ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Comparing Feature Matching Methods to Identify Persian Writers

R. Javadzadeh[1,*], E. Zahmati Iraj[1], H. R. Ghaffary[2]

[1] Department of Computer Science, Ferdows Branch, Islamic Azad University Ferdows, Iran
[2] Assistant Professor, Department of Computer Engineering, Faculty of Engineering, Islamic Azad University Ferdows Branch, Iran

ARTICLE INFO

Article History:
Received 4 Novomer 2022
Received in revised form
14 January 2023
Accepted 7 March 2023
Available online 7 March 2023

Keywords:
Writer Identification, Feature Matching, SIFT Algorithm, SURF Algorithm, FREAK Algorithm, Harris Method.

ABSTRACT

This study explores a comprehensive set of feature matching techniques to address the challenge of writer identification in Persian handwritten scripts. Writer identification, a key task in the domain of document analysis and forensic handwriting verification, has seen increasing use of local feature descriptors due to their robustness to scale, rotation, and noise. Although the literature highlights the potential of such techniques, limited comparative research has been conducted specifically for Persian script. In this work, we implement and evaluate several well-known feature matching algorithms including SIFT, SURF, BRISK, FREAK, and Harris corner detector-based hybrids such as Harris-SURF, Harris-FREAK, and Harris-BRISK as well as combinations like BRISK-SURF, SURF-FREAK, and SURF-BRISK. The writer identification process is carried out by comparing the feature points in a query image against those in a set of reference images. The reference image that exhibits the highest number of correctly matched keypoints is identified as belonging to the same writer as the query sample. Our experimental findings reveal that among the evaluated algorithms, the SIFT and SURF methods outperform others in terms of accuracy and reliability in identifying Persian writers. Nevertheless, several hybrid approaches also produce promising results, suggesting that combining feature detectors and descriptors can offer valuable performance improvements. This study provides a foundation for future research and applications in Persian handwriting analysis and biometric authentication.

1. INTRODUCTION

In the realm of Persian/Arabic writer identification and recognition, several innovative approaches have been proposed, each leveraging advanced neural network architectures. Aliakbarzadeh and Razzazi (2020) introduce a method utilizing Random Strokes representations and Gated Recurrent Unit neural networks, achieving an impressive 100% accuracy on 10 writers and 76% accuracy on 50 writers. Their work, presented in the Majlesi Journal of Electrical Engineering, showcases the effectiveness of their approach in online writer identification [1].

* corresponding author: st.javadzade@ferdowsiau.ac.ir
Department of Computer Science, Ferdows Branch, Islamic Azad University Ferdows, Iran

http://dx.doi.org/10.47176/TMI.2023.16

Similarly, Khosravi and Chalechale (2022) present a recognition approach for Persian/Arabic handwritten words, employing convolutional neural networks and autoencoders. Their method attains a recognition accuracy of 91.09%. This study, featured in Mathematical Problems in Engineering, demonstrates the potential of combining these neural network components for accurate word recognition [2].

Furthermore, Safarzadeh and Jafarzadeh (2020) contribute to offline Persian handwriting recognition by combining deep convolutional neural networks (CNN) and recurrent neural networks (RNN) with a connectionist temporal classification (CTC) loss function. Presented at the 25th International Computer Conference, their work illustrates a robust recognizer for Persian handwritten words. These diverse approaches collectively highlight the advancements in Persian/Arabic writer identification and word recognition achieved through neural network technologies [3].

Writer identification is a biometric identification task that can be utilized to solve numerous problems, such as writer identification in forensic analysis [4] and the analysis of historical archives [5]. In text-dependent writer identification, a particular text is used, while the content of the text may vary in text-independent methods. Writer identification can be categorized into two offline identifications, where scanned images are used for identification, and online identification, where additional information such as pen movements is obtained at the time of writing. In this study, we utilize an offline text-dependent method for writer identification.

Various methods have been employed to solve this classification problem, with some researchers adopting texture recognition techniques. Text: These methods extract textural features from images and use them for classification. For instance, a study by [6] found that Gabor filtering and gray level co-occurrence matrices were successful at identifying writers. Additionally, recent research [7] has demonstrated that using texture features like histograms of Local Binary Patterns, Local Ternary Patterns, and Local Phase Quantization could enhance identification accuracy. Texture recognition methods have demonstrated potential in identifying writers offline, regardless of the text they produce.

Feature matching methods can be employed to locate identical objects in different scenes by detecting and matching feature points. Feature points are identifiable points on an object that remain consistent despite transformations in different scenes. Corresponding feature descriptors or feature vectors are then computed for each point. These descriptors are used to detect and match identical feature points in multiple scenes. Recently, this method was employed for the identification of writers, showing promising results. The identification method detects significant features in the text by extracting descriptors that identify similar patterns in different texts. Subjective evaluations are excluded unless unmistakably marked as such. The sources are most alike when written by the same author, based on the number of similar feature patterns. Hedging is used to make positions clear.

This paper applies SIFT, SURF, and Harris feature detectors and various descriptors to compare and determine the optimal detector-descriptor combination. The language is formal and objective, using value-neutral language, consistent technical terms, passive tone, and impersonal construction. The structure employs clear, concise, and necessary information in simple sentences, adhering to the conventional academic structure with regular author and institution formatting. Finally, the text adheres to grammatical correctness, uses precise word choice, and avoids filler words to maintain a clear and balanced language. Section II explores current applications of feature matching techniques with an emphasis on writer identification. The paper follows a conventional format, presenting all implemented approaches in Section III, while Section IV provides experimental results and comparisons. Language biases and emotional phrasing are steered clear of, and technical terms are defined when appropriate. Moreover, consistent citation and conventional academic sections are utilized throughout. The paper concludes with Section V, providing a summary of the findings. The structure adheres to conventional academic sections with appropriate formatting features. It is pertinent to state that the language used in the paper is clear and objective, without any subjective evaluations. Technical term abbreviations have been explained wherever used. The language used is neutral, avoiding biased, emotional, figurative, or ornamental language. The vocabulary used is subject-specific, and grammar and punctuation correctness have been ensured throughout the text. The writing style is formal, and the information is presented logically with causal connections established between the statements. Passive tone and impersonal construction have been employed, and first-person perspectives have been avoided, except where necessary.

2. RELATED WORKS

In a study on Persian writer identification by [8], 40 writers were instructed to write a specific Persian text twice on separate papers. The SIFT method was utilized for writer identification, computing the number of feature descriptors matched between a query script and all reference scripts. As a result, the writer of the reference script with the highest number of matched feature points was identified as the corresponding writer of the query script. This system has reported a 100% accuracy rate for writer identification.

In a separate study by [9], the process began with separating all words from the page followed by applying the SIFT algorithm to extract descriptors from each key point. Feature descriptors were then grouped into 300 clusters to form a feature vector known as SDS. Information on the scale and orientation of each key point was utilized to create another vector referred to as the SOH features vector. Thus, writer identification can be achieved through computation of the difference between the weighted sum of SDS and SOH feature vectors across scripts. This approach has yielded accuracy levels of 95.4% on the Chinese scripts of Hit-Mw dataset [10] and 92.4% on the Firemaker dataset [11]. Results on the multilingual ICDAR2011 dataset [12] show accuracy of 96.2% for Greek scripts and 100% for English, French, and German scripts. Additionally, the IAM Handwritten Dataset [13] employs two scripts from each writer for writer identification, resulting in a reported accuracy of 98.5%.

Another study, resembling the one proposed by [9], involves segmenting all words from the page and applying the SURF algorithm to extract descriptors from each key point [14]. Consequently, SDS and SOH feature vectors are extracted and used for identifying the writer. The method achieved a reported accuracy of 95.4% on both Hit-Mw and Firemaker datasets. It was also tested on the multilingual ICDAR2011 dataset. An accuracy rate of 80.8% was achieved for English scripts, 98.1% for Greek scripts, 88.5% for French scripts, and 92.3% for German scripts. The IAM handwritten dataset incorporated two scripts from each of the writers for identification, resulting in a 99.5% accuracy rate. This method has reportedly performed better on the Firemaker dataset, modified IAM dataset, and Greek scripts of the ICDAR2011 dataset compared to other datasets. Harris corner detection was introduced as an enhancement of the Moravec corner detection algorithm [15].

This technique computes corner points in an image using a local shifting mask, successfully detecting robust interest points that were later employed in object classification [16]. Additionally, Harris corner detection has been effectively utilized to detect crucial interest points in the face, namely corner points around the eyes and nose [17]. The FAST corner detection method was introduced as a speedy alternative approach for detecting corner points that is suitable for real-time applications [18]. Essentially, this method detects corners by comparing a pixel value with its 16 adjacent pixels. The FAST method is utilized as an interest point detector in the development of image matching algorithms, such as FREAK and BRISK, where rapid interest point detection leads to fast image matching [19-20].

This is crucial for systems that require agile methods with reduced computations. Keyword spotting feature extraction can be achieved via word segmentation, then extracting arbitrary features like width to height ratio, word area density, center of gravity, vertical projection, top-bottom shape projections, upper grid features, and down grid features from each word [21]. Although feature extraction can be accomplished using image matching feature descriptors such as SIFT, as done in [22]; recently, a combination of detectors and descriptors, including Harris-FREAK, SURF-FREAK, and SURF-BRISK methods, have been used for object detection [23].

Although feature extraction can be accomplished using image matching feature descriptors such as SIFT, as done in [22]; recently, a combination of detectors and descriptors, including Harris-FREAK, SURF-FREAK, and SURF-BRISK methods, have been used for object detection [23]. However, it is unclear if these methods' performance has been tested for Persian writer identification. The Harris method and multi-scale FAST method were used to detect interest points in scripts for writer identification in this study. Additionally, several renowned feature matching methods like SIFT, SURF, FREAK, and BRISK were utilized.

3. MATERIALS AND METHODS

3.1. Preprocessing Step

In this step, all images are binarized using Otsu thresholding algorithm [24]. For this method, has found the best thresholding value for the scripts. In addition, as is shown in Figure 1, writings are minimized to single connected pixels by using thinning morphological operations [25].

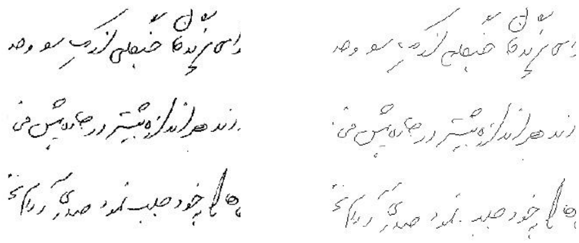

Fig. 1. Sample binary script (left) and Sample binary script after thinning morphological operations (Right)

3.2. Scale invariant feature transform (SIFT)

Lowe (2004) proposed the Scale-Invariant Feature Transform (SIFT) algorithm to detect distinctive and local feature points in images [26]. The algorithm convolves a query image with various Gaussian kernels, which identifies candidate feature points by computing the difference of two adjacent Gaussian images (forming DOG space). The gradient magnitude and orientation are computed to determine these feature points' orientation. Feature points are first rotated to maintain invariance with regard to rotation. Then, a square region is formed around each feature point, which depends on the detection scale, and rotated based on the previous step's computed orientation. Then, a square region is formed around each feature point, which depends on the detection scale, and rotated based on the previous step's computed orientation. Afterward, this region is split into 4 × 4 subregions, and the gradient orientation of each pixel in each subregion is calculated. Computed orientations for each subregion are divided into eight bins, corresponding to eight directions. By accumulating the gradient information of each subregion, a 128-dimensional feature vector is extracted for each feature point. Figure 2 provides an example of three feature points. Feature point detection is generally performed in four octaves, with each octave reducing the size of the Gaussian image by half. This ensures the detection of all extrema points, which serve as potential feature points.

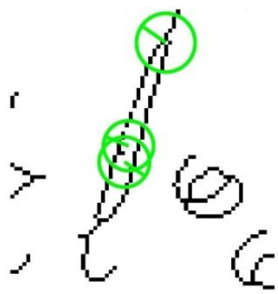

Fig. 2. A Number of SIFT Feature Points

3.3. Speeded up robust features (SURF)

This algorithm was presented as a faster alternative to SIFT method [27]. The concept of integral image is used for faster computation of convolving box filters. The determinant of Hessian detector (in multiple scales) is computed to detect blob-like structures which are the potential feature points. A Gaussian kernel of size $\sigma=1.2$ is approximated using a 9 × 9 box filter which is denoted as $s = 9$. Note that Convolving box filters are essentially different Harr wavelet filters which approximate the result of convolving Gaussian kernels. Thus, Hessian detector is computed in different scales.

Hessian detector is computed in multiple octaves to detect the maximum number of potential feature points. Next, A reproducible orientation for each feature point is identified. A square region of size 20s is constructed around the feature point and is rotated according to its orientation. This region is then divided into 4 × 4 smaller sub-regions. Then the sum of the Harr wavelet responses in horizontal and vertical directions in conjunction with the sum of the absolute values of responses in horizontal and vertical directions is computed for each sub-region. Hence, a 4-dimensional feature vector is extracted for each sub-region. Thus, by accumulating the feature vector of each sub-region, 4 × 4 × 4 = 64-dimensional feature descriptor for each feature point is extracted. Figure 3 shows an example of five located feature points located in different scales.

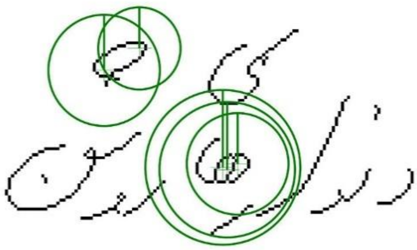

Fig. 3. Five SURF Feature Points

3.4. Binary robust invariant scalable keypoints (BRISK)

Feature points are detected using the FAST detector in scale-space concept [20]. The Scale-space is composed of octaves and intra-octaves, either of which has a different down-sampling rate. BRISK descriptor shown in Figure 4 is composed of multiple disks centered around a sampling feature point. The points are then smoothed using different Gaussian kernels over a neighborhood of feature points. The orientation of a feature point is obtained by computing the local gradients of long distance pairs. Therefore, intensity values of rotated short distance pairs are computed to obtain a BRISK descriptor. Hamming distance used to compute the difference of two feature descriptors for feature matching. A 512-bit long binary descriptor is extracted for each feature point. Figure 4 shows an example of multiple feature points detected in one scale.

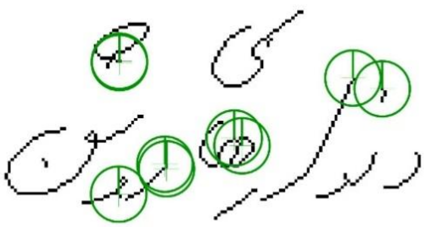

Fig. 4. Examples of both BRISK and FREAK feature points

3.5. Fast retina keypoint (FREAK)

The FREAK descriptor is based on the human visual system [19]. Like the brisk method, the FAST detector is employed to identify feature points in scale-space. It is worth noting that the same feature points displayed in Figure 4 are also detected in the FREAK method because both methods use the same detector and set of parameters. Figure 5 showcases the FREAK descriptor of a feature point, where the size of disks differs according to its Gaussian kernel size. Each disk represents the computed standard deviation of Gaussian kernels. Binary descriptors are constructed in this method by computing the difference of Gaussians between pairs of receptive fields (disks). The orientation of feature points is obtained in a similar fashion to the BRISK method. However, in the FREAK method, pairs of symmetric receptive fields are selected and the local gradients over pairs are summed to estimate the orientation. A FREAK descriptor consists of 512 bits.

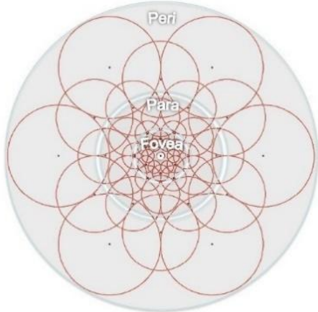

Fig .5. FREAK Sampling Pattern [19]

3.6. Combined Methods

In this section, we present different methods which are used for writer identification. The presented methods are as follows: SIFT, SURF, SURF-FREAK, SURF-BRISK, BRISK, BRISK-SURF, FREAK, Harris-SURF, Harris-FREAK and Harris-Brisk. Each method comprised of two parts, 1) the detector's name which denotes the algorithm used for feature detection and 2) the descriptor's name which denotes the algorithm used for feature extraction of corresponding points. It should be noted that while in the works proposed by [9], [14] word segmentation is done before using SURF and SIFT methods (respectively), In this study, no word segmentation is done since it is irrelevant to the subject of comparing image matching methods and moreover, it must be discussed that which word segmentation method must be chosen. Thus, it is a different topic for discussion and review, which is not covered in this study.

4. EXPERIMENTAL RESULTS

In this section, the performance of all the methods is described and compared. In Part 4.1, the collected dataset is described. The implementation details are presented in Part 4.2, and the performance evaluation of all methods is presented in Part 4.3.

4.1. Dataset

Due to lack of public dataset for Persian writer identification, 56 scripts from 28 writers had been collected [28]. It is asked from each writer, to write a specific sample text, twice, on two different pages. All scripts are written on A5 paper and then scanned at 300 dpi, and resized to 874 × 1240.

4.2. Implementations

These experiments have been performed in the MATLAB R2017a on an AMD Phenom Tripe-core at 2.40 GHz with 4GB of RAM. SIFT features were computed using the VL-SIFT library version 0.9.19 [29]. All others were implemented using built-in MATLAB functions. The optimum parameter values of all methods are determined experimentally using test-and-error method.

1. SIFT
   - Number of octaves: 3
   - Starting octave: -1

     Which means starting from the doubled size of the query image rather than the original size
   - Number of scales per octave: 2

- Starting scale of first octave: 0.8

2. SURF
   - Number of octaves: 2
   - Number of scales: 5
   - Threshold: 600

     More threshold means fewer points will be found
   - Upright: True
   - Orientation of all features is set to 90 degrees
3. SURF-FREAK
   - Number of octaves: 2
   - Number of scales: 4
   - Threshold: 700
   - Upright: True
4. SURF-BRISK
   - Number of octaves: 2
   - Number of scales: 5
   - Threshold: 700
   - Upright: True
5. BRISK
   - Octaves: (0, 1, 2)
   - Minimum Contrast: 0.1

     Accepted minimum difference of gray level value between a corner and its surrounding region, increasing this value means fewer points will be found
   - Minimum Quality: 0.1

     Minimum accepted quality of a corner point
   - Upright: True
6. FREAK
   - Octaves: (0, 1)
   - Minimum Contrast: 0.1
   - Minimum Quality: 0.1
   - Upright: True
7. Harris-SURF
   - Minimum Quality: 0.1

     Larger values remove weak corners
   - Filter Size: 3

Gaussian Smoothing with the filter dimensions of FilterSize-by-FilterSize and the standard deviation of (Filer Size) / 3

- Upright: True

8. Harris-FREAK and Harris-BRISK Methods
    - Minimum Quality: 0.01
    - Filter Size: 3
    - Upright: True

4.3. Performance Evaluation

First a query image is matched versus all images in the dataset, and the reference image which has the most number of matched feature points corresponds to the query image. Given that there are two scripts per writer, Therefore, the identity of each writer can be identified. This is done for all images in the dataset.

Performance evaluation of all methods are shown in Table 1. And Figure 6. Although, it must be noted that in rare cases, it is possible for a test script to match with more than one script due to the same number of matched feature points between the test script and a number of other scripts. Hence, more than one script may match with the test image (note that there are only two scripts per writer). Therefore, in these cases, the result is considered a failed identification, and since it is an identification error here is how it has been dealt with:

For example, if a test script of class (1) is identified and matched with scripts of classes = {1, 2, 3} at the same time, first the script would be discarded and classified as NaN (Not a Number). As a result, classification rate (CR) of the method decreases and Correct Recognition rate (CRR) would not consider these NaN cases in its computation. The Performance of Methods with all methods is shown in Figure 6. In Table 1 #N is the number of NaN outputs. The details of methods with less than 100% classified rate is shown in Table 1.

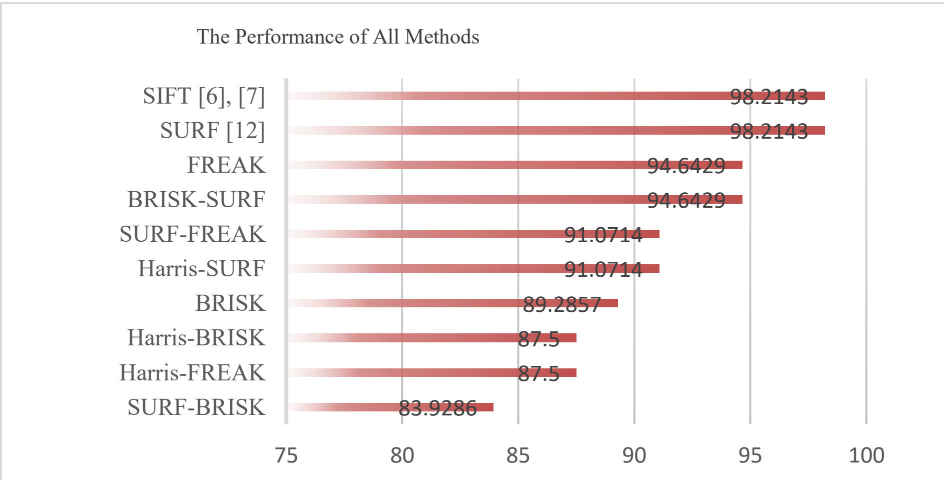

Fig. 6. Percentage of The Correct Recognition of all Methods

The findings illustrated in Figure 6 demonstrate that the SURF and SIFT methods outperformed all other methods, with each only encountering a single misclassification instance in class (24), as evidenced in Figure 7. However, in the case of the SIFT method, this misclassification was due to multiple matches for a single script. The Harris method also yielded encouraging outcomes, particularly when combined with SURF, FREAK, and BRISK descriptors. The SURF descriptor has been identified as the best option when using the Harris operator, while the other two descriptors performed similarly. Furthermore, both the SURF-FREAK and SURF-BRISK methods utilize the SURF detector; however, it has been demonstrated that the SURF descriptor outperforms the other two. Results from Table 1 and Figure 6 indicate that the SIFT and SURF methods have comparable results and outperform other methods. The FREAK method yielded superior results compared to the BRISK method, and BRISK-SURF is a more effective combination than SURF-BRISK. Both the FREAK and BRISK-SURF methods have performed equally well after SIFT and SURF.

The results depicted in Figure 7 indicate that all methods encountered difficulty identifying writer (24) at least once, and class (14) proved to be a challenging script for writer identification in most instances. It should be noted that all methods were successful in classifying the classes that are not displayed in Figure 7.

Table 1. The Performance Details of all Methods with Less Than 100% Classified Rate

Methods	CR (%)	#N	CRR by Excluding NaN Outputs (%)	CRR by Including NaN Outputs (%)
SIFT [8-9]	98.2143	1	100	98.2143
BRISK-FREAK (FREAK)	98.2143	1	96.3636	94.6429
BRISK-SURF	98.2143	1	96.3636	94.6429
SURF-BRISK	92.8571	4	90.3846	83.9286

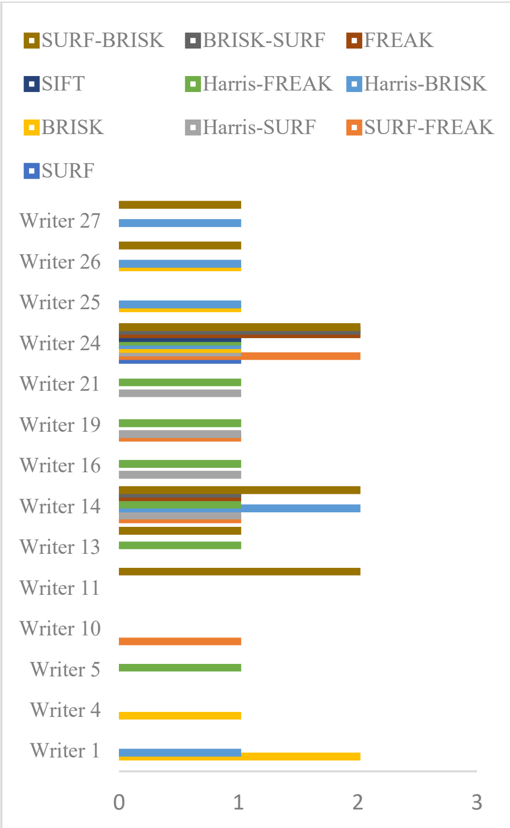

Fig. 7. Error Distribution of Methods by Class (*#Errors Per Writer*)

Figure 8 illustrates a comparison of all implemented methods in terms of time complexity. It shows how long it takes to identify the writer for one query text between all the other scripts in the dataset. Run time of all implemented methods is computed using “Run and Time” feature of MATLAB.

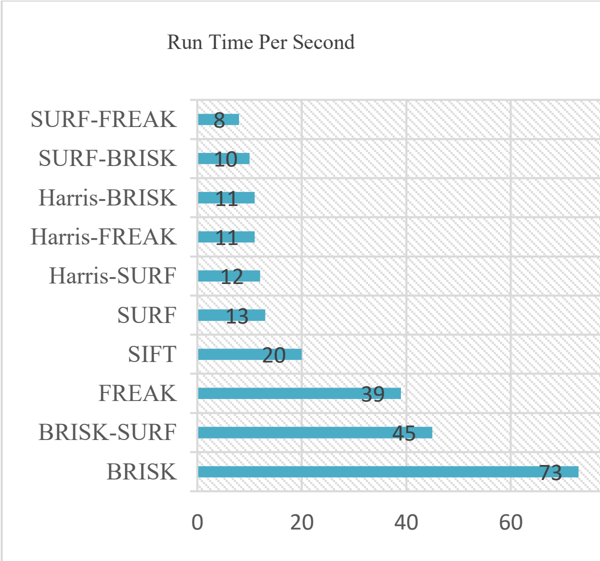

Fig. 8. Run Time of All Methods

SURF-FREAK was the fastest method, with similar running times for SURF-BRISK, Harris-BRISK, Harris-FREAK, Harris-SURF, and SURF methods. However, SIFT method had a slower running time than the other methods, including FREAK, BRISK-SURF, and BRISK. This inefficiency is a result of a high number of feature points that require the computation and matching of their descriptors with many other feature descriptors. Increasing the threshold and, consequently, reducing the number of detected feature points led to a decline in the performance of these methods.

As noted, although SIFT and SURF algorithms have demonstrated promising outcomes in recent studies of English and Persian datasets [8-9], [14], other techniques have also performed reasonably well. To the best of the authors' knowledge, no studies have compared these methods to one another, particularly in relation to more recent image matching techniques in the field of identifying Persian writers. Furthermore, the authors are not aware of any studies utilizing image matching techniques other than SIFT and SURF methods for identifying Persian writers, nor have any studies been conducted on the combination of image matching methods for this purpose.

5. CONCLUSION

Two recently proposed image matching methods for writer identification are reviewed and implemented in this paper. Consistency was maintained through conventional formatting and grammatical correctness. Additionally, several combined image matching methods are proposed specifically for Persian writer identification. The performance of all implemented methods is compared, and the top-performing methods are determined. Bias was avoided throughout the article by utilizing clear, objective, and value-neutral language. The SURF method achieved the highest recognition rate of 98% with a shorter run time than the SIFT method, which performed similarly accurately in second place. FREAK and BRISK-SURF achieve equal recognition rates of 94%, however, their high run time is a drawback. Similarly, SURF-FREAK and Harris-SURF perform equally well, achieving a 91% recognition rate. The BRISK method shows promise, yet its run time is the highest among all methods.

Experimental results indicate that SIFT and SURF techniques are optimal for identifying Persian writers, although the running time of both methods requires improvement. Additionally, all methods achieved success in feature detection by extracting feature points, as well as feature description by obtaining feature vectors for the feature points.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Aliakbarzadeh, M., & Razzazi, F. (2020). Online Persian/Arabic Writer Identification using Gated Recurrent Unit Neural Networks. Majlesi Journal of Electrical Engineering, 14, 73-79. https://doi.org/10.29252/MJEE.14.3.9.

[2] Khosravi, S., & Chalechale, A. (2022). Recognition of Persian/Arabic Handwritten Words Using a Combination of Convolutional Neural Networks and Autoencoder (AECNN). Mathematical Problems in Engineering. https://doi.org/10.1155/2022/4241016.

[3] Safarzadeh, V., & Jafarzadeh, P. (2020). Offline Persian Handwriting Recognition with CNN and RNN-CTC. 2020 25th International Computer Conference, Computer Society of Iran (CSICC), 1-10. https://doi.org/10.1109/CSICC49403.2020.9050073.

[4] Kumar, R., Chanda, B., & Sharma, J. D. (2014). A novel sparse model based forensic writer identification. Pattern Recognition Letters, 35, 105–112. doi:10.1016/j.patrec.2013.07.001.

[5] Schomaker, L., Franke, K., & Bulacu, M. (2007). Using codebooks of fragmented connected-component contours in forensic and historic writer identification. Pattern Recognition Letters, 28(6), 719–727. doi:10.1016/j.patrec.2006.08.005.

[6] Said, H. E., Tan, T. N., & Baker, K. D. (2000). Personal identification based on handwriting. Pattern Recognition, 33(1), 149-160.

[7] Hannad, Y., Siddiqi, I., & El Kettani, M. E. Y. (2016). Writer identification using texture descriptors of handwritten fragments. Expert Systems with Applications, 47, 14–22. doi:10.1016/j.eswa.2015.11.002.

[8] Peyvandi, M., & Zahedi, M. (2013). 11th Iranian Conference on Intelligent Systems. (In Persian).

[9] Wu, X., Tang, Y., & Bu, W. (2014). Offline text-independent writer identification based on scale invariant feature transform. IEEE Transactions on Information Forensics and Security, 9(3), 526–536. doi:10.1109/tifs.2014.2301274.

[10] Su, T., Zhang, T., & Guan, D. (2007). Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text. International Journal on Document Analysis and Recognition, 10(1), 27–38. doi:10.1007/s10032-006-0037-6.

[11] Bulacu, M., Schomaker, L., & Vuurpijl, L. (2005). Writer identification using edge-based directional features. Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. Presented at the Seventh International Conference on Document Analysis and Recognition, Edinburgh, UK. doi:10.1109/icdar.2003.1227797.

[12] Hassaine, A., & Maadeed, S. A. (2012). ICFHR 2012 competition on writer identification challenge 2: Arabic scripts. 2012 International Conference on Frontiers in Handwriting Recognition. Presented at the 2012 International Conference on Frontiers in Handwriting Recognition (ICFHR), Bari, Italy. doi:10.1109/icfhr.2012.218.

[13] Marti, U.-V., & Bunke, H. (2002). The IAM-database: an English sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition, 5(1), 39–46. doi:10.1007/s100320200071.

[14] Sharma, M. K., & Dhaka, V. P. (2015). Offline language-free writer identification based on speeded-up robust features. International Journal of Engineering-Transactions A: Basics, 28(7), 984–994.

[15] Harris, C., & Stephens, M. (1988). A combined corner and edge detector. Procedings of the Alvey Vision

Conference 1988. Presented at the Alvey Vision Conference 1988, Manchester. doi:10.5244/c.2.23.

[16] Ben-Musa, A. S., Singh, S. K., & Agrawal, P. (2014, July). Object detection and recognition in cluttered scene using Harris Corner Detection. 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT). India. doi:10.1109/iccicct.2014.6992953.

[17] Bhowmik, M. K., Shil, S., & Saha, P. (2013). Feature points extraction of thermal face using Harris interest point detection. Procedia Technology, 10, 724–730. doi:10.1016/j.protcy.2013.12.415.

[18] Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In Lecture Notes in Computer Science. Computer Vision – ECCV 2006 (pp. 430–443). doi:10.1007/11744023_34.

[19] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). FREAK: Fast Retina Keypoint. 2012 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI. doi:10.1109/cvpr.2012.6247715.

[20] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. 2011 International Conference on Computer Vision. Presented at the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain. doi:10.1109/iccv.2011.6126542.

[21] Keyvanpour, M., Tavoli, R., & Mozafari, S. (2014). Document image retrieval based on keyword spotting using relevance feedback.

[22] Rusinol, M., Aldavert, D., Toledo, R., & Llados, J. (2011). Browsing heterogeneous document collections by a segmentation-free word spotting method. 2011 International Conference on Document Analysis and Recognition. Beijing, China. doi:10.1109/icdar.2011.22.

[23] Madbouly, A. M. M., Wafy, M., & Mostafa, M. S. M. (2015). Performance Assessment of Feature Detector-Descriptor Combination. International Journal of Computer Science Issues (IJCSI), 12(5).

[24] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

[25] Lam, L., Lee, S.-W., & Suen, C. Y. (1992). Thinning methodologies-a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(9), 869–885. doi:10.1109/34.161346

[26] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110. doi:10.1023/b:visi.0000029664.99615.94.

[27] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer vision and image understanding, 110(3), 346-359.

[28] Javadzadeh, R. (2017). Persian Writer Identification Dataset. figshare.com.california.s/1362d5e928c349ae058a. https://figshare.com/

[29] Vedaldi, A., & Fulkerson, B. (2010, October). VLFeat: An open and portable library of computer vision algorithms. In Proceedings of the 18th ACM international conference on Multimedia (pp. 1469-1472). ACM.