

A NOVEL LEARNING-BASED IMAGE MATCHING APPROACH BASED ON
MUTUAL NEAREST NEIGHBOR SEARCH WITH RATIO TEST

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

UFUK EFE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

**A NOVEL LEARNING-BASED IMAGE MATCHING APPROACH BASED
ON MUTUAL NEAREST NEIGHBOR SEARCH WITH RATIO TEST**

submitted by **UFUK EFE** in partial fulfillment of the requirements for the degree of
**Master of Science in Electrical and Electronics Engineering Department, Middle
East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. A. Aydın Alatan
Supervisor, **Electrical and Electronics Engineering**

Examining Committee Members:

Prof. Dr. Afşar Saranlı
Department of Electrical and Electronics Engineering, METU

Prof. Dr. A. Aydın Alatan
Department of Electrical and Electronics Engineering, METU

Assoc. Prof. Dr. Elif Vural
Department of Electrical and Electronics Engineering, METU

Assoc. Prof. Dr. Sinan Kalkan
Department of Computer Engineering, METU

Assist. Prof. Dr. Gökhan Koray Gültekin
Department of Electrical and Electronics Engineering, AYBU

Date: 09.09.2021



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ufuk Efe

Signature :

ABSTRACT

A NOVEL LEARNING-BASED IMAGE MATCHING APPROACH BASED ON MUTUAL NEAREST NEIGHBOR SEARCH WITH RATIO TEST

Efe, Ufuk

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. A. Aydın Alatan

September 2021, 66 pages

This thesis proposes a novel image matching method that utilizes learned features extracted by an off-the-shelf deep neural network to obtain a promising performance. The proposed method simply uses a pre-trained VGG architecture as a feature extractor and does not require any additional training to improve matching. Inspired by well-established concepts in the psychology area, such as the Mental Rotation paradigm, an initial warping step is also performed by the help of a preliminary geometric transformation estimate. The matching estimates are based on dense matching using Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT) at the terminal layer of VGG network outputs of the images. After this initial alignment, the same approach is repeated again at every network level between reference and aligned images in a hierarchical manner to reach a good localization and matching performance. By comprehensive experiments, five classical and four learning-based methods in the literature are also compared while optimizing a single parameter, and it is shown that the proposed method achieves the state-of-the-art performance. As a result of a fair comparison, the experimental results on HPatches dataset reveal that the performance gap between classical and learning-based methods is not that signif-

icant as reported in most of the previous studies. Hence, one can conclude that our proposed method, which uses only a pre-trained network and ratio test, outperforms most well-trained learning-based methods.

Keywords: image matching, feature detection, feature description, feature matching, geometric transformation estimation



ÖZ

ORAN TESTİ İLE KARŞILIKLI EN YAKIN KOMŞU ARAMASINA DAYANAN ÖĞRENMEYE DAYALI YENİ BİR GÖRÜNTÜ EŞLEŞTİRME YAKLAŞIMI

Efe, Ufuk

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. A. Aydın Alatan

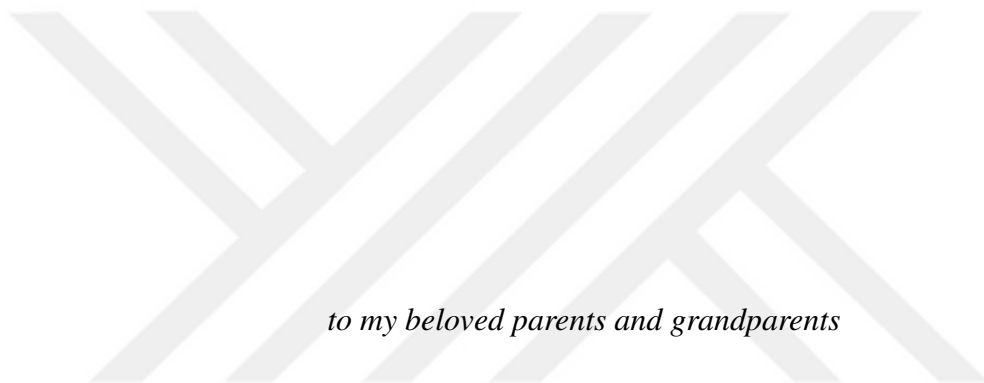
Eylül 2021 , 66 sayfa

Bu tez, kullanıma hazır bir derin sinir ağı tarafından çıkarılan öğrenilmiş özellikleri kullanan ve ümit veren bir performans elde eden yeni bir görüntü eşleştirme yöntemi önermektedir. Önerilen yöntem, özellik çıkarıcı olarak önceden eğitilmiş VGG mimarisini kullanır ve eşleştirmeyi iyileştirmek için herhangi bir ek eğitim gerektirmez. Psikoloji alanındaki Zihinsel Rotasyon paradigması gibi köklü kavramlardan esinlenerek, bir ön geometrik dönüşüm tahmini kullanılarak bir ilk çarpıtma gerçekleştirilir. Bu tahminler basitçe, eşleştirilecek görüntülerin VGG ağ çıktılarının terminal katmanında Çift Yönlü Oran Testi ile Karşılıklı En Yakın Komşu Araması (ÇYOTİ-KEYKA) kullanılarak yoğun eşleştirmeye dayanmaktadır. Bu ilk hizalamadan sonra, iyi bir yerelleştirme ve eşleştirme performansına ulaşmak için referans ve hizalanmış görüntüler arasında hiyerarşik bir şekilde aynı yaklaşım tekrarlanır. Ayrıca kapsamlı deneyler ile literatürde yer alan beş klasik ve dört öğrenme temelli yöntem de tek bir parametre optimize edilerek karşılaştırılmış ve önerilen yöntemin en gelişmiş performansı sağladığı gösterilmiştir. Adil bir karşılaştırmadan sonra, iyi bilinen bir veri seti

olan HPatches üzerindeki deneysel sonuçlar, klasik ve öğrenmeye dayalı yöntemler arasındaki performans farkının önceki çalışmalarda bildirildiği kadar önemli olmadığını ortaya koymaktadır. Genel olarak, yalnızca önceden eğitilmiş bir ağ ve oran testi kullanan önerdiğimiz yöntemimizin, iyi eğitilmiş öğrenme temelli yöntemlerin çoğundan daha iyi performans gösterdiği ve herhangi bir klasik yöntemin parametrelerinin, öğrenmeye dayalı bir teknikle karşılaştırma yapmadan önce dikkatlice analiz edilmesi gerektiği sonucuna varıyoruz.

Anahtar Kelimeler: görüntü eşleme, öznitelik tespiti, öznitelik açıklaması, öznitelik eşleştirme, geometrik dönüşüm tahmini





to my beloved parents and grandparents

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Professor Dr. Aydın Alatan, for his unique advice and guidance during my research. During this process, I learned a lot from his invaluable knowledge, and I had the opportunity to progress skillfully, thanks to his positive approach and time devotion. This thesis would not have been possible without his support.

I want to disclose a great appreciation to Dr. Kutalmış Gökalg İnce, who taught me how to make an excellent high-level design both in terms of writing methodical articles and developing proper codes in this process.

I would like to express my gratitude to Dr. Alper Koz, who was one of the first to teach me how to publish an academic work and whose invaluable technical knowledge I benefited from.

I should like to give a momentous credit to Professor Dr. Afşar Saranlı and Dr. Gökhan Gültekin, who followed my work and helped me to go in the right direction with their questions and advice during the time we spent together while doing projects.

I need to express my special thanks to Alper Kahvecioğlu from ROKETSAN, who provided an environment for my academic studies to be in line with my responsibilities at the company and ensured that all these studies were carried out in an exclusively efficient way.

I want to pay tribute to Bilgen Birkan Kurşun, Kemal Arda Özertem, and Dr. Serdar Çakır from ROKETSAN for the immense patience and tolerance they showed me and the motivation they provided me by appreciating my work.

I want to thank my company ROKETSAN for all its support and funding for this work.

I would like to express my enormous gratitude to Ahmet Arslan, with whom we shared almost all of our time together in work life, lectures, and graduate studies.

Special thanks to İrem Beril Karaçalık, who has not spared her support, faith and trust in me throughout this long journey, and who has lightened my burden in every single difficulty I have experienced.

Finally, I would like to thank my loving mother and father and my dearest elders, who have always been behind me from beginning to end and have substantially contributed every little thing I do.

This work is financially supported by the Scientific and Technological Research Council of Turkey under BİDEB 2210-A program.



TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xii
LIST OF TABLES	xv
LIST OF FIGURES	xvii
LIST OF ALGORITHMS	xxiv
LIST OF ABBREVIATIONS	xxv
0.1 Abbreviations of Technical Content	xxv
0.2 Abbreviations of Conferences and Journals	xxvi
CHAPTERS	
1 INTRODUCTION	1
1.1 Problem Definition	1
1.2 Brief Summary of Related Literature	2
1.3 Contributions	3
1.4 Outline of the Thesis	3
2 RELEVANT IMAGE MATCHING APPROACHES	7

2.1	Related Work on Conventional Approaches	7
2.2	Related Work on Learning-based Approaches	14
2.2.1	Learning-based Approaches Following Classical Pipeline . . .	14
2.2.2	Learning-based Integrated Approaches	17
2.3	Discussion	18
3	PROPOSED METHOD: DEEP FEATURE MATCHING (DFM)	21
3.1	Motivation	21
3.2	Proposed Method	22
3.2.1	Dense Nearest Neighbor Search with Ratio Test	24
3.2.2	Hierarchical Refinement	26
3.2.3	Two Stage Approach	31
4	EXPERIMENTAL EVALUATION	33
4.1	Experimental Procedure	33
4.2	Experimental Setup	34
4.2.1	Dataset	35
4.2.2	Performance Metrics	36
4.2.3	Algorithms	36
4.3	Experimental Results	38
4.3.1	Effect of Matching Threshold	38
4.3.2	Comparisons based on Best Accuracy Values	42
4.3.3	Comparisons based on Best Area Under Accuracy Curves . . .	43
4.3.4	Comparisons based on Execution Time and AUC	45
4.4	Qualitative Results	46

4.5	Discussion	54
5	CONCLUSIONS	57
5.1	Conclusions	57
5.2	Limitations	58
	REFERENCES	59



LIST OF TABLES

TABLES

Table 1.1 Conventional Image Matching Approaches , the conferences or journals in which they were published together with relevant years. For each method, the steps they focus on in the classical image matching pipeline and their input-output relationships are given. Also, for some methods, their well-known names are given in parenthesis.	4
---	---

Table 1.2 State-of-the-art Learning-based Image Matching Approaches , Specifications of Table 1.1 are also valid here.	5
---	---

Table 4.1 Definition of the Threshold Values used in experiments. For all classical algorithms and SuperPoint + NN, we define <i>threshold</i> as the <i>ratio test threshold</i> of MNNSwBRT matcher, where for SuperGlue and Patch2Pix, it is defined as <i>the complement of the confidence</i> . Finally, for DFM algorithm, the threshold is described as the <i>product of the equal ratio test thresholds used for the first three layers</i>	39
---	----

Table 4.2 Best accuracy values and their relevant thresholds for each evaluated algorithm in terms of both MMA and HEA metrics, considering the 1, 3, 5, and 10-pixel thresholds. Best-performing results are shown as bold and the second-best-performing ones are shown as bold-italic.	42
--	----

Table 4.3 **Best area under curve (AUC) percentage values** and their relevant thresholds for each algorithm considering both MMA and HEA metrics from 1 to 10 pixel threshold. Best-performing results are shown as bold, and the second-best-performing ones are shown as bold-italic. Also, the number of detected features, the number of matched features using related threshold values and the average run times are indicated. 43

Table 4.4 **Performances vs. Run Times** for different variants of DFM. Layer-n stands for the matches are from the nth Layer. Layer-1 is identical to the original DFM algorithm used in Table 4.2 and Table 4.3 46



LIST OF FIGURES

FIGURES

Figure 1.1	Classical image matching pipeline	1
Figure 2.1	Feature detection and description procedures of SIFT [8] Algorithm. On top, feature detection in scale space utilizing Difference of Gaussian (DoG) is illustrated, where at the bottom feature description logic with gradients is shown.	8
Figure 2.2	Feature detection and description procedures of SURF [9] Algorithm. Top row illustrates feature detection step with box filters, where second and third rows illustrate feature description exploiting wavelet features.	9
Figure 2.3	Feature detection and description algorithms used by ORB [10] Algorithm. On the left, feature detection with FAST [3] algorithm is illustrated, where on the right feature description with BRIEF [7] Algorithm for different test locations is shown.	10
Figure 2.4	Comparison of Gaussian and nonlinear diffusion scale space used by KAZE [12] Algorithm for varying running times. Top row is acquired by Gaussian convolutions with increasing standart deviations, where second row is constructed by nonlinear diffusion scale space with conductivity function.	10

Figure 2.5	Illustration of <i>MNNSwBRT</i> Algorithm	In both columns, the $M \times N$ distance matrix (denoted as S in Algorithm 1) is shown. Although left and right columns should have the same S matrix, we divide the figure into two columns and highlight different rows and columns to better visualization. On the left, the green match indicates the ordinal of a descriptor on image A and the ordinal of the nearest neighbor of this descriptor. The green match on the right also means the same for image B. Note that these two matches should be mutual. The yellow matches indicate the second-best matches in terms of a distance metric. The best and the second-best values should be sufficiently different from each other. This test is provided by calculating the ratio of the best and second-best matches' distance and looking for if this ratio is less than or equal to a threshold.	12
Figure 2.6	Overall Structure of SuperPoint [33] Algorithm.		14
Figure 2.7	Training strategy of SuperPoint [33] Algorithm.	a) First train MagicPoint [25] detector on a synthetic dataset, b) then use Homographic Adaptation as a self-supervision strategy to create pseudo ground-truth interest point sets, c) finally jointly train SuperPoint network in such a way that to give the keypoint confidence of each pixel and their corresponding descriptor vectors.	15
Figure 2.8	Position of the SuperGlue [41] Algorithm on image matching pipeline.		15
Figure 2.9	The architecture of SuperGlue [41] Algorithm.	SuperGlue basically improves the descriptor vectors considering the cross and self attentions by the help of a Graph Neural Network (GNN). Afterwards, utilizing these new matching descriptors, SuperGlue learns optimally matching the features by the help of the differentiable Sinkhorn algorithm [57], [58]	16

Figure 2.10 **Working Principle of D2-Net [37] Algorithm.** D2-Net first exploits features extracted by a CNN. Next, it performs non-local-maximum suppression on the feature map following by a non-maximum suppression over each descriptor. At the end D2-Net learns better features utilizing keypoint detection scores which are calculated by using α (soft local-maximum score) and β (ratio-to-maximum score). 16

Figure 2.11 **Overview of NCNet [49] Algorithm.** NCNet first performs features extraction using ResNet-101 [60]. Afterwards, NCNet generates the 4D correlation map, and first filter them using soft mutual nearest neighbour search. Then it uses a 4D CNN, namely neighbor consensus network, to output filtered matches between two images. 17

Figure 2.12 **Overview of Patch2Pix [51] Algorithm.** Patch2Pix starts by matching the deepest features of truncated ResNet-34 [60] network by employing NCNet [49] as a matcher, and it continues refining feature locations until the pixel-level by utilizing mid and fine-level regressors which are weakly supervised by epipolar geometry constraints. 18

Figure 3.1 Architecture for Deep Feature Matching (DFM) Algorithm.

Given an image pair, A and B , DFM first extracts deep features for both images using a pre-trained VGG-19 network. In Stage-0, DFM simply performs a coarse transformation estimation, using matches that come from only terminal layers, which have better semantic information. Utilizing this coarse homography, DFM warps the second image and roughly aligns it with the first image. Then, Image A and the warped version of Image B are fed to Stage-1. In Stage-1, DFM again extracts deep features for Image B warped. Then, it continues with Dense Nearest Neighbor Search (DNSS) and Hierarchical Refinement (HRA) steps. DFM utilizes the DNSS block, which is basically Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT), to find the initial matches between two feature maps at the lowest resolution. These initial matches correspond to search areas in the previous layers. For VGG-19, matching areas are 2×2 regions. Then, DNSS is fed by feature vectors inside these matching areas independently. By doing so, DFM restricts the search area and avoids looking for matches at irrelevant locations. 23

Figure 3.2 Visual illustration of Dense Nearest Neighbor Search (DNNS)

at Layer-5. Given the feature maps of F^A and F^B , which are extracted from images A and B , DNNS performs Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT), as presented in Algorithm 1, and computes corresponding matched features at the lowest resolution. These are DFM’s initial matches. The given spatial locations of matched features are the same in the figure for simplicity. . 25

Figure 3.3 Visual illustration of the first step of Hierarchical Refinement Algorithm (HRA).

On the left, matches at Layer-5 and corresponding search spaces at Layer-4 are illustrated. For VGG-19, search spaces are 2×2 regions. Then, DNNS is fed by descriptor vectors inside these search spaces ($F_4^A(\Omega^A)$ and $F_4^B(\Omega^B)$) independently as illustrated on the right. 27

Figure 3.4 **Visual illustration of the first iteration of Hierarchical Refinement Algorithm (HRA).** Inherently, a match in previous layer is expected to be better localized due to the fact that shallow layers have better localization. Black, green and red matches are examples of better localized matches. Besides, thanks to the ratio test, a match may be discarded if its descendants are not discriminative enough. The yellow match is discarded, since any of its descendants find pair in relevant search space. Moreover, if two or more descendants in a matching area find their partners, then a match in the previous layer is propagated. The purple match is an example of this, since it propagated as 2 candidates. Note that, it is also possible the number of propagated candidates as 3 or 4. Finally, DFM continues hierarchical refinement until the 1st layer. 28

Figure 3.5 **Illustration of the effect of Hierarchical Refinement Algorithm (HRA) on matching pairs.** On the left, HRA started with 2085 matched features and 691 of them are correct in terms of matching accuracy under 5-pixel threshold, hence the accuracy is 0.33. Ultimately, HRA ended up with 9002 matched features with 8245 correct matches, therefore the accuracy at the first layer is 0.92. On the right, the behaviours of outliers are investigated. Even if it starts with 0 accuracy, HRA accomplished the task with 0.9 accuracy, and besides, it increased the number of matched features. 30

Figure 3.6 **Effect of two stage approach.** On the left, DFM is enforced to use only Stage-1, where on the right, Stage-0 and Stage-1 are utilized together. Quantitative results with the metrics defined in Chapter 4 are also demonstrated on the top-left corners of the drawings. A green line is for a correctly estimated match whose distance to the groundtruth match is ≤ 5 pixels, where a red line indicates an erroneous match. . . . 32

Figure 4.1 **Example sequences of HPatches [52] dataset.** First two rows are from illumination subset, where others are from viewpoint subset. . 35

Figure 4.2	Experimental Results of 5 classical and 4 learning-based algorithms in terms of Mean Matching Accuracy (MMA) on HPatches [52] dataset. Results are shown as dashed lines for classical algorithms and straight lines for learning-based algorithms. Individual plots show the accuracy with varying threshold values while rows indicate different pixel thresholds.	40
Figure 4.3	Experimental Results of 5 classical and 4 learning-based algorithms in terms of Homography Estimation Accuracy (HEA) on HPatches [52] dataset. Results are shown as stars for classical algorithms and dots for learning-based algorithms. Individual plots show the accuracy with varying threshold values while rows indicate different pixel thresholds. .	41
Figure 4.4	Overall MMA and HEA curves that give the best AUC for each algorithm. These curves indicate the best possible AUC result for each algorithm can achieve on HPatches [52] dataset by only optimizing a single parameter ratio test or match confidence. For each setting that gives the best AUC for one metric, the relevant result of the other metric is also presented.	44
Figure 4.5	Visual Illustration of results of Classical Algorithms SIFT, SURF, ORB, KAZE, and AKAZE optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair <i>books 1 & 3</i>	48
Figure 4.6	Visual Illustration of results of Learning-based Algorithms SuperPoint, SuperGlue, Patch2Pix, and DFM optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair <i>books 1 & 3</i>	49
Figure 4.7	Visual Illustration of results of Classical Algorithms SIFT, SURF, ORB, KAZE, and AKAZE optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair <i>posters 1 & 4</i>	50

Figure 4.8 **Visual Illustration of results of Learning-based Algorithms**
SuperPoint, SuperGlue, Patch2Pix, and DFM optimized for giving
the best MMA (the left column) and the best HEA (the right column)
under 1-pixel threshold on the image pair *posters 1 & 4* 51

Figure 4.9 **Visual Illustration of results of Classical Algorithms SIFT,**
SURF, ORB, KAZE, and AKAZE optimized for giving the best MMA
(the left column) and the best HEA (the right column) under 5-pixels
threshold on the image pair *graffiti 1 & 4* 52

Figure 4.10 **Visual Illustration of results of Learning-based Algorithms**
SuperPoint, SuperGlue, Patch2Pix, and DFM optimized for giving
the best MMA (the left column) and the best HEA (the right column)
under 5-pixels threshold on the image pair *graffiti 1 & 4* 53

LIST OF ALGORITHMS

ALGORITHMS

Algorithm 1	Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT)	13
Algorithm 2	Hierarchical Refinement Algorithm (HRA)	27

LIST OF ABBREVIATIONS

0.1 Abbreviations of Technical Content

AKAZE	Accelerated KAZE
AOS	Additive Operator Splitting
AUC	Area Under Curve
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CNN	Convolutional Neural Network
D2-Net	Detect and Describe Network
DFM	Deep Feature Matching
DLT	Direct Linear Transformation
DNNS	Dense Nearest Neighbor Search
DoG	Difference of Gaussians
FAST	Features from Accelerated Segment Test
FED	Fast Explicit Diffusion
GLOH	Gradient Location and Orientation Histogram
GNN	Graph Neural Network
HEA	Homography Estimation Accuracy
HOG	Histograms of oriented gradients
HPatches	Homography-Patches
HRA	Hierarchical Refinement Algorithm
KAZE	Not an abbreviation, Japanese word means wind
LoG	Laplacian of Gaussian
LPM	Locality Preserving Matching

MMA	Mean Matching Accuracy
MNNS	Mutual Nearest Neighbor Search
MNNS _w BRT	Mutual Nearest Neighbor Search with Bidirectional Ratio Test
MSER	Maximally Stable Extremal Regions
NCNet	Neighbourhood Consensus Networks
OANet	Order-Aware network
OpenCV	Open Source Computer Vision Library
ORB	Oriented FAST and rotated BRIEF
PDE	Partial Differential Equation
RANSAC	RANdom Sample Consensus
ResNet	Residual Neural Network
SAD	Sum of Absolute Differences
SfM	Structure-from-Motion
SIFT	Scale-invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SSD	Sum of Squared Differences
SURF	Speeded Up Robust Features
SUSAN	Smallest Univalued Segment Assimilating Nucleus
VGG	Visual Geometry Group

0.2 Abbreviations of Conferences and Journals

AVC	Alvey Vision Conference
BMVC	British Machine Vision Conference
CACM	Communications of the ACM
CVPR	Conference on Computer Vision and Pattern Recognition
CVPRW	Conference on Computer Vision and Pattern Recognition Work- shops

ECCV	European Conference on Computer Vision
ICCV	International Conference on Computer Vision
IJCV	International Journal of Computer Vision
IMAVIS	Image and Vision Computing
NIPS	Conference on Neural Information Processing Systems
NeurIPS	Conference on Neural Information Processing Systems
PAMI	IEEE Transactions on Pattern Analysis and Machine Intelligence
PR	Pattern Recognition
Proc. IEEE	Proceedings of the IEEE
RA-L	IEEE Robotics and Automation Letters



CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Determining point correspondences between images is one of the vital and well-studied topics in computer vision. These correspondences are crucial for several fields and frequently used in many applications such as image matching, camera relocalization, pose estimation, image retrieval, Simultaneous Localization and Mapping (SLAM), and Structure-from-Motion (SfM).

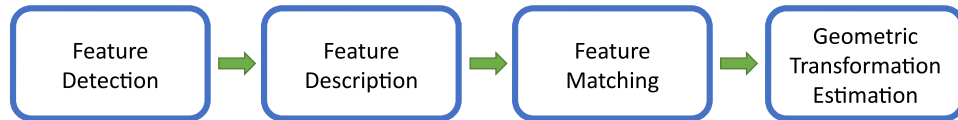


Figure 1.1: **Classical image matching pipeline**

Classical image matching pipeline consists of *Feature Detection*, *Feature Description*, *Feature Matching*, and *Geometric Transformation Estimation* steps as shown in Figure 1.1. Most of the popular algorithms before the deep learning era have focused on one or more phases in this pipeline. During recent years, image matching met deep learning, and a number of valuable efforts have been performed. However, most of these studies still rely on the classical pipeline.

1.2 Brief Summary of Related Literature

In the literature there exists various classical hand-crafted approaches, as well as some recently developed learning-based methods. While many classical techniques [1]–[4] concentrate on Feature Detection, some other approaches [5]–[7] handle Feature Description step locally. The most popular algorithms [8]–[13] work on both Feature Detection and Description. Feature Matching is usually achieved by minimizing/-maximizing some well-defined metrics, such as Sum of Squared Differences (SSD) or correlation, which is next accompanied by outlier rejection methods, such as ratio test of SIFT [8] or some outlier rejection algorithms, such as LPM [14]. In Geometric Transformation Estimation stage, either epipolar geometry or homography is estimated by Direct Linear Transformation (DLT) [15], [16] or RANdom Sample Consensus (RANSAC) [17] based algorithms [18]–[21] commonly.

While many methods [22]–[27] try to improve Feature Detection with deep architectures, some others [28]–[32] reconstruct Feature Descriptors by utilizing learned representations. On the other hand, there are also techniques [33]–[38] that tackle both problems together and jointly try to solve them. Some efforts, as in [39], [40], train the network to learn how to reject outliers in the presence of putative matches. A novel technique, namely SuperGlue [41], learns to make appropriate matches using *matching descriptors* which are obtained by benefiting both original descriptors and keypoint locations via multilayer perceptron. Another approach [42] uses reinforcement learning to optimize feature detection and description for high-level tasks like relative pose estimation. Some recent approaches [43]–[45] try to find differentiable alternatives to RANSAC in order to estimate geometric transformation better. Note that all the aforementioned efforts provide some improvements on some parts of the classical pipeline.

There exist also methods [46]–[48] that handle image matching problem by a single architecture after directly estimating the geometric transformation between two images with regression. More recently, some efforts [49]–[51] find correspondences between two images by benefiting from deep features with a single network instead of separately detect, describe and match them.

Table 1.1 and 1.2 tabulate a list of conventional and learning-based image matching approaches, respectively. These tables summarize the algorithms in terms of the subproblem they are dealing with, such as feature detection, etc. The input-output relationships are also presented to make the functionality of the algorithm clear.

1.3 Contributions

In this thesis, our contribution is twofold;

- i. We propose a novel image matching method that obtains a promising performance by only utilizing features that are extracted by pre-trained VGG-19 network. The proposed method forms a performance baseline for learning-based image matching methods by using off-the-shelf deep feature extractors with no training and only employing standard techniques.
- ii. We perform comprehensive experiments on a well-known dataset HPatches [52] by using five conventional and four learning-based image matching algorithms. We present the effect of parameter optimization on performance results in terms of widely-used metrics Mean Matching Accuracy (MMA) and Homography Estimation Accuracy (HEA).

1.4 Outline of the Thesis

This chapter is devoted defining image matching problem and briefly summarizing the related literature from age-old to state-of-the-art. We also present our contributions to the literature with this thesis.

In Chapter 2, we select the most related techniques and investigate them. Section 2.1 represents the related work on conventional approaches, where Section 2.2 describes the related work on learning-based approaches. Chapter 2 ends with the 2.3 Discussion Section in which we discuss the drawbacks of the existing techniques and the direction of the state-of-the-art.

In Chapter 3, we present our proposed method in detail by providing algorithms and

Table 1.1: **Conventional Image Matching Approaches**, the conferences or journals in which they were published together with relevant years. For each method, the steps they focus on in the classical image matching pipeline and their input-output relationships are given. Also, for some methods, their well-known names are given in parenthesis.

Method	Conference / Journal	Year	Feature Detection	Feature Description	Feature Matching	Geometric Transformation Estimation	Input	Output
A combined corner and edge detector (Harris) [1]	AVC	1988	+				Image	Features
SUSAN—a new approach to low level image processing [2]	IJCV	1997	+				Image	Features
Fusing points and lines for high performance tracking (FAST) [3]	ICCV	2005	+				Image	Features
Robust Wide-baseline Stereo from Maximally Stable Extremal Regions (MSER) [4]	IMAVIS	2004	+				Image	Features
Histograms of oriented gradients for human detection (HOG) [5]	CVPR	2005		+			Features	Descriptors
A performance evaluation of local descriptors (GLOH) [6]	PAMI	2005		+			Features	Descriptors
BRIEF: Binary Robust Independent Elementary Features [7]	ECCV	2010		+			Features	Descriptors
Distinctive image features from scale-invariant keypoints (SIFT) [8]	IJCV	2004	+	+			Image	Features + Descriptors
SURF: Speeded Up Robust Features [9]	ECCV	2006	+	+			Image	Features + Descriptors
ORB: An efficient alternative to SIFT or SURF [10]	ICCV	2011	+	+			Image	Features + Descriptors
BRISK: Binary Robust Invariant Scalable Keypoints [11]	ICCV	2011	+	+			Image	Features + Descriptors
KAZE features [12]	ECCV	2012	+	+			Image	Features + Descriptors
Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces (AKAZE) [13]	BMVC	2013	+	+			Image	Features + Descriptors
Locality Preserving Matching (LPM) [14]	IJCV	2019			+		Matches	Filtered Matches
Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography (RANSAC) [17]	CACM	1981				+	(Filtered) Matches	Homography
Direct Linear Transformation (DLT) [15], [16]	Proc. IEEE	1974				+	(Filtered) Matches	Homography
Matching with PROSAC-progressive sample consensus [18]	CVPR	2005				+	(Filtered) Matches	Homography
MLESAC: A new robust estimator with application to estimating image geometry [19]	IJCV	2019				+	(Filtered) Matches	Homography
Locally Optimized RANSAC (LO-RANSAC) [20]	Joint PR Symposium	2003				+	(Filtered) Matches	Homography
Two-view geometry estimation unaffected by a dominant plane (DEGENSAC) [21]	CVPR	2005				+	(Filtered) Matches	Homography

Table 1.2: **State-of-the-art Learning-based Image Matching Approaches**, Specifications of Table 1.1 are also valid here.

Method	Conference / Journal	Year	Feature Detection	Feature Description	Feature Matching	Geometric Transformation Estimation	Input	Output
TILDE: A Temporally Invariant Learned Detector [22]	CVPR	2015	+				Image	Features
Learning Covariant Feature Detectors [23]	ECCV	2016	+				Image	Features
Quad-Networks: Unsupervised Learning to Rank for Interest Point Detection [24]	CVPR	2017	+				Image	Features
Repeatability Is Not Enough: Learning Affine Regions via Discriminability [26]	ECCV	2018	+				Image	Features
Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters [27]	ICCV	2019	+				Image	Features
Toward geometric deep slam (MagicPoint) [25]	arXiv	2017	+				Image	Features
Learning to Compare Image Patches via Convolutional Neural Networks [28]	CVPR	2015		+			Features	Descriptors
Discriminative Learning of Deep Convolutional Feature Point Descriptors [29]	ICCV	2015		+			Features	Descriptors
PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors [30]	CVPR	2016		+			Features	Descriptors
Working hard to know your neighbor’s margins: Local descriptor learning loss [31]	NIPS	2017		+			Features	Descriptors
SOSNet: Second Order Similarity Regularization for Local Descriptor Learning [32]	CVPR	2019		+			Features	Descriptors
Learning to find good correspondences [39]	CVPR	2018			+		Matches	Filtered Matches
Learning two-view correspondences and geometry using order-aware network (OANet) [40]	ICCV	2019			+		Matches	Filtered Matches
SuperGlue: Learning Feature Matching with Graph Neural Networks [41]	CVPR	2020			+		Features + Descriptors	Matches
SuperPoint: Self-Supervised Interest Point Detection and Description [33]	CVPRW	2018	+	+			Image	Features + Descriptors
LF-Net: Learning Local Features from Images [34]	NeurIPS	2018	+	+			Image	Features + Descriptors
RF-Net: An End-to-End Image Matching Network based on Receptive Field [35]	CVPR	2019	+	+			Image	Features + Descriptors
R2D2: Repeatable and reliable detector and descriptor [36]	NeurIPS	2019	+	+			Image	Features + Descriptors
D2-Net: A Trainable CNN for Joint Description and Detection of Local Features [37]	CVPR	2019	+	+			Image	Features + Descriptors
UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description [38]	arXiv	2020	+	+			Image	Features + Descriptors
Neural-Guided RANSAC: Learning where to sample model hypotheses [43]	ICCV	2019				+	(Filtered) Matches	Homography
DSAC-Differentiable RANSAC for Camera Localization hypotheses [44]	CVPR	2017				+	(Filtered) Matches	Homography
Expert Sample Consensus Applied to Camera Re-Localization hypotheses [45]	ICCV	2019				+	(Filtered) Matches	Homography
Deep Image Homography Estimation [46]	arXiv	2016				+	Image Pairs	Homography
Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model [47]	RA-L	2018				+	Image Pairs	Homography
Convolutional neural network architecture for geometric matching [48]	CVPR	2017				+	Image Pairs	Homography
Neighbourhood Consensus Networks (NCNet) [49]	NeurIPS	2018	+	+	+		Image Pairs	Matches
Efficient Neighbourhood Consensus Networks via Sub-manifold Sparse Convolutions (Sparse NCNet) [50]	ECCV	2020	+	+	+		Image Pairs	Matches
Patch2Pix: Epipolar-Guided Pixel-Level Correspondences [51]	CVPR	2021	+	+	+		Image Pairs	Matches

figures, and explained our motivation to develop this technique.

Chapter 4, exhibits the experimental evaluation procedure, introducing the experimental setup, dataset, performance metrics, and algorithms used. Then, we show experimental results with relevant comparisons.

In Chapter 5, we first summarize our works, then made conclusions on the results and approaches. Finally, we argue the limitations of our proposed method and indications from experiments.



CHAPTER 2

RELEVANT IMAGE MATCHING APPROACHES

In this part of the thesis, the most relevant techniques from Chapter 1 are selected and examined carefully. For this purpose, the following conventional (i.e. hand-crafted) approaches are selected, as SIFT [8], SURF [9], ORB [10], KAZE [12], AKAZE [13], whereas the selected learning-based methods are SuperPoint [33], SuperGlue [41], D2-Net [37], NCNet [49] and Patch2Pix [51].

2.1 Related Work on Conventional Approaches

The most well-known conventional algorithms, SIFT, SURF, ORB, KAZE and AKAZE execute the initial two steps of the classical image matching pipeline, namely feature detection and description, in a hand-crafted manner, and return the locations of detected points together with their corresponding descriptors.

SIFT (Scale-Invariant Feature Transform) [8] detects features in the scale-space simply utilizing Difference of Gaussians (DoG), which is an approximation of Laplacian of Gaussian (LoG) [53]. Hence, SIFT is able to detect blobs in varying scales with their orientation. Next, considering the dominant orientation and creating gradient orientation histograms, SIFT outputs 128-dimensional descriptor vectors. SIFT is invariant to some amount of scale changes and even severe rotations; furthermore, it is robust to illumination changes due to the normalization of the descriptor vector.

SURF (Speeded Up Robust Features) [9] goes one step further than the SIFT and approximates LoG with low-complexity Box Filters. SURF uses a blob detector based on the Hessian matrix to find points of interest. The determinant of the Hessian matrix

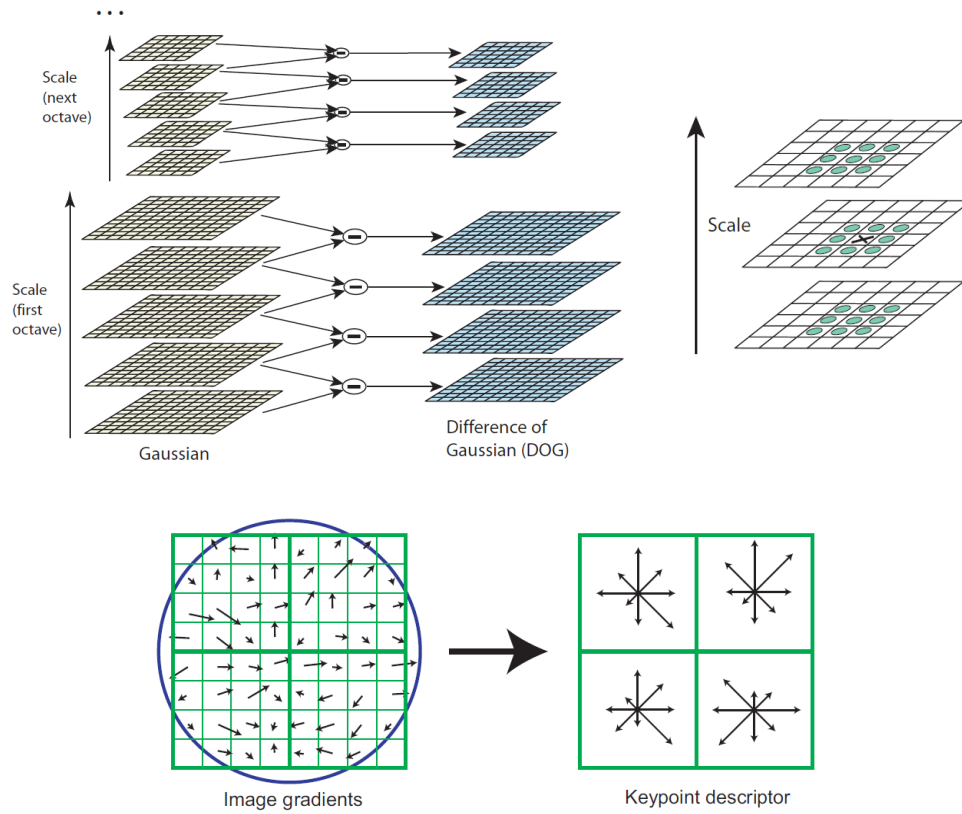


Figure 2.1: **Feature detection and description procedures of SIFT [8] Algorithm.**

On top, feature detection in scale space utilizing Difference of Gaussian (DoG) is illustrated, where at the bottom feature description logic with gradients is shown.

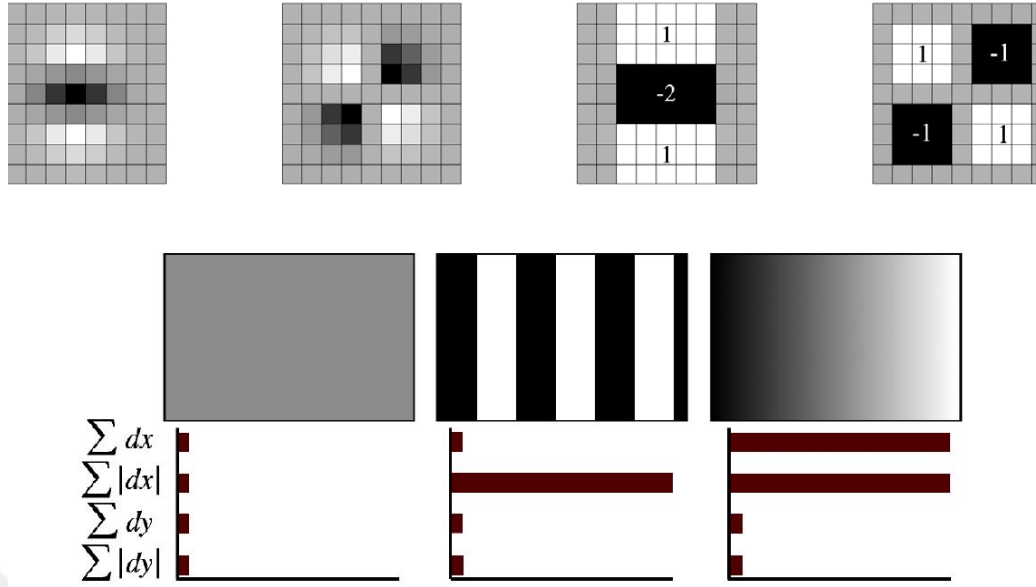


Figure 2.2: **Feature detection and description procedures of SURF [9] Algorithm.** Top row illustrates feature detection step with box filters, where second and third rows illustrate feature description exploiting wavelet features.

is used as a measure of local change around the point, and as a consequence, the points which maximize that determinant are selected. SURF also takes into account the dominant orientation of the features and exploits Haar wavelet responses in the horizontal and vertical directions to extract 64-dimensional descriptor vectors. SURF is also robust to some amount of scale, rotation, and illumination changes.

ORB (Oriented FAST and Rotated BRIEF) [10] runs well-known FAST (Features from Accelerated Segment Test) algorithm [3] as a feature point detector together with computing keypoint's orientation. FAST algorithm basically examines a circle of 16 pixels surrounding an arbitrary pixel and decides whether the pixel is a keypoint or not by using the intensity differences between the candidate pixel and 16 surrounding pixels. Then, using the computed orientation, ORB employs a rotated version of the BRIEF (Binary Robust Independent Elementary Features) [7] algorithm, which simply creates a binary feature vector of the binary test responses to build descriptor vectors.

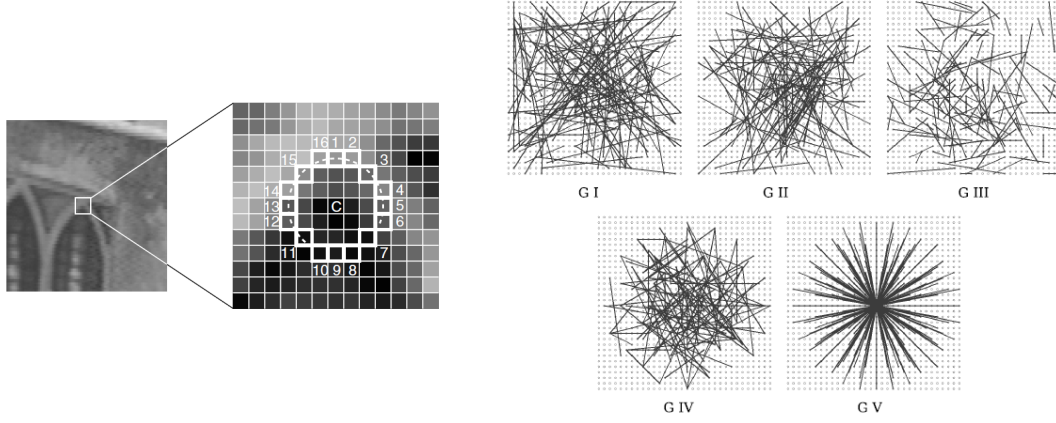


Figure 2.3: **Feature detection and description algorithms used by ORB [10] Algorithm.** On the left, feature detection with FAST [3] algorithm is illustrated, where on the right feature description with BRIEF [7] Algorithm for different test locations is shown.

KAZE [12] detector uses nonlinear scale spaces instead of Gaussian scale-space representations that are employed by SIFT by the following motivation; nonlinear scale-space considers natural boundaries of objects, unlike Gaussian scale space, which does not consider them due to smoothing the intensity details as well as noise at all scale levels to the same degree. In contrast to SIFT, which uses DoG to process the

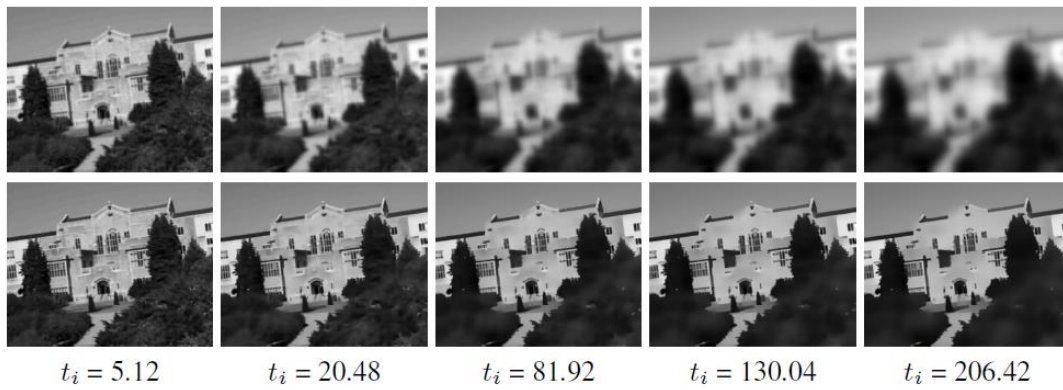


Figure 2.4: **Comparison of Gaussian and nonlinear diffusion scale space used by KAZE [12] Algorithm for varying running times.** Top row is acquired by Gaussian convolutions with increasing standart deviations, where second row is constructed by nonlinear diffusion scale space with conductivity function.

blurred images, KAZE uses AOS (Additive Operator Splitting) schemes, since there are no analytical solution of the partial differential equations (PDEs) for nonlinear diffusion filtering. KAZE uses an adapted version of the SURF descriptor. Since the descriptors must work in a nonlinear scale-space model, the derivative responses are calculated and summed into a feature descriptor vector, and then the vector is centered at the feature. Finally, the descriptor is normalized into a unit vector.

AKAZE [13] is the accelerated version of the KAZE algorithm. It benefits from FED (Fast Explicit Diffusion) in the detection step and uses Modified-Local Difference Binary (M-LDB) descriptor, which is a modified version of the original LDB descriptor [54], to create the feature descriptor vectors.

After features are detected and described with a feature extractor algorithm, they should be matched between frames. For this purpose, *Mutual Nearest Neighbor Search (MNNS)* algorithm, which is executed by measuring distances between descriptor vectors, is commonly used for feature matching. The most common distance metrics are SAD (Sum of Absolute Differences), SSD (Sum of Squared Differences), and Hamming distance. Moreover, while searching for nearest neighbors, Lowe [8] proposed a method to reject ambiguous matches by thresholding the ratio of the distance of the closest match to the distance of the second closest one. After including this extension, the whole feature matching strategy is denoted as *Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT)* which is presented as Algorithm 1 and illustrated in Figure 2.5.

As presented in Algorithm 1, MNNSwBRT Algorithm takes two sets of descriptor vectors (D^A and D^B) and ratio test threshold, then finds matches between these two sets first by looking for mutual nearest neighbors, and later by filtering matches with ratio test threshold in a bidirectional manner. In the end, MNNSwBRT returns a set of matched pairs ($P^{A,B}$) for *imageA* and *imageB*.

Figure 2.5, illustrates the working principle of the MNNSwBRT Algorithm for two sets of descriptor vectors where the first set has M and the second set has N descriptor vectors. This figure shows how the ratio test threshold is utilized by using the distances of the best and the second-best nearest neighbors.

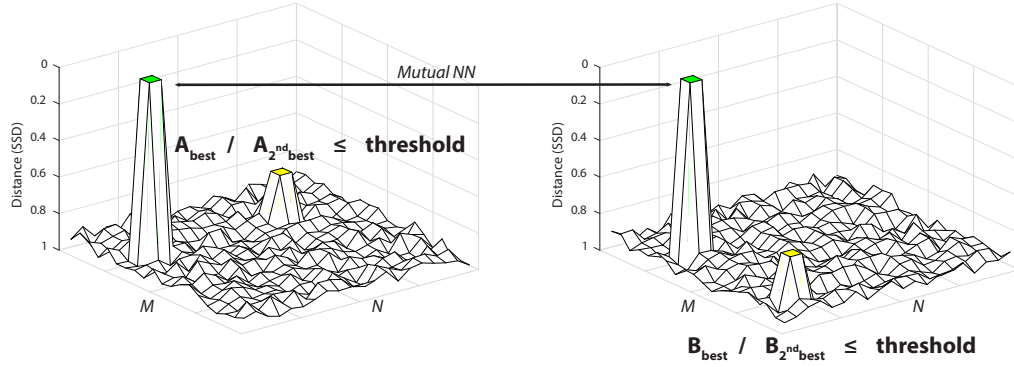


Figure 2.5: **Illustration of *MNNSwBRT* Algorithm** In both columns, the $M \times N$ distance matrix (denoted as S in Algorithm 1) is shown. Although left and right columns should have the same S matrix, we divide the figure into two columns and highlight different rows and columns to better visualization. On the left, the green match indicates the ordinal of a descriptor on image A and the ordinal of the nearest neighbor of this descriptor. The green match on the right also means the same for image B. Note that these two matches should be mutual. The yellow matches indicate the second-best matches in terms of a distance metric. The best and the second-best values should be sufficiently different from each other. This test is provided by calculating the ratio of the best and second-best matches' distance and looking for if this ratio is less than or equal to a threshold.

Algorithm 1: Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT)

inputs : $D^A, D^B \rightarrow$ sets of descriptor vectors, $thr \rightarrow$ ratio threshold

outputs: $P^{A,B} \rightarrow$ set of matched pairs for *imageA* and *imageB*

Function MNNSwBRT (D^A, D^B) :

(0) Assume D^A has M and D^B has N descriptor vectors

(1) Compute Sum of Squared Differences (SSD) between all descriptor vectors in D^A and D^B

$$S_{ij} = \text{SSD}(D_i^A, D_j^B) \text{ for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N$$

(2) Find Nearest Neighbors (NN) for A and B by picking smallest and 2nd smallest SSD scores for each rows and columns

$$(NN)_{ij}^A = \begin{cases} 1 & \text{if } S_{ij} = \min(S_{i:}) \\ 2 & \text{if } S_{ij} = 2^{\text{nd}} \min(S_{i:}) \\ 0 & \text{otherwise} \end{cases} \quad (NN)_{ij}^B = \begin{cases} 1 & \text{if } S_{ij} = \min(S_{:j}) \\ 2 & \text{if } S_{ij} = 2^{\text{nd}} \min(S_{:j}) \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq M$ and $1 \leq j \leq N$

(3) Check if these matches are Mutual Nearest Neighbors (MNN)

$$(MNN)_{ij} = \begin{cases} 1 & \text{if } (NN)_{ij}^A \& (NN)_{ij}^B = 1 \\ 2 & \text{if } (NN)_{ij}^A \parallel (NN)_{ij}^B = 2 \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq M$ and $1 \leq j \leq N$

(4) Apply Ratio Test (RT) by dividing the smallest distance to 2nd smallest distance

$$(MNNwRT)_{ij}^A = \begin{cases} 1 & \text{if } S_{i:}(\text{ind}((MNN)_{i:} == 1)) / S_{i:}(\text{ind}((MNN)_{i:} == 2)) \leq thr \\ 0 & \text{otherwise} \end{cases}$$

$$(MNNwRT)_{ij}^B = \begin{cases} 1 & \text{if } S_{:j}(\text{ind}((MNN)_{:j} == 1)) / S_{:j}(\text{ind}((MNN)_{:j} == 2)) \leq thr \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq M$ and $1 \leq j \leq N$

(5) Check for Bidirectional Ratio Test (BRT) and assign indices where the value is 1 as $P^{A,B}$

$$(MNNwBRT)_{ij} = \begin{cases} 1 & \text{if } (MNNwRT)_{ij}^A \& (MNNwRT)_{ij}^B = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$P^{A,B} = \text{ind}(MNNwBRT == 1), \text{ for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N$$

return $P^{A,B}$

end

2.2 Related Work on Learning-based Approaches

Classical Image Matching Pipeline consists of feature detection, feature description, feature matching, and geometric transformation estimation stages, as mentioned before. We investigate learning-based approaches in two subtitle in terms of following the classical image matching pipeline or not.

2.2.1 Learning-based Approaches Following Classical Pipeline

SuperPoint [33] jointly detects keypoints and computes relevant descriptor vectors. SuperPoint utilizes MagicPoint [25] detector, which is trained on a synthetic dataset and uses Homographic Adaptation as a self-supervision strategy to create pseudo ground-truth interest point sets on MS-COCO [55] dataset. Then, SuperPoint network is jointly trained in such a way that giving the keypoint confidence of each pixel and their corresponding descriptor vectors. Jointly training of this detector-descriptor network means that partly sharing parameters. This kind of combined training certainly improves the overall robustness. However, SuperPoint has limitations under some challenges, such as in-plane rotations. The authors in [56] show that due to the trade-off between invariance and discrimination, we can not enforce the network to be more rotation and illumination invariant.

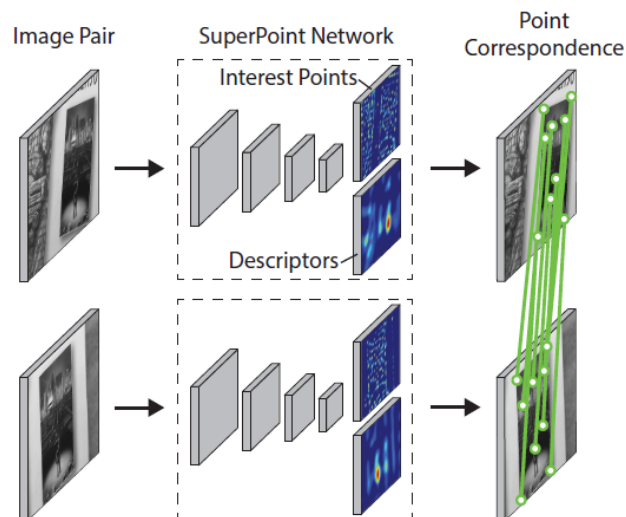


Figure 2.6: Overall Structure of SuperPoint [33] Algorithm.

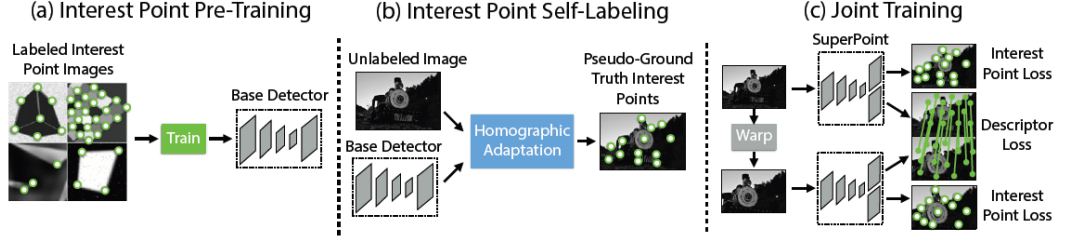


Figure 2.7: **Training strategy of SuperPoint [33] Algorithm.** a) First train Magic-Point [25] detector on a synthetic dataset, b) then use Homographic Adaptation as a self-supervision strategy to create pseudo ground-truth interest point sets, c) finally jointly train SuperPoint network in such a way that to give the keypoint confidence of each pixel and their corresponding descriptor vectors.

SuperGlue [41] can be considered as a feature matcher that computes the matches between two sets consisting of detected features and corresponding descriptor vectors. SuperGlue gets the detected features and their descriptors as inputs, then uses a Graph Neural Network (GNN) to find out cross and self attentions between features. Afterwards, this method learns optimally matching the features by the help of the differentiable Sinkhorn algorithm [57], [58]. SuperGlue argues that it could remarkably improve the matching performances of extracted features. However, its performance depends on the quality of feature detectors and descriptors in principle. Moreover, when the detector and descriptor couple change, it should be retrained to get the best performance.

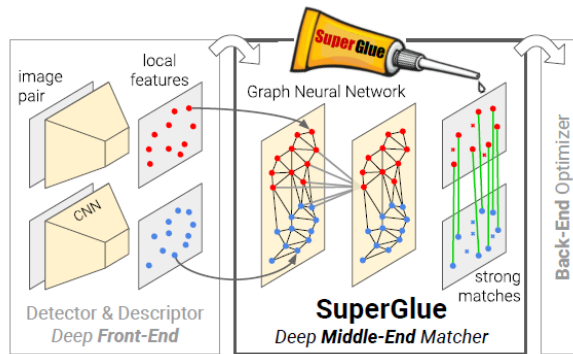


Figure 2.8: **Position of the SuperGlue [41] Algorithm on image matching pipeline.**

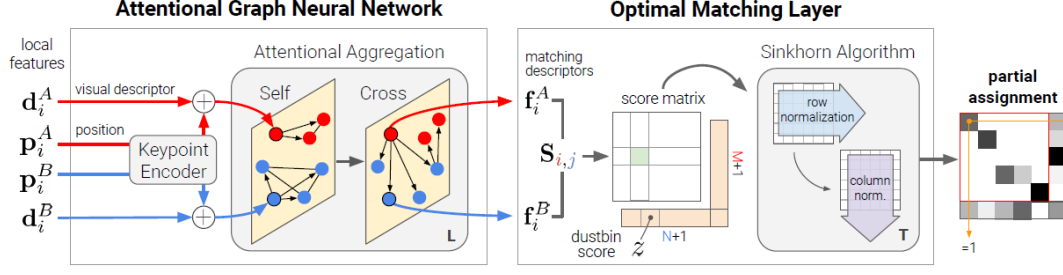


Figure 2.9: **The architecture of SuperGlue [41] Algorithm.** SuperGlue basically improves the descriptor vectors considering the cross and self attentions by the help of a Graph Neural Network (GNN). Afterwards, utilizing these new matching descriptors, SuperGlue learns optimally matching the features by the help of the differentiable Sinkhorn algorithm [57], [58]

D2-Net [37], on the other hand, uses feature maps of VGG-16 [59] to find salient points and their descriptors. Although D2-Net can work with no training, the authors argue that with well-defined losses, it can be trained and might perform better. Since D2-Net uses only the specific layers of the feature extractor network, it can not utilize all the information encoded in the feature extractor. Moreover, it tends to poorly localize the features on account of using features at a low resolution.

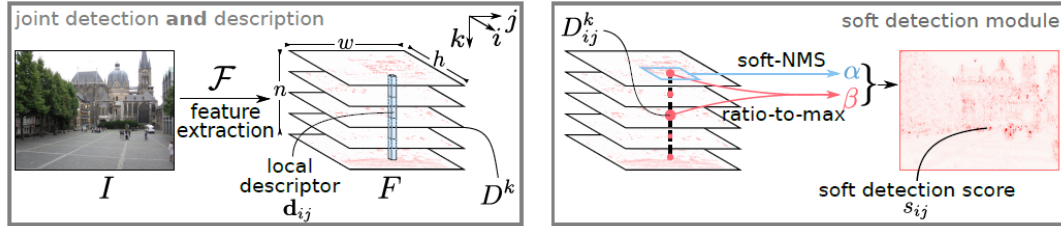


Figure 2.10: **Working Principle of D2-Net [37] Algorithm.** D2-Net first exploits features extracted by a CNN. Next, it performs non-local-maximum suppression on the feature map following by a non-maximum suppression over each descriptor. At the end D2-Net learns better features utilizing keypoint detection scores which are calculated by using α (soft local-maximum score) and β (ratio-to-maximum score).

2.2.2 Learning-based Integrated Approaches

In the integrated approaches, the algorithms only take two images as input and directly compute the correspondences between them without externally detect, describe, and match features in subsequent steps.

NCNet [49], as a typical example, takes advantage of dense CNN features extracted by ResNet-101 [60]. Next, this algorithm [49] generates the 4D correlation map, which will be processed later by a 4D CNN, namely neighbor consensus network. Finally, the algorithm outputs filtered matches between two images. NCNet algorithm is a novel idea and it works well especially in semantic matching scenarios; however, the localization performance is not good, since NCNet works in low resolutions due to the large memory requirement.

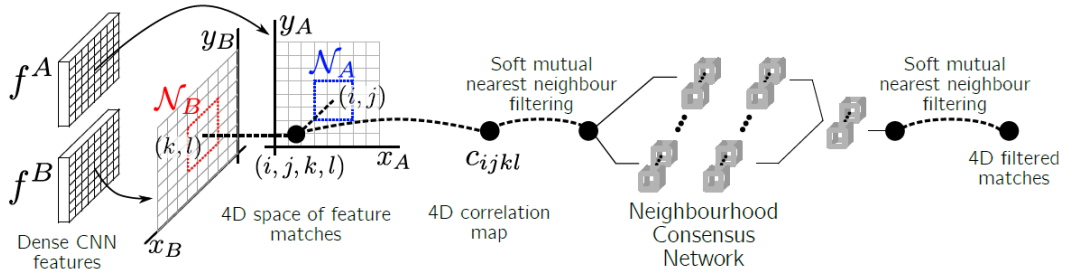


Figure 2.11: **Overview of NCNet [49] Algorithm.** NCNet first performs features extraction using ResNet-101 [60]. Afterwards, NCNet generates the 4D correlation map, and first filter them using soft mutual nearest neighbour search. Then it uses a 4D CNN, namely neighbor consensus network, to output filtered matches between two images.

Patch2Pix [51], as an alternative integrated approach, uses NCNet as a baseline and, starting from local patches, it predicts pixel-level matches by using epipolar loss. The algorithm first extracts features by adapted ResNet-34 [60] until the 4th convolutional layer, detects match proposals using the last layer and finally, refines these proposals by using mid and fine-level regressors. However, this approach does not entirely benefit from hierarchical refinement and coarse warping stages, which are well-known procedures in vision research. Moreover, Patch2Pix does not use the very last layers of ResNet-34 in which more semantic information is encoded, and requires training,

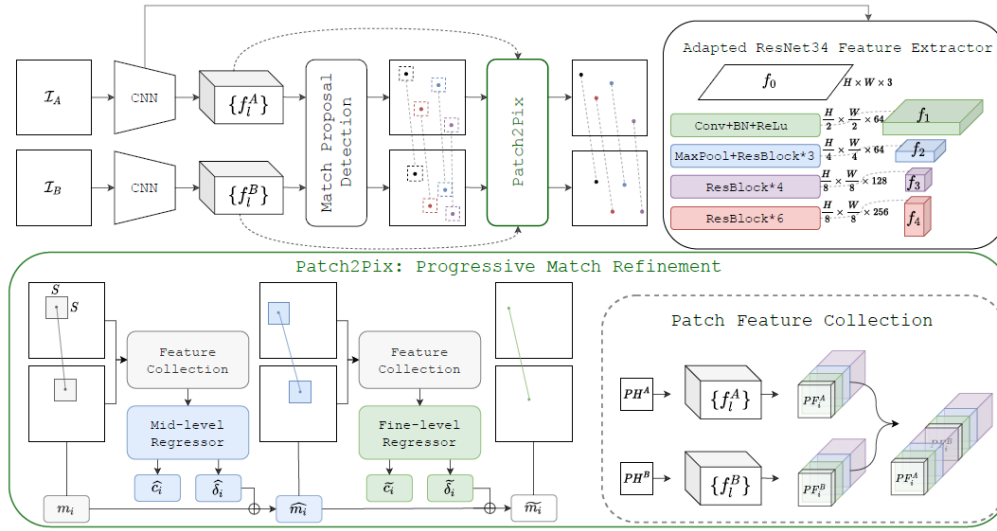


Figure 2.12: **Overview of Patch2Pix [51] Algorithm.** Patch2Pix starts by matching the deepest features of truncated ResNet-34 [60] network by employing NCNet [49] as a matcher, and it continues refining feature locations until the pixel-level by utilizing mid and fine-level regressors which are weakly supervised by epipolar geometry constraints.

whenever the feature extractor backbone change. In their ablation study, Patch2Pix clearly presents that starting with the last layer and using the information of all layers before the last layer provides a better performance compared to using some previous layers or using all layers containing the last one. These experimental results also indicate the possible advantages of hierarchical refinement that is used in our proposed method.

It should be noted that while Patch2Pix and NCNet directly output the putative matches, SuperPoint and D2-Net require a feature matcher at its final stage as in classical algorithms; this matcher step might be either MNNSwBRT or SuperGlue.

2.3 Discussion

The performance of the overall conventional system is dictated by the weakest sub-block in this serial pipeline during image matching. Therefore, trying to enhance the performance of each stage would be sub-optimal as the output of each block is the

input to another, and every single step is prone to error in a different scenario.

Hence, it can be observed that some recent works [49], [51] in the literature have abandoned the classical pipeline and started to propose combining many stages together in their solutions to overcome this bottleneck.

It should be noted that the prior conventional matching approaches have many powerful techniques to ensure high-quality matching with hand-crafted features. Although deep features improve the discriminative performance over conventional ones, the utilized matching techniques, such as MNNSwBRT, can still be used with off-the-shelf deep feature extractors.





CHAPTER 3

PROPOSED METHOD: DEEP FEATURE MATCHING (DFM)

3.1 Motivation

The representations that are obtained by off-the-shelf feature extractors are shown to be quite effective for complex tasks, such as object detection [61]–[63] and semantic segmentation [64]; however, according to our knowledge, currently the only method employing off-the-shelf feature extractor is D2-Net [37]. In other words, the requirement of designing new deep networks for image matching is not clearly justified by demonstrating the insufficiency of off-the-shelf feature extractors as they are not examined in detail. On the contrary, it is firmly believed that the human-level performance of these feature extractors for image classification task [65] indicates their semantic abstraction capability. Moreover, their success in object detection and semantic segmentation tasks reveals their good localization capability. If these pre-trained extractors are utilized in a way for exploiting both their semantic abstraction and good localization capabilities, they might have a chance to achieve satisfactory performance for image matching task as well. If these pre-trained extractors are utilized in such a way exploiting both their semantic abstraction and good localization capabilities, these extractors still might have a chance to achieve satisfactory performance for image matching task.

Recognition-by-components theory [66] advocates that our way of perception of objects are separating them into parts, which supports that humans might accomplish image matching task using objects as they are the main components of images. We have also been inspired by the Mental Rotation [67] which states that the human brain does some initial rotations while looking for similarities between two objects.

Hence, our main motivation is to reveal the performance of a pre-trained off-the-shelf feature extractor on image matching problem by applying simple, but effective vision techniques, such as hierarchical search, coarse-to-fine strategy or preliminary alignment, since human visual system possibly employs these fundamental approaches.

3.2 Proposed Method

We propose a method that is able to find high-quality correspondences between images, while not requiring any training step. The proposed method only needs pre-trained VGG [59] network to extract its features from images. We first leverage the invariances of the deepest (i.e. final) layers to find matches between images at a more semantic level. As a result of this initial matching, we might easily obtain a rough estimate of the geometric transformation between the images to apply an initial warping, as in Mental Rotation phenomena [67]. After warping one of the images by the estimated transformation, we start performing an hierarchical matching, which is a popular technique, from the deepest level of the feature map to the shallowest layer. With this coarse to fine strategy, we basically exploit the properties of all layers together, such as the powerful semantic property of the deep layers, as well as the precise localization of the shallow layers.

In Figure 3.1, our proposed architecture to perform matching between two images is shown. Our key idea is to extract features by a pre-trained VGG-19 network, aligning images with rough geometric transformation estimation then applying a coarse to fine strategy for better localization. Opposing to the classical feature matching pipeline, rather than constraining the set of feature points by a detection step, we exploit the feature points, which are successfully matched, as in [49], [51].

In Stage-0 of Figure 3.1, we perform *Dense Nearest Neighbor Search* (DNNS) in low spatial resolution to obtain a rough estimation of the geometric transformation between given images. This dense search is still achievable in terms of complexity, since it is only applied at the final layers with small number of outputs. Using this transformation estimate, we warp the second image to form the warped-image that we use in the next stage.

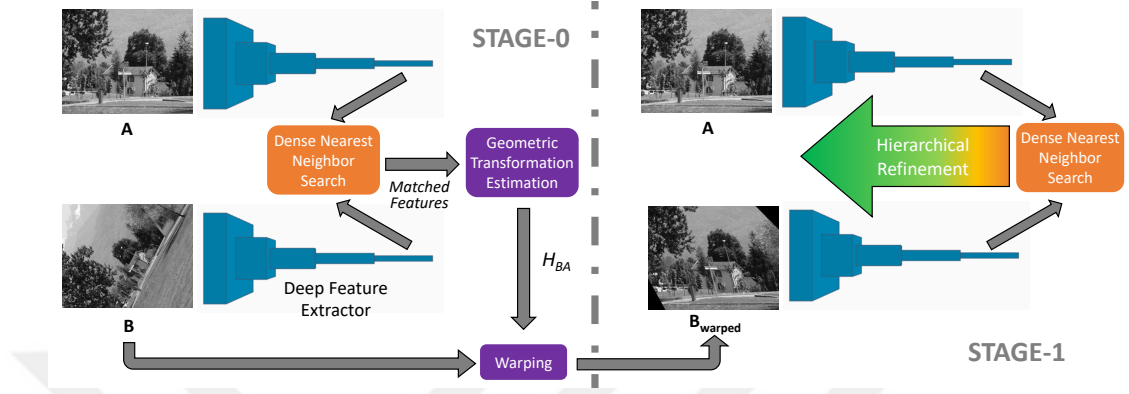


Figure 3.1: **Architecture for Deep Feature Matching (DFM) Algorithm.** Given an image pair, A and B, DFM first extracts deep features for both images using a pre-trained VGG-19 network. In Stage-0, DFM simply performs a coarse transformation estimation, using matches that come from only terminal layers, which have better semantic information. Utilizing this coarse homography, DFM warps the second image and roughly aligns it with the first image. Then, Image A and the warped version of Image B are fed to Stage-1. In Stage-1, DFM again extracts deep features for Image B warped. Then, it continues with Dense Nearest Neighbor Search (DNSS) and Hierarchical Refinement (HRA) steps. DFM utilizes the DNSS block, which is basically Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT), to find the initial matches between two feature maps at the lowest resolution. These initial matches correspond to search areas in the previous layers. For VGG-19, matching areas are 2x2 regions. Then, DNSS is fed by feature vectors inside these matching areas independently. By doing so, DFM restricts the search area and avoids looking for matches at irrelevant locations.

In Stage-1 of Figure 3.1, we first perform DNNS using the feature maps of the reference image and the warped image at the last layers. Starting from the coarse matches of the last layer, we refine and propagate these initial matches hierarchically by moving to the finer resolutions at each step.

Overall, given an image pair, A and B as in Figure 3.1, we first perform deep feature extraction for both images using a pre-trained VGG-19 network. This network is not fine-tuned by any matching cost, as applied in many other state-of-the-art techniques. Then, we continue with a) Dense Nearest Neighbor Search and b) Hierarchical Refinement, which are explained in the following chapters in detail. After explaining these two basic steps, we next present the two-stage architecture and how these basic steps are employed.

3.2.1 Dense Nearest Neighbor Search with Ratio Test

Dense Nearest Neighbor Search (DNNS) technique simply searches for matches in dense feature maps by using a mutual nearest neighbor search with a bidirectional ratio test (Algorithm 1) [8], as employed in [37]. Given the feature patches F^A and F^B extracted from images A and B , DNNS searches F^B to find the best match for each element of F^A . The potential matches are defined as the nearest neighbors in terms of L_2 distance. For a point p^A in feature map F^A , if the ratio of the distance to best match p^B and the second-best match is below the given threshold, then the point p^A is matched to p^B . However, the pair is accepted only if the match is mutual, i.e., p^A and p^B are returned as a matched pair if p^B is matched to p^A as well.

As shown in Figure 3.2, we first employ DNNS to initialize matches at the terminal layer of the feature extractor. These are DFM’s initial matches, and they will be further processed. Note that, downsampled size of this layer makes them suitable for DNSS. For VGG feature extractor, the size of the feature maps at the 5th convolutional layer is $W/16 \times H/16$ where W and H are the original width and height of the image, which makes DNNS feasible in terms of search complexity. We also employ DNNS to refine the matches at finer spatial resolutions, as explained next.

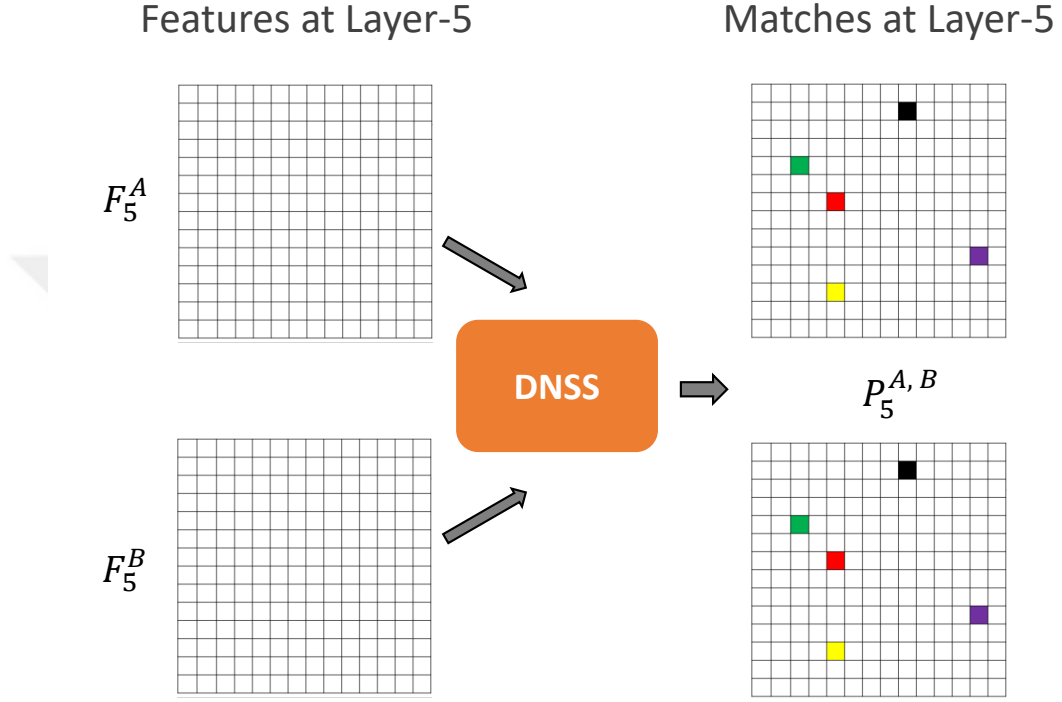


Figure 3.2: **Visual illustration of Dense Nearest Neighbor Search (DNNS) at Layer-5.** Given the feature maps of F^A and F^B , which are extracted from images A and B , DNNS performs Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT), as presented in Algorithm 1, and computes corresponding matched features at the lowest resolution. These are DFM’s initial matches. The given spatial locations of matched features are the same in the figure for simplicity.

3.2.2 Hierarchical Refinement

The output features of DNNS at the final layers are robust to geometric transformations and illumination changes, but they are poorly localized. Directly upsampling them to their original resolution is also expected to yield poor matching performance.

As an imperfect alternative, DNNS can also be employed in former convolutional layers without any constraint. However, finer spatial resolutions of these layers would increase the computational complexity exponentially as we move towards the original resolution. Moreover, the features at deeper layers tend to carry semantic information more, which makes them more discriminative, while the features in shallower layers have smaller receptive fields and therefore tend to be more repetitive. Hence, applying DNNS at these layers brings the risk of mismatches and erroneous elimination of feature points. Patch2Pix [51] solves the latter problem by appending the features of different layers in the original resolution at the expense of computational complexity, which worsens the first problem more. However, coarse-to-fine strategy (or hierarchical search/refinement), which is a well-known technique in computer vision and practiced in many algorithms [68], [69], can be employed to solve both problems at the same time. A similar hierarchical matching strategy is also used in [70].

As presented in Algorithm 2, given a set of matched pairs, $P_n^{A,B}$ at layer n and feature maps F_{n-1}^A and F_{n-1}^B at the layer $n - 1$, Hierarchical Refinement Algorithm (HRA) finds the matched points at the current layer by employing DNNS in a limited area. For each matched pair, p^A and p^B , HRA first constructs point sets Ω_A and Ω_B which are the corresponding search spaces of the p^A and p^B at layer $n - 1$, respectively. For VGG-19, these search spaces can be easily obtained by upsampling the input points by 2 and including the one-pixel right and bottom neighborhood for each upsampled point. By restricting the search area as such, DNNS can be applied with none of the aforementioned problems. Therefore, given a set of matched pairs in layer n , for each pair, we feed the patches of feature maps F^A and F^B to DNNS and get the matched pairs in layer $n - 1$.

Algorithm 2: Hierarchical Refinement Algorithm (HRA)

input : $F_{n-1}^A, F_{n-1}^B \rightarrow$ feature maps at layer $n - 1$ $P_n^{A,B} \rightarrow$ set of matched pairs at layer n

output: $P_{n-1}^{A,B} \rightarrow$ set of matched pairs at layer $n - 1$

Function $\text{HRef}(F_{n-1}^A, F_{n-1}^B, P_n^{A,B}) :$

```
for  $p^A, p^B$  in  $P_n^{A,B}$  do
  (1) Define 2x2 search spaces at layer  $n - 1$  for points at layer  $n$ 
     $\Omega^A = \text{search\_space}(p^A), \Omega^B = \text{search\_space}(p^B)$ 
  (2) Perform Dense Nearest Neighbor Search
     $M^{A,B} = \text{DNNS}(F_{n-1}^A(\Omega^A), F_{n-1}^B(\Omega^B))$ 
  (3) Record the matched pair at layer  $n - 1$ 
     $P_{n-1}^{A,B}.append(\text{transform}(M^{A,B}))$ 
end
return  $P_{n-1}^{A,B}$ 
```

end

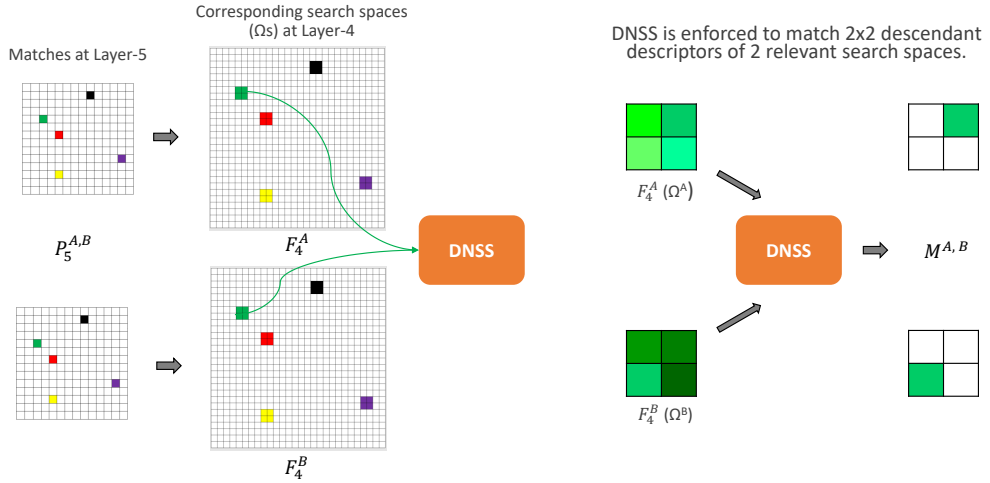


Figure 3.3: **Visual illustration of the first step of Hierarchical Refinement Algorithm (HRA).** On the left, matches at Layer-5 and corresponding search spaces at Layer-4 are illustrated. For VGG-19, search spaces are 2x2 regions. Then, DNNS is fed by descriptor vectors inside these search spaces ($F_4^A(\Omega^A)$ and $F_4^B(\Omega^B)$) independently as illustrated on the right.

The first step of the HRA is illustrated in Figure 3.3. Given the initial matches from DNNS block $P_5^{A,B}$, for VGG these initial matches are at the 5th-Layer and feature maps F_4^A and F_4^B at layer 4, Hierarchical Refinement Algorithm (HRA) finds the matched points at the current layer by employing DNNS in a limited search space (Ω). On the left of the Figure 3.3, matches at Layer-5 and corresponding search spaces at Layer-4 are illustrated. For VGG-19, search spaces are 2x2 regions. Then, DNNS is fed by descriptor vectors inside these search spaces ($F_4^A(\Omega^A)$ and $F_4^B(\Omega^B)$) independently as illustrated on the right of the Figure 3.3. By doing so, we restrict the search area, and avoid looking for matches at irrelevant locations.

As illustrated in Figure 3.4, HRA changes the number of feature points in each refinement step by selecting the correct matches and rejecting the outliers. In other words, correct and robust matches are expected to survive and might even populate

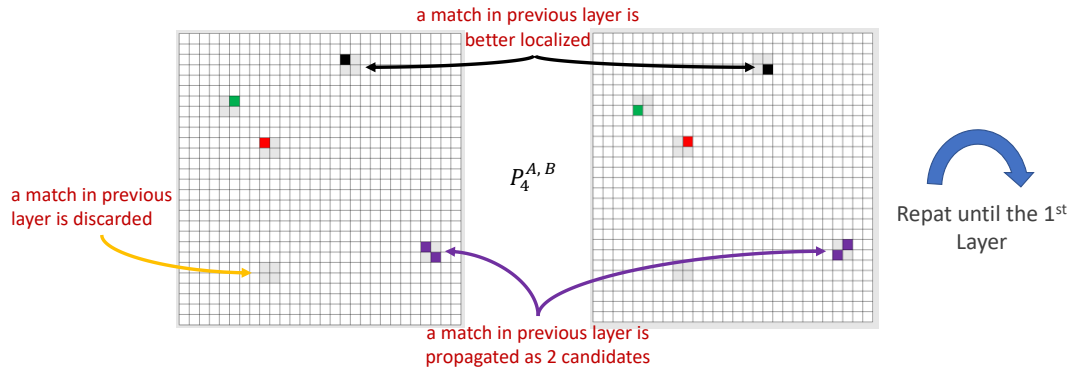


Figure 3.4: **Visual illustration of the first iteration of Hierarchical Refinement Algorithm (HRA).** Inherently, a match in previous layer is expected to be better localized due to the fact that shallow layers have better localization. Black, green and red matches are examples of better localized matches. Besides, thanks to the ratio test, a match may be discarded if its descendants are not discriminative enough. The yellow match is discarded, since any of its descendants find pair in relevant search space. Moreover, if two or more descendants in a matching area find their partners, then a match in the previous layer is propagated. The purple match is an example of this, since it propagated as 2 candidates. Note that, it is also possible the number of propagated candidates as 3 or 4. Finally, DFM continues hierarchical refinement until the 1st layer.

through hierarchical refinement, while the outliers are expected to be rejected. Inherently, a match in previous layer is expected to be better localized due to the fact that shallow layers have better localization. Black, green and red matches in Figure 3.4 are examples of better localized matches. Besides, thanks to the ratio test, a match may be discarded, if its descendants are not discriminative enough. In Figure 3.4, the yellow match is discarded, since any of its descendants find pair in relevant search space. Moreover, if two or more descendants in a matching area find their partners, then a match in the previous layer is propagated. The purple match of Figure 3.4 is an example of this, since it is propagated as 2 candidates. It should be noted that it is also possible to propagate the number of candidates as 3 or 4. In this manner, DFM continues hierarchical refinement until the 1st layer and completes the algorithm.

In Figure 3.5, the effect of the HRA on the matched features from the terminal layer to the shallowest layer is shown. In this figure, how the image matching performance is improved by using a subsequent block that hierarchically mimics the previous stage to refine positions of dense matches from the terminal layers. As it can be observed in the left column, there are some poorly localized or complete failure matches in *Layer*–5. As moving forward to the first layer, incorrect matched features are eliminated, and other matched features are better localized where also the number of matched features are propagated. Hence, finally we ended up with better localized and more accurate matches in *Layer*–1. On the right column, how the hierarchical refinement deals with outliers, that are coming from DNSS at the last layer is shown. Starting with such an outlier set, that consists of poorly localized or complete failure matches, HRA is still able to find good matches, either discarding the outliers or enhancing their localizations.

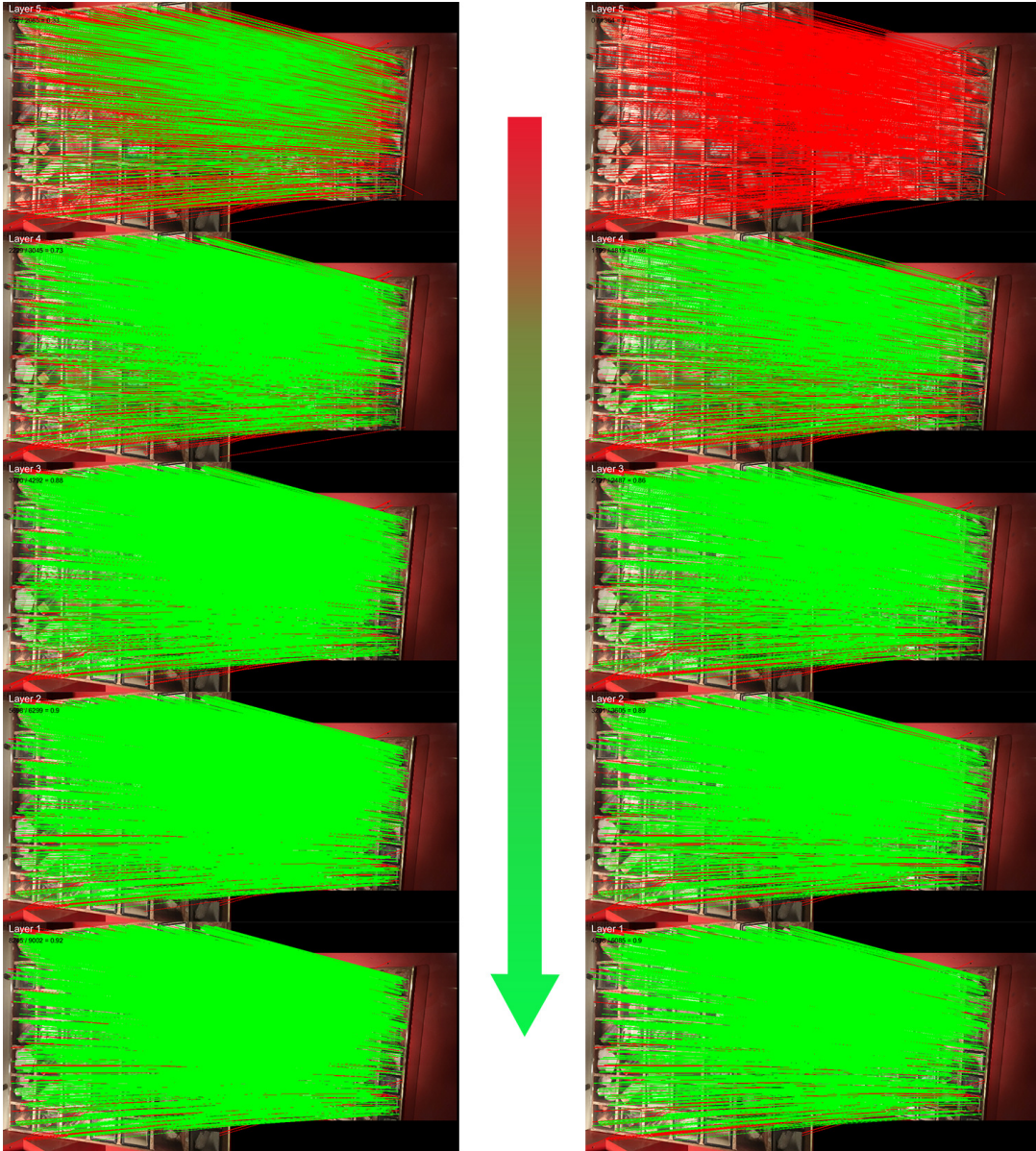


Figure 3.5: **Illustration of the effect of Hierarchical Refinement Algorithm (HRA) on matching pairs.** On the left, HRA started with 2085 matched features and 691 of them are correct in terms of matching accuracy under 5-pixel threshold, hence the accuracy is 0.33. Ultimately, HRA ended up with 9002 matched features with 8245 correct matches, therefore the accuracy at the first layer is 0.92. On the right, the behaviours of outliers are investigated. Even if it starts with 0 accuracy, HRA accomplished the task with 0.9 accuracy, and besides, it increased the number of matched features.

3.2.3 Two Stage Approach

Given an image pair, one can extract the features by using pre-trained VGG-19 feature extractor, perform DNNS at the terminal layer to initialize the matched pairs, and refine them up to the original resolution using HRA. The single-stage solution, i.e., Stage-1 only DFM, is able to find correspondences between the given image pair. However, we realized that this one-step solution fails during hierarchical refinement in case of severe geometric transformations. This result is due to the fact that features at shallower layers are not as robust as deep layers against geometric transformations. We also observed that even if the method is able to generate correct matches at deeper layers, these matches are eliminated while moving towards shallower layers. To overcome this issue, we first predict a homography matrix H_{BA} using the matched set of points at the terminal layer and warp B using H_{BA} to obtain warped image B_w . Next, we apply the solution mentioned above by using the reference image A and B_w . With such an approach, we are effectively first aligning the two images, then looking for possible matches, as proposed in [48]. Recently, this approach is successfully applied in [71]–[73].

When VGG-19 network is employed as the feature extractor, we first obtain a set of matched points by feeding P_5^A and P_5^B to DNNS. Using this set of matched pairs, we predict a homography matrix H_{BA} and obtain warped image B_{warped} . We extract features one more time for the warped image. Then, we initialize a set of matched points utilizing the terminal layers' features of the warped and the reference image with DNNS. For VGG-19, each point of the pair is the parent of 4 points in the former layer. To refine the pairs, HRA applies DNNS on 2x2 feature patches iteratively until the first layer. Note that each level number of feature points might change; in other words, P_1^A may have more points than P_5^A , or vice versa.

The effect of this two stage approach is illustrated in Figure 3.6. On the left column, DFM is enforced to use only Stage-1, where on the right column, Stage-0 and 1 are utilized together. It can be observed that for image pairs that have a significant amount of rotation in between, the two stage approach has a critical improvement on the performance of the DFM. Figure 3.6 shows the value the making a rough alignment with warping by using estimated transformation as a preliminary step.

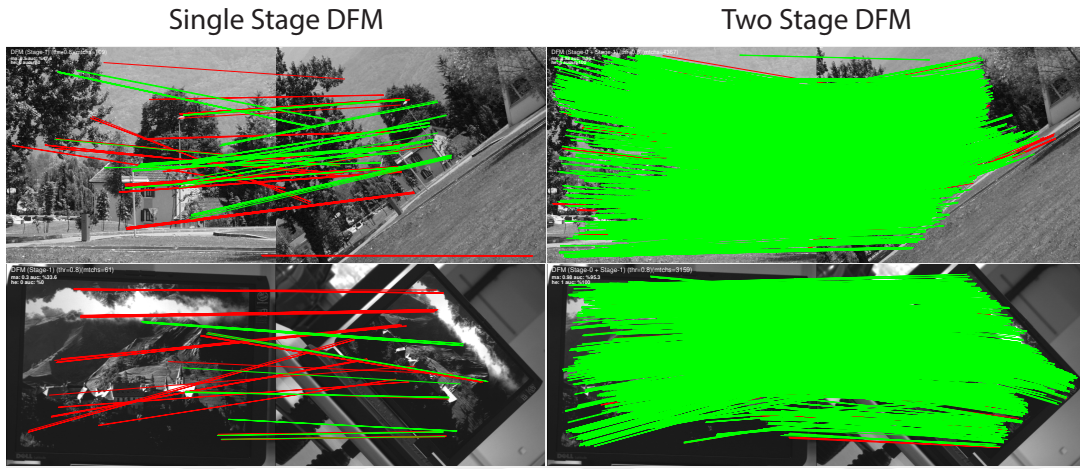


Figure 3.6: **Effect of two stage approach.** On the left, DFM is enforced to use only Stage-1, where on the right, Stage-0 and Stage-1 are utilized together. Quantitative results with the metrics defined in Chapter 4 are also demonstrated on the top-left corners of the drawings. A green line is for a correctly estimated match whose distance to the groundtruth match is ≤ 5 pixels, where a red line indicates an erroneous match.

CHAPTER 4

EXPERIMENTAL EVALUATION

4.1 Experimental Procedure

Testing the performance of an algorithm in a fair way for different applications is a challenging task. In addition, comparing and evaluating these algorithms on well-known datasets is also not straightforward due to hyper-parameter selection. This step is known to be crucial specifically for classical algorithms due to numerous parameters of such algorithms. In case of a comparison between a novel learning-based method against the classical methods, this parameter adjustment procedure is usually not examined in detail. Moreover, most of the time, the hyper-parameters of the classical methods are not even specified in the manuscripts. In this thesis, we try to compare classical and learning-based image matching algorithms in a fair manner on well-known HPatches dataset [52] by optimizing the hyper-parameters of each algorithm on the selected dataset.

In our study [74], we demonstrate that applying mutual nearest neighbor search that utilizes the ratio test on pre-trained VGG [59] features achieves the state-of-the-art performance. In order to examine the resulting effect of this ratio threshold on some classical methods, we have also examined five classical image matching algorithms [8]–[10], [12], [13] on HPatches dataset in terms of *Mean Matching Accuracy (MMA)* [37] and *Homography Estimation Accuracy (HEA)* [33]. We compare the conventional algorithms against four popular learning-based [33], [41], [51], [74] methods. We observe that ratio test threshold have a significant impact, as in [75], [76], on the performance of the methods SIFT, SURF, ORB, KAZE, AKAZE, SuperPoint and DFM. Similarly, the confidence parameter utilized in SuperGlue and Patch2Pix al-

gorithms also affects their performance significantly. This thesis presents the result of comprehensive experiments on nine algorithms with different ratio tests and confidence thresholds to reveal MMA and HEA performances of those algorithms between 1-10 pixel error thresholds. As a result of these results, we present the optimal parameters for each algorithm to maximize MMA and HEA for the required pixel threshold.

In the literature, most of the new algorithm proposals claim to outperform the preceding ones at least for some specific pixel thresholds. As we present through the experiments, most of the time by adjusting the hyper-parameters, it is possible one algorithm to outperform the rest for a specific pixel threshold, resulting in multiple state-of-the-art methods. To minimize this ambiguity, beyond the pixel threshold specific optimal parameters, Area Under Curve (AUC) values for MMA and HEA using the whole accuracy range (1-10 pixel thresholds) are also provided to reveal an average performance. We also present optimal parameters for MMA and HEA individually, since the optimal parameters for these two metrics might be different as demonstrated in [51], [74].

Throughout the experiments, we observe that changing only a single parameter of an algorithm yields this algorithm reaching the state-of-the-art performance. This circumstance is mostly observed for classical methods, which usually argued to have inferior performance compared to the recent learning-based algorithms. In other words, we demonstrate and argue that classical algorithms might still perform close to the state-of-the-art for image matching task, at least for HPatches dataset, and the performance gap between the classical and deep learning-based methods is not that significant, as it is accepted and reported before in the literature.

4.2 Experimental Setup

We construct an experimental setup in order to measure the performances of five classical (SIFT [8], SURF [9], ORB [10], KAZE [12], AKAZE [13]) and four learning-based (SuperPoint [33], SuperGlue [41], Patch2Pix [51] and our proposed DFM method [74]) image matching algorithms on HPatches dataset in terms of widely used metrics MMA and HEA by varying only one parameter, either ratio test or confidence.

4.2.1 Dataset

HPatches dataset [52] consists of 116 sequences of two subsets, namely illumination and viewpoint sets. The illumination subset includes 57 sequences, and each has 6 images and 5 ground-truth homographies between the first image and others. The viewpoint subset has 59 sequences with the same structure. The sequences in the illumination subset have significant illumination variation with the same viewpoints antithetical to the viewpoint subset in which sequences have significant viewpoint changes with similar illuminations. Following D2-Net [37], we leave out large images and made evaluations on 52 illumination sequences and 56 viewpoint sequences, in order to make all algorithms execute and to be coherent with the literature.

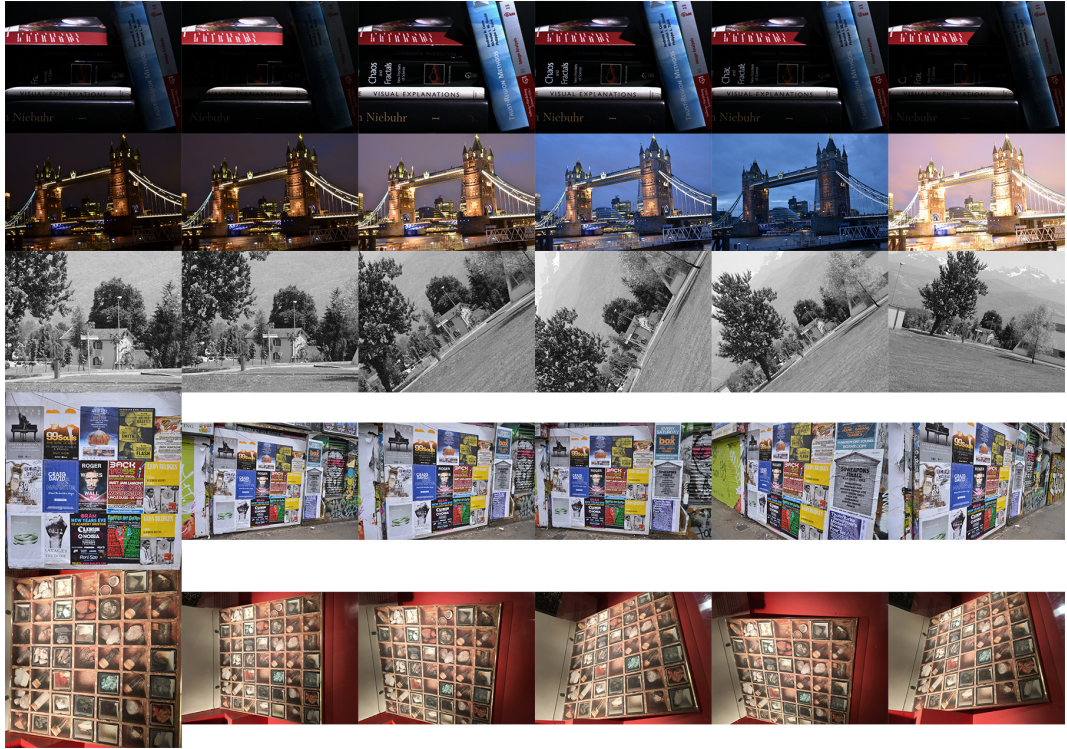


Figure 4.1: **Example sequences of HPatches [52] dataset.** First two rows are from illumination subset, where others are from viewpoint subset.

4.2.2 Performance Metrics

In this thesis, two different performance metrics are preferred, which are Mean Matching Accuracy (MMA) and Homography Estimation Accuracy (HEA).

Mean Matching Accuracy (MMA) is a widely used performance metric; recently, many state-of-the-art works [37], [51], [72], [74] report their performances in terms of MMA on HPatches dataset. This metric basically measures the average accuracy of the matched features over the dataset. Given an image pair and matched features between them, matching accuracy is defined as the percentage of the correctly matched features. A match is accepted as correct, if the distance between the reprojected feature point with ground-truth homography and its corresponding match point is less than a given pixel threshold. In our experiments, we vary the threshold from 1 pixel to 10 pixels as applied in the literature.

Homography Estimation Accuracy (HEA) is another widely used metric for image matching evaluation and used in [33], [51], [74], [77] as a performance metric. MMA solely may not be sufficient for image matching evaluation, since an algorithm with a very limited number of and poorly distributed matches may make a high score in terms of MMA, but it is likely to fail at geometric transformation estimation. Hence, we take into account HEA and use it as the second performance metric during all of our experiments. In order to compute HEA, four corners of one image is reprojected onto the other image with the estimated and the ground-truth homographies. Then we take the average distance between these projected four points and accept the estimated homography correct, if the reprojection error is smaller than the given threshold. HEA is the rate of correctly estimated homographies over whole dataset.

4.2.3 Algorithms

We use OpenCV [78] 4.5.2 implementations of the classical algorithms SIFT, ORB, KAZE, and AKAZE with the default parameters for feature detection and description. For SURF, we use OpenCV 3.4.2 with the default parameters. For SuperPoint, we utilize SuperGlue GitHub repository [79] in order to obtain keypoint locations and their descriptors. All the algorithms mentioned above perform Mutual Nearest Neighbor

Search with Bidirectional Ratio Test (MNNSwBRT) to find matches between extracted features. We measure the performance for different ratio test thresholds from 0.1 to 1.0 with steps of 0.1.

For DFM, we also use the original implementation [80] of the algorithm which again takes the advantage of MNNSwBRT. However, since DFM benefits from MNNSwBRT multiple times, to be fair against all algorithms, we keep the ratio test thresholds for the deepest two layer’s descriptors fixed in the act of 0.95 and 0.90 as in the original paper, and only optimize the ratio test thresholds of the first three shallowest layer as their multiplication results with the threshold value. To illustrate, we have used the threshold set [0.80, 0.80, 0.80, 0.90, 0.95] for the threshold value 0.5 since $0.5^{(1/3)} \approx 0.80$.

For SuperGlue, we have used the GitHub repository [79] with the default parameters except for not resizing the input images and using the ‘outdoor’ setting, which gives better performance HPatches dataset. For Patch2Pix, we use the official GitHub repository [81] with default parameter settings. For both SuperGlue and Patch2Pix we measure the performance for different confidence thresholds from 0.9 to 0.0 with steps of 0.1.

Computing MMA is straightforward and reported with the same procedure in many works. Nonetheless, Homography Estimation performance depends on utilized homography estimation method, many image matching algorithms reported their results using different homography estimation methods. For example, Patch2Pix has used pydegensac, where DFM has used MATLAB’s estimateGeometricTransform function, both are different versions of RANSAC [17] algorithm. In this work, we use OpenCV’s findHomography function with newly introduced cv.USAC_MAGSAC method [82], which is available from OpenCV 4.5.2 version, and whose success is demonstrated in [83]. We adjust other RANSAC parameters as follows: ransacReprojThreshold=3.0, maxIters=5000, confidence=0.9999.

4.3 Experimental Results

We follow the procedure explained in Section 4.2 and evaluate nine algorithms on HPatches dataset, with varying thresholds, which are different for each algorithm, as defined in Table 4.1.

4.3.1 Effect of Matching Threshold

Figure 4.2 and Figure 4.3 give the performances of 5 classical and 4 learning-based image matching algorithms for varying algorithm-based thresholds in terms of MMA and HEA with different pixel thresholds on HPatches dataset. The algorithm-based threshold is defined as the ratio test threshold for MNNSwBRT matcher for SIFT, SURF, ORB, KAZE, AKAZE, and SuperPoint algorithms, $(\text{ratio test threshold})^3$ for DFM and $(1 - \text{confidence})$ for SuperGlue and Patch2Pix algorithms. It should be noted that a smaller threshold yields strict matching and returns fewer matches, while larger threshold values mean loose matching and return more matches.

From Figures 4.2 and 4.3, it can be clearly observed that the algorithm-based threshold has a notable effect on the performance. For example, for a specific threshold (0.4), SIFT becomes the best-performing method, while it takes 4th place, when the threshold value is 1.0 in terms of MMA under 1 pixel threshold. A similar pattern is observed for most of the algorithms evaluated. Another interesting observation is that most of the classical algorithms have a bell-shaped curve, similar to the observations in [75]. Moreover, the effect of the algorithm-based threshold is more significant than learning-based algorithms in terms of MMA, which claims that when evaluating a classical method, at least the ratio test threshold parameter should be optimized for the given dataset. An additional important observation is the almost monotonic increase in HEA for every algorithm, indicating that RANSAC algorithm is powerful enough to handle putative match sets consists of some amount of outliers and further giving better results.

Table 4.1: **Definition of the Threshold Values** used in experiments. For all classical algorithms and SuperPoint + NN, we define *threshold* as the *ratio test threshold* of MNNSwBRT matcher, where for SuperGlue and Patch2Pix, it is defined as *the complement of the confidence*. Finally, for DFM algorithm, the threshold is described as *the product of the equal ratio test thresholds used for the first three layers*.

Method	Threshold
SIFT [8] + NN	Ratio Test Threshold
SURF [9] + NN	Ratio Test Threshold
ORB [10] + NN	Ratio Test Threshold
KAZE [12] + NN	Ratio Test Threshold
AKAZE [13] + NN	Ratio Test Threshold
SuperPoint [33] + NN	Ratio Test Threshold
SuperPoint + SuperGlue [41]	(1 - Confidence)
Patch2Pix [51]	(1 - Confidence)
DFM [74]	(Ratio Test Threshold) ³ for first 3 layers

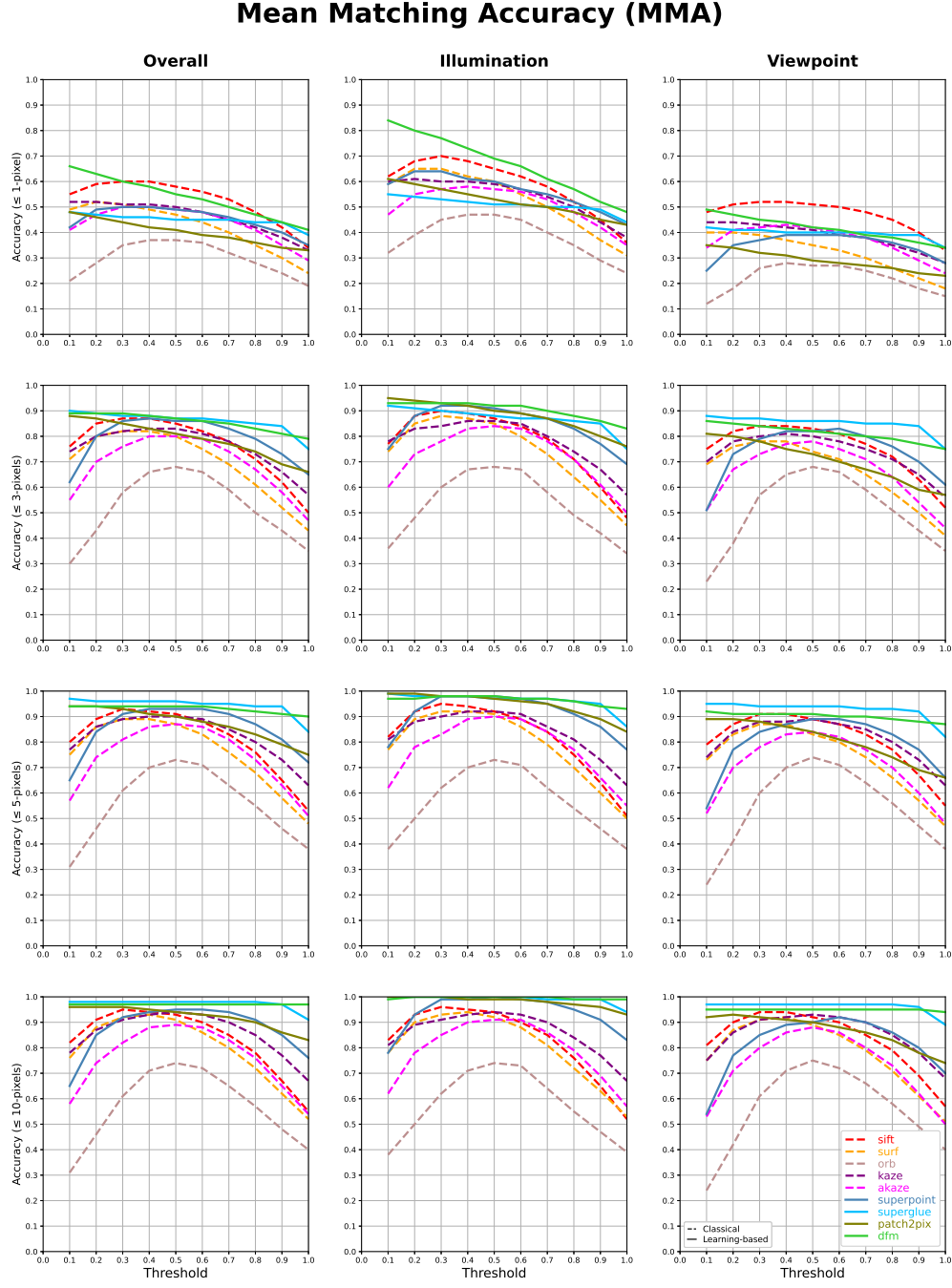


Figure 4.2: **Experimental Results** of 5 classical and 4 learning-based algorithms in terms of Mean Matching Accuracy (MMA) on HPatches [52] dataset. Results are shown as dashed lines for classical algorithms and straight lines for learning-based algorithms. Individual plots show the accuracy with varying threshold values while rows indicate different pixel thresholds.

Homography Estimation Accuracy (HEA)

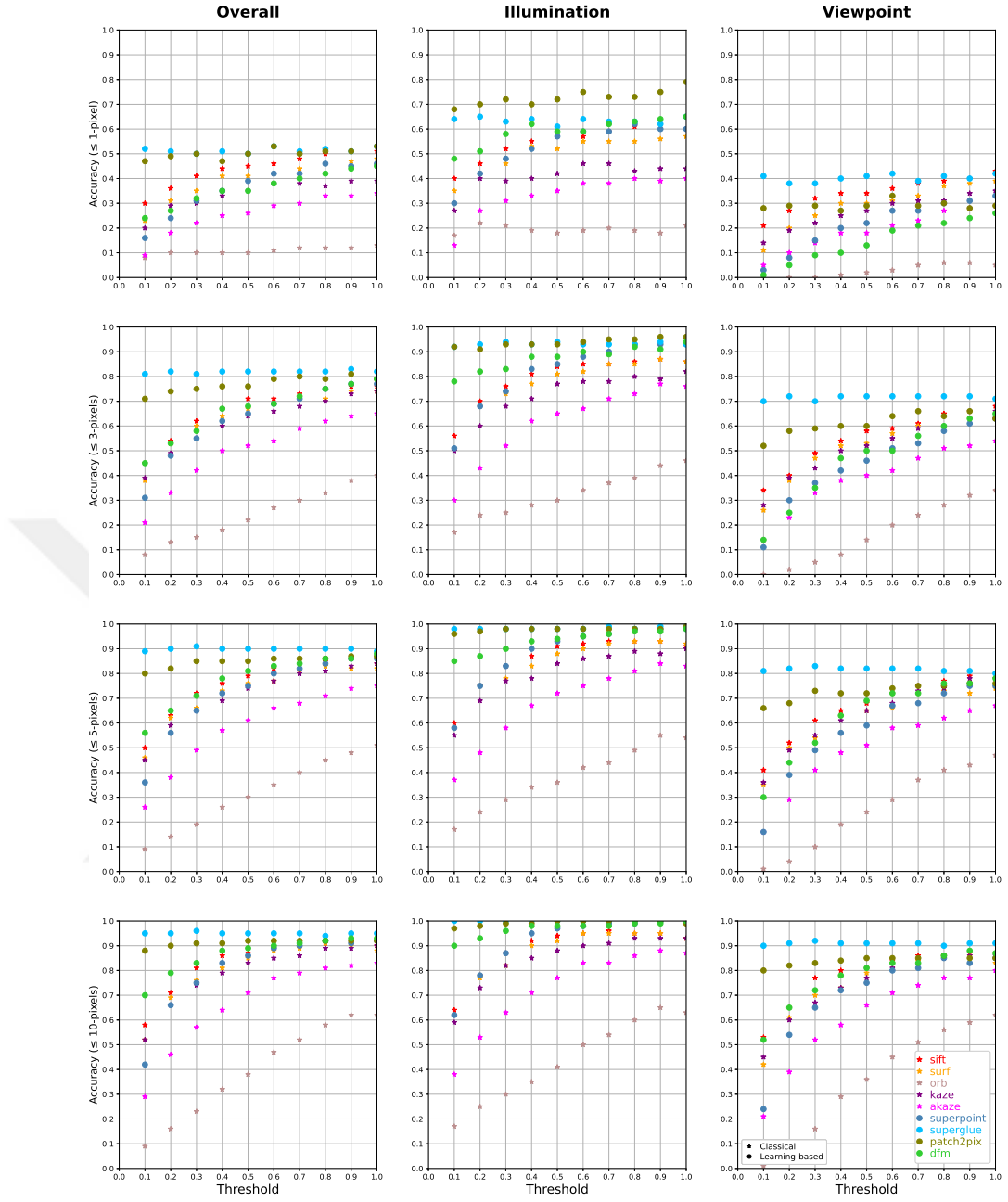


Figure 4.3: **Experimental Results** of 5 classical and 4 learning-based algorithms in terms of Homography Estimation Accuracy (HEA) on HPatches [52] dataset. Results are shown as stars for classical algorithms and dots for learning-based algorithms. Individual plots show the accuracy with varying threshold values while rows indicate different pixel thresholds.

4.3.2 Comparisons based on Best Accuracy Values

Table 4.2 is tabulated by picking the best accuracy in terms of both MMA and HEA and the selected thresholds for each algorithm, considering the 1, 3, 5, and 10-pixel threshold. From this table, we again notice the monotonic increasing behavior of HEA with respect to threshold, and in contrast, MMA maximizes in lower threshold values. This observation indicates that tight thresholds result in correct matches while loose thresholds result in more matches and increase the accuracy of the homography estimation by the help of RANSAC. The last but not the least, the implication from the table is that there is only a small performance gap between recently developed state-of-the-art learning-based algorithms and the classical algorithms by only adjusting a single parameter. As an example, SIFT is the second-best performing algorithm under 1-pixel accuracy, and it has only a few percent away from the best-performing methods for the other pixel thresholds. Noting that we did not attempt to optimize any other parameters of the classical algorithms, such an optimization over the dataset might yield these algorithms outperform the state-of-the-art.

Table 4.2: **Best accuracy values** and their relevant thresholds for each evaluated algorithm in terms of both MMA and HEA metrics, considering the 1, 3, 5, and 10-pixel thresholds. Best-performing results are shown as bold and the second-best-performing ones are shown as bold-italic.

Method	Mean Matching Accuracy (MMA)								Homography Estimation Accuracy (HEA)							
	$\leq 1\text{px}$		$\leq 3\text{px}$		$\leq 5\text{px}$		$\leq 10\text{px}$		$\leq 1\text{px}$		$\leq 3\text{px}$		$\leq 5\text{px}$		$\leq 10\text{px}$	
	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.
SIFT [8] + NN	0.60	0.3	0.87	0.3	0.93	0.3	0.95	0.3	0.51	0.9	0.76	0.9	0.86	0.9	0.91	0.7
SURF [9] + NN	0.52	0.2	0.82	0.3	0.89	0.3	0.93	0.4	0.48	1.0	0.74	0.9	0.83	0.8	0.91	0.8
ORB [10] + NN	0.37	0.4	0.68	0.5	0.73	0.5	0.74	0.5	0.13	1.0	0.4	1.0	0.51	1.0	0.62	0.9
KAZE [12] + NN	0.52	0.1	0.83	0.4	0.90	0.4	0.94	0.5	0.39	0.9	0.74	1.0	0.84	1.0	0.90	1.0
AKAZE [13] + NN	0.50	0.3	0.80	0.4	0.87	0.5	0.89	0.5	0.34	1.0	0.65	1.0	0.75	1.0	0.83	1.0
SuperPoint [33] + NN	0.50	0.3	0.87	0.4	0.93	0.4	0.95	0.5	0.46	0.8	0.70	0.9	0.86	0.9	0.92	0.8
SuperPoint + SuperGlue [41]	0.48	0.1	0.90	0.1	0.97	0.1	0.98	0.1	0.53	0.6	0.83	0.9	0.91	0.3	0.96	0.3
Patch2Pix [51]	0.48	0.1	0.88	0.1	0.94	0.1	0.96	0.1	0.53	0.6	0.81	0.9	0.87	0.9	0.92	0.5
DFM [74]	0.66	0.1	0.89	0.1	0.94	0.1	0.97	0.1	0.45	1.0	0.79	1.0	0.88	1.0	0.93	0.9

4.3.3 Comparisons based on Best Area Under Accuracy Curves

Table 4.3 is obtained by selecting an algorithm-based threshold for each algorithm to maximize the average accuracy for different pixel thresholds from 1 to 10 pixels. This average also gives area under the accuracy curve (AUC) shown in Figure 4.4. Table 4.3 also illustrates the number of matches for relevant thresholds, indicating the fact that high MMA can be achieved with higher confidence and less number of matches, while high HEA can be obtained with less confidence and hence a vast number of matches. The most cardinal demonstration of Table 4.3 is a classical algorithm SIFT is the second best performing algorithm in terms of AUC of mean matching accuracy. Also, its performance in terms of AUC of homography estimation accuracy is only two percentage below the second best performing algorithm, Patch2Pix. In addition, SURF and KAZE algorithms also have comparable performances with the state-of-the-art algorithms. Finally, note that although we categorize proposed DFM as a learning-based method due to the fact that it utilizes the deep features extracted by a pre-trained VGG-19 network, it can be considered as almost a classical approach,

Table 4.3: **Best area under curve (AUC) percentage values** and their relevant thresholds for each algorithm considering both MMA and HEA metrics from 1 to 10 pixel threshold. Best-performing results are shown as bold, and the second-best-performing ones are shown as bold-italic. Also, the number of detected features, the number of matched features using related threshold values and the average run times are indicated.

Method	MMA					HEA			
	#Features	AUC	Thr.	#Matches	Avg. Time (s)	AUC	Thr.	#Matches	Avg. Time (s)
SIFT [8] + NN	4572	89.6	0.3	478	0.741	82.1	0.9	1293	0.708
SURF [9] + NN	6003	85.6	0.3	276	0.706	80.0	0.9	1378	0.703
ORB [10] + NN	499	69.1	0.5	19	0.078	48.0	1.0	170	0.079
KAZE [12] + NN	3120	86.3	0.5	544	1.281	79.8	1.0	1287	1.257
AKAZE [13] + NN	2694	82.9	0.5	219	0.287	72.0	1.0	1011	0.289
SuperPoint [33] + NN	921	88.3	0.5	253	0.091	82.6	1.0	506	0.091
SuperPoint + SuperGlue [41]	921	91.6	0.1	441	0.075	86.8	0.6	482	0.075
Patch2Pix [51]	-	89.2	0.1	723	0.596	84.1	0.9	1434	0.592
DFM [74]	-	91.6	0.1	881	0.321	84.0	1.0	16619	0.556

since it only employs the well established classical computer vision algorithms, such as initial warping, hierarchical refinement and nearest neighbor search with ratio test in a very simple framework; and note that DFM has no specific training procedure for image matching task.

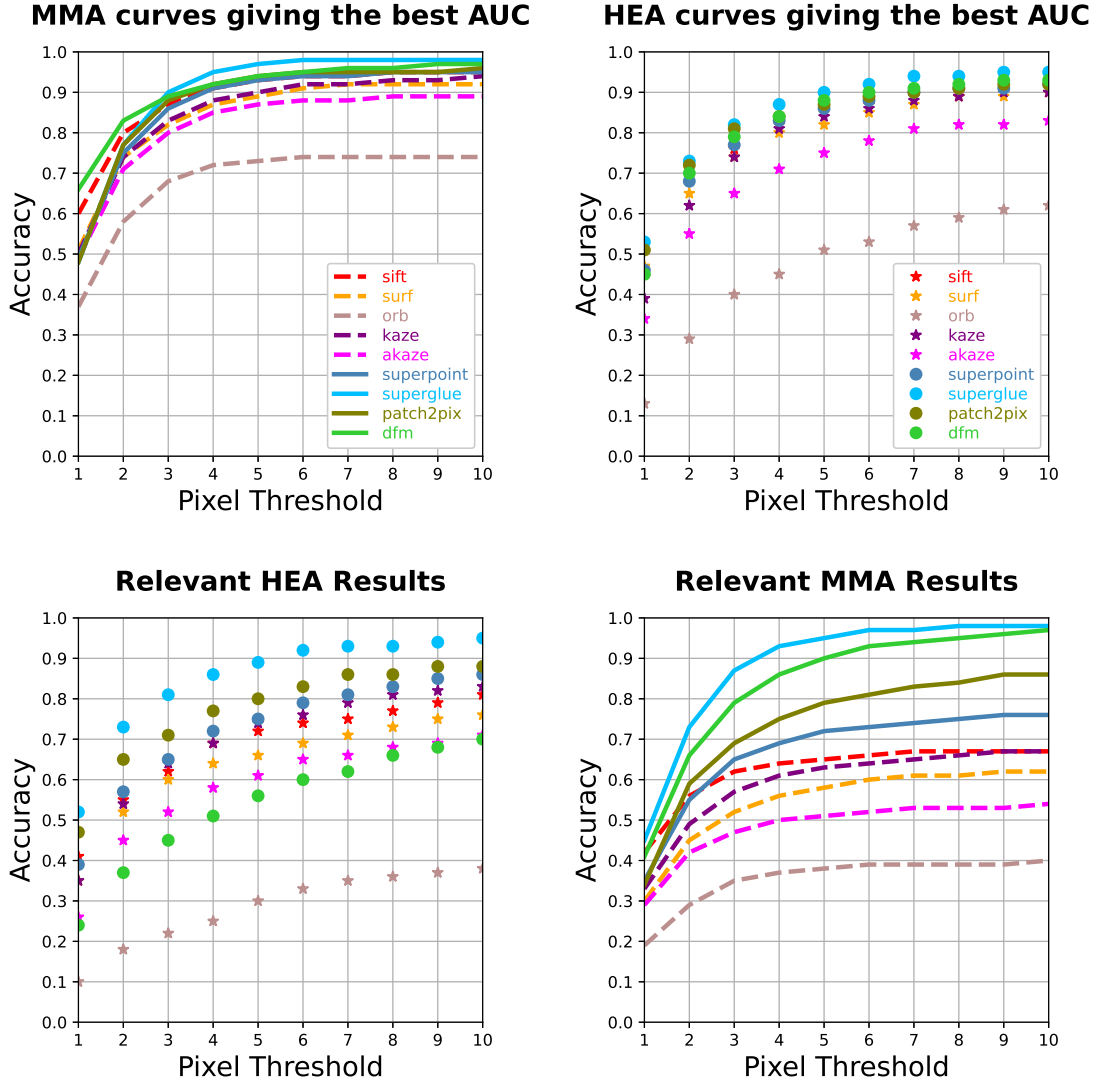


Figure 4.4: **Overall MMA and HEA curves** that give the best AUC for each algorithm. These curves indicate the best possible AUC result for each algorithm can achieve on HPatches [52] dataset by only optimizing a single parameter ratio test or match confidence. For each setting that gives the best AUC for one metric, the relevant result of the other metric is also presented.

Here, it should be noted that given optimal algorithm-based thresholds are valid for only the HPatches dataset and could not be generalized for any other dataset. These

experiments with varying algorithm-based thresholds should be repeated for other datasets to find the optimal algorithm-based thresholds.

Figure 4.4 exhibits the MMA and HEA curves that give the best AUC for each algorithm. The figure utilizes the thresholds given in Table 4.3 and illustrates the MMA and HEA curves. Note that it also shows the curve of the other metric by using the thresholds optimized for one metric, i.e., it displays the HEA curve with the thresholds that maximize the AUC of the MMA curve and vice-versa. It can be seen from the figure, for the most of the algorithms, the best threshold for one metric is not good enough for the other metric. For example, when DFM’s ratio test threshold is optimized for MMA, its performance is poor for HEA. Except that, Patch2Pix and specifically SuperGlue algorithms are robust to such a parameter variation, performing well in both metrics. It may result from the fact that these two algorithms inherently learn the scene geometry so that optimizing these algorithms to obtain high MMA will naturally result in a high performance for HEA as well. Most importantly, the figure claims that classical methods can still perform very close to the well-trained state-of-the-art algorithms by adjusting just a single parameter. In fact, SIFT still achieves state-of-the-art performance, being under only DFM in terms of MMA for 1 and 2-pixel thresholds, and there is not a significant performance gap between the state-of-the-art algorithms in terms of HEA. This kind of result, meaningly a classical algorithm performs this much closer to the recent methods, is not reported in previous studies [37], [51], [72], [74].

4.3.4 Comparisons based on Execution Time and AUC

Table 4.3 demonstrates the average run times of each algorithm. All these algorithms are tested on Supermicro SuperWorkstation 5049A-T, which has Intel Xeon Silver 4215R 3.20 GHz CPU and NVIDIA GeForce RTX 2080 TI GPU. These results reveal that SuperPoint + SuperGlue is the leading algorithm in terms of the run time. Also, note that all the algorithms have approximately the same run times for MMA and HEA, where our method DFM is the only affected one. That is because DFM’s run time depends on the number of matches found on the previous layer.

To observe the effect of the Hierarchical Refinement Algorithm [2] on the run time

and the accuracy, we set up an experimental setup in which we measure the AUC, the number of matches, and average execution time for the different variants of our proposed DFM method. We perform experiments for matches that come from different layers as demonstrated in Table 4.4. In this table, it can be observed that using only the 5th layers matches results with a poor AUC but faster. Going through to the first layer, AUC and the number of matches increase where the speed of the algorithm decreases. So, there is a trade-off between accuracy and algorithm efficiency. According to 4.4 our proposed DFM can be used for the applications that require time efficiency by renouncing some amount of accuracy. In addition, it can be seen from the Table 4.4, going through to the first layer the number of matches propagates with increasing accuracy which shows the strength of the hierarchical refinement algorithm.

Table 4.4: **Performances vs. Run Times** for different variants of DFM. Layer-n stands for the matches are from the nth Layer. Layer-1 is identical to the original DFM algorithm used in Table 4.2 and Table 4.3

Variants of DFM [74]	AUC of HEA	#Matches	Avg. Time (s)
Layer-5	56.8	1589	0.184
Layer-4	62.1	2659	0.217
Layer-3	73.4	5078	0.279
Layer-2	82.6	9444	0.377
Layer-1 (original)	84.0	16619	0.556

4.4 Qualitative Results

In this section, visual results of image pairs *books1* & *books3* in illumination subset together with *posters1* & *posters4* and *graffiti1* & *graffiti4* pairs in viewpoint subset from HPatches dataset [52] are shown by using five classical and four learning-based algorithms.

In all the figures, the algorithms optimized for giving the best MMA as well as HEA are shown together to illustrate the qualitative effect of the parameter selection. On

the top left corners of the drawings, the information about relevant algorithm and the selected threshold values are provided together with the total number of matched features. The total number of matched features is tend to be more for algorithms optimized for HEA, since RANSAC can eliminate outliers and estimate better homography using more of matched features. In each figure, under the name of the algorithm, *matching accuracy (ma)* at the specific pixel threshold and the *area under curve (auc)* value of *ma* curve from 1-10 pixels are given. At the bottom line, *homography estimation (he)* at the specific pixel threshold (it can be either 0 or 1) and the *area under curve (auc)* value of *he* curve from 1-10 pixels are given. A green line is for a correctly estimated match whose distance to the groundtruth match is ≤ 1 pixel for *books* and *posters* pairs, ≤ 5 pixels for the *graffiti* pair where a red line denotes an outlier match.

Optimized for best MMA (at 1 px)

Optimized for best HEA (at 1 px)

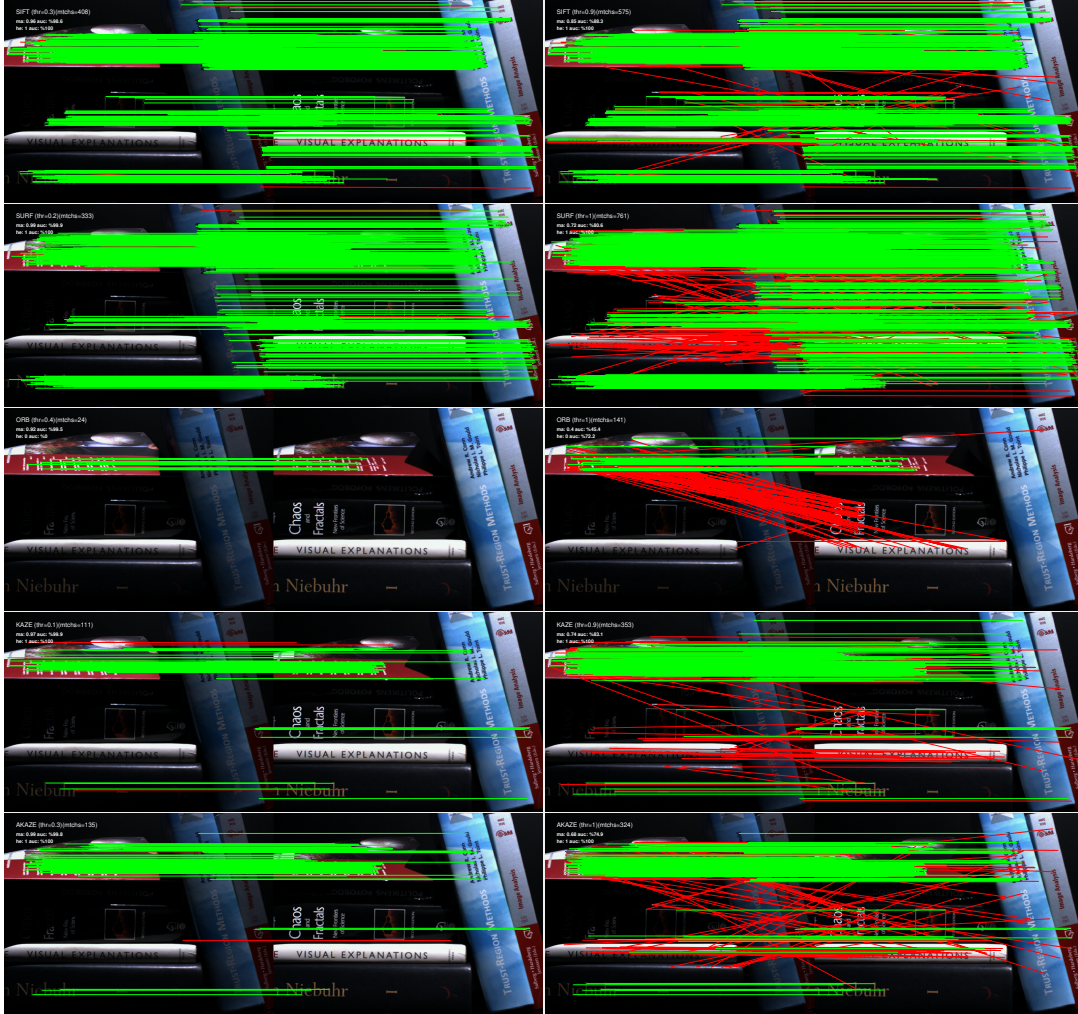


Figure 4.5: **Visual Illustration of results of Classical Algorithms SIFT, SURF, ORB, KAZE, and AKAZE** optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair books 1 & 3

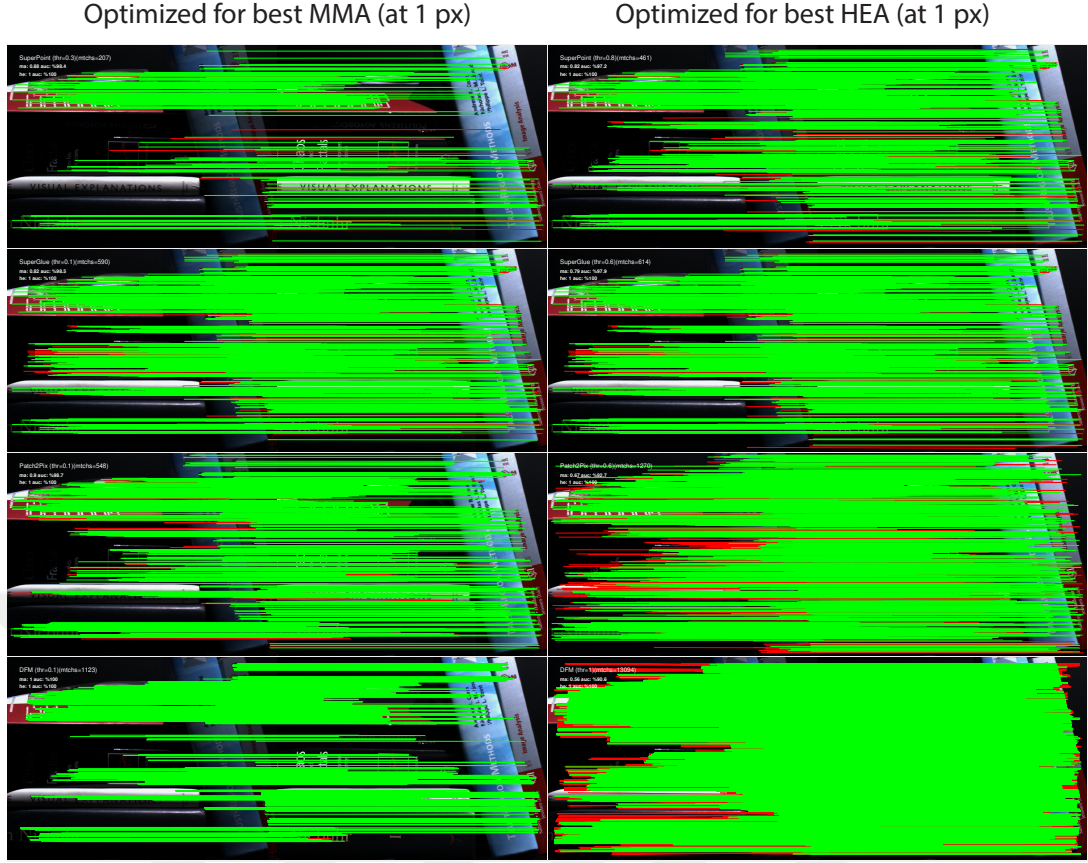


Figure 4.6: **Visual Illustration of results of Learning-based Algorithms SuperPoint, SuperGlue, Patch2Pix, and DFM** optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair *books 1 & 3*

Figures 4.5 and 4.6 display the performances of nine algorithms in a presence of illumination change between images. These figures reveal that matches should be properly distributed on the images to get a good homography estimation score. Such a good scattering behaviour for features can be observed in all learning-based algorithms, and classical algorithms SIFT and SURF. Note that, in terms of the number of matched features and matching accuracy under 1-pixel threshold, our method, the proposed DFM is the leading algorithm.



Figure 4.7: **Visual Illustration of results of Classical Algorithms SIFT, SURF, ORB, KAZE, and AKAZE** optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair *posters 1 & 4*



Figure 4.8: **Visual Illustration of results of Learning-based Algorithms Super-Point, SuperGlue, Patch2Pix, and DFM** optimized for giving the best MMA (the left column) and the best HEA (the right column) under 1-pixel threshold on the image pair *posters 1 & 4*

Figures 4.7 and 4.8 show the performances of nine algorithm in a presence of some amount of viewpoint change between images. In terms of matching accuracy under 1-pixel threshold, a well-known classical algorithm SIFT is the prominent algorithm that supports our claim that classical algorithms work well even outperform the learning-based ones when a convenient parameter adjustment is applied. Moreover,

note that classical algorithms are able to find more matched features than the learning-based methods, except our proposed method DFM, which is still the superior method finding of 6 times more matches than the second-best algorithm SURF in terms of match count.

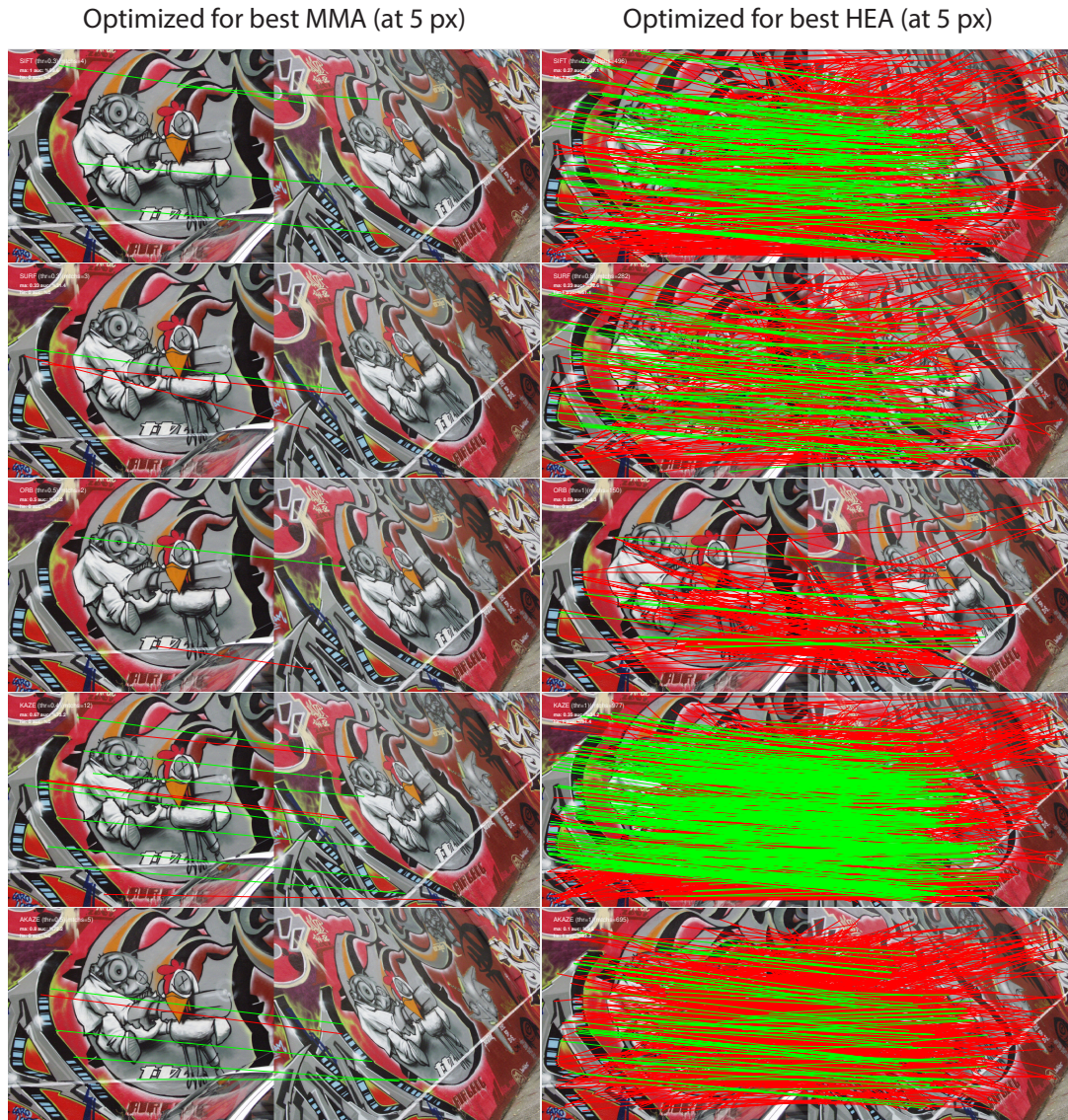


Figure 4.9: **Visual Illustration of results of Classical Algorithms SIFT, SURF, ORB, KAZE, and AKAZE** optimized for giving the best MMA (the left column) and the best HEA (the right column) under 5-pixels threshold on the image pair *graffiti 1 & 4*

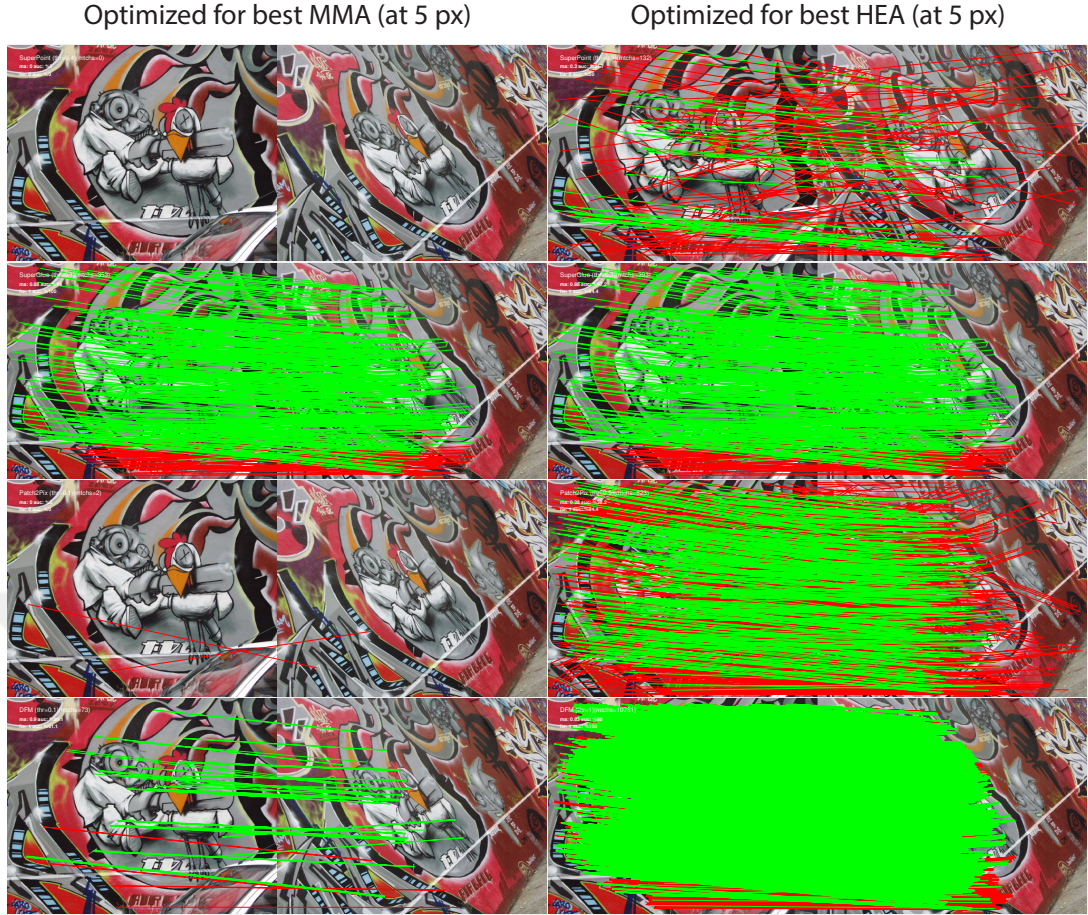


Figure 4.10: **Visual Illustration of results of Learning-based Algorithms SuperPoint, SuperGlue, Patch2Pix, and DFM** optimized for giving the best MMA (the left column) and the best HEA (the right column) under 5-pixels threshold on the image pair *graffiti 1 & 4*

Figures 4.9 and 4.10 present the performances of nine algorithms in the presence of a large amount of viewpoint change between images with smooth textures. We prefer to use 5-pixels threshold instead of 1-pixel threshold employed previously, since this pair is a bit harder than other pairs. It can be observed that from SIFT's results, using only the MMA metric is not sufficient enough for evaluating the algorithms. Another observation is that SuperGlue is quite robust to parameter optimization for giving the best MMA and HEA. It works well when estimating homography, even if the algorithm is optimized for providing the best MMA. These experiments are also demonstrating the powerfulness of the learning-based integrated approaches on the homography estimation task. It should also be noted that the KAZE algorithm

performs quite well on this image pair, which is probably due to the exploiting of nonlinear diffusion scale space, unlike SIFT, which uses Gaussian scale space.

In terms of matching accuracy under 1-pixel threshold, a well-known classical algorithm SIFT is best performing algorithm that supports our claim that classical algorithms can work well even outperform the learning-based ones with a convenient parameter adjustment. Moreover, note that classical algorithms are able to find more matched features than the learning-based methods except our method, DFM, which is still the superior method finding of 6 times more matches than the second-best algorithm SURF in terms of match count.

4.5 Discussion

With these experiments, we argue that our proposed novel technique, DFM, which only uses off-the-shelf deep feature extractors with no training and employs standard techniques, is able to form a performance baseline for learning-based image matching methods. These experimental results also reveal that a simple dense nearest neighbor search of the deep features at the terminal layer of a frozen VGG-19 network might still yield sufficient matches for obtaining a geometric transformation- and illumination-invariant matching.

Moreover, we examine the performance of the classical methods in detail by employing mutual nearest neighbor search with bidirectional ratio test and optimizing the ratio test threshold to achieve the best performance on two different performance metrics. We compare classical and learning-based methods and after a fair comparison, the experimental results on HPatches dataset reveal that the performance gap between classical and learning-based methods is not that significant, which is by contrast with learning-based methods' reports that claim to outperform the classical techniques with a significant margin.

We discuss that, by only adjusting a single hyper-parameter, namely ratio test threshold, classical algorithms still competes with the state-of-the-art learning-based methods in terms of MMA and HEA metrics on HPatches dataset.

Further, by revisiting the existing methods, we propose the optimal parameter settings for classical and learning-based algorithms and present the optimal MMA and HEA performance on HPatches dataset for each method.

Finally, we provide an experimental setup to determine the optimal parameters of 9 well-known image matching algorithms using MMA and HEA metrics not only for HPatches but also for any dataset in which these performance metrics can be employed.





CHAPTER 5

CONCLUSIONS

In this thesis, we propose a method for finding point correspondences between two images by using the features that are extracted by a pre-trained VGG-19 network. We also evaluate and compare five classical and four learning-based algorithms for image matching task by using well-known HPatches dataset [52]. The conclusions from these experiments are provided in the next section.

5.1 Conclusions

We propose a novel image matching method by motivating from the recent tendency for integrating the feature detection, description, and matching steps to not stick to the classical image matching pipeline. Considering the performance of all experiments, one can conclude that benefiting from basic vision techniques and using the features extracted by an off-the-shelf VGG-19 network, state-of-the-art performance can be achieved without any special training for feature matching. To be more specific, we also demonstrate that the features extracted by VGG network at its deepest layers are quite discriminative that a simple nearest neighbor search can be applied densely to initiate matches, decreasing the necessity of a learning-based dense feature matcher, such as NCNet [49]. Based on the presented results, one can also argue that the initial matches can be refined hierarchically through shallower layers yielding quite good localization for features.

Based on our experiments, we demonstrate the significant effect of the ratio test threshold for feature matching. Furthermore, the experimental results indicate that classical methods are quite powerful, so that they still perform close to the state-

of-the-art techniques. Specifically, one of the most popular methods for nearly two decades, SIFT [8], achieves almost the state-of-the-art performance by only optimizing the ratio test threshold, which is mostly neglected in the previous studies. Proposed DFM method [74] also demonstrates the feature matching capability of conventional techniques by achieving state-of-the-art performance with only practicing well-established computer vision techniques on top of a pre-trained deep learning backbone.

5.2 Limitations

Apart from all the aforementioned advantages for the proposed DFM method, as a first limitation, in DFM’s initial warping stage, we assume that the matched scenes are planar or the camera pose between frames is only due to rotation. Hence, our two-stage solution tends to perform relatively well in planar scenes, as demonstrated on HPatches dataset [52] and may be useful for aerial image matching. DFM method might perform poorly in the presence of non-planar image pairs captured by a translating camera. For such scenarios, the proposed method should be either modified to solve appropriate geometric transformation, instead of homography, or should be utilized with one-stage version. For the one-stage case, weakness of features at shallow layers limit the performance by preventing the method from finding matches at those layers, which we observed in the HPatches dataset, in case of severe geometric transformations.

Although our conclusions from the experiments may be limited to HPatches dataset, we argue that classical methods should be carefully analyzed before immediately working on a learning-based method. Despite the promising performance of the traditional methods, we should remind that a learning-based technique, SuperPoint [33] together with SuperGlue [41], yields the best-performing method on HPatches dataset in terms of area under curves for mean matching accuracy and homography estimation accuracy along with its robustness to the varying confidence thresholds. Also, it should be emphasized that optimal algorithm-based thresholds found from experiments are valid for only the HPatches dataset and could not be generalized for any other dataset.

REFERENCES

- [1]C. G. Harris, M. Stephens, *et al.*, “A combined corner and edge detector.”, in *Alvey vision conference*, Citeseer, vol. 15, 1988, pp. 10–5244.
- [2]S. M. Smith and J. M. Brady, “Susan—a new approach to low level image processing”, *International journal of computer vision*, vol. 23, no. 1, pp. 45–78, 1997.
- [3]E. Rosten and T. Drummond, “Fusing points and lines for high performance tracking”, in *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, Ieee, vol. 2, 2005, pp. 1508–1515.
- [4]J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions”, *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.
- [5]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)*, Ieee, vol. 1, 2005, pp. 886–893.
- [6]K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [7]M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features”, in *European conference on computer vision*, Springer, 2010, pp. 778–792.
- [8]D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [9]H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features”, in *European conference on computer vision*, Springer, 2006, pp. 404–417.
- [10]E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf”, in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

- [11]S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints”, in *2011 International conference on computer vision*, Ieee, 2011, pp. 2548–2555.
- [12]P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “Kaze features”, in *European conference on computer vision*, Springer, 2012, pp. 214–227.
- [13]P. F. Alcantarilla and T. Solutions, “Fast explicit diffusion for accelerated features in nonlinear scale spaces”, 2013. DOI: 10.5244/C.27.13.
- [14]J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, “Locality preserving matching”, *International Journal of Computer Vision*, vol. 127, no. 5, pp. 512–531, 2019.
- [15]I. E. Sutherland, “Three-dimensional data input by tablet”, *Proceedings of the IEEE*, vol. 62, no. 4, pp. 453–461, 1974.
- [16]R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. USA: Cambridge University Press, 2003, ISBN: 0521540518.
- [17]M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”, *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [18]O. Chum and J. Matas, “Matching with prosac-progressive sample consensus”, in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, IEEE, vol. 1, 2005, pp. 220–226.
- [19]P. H. Torr and A. Zisserman, “Mlesac: A new robust estimator with application to estimating image geometry”, *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.
- [20]O. Chum, J. Matas, and J. Kittler, “Locally optimized ransac”, in *Joint Pattern Recognition Symposium*, Springer, 2003, pp. 236–243.
- [21]O. Chum, T. Werner, and J. Matas, “Two-view geometry estimation unaffected by a dominant plane”, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, IEEE, vol. 1, 2005, pp. 772–779.
- [22]Y. Verdie, K. Yi, P. Fua, and V. Lepetit, “Tilde: A temporally invariant learned detector”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5279–5288.
- [23]K. Lenc and A. Vedaldi, “Learning covariant feature detectors”, in *European conference on computer vision*, Springer, 2016, pp. 100–117.

- [24]N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, “Quad-networks: Unsupervised learning to rank for interest point detection”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1822–1830.
- [25]D. DeTone, T. Malisiewicz, and A. Rabinovich, “Toward geometric deep slam”, *arXiv preprint arXiv:1707.07410*, 2017.
- [26]D. Mishkin, F. Radenovic, and J. Matas, “Repeatability is not enough: Learning affine regions via discriminability”, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 284–300.
- [27]A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, “Key. net: Key-point detection by handcrafted and learned cnn filters”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5836–5844.
- [28]S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4353–4361.
- [29]E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, “Discriminative learning of deep convolutional feature point descriptors”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 118–126.
- [30]V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, “Pn-net: Conjoined triple deep network for learning local image descriptors”, *arXiv preprint arXiv:1601.05030*, 2016.
- [31]A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, “Working hard to know your neighbor’s margins: Local descriptor learning loss”, *arXiv preprint arXiv:1705.10872*, 2017.
- [32]Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, “Sosnet: Second order similarity regularization for local descriptor learning”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 016–11 025.
- [33]D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description”, in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

- [34]Y. Ono, E. Trulls, P. Fua, and K. M. Yi, “Lf-net: Learning local features from images”, *arXiv preprint arXiv:1805.09662*, 2018.
- [35]X. Shen, C. Wang, X. Li, Z. Yu, J. Li, C. Wen, M. Cheng, and Z. He, “Rf-net: An end-to-end image matching network based on receptive field”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8132–8140.
- [36]J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, “R2d2: Repeatable and reliable detector and descriptor”, *arXiv preprint arXiv:1906.06195*, 2019.
- [37]M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable cnn for joint description and detection of local features”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101.
- [38]T.-Y. Yang, D.-K. Nguyen, H. Heijnen, and V. Balntas, “Ur2kid: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision”, *arXiv preprint arXiv:2001.07252*, 2020.
- [39]K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, “Learning to find good correspondences”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2666–2674.
- [40]J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, “Learning two-view correspondences and geometry using order-aware network”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5845–5854.
- [41]P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.
- [42]A. Bhowmik, S. Gumhold, C. Rother, and E. Brachmann, “Reinforced feature points: Optimizing feature detection and description for a high-level task”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4948–4957.
- [43]E. Brachmann and C. Rother, “Neural-guided ransac: Learning where to sample model hypotheses”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4322–4331.

- [44]E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, “Dsac-differentiable ransac for camera localization”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.
- [45]E. Brachmann and C. Rother, “Expert sample consensus applied to camera re-localization”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7525–7534.
- [46]D. DeTone, T. Malisiewicz, and A. Rabinovich, “Deep image homography estimation”, *arXiv preprint arXiv:1606.03798*, 2016.
- [47]T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar, “Unsupervised deep homography: A fast and robust homography estimation model”, *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2346–2353, 2018.
- [48]I. Rocco, R. Arandjelovic, and J. Sivic, “Convolutional neural network architecture for geometric matching”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6148–6157.
- [49]I. Rocco, M. Cimpoi, R. Arandjelovic, A. Torii, T. Pajdla, and J. Sivic, “Nc-net: Neighbourhood consensus networks for estimating image correspondences”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [50]I. Rocco, R. Arandjelović, and J. Sivic, “Efficient neighbourhood consensus networks via submanifold sparse convolutions”, in *European Conference on Computer Vision*, Springer, 2020, pp. 605–621.
- [51]Q. Zhou, T. Sattler, and L. Leal-Taixe, “Patch2pix: Epipolar-guided pixel-level correspondences”, in *CVPR*, 2021.
- [52]V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, “Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173–5182.
- [53]T. Lindeberg, “Scale-space theory: A basic tool for analyzing structures at different scales”, *Journal of applied statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.
- [54]X. Yang and K.-T. Cheng, “Ldb: An ultra-fast feature for scalable augmented reality on mobile devices”, in *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*, IEEE, 2012, pp. 49–57.

- [55]T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context”, in *European conference on computer vision*, Springer, 2014, pp. 740–755.
- [56]R. Pautrat, V. Larsson, M. R. Oswald, and M. Pollefeys, “Online invariance selection for local feature descriptors”, in *European Conference on Computer Vision*, Springer, 2020, pp. 707–724.
- [57]R. Sinkhorn and P. Knopp, “Concerning nonnegative matrices and doubly stochastic matrices”, *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.
- [58]M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport”, *Advances in neural information processing systems*, vol. 26, pp. 2292–2300, 2013.
- [59]K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.
- [60]K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [61]S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks”, *arXiv preprint arXiv:1506.01497*, 2015.
- [62]A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection”, *arXiv preprint arXiv:2004.10934*, 2020.
- [63]T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection”, in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [64]K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn”, in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [65]K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”, in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [66]I. Biederman, “Recognition-by-components: A theory of human image understanding.”, *Psychological review*, vol. 94, no. 2, p. 115, 1987.
- [67]R. N. Shepard and J. Metzler, “Mental rotation of three-dimensional objects”, *Science*, vol. 171, no. 3972, pp. 701–703, 1971.

- [68]A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4161–4170.
- [69]D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.
- [70]J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “Deepmatching: Hierarchical deformable dense matching”, *International Journal of Computer Vision*, vol. 120, no. 3, pp. 300–323, 2016.
- [71]X. Shen, F. Darmon, A. A. Efros, and M. Aubry, “Ransac-flow: Generic two-stage image alignment”, *arXiv preprint arXiv:2004.01526*, 2020.
- [72]U. S. Parihar, A. Gujarathi, K. Mehta, S. Tourani, S. Garg, M. Milford, and K. Krishna, “Rord: Rotation-robust descriptors and orthographic views for local feature matching”, 2021.
- [73]P. Truong, M. Danelljan, and R. Timofte, “Glu-net: Global-local universal network for dense flow and correspondences”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6258–6268.
- [74]U. Efe, K. G. Ince, and A. Alatan, “Dfm: A performance baseline for deep feature matching”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2021, pp. 4284–4293.
- [75]Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, “Image matching across wide baselines: From paper to practice”, *International Journal of Computer Vision*, vol. 129, no. 2, pp. 517–547, 2021.
- [76]M. Dusmanu, J. L. Schönberger, and M. Pollefeys, “Multi-view optimization of local feature geometry”, in *European Conference on Computer Vision*, Springer, 2020, pp. 670–686.
- [77]J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “LoFTR: Detector-free local feature matching with transformers”, *CVPR*, 2021.
- [78]G. Bradski, “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools*, 2000.
- [79]P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, *Superglue github repository*, <https://github.com/magicleap/SuperGluePretrainedNetwork>, 2020. (visited on 07/24/2021).

- [80]U. Efe, K. G. Ince, and A. Alatan, *Dfm github repository*, <https://github.com/ufukefe/DFM>, 2021. (visited on 07/24/2021).
- [81]Q. Zhou, T. Sattler, and L. Leal-Taixe, *Patch2pix github repository*, <https://github.com/GrumpyZhou/patch2pix>, 2021. (visited on 07/24/2021).
- [82]D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, “Magsac++, a fast, reliable and accurate robust estimator”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1304–1312.
- [83]D. Mishkin, *Evaluating opencv’s new ransacs*, <https://opencv.org/evaluating-opencvs-new-ransacs/>, 2021. (visited on 07/24/2021).

