Effect of Parameter Optimization on Classical and Learning-based Image Matching Methods

Ufuk Efe, Kutalmis Gokalp Ince, A. Aydin Alatan
Department of Electrical and Electronics Engineering, Center for Image Analysis (OGAM)
Middle East Technical University, Ankara, Turkey
ufuk.efe, kutalmis, alatan @ metu.edu.tr

Abstract

Deep learning-based image matching methods are improved significantly during the recent years. Although these methods are reported to outperform the classical techniques, the performance of the classical methods is not examined in detail. In this study, we compare classical and learning-based methods by employing mutual nearest neighbor search with ratio test and optimizing the ratio test threshold to achieve the best performance on two different performance metrics. 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. Throughout the experiments, we demonstrated that SuperGlue is the state-of-the-art technique for the image matching problem on HPatches dataset. However, if a single parameter, namely ratio test threshold, is carefully optimized, a well-known traditional method SIFT performs quite close to SuperGlue and even outperforms in terms of mean matching accuracy (MMA) under 1 and 2 pixel thresholds. Moreover, a recent approach, DFM, which only uses pre-trained VGG features as descriptors and ratio test, is shown to outperform most of the well-trained learning-based methods. Therefore, we conclude that the parameters of any classical method should be analyzed carefully before comparing against a learning-based technique.

1. Introduction

Determining pixel-to-pixel correspondences between two images is one of the fundamental problems in computer vision. There exists various classical ”hand-crafted” approaches, such as SIFT [23], SURF [5], ORB [29], KAZE [1], AKAZE [2], as well as some recently developed learning-based methods, SuperPoint [12], SuperGlue [31], Patch2Pix [36] and DFM [15]. All these techniques are frequently used in many applications, such as image matching, camera relocalization, pose estimation, Simultaneous Localization and Mapping (SLAM), and Structure-from-Motion (SfM).

Testing the performance of an algorithm in a fair manner 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 new learning-based method to the classical methods, this parameter adjustment procedure is 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 paper, we tried to compare classical and learning-based image matching algorithms in a fair manner on a well-known (HPatches) dataset [3] by optimizing the hyper-parameters of each algorithm on the selected dataset.

In a recent study [15], we demonstrated that applying mutual nearest neighbor search that exploits the ratio test on pre-trained VGG [32] features achieves the state-of-the-art performance. In order to investigate the resulting effect of this ratio threshold on some classical methods, we have also examined five classical image matching algorithms [23, 5, 29, 1, 2] on HPatches dataset in terms of *Mean Matching Accuracy (MMA)* [13] and *Homography Estimation Accuracy (HEA)* [12]. We have compared these conventional algorithms against four popular learning-based [12, 31, 36, 15] methods. We observed that ratio test threshold have a significant impact, as in [19, 14], on the performance of the methods SIFT, SURF, ORB, KAZE, AKAZE, SuperPoint and DFM. Similarly, the confidence parameter utilized in SuperGlue and Patch2Pix algorithms also affects their performance. This study presents the result of comprehensive experiments on these nine algorithms with different ratio test and confidence thresholds to reveal MMA and HEA performances of those algorithms for 1-10 pixel thresholds. By using these results, we present the optimal parameters for each algorithm to maximize MMA and HEA for required pixel threshold.

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 the 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 [36, 15].

Throughout the experiments, we observed 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 learning-based algorithms. In other words, we demonstrate and argue that classical algorithms can still perform close to the state-of-the-art for image matching task, at least in 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.

Hence, in this study, our contribution is threefold;

i. We show 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.

ii. 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.

iii. We provide an experimental setup on `https://github.com/ufukefe/IME` 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.

2. Related Work

In order to determine pixel-wise correspondences between two images, the classical approaches follow detection, description, and matching steps, while learning-based methods either follow these steps or carry out all of these steps in a single stage.

2.1. Classical Image Matching Algorithms

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

SIFT (Scale-Invariant Feature Transform) [23] detects features in the scale-space simply utilizing Difference of Gaussians (DoG), which is an approximation of Laplacian of Gaussian (LoG) [22]. In this manner, SIFT is able to detect blobs in varying scales with their orientation. Next, considering the dominant orientation and creating gradient 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) [5] goes a little 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 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) [29] runs FAST (Features from Accelerated Segment Test) [28] as a feature point detector together with computing keypoint's orientation. FAST 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.

KAZE [1] detector uses nonlinear scale spaces instead of Gaussian scale-space representations that are employed by SIFT. The motivation is that; nonlinear scale-space considers objects' natural boundaries, unlike Gaussian scale-space, which does not regard them due to smoothing the details and noise at all scale levels to the same degree. In contrast to SIFT, which uses the (DoG) to process the 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 [2] 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 [35], to create the feature descriptor vectors.

After features are detected and described with a feature extractor algorithm, they should be matched between frames. *Mutual Nearest Neighbor Search (MNNS)* 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 [23] 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. This whole feature matching strategy is denoted as *Mutual Nearest Neighbor Search with Bidirectional Ratio Test (MNNSwBRT)*.

2.2. Learning-based Image Matching Algorithms

Among the recently proposed learning-based techniques, SuperPoint and SuperGlue algorithms follow the classical image matching pipeline, while Patch2Pix and DFM techniques directly output the matched features between two images.

SuperPoint [12] jointly detects keypoints and computes relevant descriptor vectors. In this method, the outputs of the MagicPoint [11] detector is first exploited and a novel self-supervision strategy, namely Homographic Adaptation, is utilized to create a pseudo ground-truth interest point. Then, SuperPoint network is jointly trained in such a way that giving the keypoint confidence of each pixel and their corresponding descriptor vectors.

SuperGlue [31] can be considered as a feature matcher that computes the matches between two sets consisting of detected features and corresponding descriptor vectors. SuperGlue basically improves the descriptor vectors considering the cross and self attentions by the help of a graph neural network. At the final step, SuperGlue algorithm learns to match features optimally by using differentiable Sinkhorn algorithm [33, 8].

Patch2Pix [36] starts by matching the deepest features of truncated ResNet-34 [18] network by employing NCNet [27] 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.

The recent DFM method [15] uses only a pre-trained classification network and well-studied conventional computer vision techniques, such as hierarchical refinement and ratio test. DFM first aligns two images by matching the terminal layers of the pre-trained VGG19 [32] network; next, by the ratio test, DFM starts to match features of the terminal layers and refines those matches in a hierarchical way upto the first layers of VGG.

It is an essential fact that all these learning-based approaches need training data. Specifically, SuperPoint exploits MagicPoint [11] detector, which is trained using Synthetic Shapes dataset [11], and uses MS-COCO dataset [21] after labeling in a self-supervised manner. SuperGlue, on the other hand, is separately trained on different datasets, which are Oxford and Paris [26], ScanNet [9], and MegaDepth [20], for every particular problem, namely homography, indoor and outdoor. Patch2Pix utilizes ResNet34 [18] backbone, trained on ImageNet [10], and its refinement network is trained on MegaDepth. Even DFM, which uses a pre-trained VGG-19 [32] extractor, naturally needs this off-the-shelf network trained on ImageNet. Hence, all learning-based methods depend on the dataset characteristics, such as content and annotation quality.

Finally, it should be noted that while Patch2Pix and DFM directly output the putative matches, SuperPoint requires a feature matcher at its final stage as classical algorithms; this matcher step might be either MNNSwBRT or SuperGlue.

3. Experimental Setup

We have constructed an experimental setup in order to measure the performances of 5 classical and 4 learning-based image matching algorithms on HPatches dataset in terms of widely used metrics MMA and HEA by sliding only one parameter, either ratio test or confidence.

3.1. Dataset

HPatches dataset [3] 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 [13], we left out large images and made evaluations on 52 illumination sequences and 56 viewpoint sequences in order to make all algorithms work and to be coherent with the literature.

3.2. Performance Metrics

3.2.1 Mean Matching Accuracy (MMA)

Mean Matching Accuracy (MMA) is a widely used performance metric, and recently many state-of-the-art works [13, 36, 25, 15] reported 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 a correct match if the distance between the reprojected feature point with ground-truth homography and its corresponding match point is less than given pixel threshold. In our experiments, we vary the threshold from 1 pixel to 10 pixels as in the literature.

3.2.2 Homography Estimation Accuracy (HEA)

Homography Estimation Accuracy (HEA) is another widely used metric for image matching evaluation and used in [12, 36, 34, 15] 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 in all of our experiments. 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 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.

3.3. Algorithms

We use OpenCV [6] 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 [30] 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 used the original implementation [16] of the algorithm which again takes the advantage of MNNSwBRT. However, since DFM benefits from MNNSwBRT multiple times, to be fair, we kept 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 $\mathtt{0.5}^{(1/3)} \approx \mathtt{0.80}$.

For SuperGlue, we have used the GitHub repository [30] 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 [37] 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.

Method	Threshold
SIFT [23] + NN	Ratio Test Threshold
SURF [5] + NN	Ratio Test Threshold
ORB [29] + NN	Ratio Test Threshold
KAZE [1] + NN	Ratio Test Threshold
AKAZE [2] + NN	Ratio Test Threshold
SuperPoint [12] + NN	Ratio Test Threshold
SuperPoint + SuperGlue [31]	(1 - Confidence)
Patch2Pix [36]	(1 - Confidence)
DFM [15]	(Ratio Test Threshold)^3 for first 3 layers

Table 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*.

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 [4], which is available from OpenCV 4.5.2 version, and whose success is demonstrated in [24]. We adjust the other RANSAC's parameters as following: ransacReprojThreshold=3.0, maxIters=5000, confidence=0.9999.

4. Experimental Results

We followed the procedure explained in Section 3 and evaluated nine algorithms on HPatches dataset, with varying thresholds, which are different for each algorithm and defined in Table 1.

4.1. Effect of Matching Threshold

Figure 1 illustrates the performances of 5 classical and 4 learning-based image matching algorithms with varying thresholds in terms of MMA and HEA with different pixel thresholds on HPatches dataset. The threshold is defined as the ratio test threshold for MNNSwBRT matcher for SIFT, SURF, ORB, KAZE, AKAZE, and SuperPoint al-

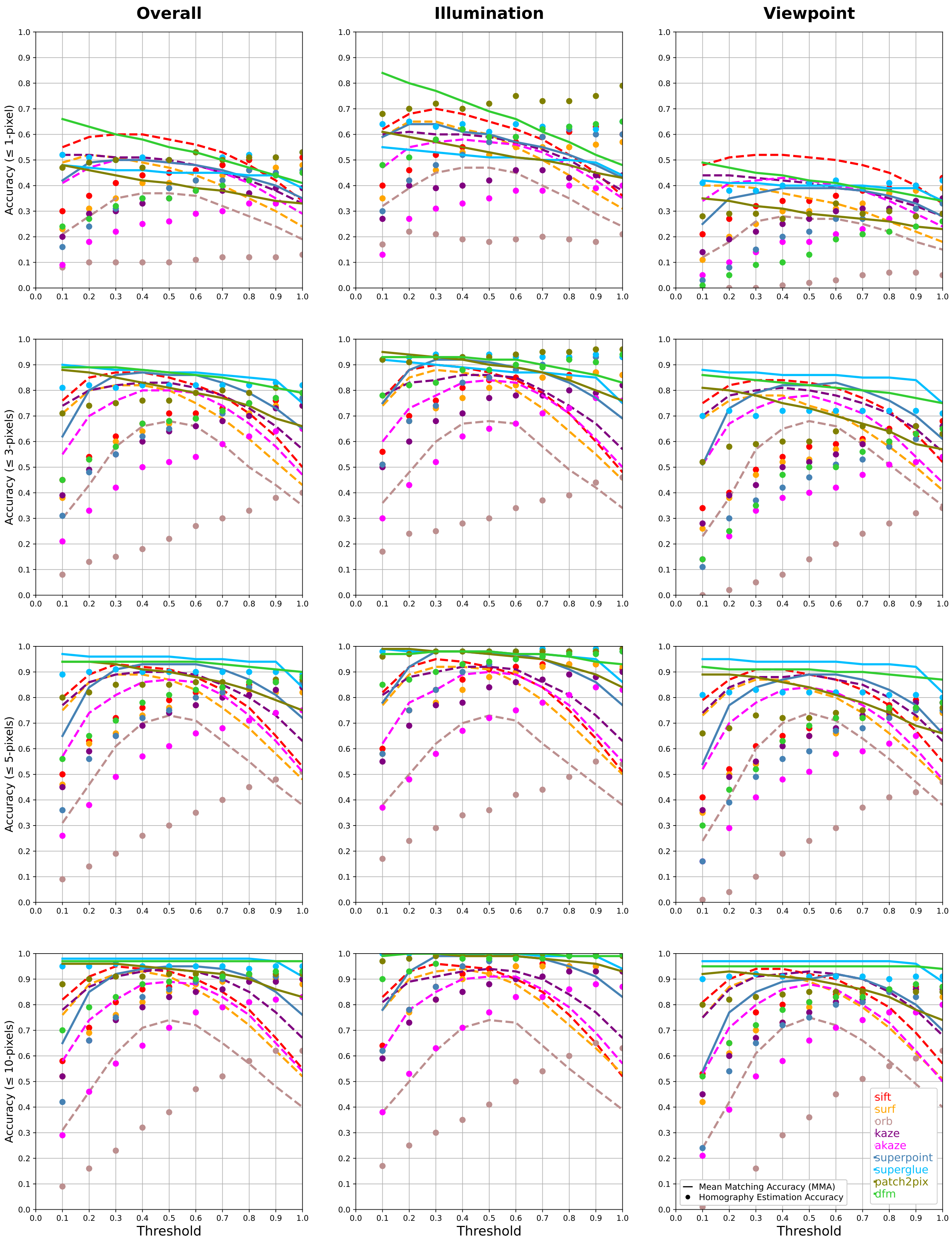

Figure 1. **Experimental Results** of 5 classical and 4 learning-based algorithms in terms of MMA and HEA on HPatches [3] dataset. MMA results are shown as dashed lines for classical algorithms and straight lines for learning-based algorithms, where HEA results are shown as dots. Individual plots show the accuracy with varying threshold values while rows indicate different pixel thresholds.

Method	Mean Matching Accuracy (MMA)								Homography Estimation Accuracy (HEA)							
	≤1px		≤3px		≤5px		≤10px		≤1px		≤3px		≤5px		≤10px	
	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.	Acc.	Thr.
SIFT [23] + 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 [5] + 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 [29] + 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 [1] + 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 [2] + 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 [12] + 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 [31]	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 [36]	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 [15]	**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

Table 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.

gorithms, (ratio test threshold)^3 for DFM and (1 − confidence) for SuperGlue and Patch2Pix algorithms. A smaller threshold means strict matching and returns fewer matches, while larger threshold values mean loose matching and return more matches.

In Figure 1, it can be clearly observed that the 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 [19]. Moreover, the effect of the 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, meaning that RANSAC is powerful enough to handle putative match sets consists of some amount of outliers and further giving better results.

4.2. Comparisons based on Best Accuracy Values

Table 2 is established 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 the table, we again notice the monotonic increasing behavior of HEA with respect to threshold, and in contrast, MMA maximizes in lower threshold values. That means strict thresholds result in correct matches while loose thresholds result in more matches and increase the accuracy of the homography estimation with the help of RANSAC. Last but not 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 with only adjusting a single parameter. For 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 make these algorithms outperform the state-of-the-art.

4.3. Comparisons based on Best Area Under Accuracy Curves

Table 3 is constructed by selecting a threshold for each algorithm to maximize the average accuracy for different pixel thresholds from 1 to 10 pixels. This average gives area under the accuracy curve (AUC) shown in Figure 2. Table 3 also illustrates the number of matches for relevant thresholds, indicating the fact that high MMA can be achieved

Method		MMA			HEA		
	#Features	AUC	Thr.	#Matches	AUC	Thr.	#Matches
SIFT [23] + NN	4572	***89.6***	0.3	478	82.1	0.9	1293
SURF [5] + NN	6003	85.6	0.3	276	80.0	0.9	1378
ORB [29] + NN	499	69.1	0.5	19	48.0	1.0	170
KAZE [1] + NN	3120	86.3	0.5	544	79.8	1.0	1287
AKAZE [2] + NN	2694	82.9	0.5	219	72.0	1.0	1011
SuperPoint [12] + NN	921	88.3	0.5	253	82.6	1.0	506
SuperPoint + SuperGlue [31]	921	**91.6**	0.1	441	**86.8**	0.6	482
Patch2Pix [36]	-	89.2	0.1	723	***84.1***	0.9	1434
DFM [15]	-	**91.6**	0.1	881	84.0	1.0	16619

Table 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 and the number of matched features using related threshold values are indicated.

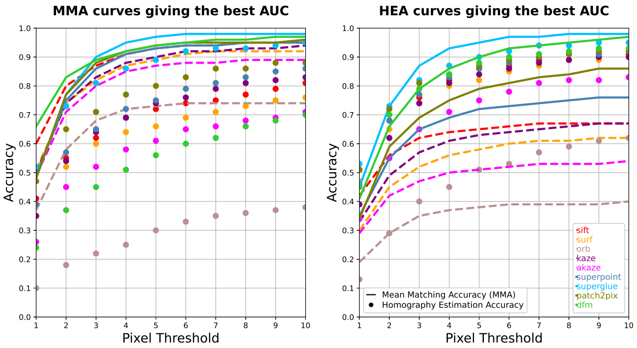

Figure 2. **Overall MMA and HEA curves** that give the best AUC for each algorithm. These curves indicate the best possible AUC result for each algorithm can achieve on HPatches [3] dataset by only optimizing a single parameter ratio test or match confidence. For each setting that gives the best AUC for one metric, the result of the other metric is also presented.

with higher confidence and less number of matches, while high HEA can be obtained with loose confidence and hence a vast number of matches. The most cardinal demonstration of Table 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 DFM as a learning-based method, it can be considered as almost a classical approach as it utilizes the deep features extracted by a pre-trained VGG-19 network, 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 underline that DFM has no specific training procedure for image matching task.

Figure 2 exhibits the MMA and HEA curves that give the best AUC for each algorithm. The figure utilizes the thresholds given in Table 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 [13, 36, 25, 15].

5. Conclusions

In this study, we have evaluated five classical and four learning-based algorithms for image matching task on well-known HPatches dataset [3]. We demonstrated the effect of the ratio test threshold for feature matching through experiments. Furthermore, we showed that classical methods are quite powerful so that they still perform very close to state-of-the-art. Specifically, one of the most popular methods nearly two decades, SIFT [23] achieves almost state-of-the-art performance by only optimizing ratio test threshold, which is neglected in previous studies. DFM [15] algorithm also demonstrates the feature matching capability of classical techniques by achieving state-of-the-art performance with only practicing well-established classical computer vision techniques on top of a pre-trained deep learning backbone.

Although our arguments may be limited with 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 emphasize a learning-based technique, SuperPoint [12] + SuperGlue [31], as 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.

References

[1] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In *European conference on computer vision*, pages 214–227. Springer, 2012.

[2] P. F. Alcantarilla and T. Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.

[3] 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*, pages 5173–5182, 2017.

[4] 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*, pages 1304–1312, 2020.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] D. DeTone, T. Malisiewicz, and A. Rabinovich. Toward geometric deep slam. *arXiv preprint arXiv:1707.07410*, 2017.

[12] 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*, pages 224–236, 2018.

[13] 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*, pages 8092–8101, 2019.

[14] M. Dusmanu, J. L. Schönberger, and M. Pollefeys. Multi-view optimization of local feature geometry. In *European Conference on Computer Vision*, pages 670–686. Springer, 2020.

[15] 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*, pages 4284–4293, June 2021.

[16] U. Efe, K. G. Ince, and A. Alatan. Dfm github repository. `https://github.com/ufukefe/DFM`, 2021.

[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*, 24(6):381–395, 1981.

[18] 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*, pages 770–778, 2016.

[19] 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*, 129(2):517–547, 2021.

[20] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[21] 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*, pages 740–755. Springer, 2014.

[22] T. Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[24] D. Mishkin. Evaluating opencv's new ransacs. `https://opencv.org/evaluating-opencvs-new-ransacs/`, 2021.

[25] 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.

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

[27] I. Rocco, M. Cimpoi, R. Arandjelovic, A. Torii, T. Pajdla, and J. Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[28] 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*, volume 2, pages 1508–1515. Ieee, 2005.

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

[30] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue github repository. `https://github.com/magicleap/SuperGluePretrainedNetwo` 2020.

[31] 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*, pages 4938–4947, 2020.

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

[33] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[34] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021.

[35] 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)*, pages 49–57. IEEE, 2012.

[36] Q. Zhou, T. Sattler, and L. Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, 2021.

[37] Q. Zhou, T. Sattler, and L. Leal-Taixe. Patch2pix github repository. https://github.com/GrumpyZhou/patch2pix, 2021.