

COMPARING ROTATION-ROBUST MECHANISMS IN LOCAL FEATURE MATCHING: HAND-CRAFTED VS. DEEP LEARNING ALGORITHMS

A. Rahman¹, L. G. Lie², H. Wahyudi³, and F. Heltha^{1,2*}

¹Department of Electrical and Computer Engineering, Universitas Syiah Kuala, Banda Aceh 23111, INDONESIA

²Faculty of Engineering, UCSI University, Cheras, Kuala Lumpur 56000, MALAYSIA

³Department of Mechanical Engineering, Universitas Mercu Buana, Kembangan, Jakarta 11650, INDONESIA

Abstract

The objective of this research is to conduct a performance comparison between hand-crafted feature matching algorithms and deep learning-based counterparts in the context of rotational variances. Hand-crafted algorithms underwent testing utilizing FLANN (Fast Library for Approximate Nearest Neighbors) as the matcher and RANSAC (Random sample consensus) for outlier detection and elimination, contributing to enhanced accuracy in the results. Surprisingly, experiments revealed that hand-crafted algorithms could yield comparable or superior results to deep learning-based algorithms when exposed to rotational variances. Notably, the application of horizontally flipped images showcased a distinct advantage for deep learning-based algorithms, demonstrating significantly improved results compared to their hand-crafted counterparts. While deep learning-based algorithms exhibit technological advancements, the study found that hand-crafted algorithms like AKAZE and AKAZE-SIFT could effectively compete with their deep learning counterparts, particularly in scenarios involving rotational variances. However, the same level of competitiveness was not observed in horizontally flipped cases, where hand-crafted algorithms exhibited suboptimal results. Conversely, deep learning algorithms such as DELF demonstrated superior results and accuracy in horizontally flipped scenarios. The research underscores that the choice between hand-crafted and deep learning-based algorithms depends on the specific use case. Hand-crafted algorithms exhibit competitiveness, especially in addressing rotational variances, while deep learning-based algorithms, exemplified by DELF, excel in scenarios involving horizontally flipped images, showcasing the unique advantages each approach holds in different contexts.

Keywords: Local Feature Matching, Deep Learning, Rotation Robust Feature Matching, CNN

*Corresponding author: Tel. +62 651 7553205

E-mail address: fheltha@gmail.com

DOI: 10.22441/ijimeam.v5i3.24794

1. Introduction

Local feature matching, a method for identifying and correlating features in two similar images through algorithms, has found extensive use in computer vision applications. Its applications span Simultaneous Localization and Mapping (SLAM), image processing, pattern recognition, and object detection, making it integral in robotics, mechatronics, and intelligent control systems [1]. Notable instances include medical image registration using SURF (Speeded Up Robust Features) for biomedical purposes [2] in 2016, early lung cancer detection using SVM (Support Vector Machine) classifiers in biomedical image processing [3] in 2017, and the prediction of CT images from MRI data through feature matching [4] in 2018. In Geographic Information Systems, feature matching has been applied to monitor and protect cultivated land [5] in 2019 and [6] in 2020.

The prevailing matching techniques typically involve three stages: feature detection, feature description, and feature matching. During detection, salient points like corners are identified as interest points in each image. Local descriptors are then extracted from areas near these points. The feature identification and description phases produce two sets of interest points with descriptors, facilitating processes like nearest neighbor searches or more complex matching algorithms that ultimately identify point-to-point correspondences [7].

Deep learning methods have revolutionized feature matching in image processing, addressing challenges like image rotation. Recent attention has focused on learned local feature techniques such as SuperPoint, LoFTR, D2-Net, R2D2, and RoRD, known for their enhanced robustness in handling rotational variations compared to traditional hand-crafted methods like SIFT. A comprehensive analysis

of the performance disparity between classical hand-crafted algorithms and deep learning-based approaches, particularly under rotational variations, is crucial for advancing our understanding of rotational robust feature matching.

Local feature matching typically involves three steps: feature detection, feature description, and feature matching. For feature detection and description, algorithms providing detectors and descriptors play a pivotal role. Classical detectors like SIFT and SURF, as well as binary-based detectors like BRIEF and ORB, are commonly employed. Meanwhile, feature matching can be accomplished through methods such as Brute Force Matching, kNN (k-Nearest Neighbor), or FLANN Based Matcher (Fast Library for Approximate Nearest Neighbors). RANSAC (Random sample consensus) is commonly applied to optimize matching results from pre-matched feature points.

1.1 Classical Feature Detector and Descriptor Algorithm

Feature detection in image processing relies on various algorithms, often emphasizing invariant features to ensure robust matching, independent of imposed constraints. A fundamental requirement for these features is resistance to scale, position, rotation, or viewpoint modifications [1]. Notable algorithms commonly employed in feature matching include:

a. SIFT (Scale-Invariant Feature Transform)

SIFT, introduced by Lowe, provides both a detector and a descriptor, offering invariance to scale, rotation, illumination, and viewpoint changes [1]. Utilizing the difference of Gaussian (DOG) pyramids, SIFT identifies feature points, significantly reducing scale discrepancies between stereo images. Its high-dimensional descriptors, however, result in a computationally intensive process [8].

b. SURF (Speeded Up Robust Feature)

A faster alternative to SIFT, SURF approximates the second-order derivative of the Gaussian filter using integral images. Employing box filters for DoG replication, SURF utilizes a BLOB detector based on the Hessian matrix and wavelet responses for orientation assignment and feature description. Its use of wavelet responses enhances matching speed [9, 10].

c. BRIEF Features (Binary Robust Independent Elementary)

BRIEF employs various pixel coordinates in the smoothed feature support window, generating short binary descriptors. The Hamming distance replaces

the traditional Euclidean distance in binary descriptors. After Gaussian smoothing and random pixel selection, BRIEF creates a descriptor vector. Invariance to scale and rotation requires pairing with an appropriate detector, and unnecessary orientation detection should be avoided [11].

d. ORB (Oriented FAST and Rotated BRIEF)

Introduced in 2011, ORB combines direction-normalized BRIEF description with modified FAST detection techniques. ORB is invariant to scale, rotation, and some affine changes. An alternative affine transformation-based ORB technique improves feature point extraction speed but may introduce redundant points and longer matching times [12, 13].

1.2 Feature Matching

1) Brute Force Matching

The simplest technique that may be used for pattern finding is the brute-force algorithm, often known as the "naive" algorithm. It doesn't need the pattern or the text to be pre-processed [17]. The Euclidean distance is used by Brute Force to determine the distance between two points. Brute Force matches N feature key points from a source image with N feature key points of the target image. For each key point descriptor of the feature set from the source image, brute force matcher will find the closest key point descriptor in the feature set of the target image by trying to pair each descriptor in the target set [1].

2) kNN (K-Nearest Neighbor) Based Matching

The non-parametric K-Nearest Neighbors algorithm is used for regression and classification. The K-Nearest Neighbor (kNN) technique has been extensively used in data mining and machine learning since it is simple yet highly useful and works brilliantly. To predict the labels of test data points, classification is employed after training sample data. Although academics have offered additional classification methods over the past few decades, KNN (K-Nearest Neighbor) continues to be one of the most popular methods [18].

The basic idea behind the approach is that a query point q can be determined to fall into a particular category if the majority of the k most similar samples to a query point q in the feature space belong to the same category. This approach is known as the K Nearest Neighbor algorithm because it uses the distance in the feature space to estimate similarity. At the outset of the method, a train data set with precise classification labels should be known. Then, the distances between each point in the train data set and

the query data q , whose label is unknown, and which is represented by a vector in the feature space should be determined. After sorting the distance calculation results, the test point q class label may be determined based on the labels of the k closest points in the train data set [19].

3) FLANN Based Matcher

In many applications, including image identification, data compression, pattern recognition and classification, machine learning, document retrieval systems, data analysis and statistics etc. The problem of nearest neighbor search is of much importance. Unfortunately, it appears to be very challenging to solve this problem in high dimensional spaces, as no solution appears to outperform the traditional brute-force search. As a result, there is growing interest in a class of algorithms that perform approximate nearest neighbor searches. These algorithms have shown to be orders of magnitude faster than exact nearest neighbor search algorithms and to be a good enough approximation in the majority of practical applications [20].

FLANN is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. Flann is based on approximate nearest neighbor algorithm [21]. A matching should be performed once distinct key points and descriptors are retrieved from both photos. The FLANN library has a selection of methods that have been enhanced for high dimensional features and fast nearest neighbor search in huge datasets [22].

In 2009, [20] developed the FLANN algorithm, which is based on the KD tree operation or K-means tree. The known characteristics of the distribution of the data set and the necessary spatial resource consumption establish the most appropriate search type and retrieval parameters. The FLANN algorithm typically requires an n -dimensional real vector space as the feature space. The key is to use Euclidean distance to find the closest point from the instance point's closest neighborhood. Equation to calculate Euclidean distance is shown on equation (1) below:

$$D(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}, \mathbf{y}\| = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (1)$$

The distance between the existing feature point pairs is "near" if the D value returned is lower, indicating a higher degree of similarity between the feature point pairs in the image. This approach uses the KD-Tree method to separate the data points in the n -dimensional spatial query brush into a number of distinct regions. Its purpose is to get the neighborhood's closest Euclidean distance to the query location. The KD-tree structure in the n -

dimensional space brush stores all of the Euclidean distances, making it simple and efficient to locate the location that is closest to the reference point. KD-tree structures use a recursive top-to-bottom search mechanism [23].

1.3 Deep Learning Based Feature Matching Algorithm

1) RoRD: Rotation-Robust Descriptors and Orthographic Views

In 2021, the RoRD method, as introduced by Reference [27], aimed at enhancing local feature matching by focusing on precise, pixel-level correspondences in images with highly variable viewpoints. The primary objective was to achieve accurate local feature matching by learning rotation-robust descriptors that remain invariant to significant changes in viewpoint. To broaden the application of these descriptors, the method incorporated a correspondence ensemble and an orthographic view generation technique. This approach maintained the lighting and scale invariance of the original D2-Net through an ensemble architecture, making the network robust to substantial changes in viewpoint.

The method's strategy involved a training pipeline and straightforward geometric operations, specifically in-plane rotations. These features enabled the researchers to demonstrate the acquisition of accurate feature correspondences across image pairs, even in the presence of large differences in viewpoint. The introduction of the ensemble approach RoRD + CE, which combined the initial knowledge from D2-Net with the recently acquired rotation-robust features, significantly enhanced the potential of RoRD. This combination outperformed both previous methods and established baselines on the HPatches dataset. Notably, for the Oxford RobotCar and the DiverseView Datasets, the method's performance was greatly improved by replacing perspective photos with orthographically modified images. This emphasized the necessity of rotation-robust descriptors in achieving superior performance in tasks such as pose estimation and opposite-viewpoint location recognition, indicating the method's effectiveness in challenging scenarios.

2) D2-Net

D2-Net, proposed by Reference [28], is a deep-learning-based feature matching algorithm designed to address the challenges associated with correspondence estimation using sparse local features. Traditional approaches often use a detect-then-describe strategy, identifying key points through a feature detector and subsequently describing them

with image patches. While sparse features have benefits such as effective Euclidean distance matching and memory-efficient representation, they may struggle in scenarios with weak textures or significant appearance changes.

The observed decline in performance is attributed to the lack of repeatability in the key point detector, which focuses on small image portions. In contrast, local descriptors consider larger patches, capturing higher-level structures. This discrepancy results in unstable detections during significant appearance changes. Despite this, it has been demonstrated that local descriptors, rather than key points, can still be successfully matched. Techniques skipping the detection stage and densely extracting descriptors show better performance in challenging situations, albeit with increased matching times and memory usage.

In addressing these challenges, the researchers proposed a describe-and-detect strategy for sparse local feature detection and description. This involved utilizing a Deep Convolutional Neural Network (CNN) to construct feature maps, from which descriptors were computed, and key points were discovered. Although this method may be less effective than traditional sparse features, the researchers showcased its precision for visual localization and Structure-from-Motion (SfM). The approach, while demonstrating a trade-off between robustness and accuracy, outperformed existing methods in camera localization, particularly in challenging scenarios like day-night transitions and indoor scenes. Despite the features being less precisely localized than classical detectors, they proved suitable for 3D reconstruction.

The researchers acknowledge the need for further improvements in key points' detection accuracy. They suggest potential enhancements, such as providing higher spatial resolution to CNN feature maps or incorporating a ratio test-type objective into their loss function, especially in SfM applications. This indicates a commitment to refining the method for broader and more accurate applications in feature matching and reconstruction tasks.

3) R2D2: Repeatability and Reliable Descriptor

The advancement in metric learning algorithms has surpassed traditional detectors and classic descriptors, such as SIFT, as highlighted by Reference [29]. These classical methods may struggle in repeating regions, hindering accurate matching, as they are trained on places that the detector deems repeatable. Common textures like tree leaves, skyscraper windows, or sea waves pose challenges for matching in natural photos.

To address these issues and enhance the feature matching process, the proposed method employs a unique approach. Confidence maps are individually estimated for repeatability and reliability, targeting two critical elements of key points and descriptors. The network generates associated repeatability and reliability confidence maps, along with dense local descriptors for each pixel. This ensures that the key points selected are both repeatable and trustworthy. The estimation of the odds that a key point is repeatable, and its descriptor is discriminative is achieved through the maximization of both confidence maps.

Reference [29] employs a distinctive unsupervised loss to train the key point detector, emphasizing repeatability, sparsity, and uniform coverage of the image. Departing from traditional triplet or contrastive loss methods, the local descriptor is trained using a listwise ranking loss, leveraging recent developments in metric learning based on an estimated Average Precision (AP) metric. To identify pixels with descriptors possessing high AP, indicating both discriminative and robust characteristics conducive to successful matching, a reliability confidence value is simultaneously learned.

The effectiveness of this formulation is demonstrated through extensive tests on various benchmarks. The approach unifies the repeatability and sparsity of the detector with a robust and discriminative descriptor. This showcases the method's capability to address challenges in feature matching, particularly in scenarios with repeating patterns or challenging textures, ensuring reliable and precise matching results.

The objective of this research is to conduct a performance comparison between hand-crafted feature matching algorithms and deep learning-based counterparts in the context of rotational variances. Hand-crafted algorithms underwent testing utilizing FLANN (Fast Library for Approximate Nearest Neighbors) as the matcher and RANSAC (Random sample consensus) for outlier detection and elimination, contributing to enhanced accuracy.

2. Experimental and Procedures

The research follows [31]. The feature-detector descriptions for SIFT, SURF, KAZE, AKAZE, ORB, and BRISK [31] are thoroughly compared in this article. The experimental findings offer detailed information and a variety of fresh perspectives that are important for making crucial choices in vision-based applications. Based on repeatability, SIFT, SURF, and BRISK were discovered to be the most scale-invariant feature detectors that have endured widely dispersed scale fluctuations. It is found that

ORB is the least scale invariant. Compared to other functions, ORB (1000), BRISK (1000), and AKAZE are more rotation invariant. ORB and BRISK are generally more resistant to affine alterations than the others. Compared to the others, SIFT, KAZE, AKAZE, and BRISK have greater picture rotation accuracy.

Even though ORB and BRISK are the most effective algorithms for detecting a large number of features, the time required to match such a large number of features increases the overall image-matching time. Contrarily, ORB (1000) and BRISK (1000) match images the quickest, but at the expense of precision. SIFT and BRISK are determined to have the highest overall accuracy for all kinds of geometric transformations and SIFT is declared the most accurate method.

In 2017, [32] published a journal that analyses feature matching performance with different combinations of detector and descriptors, with a variance of rotational angles on the images. Their findings showed trade-offs among metrics and performance standards when various feature detector-descriptor combinations were examined. The method detected weak features when a wide rotation range needed to be matched, and matches were not realized due to predefined threshold values. Conversely, the method finds too many features to match if a tight rotation range is sought, increasing the overall loop duration due to increased correct matches per unit of time.

The result of the experiment suggests that BRISK-BRIEF had the shortest overall running time, while the combination of SURF and SIFT had the longest running time. In addition, SIFT-SURF had a high accuracy rate of 98.41% over 35319.080 seconds. Also, for comparative purposes, the FAST-SURF combination method yields the best results for angular rotations of 30°, 15°, -15°, and -30°. Also, when comparing identical images, SURF-SURF and SURF-SIFT combinations produce the greatest results regarding the proportion of correct matches, the average angle at which key points are correctly matched, and minimum distance metrics.

The feature-matching parameters used in this research include:

1) Repeatability

The repeatability for a pair of photos is calculated by dividing the number of feature correspondences discovered between the pair of images by the lowest number of features discovered in the pair. The lowest number of features between the two images will be used as the denominator [33]. The repeatability metrics are shown in equation (2):

$$\text{Repeatability} = (\# \text{ of feature matches}) / (\# \text{ of minimum features between the two images}) \quad (2)$$

2) Matching Score

The matching score is the average ratio of detected features in a shared perspective region to the ground truth correspondences. By removing outliers, the matching score also shows the precision of the matching process. The matched features are perfectly inlier and perfectly outlier, respectively, according to matching scores of 1 and 0 [33]. The matching score metric is shown in equation (3):

$$\text{Matching Score} = (\# \text{ of inliers}) / (\# \text{ of minimum features between the two images}) \quad (3)$$

The accuracy and the run time metric show in equation (4) and (5):

$$\text{Accuracy} = (\# \text{ of inliers}) / (\# \text{ of matches}) \quad (4)$$

$$\text{Average Run Time} = (\text{sum of run time of tests done in the algorithm}) / (\text{total tests done in the algorithm}) \quad (5)$$

3) Mean Matching Accuracy (MMA)

The pixel-wise distances between the matched features and their ground-truth projections on the pair pictures are used to gauge how well the feature-matching task performed. After these values have been averaged throughout the whole dataset, the percentage of matches with matching errors less than the chosen threshold is then presented for thresholds ranging from 1 pixel to 10 pixels. The mean matching accuracy (MMA) metric is used to measure this [34].

The handcrafted feature-matching and deep learning-based algorithms' rotation robustness were compared. First, handcrafted algorithms were picked based on their reputation for being rotating robust. The algorithms picked were SIFT, AKAZE, ORB, and BRISK. Hybrid combinations of the algorithms were also tested between the four algorithms, though some combinations could not be tested due to the difference in data types produced by the detector. The possible hybrid combinations include SIFT- BRISK, ORB-SIFT, ORB-BRISK, AKAZE-BRISK, AKAZE- SIFT, AKAZE-ORB, BRISK-ORB, and BRISK-SIFT where the left algorithm in the name acts as the detector and the right algorithm in the name serves as the descriptor. Two algorithms that are known to be non-rotation robust, namely BRIEF and FREAK, were picked to be tested as well to compare the performance between the rotation-robust algorithms and non-rotation robust algorithms. Meanwhile, using the same reasoning, D2-Net, R2D2, DELF, and RoRD were picked as the deep

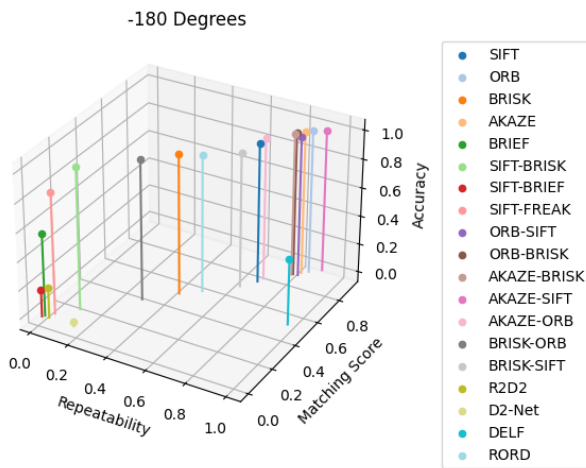


Fig. 1. Graph representation of parameter values at -180 degree rotation

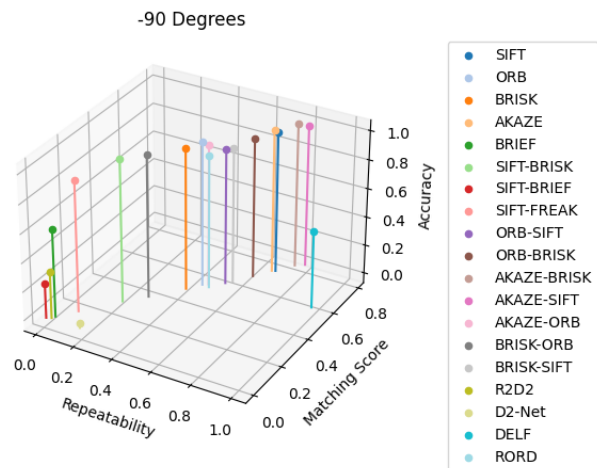


Fig. 3. Graph representation of parameter values at -90 degree rotation

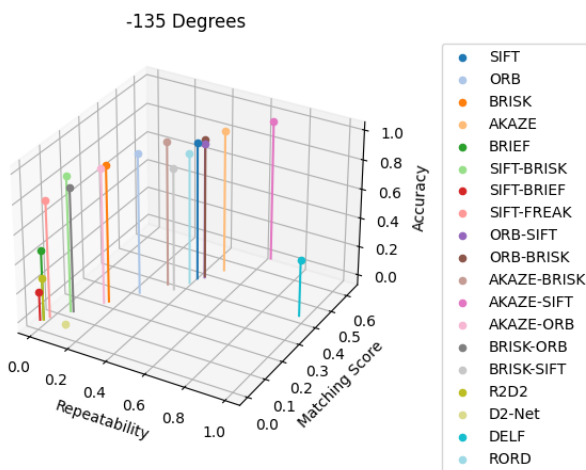


Fig. 2. Graph representation of parameter values at -135 degree rotation

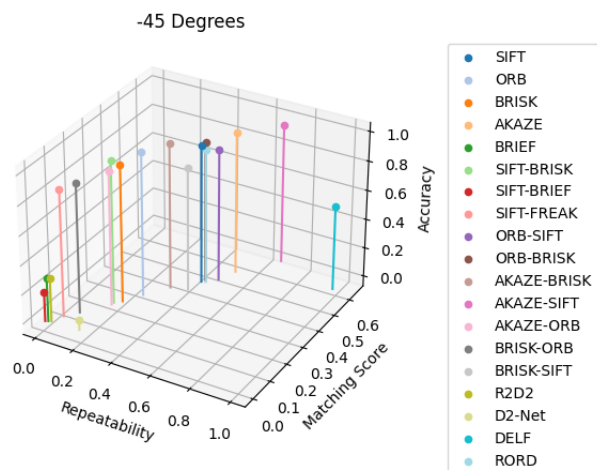


Fig. 4. Graph representation of parameter values at -45 degree rotation

learning-based algorithms to be tested because of their reputation for being rotation-robust.

In the experiments, there were a few control variables used. First, all experiments were to be done utilizing the free version of Google Colab using the Python3 runtime type settings and a GPU hardware accelerator with a GPU type of T4 to maintain a stable testing environment. For the handcrafted algorithms, matches were obtained with a FLANN-based matcher that returns the two nearest neighbors as the best matches. The ratio test was also applied to filter ambiguous matches. Then, RANSAC, which was set to have a residual threshold of 4, was used to obtain the inlier matches. On the other hand, the deep learning-based algorithms were tested using the original code provided in the algorithms' respective GitHub Repository.

Tests were done by subjecting the algorithms with rotated images in the range of $[-180^\circ, 135^\circ]$ with 45-degree intervals and lastly with a horizontally flipped

image, resulting in a total of 8 tests for each algorithm. All codes were modified to obtain the parameter values, such as the number of key points in the source image, the number of key points in the target image, the total number of matches, the number of inliers, repeatability, matching score, accuracy, and run time. Obtained results were tabulated, and graphs representing the performance of the algorithms were generated based on the results.

3. Results and Discussion

The research evaluated each method using the metrics mentioned above to measure the performance of that method where the image was rotated -180, -135, -90, -45, 45, 90, and 135 degrees and flipped the image horizontally. These metrics are shown in Fig. 1 – Fig. 8, respectively.

Three-dimensional graphs that represent the performance of each algorithm with regards to their repeatability, matching score, and accuracy value

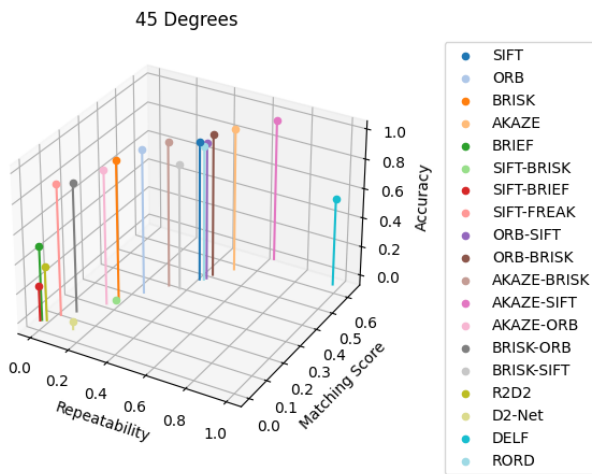


Fig. 5. Graph representation of parameter values at 45 degree rotation

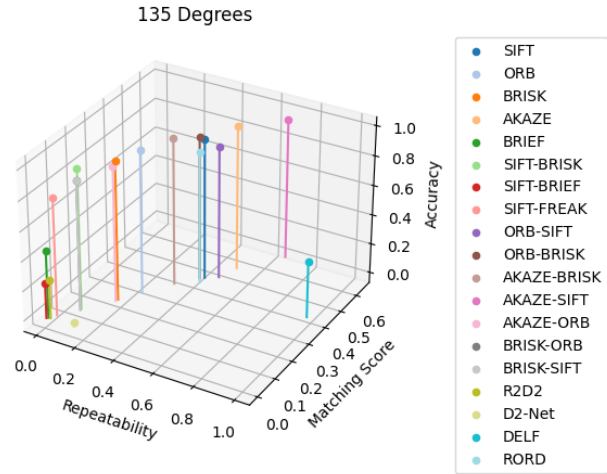


Fig. 7. Graph representation of parameter values at 135 degree rotation

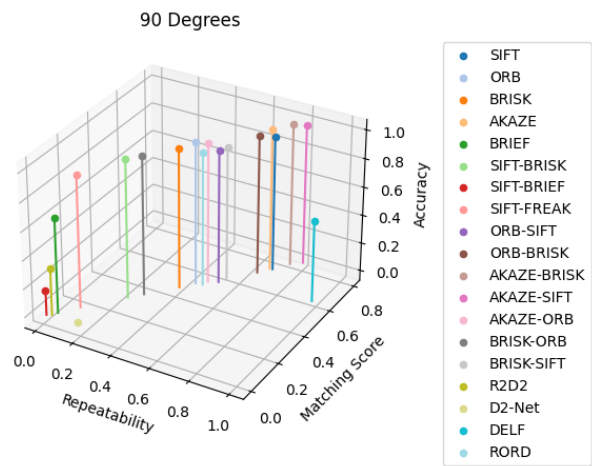


Fig. 6. Graph representation of parameter values at 90 degree rotation

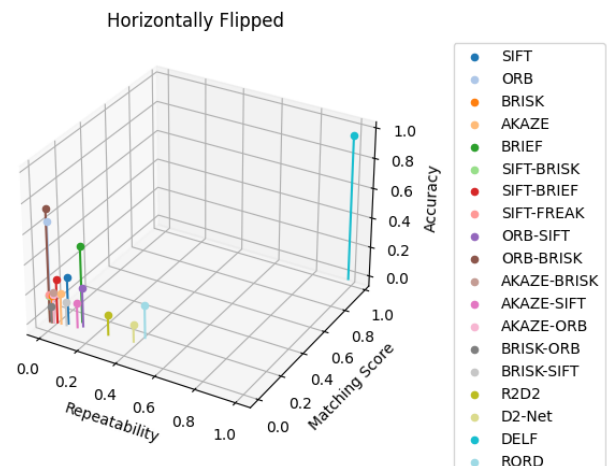


Fig. 8. Graph representation of parameter values when flipped horizontally

were generated, as seen in Fig. 1 to Fig. 9. From the generated figures, it could be noticed that the values obtained in 90 degrees intervals (-180° , 90° , -90°) tests had considerably better results than the values obtained in 45 degrees intervals (-135° , -45° , 45° , 135°). These differences could be observed in every algorithm tested.

In terms of repeatability, the DELF algorithm, being a deep learning-based feature matching algorithm, had the best repeatability value in every rotational degree variable, with every test resulting in one value. RoRD algorithm, on the other hand, had a mid-performance, with its repeatability value ranging from around 0.4 to 0.5. Meanwhile, R2D2 and D2-Net had the worst repeatability values among the algorithms, with most values under 0.2. Three hand-crafted feature matching algorithms were noticed to constantly have better repeatability values among the other algorithms as well, namely AKAZE and AKAZE-SIFT algorithms. As mentioned before, the

performance of the algorithms had a considerable decrease when subjected to 45-degree rotation intervals, as seen in the highest repeatability value that the algorithms could obtain, which was around 0.6. Meanwhile, a peak value around 0.8 could be observed on images with 90-degree rotation intervals. Regarding matching scores, both the DELF and RoRD algorithms as deep learning-based algorithms had mid-performance matching score values, ranging from around 0.3 to 0.5. Similar to repeatability value performance, AKAZE and AKAZE-SIFT algorithms also showed the best results in matching score values among other algorithms. The performance of the algorithms concerning the degree intervals was also similar to the performance in repeatability, where the peak matching score value was around 0.6 for 45-degree rotation intervals, while the peak matching score value for 90-degree rotation intervals was around 0.9.

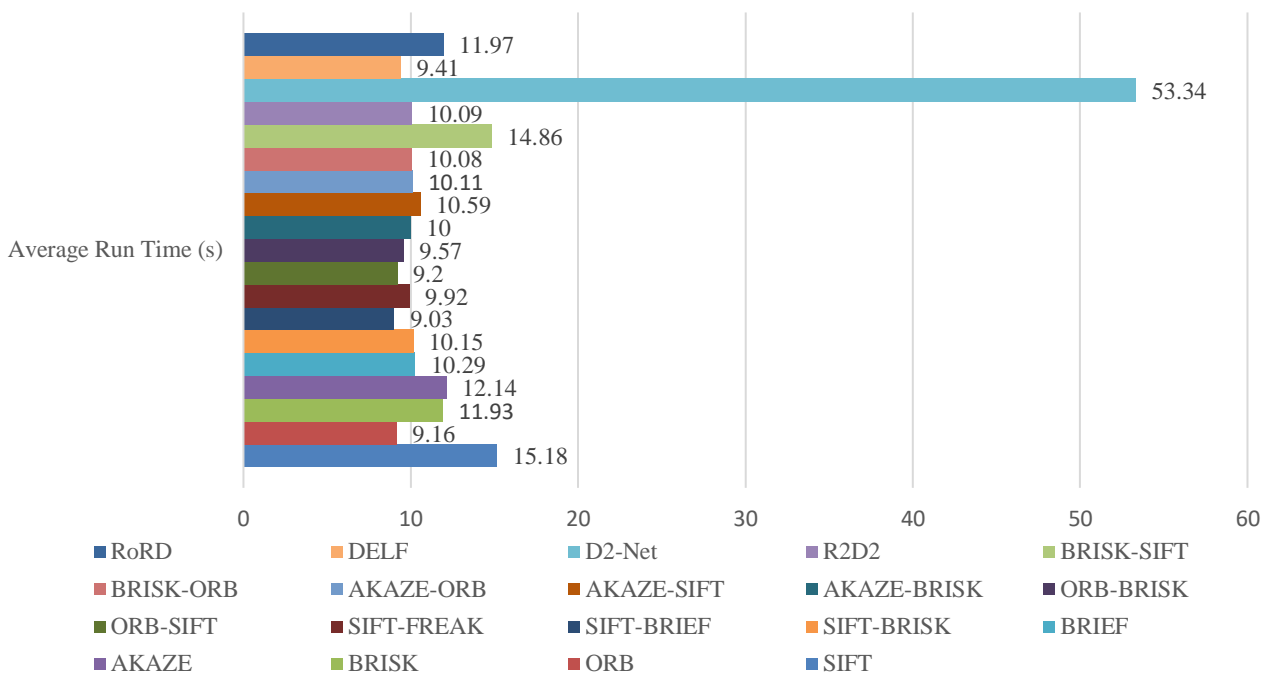


Fig. 9. Graph representation of average run time of each algorithm

RoRD algorithm showed a high accuracy value among the deep learning-based algorithms with values around 0.9, while DELFL had mid-values with numbers around 0.5. Meanwhile, R2D2 and D2-Net had low values below 0.3. Most classical hand-crafted algorithms, on the other hand, showed high accuracy values above 0.8.

In tests involving horizontally flipped images, only the DELFL algorithm had high repeatability, matching score, and accuracy parameter values, with every value close to 1. Comparing the deep learning-based algorithm and the hand-crafted algorithms when subjected to horizontally flipped images, the deep learning algorithms had slightly better results in their repeatability values around 0.2 to 0.4, while hand-crafted algorithms had values under 0.2. Most algorithms, including both deep learning-based and hand-crafted, had low matching score values of around 0.2 and below.

As seen in Fig. 9, the SIFT algorithm has one of the longest average run times among the hand-crafted algorithms. Even though BRIEF, SIFT-BRIEF, and SIFT-FREAK are non-rotational robust algorithms, these algorithms proved to have a short computational time. Excluding the non-rotational robust algorithms, the rotation robust hand-crafted algorithm with the shortest run time was ORB with a 9.16 second average run time. Other algorithms with ORB as the detector also had a shorter average run time, such as the ORB-SIFT algorithm with a 9.2-second average run time. Meanwhile, D2-Net had a considerably high average run time of 53.34 seconds among the deep

learning-based algorithms. On the other hand, DELFL had the shortest average run time of 9.41 seconds.

Comparing the overall performance of both the hand-crafted algorithms and the deep learning-based algorithms, it could be said that the hand-crafted algorithms have better performance in rotation variances, especially seen on SIFT, AKAZE, and AKAZE-SIFT. RoRD and DELFL algorithms as deep learning algorithms have passable performance but are not on par with the three mentioned hand-crafted algorithms. But in terms of horizontally flipped images, deep learning algorithms have the upper hand in feature point detection, especially when considering the number of matches they could find.

The performance of the algorithms could not be evaluated solely from their repeatability, matching score, and accuracy, as the number of key points and number of matches generated by the algorithms vary and greatly affect the three parameter values discussed earlier. To explain further, as SIFT has the highest key point detection accuracy by theory, the number of key points detected in the source image by the SIFT algorithm could be assumed to be the ground truth. In the test, the SIFT algorithm could detect up to 5442 key points in the source image. It could be expected for other algorithms to be capable of detecting a similar number of key points from the same source image. But some algorithms limit the number of key points they detect, such as ORB and DELFL, or algorithms that simply cannot detect key points as much as SIFT, as seen on AKAZE. Other factors could also affect the performance of the

algorithms, such as applying ratio tests on the algorithms, the applied matching threshold and matching method, or even the image used.

4. Conclusions

In the case of rotational variances, even though deep learning-based algorithms are more advanced in terms of technology, hand crafted algorithms such as AKAZE and AKAZE-SIFT also proved to be able to compete well with the deep learning-based algorithms. But the same could not be said for horizontally flipped cases, as hand crafted algorithms showed terrible results. On the other hand, deep learning algorithms such as DELF showed better results and accuracy. In conclusion, depending on the use case, hand crafted algorithms and deep learning-based algorithms have their own advantages.

References

- [1] Ali, H. K., & Whitehead, A. (2014). Modern to Historic Image Matching: ORB/SURF an Effective Matching Technique. In *Proceedings of International Conference on Computers and Their Application (CATA 2014)*.
- [2] Gupta, S., Chakarvarti, S. K., & Zaheeruddin. (2016). Medical image registration based on fuzzy c-means clustering segmentation approach using SURF. *International Journal of Biomedical Engineering and Technology*, 20(1), 33-50.
- [3] Kaucha, D. P., Prasad, P. W. C., Alsadoon, A., Elchouemi, A., & Sreedharan, S. (2017, September). Early detection of lung cancer using SVM classifier in biomedical image processing. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (pp. 3143-3148). IEEE.
- [4] Yang, W., Zhong, L., Chen, Y., Lin, L., Lu, Z., Liu, S., ... & Chen, W. (2018). Predicting CT image from MRI data through feature matching with learned nonlinear local descriptors. *IEEE Transactions on Medical Imaging*, 37(4), 977-987.
- [5] Song, F., Dan, T., Yu, R., Yang, K., Yang, Y., Chen, W., ... & Ong, S. H. (2019). Small UAV-based multi-temporal change detection for monitoring cultivated land cover changes in mountainous terrain. *Remote Sensing Letters*, 10(6), 573-582.
- [6] Shao, Z., Li, C., Li, D., Altan, O., Zhang, L., & Ding, L. (2020). An accurate matching method for projecting vector data into surveillance video to monitor and protect cultivated land. *ISPRS International Journal of Geo-Information*, 9(7), 448.
- [7] Sun, J., Shen, Z., Wang, Y., Bao, H., & Zhou, X. (2021). LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8922-8931).
- [8] Huang, X., Wan, X., & Peng, D. (2020). Robust feature matching with spatial smoothness constraints. *Remote Sensing*, 12(19), 3158.
- [9] Bojanić, D., Bartol, K., Pribanić, T., Petković, T., Donoso, Y. D., & Mas, J. S. (2019, September). On the comparison of classic and deep keypoint detector and descriptor methods. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp. 64-69). IEEE.
- [10] Karami, E., Prasad, S., & Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*.
- [11] Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. In *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11* (pp. 778-792). Springer Berlin Heidelberg.
- [12] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision* (pp. 2564-2571). IEEE.
- [13] Cai, L., Ye, Y., Gao, X., Li, Z., & Zhang, C. (2021). An improved visual SLAM based on affine transformation for ORB feature extraction. *Optik*, 227, 165421.
- [14] Alcantarilla, P. F., & Solutions, T. (2011). Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell.*, 34(7), 1281-1298.
- [15] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). BRISK: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision* (pp. 2548-2555). IEEE.
- [16] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012, June). Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 510-517). IEEE.
- [17] Mohammad, A., Saleh, O., & Abdeen, R. A. (2006). Occurrences algorithm for string searching based on brute-force algorithm. *Journal of Computer Science*, 2(1), 82-85.
- [18] Pandey, A., & Jain, A. (2017). Comparative



- analysis of KNN algorithm using various normalization techniques. *International Journal of Computer Network and Information Security*, 11(11), 36.
- [19] Kuang, Q., & Zhao, L. (2009). A practical GPU based kNN algorithm. In *Proceedings. The 2009 International Symposium on Computer Science and Computational Technology (ISCSCI 2009)* (p. 151). Academy Publisher.
- [20] Muja, M., & Lowe, D. (2009). Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 5, 6.
- [21] Megalingam, R. K., Sriteja, G., Kashyap, A., Apuroop, K. G. S., Gedala, V. V., & Badhyopadhyay, S. (2018). Performance Evaluation of SIFT & FLANN and HAAR Cascade Image Processing Algorithms for Object Identification in Robotic Applications. *International Journal of Pure and Applied Mathematics*, 118(18), 2605-2612.
- [22] George, J., & Raj, S. G. (2021). Leaf Identification using Harris Corner Detection, SURF Feature and FLANN Matcher. *Int. J. Innov. Technol. Explor. Eng.*, 8(8).
- [23] Wang, S., Guo, Z., & Liu, Y. (2021, September). An image matching method based on sift feature extraction and FLANN search algorithm improvement. In *Journal of Physics: Conference Series* (Vol. 2037, No. 1, p. 012122). IOP Publishing.
- [24] Bolles, R. C., & Fischler, M. A. (1981, August). A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI* (Vol. 1981, pp. 637-643).
- [25] Liu, J., & Bu, F. (2019). Improved RANSAC features image-matching method based on SURF. *The Journal of Engineering*, 2019(23), 9118-9122.
- [26] Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. *Image Rochester NY*, 4(1), 2-3.
- [27] Parihar, U. S., Gujarathi, A., Mehta, K., Tourani, S., Garg, S., Milford, M., & Krishna, K. M. (2021, September). RoRD: Rotation-robust descriptors and orthographic views for local feature matching. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1593-1600). IEEE.
- [28] Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., & Sattler, T. (2019). 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* (pp. 8092-8101).
- [29] Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., & Humenberger, M. (2019). R2D2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*.
- [30] Noh, H., Araujo, A., Sim, J., Weyand, T., & Han, B. (2017). Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3456-3465).
- [31] Tareen, S. A. K., & Saleem, Z. (2018, March). A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-10). IEEE.
- [32] Bayraktar, E., & Boyraz, P. (2017). Analysis of feature detector and descriptor combinations with a localization experiment for various performance metrics. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(3), 2444-2454.
- [33] Ha, Y. S., Lee, J., & Kim, Y. T. (2022). Performance Evaluation of Feature Matching Techniques for Detecting Reinforced Soil Retaining Wall Displacement. *Remote Sensing*, 14(7), 1697.
- [34] Efe, U., Ince, K. G., & Alatan, A. (2021). Dfm: A performance baseline for deep feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4284-4293).