❏   4095

Offline signature recognition system using oriented FAST and rotated BRIEF

Abdel-Karim Al-Tamimi[1], Asseel Qasaimeh[2], Kefaya Qaddoum[3]
[1,3]CIS Department, Higher Colleges of Technology (HCT), United Arab Emirates
[1,2]Computer Engineering Department, Yarmouk University, Jordan

Article Info	ABSTRACT
	Despite recent developments in offline signature recognition systems, there is however limited focus on the recognition problem facet of using an inadequate sample size for training that could deliver reliable and easy to use authentication systems. Signature recognition systems are one of the most popular biometric authentication systems. They are regarded as non-invasive, socially accepted, and adequately precise. Research on offline signature recognition systems still has not shown competent results when a limited number of signatures are used. This paper describes our proposed practical offline signature recognition system using the oriented FAST and rotated BRIEF (ORB) feature extraction algorithm. We focus on the practicality of the proposed system, which requires only the minimum number of signatures per user to achieve a high level of fidelity. We manifest the practicality of our approach with a signature database of 300 signatures from 100 different individuals, implying that only two signatures are needed per person to train the proposed system. Our proposed solution achieves a 91% recognition rate with a median matching time of only 7 ms.

Corresponding Author:

Abdel-Karim Al-Tamimi
Computer Engineering Department
Yarmouk University
Irbid, Jordan
Email: altamimi@yu.edu.jo

1. INTRODUCTION

Biometrics technologies are used for recognizing people based on their physiological traits, including fingerprints, or behavioral traits such as voice, and handwritten signature. Biometric systems can be used for both verification and identification tasks, which are vital for security applications. Although technology has developed, signing is still a common authentication method today. The handwritten signature is widely used in a variety of security systems for authentication, in some bureaucratic transactions, contracts, and to validate financial processes and elections. Signature verification aims to detect whether a given signature is genuine or forged. There are two methods of signature recognition: Online (dynamic) and offline (static). In the online method, an acquisition device such as an electronic tablet, a pressure-sensitive pen, or a glove-based system is needed to obtain the signature and capture its defining characteristics. In literature, there are several proposed offline signature verification systems that are based on texture description and interest point matching [1]-[4]. Offline systems lacks the access to the riach identification features that are obtainable using the mor invasive online systesm, and thus they need to rely solely on signature 2D images. Despite the advancements in this subject, researchers are still working towards producing a practical solution for the recognition problem of offline signatures, particularly for large-scale

data [4], [5]. Section 2 provides a throughout literature review of the currently used matching algorithms. Section 3 dicusses our research methodology, and section 4 demnostrates our findings and testing results. Section 5 discusses the impact of our research findings, and section 6 concludes and lists the main contributions of our paper.

2. LITRATURE REVIEW

In this section, we discuss the features of SIFT, SURF, and ORB 2D feature extraction and matching algorithms to help deduct the differences in performance among these algorithms.

2.1. Scale invariant feature transform (SIFT) descriptor

In 2004, SIFT [6], [7] was proposed by D. Lowe an invariant feature detector. SIFT uses a cascade filtering concept to detect the features and convert image data into scale-invariant features. SIFT detects local features which are robust against illumination changes, minor changes in viewpoint, and noise. In general, SIFT consist of four main stages: scale-space detection, key-points localization, orientation assignment, and extraction of the key-point descriptor.

In the scale-space detection stage, SIFT decomposes the original image using a Gaussian pyramid, which has multiple levels called octaves. Each octave is also decomposed into multiple sub-levels through convolving the original image with Gaussian filters with different scales. Each pixel DoG is compared with its eight neighbors; when the pixel has the maximum or the minimum value among all the eight neighbors' pixels, it is considered as a *key-point*. SIFT uses the quadratic Taylor expansion of the DoG scale-space function. Around the key-point, the direction and the magnitude of the gradient are calculated for each pixel and the orientation histogram is formed. Once this process is completed, the highest value is considered as the orientation of the key-point.

2.2. Speeded-up robust features (SURF) descriptor

In 2006, Herbert Bay *et al.* presented SURF [8] algorithm. This algorithm contains four main steps: interest point detection, location and scale-space representation of interest points, local neighborhood description, and key-points matching. To detect the interest points, as a first step, SURF uses square-shaped filters to compute Gaussian approximation after the image was already cropped and discretized. Then, the Hessian blob detector [9] is used, which computes the determinant of the Hessian matrix around each point. The point that gets the highest determinant is considered as an *interest* point. A determinant is also used to select the scale of the interest point. For a point $P(x, y)$ in image I, the Hessian matrix $H(P, \sigma)$ at point $P(x, y)$ and scale σ is defined as (1):

$$H(P,\sigma) = \begin{pmatrix} L_{xx}(P,\sigma) & L_{xy}(P,\sigma) \\ L_{xy}(P,\sigma) & L_{yy}(P,\sigma) \end{pmatrix} \tag{1}$$

where $L_{xx}(P, \sigma)$ is the x second-order derivative of the gray-scaled image, and since SURF uses the square-shaped filter, the expression of Hessian's determinant is simplified as (2):

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{2}$$

To compute the location and the scale-space representation of interest points, SURF applies different filter sizes to represent the scale-space representation, then the highest determinant of the Hessian matrix is added in the image space and scaled as Brown *et al.* proposed [9].

In order to identify the rotational invariance, the orientation of interest points is found. After SURF computes the Haar wavelet [8], we collect responses in the circular neighborhood around the interest point and weighting them by a Gaussian function. In order to evaluate the primary orientation, all responses are calculated within a sliding window of $\pi/3$ size, and the sliding window's size is chosen carefully to maintain a balance between angular resolution and robustness. SURF is widely used in image matching and recognition systems, including steganography [10], face liveness detection or face anti-spoofing [11].

2.3. Oriented FAST and rotated BRIEF (ORB) descriptor

In 2011, Rublee proposed oriented FAST and rotated BRIEF (ORB) that is built on FAST key-point detector and BRIEF descriptor. These two algorithms are attractive because of their superior performance and low time requirements [12], [13]. FAST detector [14], [15] is a technique that finds key-points in real-

time that match specific visual features [16]. FAST detector measures the intensity threshold between the center pixel and the pixels in the circular ring region around the center [17]. Because FAST detector does not measure key-points of the corner, ORB employs Harris corner measure [18], [19] to order the detected key-points. To detect N, the number of key-points, ORB set a low threshold to get more than N key-points in the first step, and then it uses Harris measure to order them, and select the top N points.

FAST detector does not produce multi-scale features. Alternatively, a scale pyramid of the image is employed, and FAST features are measured and filtered at each level. ORB measures corner orientation using the simple and effective technique: intensity centroid approach. This approach assumes the intensity of a corner is an offset of its center, and this vector can be used to assign an orientation. Rosin [19] computed the moments of a patch using (3), (4):

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \tag{3}$$

Then, the centroid will be:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{4}$$

$\left(\vec{OC} \right)$ the vector from the corner's center (O) to the centroid (C) can be constructed, and the orientation of this patch is (5):

$$\theta = a \tan 2(m_{01}, m_{10}) \tag{5}$$

where $(a\,tan\,2)$ is the quadrant-aware version of arc-tan. The moments are measured with x and y directions within the specified circular region (of radius equals to r) to improve the rotation invariance.

BRIEF descriptor [20] is a features descriptor, which uses straightforward binary tests between pixels in a smoothed image area. Binary descriptors have shorter computation time, smaller memory footprint, and higher efficiency in image comparisons when compared to vector-based features descriptors. Vector-based features detectors are based on the nearest-neighbor search, while binary features detectors are based on the priority search of multiple hierarchical clustering trees [21], [22].

ORB can match signature images using low-power devices without the use of GPU acceleration. Therefore, it performs as well as SIFT and better than SURF with almost two orders of magnitude [17]. Image patches are sets of binary intensity tests that BRIEF descriptor [20] makes a bit-string description of these patches. Afterward, Gaussian distribution is performed around the center of the image's patch. In ORB, in order to use BRIEF descriptor on the orientation of key-points, an efficient method is performed to steer BRIEF regarding the orientation. For n binary tests, a feature set at (x_i, y_i) can be represented as $2 \times n$ a matrix as (6):

$$S = \begin{pmatrix} x_1 \cdots x_n \\ y_1 \cdots y_n \end{pmatrix} \tag{6}$$

The steered version (S_θ) of (S) using the patch's orientation (θ) and the corresponding rotation matrix (R^θ) is calculated as (7):

$$S_\theta = R^\theta S \tag{7}$$

then the steered BRIEF operator is (8):

$$g_n(P, \theta) = f_n(P)|(x_i, y_i) \in S_\theta \tag{8}$$

While SURF and SIFT algorithms are based on histograms of gradients, ORB is a binary descriptor that is based on image intensity comparisons to encode patch's information as a binary string; which makes it relatively faster. ORB can match two images in a single instruction by using the hamming distance only.

3. METHOD

The poroposed ORB algorithm process is divided into several steps as depicted in Figure 1, more details about these steps are given in the following sub-sections. We start with acquiring offlines singatures from our users, then we proceed to apply several pre-processing steps to normalize the system inputs and remove any unnecessary data features. Afterwards, we apply several features extraction techniques, (SIFT, SURF, and ORB) to extract the signatures features, and then we perform the features matching comparison to evaluate the proposed system performance.
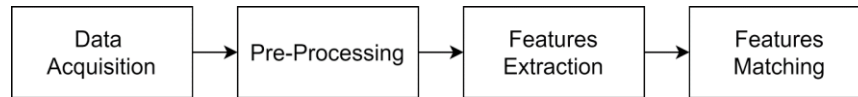

Figure 1. Proposed system flow chart

3.1. Data collection

In our data collection/acquisition phase, three handwritten signatures were scanned from Arabic users/signers. As this research approach aims to recognize users with a minimum number of required signatures. In contrast, related litrature research works base their findings on a relatively large number of signatures per signer as shown in Table 1.

Table 1. A comparison of the number of signatures required per person/signer

Related Works	Year	Number of required signatures (per signer)
Prasad et al. [1]	2013	7
Shikha et al. [2]	2013	20
Fazli et al. [3]	2015	14
Bhausaheb et al. [4]	2015	16
Our proposed work	2020	3

In our proposed system, users are asked to sign the same signature three times. Two of these signatures will be saved as a reference point, while the third signature will be used to test the system. Handwritten signatures were written/signed on white papers, and then scanned using a digital scanner using 600 dpi (dot per inch) resolution. We collected 300 signatures from 100 different individuals from students and faculty members of Yarmouk University. To preserve the integrity of the scanned images, they were saved using digital images in PNG format, as it is the best available format for binary images. Figure 2(a) to Figure 2(d) shows four examples of the required three signatures per user. As can be noted, signers used both Arabic and English characters when signing.
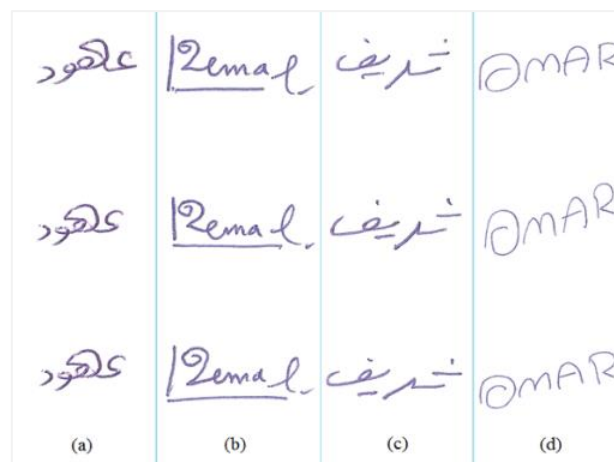

Figure 2. Examples of collected signatures from different users (a, b, c, d)

3.2. Data pre-processing

Once the signatures are converted to digital images; these images are pre-processed to eliminate undesired areas and impurities that would affect the system's performance [18]. Signature images would be cropped, resized, and filtered to make sure that all the collected signatures are pre-processed before the system extracts the required features. This step minimizes the number of false matches, maintains the high performance of the system, and reduces the processing time by reducing the image size [23]. The resulting images are resized to a unified size of (512*512 pixels). Finally, they are converted to black-and-white images, since ORB is a binary detector and descriptor. Figure 3 summarizes the signature pre-processing steps, and Figure 4(a) and Figure 4(b) shows the difference between the original signature image before and after applying the pre-processing steps. As can be noted in Figure 4, all processed images/singatures have been uniformly resized, color adjusted, and all unwanted features or noisy elements have been removed and cleaned.

Figure 3. Pre-processing steps applied to all acquired signatures

(a)   (b)

Figure 4. This figure are, (a) Original signature's image, (b) Pre-processed signature's image

3.3. Feature extraction

In the feature extraction stage, the system applies ORB, SURF, and SIFT algorithms to extract the signatures features and saves them in two byte-arrays: the serialized image as a byte array, and its associated features also as a byte array. These arrays are then stored into a custom database to facilitate storing and retrieving them. Figure 5 depicts the steps of the feature extraction stage.
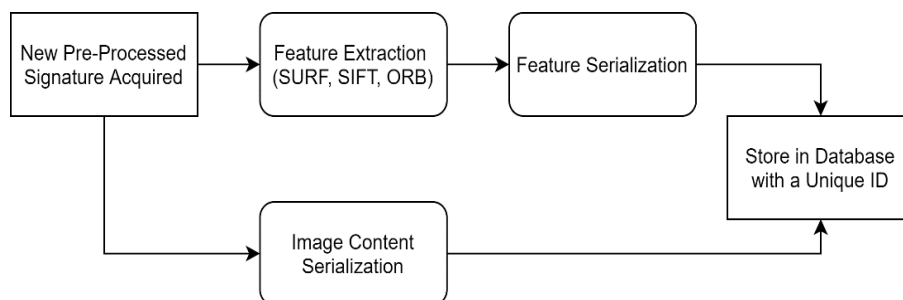

Figure 5. The process of features extraction process using SIFT, SURF, and ORB

3.4. Features matching

At this stage, the system aims to match the input signature with a stored template/reference signature. Each input signature is matched with every saved template and the number of matched features is computed. The template that achieves the highest numbers of matched features between all saved templates is retrieved as the closest one to resemble the matched template. Each time the system retrieves the correct template, the recognition ratio is increased. Feature's matching is performed using two matchers: Brute-Force matcher and fast library for approximate nearest neighbors (FLANN) matcher.

4. SYSTEM IMPLEMENTATION AND RESULTS

Our proposed system was developed using C# language and OpenCV 2.4.1. We have chosen a low-end PC to implement the proposed system to help demonstrate its efficiency. The used PC is an HP laptop equipped with AMD A4 processor clocked at 1.90 GHz and 6 GB installed memory. As shown in Figure 6, we have implemented a simple GUI interface to verify the accuracy of the implemented system and perform exploratory analyses on the system behavior during the detection and matching phases. The system matches features of the new signature (left-side picture-box) with features of each template saved in the system's database. The number of matched features is calculated each time to obtain the highest number of matched features. The highest matched template is displayed in the right-side picture-box.
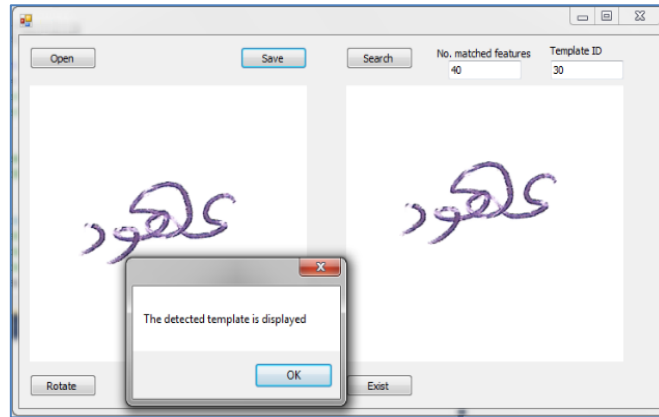

Figure 6. Simple GUI developed for verification purposes

The proposed system has been tested using 100 signatures, we measured the recognition ratio (RR), the speed of matching, the false acceptance rate (FAR), and the false rejection rate (FRR). These metrics are indicators of the system's accuracy and robustness. These performance metrics are calculated as (9):

$$FAR(\%) = \frac{\#False_{accepted}}{\#Tests_{All}} * 100\% \qquad (9)$$

where $\#False_{accepted}$ is the number of the false signatures (from an unauthorized user) that has been incorrectly accepted by the system, and $\#Tests_{All}$ is the number of all tested signatures. Lower FAR values are better as they indicated lower rates of false positive cases.

FRR is the ratio between the numbers of times when the system rejects signatures by an authorized user and does not retrieve their signature template compared to the number of all tested signatures.

$$FRR(\%) = \frac{\#Correct_{rejected}}{\#Tests_{All}} * 100\% \qquad (10)$$

where $\#Correct_{rejected}$ is the number of saved signatures (from authorized users) that were rejected by the system. Lower FRR values are also desirable as they indicated lower rates of false-negative cases. Recognition ratio (RR) is the ratio between the numbers of times the system retrieves the correct template (the same signer) and the number of all tests (the number of all used signatures in system testing). The following equation calculates recognition ratio (RR):

$$RR(\%) = \frac{\#Templates_{correct}}{\#Tests_{All}} * 100\% \qquad (11)$$

where $\#Templates_{correct}$ is the number of correct templates that were retrieved. Higher values of RR are desirable as they indicate the number of true positive cases achieved by the system.

The speed of matching is represented by the median matching time. Matching time is defined as the required time to complete a single match between the input signature and one template. Each signature has a

different number of extracted features; therefore, matching time may differ between one template and another. As a measure of central tendency for matching time values, we use the median of the matching times because of the skewed distribution of their values.

After performing the required preprocessing steps on our testing set of 100 signatures, we compared SURF, SIFT, and ORB algorithms using their FAR and FRR values. We have conducted our comparison using the Brute-Force matcher that will exhaustively try all possibilities in matching the two images. As can be noted from the results listed in Table 2, ORB has achieved the best results with the lowest ratio values for false positive (FAR) and false negatives (FRR) rates. Based on these experiments, the achieved RR for ORB is 91%.

Table 2. FAR and FRR comparison for ORB, SURF, and SIFT using brute-force matcher

Performance Metric/Algorithm	ORB	SURF	SIFT
FAR	26.6%	63%	90%
FRR	9%	60%	73%

As discussed before, FAR and FRR values impacts greatly RR ratio values. As shown in Table 3, ORB has outperformed both algorithms. In addition, it has a relatively short matching time especially when compared to SURF (7 ms vs 29 ms). While SIFT was the fastest (with 1 ms matching time), it has produced the worst matching ratios among the tested algorithms. These variations in matching times can be explained by relating them to the number of extracted features in each algorithm: using SIFT algorithm on average we extract 120 features, in SURF we extract between 200 to 3000 features, and in ORB we extract around 500 features on average.

Table 3. Recognition ratios and median matching times for ORB, SURF, and SIFT

Performance Metric/Algorithm	ORB	SURF	SIFT
Recognition ratio (RR%)	91%	40%	27%
Median Matching Time *(per template)*	7 ms	29 ms	1 ms

To assess the effects of the proposed pre-processing steps and the use of Brute-Force matcher on the obtained RR values, we repeated the testing experiment for ORB using the original images without applying preprocessing and using FLANN matcher instead of Brute-Force matcher. Fast library for approximate nearest neighbors (FLANN) matcher is much faster than Brute-Force matcher as it is designed to only find an approximate nearest neighbor match using clustering [24]-[26]. As depicted in Table 4, Brute-Force matcher outperformed FLANN significantly (91% to 64%). In addition, we can note the importance of applying the proposed pre-processing steps to improve the matching accuracy. Recognition ratio values decreased when the system used original (without pre-processing) signatures images, as original signature images may contain undesired features, e.g. dust particles or ink spots, that may lead to false matching and thus reduce the recognition ratio. Table 5 shows a comparison of the measured median matching times when using original versus pre-processed images, and when using Brute-Force matcher versus FLANN matcher.

Table 4. Comparison of recognition ratios of ORB using original versus pre-processing signatures, and brute-force versus FLANN matchers

Matcher	Pre-processed signature images (RR %)	Original signature images (RR %)
Brute-Force matcher	91 %	57 %
FLANN matcher	64 %	36 %

Table 5. Median matching time of ORB algorithm using original versus pre-processing signatures, and using brute-force versus FLANN matchers (per signature)

Matcher	Pre-processed signature images (median time)	Original signature images (median time)
Brute-Force	7 ms	7 ms
FLANN	3 ms	5 ms

As shown in Table 5, the median matching time decreased (matching speed increased) when using FLANN matcher as expected. However, when using FLANN the median matching time increased (matching

speed decreased) when using original signatures images without reprocessing. While these differences are relatively small, we would like to note that these values are the median matching time for a single matching or pairing process. In large datasets, FLANN is expected to significantly outperforms Brute-Force in terms of matching speed. Improving the matching speed under these experiment conditions is outside the scope of this paper. Table 6 summarizes the performed experiments using various experiment settings. As can be noted, ORB outperforms other algorithms especially when using Brute-Force matcher with 91% recognition rate and 7ms median matching time per signature.

Table 6. Full comparison results between ORB, SURF and SIFT

Input Images	Algorithm / Matcher	ORB		SURF		SIFT	
		RR (%)	Time (ms)	RR (%)	Time (ms)	RR (%)	Time (ms)
Pre-Processed	Brute-Force	**91%**	7 ms	40%	29 ms	27%	1 ms
	FLANN	64%	3%	34%	4.5 ms	23%	1 ms
Original	Brute-Force	57%	7 ms	23%	33 ms	22%	3 ms
	FLANN	36%	5 ms	16%	10 ms	16%	4 ms

5. DISCUSSION

The main goal of this paper is to show the effectiveness of ORB as a 2D feature extraction and matching algorithm in offline signature recognition. ORB as a rotation-invariant and scale-invariant matching algorithm is a very promising candidate for offline signature authentication systems. The usability of an authentication system is also crucial to determine its applicability in real-world scenarios. Requesting that users input identical or similar signatures upwards of 20 per user, as assumed in previous related works, is infeasible and undesirable. As shown in this paper, our proposed offline signature authentication system using ORB algorithm achieved, with limited pre-processing steps, a recognition rate (RR) of 91% based only on two signatures for training the system with a median processing time of 7ms per matching step. We also showed the importance of using the proper pre-processing steps and the effects of using Brute-Force matcher on the system's results.

6. CONCLUSION

Offline recognition systems are more accessible and more applicable in comparison with online signature systems as they do not need the presence of signers during the verification process, and they do not need any special tools like stylus and high precision acquiring systems. This research proposes the use of oriented FAST and rotated BRIEF (ORB) algorithm to detect and match signatures features for authentication purposes. The proposed system acquires signatures images and detect their 2D features after performing a minimal number of pre-processing steps. Once the system acquires a new signature, it matches the input signature's features with the features database to find the highest similarity among signatures and retrieve it.

As can be noted from our discussions, we designed our system without any special considerations to achieve lower FAR and FRR ratios. This allows for further possible improvements on the achieved RR ratio, especially when more pre-processing features are integrated. In general, we can observe that ORB algorithm achieved the best FAR and FRR ratios, which indicate that our system using the ORB algorithm is superior to the compared systems using either SURF or SIFT algorithms.

REFERENCES

[1]   R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195–216, May 2018.
[2]   T. Sari, L. S. Meslati, and M. Sellami, "Off-line handwritten arabic character segmentation algorithm: Acsa," in *Eighth International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 452–457.
[3]   L. M. Lorigo and V. Govindaraju, "Offline arabic handwriting recognition: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712–724, 2006.
[4]   N. Hamid, A. Yahya, R. B. Ahmad, and O. Al-Qershi, "Characteristic region based image steganography using speeded-up robust features technique," in *2012 International Conference on Future Communication Networks*, 2012, pp. 141–146.
[5]   R. Doroz, P. Porwik, T. Para, and K. Wrobel, "Dynamic signature recognition based on velocity changes of some features," *International Journal of Biometrics*, vol. 1, no. 1, pp. 47–14, 2008.

[6] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," *International Conference on Computer Vision (ICCV)*, 2001, pp. 525–531.

[7] R. M. Kumar and K. Sreekumar, "A survey on image feature descriptors," *Int J Comput Sci Inf Technol*, vol. 5, pp. 7668–7673, 2014.

[8] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Japan, 2007, pp. 225–234.

[9] C. Harris and M. Stephens, "A combined corner and edge detector," *Alvey Vision Conference*, 1988, pp. 147–151.

[10] F. Ferreira, G. Veruggio, M. Caccia, and G. Bruzzone, "ROV vision-based motion estimation: a comparison study," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 96–101, 2012.

[11] Z. Li, "The study of security application of LOGO recognition technology in sports video," *EURASIP Journal on Image and Video Processing*, vol. 1, 2019, Art. No. 46, doi: 10.1186/s13640-019-0441-8.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," *International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[13] R. Brunelli, "Template matching techniques in computer vision: theory and practice," Wiley, 2009.

[14] A. Prasad and V. Amaresh, "An offline signature verification system," *IEEE International Conference on Signal and Image Processing Applications*, 2009, pp. 95–100.

[15] P. Shikha and S. Shailja, "Neural network based offline signature recognition and verification system," *Research journal of engineering sciences*, vol. 2, no. 2, pp. 1-4, 2013.

[16] S. Fazli, S. Pouyan, and H. Fathi, "High performance offline signature verification and recognition method using neural network," *International journal of advance studies in computer science and engineering*, vol. 4, no. 6, pp. 9–13, 2015.

[17] R. R. Bhausaheb, S. Kumar, and K. Sachin, "Centroidal distance based offline signature recognition using global and local features," *IPASJ International journal of computer science (IIJCS)*, vol. 3, no. 3, pp. 036–042, Mar. 2015.

[18] P. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, pp. 291–307, 1999.

[19] H. Patel, S. Desai, P. Desai, and A. Damani, "Review on Offline Signature Recognition and Verification Techniques," *International Journal of Computer Applications*, vol. 179, no. 53, pp. 35–41, Jun. 2018.

[20] D. Lowe, "Distinctive image features from scale-invariant key-points," *International Journal of Computer Vision*, vol. 12, pp. 91–110, 2004.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[22] P. Loncomilla, J. R. del Solar, and L. Martinez, "Object recognition using local invariant features for robotic applications: A survey," *Pattern Recognition*, vol. 60, pp. 499–514, Dec. 2016.

[23] O. S. Vaidya and S. T. Gandhe, "The study of preprocessing and postprocessing techniques of image stitching," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 2018, pp. 431–435.

[24] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face antispoofing using speeded-up robust features and fisher vector encoding," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 141–145, 2016.

[25] A. Al-Tamimi, M. Jaradat, N. Aljarrah, and S. Ghanim, "AARI: automatic arabic readability index," *Int. Arab J. Inf. Technol.*, vol. 11, no. 4, pp. 370–378, 2014.

[26] A. Al-Tamimi, A. M. Shatnawi, and E. B. Issa, "Arabic sentiment analysis of YouTube comments," *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Aqaba, 2017, pp. 1–6, doi: 10.1109/AEECT.2017.8257766.