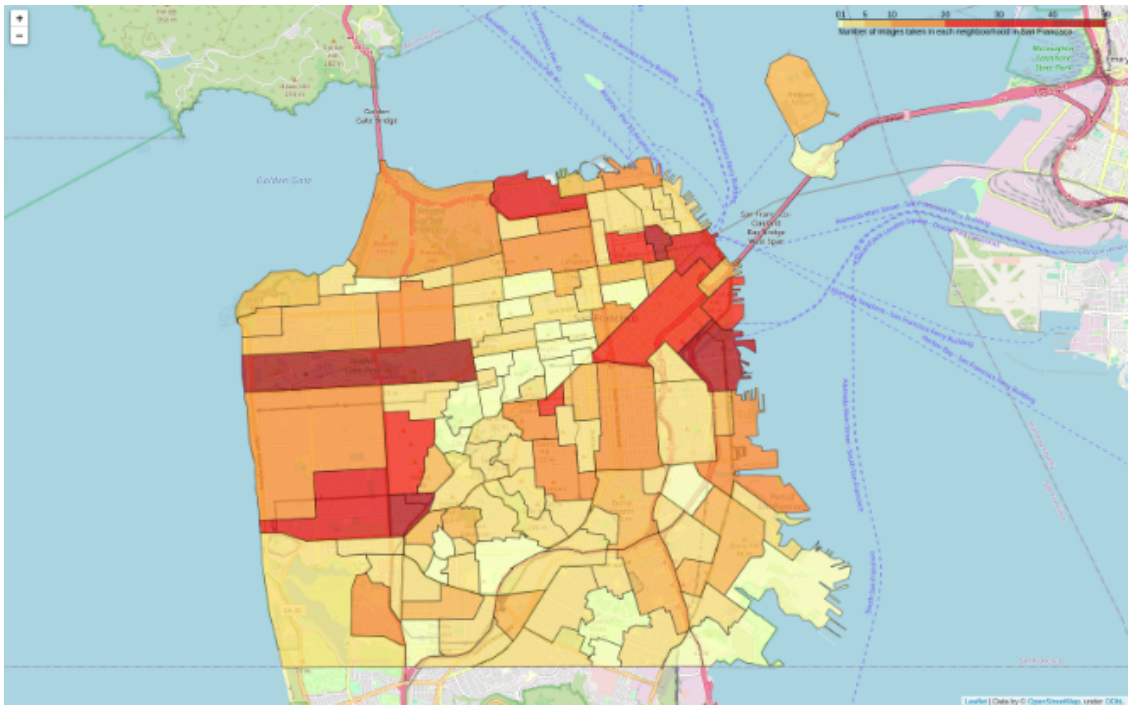

Figure 3.2: **Choropleth map.** It shows the distribution from which neighbors the image are from after cleaning.

3.2 SVOX Night

Street View Oxford Dataset (SVOX) was proposed by [23] for the purpose of evaluating a VG model for cross-domains, it is a large-scale dataset with a wide coverage of Oxford. The database and query with a single domain were collected from Google Street View taken from two different years where for each query at least a single positive database image exists. To address multiple domains, a subset of each of the five domain were used from Oxford RobotCar Dataset [32] with 5 meters distance between consecutive images to avoid redundancy, these images also have the hood of the car visible. For our work, we used the night queries provided by SVOX and the database from the test split, and we call it SVOX night.

3.3 Tokyo Night

Tokyo Night is a subset of Tokyo24/7 that was introduced in [26] to address the challenge of illumination changes and the variability like construction of

buildings, at each location three images were taken by smartphone, each in a different direction than the other and at three different times, the query set has for the same scene a day, sunset, and night image. Tokyo Night has the same database as Tokyo24/7 but just the night image subset of the queries.

3.4 San Francisco Extra Large

San Francisco Extra Large Dataset (SF-XL) [5] was built as a largely dense dataset that covers the whole city of San Francisco with challenging cases including the variability due to its collection between long-term. The Database is collected by splitting panoramas from Google Street View imagery. The two test query sets suggested in the papers were used which are associated with the test split of the database set. Test set v1 contains images collected from Flickr which contain some illumination and viewpoint challenges, and their given coordinates were verified, and test set v2 is from the queries of San Francisco Landmark Dataset [33] taken with smartphones where its 6 DoF coordinates were generated by [34].

Datasets	# Query images	# Database images
Tokyo24/7	315	75 K
Tokyo Night	105	
SVOX Night	823	17 K
SF-XL test v1	1000	2.8 M
SF-XL test v2	598	
SF-XL test night (ours)	466	

Table 3.1: **Datasets statistics**

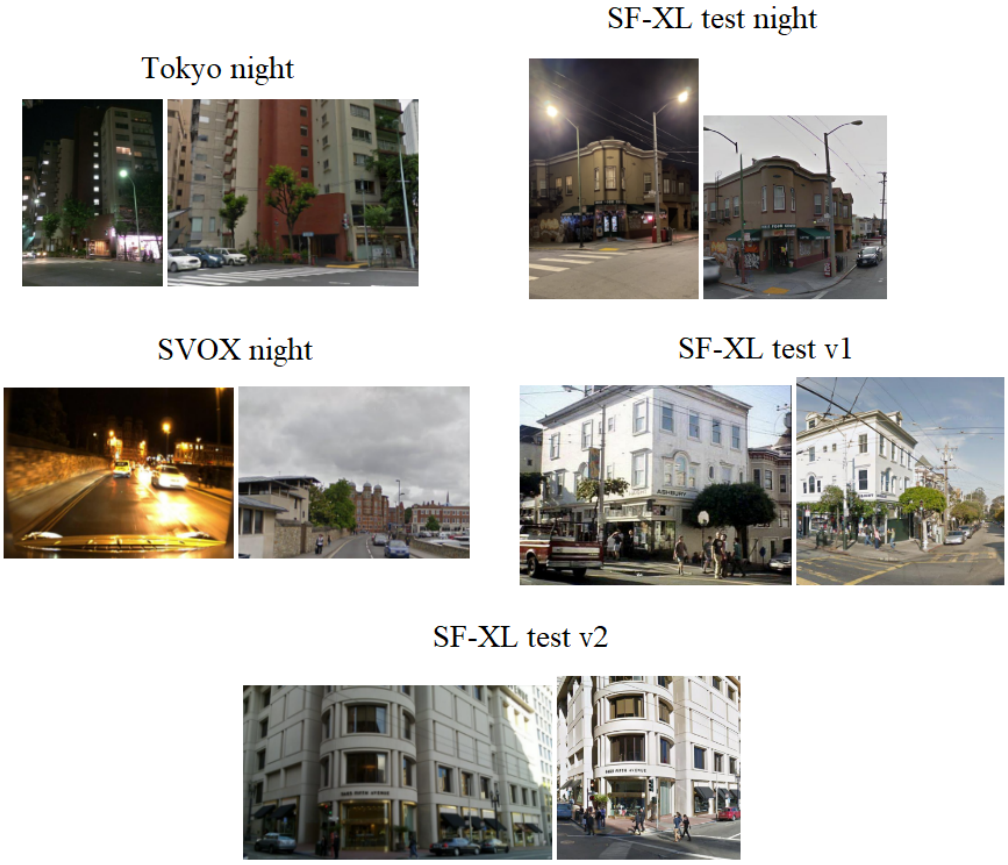

Figure 3.3: **Datasets sample.** It shows a query sample (left image) and its corresponding positive reference image (right image).

Part II

Chapter 4

Architectures and Experiments

4.1 Pipeline Overview

The used approach in these experiments is the re-ranking approach which is composed of several components (Fig. 4.1). For the first component, we choose state-of-the-art VG model CosPlace [5] which will act as the baseline for this benchmark without any re-ranking, with the goal of retrieving top-k candidates which in our case we chose $k = 100$, for the second phase, the re-ranking includes using a model which extracts local features of hybrid features which will be further discussed in 4.3, these features are then matched by selecting mutual neighbors which will be followed by scoring methods discussed in 4.4 which will refine the retrieved candidates producing the final predictions. Moreover, we mention another approach that is complete including their own extractor, matching, and scoring in 4.5. Finally, the combinations of scoring and feature extractors used in addition to the hyperparameters associated with them are used as suggested by their authors.

4.2 VG Retrieval Model

CosPlace [5] has recently achieved state-of-the-art performance for large-scale VG tasks on datasets like SF-XL [5] and Tokyo 24/7 [26]. It is the first block of the VG pipeline we used. CosPlace is a highly scalable and memory-efficient VG model where at training the model needs to compute global

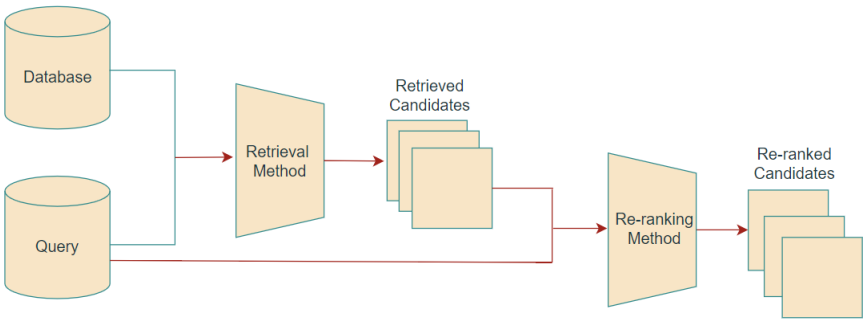

Figure 4.1: **The re-ranking pipeline used in our experiments.**

descriptors for all images in the database and store them without having time or space issues or the need to use dimensionality reduction methods, [5] approaches the training phase of VG as a classification problem which makes it suitable for this task. At inference time it uses the image retrieval approach by extracting features from a query and database where matching can be applied to choose the closest image where the label is known to predict the query's location.

4.2.1 Training Process

CosPlace follows a similar approach to cosFace [35] which requires having different classes that are not given with VG datasets since their labels are coordinates. The author suggests performing splits into the dataset following specific criteria:

- Divide the area that the dataset covers into squares with side M
- Map the coordinate labels with longitude, latitude (represented by east and north respectively), and heading angle into new coordinates according to the square cells division defining each class C_{e_i,n_j,h_k} where $e_i = \lfloor \frac{east}{M} \rfloor$, $\lfloor n_j = \frac{north}{M} \rfloor$, $\lfloor h_k = \frac{heading}{\alpha} \rfloor$ with α the parameter for heading extent of a class.
- Divide classes in CosPlace Groups and train over them iteratively, where each group contains classes separated with a minimum number of cells N and within a minimum heading angle L
- Train on the $N \times N \times L$ groups using Large Margin Cosine Loss (LCML) [35]

CosPlace can use different Convolutional Neural Network (CNN) backbones followed by Generalized Mean (GeM) pooling [7] giving an output of a 512-global descriptor.

4.2.2 Inference Time

During inference time, the fully connected layer is discarded since there are no classes considered with the image retrieval approach, but only to extract the global descriptors using the trained backbone so that matching can be applied with nearest neighbor search using Faiss [1], developed by the Facebook AI Research team. The backbone we used is ResNet-50 [36] and the weights pre-trained on SF-XL Dataset [5].

4.3 Feature Extractors for Re-ranking

4.3.1 SuperPoint

SuperPoint [17] is one of the re-ranking networks we used, which is a fully convolutional self-supervised model trained on the whole image that jointly computes keypoint locations and the corresponding local descriptors. The network architecture overview is shown in figure 4.2 where there are three main components that this is built from. SuperPoint takes a gray-scale image as input where a shared encoder is applied to outputting two similar feature maps, one for the descriptor extractor head and the other for the keypoint detector head where each head results in a specific feature map to the task required. This will be further discussed in more detail in the next sections.

4.3.1.1 Shared Encoder

This is the first component that will be applied to the input gray-scale image, It has similar architecture to VGG [37] having eight 3×3 convolutional layers where every layer is followed by ReLU non-linear activation function, and every two convolutional layers there is a 2×2 max-pooling layer for the goal of reducing the dimensionality of the input image, which means for this encoder there are 3 max-pooling layers with kernel 2×2 leading to an output feature map with 1/8 the size of the input image and with 128 channels, so given an input gray-scale image $I \in \mathbb{R}^{H \times W}$ where $H \times W$ is the input image's size, denoting the encoder function as E, the output feature map is $E(I) = \mathcal{B} \in \mathbb{R}^{H_c \times W_c \times 128}$ where $H_c = H/8$ and $W_c = W/8$.

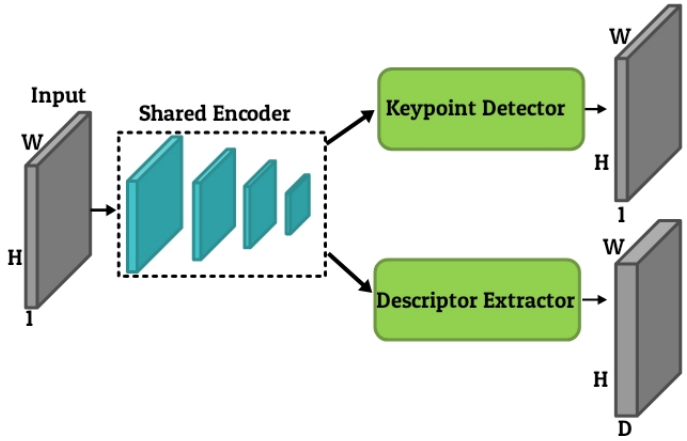

Figure 4.2: **Overview of the SuperPoint architecture.**

4.3.1.2 Keypoint Detection

The keypoint detection process is a challenging part because the model should detect keypoints in a repeatable way even if the viewpoint of illumination changes for a certain image scene. The keypoint decoder head takes $\mathcal{B}$ as input and outputs an upscaled feature map with the same size as the input image I by applying two convolutional layers to $\mathcal{B}$ to get a feature map $\mathcal{X} \in \mathbb{R}^{H_c \times W_c \times 65}$ followed by a channel-wise softmax which will give probability scores, then an upscale process is applied to get a dense output probability scores for each pixel from the input following sub-pixel convolution from [38]. To train the detector the author followed a self-supervised framework to make it more robust. The main problem was that there is no common definition of keypoints or interest points of an image for all tasks that a model can agree on. The semantics that a model is trained on for face recognition to define interest points focusing on the corner of an eye and other facial features is different than what a model should focus on in VG where buildings and other structures and vegetation are the regions that matter to check the matching of two images, especially with the presence of occlusions from vehicles and pedestrians that should be ignored while detecting keypoints because they are dynamic objects that cannot be matched or will lead to a wrong matching. To solve this problem the author worked on creating a synthetic dataset called Synthetic Shapes which is created by rendering simple

shapes like triangles, quadrilaterals, and ellipses and labeling their keypoints without ambiguity by using simple junctions types. The detector path of SuperPoint is pre-trained on this dataset resulting in a model called MagicPoint with high performance compared to other traditional corner detection approaches on scenes having corner-like structured objects. but it is still not robust enough to generalize to real images with natural image scenery. To make the detector more robust on real images, homographic adaptation was introduced. SuperPoint was trained on MS-COCO, but since it is unlabeled for keypoint detection, MagicPoint is used to generate pseudo-ground truth labels for the MS-COCO dataset with the use of homographic adaptation. To perform homographic adaptation starting with MagicPoint and unlabeled images from MS-COCO. The author generates a set of homographies such as scaling, translation, in-plane rotations, etc., where a homography is a geometric transformation mapping points from one image to corresponding points from another. The aim is to have a detector that results in corresponding keypoints when applied to different viewpoints which are performed by homography in this case. So let $f_\theta(.)$ be the keypoint detector function, $\mathcal{H}$ be a random homography, I the input image, and $\mathbf{x}$ the detected keypoints, then:

$$\mathbf{x} = f_\theta(I). \tag{4.1}$$

And the keypoint extracted by the detector should be covariant to the homography:

$$\mathcal{H}\mathbf{x} = f_\theta(\mathcal{H}(I)), \tag{4.2}$$

which gives:

$$\mathbf{x} = \mathcal{H}^{-1} f_\theta(\mathcal{H}(I)). \tag{4.3}$$

And since in practice, the detector is not perfectly covariant, Homographic Adaptation was introduced with the following steps: Given H a set of random homographies, MagicPoint $f_\theta(.)$, unlabeled image I, $\mathbf{x}_I$ empty list of keypoints, and N_h number of homographies to sample:

- sample a random homography $\mathcal{H}_i$
- warp the unlabeled image $\mathcal{H}_i(I)$
- apply MagicPoint to get the keypoints for the warped image $\mathbf{x}_{warp} = f_\theta(\mathcal{H}_i(I))$
- compute the unwarped keypoints $\mathbf{x}_{unwarp} = \mathcal{H}_i^{-1} f_\theta(\mathcal{H}_i(I))$

- store $\mathbf{x}_{unwarp}$ into $\mathbf{x}_I$
- repeat the previous steps N_h times
- aggregate the keypoints stored in $\mathbf{x}_I$ to produce the super-point detector $\hat{F}(.)$:

$$\hat{F}(I; f_\theta) = \frac{1}{N_h} \sum_{i=1}^{N_h} \mathcal{H}_i^{-1} f_\theta(\mathcal{H}_i(I)). \tag{4.4}$$

The detector is trained with convolutional cross-entropy loss given two synthetically warped images having pseudo-ground-truth keypoints generated by MagicPoint and ground truth correspondence generated by the homography transforming one of the images to the other, we define the loss as:

$$\mathcal{L}_p(\mathcal{X}, Y) = \frac{1}{H_c W_c} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_C} l_p(\mathbf{x}_{hw}; y_{hw}), \tag{4.5}$$

where:

$$l_p(\mathbf{x}_{hw}; y) = -\log\left(\frac{\exp(\mathbf{x}_{hwy})}{\sum_{k=1}^{65} \exp(\mathbf{x}_{hwk})}\right). \tag{4.6}$$

4.3.1.3 Descriptor Extractor

The descriptor head takes $\mathcal{B}$ as input from the shared encoder and applies two convolutional layers to it to give a feature map $\mathcal{D} \in \mathbb{R}^{H_c \times W_c \times D}$ where D is fixed to 256, then perform Bi-Cubic Interpolation to upscale the feature map to the original size as the input image I and normalize it with L2-normalization leading to a fixed length dense descriptor. After the pseudo-ground truth labels have been generated by the MagicPoint with homographic adaptation, a homography is sampled from a different set than that for homographic adaptation to transform the image and its corresponding pseudo-ground truth to produce the pair of inputs which the SuperPoint model will be jointly trained on using a loss applied to descriptors from the resulting input $\mathbf{d}_{hw} \in \mathcal{D}$ and $\mathbf{d}'_{hw} \in \mathcal{D}'$ from the original and transformed input images respectively. The correspondences between two cells (h, w) and (h', w') from both images transformed by the homography are defined as:

$$s_{hwh'w'} = \begin{cases} 1, & \text{if } \|\widehat{\mathcal{H}\mathbf{p}_{hw}} - \mathbf{p}_{\mathbf{h'w'}}\| \leqslant 8 \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

where $\mathbf{p}_{hw}$ denotes the location of the center pixel in the (h, w) cell. The set of correspondences for a pair of images is represented with S. The descriptor loss uses hinge loss as:

$$\mathcal{L}_d(\mathcal{D}, \mathcal{D}', S) = \frac{1}{(H_c W_c)^2} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_C} \sum_{\substack{h'=1 \\ w'=1}}^{H_c, W_C} l_d(\mathbf{d}_{hw}, \mathbf{d'}_{h'w'}; s_{hwh'w'}), \tag{4.8}$$

where

$$l_d(\mathbf{d}, \mathbf{d}'; s) = \lambda_d * s * \max(0, m_p - \mathbf{d}^T \mathbf{d}') + (1 - s) * \max(0, \mathbf{d}^T \mathbf{d}' - m_n). \tag{4.9}$$

m_p and m_n are the positive and negative margins respectively.

Since the model is trained jointly, the final loss is the sum of the detector loss from 4.3.1.2 and the descriptor loss 4.9:

$$\mathcal{L}(\mathcal{X}, \mathcal{X}', \mathcal{D}, \mathcal{D}'; Y, Y', S) = \mathcal{L}_p(\mathcal{X}, Y) + \mathcal{L}_p(\mathcal{X}', Y') + \lambda \mathcal{L}_d(\mathcal{D}, \mathcal{D}', S). \tag{4.10}$$

4.3.2 D2-Net

Proposed by [16], it was aimed at finding stable pixel-level correspondences extracted from Structure from Motion (SfM) reconstructions where the corresponding local features can be used to match images reliably under challenging conditions like illumination or viewpoint changes. The idea is based on having characteristics of a sparse local features approach leading to an efficient model in addition to being able to perform better on images with different viewpoints or illuminations changes since sparse local features approach uses small regions to detect keypoints which is unstable on challenging conditions to be extracted with repeatability. The model is a single convolutional neural network trained jointly for both tasks following the describe-and-detect approach where detection is postponed into a later stage as shown in figure 4.3.

4.3.2.1 Descriptor Extraction

The D2-Net pipeline starts with a descriptors extraction phase where a convolutional neural network with a VGG-like architecture [37] was used to truncate the network until `conv4_3` pre-trained on ImageNet [30] represents the feature extractor defined as $\mathcal{F}$ is to be applied on an input image I to output a feature map $F = \mathcal{F}(I), F \in \mathbb{R}^{h \times w \times n}$ where n is the number of

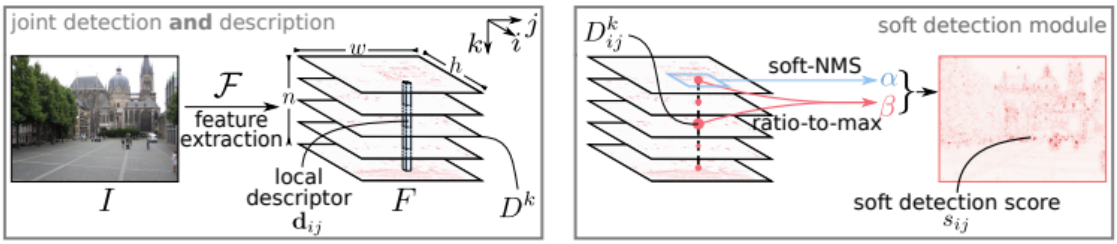

Figure 4.3: **D2-Net training pipeline.** It follows the detect and describe approach. Image from [16].

channels and $h \times w$ is the feature map resolution. Each element (i, j) in F represents a descriptor vector $\mathbf{d}$ across its channels as:

$$\mathbf{d}_{ij} = F_{ij:}, \mathbf{d} \in \mathbb{R}_n. \tag{4.11}$$

These descriptors are adjusted at training so that the same points in a scene lead to similar descriptors even under challenging conditions, followed by L2-normalization.

4.3.2.2 Feature Detection

For this phase the author suggested a different annotation to help interpret the detection stage by defining each channel of the feature map F as:

$$D^k = F_{::k}, D^k \in \mathbb{R}^{h \times w} \tag{4.12}$$

For this approach **Hard Feature Detection** is used to determine if a point (i, j) is to be detected if and only if:

$$D^k_{ij} \text{ is a local maximum in } D^k, \text{with } k = \arg\max_t D^t_{ij}. \tag{4.13}$$

This approach is further softened to be differentiable for training with back-propagation suggesting a **Soft Feature Detection** where they start by computing a soft local maximum score:

$$\alpha^k_{ij} = \frac{\exp D^k_{ij}}{\sum_{(i',j') \in \mathcal{N}(i,j)} \exp D^k_{i',j'}}, \tag{4.14}$$

where $\mathcal{N}(i, j)$ is the set of 9 neighbours of the pixel (i, j) including it. Then defining the soft channel selection that scales each value in the 2D response with respect to the max channel-wise to act as non-maximum suppression:

$$\beta^k_{ij} = D^k_{ij} / \max_t D^t_{ij}. \tag{4.15}$$

Both criteria are then taken into account to obtain a single score map:

$$\gamma_{ij} = \max_k (\alpha_{ij}^k \beta_{ij}^k). \tag{4.16}$$

The normalization is applied to obtain the soft detection score s_{ij}:

$$s_{ij} = \gamma_{ij} \Big/ \sum_{(i',j')} \gamma_{i'j'} \ . \tag{4.17}$$

Due to the lack of data labeled for pixel-wise correspondences between images. The author used MegaDepth Dataset [39] considering pairs of images with at least 50% overlap in the sparse SfM point cloud.

4.3.2.3 Training Loss

The main attributes that the model should be trained for that the author suggested is producing repeatable keypoints under challenging changes and distinctive descriptors, so they used an extended version of triplet margin ranking loss used by [40, 41] that jointly optimizes for both attributes. The pixel-wise correspondences between two images I_1 and I_2 are defined by $c : A \leftrightarrow B$ where $A \in I_1$ and $B \in I_2$, the idea is to minimize the distance between their descriptors $\hat{\mathbf{d}}_A^{(1)}$ and $\hat{\mathbf{d}}_B^{(2)}$ while maximizing the distance to other descriptors that are the hardest negative samples to them $\hat{\mathbf{d}}_{N_1}^{(1)}$ and $\hat{\mathbf{d}}_{N_2}^{(2)}$. The positive descriptor distance between corresponding descriptors is:

$$p(c) = ||\hat{\mathbf{d}}_A^{(1)} - \hat{\mathbf{d}}_B^{(2)}||_2, \tag{4.18}$$

whereas the negative distance is:

$$n(c) = \min \left(||\hat{\mathbf{d}}_A^{(1)} - \hat{\mathbf{d}}_{N_2}^{(2)}||_2, ||\hat{\mathbf{d}}_{N_1}^{(1)} - \hat{\mathbf{d}}_B^{(2)}||_2 \right) \tag{4.19}$$

The triple margin ranking loss is:

$$m(c) = \max \left(0, M + p(c)^2 - n(c)^2 \right). \tag{4.20}$$

which is extended leading to the proposed loss:

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_{q \in \mathcal{C}} s_q^{(1)} s_q^{(2)}} m(p(c), n(c)) \tag{4.21}$$

where $s_c^{(1)}$ and $s_c^{(2)}$ are the soft detection scores in 4.17 and $\mathcal{C}$ is the set of correspondences between the two images.

4.3.2.4 Test time

At test time the feature extractor is modified to output a larger resolution to improve localization. Additionally, multiple scales are considered, $\{0.5, 1, 2\}$; to build an image pyramid as in [42] to obtain a more robust feature extractor by propagating image structures from lower to higher feature maps as in [43].

4.3.3 R2D2

Repeatable and Reliable Detector and Descriptor (R2D2) [15] is another model that aims at jointly learning to detect keypoints and compute image descriptors. In addition to finding repeatable and sparse keypoints with unsupervised loss, the model has to predict reliable descriptors that are discriminative with high confidence that can accurately match a pair of images while overcoming the repetitiveness of some unnecessary salient regions that a model might focus on like windows. The model takes an image as input and outputs pixel-wise descriptors and pixel-wise confidence maps (Fig. 4.4).

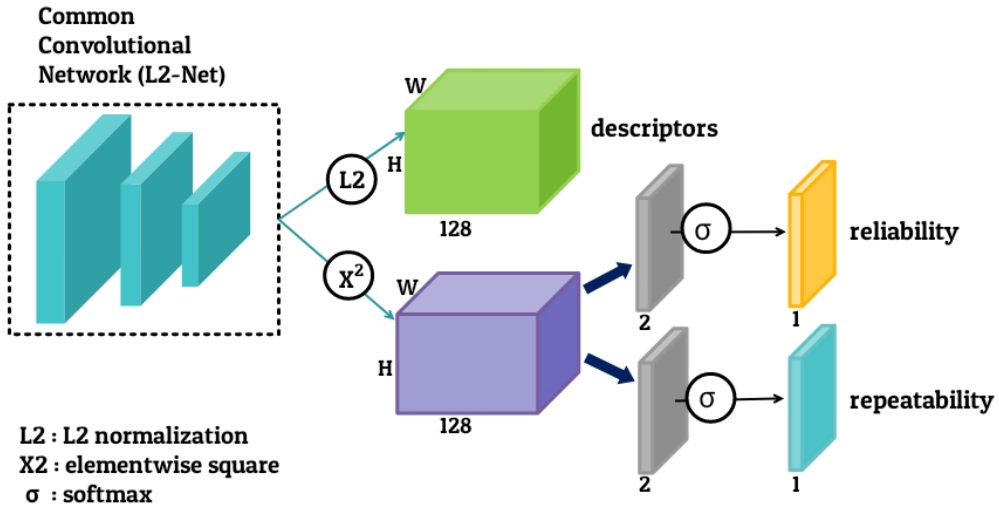

Figure 4.4: **R2D2 architecture overview.**

4.3.3.1 Convolutional Network

The first component in this network is a fully convolutional network that is common for the two paths that the R2D2 has. This component is based on L2-Net [44] with the addition to replacing the final 8×8 convolutional layer with 3 successive 2×2 convolutional layers to reduce the number of parameters. This convolutional network takes an image $I \in \mathbb{R}^{H \times W \times 3}$ and outputs a feature map $\mathcal{B} \in \mathbb{R}^{H \times W \times 128}$.

4.3.3.2 Descriptor path

The output is produced by the means of L2-normalization applied to the final layer of the common convolutional network $\mathcal{B}$ from 4.3.3.1, the resulting feature map is defined as $\mathbf{X} \in \mathbb{R}^{H \times W \times 128}$ where each vector $\mathbf{X}_{ij}$ with $i = 1...W$ and $j = 1...H$ correspond to the (i, j) pixel in the input image I.

4.3.3.3 Repeatability and Reliability path

The second path of the common convolutional network results in two confidence maps, one for repeatability and the other for reliability. $\mathcal{B}$ is squared element-wise giving a feature map $\mathcal{K} \in \mathbb{R}^{H \times W \times 128}$.

- **Repeatability Sub-Path** The output feature map of this sub-path is produced by applying 1×1 convolutional layer on $\mathcal{K}$ that outputs a 2-channel feature map where a softmax is applied after that to produce a one-channel heatmap $\mathbf{S} \in [0, 1]^{H \times W}$. This path aims at detecting sparse repeatable keypoints. Training for keypoints detection is not an easy task due to the lack of labeled datasets for this task. The author mentioned 'We thus treat the repeatability as a self-supervised task and train the network such that the positions of local maxima in $\mathbf{S}$ are covariant to natural image transformations like viewpoint or illumination changes' to handle this problem. So they define a 3D tensor $U \in \mathbb{R}^{H \times W \times 2}$ such that $U_{ij} = (i', j')$ where $(i, j)$and(i', j') are pixels corresponding to each other from images I and I' respectively. Similarly, $\mathbf{S'}_U$ is the correspondences from repeatability maps $\mathbf{S}$ and $\mathbf{S'}$. In order to make local maxima covariant, they aim to maximize the cosine similarity between $\mathbf{S}$ and $\mathbf{S'}_U$. Moreover, to overcome occlusion which impacts this goal, the maximization objective is done locally and then averaged. So they define a set of

overlapping patches $\mathcal{P} = p$ giving a loss:

$$\mathcal{L}_{cosim}(I, I', U) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} cosim(\mathbf{S}[p], \mathbf{S'}_U[p]), \tag{4.22}$$

where $\mathbf{S}[p] \in \mathbb{R}_{N^2}$ is the flattened vectorized $N \times N$ patch p extracted from $\mathbf{S}$, similarly for $\mathbf{S'}_U[p]$. To avoid having the heatmaps constant leading to minimum loss a second loss is employed to maximize the local peakiness of the repeatability map:

$$\mathcal{L}_{peaky}(I) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left(\max_{(i,j) \in p} \mathbf{S}_{ij} - \underset{(i,j) \in p}{\text{mean}} \mathbf{S}_{ij} \right). \tag{4.23}$$

Leading to the final repeatability loss:

$$\mathcal{L}_{rep}(I, I', U) = \mathcal{L}_{cosim}(I, I', U) + \lambda(\mathcal{L}_{peaky}(I) + \mathcal{L}_{peaky}(I')). \tag{4.24}$$

- **Reliability Sub-Path** It follows a similar operation as repeatability where 1×1 convolutional layer followed by softmax is applied to $\mathcal{K}$ leading to a confidence map $\mathbf{R}_{ij} \in [0, 1]$ representing the discriminativeness of each descriptor in $\mathbf{X}$ since only descriptors belonging to salient regions with high confidence are to be used for matching. The author treats the problem as a ranking optimization problem where each descriptor from an image is ranked with respect to one of the descriptors from another image. However, following the work of [45] where an exhaustive Euclidean Distance is computed between query patches from a given batch to all patches from another to maximize the average precision. The author uses L2-Net [44] in a fully convolutional way where each pixel from an image defining a patch is compared to others in another image, then proposes a loss that ignores regions with repeatability patterns or lacking distinction as:

$$\mathcal{L}_{AP\kappa}(i, j) = 1 - [AP(i, j)\mathbf{R}_{ij} + \kappa(1 - \mathbf{R}_{ij})], \tag{4.25}$$

with $\kappa \in [0, 1]$ indicating the minimum AP per batch chosen as 0.5 because it yielded in good results.

4.3.4 DELG

Deep Local and Global features (DELG) [20] is a model similar to the VG pipeline we are following with image-level supervision, where global features

are extracted and top candidates from the database are retrieved by matching the database global descriptors with that of the query, then local feature extractor is used to re-rank the retrieved candidates. There are some additional operations suggested in DELG that reduces the dimensionality of the local features which the author introduced and an attention-based keypoint detection. The final re-ranked candidates suggested by DELG depend on the scores computed using both global and local descriptors. In our pipeline, we used the global descriptors of DELG to compute this score instead of that extracted by CosPlace. The pipeline overview is shown in figure 4.5

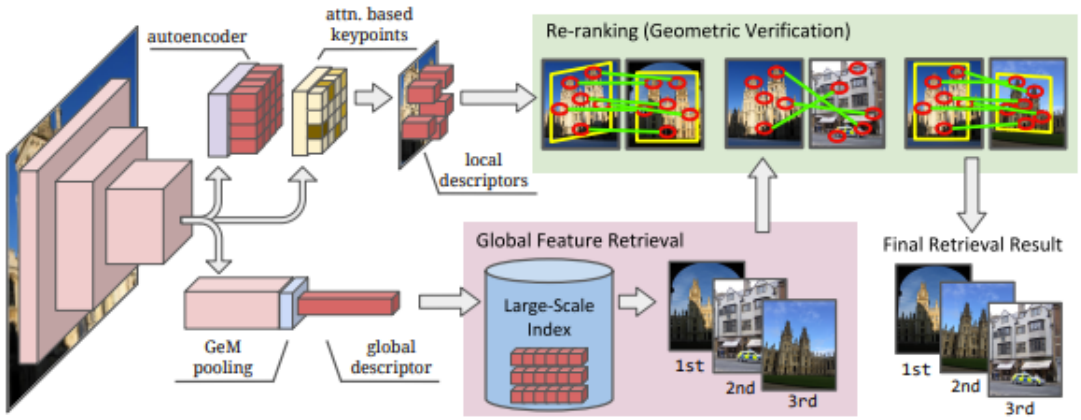

Figure 4.5: **DELG architecture overview.** Image from [20]

4.3.4.1 Common Convolutional Network

The pipeline starts with a common convolutional network backbone which is ResNet-50 where a deeper feature map $\mathcal{D} \in \mathbb{R}^{H_D \times W_D \times C_D}$ obtained from the output of *conv*5 layer with $C_D = 2048$ will be used for the global descriptors and shallower one $\mathcal{S} \in \mathbb{R}^{H_S \times W_S \times C_S}$ obtained from the output of *conv*4 is used for the local descriptors with $C_S = 1024$.

4.3.4.2 Global Descriptors

Since a global descriptor is a compact vector representation that should have a high-level semantic context of an image which is why we use the feature maps belonging to deeper layer $\mathcal{D}$. To obtain global descriptors generalized mean pooling (GeM) is used as in [7], then whitening of the obtained aggregated features by applying fully connected layer $F \in \mathbb{R}^{C_F \times C_D}$ with $C_F = 2048$

and learned bias $b_F \in \mathbb{R}^{C_F}$ as in [46] to produce global feature $g \in \mathbb{R}^{C_F}$, this path can be summarized as:

$$g = F \times \left(\frac{1}{H_D W_D} \sum_{h,w} d_{h,w}^p \right)^{1/p} + b_F \tag{4.26}$$

where p is the generalized mean pooling power parameter, and $d_{h,w} \in \mathbb{R}^{C_D}$ are features from map $\mathcal{D}$ at location (h, w). g is then L2-normalized into $\hat{g}$. The global feature path uses ArcFace margin loss [47]:

$$\mathrm{AF}(u, c) = \begin{cases} \cos(\mathrm{acos}(u) + m), & \text{if } c = 1 \\ u, & \text{if } c = 0 \end{cases} \tag{4.27}$$

where u is the cosine similarity, m is the ArcFace margin, and c representing the binary ground-truth class. In addition, the cosine classifier which is cross-entropy loss and scaled softmax normalization applied to L2-normalized classifier weights $\hat{\mathcal{W}}$ is computed as:

$$L_g(\hat{g}, y) = -\log \left(\frac{\exp\left(\gamma \times \mathrm{AF}(\hat{w}_k^T \hat{g}, 1)\right)}{\sum_n \exp\left(\gamma \times \mathrm{AF}(\hat{w}_n^T \hat{g}, y_n)\right)} \right) \tag{4.28}$$

where $\hat{w}_i$ is the L2-normalized weights for class i, γ is a learnable scalar, y is a one-hot label vector and k is the index of the ground-truth class.

4.3.4.3 Dimensionality Reduction

An autoencoder module [48] is learned jointly with the other components without extra supervision to reduce the dimensionality of the local features trained with reconstruction loss. The encoder T is applied to $\mathcal{S}$ to obtain the local descriptors $\mathcal{L} \in \mathbb{R}^{H_S \times W_S \times C_T}$. The autoencoder learn to reconstruct $\mathcal{S}$ as $\mathcal{S}'$ using mean-squared error regression loss:

$$L_r(\mathcal{S}', \mathcal{S}) = \frac{1}{H_S W_S C_S} \sum_{h,w} ||s'_{h,w} - s_{h,w}||^2. \tag{4.29}$$

4.3.4.4 Keypoint Detection

To detect keypoints, an attention model M [49] is used to extract local features for relevant regions only instead of dense extraction. M is based on a small convolutional network that outputs an attention score map $\mathcal{A} \in \mathbb{R}^{C_F}$:

$$\mathcal{A} = M(\mathcal{S}). \tag{4.30}$$

. M is trained with cross-entropy classification loss after pooling the reconstructed features $\mathcal{S}'$:

$$a' = \sum_{h,w} a_{h,w} s'_{h,w} \tag{4.31}$$

where $a_{h,w} \in \mathcal{A}$ are the attention weights. Softmax cross-entropy loss is then used:

$$L_a(a', k) = -log\left(\frac{\exp\left(v_k^T a' + b_k\right)}{\sum_n \exp\left(v_n^T a' + b_n\right)}\right). \tag{4.32}$$

where v_i and b_i are the weights and biases for class i and k is the index of ground-truth class.

4.3.4.5 Local descriptors

Local descriptors are extracted at locations with high keypoints detection scores from the attention map produced by M from 4.3.4.4 taking the center of the receptive field computed. Local descriptors are the ones with reduced dimensionality obtained by the encoder T in 4.3.4.3. The local descriptors are defined as $l_{h,w} \in \mathcal{L}$ where h, w represent the location of the extracted descriptor, then the set of descriptors is further L2-normalized to give $\hat{l}_{h,w}$. The total loss is:

$$L_{tot} = L_g + \lambda L_r + \beta L_a \tag{4.33}$$

where L_{tot} is not optimized completely, since the reconstruction loss L_r and attention loss L_a makes $\mathcal{S}$ worse for localization so during back-propagation the gradients are stopped from both losses to $\mathcal{S}$ while optimizing L_g only.

4.3.4.6 Test Time

DELG uses image pyramid [42] during test time for both global and local features by applying L2-Normalization to the global feature at each of the three scales, $\{1/\sqrt{2}, 1, \sqrt{2}\}$; followed by average pooling the thee resulting global features and another L2-normalization. On the other hand, the local features use seven scales which are extracted as described in 4.3.4.5 selecting the ones with attention scores higher than a specified threshold equal to that in the final epoch of training, with a maximum of 1k local features.

4.3.5 TransVPR

This model is one of the hybrid models that extract both global and patch-level descriptors for the task of Visual Geo-localization based on visual transformers with the use of self-attention, where global descriptors are integrated by aggregating multi-level attention on task-relevant features to retrieve the top candidates and then patch-level descriptors are used for spatial verification to re-rank the retrieved candidates. The advantage of such an architecture is the fact that it does not consider all regions in the image by selecting task-relevant regions, which is one of the main challenges in VG.

4.3.5.1 Patch-Level Descriptors

First, the model extracts patch descriptors using a four-layer convolutional neural network performed on a given input image, where for each layer the output feature map is computed according to the following:

$$\mathbf{F_i} = \text{MaxPool}(\text{ReLU}(\text{BN}(\text{Conv}(\mathbf{F_{i-1}})))), \tag{4.34}$$

where $\mathbf{F_i}$ is half the spatial size of $\mathbf{F_{i-1}}$. Then patch embedding is performed on each feature map as in [50] where all feature maps have the same number of patches which are flattened but with patch area difference of a factor of 4 between each consecutive feature maps, the flattened patches are then mapped to a different dimension related to the latent embedding dimension of the transformer blocks. Then the patches at the same positions of different feature maps are concatenated to obtain raw patch-level descriptors $\mathbf{P_0}$ that will be the input tokens of multiple transformer encoder layers to give multi-level output patch tokens $\mathbf{P}$ by concatenating the patch tokens from low-level, mid-level, high-level defined as $\mathbf{P_L}, \mathbf{P_M}, \mathbf{P_H}$ in order to generate a multi-level attention map $\mathbf{A}$ by merging the three attention maps:

$$\mathbf{A} = \text{MinMaxNorm}(\sum_i \text{MinMaxNorm}(\mathbf{a_i})). \tag{4.35}$$

where $i \in L, M, H$, and $\mathbf{a_i}$ is defined as:

$$\mathbf{a_i} = \text{softmax}(\mathbf{P}\mathbf{W_i^a}) \tag{4.36}$$

where $\mathbf{W_i^a}$ maps the concatenated patch tokens to scalar with multi-level. To re-rank the retrieved candidates with local descriptors, the author found experimentally that using patch tokens from the mid-level led to the best performance, and to choose keypoints from where to extract the local descriptors, the multi-level attention map $\mathbf{A}$ is used where scores larger than a given threshold are to be selected.

4.3.5.2 Global Descriptors

To compute the global descriptors, the author suggested using the output patch tokens weighted with the attention masks obtained from 4.3.5.1 from the multiple levels as in:

$$\mathbf{G_i} = \mathbf{a_i}^T \mathbf{P_i} \tag{4.37}$$

These level-wise global descriptors are concatenated to get $\mathbf{G}^*$ and post-processed to get the final global descriptor $\mathbf{G}$:

$$\mathbf{G} = \text{L2Norm}(\text{L2Norm}(\mathbf{G}^*)\mathbf{W_g}), \tag{4.38}$$

where $\mathbf{W_g}$ is used for reducing dimensionality.

4.3.6 Patch-NetVLAD

One of the hybrid methods that follow the re-ranking approach is Patch-NetVLAD [51] that uses NetVLAD [6] descriptors first to retrieve top-k candidates, then patch-level descriptors with the multi-scale approach are used to refine the retrieved candidates.

4.3.6.1 Patch-level Global Features

Patch-NetVLAD has a similar approach to the original NetVLAD but it considers the aggregation of descriptors at a patch level which are densely sampled. So from the original NetVLAD, starting from a pre-trained CNN that extracts from an image I a feature map $F \in \mathbb{R}^{H \times W \times D}$, then patch extraction is applied from F leading to a set of patch features P_i of n_p square patches (patches can have different shapes but square shape gave better results as experimented in [51]). Then VLAD aggregation [52] and projection layers are applied on each patch feature $\mathbf{f_i} as NetVLAD$:

$$\mathbf{f}_i = f_{proj}(f_{VLAD}(P_i)) \tag{4.39}$$

For extra efficiency, IntegralVLAD similar to [53] is used after the VLAD aggregation of the features which is necessary to compute an integral feature map, followed by a depth-wise dilated convolution is used to produce the patch feature for multiple scales.

4.4 Scoring Methods

4.4.1 RANSAC

Random Sample Consensus (RANSAC) [2] is a widely used algorithm for computing scores given to the matched extracted local features which we used in the re-ranking of the retrieved candidates. The most important advantage is its robustness to outliers. An overview of how the RANSAC algorithm works for image matching:

- select four pairs of points at random from the mutual matches of both images.
- use these points to compute the homography transformation matrix between those images.
- calculate the error between estimated matches transformed by applying the homography matrix on matches from the first image and the corresponding matches from the second image.
- count the number of inliers, that are the points that have an error below a specified threshold which is a hyperparameter for this algorithm.
- repeat the first four steps for a specified number of iterations, selecting different random subsets of points each time.
- re-compute the homography using all the inliers counted by the homography with the highest number of inliers computed in the previous iterations.

The final homography computed is to be used to calculate the score that will be used to find the closest match while re-ranking.

4.4.2 Rapid Spatial Scoring

An efficient scoring method was suggested by [51] based on the translation transformation considering horizontal and vertical displacements in patches where the set of horizontal displacements is defined as:

$$x_d = \{x_i^r - x_j^q\}_{(i,j)\in\mathcal{P}} \tag{4.40}$$

for a pair of matched patches $(i, j) \in \mathcal{P}$ from two given images r and q for query and reference, similarly for the vertical displacement set represented

by y_d. Then the mean displacement in both directions is computed over all the set of matched patches as $\bar{x}_d$ and $\bar{y}_d$ for horizontal and vertical directions respectively. Which leads to the rapid spatial score definition:

$$s_{spatial} = \frac{1}{n_p} \sum_{i \in \mathcal{P}} \left(|\max_{j \in \mathcal{P}} x_{d,j}| - |x_{d,i} - \bar{x}_d| \right)^2 + \left(|\max_{j \in \mathcal{P}} y_{d,j}| - |y_{d,i} - \bar{y}_d| \right)^2 . \tag{4.41}$$

4.4.3 RRT

Reranking transformers (RRTs) [54] is a model that replaces the process of a geometric verification in the image retrieval pipeline that uses both the global and local descriptors extracted from an image. This model learns to predict a similarity score between images instead of estimating a homography, which may be very challenging. Moreover, it can be used as a scoring block for various re-ranking models. It is based on the transformer architecture [55] .

4.4.3.1 Model

For a given image I use the feature extractor of DELG [20] that outputs both global and local descriptors represented as $\mathbf{x}_g \in \mathbb{R}^{d_g}$ with $d_g = 2048$ extracted with multi-scale followed by a linear projection to reduce the dimensionality to 128, and $L = 500$ local descriptors $\mathbf{x}_l = \{\mathbf{x}_{l,i} \in \mathbb{R}^{d_l}\}_{i=1}^{L}$ with $d_l = 128$ respectively. Each local descriptor may have a coordinate tuple $\mathbf{p_{l,i}} = (u, v) \in \mathbb{R}^2$ specifying its location and a scale factor $s_{l,i}$. Similar to BERT transformer encoder [56] the input tokens are represented by the descriptors obtained by the feature extractor as:

$$\begin{aligned} \mathbf{X}(\mathbf{I}, \bar{\mathbf{I}}) := [\ & \langle \text{CLS} \rangle; f_g(\mathbf{x}_g); f_l(\mathbf{x}_{l,1}); \cdots ; f_l(\mathbf{x}_{l,L}); \\ & \langle \text{SEP} \rangle; \bar{f}_g(\bar{\mathbf{x}}_g); \bar{f}_l(\bar{\mathbf{x}}_{l,1}); \cdots ; \bar{f}_l(\bar{\mathbf{x}}_{l,L})\ ], \end{aligned} \tag{4.42}$$

where:

$$\begin{aligned} f_g(\mathbf{x}_g) &:= \mathbf{x}_g + \alpha; \\ f_l(\mathbf{x}_{l,i}) &:= \mathbf{x}_{l,i} + \varphi(\mathbf{p}_{l,i}) + \psi(s_{l,i}) + \beta \\ \bar{f}_g(\bar{\mathbf{x}}_g) &:= \bar{\mathbf{x}}_g + \bar{\alpha}; \\ \bar{f}_l(\bar{\mathbf{x}}_{l,i}) &:= \bar{\mathbf{x}}_{l,i} + \varphi(\bar{\mathbf{p}}_{l,i}) + \psi(\bar{s}_{l,i}) + \bar{\beta}. \end{aligned} \tag{4.43}$$

$\alpha, \bar{\alpha}, \beta, \bar{\beta}$ are segment embeddings to differentiate between global and local descriptors, φ is a positional embedding function as in [57], and ψ is linear

embedding that takes the scale factor as input. Each layer of the multi-layer transformer is defined as:

$$\begin{aligned} \bar{\mathbf{Z}}_{i+1} &= \text{LayerNorm}(\mathbf{Z}_i + \text{MHA}(\mathbf{Z}_i)), \\ \mathbf{Z}_{i+1} &= \text{LayerNorm}(\text{MLP}(\bar{\mathbf{Z}}_{i+1})), \\ \text{MLP}(\bar{\mathbf{Z}}_{i+1}) &= \text{ReLU}(\bar{\mathbf{Z}}_{i+1} W_1^T) W_2^T, \\ i &= 0, \cdots, C-1. \end{aligned} \tag{4.44}$$

with $\mathbf{Z}_0 = \mathbf{X}(\mathbf{I}, \bar{\mathbf{I}})$, W_1 and W_2 the parameters of the multi-layer perceptron (MLP), MHA is a multi-headed attention block and C transformer layers The model is trained with binary cross entropy loss for the objective of predicting if the pair of images are a match or not:

$$\text{E}(\mathbf{I}, \bar{\mathbf{I}}) = \text{BCE}(\text{sigmoid}(\mathbf{Z}_C^{\langle \text{CLS} \rangle} W_z^T), \mathbb{1}(\mathbf{I}, \bar{\mathbf{I}})), \tag{4.45}$$

where $\mathbf{Z}_C^{\langle \text{CLS} \rangle}$ is a feature vector corresponding to the $\langle \text{CLS} \rangle$ token and W_z^T is a linear function mapping from $\mathbf{Z}_C^{\langle \text{CLS} \rangle}$ to a logit scalar. $\mathbb{1}(\mathbf{I}, \bar{\mathbf{I}})$ indicates if both images are the same or not with values one or zero respectively. The author used 500 local descriptors from DELG which is one of the advantages of RRT not needing all the local descriptors.

4.4.4 SuperGlue

SuperGlue [58] uses a graph neural network to solve an optimal transport problem which is a relaxation of a linear assignment problem to match the local features of two images, and benefits from self and cross attention that pertains spatial relationships of keypoints and their visual appearance. It is robust against viewpoint changes and occlusion.

4.4.4.1 Architecture

SuperGlue uses SuperPoint [17] as a feature extractor to extract descriptors $\mathbf{d}_i \in \mathbb{R}^D$ and $\mathbf{p}$ the corresponding position represented by (x, y) coordinates and detection confidence c, each keypoint can be considered as a node for the graph. The first component of the architecture is an Attentional Graph Neural Network that uses the initial local features extracted by SuperPoint from both images and applies a keypoint encoder to embed the local feature tuple $(\mathbf{p}, \mathbf{d})$ to take the visual appearance and spatial positioning into consideration as:

$$^{(0)}\mathbf{x}_i = \mathbf{d}_i + \text{MLP}_{\text{enc}}(\mathbf{p_i}) \tag{4.46}$$

where ${}^{(0)}\mathbf{x}_i$ is for the initial representation at layer zero. Keypoints from both images form a single complete graph where $\mathcal{E}_{\text{self}}$ are self edges connecting keypoints within the same image and $\mathcal{E}_{\text{cross}}$ are cross edges that connects keypoints from an image to keypoints in another. The idea proposed was suggested by [59, 60] which aims to update the representations by passing messages across keypoints, for keypoints in image A:

$$ {}^{(l+1)}\mathbf{x}_i^A = {}^{(l)}\ \mathbf{x}_i^A + \text{MLP}\left(\left[{}^{(l)}\mathbf{x}_i^A || \mathbf{m}_{\mathcal{E}\to i}\right]\right) \tag{4.47} $$

where [.||.] is the notation for concatenation. After the keypoint encoder, an Attentional Aggregation is performed where self-attention and cross-attention are performed for self edges and cross edges respectively. The message is an aggregation from all keypoints from both images:

$$ \mathbf{m}_{\mathcal{E}\to i} = \sum_{j:(i,j)\in\mathcal{E}} \alpha_{ij}\mathbf{v}_j \tag{4.48} $$

where $\mathcal{E}$ is the set of both self and cross edges, $\alpha_{ij} = \text{Softmax}_j(\mathbf{q}_i^\intercal \mathbf{k}_j)$, and $\mathbf{q}, \mathbf{k}, \mathbf{v}$ are the query, key, and value computed as linear projections from the feature representation. The final matching descriptors for image A are:

$$ \mathbf{f}_i^A = \mathbf{W}.{}^{(L)}\mathbf{x}_i^A + \mathbf{b}, \tag{4.49} $$

The same goes for image B.

4.4.4.2 Optimal Matching Layer

The second component of SuperGlue results in a partial assignment matrix $\mathbf{P} \in [0,1]^{M\times N}$ by maximizing the following:

$$ \sum_{i,j} \mathbf{S}_{i,j}\mathbf{P}_{i,j}, \tag{4.50} $$

under the following constraints:

$$ \mathbf{P}\mathbf{1}_N \leq \mathbf{1}_M \quad \text{and} \quad \mathbf{P}^\top \mathbf{1}_M \leq \mathbf{1}_N \tag{4.51} $$

where M and N are the numbers of local features for a pair of images, $\mathbf{S} \in \mathbb{R}^{M\times N}$ is the score matrix obtained by computing the inner product between local features corresponding to matches from the image pair. To handle occlusion and visibility the author suggests solving an optimal transport problem [61] using Sinkhorn Algorithm [62, 63] on augmented score $\bar{\mathbf{S}}$ and

assignment matrix $\bar{\mathbf{P}}$. SuperGlue is trained in supervised manner where the set of ground truth matches are represented by $\mathcal{M} = \{(i, j)\} \subset \mathcal{A} \times \mathcal{B}$ and unmatched keypoints in both images as $\mathcal{I} \subseteq \mathcal{A}$ and $\mathcal{J} \subseteq \mathcal{B}$, where the loss used is:

$$\text{Loss} = -\sum_{(i,j)\in\mathcal{M}} \log \bar{\mathbf{P}}_{i,j} - \sum_{i\in\mathcal{I}} \log \bar{\mathbf{P}}_{i,N+1} - \sum_{j\in\mathcal{J}} \log \bar{\mathbf{P}}_{M+1,j}. \tag{4.52}$$

4.5 Other Re-ranking Methods with Built-in Scoring

4.5.1 CVNet

Correlation Verification Network [21] (CVNet) is a hybrid approach for image retrieval with its own scoring method that substitutes geometric verification component for re-ranking models where a scoring method like RANSAC [2] is applied on the extracted local features from the retrieved candidates obtained based on similarity scores computed from the extracted global descriptors. This method benefits from the 4D convolution that compresses feature correlation from image pair into an image similarity, it also uses cross-scale correlation with a single inference to replace multi-scale inference. Moreover, CVNet uses curriculum learning with hard negative mining and Hide and Seek technique [64] that handles hard samples like images with occlusion.

4.5.1.1 CVNet Global

The Global backbone network of CVNet includes two ResNet networks [36] f and $\bar{f}$ pre-trained on ImageNet [30] where each is used to extract global descriptor $\mathbf{d}_g \in \mathbb{R}^{C_g}$ where $C_g = 2048$ from an image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, the use of two networks is a structure inspired from [65]. Each ResNet network consists of four ResNet blocks. The whole global architecture is composed of two main paths trained jointly. The first path takes the global descriptor extracted from a query image and applies GeM Pooling [7] and a whitening layer [66] followed by L2-Normalization to result in the query descriptor $\mathbf{d}_g^q$ which will be trained for classification using the CurriculurFace-margined classification loss [67] $\mathcal{L}_{cls}$. The second path uses the momentum network $\bar{f}$ to extract positive momentum global descriptor $\bar{\mathbf{d}}_g^p$ from a sampled positive image $\mathbf{I}_p$ with the same label as the query image $\mathbf{I}_q$, the same processing is applied with GeM Pooling, whitening layer, and L2-Normalization, the

resulting global descriptor is enqueued to a queue $\mathbf{Q}$, this is done iteratively for each sample and while dequeuing the last element. A CurriculurFace-margined momentum contrastive loss $\mathcal{L}_{con}$ is used without being updated by the optimizer but with a momentum update. The total loss is obtained by a weighted sum of the stated losses:

$$\mathcal{L}_g = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{con}\mathcal{L}_{con}. \tag{4.53}$$

4.5.1.2 CVNet-Rerank

Local feature maps $F \in \mathbb{R}^{C_l \times W_l \times H_l}$ are extracted using the third ResNet Block f_3 from the query and candidate image pairs to predict the similarity score $s_l^{q,k}$ where q and k are for query and key, the similarity score is obtained using a classifier composed of 4D convolutional layers. Moreover, to build a scale indifferent features, the author suggests replacing standard image pyramid [42] with feature pyramid [68] $\{\mathbf{F}^s\}_{s=1}^{S}$ by resizing the feature map $\mathbf{F}$ with multiple scales $S = \{1/2, 1/\sqrt{2}, 1\}$ followed by scale-wise convolutional layer to produce a feature map with channel number $C_l' = 256$. Then using the computed feature maps from the query and key images the cross-scale correlation set $\mathbf{C}_{qk}^{s_q,s_k}$ is obtained by applying ReLU to the cosine similarity of local features taken from different pixel positions from the feature maps. A sequence of 4D convolutional blocks followed by a global average pooling layer and a 2-layer MLP is applied to output a binary class logit. The author suggested adopting a center-pivot 4D convolution [69] to be more computationally efficient. The final score used for re-ranking is the sum of cosine similarity of the global descriptors with the weighted output score of the re-ranking network. The re-ranking network is trained with total loss:

$$\mathcal{L}_r = (\mathcal{L}_r^{qp} + \mathcal{L}_r^{pq} + \mathcal{L}_r^{qn} + \mathcal{L}_r^{nq})/4. \tag{4.54}$$

where:

$$\mathcal{L}_r^{qk} = \mathbf{CE}(\mathbf{Softmax}(\mathbf{Z}_{qk}), \mathbb{1}_q^k). \tag{4.55}$$

To build a more robust re-ranking network, the author makes use of hard negative mining where negative images with the highest global descriptor matching score are sampled, another strategy used is the Hide-and-Seek strategy that augments an image by deactivating some grids with a certain probability to synthesize the occlusion concept. Both mentioned strategies are learned in curriculum manner where the rate of selecting hard negatives or applying the augmentation increases as the learning progresses.

4.5.2 LoFTR

LoFTR [70] is a method that consists of local feature extraction and matching while skipping the detection phase that existing methods like [71, 72] use. The detection-free approach avoids the challenges associated with this step from being able to extract repeatable interest points. Moreover, another issue is that images with low-texture regions are challenging when using dense features because they may lack context locally, so the global context is crucial to be considered, so a coarse feature map is extracted from a CNN backbone and used to simulate a larger receptive field which is then updated at a finer level by a fine feature map after finding the matches. The local features are updated by a Transformer using self and cross-attention to make them conditioned on both input images. The overview of the pipeline is shown in Figure 4.6.

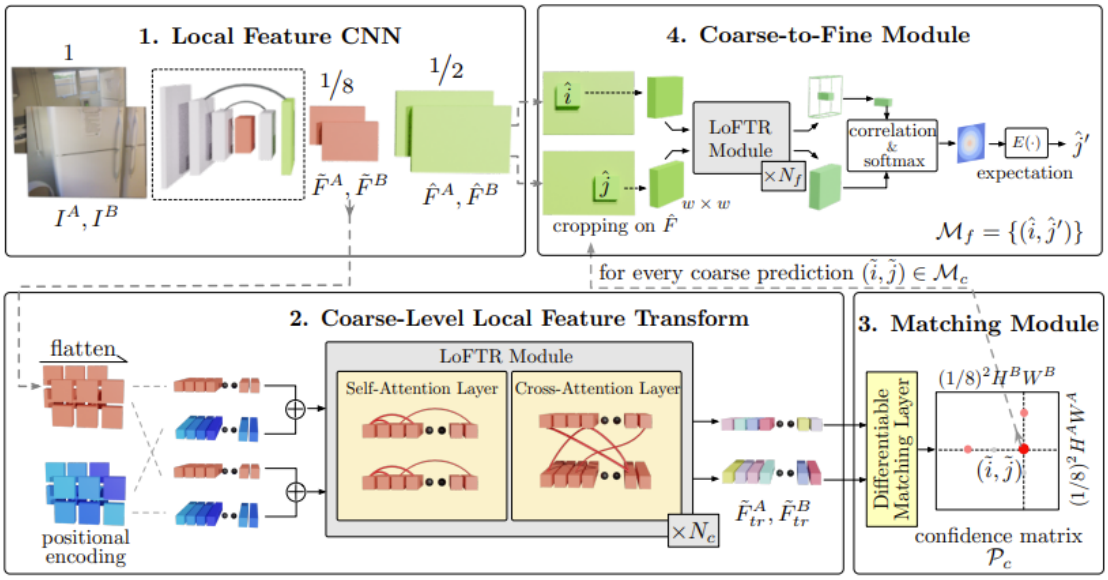

Figure 4.6: **LoFTR pipeline overview.** Image from [70]

4.5.2.1 Local Feature Extraction

First, a CNN based on FPN [73] is used to extract a pair of coarse-level feature maps (1/8 the original dimension) $\tilde{F}^A$ and $\tilde{F}^B$ and a pair of fine-level feature maps (1/2 the original dimension $\hat{F}^A$ and $\hat{F}^B$ from an input image I^A and I^B.

4.5.2.2 Coarse-Level Local Feature Transform

The coarse-level local features are flattened and a positional encoding is added to both feature maps to account for the position factor, which is followed by a LoFTR Module that performs Self-Attention Layer and Cross-Attention Layer while substituting the normal Dot-Product Attention [55] with a Linear Attention [74] that is more computationally efficient. This phase outputs a transformed feature maps $\tilde{F}^A_{tr}$ and $\tilde{F}^B_{tr}$.

4.5.2.3 Matching Module at Coarse Level

After computing the score matrix $\mathcal{S}$ between the transformed features as:

$$\mathcal{S}(i,j) = \frac{1}{\tau} \cdot \langle \tilde{F}^A_{tr}(i), \tilde{F}^B_{tr}(j) \rangle, \tag{4.56}$$

two approaches can be used to perform matching, the first approach is described in SuperGlue [58] by using an optimal transport layer where $-\mathcal{S}$ is the cost matrix for the partial assignment problem, and the second approach is using dual-softmax operator [75, 76] on $\mathcal{S}$, where we then obtain the matching probability $\mathcal{P}_c$ as:

$$\mathcal{P}_c(i,j) = \text{softmax}(\mathcal{S}(i,\cdot))_j \cdot \text{softmax}(\mathcal{S}(\cdot,j))_i. \tag{4.57}$$

Matches are selected according to $\mathcal{P}_c$ where their value is higher than a specified threshold θ_c and constraining the matches to be mutual.

4.5.2.4 Coarse-to-Fine Module

To update the coarse matches' locations into the original image size the coarse-to-fine module finds the location $(\hat{i}, \hat{j})$ in the fine feature maps $\hat{F}^A$ and $\hat{F}^B$ corresponding to $\tilde{i}, \tilde{j}$ then crop two windows sets and apply LoFTR transformer module as in 4.5.2.2 into $\hat{F}^A(\hat{i})$ and $\hat{F}^B(\hat{j})$ where $\hat{i}$ and $\hat{j}$ are their centers respectively, then correlating the center vector from the first transformed feature maps to all vectors in the other and getting the final location $\hat{j}'$ according to the calculated expectation over the probability distribution of the previous correlation to produce the set of fine-level matches $\{(\hat{i}, \hat{j}')\}$.

4.6 Results

We provide a benchmark to evaluate the effect of the re-ranking pipeline by comparing several combinations of feature extraction models and scoring

methods that use nearest neighbor search to match the extracted features to re-rank the top 100 retrieved candidates provided by CosPlace [5] against the baseline. The choice of feature extractor, scoring methods, and their corresponding best configuration of hyperparameters are as suggested by their author except for the backbones where we chose ResNet-50 over ResNet-100. Furthermore, the feature extractors used are not all designed for re-ranking. The characteristics of the feature extractors are summarized in 4.2.

We also mention that DELG uses an affine version of RANSAC with a limited number of transformation restricted to estimating a homography with 6 parameters whereas all other models using RANSAC uses the general version with 8 parameters. Also, Patch-NetVLAD uses RANSAC on each of its scales and averages the score.

The metric used in all experiments is Recall@N (R@N) which was used in [6, 77] which represents the percentage of queries having a correct prediction in the top-N predicted candidates within a specific threshold. We select a value of 100 for N and 25 meters as a threshold.

4.6.1 Quantitative Evaluation

Table 4.1 shows the results achieved by the re-ranking models on the datasets described in 3. It is observed that re-ranking models have a significant improvement on datasets with both night and day domain queries with respect to the baseline. For datasets with day queries, there is no clear winner. For example, on SF-XL test v1 with SuperGlue implementation having SuperPoint as a local extractor achieves the highest R@1, R@5, and R@10 with 11.9% improvement over the baseline for R@1 and with DELG immediately behind it with 0.1% difference, while on SF-XL test v2, R2D2 with RANSAC increases the R@1 by 15.1% with 0.1% higher than D2-Net with RANSAC. On datasets with night queries, SuperPoint with SuperGlue achieves the highest R@1 on Tokyo Night and SF-XL test night with 15.2% and 9.2% increase over the baseline respectively, and DELG achieves the highest R@1 on SVOX Night with 28.5% higher R@1 than the baseline, while CVNET achieving the highest R@5 and R@10 on all night datasets. On average, DELG shows the highest improvements in R@1 for both domains with 8.3% and 17% increases on both day and night domains respectively, followed by SuperGlue and LoFTR. Additionally, comparing RRT against RANSAC for feature matching with local features extracted by DELG shows a great improvement using RANSAC over RRT, especially for night datasets with the highest difference on SVOX Night with 15% R@1 improvement. Also

comparing Rapid Scoring against RANSAC with Patch-Netvlad as a local feature extractor shows that the approach of rapid scoring performs worse than the baseline for all datasets except SF-XL test v2 with an increase of 1%. This shows how good RANSAC is compared to the other matching methods considered. SuperGlue on the other hand seems to have a very good performance too in general competing with RANSAC in performance. Finally, we notice how challenging the SF-XL test night dataset with the best combination achieving 33% R@1 which compared to other night datasets is significant which might mean that the illumination challenge is not the only reason for the low results, but the images themselves with the high viewpoint shifts, and others focusing on signboards and some images with decorations and lights are challenging for these models.

Regarding the computational cost of these methods, the trade-off between the performance and the efficiency can be observed in 4.7, where we show the R@1 for the re-ranking models with respect to the time it required to re-rank the top-100 candidates for a single query from SF-XL test night dataset not considering the time it takes CosPlace to retrieve these candidates which are computed by summing the time it takes to extract feature from the query and the 100 candidates and the time to match and score each of the candidates with the query. We notice that although DELG had the best performance overall, the time it requires is very high compared to other high-performing methods like SuperGlue and LoFTR which takes less than half the time DELG requires, they are still not feasible for applications that require real-time retrieval like autonomous driving for example which can be handled by changing the number of candidates considered while still maintaining a noticeable increase in performance. We can see in 4.8 the effect of changing the number of retrieved candidates to re-rank (K) on the R@1 against the upper-bound on SF-XL test v1 and SF-XL test night within 25 meters threshold. In general, as K increases the R@1 increase reaching an asymptote which is reached quicker for night domain as we see for SF-XL test night. So we can decrease the number of candidates to make the re-ranking faster while not sacrificing the performance much, or even in case the application does not require very high accuracy, a lower number of candidates might be more suitable.

4.6.2 Qualitative Evaluation

Most of the re-ranking models have a keypoint detection phase, so they use their local features to perform matching using a homography estimation using

Features Extractor	Features Matching	Tokyo Night R@100 = 96.2			SVOX Night R@100 = 90.3			SF-XL test v1 R@100 = 92.5			SF-XL test v2 R@100 = 97.7			SF-XL test night R@100 = 41.6		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
-	-	80.0	88.6	91.4	51.6	68.8	76.1	76.7	82.5	85.6	89.0	95.3	96.3	23.8	29.0	31.5
SuperPoint	SuperGlue	**95.2**	95.2	95.2	77.9	**85.2**	**86.5**	**88.6**	**91.6**	**91.9**	92.8	96.7	**97.7**	**33.0**	38.0	39.1
D2-net	RANSAC	92.4	**96.2**	**96.2**	78.9	85.1	86.4	87.5	90.3	90.8	94.0	96.3	97.0	32.6	38.2	39.5
R2D2	RANSAC	86.7	90.5	92.4	72.5	80.7	82.9	85.1	88.2	89.6	**94.1**	**96.8**	96.8	26.2	32.2	33.9
DELG	RANSAC	94.3	95.2	**96.2**	**80.1**	84.1	86.0	88.5	91.2	91.5	93.8	96.2	97.0	32.2	37.6	39.2
DELG	RRT	84.8	94.3	95.2	66.3	81.7	85.7	85.3	89.6	90.4	88.6	96.0	97.2	27.3	35.6	38.6
Patch-NetVLAD	RANSAC	90.5	94.3	94.3	67.2	80.6	83.6	77.0	84.7	87.0	91.0	95.2	96.2	31.8	37.3	38.4
Patch-NetVLAD	Rapid Scoring	73.3	87.6	92.4	42.2	66.3	73.1	69.3	80.3	84.1	90.0	94.6	95.8	21.7	31.3	35.4
TransVPR	RANSAC	88.6	95.2	95.2	63.8	79.2	83.2	84.0	87.6	89.1	92.5	96.2	96.7	27.3	34.3	36.7
	LoFTR	93.3	95.2	95.2	80.0	84.0	85.3	87.9	89.8	90.7	93.3	96.3	97.2	32.6	37.6	38.2
	CVNet	94.3	**96.2**	**96.2**	74.6	**85.2**	**86.5**	84.8	91.0	91.6	88.0	95.8	97.0	31.5	**39.3**	**39.9**

Table 4.1: **Results of the re-ranking methods against the baseline with CosPlace.** The first 100 candidates retrieved by CosPlace were used to be re-ranked by the re-ranking methods. The upper bound defined as Recall@100 is shown under the dataset name for each dataset.

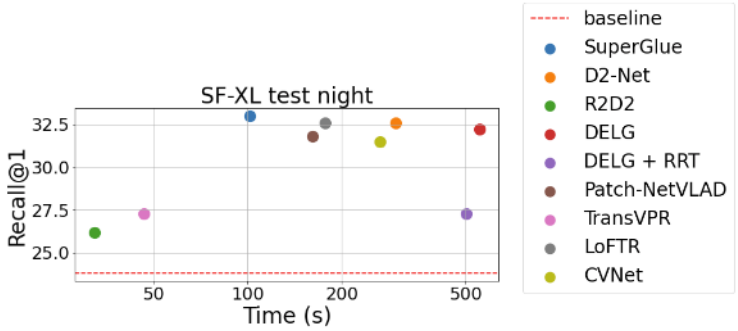

Figure 4.7: **Recall@1 and Time.** The Recall@1 is considered for the re-ranking methods on SF-XL test night dataset. The time is for the latency it takes the re-ranking method to re-rank the 100 candidates for a single query considering feature extraction, matching, and scoring.

RANSAC for example. For this reason, we visualize the keypoints detected by these models except for CVNet which has no keypoint detection phase, also for LoFTR which is a detector-free model, we try to visualize what points it matches given two images which is not exactly a keypoint detection stage but we visualize the matches chosen which are dependent on the image pair. We can observe from figure 4.9 that D2-Net and SuperPoint extract keypoints without avoiding irrelevant objects like sky, pedestrians, and cars, even for the night query image, whereas R2D2 is better at avoiding the sky, especially for the night domain, but still it does not ignore pedestrians and cars, then there is DELG that has a significant improvement over the other models in detecting interest points avoiding unnecessary objects much

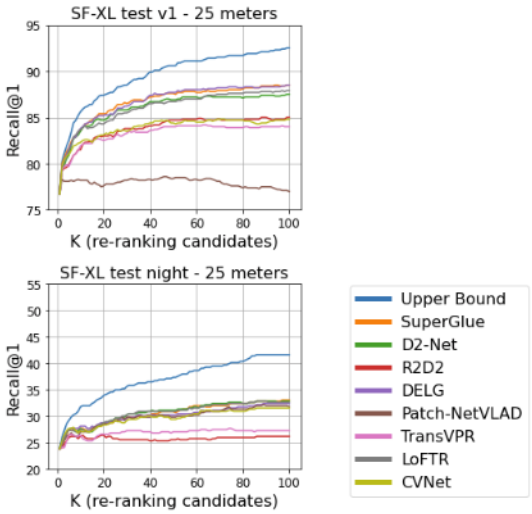

Figure 4.8: **Plot for the effect of candidates number on the recall.** DELG and Patch-NetVLAD with RANSAC are considered.

better and focusing on the important regions that are useful for the VG task which might be explained due to the fact that not all models were trained for the VG task (Table 4.2), as DELG is for VG, other models are not. LoFTR also behaves accurately with respect toother models (Fig. 4.10) and we can see the difference of the chosen points for matching for the same query image when the reference image changes. TransVPR does keypoint detection depending on the image size by dividing the height and width by patch size which gives the number of patches for both dimensions N_h and N_w respectively and selecting the equidistant keypoints equal to the number of patches $N_h \times N_w$ (Fig. 4.11. We also show the matching between the same image pair with a query in both domains using the best scoring methods compared to mutual matches (Fig. 4.12). Moreover, a sample of predictions on SF-XL test night was given using the best-performing re-ranking methods showing their performance against the baseline (Fig. 4.13).

Model	Descriptors size (num. × dim.)	Backbone	Designed for re-ranking	Sparse Keypoints
DELG	1000 x 128	ResNet-50	✓	✓
Patch-NetVLAD	2826 x 4096	VGG-16	✓	✗
TransVPR	522 x 256	Custom CNN+transformer	✓	✓
R2D2	4126 x 128	custom L2-Net[44]	✗	✓
D2Net	2775 x 512	VGG-16	✗	✓
SuperPoint	1034 x 256	custom VGG	✗	✓

Table 4.2: **Characteristics of feature extractors used for re-ranking.** The number of descriptors for each method may vary depending on the image resolution or the visual content.

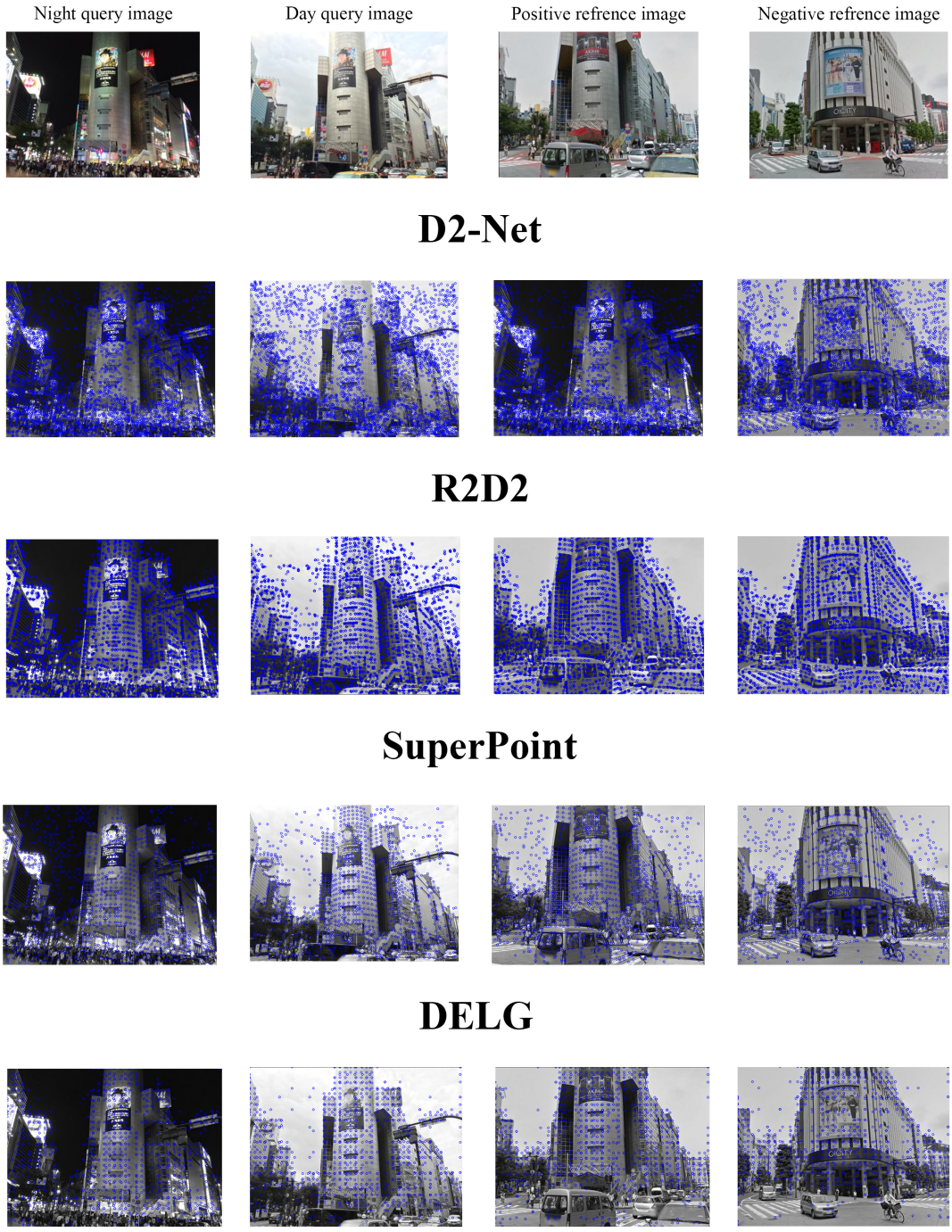

Figure 4.9: **Visualization of the keypoint detection phase of the re-ranking methods.**

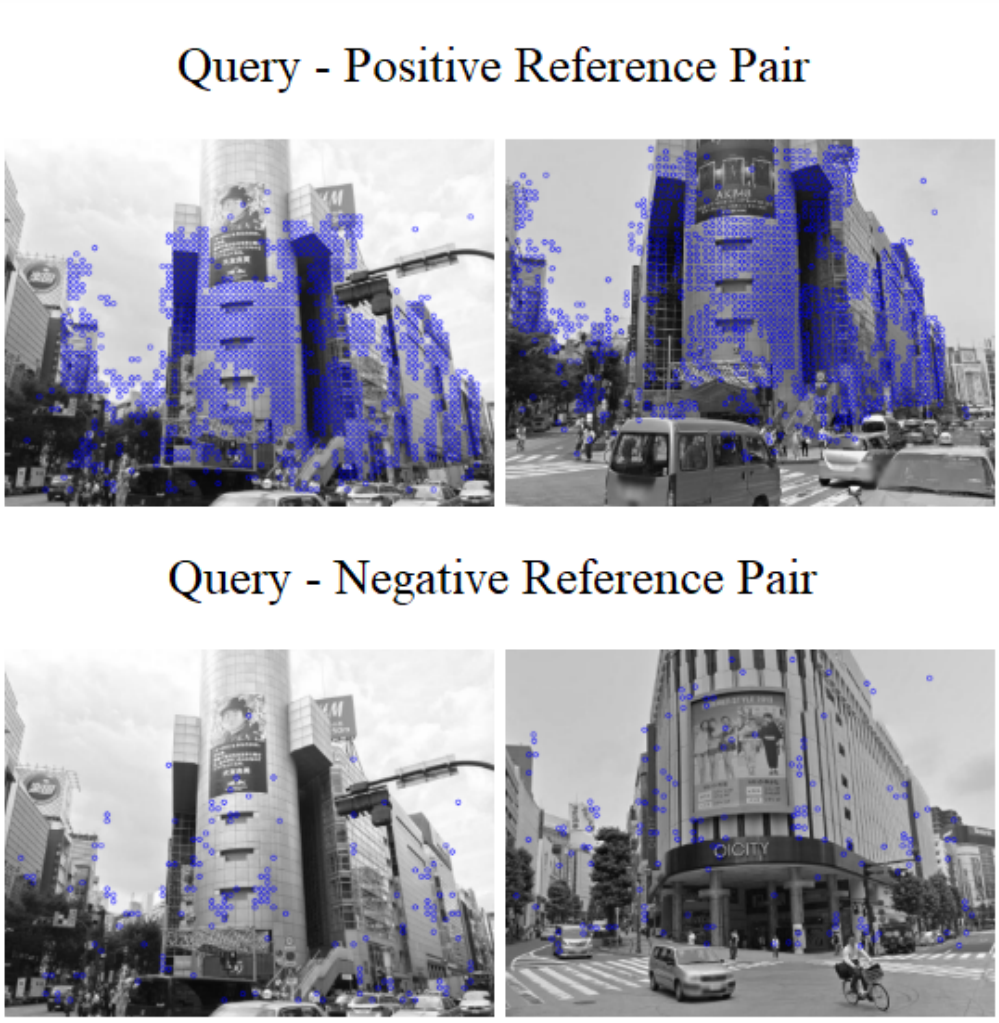

Figure 4.10: **LoFTR keypoints.** LoFTR is a detector-free model, so we show what points of interest are being considered for matching since it is dependent on image pair.

Figure 4.11: **Visualization of the keypoints used by TransVPR.**

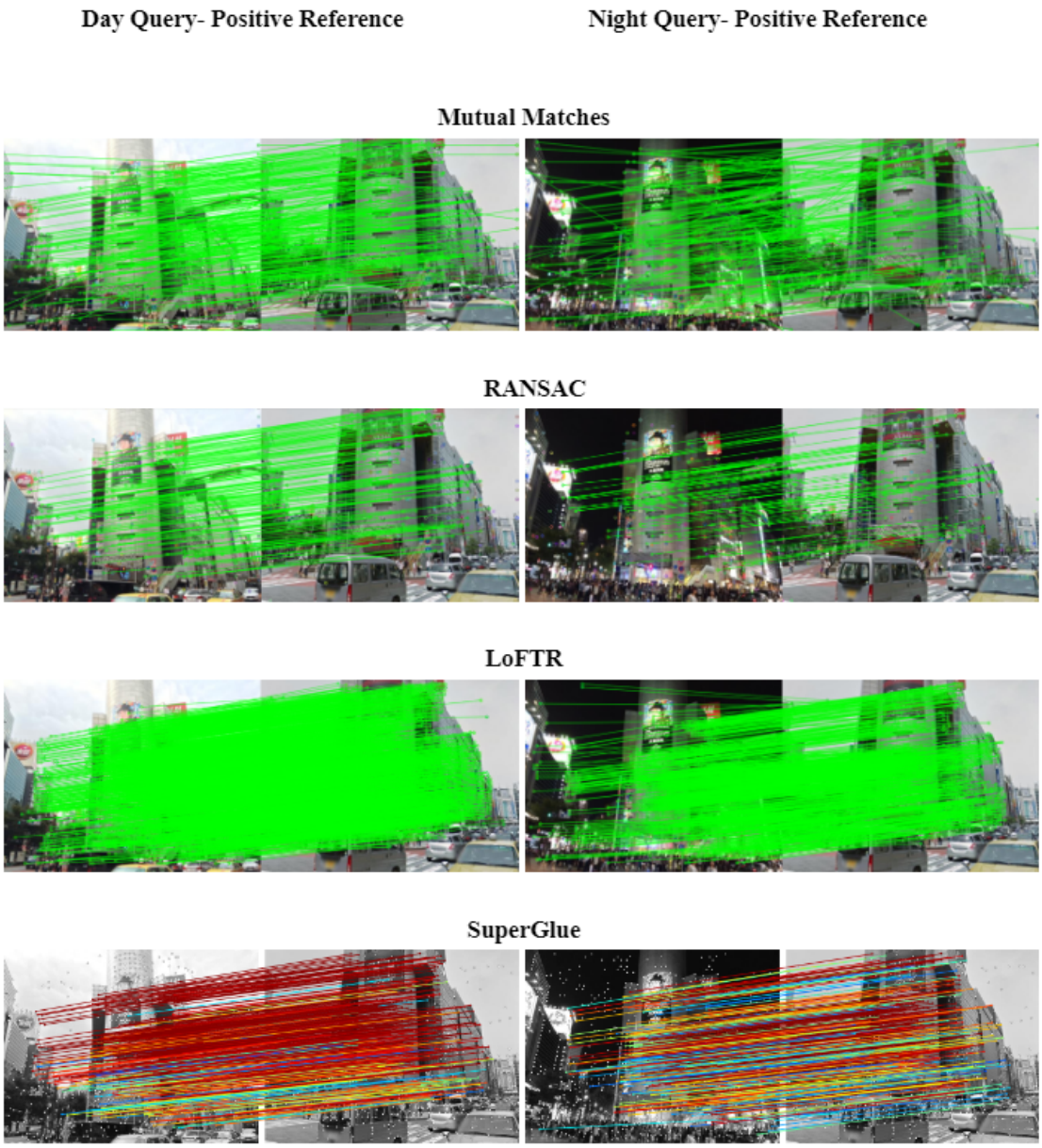

Figure 4.12: **Matching visualization using different methods for night and day queries.**

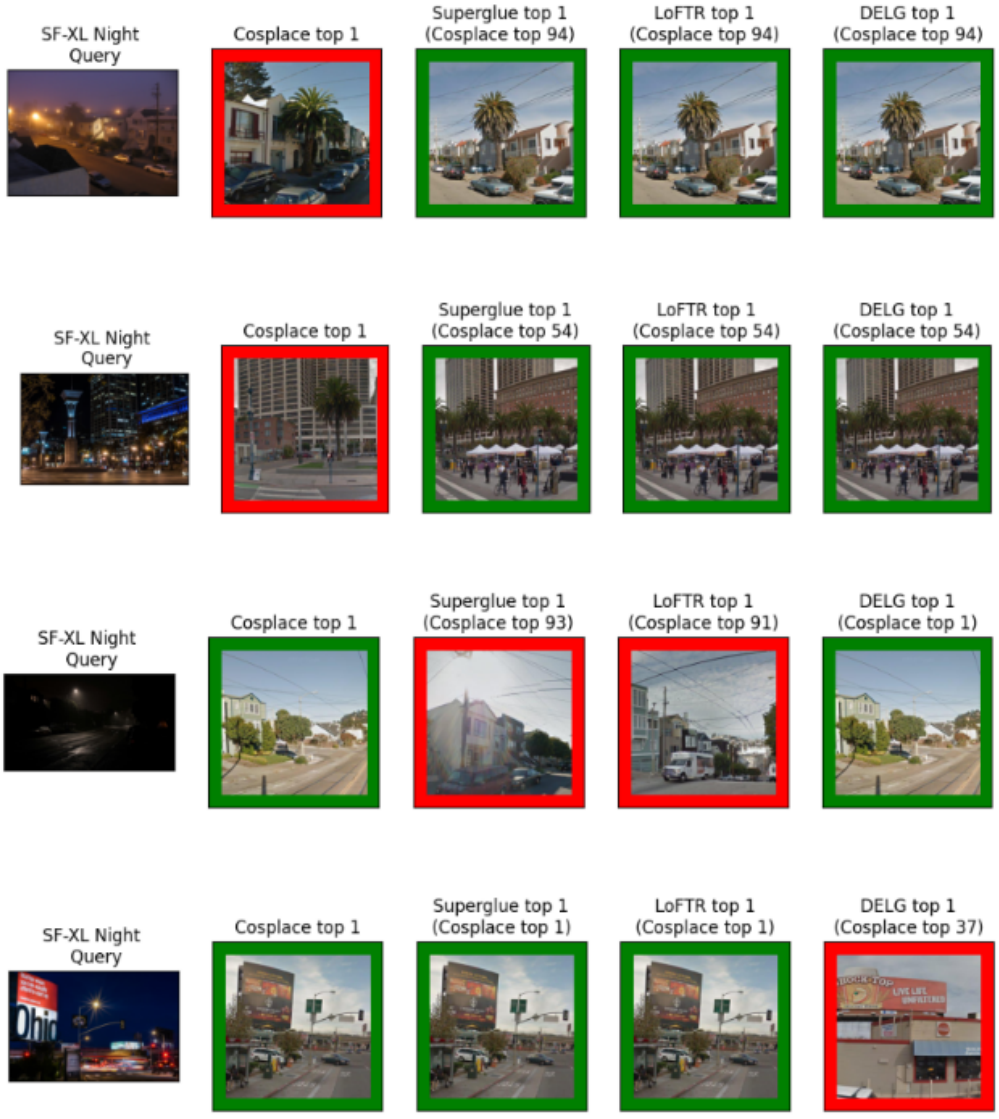

Figure 4.13: **Predictions example for the best re-ranking methods.**

Chapter 5

Conclusions and future works

The main purpose of this thesis was to provide a benchmark that evaluates the effects of applying a re-ranking approach on the task of visual geo-localization (VG), especially on a major challenge that some applications require which is the domain shift problem, specifically the night-day domain shift. Starting from an image retrieval method, we experimented with various re-ranking methods that not necessarily are trained for the VG task, but might also be trained for image matching or other similar tasks, and analyzed their performance and efficiency for various challenging datasets, especially on night domain which showed a significant improvement. Moreover, we introduced a new clean dataset SF-XL test night having night queries which proved to be challenging for the suggested methods due to other challenges in addition to the illumination change like some lighting effects from decorations or signboards, and this helps for further experimenting for challenges that might be faced in real life applications. Some additional work might be used to extend what has been evaluated in this thesis:

- try to use the global features provided by the best retrieval method instead of those used by re-ranking models that perform scoring for both global and local features (like DELG).
- provide an alternative scoring method that estimates homography which is challenging since it is hard to choose pairs of keypoints that are co-planar which is important for estimating the homography matrix.
- Use Generative Adversarial networks that can help with homography

estimation to help with scoring [78]

Bibliography

[1] Jeff Johnson, Matthijs Douze, and Hervé Jégou. *Billion-scale similarity search with GPUs*. 2017. arXiv: 1702.08734 [cs.CV].

[2] Martin A Fischler and Robert C Bolles. «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography». In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[3] Aude Oliva and Antonio Torralba. «Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope». In: *International Journal of Computer Vision* 42 (May 2001), pp. 145–175. DOI: 10.1023/A:1011139631724.

[4] N. Dalal and B. Triggs. «Histograms of oriented gradients for human detection». In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[5] Gabriele Berton, Carlo Masone, and Barbara Caputo. «Rethinking Visual Geo-localization for Large-Scale Applications». In: *CVPR*. June 2022.

[6] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. «NetVLAD: CNN Architecture for Weakly Supervised Place Recognition». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1437–1451. DOI: 10.1109/TPAMI.2017.2711011.

[7] F. Radenović, G. Tolias, and O. Chum. «Fine-tuning CNN Image Retrieval with No Human Annotation». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[8] Krystian Mikolajczyk and Cordelia Schmid. «An Affine Invariant Interest Point Detector». In: *Computer Vision — ECCV 2002*. Ed. by Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 128–142. ISBN: 978-3-540-47969-7.

[9] J Matas, O Chum, M Urban, and T Pajdla. «Robust wide-baseline stereo from maximally stable extremal regions». In: *Image and Vision Computing* 22.10 (2004). British Machine Vision Computing 2002, pp. 761–767. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2004.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0262885604000435.

[10] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. *LIFT: Learned Invariant Feature Transform*. 2016. arXiv: 1603.09114 [cs.CV].

[11] David Lowe. «Distinctive Image Features from Scale-Invariant Keypoints». In: *International Journal of Computer Vision* 60 (Nov. 2004), pp. 91–. DOI: 10.1023/B:VISI.0000029664.99615.94.

[12] David G. Lowe. «Distinctive Image Features from Scale-Invariant Keypoints». In: *Int. J. Comput. Vision* 60.2 (2004), pp. 91–110. URL: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. «Speeded-up robust features (SURF)». In: *Computer Vision and Image Understanding* 110 (June 2008), pp. 346–359. DOI: 10.1016/j.cviu.2007.09.014.

[14] R. Arandjelović and Andrew Zisserman. «Three things everyone should know to improve object retrieval». In: 2012, pp. 2911–2918.

[15] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. «R2D2: Repeatable and Reliable Detector and Descriptor». In: *NeurIPS*. 2019.

[16] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. «D2-Net: A Trainable CNN for Joint Detection and Description of Local Features». In: *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[17] Tomasz Malisiewicz Daniel DeTone and Andrew Rabinovich. «SuperPoint: Self-Supervised Interest Point Detection and Description». In: *Computer Vision and Pattern Recognition Workshop*. 2018.

[18] Tobias Weyand, Ilya Kostrikov, and James Philbin. «PlaNet - Photo Geolocation with Convolutional Neural Networks». In: *European Conference on Computer Vision*. 2016.

[19] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. «CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps». In: *ECCV*. 2018.

[20] B. Cao, A. Araujo, and J. Sim. «Unifying Deep Local and Global Features for Image Search». In: *European Conference on Computer Vision*. Springer Int. Publishing, 2020, pp. 726–743. ISBN: 978-3-030-58564-8.

[21] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. «Correlation Verification for Image Retrieval». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 5374–5384.

[22] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. «TransVPR: Transformer-Based Place Recognition With Multi-Level Attention Aggregation». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 13648–13657.

[23] Gabriele Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. «Adaptive-Attentive Geolocalization From Few Queries: A Hybrid Approach». In: *IEEE Winter Conference on Applications of Computer Vision*. Jan. 2021, pp. 2918–2927.

[24] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. «Night-to-day image translation for retrieval-based localization». In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5958–5964.

[25] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. *ComboGAN: Unrestrained Scalability for Image Domain Translation*. 2017. arXiv: 1712.06909 [cs.CV].

[26] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. «24/7 Place Recognition by View Synthesis». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.2 (2018), pp. 257–271.

[27] Horia Porav, Will Maddern, and Paul Newman. *Adversarial Training for Adverse Conditions: Robust Metric Localisation using Appearance Transfer*. 2018. arXiv: 1803.03341 [cs.CV].

[28] Mingxing Tan and Quoc Le. «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks». In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[29] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. arXiv: 1801.04381 [cs.CV].

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*. 2015. arXiv: 1409.0575 [cs.CV].

[31] *Day-Night-Classifier*. URL: https://github.com/jayeshsaita/Day-Night-Classifier.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. «1 Year, 1000km: The Oxford RobotCar Dataset». In: *The International Journal of Robotics Research* (2017).

[33] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. «City-scale landmark identification on mobile devices». In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 737–744. DOI: 10.1109/CVPR.2011.5995610.

[34] A. Torii, Hajime Taira, Josef Sivic, M. Pollefeys, M. Okutomi, T. Pajdla, and Torsten Sattler. «Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization?» In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2021), pp. 814–829.

[35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. «CosFace: Large Margin Cosine Loss for Deep Face Recognition». In: *IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 5265–5274.

[36] K. He, X. Zhang, S. Ren, and J. Sun. «Deep Residual Learning for Image Recognition». In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[37] Karen Simonyan and Andrew Zisserman. «Very Deep Convolutional Networks for Large-Scale Image Recognition». In: *International Conference on Learning Representations*. 2015.

[38] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*. 2016. arXiv: 1609.05158 [cs.CV].

[39] Zhengqi Li and Noah Snavely. «MegaDepth: Learning Single-View Depth Prediction from Internet Photos». In: *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[40] Daniel Ponsa Vassileios Balntas Edgar Riba and Krystian Mikolajczyk. «Learning local feature descriptors with triplets and shallow convolutional neural networks». In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Edwin R. Hancock Richard C. Wilson and William A. P. Smith. BMVA Press, Sept. 2016, pp. 119.1–119.11. ISBN: 1-901725-59-6. DOI: 10.5244/C.30.119. URL: https://dx.doi.org/10.5244/C.30.119.

[41] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. *Working hard to know your neighbor's margins: Local descriptor learning loss*. 2018. arXiv: 1705.10872 [cs.CV].

[42] Edward Adelson, Charles Anderson, James Bergen, Peter Burt, and Joan Ogden. «Pyramid Methods in Image Processing». In: *RCA Eng.* 29 (Nov. 1983).

[43] Tony Lindeberg. «Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales». In: *Journal of Applied Statistics* 21 (Sept. 1994), pp. 224–270. DOI: 10.1080/757582976.

[44] Yurun Tian, Bin Fan, and Fuchao Wu. «L2-net: Deep learning of discriminative patch descriptor in euclidean space». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 661–669.

[45] Kun He, Yan Lu, and Stan Sclaroff. *Local Descriptors Optimized for Average Precision*. 2018. arXiv: 1804.05312 [cs.CV].

[46] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. *Deep Image Retrieval: Learning global representations for image search*. 2016. arXiv: 1604.01325 [cs.CV].

[47] Jiankang Deng, J. Guo, and S. Zafeiriou. «ArcFace: Additive Angular Margin Loss for Deep Face Recognition». In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4685–4694.

[48] Geoffrey E. Hinton. «Connectionist learning procedures». In: *Artificial Intelligence* 40.1 (1989), pp. 185–234. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(89)90049-0. URL: https://www.sciencedirect.com/science/article/pii/0004370289900490.

[49] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. «Large-Scale Image Retrieval with Attentive Deep Local Features». In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 3476–3485. DOI: 10.1109/ICCV.2017.374.

[50] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». In: *ArXiv* abs/2010.11929 (2021).

[51] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. «Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition». In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14141–14152.

[52] Hervé Jégou, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. «Aggregating Local Image Descriptors into Compact Codes». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (Dec. 2011). DOI: 10.1109/TPAMI.2011.235.

[53] Franklin C. Crow. «Summed-Area Tables for Texture Mapping». In: *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '84. New York, NY, USA: Association for Computing Machinery, 1984, pp. 207–212. ISBN: 0897911385. DOI: 10.1145/800031.808600. URL: https://doi.org/10.1145/800031.808600.

[54] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. «Instance-level Image Retrieval using Reranking Transformers». In: *IEEE International Conference on Computer Vision*. 2021.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. «Attention is All You Need». In: NIPS'17. Long Beach, California, USA, 2017, pp. 6000–6010.

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[57] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].

[58] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. «SuperGlue: Learning Feature Matching with Graph Neural Networks». In: *CVPR*. 2020. URL: https://arxiv.org/abs/1911.11763.

[59] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: 1704.01212 [cs.LG].

[60] Peter W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. 2018. arXiv: 1806.01261 [cs.LG].

[61] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*. 2020. arXiv: 1803.00567 [stat.ML].

[62] Paul Knopp and Richard Sinkhorn. «Concerning nonnegative matrices and doubly stochastic matrices.» In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348.

[63] Marco Cuturi. «Sinkhorn Distances: Lightspeed Computation of Optimal Transport». In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

[64] Krishna Kumar Singh and Yong Jae Lee. «Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-Supervised Object and Action Localization». In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 3544–3553.

[65] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. *Momentum Contrast for Unsupervised Visual Representation Learning*. 2020. arXiv: 1911.05722 [cs.CV].

[66] Albert Gordoa, José A. Rodríguez-Serrano, Florent Perronnin, and Ernest Valveny. «Leveraging category-level labels for instance-level image retrieval». In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3045–3052. DOI: 10.1109/CVPR.2012.6248035.

[67] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. *CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition*. 2020. arXiv: 2004.00288 [cs.CV].

[68] Juhong Min and Minsu Cho. *Convolutional Hough Matching Networks*. 2021. arXiv: 2103.16831 [cs.CV].

[69] Juhong Min, Dahyun Kang, and Minsu Cho. *Hypercorrelation Squeeze for Few-Shot Segmentation*. 2021. arXiv: 2104.01538 [cs.CV].

[70] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. «LoFTR: Detector-Free Local Feature Matching with Transformers». In: *CVPR* (2021).

[71] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. «Lift: Learned invariant feature transform». In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, pp. 467–483.

[72] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. «Toward geometric deep slam». In: *arXiv preprint arXiv:1707.07410* (2017).

[73] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. *Feature Pyramid Networks for Object Detection*. 2017. arXiv: 1612.03144 [cs.CV].

[74] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention*. 2020. arXiv: 2006.16236 [cs.LG].

[75] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. *Neighbourhood Consensus Networks*. 2018. arXiv: 1810.10510 [cs.CV].

[76] Michał J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. *DISK: Learning local features with policy gradient*. 2020. arXiv: 2006.13566 [cs.CV].

[77] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. «Deep Visual Geo-localization Benchmark». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022.

[78] Mingbo Hong, Yuhang Lu, Nianjin Ye, Chunyu Lin, Qijun Zhao, and Shuaicheng Liu. *Unsupervised Homography Estimation with Coplanarity-Aware GAN*. 2022. arXiv: 2205.03821 [cs.CV].