UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Learning Discriminative Context-Aware Keypoints Representations for Resolving Ambiguous Matches

Author:
Ostap VINIAVSKYI

Supervisor:
PhD James PRITTS

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2021

Declaration of Authorship

I, Ostap VINIAVSKYI, declare that this thesis titled, "Learning Discriminative Context-Aware Keypoints Representations for Resolving Ambiguous Matches" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“Mistakes are, after all, the foundations of truth, and if a man does not know what a thing is, it is at least an increase in knowledge if he knows what it is not.”

Carl Jung

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Learning Discriminative Context-Aware Keypoints Representations for Resolving Ambiguous Matches

by Ostap VINIAVSKYI

Abstract

In the feature matching problem, local keypoint representations are often not sufficiently distinctive to disambiguate repetitive textures. State-of-the-art matching pipelines encode global information and embed context into keypoint descriptors to resolve this issue. In this thesis, we evaluate the failure modes of the state-of-the-art method for image matching. We identify the problem that including global context to keypoint representations can sometimes eliminate their distinctiveness. We propose to enhance the learning of the state-of-the-art pipeline by adding a metric learning component to its objective function. By learning more distinctive global context-aware keypoint descriptors, we recover the filtered matches without the loss in matching precision.

Acknowledgements

I am very grateful to my supervisor James Pritts, who was mentoring me and giving precious advice on all stages of development of this thesis.

Big thanks to Maria Dobko, Oles Dobosevych, and Rostyslav Hryniv, who were open to questions and always found time to help me when needed.

Lastly, I want to thank all my family and friend for providing a lot of support, being patient, and motivating me during the whole period of my studies.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
RANSAC	**RA**ndom **SA**mpling **C**onsensus
AGNN	**A**ttentional **G**raph **N**eural **N**etwork
MLP	**M**ulti-**L**ayer **P**erceptron
MS	**M**atching **S**core
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise

To my family

Chapter 1

Introduction

Image matching is a fundamental problem in the whole computer vision and multiple-view geometry fields which tries to overlay two or more images of one scene or object. Many emerging technologies like augmented reality, self-driving cars, robotics, or 3D reconstruction of objects rely deeply on the process of matching. These technologies are used ubiquitously in medicine, agriculture, education, entertainment, and many more.

Detecting correspondences between images is a primary step towards solving a large number of problems in computer vision. Among them are scene reconstruction, visual localization, camera parameters estimation, visual navigation, odometry, pose estimation, image retrieval. Usually, algorithms that try to solve these problems should operate with a large amount of imaging data acquired under different conditions. This assumption poses a strict requirement for the image matching component to be both efficient and robust.

The complexity of a particular image matching problem strongly depends on the assumptions posed on the problem and the domain where it is used. In this thesis, we focus our attention on the problem of finding point (pixel) correspondences in two images in real-world outdoor environments. This case is the most basic yet very important. Also, we assume significant viewpoint, illumination, and scale changes across the images, which allows for a wide range of potential applications. Further, throughout this work, we will refer to this particular problem simply as image matching if the opposite is not explicitly stated.

Finding correspondences between any two pixels in the image pair will be very computationally intensive and redundant for many downstream algorithms. The most widely adopted pipeline for image matching operates on the sparse set of points from both images and tries to match them based on the local appearance around them. This method allows for efficient image matching, but poses the new problems of detecting a repeatable set of keypoints in both images and creating invariant representations for them, called local descriptors.

Keypoints matching is a challenging task when working with the images of artificial outdoor environments because of the presence of a significant number of repeated textures. Human-made objects, such as buildings, are usually full of symmetrical patterns, which are very hard to match based only on local descriptors. The typical approach to dealing with matches of keypoints that belong to repeated textures is to filter them as they cannot be considered confident. Throwing away important information hurts the overall performance of the matching algorithms, especially for such downstream tasks as Structure-from-Motion and Multiple View Stereo. Recovering more correct matches would be very important for constructing denser 3D models of the scenes, which would be beneficial for further analysis.

In recent years deep learning has found enormous success in the computer vision area. Neural network-based algorithms showed state-of-the-art performance on image classification, object detection, semantic segmentation, and many other related tasks. Image machining also benefited a lot. Using Convolutional Neural Networks (CNN) allowed researchers to create better keypoint detectors and descriptors in repeatability and invariance to different transformations. Nevertheless, the matching process itself was still mainly done by different variations of nearest-neighbors search among the local descriptors with post-processing filtering techniques.

A recent state-of-the-art approach called SuperGlue [29] tackles keypoints matching in an entirely new way by formulating it as an end-to-end prediction task solved by a neural network. Also, it proposes a novel idea to establish matching based not only on local visual information but also on contextual information from other keypoints and their location relative to each other.

In this work, we explore the capabilities of recent state-of-the-art keypoint matching methods in terms of matching images of outdoor environments rich in repeated textures. We make the investigation of cases where they fail, especially when matching repeated patterns. In particular, we investigate the performance of the SuperGlue matching technique and identify the problems with its global keypoint representations. We proposed adding a metric learning component to the training objective, which can potentially solve them. By implementing the SuperGlue pipeline, we were able to compare the performance of the original algorithm with the proposed approach.

In Chapter 2, we provide a general overview of the standard keypoint-based image matching pipeline. Chapter 3 reviews some of the recent deep-learning-based techniques, which are of special relevance to our work. In particular, we review the SuperGlue method, which proposes to match keypoints based on global context-aware descriptors. In Chapter 4, we describe the problem of matching images with a large number of repetitions. In Chapter 5, we perform a more careful analysis of the strengths and weaknesses of using global keypoints representations provided by SuperGlue, especially when repeated textures are present in the images. By identifying potential problems, we propose a metric-learning-based solution to them. In Chapter 6, we describe the experiments performed on SuperGlue enhanced by the metric learning objective function. Finally, in Chapter 7, we make the conclusion about the performance of our approach and identify the directions for future work.

Chapter 2

Image matching problem

2.1 Image matching pipeline

The most widely adopted pipeline for image matching was first formalized in the survey paper on image registration [39]. It consists of four consecutive steps: feature detection, feature description, feature matching, transformation estimation. Here, we will provide a brief overview of existing and most widely used methods for each of the steps.

2.1.1 Features detection and description

A good feature is invariant to transformations, repeatable, and efficient both to compute and store. Usually, features are detected at locations robustly found under geometric transformations, viewpoint, or illumination changes. Such features are edges, corners, blobs, or keypoints. The latter is of specific interest since they are simple to define and store. Keypoints are generally described based on the local area around them. Following the same line of reasoning, as with detection, descriptors are built such that they are invariant to a wide range of transformations, as well distinctive enough to perform matching.

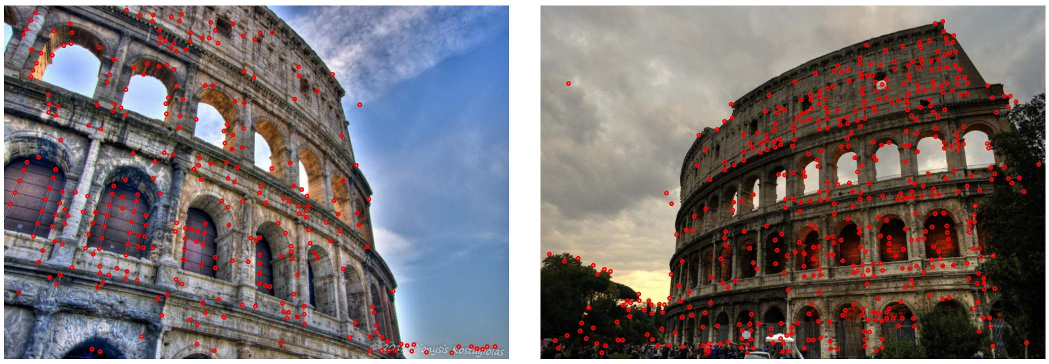
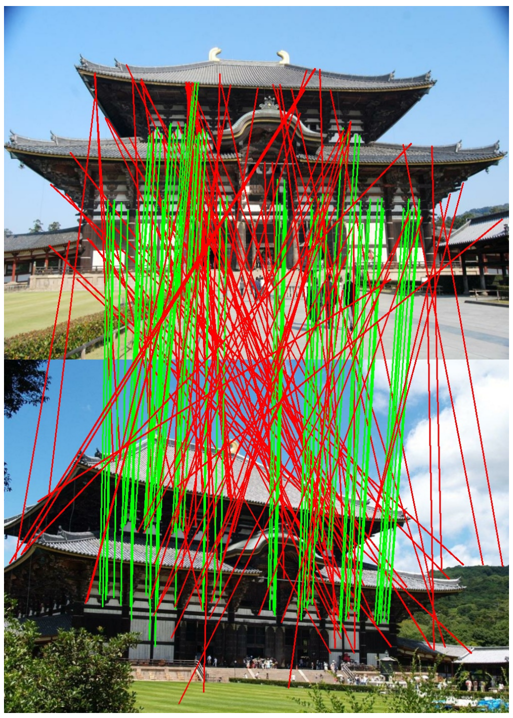


FIGURE 2.1: Example of keypoints detected by SuperPoint [6]. Produced set of keypoints is repeatable under large viewpoint and illumination changes.

More formally, given two images A and B, one tries to detect a set of keypoint locations $\mathbf{p}$ and corresponding local descriptors $\mathbf{d}$ on each of them. We will refer to individual keypoint location and local description on the images A and B as $\mathbf{p_i}^A$, $\mathbf{d_i}^A$ and $\mathbf{p_j}^B$, $\mathbf{d_j}^B$, respectively, where $i = 1 \ldots M$ and $j = 1 \ldots N$.

In the past few decades, many handcrafted methods for keypoints detection and description were invented. Some of them perform only detection task: Harris corner detector [14], FAST [27], MSER [22]; some can only make description: BRIEF [3], while others can do both: ORB [28], SIFT [21], SURF [1].

Feature detection and description steps benefited a lot from applying deep learning approaches. Features produced by such methods as SuperPoint [6], D2-Net [8], HardNet [23] set new state-of-the-art results in various image matching benchmarks.


(A) NN matcher with mutual nearest neighbors check

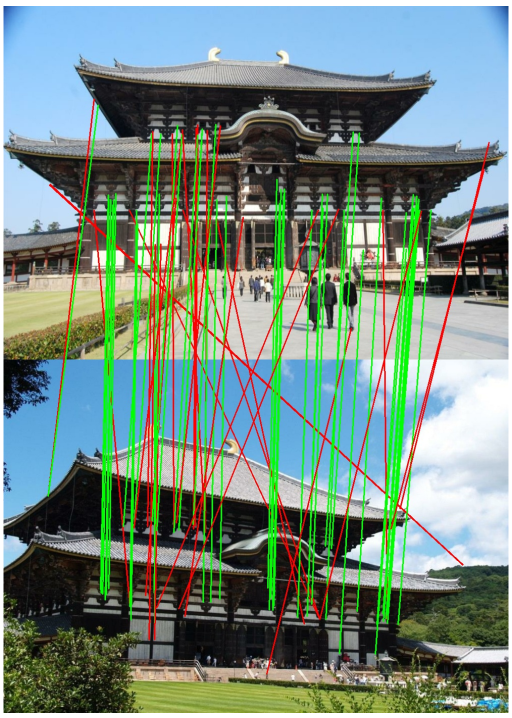

(B) NN matcher with Lowe's ratio test

FIGURE 2.2: Image matching performed by Nearest Neighbors matcher. Keypoints are detected and described with SIFT. We used RANSAC to robustly estimate the homography transformation between two images. Matches that are inliers and outliers according to the transformation are shown in green and red respectively. Lowe's ratio test tends to filter both many inliers and outliers.

2.1.2 Features matching

Traditionally descriptors are matched based on the Nearest Neighbors (NN) search. The matches produced this way are further filtered using such techniques as mutual nearest neighbors check or nearest neighbors consensus [34]. Lowe's ratio test [20] is one of the filtering approaches of particular interest to us. It discards a match as non-confident if the distance to the second NN is not substantially bigger than the distance to the first NN. The match is discarded if:

$$\left\| \mathbf{d}_\mathbf{i}^\mathbf{A} - \mathbf{d}_{\mathbf{j}_1}^\mathbf{B} \right\|_2 > c \cdot \left\| \mathbf{d}_\mathbf{i}^\mathbf{A} - \mathbf{d}_{\mathbf{j}_2}^\mathbf{B} \right\|_2, \tag{2.1}$$

where j_1 and j_2 are indices of the first and the second nearest neighbors in image B of $\mathbf{d}_\mathbf{i}^\mathbf{A}$ and $c \in (0, 1)$ is predefined constant.

Recent Neural Networks-based approaches to matching, in general, followed the NN paradigm. [38, 26, 2] proposed deep learning-based filtering techniques but still relied on the NN search.

2.1.3 Transformation estimation

Tentative correspondences established in the previous step are used to estimate some geometric transformation between the structures on two images. Usually, one needs a small number of correct point correspondences to estimate this transformation. For example, one needs only 4 points matches to estimate the homography matrix in a non-degenerate case. On the other hand, there are often much more tentative matches, a large fraction of which may be incorrect. Robust estimation techniques such as RANSAC [10] were designed to operate under such assumptions. RANSAC and its many variations show excellent performance when estimating transformation models in scenarios with many outlier correspondences.

2.2 Metric Learning in image matching

Metric learning approaches have enormous success in lots of Computer Vision problems. It helped to boost the performance of many classification systems by learning better image representations. Training a model in a metric learning fashion allows it to generalize well even to the instances of classes unseen before and produce meaningful representations vectors for them. Such problems as face verification and face identification benefited a lot because usually, one cannot access instances of all possible classes during training.

Metric learning has become a popular way to learning local descriptors using deep neural networks. Such methods as SuperPoint, D2-Net, or HardNet learn to describe local regions around keypoints centers such that the output descriptors are invariant to geometric transformations and illumination changes while being discriminative enough to match. This is usually achieved by variants of triplet margin loss [4].

Chapter 3

Related Works

In this chapter, we will review state-of-the-art approaches to keypoints detection, description, and matching, which are of particular relevance to us. The work presented in this thesis is based mostly on the SuperGlue matching algorithm. Therefore here, we will also provide an extensive overview of its components, as well as identify its weaknesses.

3.1 Keypoints detection and description

3.1.1 SuperPoint

SuperPoint [6] aims to create an end-to-end learnable framework for keypoints detection and description. Detection and description are done with a single CNN model that shares a backbone and has separate heads for predicting keypoints and describing them.

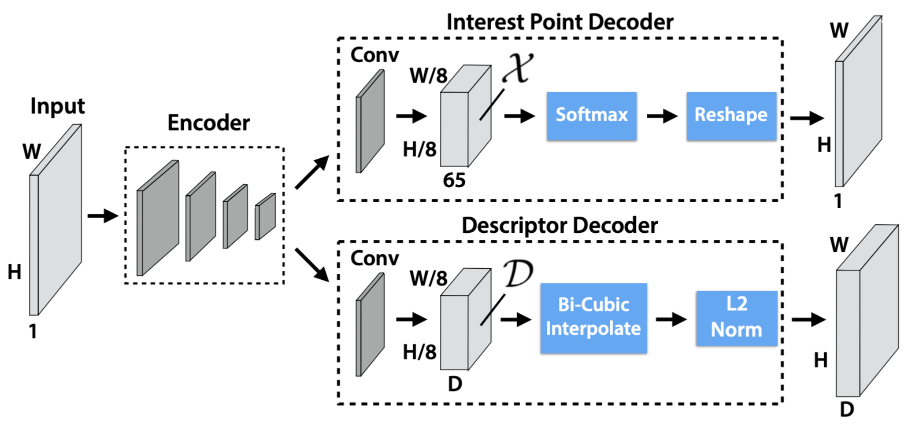


FIGURE 3.1: SuperPoint model architecture. Separate heads produce the detection and description of keypoints in a single forward pass of a model. Image taken from [6].

The detector head is trained in a self-supervised manner requiring no human-labeled data. Firstly, it is pretrained on synthetically generated shapes with predefined keypoints locations (angles, joints, etc.). Then, the model is further trained on a large set of unlabeled images. The model provides supervision for itself by using a process called Homographic Adaptation, where ground-truth keypoints are generated by making detections on different warped versions of the same image and combining them together. The model learns to predict keypoints locations by using cross-entropy loss which is used for pixel-wise classification.

The descriptor head is trained by using separate loss functions. By applying random homographies, one can establish ground-truth correspondences between points in an image and its warped version. The objective function tries to minimize the distance between descriptors of such two keypoints. The distance to descriptors of all others keypoints is maximized. Thus, the model aims to learn descriptors that are invariant to generic homographic transformations and distinctive enough to be robustly matched.

3.1.2 HardNet

HardNet [23], on the other hand, only performs a description task based on the local patches around keypoints. Simple and lightweight CNN architecture allows for fast and efficient keypoints descriptions.

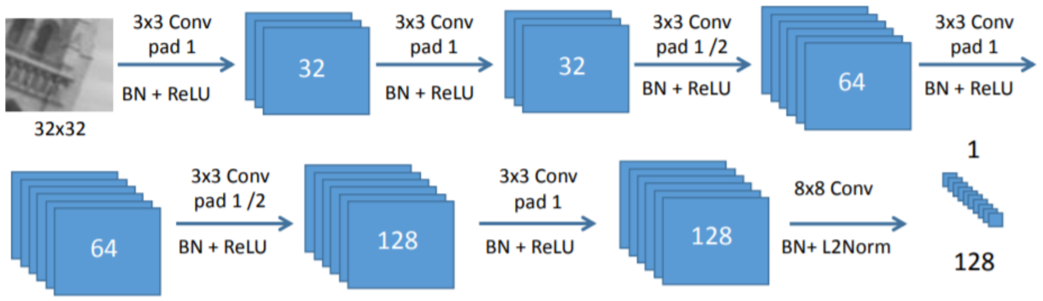


FIGURE 3.2: Lightweight architecture of HardNet, adopted from L2Net [33] model. Importantly, descriptors for keypoints are produced based only on the local neighborhood. Image taken from [23].

Descriptors are learned by using triplet margin loss with hard-negative mining [15]. The whole process is inspired by Lowe's ratio test. HardNet objective tries to minimize the descriptor's distance to its ground truth match while simultaneously maximizing the distance to the second nearest neighbor.

3.2 Keypoints matching using SuperGlue

The SuperGlue approach to image matching fits into the standard image matching pipeline while replacing classical Nearest Neighbor-based methods. As its input it expects a set of keypoints on both images $\mathbf{p}^A$ and $\mathbf{p}^B$ along with their local descriptors $\mathbf{d}^A$ and $\mathbf{d}^B$. Keypoints and their descriptors can be provided by any existing method or combination of them. In this work, we use only SuperPoint keypoints as they provide the best accuracy when matching.

SuperGlue consists of two main components: Attentional Graph Neural Network (AGNN) and Optimal Matching Layer. The former is responsible for context aggregation and provides global context-aware descriptors enhanced with positional information of keypoint. The latter component allows formulating descriptors matching as a prediction task. Using a differentiable Sinkhorn algorithm [5], the model can propagate gradients directly from its prediction to AGNN, allowing to train it and make inferences in an end-to-end manner.

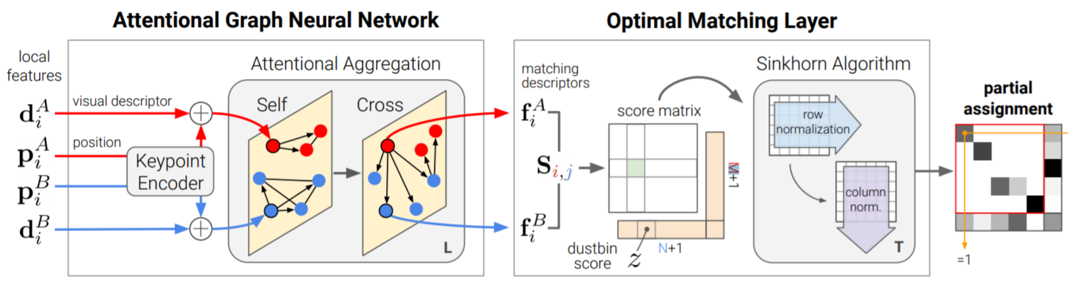


FIGURE 3.3: SuperGlue matching pipeline. Attentional Graph Neural Network produces context-aware descriptors by applying attentional aggregation. Then, matching assignment matrix is constructed by differentiable Sinkhorn algorithm. Image taken from [29].

3.2.1 Attention Mechanisms

Attention mechanisms were firstly utilized in Natural Language Processing field for sequence-to-sequence tasks but soon gained more popularity in vision problems. In particular, dot-product attention was firstly introduced in the Transformers architecture [36]. However, nowadays, it is commonly used to replace convolution operations in image processing architectures [24, 12].

Dot-product attention can be seen as an operation that maps a query vector and set of key-value pairs to the output. In practice, there are multiple queries that are stacked into one matrix Q. Key and value pairs can also be represented as matrices K and V. Then

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{3.1}$$

where d_k is the dimensionality of queries and keys.

In Natural Language Processing, dot-product attention allows building context embeddings [7] of words out of independent word-level embeddings. The same ideas can be applied to keypoints descriptors. By applying attention operations to individual local descriptors, we can build global context-aware features. Also, by incorporating positional information about keypoints location, the model can reason about their relative location and co-occurrences. These techniques show a promising way to resolve ambiguous matches predicted based only on the local information.

3.2.2 Attentional Graph Neural Network

SuperGlue uses a complete undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}_{all}\}$ over all the keypoints in both images, where $\mathcal{V} = \mathbf{p}^A \cup \mathbf{p}^B$. The edges are split into two separate sets $\mathcal{E}_{all} = \mathcal{E}_{self} \cup \mathcal{E}_{cross}$. Edges from $\mathcal{E}_{self}$ connect keypoints that lie in a single image, while edges from $\mathcal{E}_{cross}$ connects keypoints across images. AGNN propagates information only from keypoints that are connected by one type of edges at a single layer. At odd layers, information is propagated using self-edges, while at even layers from cross-edges.

AGNN propagates information from one node to the other by using message passing [11]. If we denote the representation of each keypoint in an image A after layer l as ${}^{(l)}\mathbf{x}_i^A$ and the message passed from keypoints connected by vertices $\mathcal{E} = \{\mathcal{E}_{self}, \mathcal{E}_{cross}\}$ as $\mathbf{m}_{\mathcal{E}\to i}$, then the update is given by:

$${}^{(l+1)}\mathbf{x}_i^A = {}^{(l)}\mathbf{x}_i^A + MLP([{}^{(l)}\mathbf{x}_i^A \| \mathbf{m}_{\mathcal{E}\to i}]) \tag{3.2}$$

The same computations are performed for keypoints in the image B.

The message $\mathbf{m}_{\mathcal{E}\rightarrow i}$ is computed by using dot-product attention, where the query is an intermediate representation of keypoint i, and keys and values are computed based on intermediate representations of message sources.

The original representation of each keypoint ${}^{(0)}\mathbf{x}_i^A$ is composed of visual descriptor along with positional information embedded in a higher-dimensional space with Multi-Layer Perceptron:

$$ {}^{(0)}\mathbf{x}_i^A = \mathbf{d}_i^A + MLP(\mathbf{p}_i^A) \tag{3.3}$$

The representation after the last layer L is linearly projected to obtain the final global context-aware descriptor for each keypoint:

$$\mathbf{f}_i^A = \mathbf{W} \cdot {}^{(L)}\mathbf{x}_i^A + \mathbf{b} \tag{3.4}$$

Descriptors $\mathbf{f}_i^A$ and $\mathbf{f}_i^B$ are kept unnormalized. The norm reflects the matching confidence for each keypoint.

3.2.3 Optimal Matching Layer

Optimal Matching Layer computes an assignment matrix $\mathbf{P}$, representing the matching probabilities for each possible correspondence. Each keypoints in one image can be matched to only one keypoint in the other image or will have no match at all because of viewpoint change and occlusion. To account for keypoints that cannot be matched, SuperGlue adds an auxiliary dustbin keypoint on each image that will match any unmatchable keypoints in other images. Given that there are M keypoints on the image A and N keypoints on the image B, $\mathbf{P}$ is $(M+1) \times (N+1)$ matrix such that

$$\mathbf{P}\mathbf{1}_{N+1} = \mathbf{a},\ \mathbf{P}^{\mathbf{T}}\mathbf{1}_{M+1} = \mathbf{b} \tag{3.5}$$

$$\mathbf{a} = [\mathbf{1}_M^T\ N],\ \mathbf{b} = [\mathbf{1}_N^T\ M] \tag{3.6}$$

Given the matching scores $\mathbf{S}$ between any two keypoints, the task is to maximize $\sum_{i,j} \mathbf{S}_{i,j}\mathbf{P}_{i,j}$. The score for any two keypoints is given as dot product between their context-aware descriptors.

$$\mathbf{S}_{i,j} =< \mathbf{f}_i^A, \mathbf{f}_j^B >, \text{ for } \forall i = 1 \ldots M, j = 1 \ldots N \tag{3.7}$$

Matching score between any keypoint and dustbin, as well as between two dustbin is a learnable parameter.

The partial assignment matrix is computed using a differentiable Sinkhorn algorithm.

3.2.4 Objective function

Original SuperGlue implementation optimizes simple negative log-likelihood loss in a completely supervised fashion. Given a set of ground truth correspondence $\mathcal{M} = \{(i,j)\}$ and two sets of unmatchable keypoints from both images $\mathcal{I}$ and $\mathcal{J}$ the loss is computed as:

$$L = -\sum_{(i,j)\in\mathcal{M}} \log\mathbf{P}_{i,j} - \sum_{i\in\mathcal{I}} \log\mathbf{P}_{i,N+1} - \sum_{j\in\mathcal{J}} \log\mathbf{P}_{M+1,j} \tag{3.8}$$

In practice, the loss is averaged, such that true correspondences and unmatchable keypoints have an equal contribution.

$$L = -\frac{1}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} \log \mathbf{P}_{i,j} - \frac{1}{2|\mathcal{I}|} \sum_{i\in\mathcal{I}} \log \mathbf{P}_{i,N+1} - \frac{1}{2|\mathcal{J}|} \sum_{j\in\mathcal{J}} \log \mathbf{P}_{M+1,j} \tag{3.9}$$

Chapter 4

Problem Formulation

In this work, we focus our attention on the problem of matching two images of one scene or object in the outdoor environment. This problem is especially hard because of the large number of repeated structures in man-made environments. Repetition on the image can be defined as a region that repeats itself according to some rule (e.g. translation, rotation, reflection, etc.). The presence of repeated patterns in the image matching pipeline will cause the detection of keypoints with roughly the same descriptors. An example image of a scene with a large number of translation repeated textures is given below.

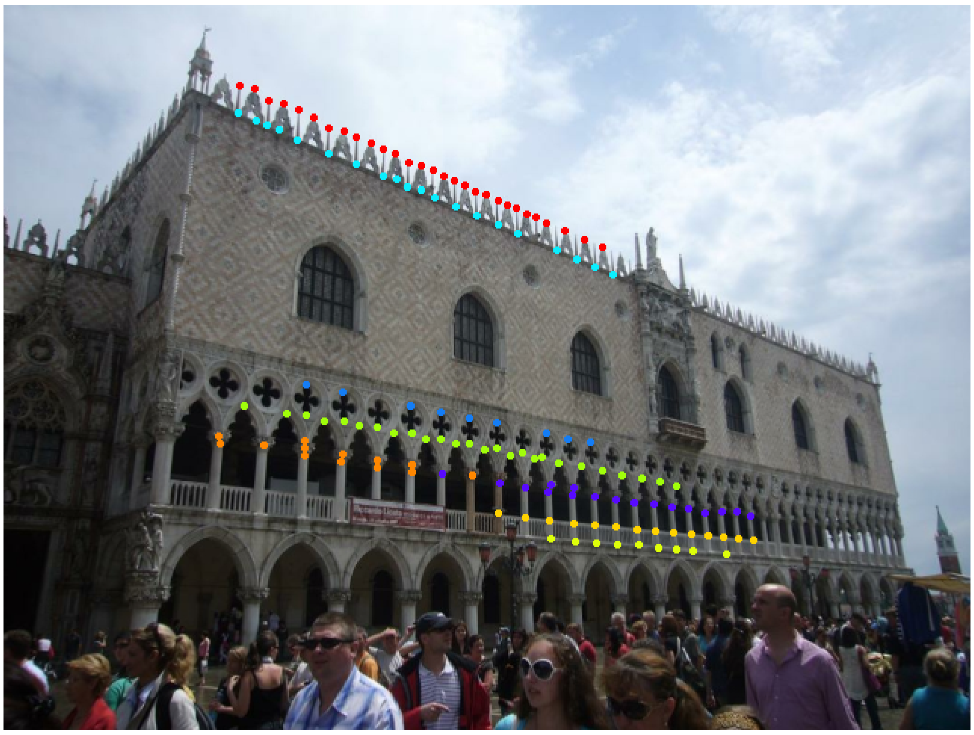


FIGURE 4.1: Example of repeated features detected by DBSCAN [9] using keypoint descriptors from SuperPoint. Clusters of repetitions are represented in different colors.

Methods that perform matching based only on the local descriptors and Nearest Neighbor search are doomed to fail to disambiguate repetitions since the local representation for each keypoint is roughly the same. The easiest way to deal with this problem is to filter out repeated pattern matches. Lowe's ratio test will do precisely this. Since the first and second nearest neighbors will have very similar descriptors, the distance to both of them will be similar too. Thus a lot of repeated pattern matches will be filtered, resulting in a loss of useful information.

In this thesis, we try to improve existing matching algorithms by paying more attention to repeated patterns in the images. As a backbone for our solution, we use the SuperGlue method, which shows state-of-the-art performance on image matching tasks. The SuperGlue algorithm has no explicit mechanism for dealing with repeated patterns. Nevertheless, there are two components inside of SuperGlue that can implicitly help matching repetitions:

- positional information encoded in local descriptors
- context-aware keypoints descriptors.

Building global contextual descriptors and incorporating relative positional information about keypoints can help resolve ambiguous matches, even when local descriptors are precisely the same.

In chapter 5 we will provide an error analysis of SuperGlue in terms of repeated patterns to identify how well it deals with the problem of repetitions. We noticed that SuperGlue shows high precision when matching, meaning that matches predicted by this algorithm are mostly accurate. Nevertheless, SuperGlue still seems to miss a lot of matches present in the image pair.

The problem with unmatched keypoints can be seen on the repeated features matches, a significant amount of which is filtered by SuperGlue. In this work, we try to answer the question, whether those correct matches can be recovered. By using metric learning approaches, we try to explicitly make SuperGlue learn more distinctive descriptors, particularly for repeated patterns, such that their matching will be easier.

Summarizing, the contributions of this work are the following:

1. Error analysis of existing SuperGlue method on the problem of matching repeated patterns.
2. Implementation of SuperGlue training procedures.
3. Improvement to SuperGlue method by using metric learning.
4. Comparison of SuperGlue enhanced with the metric learning objective to the original method.

Chapter 5

Method

In order to analyze the performance of SuperGlue and add metric learning components to it, we first had to implement the original solution. Unfortunately, the authors did not provide the code for dataset preparation and training. Only inference code and pretrained model weights were available.

Firstly, we will describe the process of dataset preparation, present details of establishing ground-truth correspondences, and reason about evaluation metrics used further in the thesis.

5.1 Establishing ground truth correspondences

We used the MegaDepth [19] dataset for training and evaluation experiments. In the original SuperGlue implementation, it was used only for the training process. The dataset is split into images of separate scenes from different viewpoints. For each image, camera intrinsic and extrinsic parameters relative to the world coordinates are provided. Also, dense depth maps for the images are available, even though depth information can be present only for part of the image. This information is computed by the COLMAP [30, 31] software which can perform accurate Structure-from-Motion and Multiple View stereo from the sequence of images of a single scene.

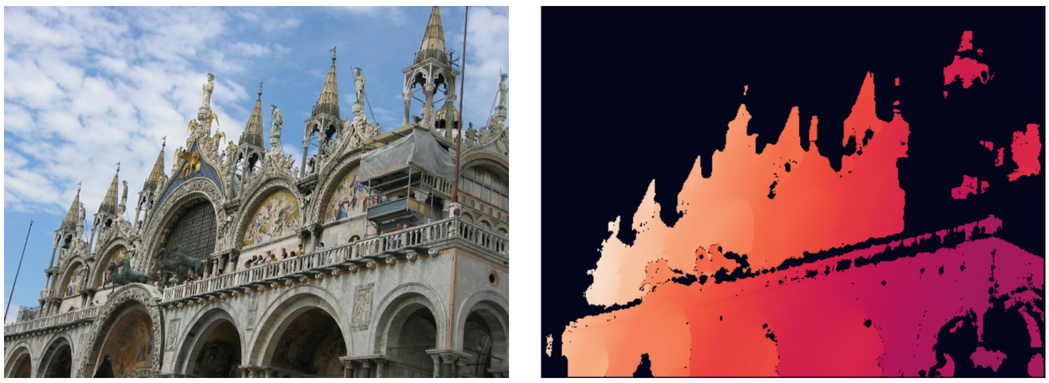


FIGURE 5.1: Example of the image (left) and corresponding depth map (right) from MegaDepth dataset. Brighter values in the depth map represent larger depth values. Missing depth information is shown in black.

To establish the ground truth point-to-point correspondences under the pinhole camera model assumption, one needs calibration parameters of both cameras, camera motion from one view to another, as well as depth information. Relative rotation and translation between both view A and B can be found from rotation and translation of individual cameras in the world coordinates as following:

$$\begin{aligned} R_{AB} &= R_B R_A^T \\ T_{AB} &= -R_B R_A^T T_A + T_B \end{aligned} \tag{5.1}$$

Given two sets of keypoints $\mathbf{p}^A$ and $\mathbf{p}^B$ on images A and B, respectively, we first reproject keypoints coordinates from the image A to the image B. Taking a point p with depth z on the image A we transform it with the next equations:

$$\begin{aligned} P_A &= z \cdot K_A^{-1} [p(x) \ \ p(y) \ \ 1]^T \\ P_B &= R_{AB} P_A + T_{AB} \\ \hat{p} &= K_B P_B \end{aligned} \tag{5.2}$$

Next, we symmetrically repeat the same process with keypoints in the image B. If reprojection of $\mathbf{p_i^A}$ is the nearest neighbor of $\mathbf{p_j^B}$ and vice versa, and symmetric transfer distance [32] between them is less than the predefined small threshold chosen as 3 px, we say that these two keypoints establish a match.

Also, some keypoints may not have any match because of viewpoint change, occlusion, or just inaccuracies of the detector. If the distance from the keypoint's reprojection to its nearest neighbor is bigger than 5 px, we call this keypoint unmatchable. Keypoint, for which we don't possess depth information, or the distance of keypoint's reprojection to its nearest neighbor lies in a range from 3 px to 5 px, is considered not confident, so we can't say whether it has a match or not. Such keypoints are not included in SuperGlue loss computations.

To summarize, the process described above allows us to obtain a set of confident ground-truth correspondences $\mathcal{M}$, two sets of unmatchable keypoints $\mathcal{I}$ and $\mathcal{J}$ and set of ambiguous keypoints.

5.2 Selecting evaluation metrics

As can be seen from the previous section, establishing ground truth point-to-point correspondences requires accurate depth information. In our case, we have noisy depth maps with many missing locations. The resulting set of ground truth correspondences consists of many ambiguous matches, making no problem when training the model since we can ignore those matches when computing the loss. On the other hand, ignoring those matches at evaluation time can dramatically influence the metrics and result in a very inaccurate estimate of model performance.

A common strategy when evaluating the image matching method is to measure its performance on the downstream task. In our case, we will choose the task of relative pose estimation since we possess accurate ground truth poses information for any image pair. Following the SuperGlue paper, we will use the AUC of the pose error at (5°, 10°, 20°) thresholds. We predict relative pose information by estimating the Essential matrix using RANSAC [10]. The pose error is computed by accounting for angular error in rotation and translation and taking the maximum of two.

To analyze the performance of SuperGlue at the level of individual keypoints matches, we will use precision (P) and matching score (MS) metrics.

Precision	MS	AUC@5°	AUC@10°	AUC@20°
96.05	26.68	43.63	58.93	71.55

TABLE 5.1: Performance of the pretrained SuperGlue model on the validation split of the MegaDepth dataset.

Precision	MS	AUC@5°	AUC@10°	AUC@20°
94.62	24.01	36.91	51.19	63.96

TABLE 5.2: Performance of the pretrained SuperGlue model on the test split of the MegaDepth dataset.

$$\text{Precision (P)} = \frac{\text{True Positives}}{\text{Predicted Matches}}$$

$$\text{Matching Score (MS)} = \frac{\text{True Positives}}{\text{Total Keypoints}}$$

The match is considered true positive if the corresponding epipolar error in a calibrated ray space is smaller than the predefined threshold $5e^{-4}$. This constant is chosen the same as in the original paper for the consistency purposes. The recall of matching cannot be estimated since we do not have an accurate set of ground-truth correspondences. Nevertheless, MS can be used to estimate the method's performance in terms of false negative predictions.

5.3 SuperGlue error analysis

In this section, we identify existing problems with SuperGlue. The following analysis will be done using a pretrained SuperGlue model with the weights provided by the authors. Keypoints are detected and described using the SuperPoint model. The images are taken from the validation split of the MegaDepth dataset [19]. All images are resized to the size of 960×720 pixels and converted to grayscale.

As shown in the Tables 5.1 and 5.2, SuperGlue achieves very high precision scores, meaning that most of the predicted matches are correct. Nevertheless, MS has much room for improvement. Low MS combined with high precision is a direct indicator of many false negatives predictions, meaning that correct ground-truth correspondences are present in the image, but SuperGlue discards them.

When making a prediction, SuperGlue leaves the keypoint unmatched if none of the matching probabilities for a given keypoint passes confidence threshold $c = 0.2$. Setting up this confidence threshold has a filtering effect, similar to other filtering techniques used by classical matching methods.

We discover that unmatched keypoints have their matching scores more uniform then those that are matched by SuperGlue. For each keypoint i in the image A, we took its matching scores $S_{i,j}$ with all the keypoints $j = 1 \dots N$, $N = 1024$ in the image B and converted them to the distribution using softmax function. Analysing the mean of entropies of this distributions for each keypoint in the validation subset shows that keypoints that are unmatched have 10 times larger entropy: 0.0094 for matched keypoints compared to 0.096 for unmatched. Distributing the matching probability among many potential matches results in the effect that this keypoint cannot pass the confidence threshold.

We link the problem described above directly to the problem of repeated textures. Suppose that keypoint $\mathbf{p}_i^A$ matches $\mathbf{p}_j^B$, while $\mathbf{p}_j^B$ belongs to the repetition. Then there are several keypoints in image B, which have similar descriptors, and their matching scores with $\mathbf{p}_i^A$ would be almost the same. This will result in matching probability distributing across repetition. Therefore the maximum probability in the distribution will become smaller than the threshold, and $\mathbf{p}_i^A$ will be treated as unmatched.

5.3.1 Context-aware descriptors of repeated patterns

Here, we will analyze the effect of SuperGlue on the descriptors of repeated patterns. The first problem in question is to identify repeated patterns. We do this by clustering descriptors of detected keypoints with the DBSCAN [9] clustering algorithm. The idea is that keypoints detected on the repeated textures will form high-density regions in the descriptors space. These regions will be detected by DBSCAN and packed into clusters, while others descriptors that do not belong to repeated patterns will be marked as outliers.

Attentional mechanisms with positional encoding provided by Graph Neural Network should output for each keypoint descriptor, which will be distinctive even when keypoint belongs to a repeated pattern. In practice, this is not the case. We noticed that for at least 30% of repetitions the descriptors $\mathbf{f}_i^A$, $\mathbf{f}_j^B$ after AGNN have the same or even smaller pairwise distances than local descriptors $\mathbf{d}_i^A$, $\mathbf{d}_j^B$.

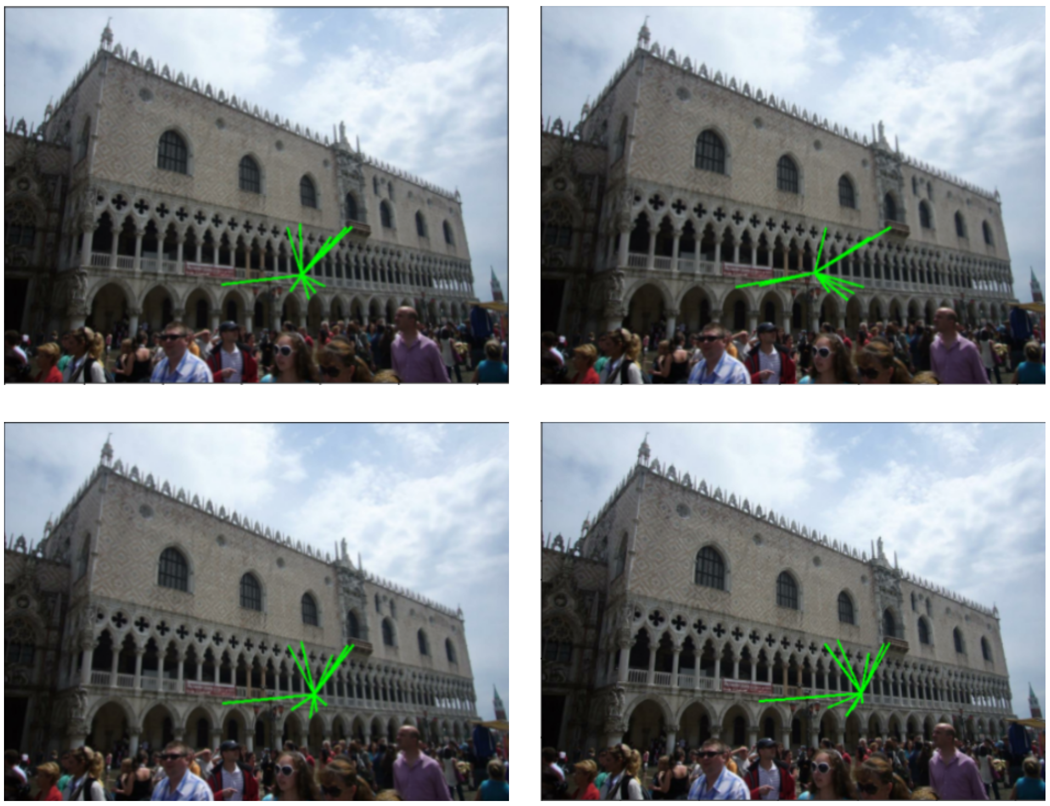


FIGURE 5.2: Visualisation of self-attention for points that belong to a single repeated texture at the first layer in AGNN. For each keypoint in repetition only top 10 attention weights are shown as lines between query and key. Keypoints from the same repetition attend to very similar set of keys independent of their position.

This effect can be explained by the symmetry of computations inside of AGNN. Because we operate on the complete graph, all keypoints from the repeated patterns attend to the same set of message sources, thus receive similar messages. Continuing this process several times has an even more smoothing effect, making keypoint less distinctive. Positional encoding added to the local descriptors should have resolved this issues, but, empirically, we can see this is not the case.

5.4 Metric learning of context-aware descriptors

To alleviate the problem of identical descriptors output by AGNN, we add a metric learning component to the SuperGlue's objective function. Inspired by HardNet [23] approach to learning local descriptors, we apply metric learning loss functions with hard negative mining to learn more distinctive context-aware descriptors, which will be easier to match in the end. The main difference compared to HardNet is that our approach operates on the global descriptors. Therefore we can learn substantially different context-aware descriptors even in the case when local descriptors are identical. HardNet, in turn, works only with local patches and thus will describe identical local patches with identical local descriptors. To our knowledge, this is the first work that tries to enhance SuperGlue's context-aware descriptors by adding the metric learning component to the objective function.

5.4.1 Triplet loss for positive matches

The HardNet approach is motivated by a classical Lowe's ratio test [20]. It uses triplet margin loss with hard negative mining technique to learn the embeddings of local patches around keypoints. Given a ground truth match between keypoints i in the image A and keypoint j in the image B, the triplet is formed in the following way: $\mathbf{d}_i^A$ - anchor, $\mathbf{d}_j^B$ - positive sample, and $\mathbf{d}_{k_i}^B$ - negative, where k_i is the index of the nearest neighbor of keypoint i in the descriptors space, of all keypoints that do not match i. The triplet loss is then given as:

$$L_{triplet} = max(\left\|\mathbf{d}_i^A - \mathbf{d}_j^B\right\|_2 - \left\|\mathbf{d}_i^A - \mathbf{d}_{k_i}^B\right\|_2 + m, 0) \tag{5.3}$$

Compared to the Lowe's ratio test, HardNet uses positive additive margin m instead of multiplicative constant c as in 2.1. The authors claim that it adds stability to training. We follow their line of reasoning and also use additive margin.

The other important question is a choice of distance measure. HardNet uses for its descriptors L_2-norm, which is not applicable in our case because the descriptors $\mathbf{f}^A, \mathbf{f}^B$ are not normalized. Instead, we use cosine distance, which is actually not a proper distance metric but is successfully used in metric learning. Cosine distance is normalized such that it lies in the range $[0, 1]$.

$$d_{cos}(x, y) = \frac{1 - cos(\theta)}{2}, \text{ where } \theta \text{ is angle between } x \text{ and } y. \tag{5.4}$$

Following HardNet, we use hard negative mining approach to create training triplets. Also, we define the triplet loss symmetrically for keypoints in both images, meaning that from one ground truth (i, j) match we make two triplets: ($\mathbf{f}_i^A$, $\mathbf{f}_j^B$, $\mathbf{f}_{k_i}^B$) and ($\mathbf{f}_j^B$, $\mathbf{f}_i^A$, $\mathbf{f}_{l_j}^A$), where k_i and l_j are the indices of the nearest neighbors of keypoints i and j out of all keypoints that do not match i and j respectively. The symmetrical triplet loss can be written as:

$$\begin{aligned} l_{pos}(i,j) &= max(d_{cos}(\mathbf{f}_i^A - \mathbf{f}_j^B) - d_{cos}(\mathbf{f}_i^A - \mathbf{f}_{k_i}^B) + m, 0) + \\ &+ max(d_{cos}(\mathbf{f}_j^B - \mathbf{f}_i^A) - d_{cos}(\mathbf{f}_j^B - \mathbf{f}_{l_j}^A) + m, 0) \end{aligned} \tag{5.5}$$

The total loss is computed as average over all ground truth matches $(i,j) \in \mathcal{M}$:

$$L_{pos} = \frac{1}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} l_{pos}(i,j) \tag{5.6}$$

5.4.2 Pairwise ranking loss for unmatched keypoint

We also add a pairwise ranking loss component to learn better descriptors for unmatchable keypoints. The aim is to separate them from all the descriptors in another image by a margin of m. Analogously to the previous section, we apply hard negative mining for selecting descriptor pairs. For any unmatchable keypoint i in image A the loss is given as following:

$$l_{neg}^A(i) = max(-d_{cos}(\mathbf{f}_i^A - \mathbf{f}_{k_i}^B) + m, 0), \tag{5.7}$$

where k_i is nearest neighbor of i from image B. The loss for unmatchable keypoints in image B is defined the same way:

$$l_{neg}^B(j) = max(-d_{cos}(\mathbf{f}_j^B - \mathbf{f}_{l_j}^A) + m, 0) \tag{5.8}$$

The total loss is averaged over the two sets of unmatchable keypoints $\mathcal{I}$ and $\mathcal{J}$:

$$L_{neg} = \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} l_{neg}^A(i) + \frac{1}{|\mathcal{J}|} \sum_{j\in\mathcal{J}} l_{neg}^B(j) \tag{5.9}$$

The margin m can be chosen as a different value for $l_{pos}(i,j)$, $l_{neg}^A(i)$, $l_{neg}^B(j)$, but in practice we selected a single value for all loss components.

The final objective for training the SuperGlue enhanced with the metric learning is a weighted sum of the original cross-entropy loss defined in equation 3.9 and the loss functions defined above in equations 5.6 and 5.9:

$$L_{total} = L + w(L_{pow} + L_{neg}) \tag{5.10}$$

The weight w is chosen as a hyperparameter.

Chapter 6

Experiments

6.1 Implementation details

The work presented in this thesis is implemented using Python 3.7 [35] programming language, which is the most popular language for data science and deep learning in particular. Our implementation is based on the official SuperGlue inference model implementation, which was written in PyTorch [25] framework for deep learning. We also utilized OpenCV [17] library for estimation of essential matrix using its RANSAC implementation. Other libraries like NumPy [13], Matplotlib [16] and Seaborn [37] were used to provide fancy visualizations. In our experiments, we tried to follow the original implementation as much as possible in order to be able to measure the impact of added metric learning component. Nevertheless, some changes had to be made due to the limitation of available computing and storage resources. Also, it is important to note that we implemented training procedures for SuperGlue from scratch based only on the original paper. Thus, some discrepancies between our and original implementation might exist due to the lack of training details.

6.2 Dataset

We used MegaDepth [19] dataset for error analysis, training, validation, and testing. We split the dataset into 153 training, 18 validation, and 18 test scenes. The training scenes are the same as used in the original implementation. The training split consists of 101,998 images with available depth information. Validation and test subsets have 2081 and 2528 images, respectively.

Not all possible image pairs were created using the available images. Following [8], we selected only images with overlap score $[0.1, 0.7]$ to create pairs. After applying this filtering step, we get 7,458,696/21,942/20,740 image pairs for train/val/test splits, respectively.

Ground-truth matches were generated on the fly, as the model is training. A detailed explanation of the generation process was provided in section 5.1. For keypoints detection and description, we used the SuperPoint model with pretrained weights. It is possible to propagate gradients to the SuperPoint description head, yet we kept it frozen for the purpose of fair comparison with the original implementation.

6.3 Training details

All the experiments were conducted on a single NVIDIA RTX 2080TI GPU. The input images were both resize to the size of 960×720 pixels. We used a batch size of 4 along with gradient accumulation techniques for four steps. This trick emulates the batch size of 16 as used in the original implementation. We have chosen the Adam [18] optimizer with a step scheduler for learning rate.

The SuperPoint model was used with the following parameters: 1024 maximum keypoints detected in each image and non-maximum suppression filtering with 3 px radius. Also, the final matching layer of SuperGlue performed 20 iterations of the Sinkhorn algorithm.

Next, we will describe the conducted experiments. For a fair comparison, we trained the SuperGlue model from scratch with and without metric learning component in the objective to compare how it affects the training process. Also, we took pretrained weights from the original implementation and trained the model further with metric learning components in the objective function. We used the same metrics and dataset split as in the Section 5.2.

6.3.1 Finetuning SuperGlue

The first set of experiments we conducted aimed to improve the model provided by the authors of SuperGlue. We decided to finetune the model with the metric learning component in the objective function. For this purpose, we selected a small starting learning rate $1e^{-6}$ with exponential decay of 0.9995. Next, we experimented with the margin parameter in triplet and pairwise ranking losses m. Early experiments showed that selecting m larger than 0.2 makes the model diverge fast and worsens the results.

For a fair comparison, we also finetuned the model without the metric learning component. We outperform the original model in Matching Score even with $w = 0$. This could be related to the specifics of ground-truth generation for training data, as well as potential differences in balancing positive and negative cases in the objective function. Training with the metric learning component improves matching scores even further, suggesting that metric learning is indeed beneficial for recovering more matches. More details about experiment settings and final results are presented in the Table 6.1 and 6.2.

m	w	Precision	MS	AUC@5°	AUC@10°	AUC@20°
-	0	96.00	30.86	42.83	57.88	70.41
0.05	$7e^3$	95.90	**32.00**	42.80	57.98	70.61
0.1	$3e^3$	95.20	29.88	41.20	56.07	68.64
0.15	$0.5e^3$	95.82	31.73	42.99	58.06	70.50

TABLE 6.1: Performance of the finetuned SuperGlue model on the validation split of MegaDepth dataset.

m	w	Precision	MS	AUC@5°	AUC@10°	AUC@20°
-	0	94.62	28.24	35.93	50.09	63.00
0.05	$7e^3$	94.49	29.05	35.51	49.53	62.48
0.1	$3e^3$	94.39	27.21	33.99	47.87	60.84
0.15	$0.5e^3$	94.63	**29.11**	35.28	49.43	62.51

TABLE 6.2: Performance of the finetuned SuperGlue model on the test split of MegaDepth dataset.

6.3.2 Training SuperGlue from scratch

The next experiments compare SuperGlue trained from scratch with and without metric learning component in the objective function. In the original implementation, SuperGlue was trained for 1 million iterations. When we use a batch size of 4 and 4 steps for gradient accumulation in our setting, it would require 4 million steps. With the available hardware, it would take around 20 days to complete the training. Considering the time constraints, we trained the model only for 1 million steps. Even though we did not beat the score of original implementation in such little number of iterations, we showed that adding metric learning component consistently outperforms the model without it in terms of matching score. Precision and other pose estimation metrics remain on the same level. We assume, that there might be the improvement in them as well when the number of iterations is increased. This remains to be tested in the future work.

m	w	Precision	MS	AUC@5°	AUC@10°	AUC@20°
-	0	94.13	29.13	39.55	54.30	67.03
0.15	10	94.47	29.77	39.35	54.13	66.84
0.2	10	93.28	31.01	40.20	54.92	67.39

TABLE 6.3: Performance of the SuperGlue model trained from scratch on the validation split of MegaDepth dataset.

m	w	Precision	MS	AUC@5°	AUC@10°	AUC@20°
-	0	92.63	27.44	33.25	47.06	60.00
0.15	10	93.15	27.86	33.78	47.58	60.44
0.2	10	91.98	29.03	33.8	47.56	60.51

TABLE 6.4: Performance of the SuperGlue model trained from scratch on the test split of MegaDepth dataset.

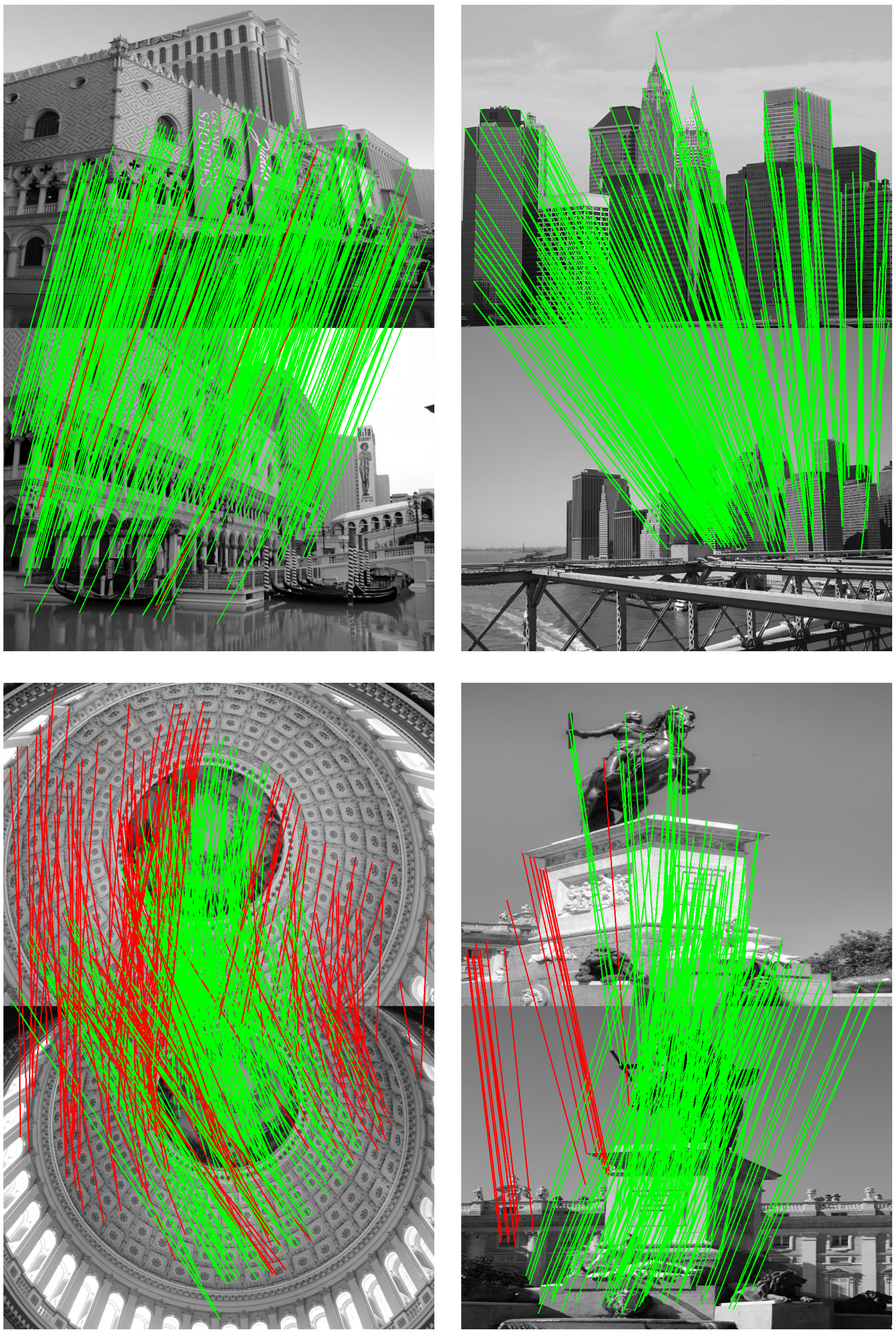


FIGURE 6.1: Examples of the matches predicted by SuperGlue, which was fine-tuned with metric learning component in the objective. Correct matches are shown in green, while incorrect in red.

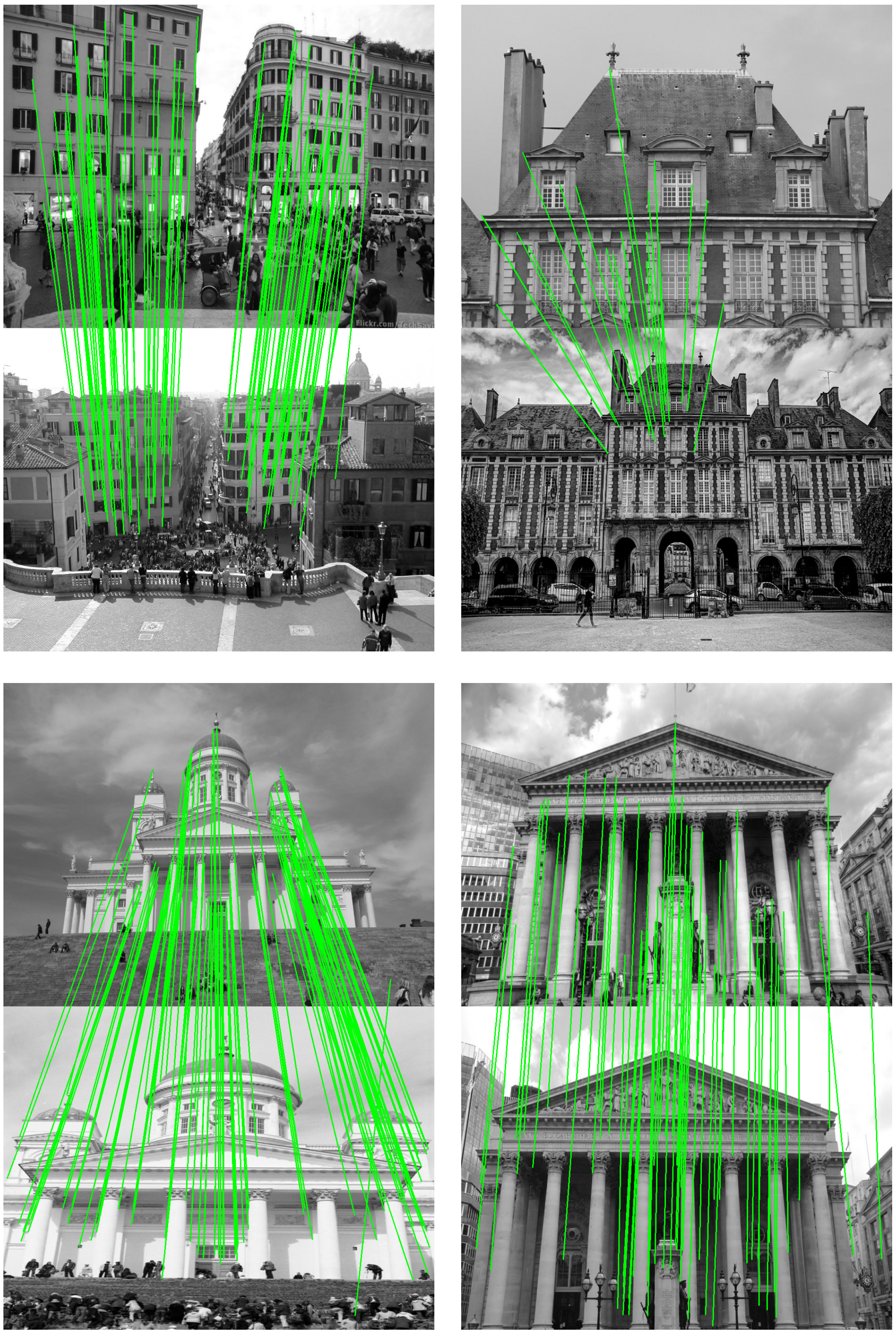


FIGURE 6.2: Examples of the correct matches recovered by SuperGlue, which was finetuned with metric learning component in the objective. In the images are shown only those matches which were missed by the original implementation.

Chapter 7

Conclusions and Future work

7.1 Conclusions

In this work, we tackled the problem of matching images of the outdoor environments, which are rich in repeated textures. We analyzed the SuperGlue, the state-of-the-art approach to image matching, and its performance under the given assumptions.

We identified the problems with the existing method for building global context-aware descriptors for keypoints and proposed to modify the objective function of SuperGlue to solve those problems. By adding triplet and pairwise ranking components to the loss functions, we improved the performance of the state-of-the-art method in terms of Matching Score without the loss in the precision of matching.

7.2 Future work

We consider two directions for future work. Firstly, we plan to perform a more thorough evaluation of the improved model on other downstream tasks. Also, we plan measure the effect of proposed improvements on the Structure-from-Motion pipeline.

The other direction involves further improving the state-of-the-art pipeline for image matching. Utilizing global context-aware descriptors is a promising way to finding better image correspondences. We believe that by building more sophisticated network architectures present in the literature and further improving learning objectives, we can obtain even more distinctive keypoints representations.

Bibliography

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features". In: *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*. Ed. by Ales Leonardis, Horst Bischof, and Axel Pinz. Vol. 3951. Lecture Notes in Computer Science. Springer, 2006, pp. 404–417. DOI: 10.1007/11744023_32. URL: https://doi.org/10.1007/11744023_32.

[2] Eric Brachmann and Carsten Rother. "Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 4321–4330. DOI: 10.1109/ICCV.2019.00442. URL: https://doi.org/10.1109/ICCV.2019.00442.

[3] Michael Calonder, Vincent Lepetit, and Pascal Fua. *BRIEF: Binary robust independent elementary features*. 2010.

[4] Gal Chechik et al. "Large Scale Online Learning of Image Similarity Through Ranking". In: *J. Mach. Learn. Res.* 11 (2010), pp. 1109–1135. URL: https://dl.acm.org/citation.cfm?id=1756042.

[5] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al. 2013, pp. 2292–2300. URL: https://proceedings.neurips.cc/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 224–236. DOI: 10.1109/CVPRW.2018.00060. URL: http://openaccess.thecvf.com/content_cvpr_2018_workshops/w9/html/DeTone_SuperPoint_Self-Supervised_Interest_CVPR_2018_paper.html.

[7] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

[8] Mihai Dusmanu et al. "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 8092–8101. DOI: 10.1109/CVPR.2019.00828. URL: http://openaccess.thecvf.com/content_CVPR_2019/html/

Dusmanu_D2-Net_A_Trainable_CNN_for_Joint_Description_and_Detection_of_CVPR_2019_paper.html.

[9] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*. Ed. by Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad. AAAI Press, 1996, pp. 226–231. URL: http://www.aaai.org/Library/KDD/1996/kdd96-037.php.

[10] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692. URL: http://doi.acm.org/10.1145/358669.358692.

[11] Justin Gilmer et al. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1263–1272. URL: http://proceedings.mlr.press/v70/gilmer17a.html.

[12] Qingji Guan et al. *Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification*. 2018. arXiv: 1801.09927 [cs.CV].

[13] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[14] Chris Harris and Mike Stephens. "A combined corner and edge detector". In: *In Proc. of Fourth Alvey Vision Conference*. 1988, pp. 147–151.

[15] Alexander Hermans, Lucas Beyer, and Bastian Leibe. "In Defense of the Triplet Loss for Person Re-Identification". In: *CoRR* abs/1703.07737 (2017). arXiv: 1703.07737. URL: http://arxiv.org/abs/1703.07737.

[16] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[17] Itseez. *Open Source Computer Vision Library*. https://github.com/itseez/opencv. 2015.

[18] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

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

[20] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vis.* 60.2 (2004), pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[21] David G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*. IEEE Computer Society, 1999, pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410. URL: https://doi.org/10.1109/ICCV.1999.790410.

[22] J. Matas et al. "Robust wide baseline stereo from maximally stable extremal regions". In: *In Proc. BMVC*. 2002, pp. 384–393.

[23] Anastasya Mishchuk et al. "Working hard to know your neighbor's margins: Local descriptor learning loss". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 4826–4837. URL: https://proceedings.neurips.cc/paper/2017/hash/831caa1b600f852b7844499430ecac17-Abstract.html.

[24] Niki Parmar et al. "Stand-Alone Self-Attention in Vision Models". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 68–80. URL: https://proceedings.neurips.cc/paper/2019/hash/3416a75f4cea9109507cacd8e2f2aefc-Abstract.html.

[25] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[26] René Ranftl and Vladlen Koltun. "Deep Fundamental Matrix Estimation". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Ed. by Vittorio Ferrari et al. Vol. 11205. Lecture Notes in Computer Science. Springer, 2018, pp. 292–309. DOI: 10.1007/978-3-030-01246-5_18. URL: https://doi.org/10.1007/978-3-030-01246-5_18.

[27] Edward Rosten and Tom Drummond. "Machine learning for high-speed corner detection". In: *In European Conference on Computer Vision*. 2006, pp. 430–443.

[28] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[29] Paul-Edouard Sarlin et al. "SuperGlue: Learning Feature Matching with Graph Neural Networks". In: *CVPR*. 2020. URL: https://arxiv.org/abs/1911.11763.

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[31] Johannes Lutz Schönberger et al. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*. 2016.

[32] Meghna Singh et al. "Optimization of Symmetric Transfer Error for Sub-frame Video Synchronization". In: *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II*. Ed. by David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman. Vol. 5303. Lecture Notes in Computer Science. Springer, 2008, pp. 554–567. DOI: 10.1007/978-3-540-88688-4_41. URL: https://doi.org/10.1007/978-3-540-88688-4_41.

[33] Yurun Tian, Bin Fan, and Fuchao Wu. "L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 6128–6136. DOI: `10.1109/CVPR.2017.649`. URL: `https://doi.org/10.1109/CVPR.2017.649`.

[34] Tinne Tuytelaars and Luc Van Gool. "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions". In: *Proceedings of the British Machine Vision Conference 2000, BMVC 2000, Bristol, UK, 11-14 September 2000*. Ed. by Majid Mirmehdi and Barry T. Thomas. British Machine Vision Association, 2000, pp. 1–14. DOI: `10.5244/C.14.38`. URL: `https://doi.org/10.5244/C.14.38`.

[35] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[36] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 5998–6008. URL: `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

[37] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: `10.21105/joss.03021`. URL: `https://doi.org/10.21105/joss.03021`.

[38] Kwang Moo Yi et al. *Learning to Find Good Correspondences*. 2018. arXiv: `1711.05971 [cs.CV]`.

[39] Barbara Zitová and Jan Flusser. "Image registration methods: a survey". In: *IMAGE AND VISION COMPUTING* 21 (2003), pp. 977–1000.