Performance Comparison of Feature Descriptors in Offline Signature Verification

ABM Ashikur Rahman*, Md Redwan Karim, Rafsanjany Kushol and Md. Hasanul Kabir

***Abstract*—Handwritten Signature is a widely used biometric in daily life as a mean of identity verification of an individual. For offline signature verification both accuracy and speed are important parameters. Accuracy may vary as the samples from signature datasets show a high intra-class variability. As these properties depend on the feature descriptor taken to represent the signature image, this selection is very important. In this study we provide a comparative performance evaluation of well-known histogram based descriptors like SIFT and SURF and a wide variety of binary descriptors like BRIEF, ORB, BRISK and FREAK in the application of handwritten signature verification. We compare the performance of these feature descriptors against speed and accuracy. After the experimental analysis we have observed that binary features like ORB is faster with moderate accuracy but SIFT-like descriptors give better accuracy. Among them the combination of FAST feature detection and BRIEF descriptor is the fastest one but with lowest accuracy.**

***Keywords*—signature verification, feature detection, feature descriptors, SIFT, SURF, FAST, BRIEF, ORB, BRISK, FREAK.**

I. Introduction

Biometric technology is used in a wide variety of security applications. The main target of such systems is to verify the identity of a person based on physiological or behavioral traits. The first case is based on measurements of biological traits, such as the fingerprint, face, iris, etc. The other one is concerned with behavioral traits such as voice and the handwritten signature. Offline handwritten signature has been one of the most used methods because of its simplicity and ease of the user.

The problem of offline signature verification is usually modelled as a verification task. If a single model is used to classify the signature images of any user, it is called Writer Independent (WI) system. In contrast, The model used to train for different individual user is known as Writer Dependent (WD) system.[1] Offline signature verification is very challenging task as a few amount of information can be extracted from static signature images compared to the online signature systems. Signature images are often prone to noises which distorts the image. The main challenge for this system is having a very high intra-class variability which means authentic signatures of a person may vary from one another. Compared to physical biometric traits, such as fingerprint or iris, handwritten signatures from the same user often show a large variability between samples. This problem is illustrated in Fig.1. To overcome these challenges it is important to find a good representation of those signature images which is distinct and unique for each person. Features that represent signature images should be distinctive to differentiate the person giving the signature as well as robust to scale, rotation, and noise.

Over the last few decades, a number of key papers have surveyed and summarized the advancements in this field, in the late 80s [3], 90s [4], 2000s [5]. However, over the last decades mention worthy surveys are hafeman et al. [1], Impedovo et al. [6], Okawa et al. [7] and Ruiz-del-solar et al. [8]. These surveys mostly focus on the methodologies used in the verification process. Over the years, many features descriptors have been proposed to serve the purpose of matching. Some key survey papers evaluated their performance and summarized them. [9], [10], [11], [12], [13], [14] Most of them performed the survey from a general point of view. They tested the performance over a benchmark dataset of images.[15] But a study regarding the performance of the feature descriptors in the domain of offline signature verification is rare.

In this paper, we are going to provide a comparative analysis of the performance of different feature descriptors for the verification of offline signature. We have taken 6 combinations of popular feature detection and descriptor methods such as SIFT, SURF, BRIEF, ORB, BRISK and FREAK. For testing we have used ICDAR 2011 dataset. [16] We have taken Area Under the ROC Curve (AUC), Accuracy, Precision vs Recall graph, F-score and running time as the metrics of comparison. Our study in this paper found that: Binary descriptors work faster and descriptors occupy less amount of memory space. Combination of FAST as keypoint detection method and BRIEF as feature descriptor is the fastest one but it lacks at performance accuracy. On the other hand, SIFT-like descriptors take more time and space but result in better accuracy. In section 2 we introduce an overview of the feature

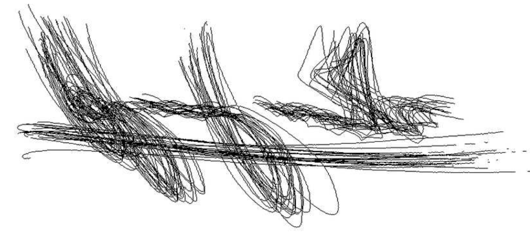


Fig. 1. Intra-class variability of signature image [2]

*Corresponding author.

ABM Ashikur Rahman*, Md Redwan Karim, Rafsanjany Kushol and Md. Hasanul Kabir are with the Deptartment of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. e-mail: ashikiut@iut-dhaka.edu, redwankarim@iut-dhaka.edu, kushol@iut-dhaka.edu and hasanul@iut-dhaka.edu.

Manuscript received 20 August, 2018; revised 20 November,2018; Accepted 03 December,2018.

detection and descriptor methods. Following section 3 has details about the dataset and experimental setup. Then we provide description of the evaluation metrics and performance analysis. Finally a conclusion is drawn and future works are discussed in the very last section.

II. Feature Descriptors

For different image matching and detection applications, a big question is how the image will be represented and what features we should use to find the distinction between images. Over the years, researchers have proposed many algorithms to detect and describe feature of an image. A popular approach is to find interest points, widely known as keypoints. Then a suitable descriptor of these keypoints are found to represent the image. Feature detection is the process of computing the abstraction of the image information and making a local decision at every image point to see if there is an image feature of the given type existing at that point. Feature detection and image matching have been two important problems in machine vision and robotics, and their applications continue to grow in various fields. An ideal feature detection technique should be robust to image transformations such as rotation, scale, illumination, noise and affine transformations. In addition, ideal features must be highly distinctive, such that a single feature to be correctly matched with high probability. The features evaluated in this paper have five major stages. Most of the keypoint based feature descriptors for the image can be classified into two groups.

1. Float point Image descriptors / Histogram of Oriented Gradients (HOG) based Descriptors
2. Binary Image Descriptors

A. Histogram of Oriented Gradients (HOG) based Descriptors

Histograms of this class are prominent for their performance in accuracy. They calculate the features in float points. Members of this family are- SIFT, SURF and GLOH. The most popular member of them is SIFT.

1) Scale Invariant Feature Transform (SIFT) [17] : SIFT includes both keypoint detector and descriptor. It has four major stages.

1) Scale-space Extreme Detection
2) Keypoint Localization
3) Orientation Assignment
4) Keypoint Descriptor

First the scale space extrema is calculated using Difference of Gaussian (DoG) for potential Keypoints for the image. Then these keypoints are refined for better accuracy. Low contrast points is eliminated by thresholding. The edge response which is resulted from DoG are removed using the concept of Harris Corner Detector and Hessian Matrix. Thirdly, an orientation is assigned to achieve rotation invariance. The Last step is to generate keypoint descriptor using local gradient magnitude and orientation. For this, the region around a given keypoint is warped to a 16x16 pixels putting the keypoint at the center. Then, gradients for each and every pixel is computed. SIFT divides this region into 16(4x4) sub-regions. Each sub region has 8 histograms which create a 128 bin long feature vectors for the keypoints. A method to use this local gradient feature for offline signature verification was developed by Ruiz-del-Solar et al. [18]

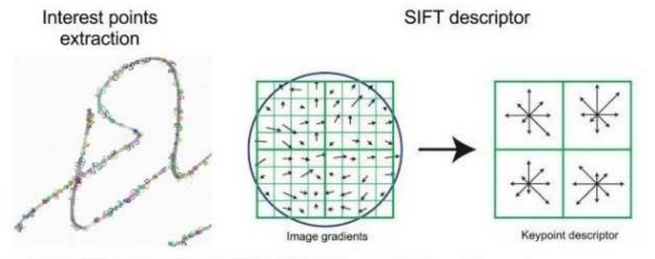


Fig. 2. Keypoint detected by SIFT.

2) Speeded Up Robust Feature (SURF) [19]: Though SIFT performs well, still there was a question about the speed of the processing. To answer that question SURF was proposed in 2006 by Bay et al.[19]. To find the scale space SIFT approximates Laplacian of Gaussian (LoG) using Difference of Gaussian (DoG) which is costly. SURF proposed to approximate that applying a Box-filter. Convolution with Box filter can be calculated very efficiently using integral images. One of the major advantages of Box filter is that it can be done parallel for different scales. For both the scale and location, SURF depends on the determinant of Hessian Matrix. For assigning the orientation SURF calculates the wavelet responses in horizontal and vertical direction for a neighborhood. For feature descriptor, a neighborhood of 20s x 20s is taken around a keypoint where s is the size. It is then divided into 4x4 subregions. For each of the subregions, vertical and horizontal wavelet responses are combined to create a vector like equation 1.

$$V = (dx, dy, |dx|, |dy|) \tag{1}$$

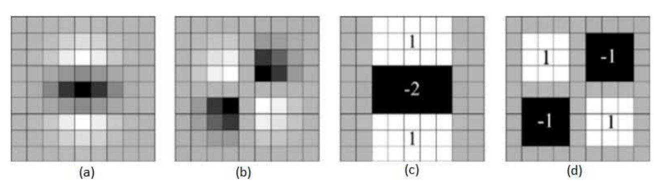


Fig. 3. (a)(b) Gaussian second order partial derivative in y-direction and xy-direction respectively, (c)(d) Approximation of the second order Gaussian partial derivative by SURF. [19]

In total 16 regions contribute to create a feature descriptor of dimension 64. Another variation of SURF uses a descriptor of dimension 128 to provide better distinctiveness. SURF features has been used for a number of image matching application. Malik et al. proposed to use it for handwritten signature verification. [20]

B. Binary Image Descriptors

Though Histogram based descriptors are very good at performance, the question comes about their efficiency. The gradients of all the pixels in the patch need to be computed which is very costly. Even the speed boost up by SURF is not enough. These methods give the descriptor in float numbers and dissimilarity metric is L2 norm. Computing L2 norm is also a costly operation. Moreover, encoding these float point descriptors takes much memory space. These are the aspects where Binary descriptors come in handy.[21]

Binary descriptors encode the information of a patch in binary strings. It compares the intensity of the reference points in the patches and stores the result in either 0 or 1. This operations are fast and can be stored using very less amount of space. A Distance measure between two binary strings is computed using Hamming distance. Then matching of binary strings can be done using a single XOR operation of the processor. These are the exact motivation behind the binary descriptors.

Most of the binary descriptors work in similar fashion with small differences. These descriptors are composed of three parts mostly: [21]

1) A sampling pattern: to find the sampling points around the keypoints
2) Orientation Compensation: to measure the orientation of the keypoint to make it rotation invariant
3) Sampling Pairs: to find which pairs to compare to build the final descriptor

1) FAST (Features from Accelerated Segment Test): FAST corner detection algorithm was presented on 2010 by Rosten et al. [22] It is proposed based on SUSAN corner criterion [23]. Similar to SUSAN, FAST takes a circle of 16 pixels from the neighborhood of a potential keypoint candidate. These 16 pixels can be chosen by a Bresenham circle [24] of radius 3. Based on the value of these pixels, it is determined whether the candidate is a keypoint or not. As plotted in figure x, the intensity values of the chosen pixels are compared with the pixel of the candidate. If n pixels among the 16 fulfills the threshold criterion then the candidate is taken as interest point the value of n is usually taken as 12. To make it faster, all 16 pixels are indexed clockwise and FAST compares the intensity values of pixels 1, 5, 9, 13 from the circle. At least any three of these four pixels should be within the threshold for the candidate to be keypoint. If a candidate pass this test, FAST goes for further testing. Otherwise it rejects the candidate. This faster approach works with good speed but has a few weakness. Along with the other weakness, numbering the pixels order is an overhead and multiple features are detected adjacent to one another. To overcome those weakness machine learning approach is taken and the keypoints being adjacent to each other is addressed using non-maximal suppression. Noting that Fig.4 is taken from website [25].

2) Binary Robust Independent Elementary Features (BRIEF): BRIEF is one of the first of its kind. It is a comparison between the intensities of random pixel pairs in the patch centered at a detected keypoints. Firstly, the patch is smoothed using a Gaussian filter to make it less sensitive to noise. Then to make a length ***n*** BRIEF descriptor, ***n*** pairs

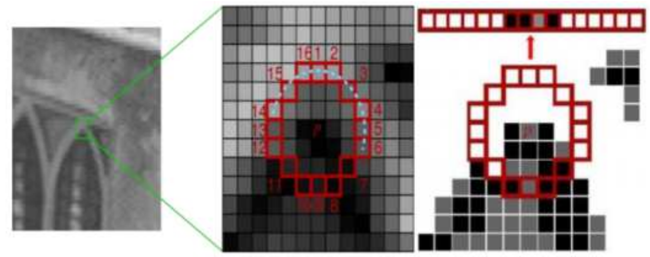


Fig. 4. (a) A processed interest point and 16 pixels surrounding on it, (b) the demonstration of storing 16 values surrounding pixels in a vector form. [25]

are determined using any of the five methods shown in Fig. 5.[26] Now, the comparisons between the pairs are encoded in binary to build the descriptors. As BRIEF is created using comparisons only instead of computing gradients and Histogram pooling, it is faster than SIFT-like descriptors. And using not more than 512 bits BRIEF descriptors can be stored in less space comparing to its floating point alternatives.

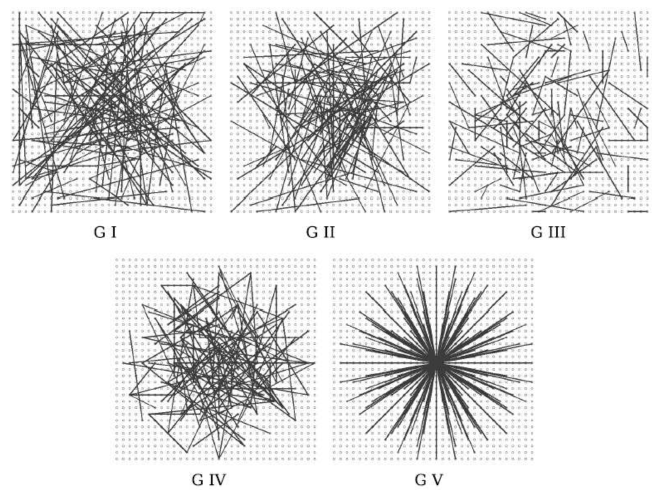


Fig. 5. Different approaches used by BRIEF to choose the test locations. [26]

3) Oriented FAST and Rotated BRIEF (ORB): ORB [27] is a combination of two popular Algorithms (FAST [22] and BRIEF[26]). To find the keypoints it uses FAST algorithm. As FAST does not produce multi-scale features, ORB constructs scale pyramids and finds keypoints at each scale. Once the keypoints are detected, to sort them and to remove the edge responses Harris Corner detection [28] is used. After detecting the keypoints, descriptor is constructed by the idea of BRIEF. BRIEF does not have orientation information and is rotation variant. To compensate orientation local first order moments are used. Another important modification made by ORB is to propose an unsupervised learning for choosing sampling pairs instead of a random selection. Fig.6 shows all the pairs for ORB.

4) Binary Robust Invariant Scalable Keypoints (BRISK) : Another member of binary descriptors family is BRISK [29] which has same the structure as BRIEF[26] and ORB[27]. Unlike its ancestors using random sampling or unsupervised learning of pairs, BRISK uses a specially crafted concentric-

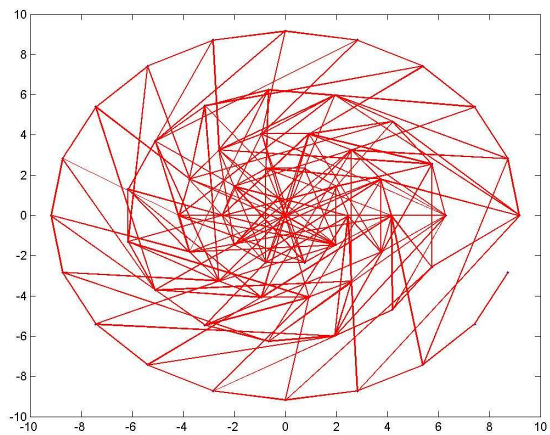


Fig. 6. Sampling pattern of ORB [21]

rings sampling pattern which is shown in Fig.7. While using this sampling patterns, all the pairs are grouped into either a short pair or a long pair based on the distance between them. Long pairs are used to compensate for the orientation of the patch while the short ones are used to build the descriptor comparing the intensity. For assigning orientation to the keypoints, all the local gradients between the long pairs are summed up and arctan is taken. After finding the angle, short pairs are rotated accordingly to achieve rotation invariance. To finally build the descriptor, intensity comparison of the short pairs are done just like BRIEF and ORB which gives a descriptor of length 512 bits. Hamming distance is used as a dissimilarity metric.

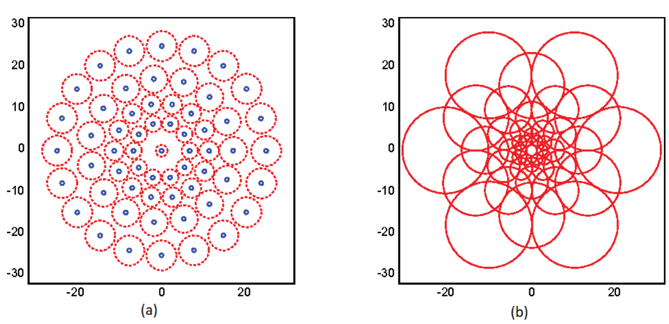


Fig. 7. (a) The BRISK Sampling pattern with 60 points, [29] (b) The FREAK sampling pattern [30]

5) Fast REtinA Keypoint (FREAK) : FREAK [30] is another binary descriptor which uses methods similar to BRISK. Instead of using hand-crafted concentric-ring sampling pattern, it uses retinal sampling grid. Retinal sampling grid is also circular but the density of points is higher near the center of the patch and decreases exponentially as we go distant from the center. This pattern is inspired by the Retinal pattern of human eye. Computing the orientation is same as BRISK[29] with the difference of using a predefined set of symmetric sampling pairs instead of long pairs. This coarse-to-fine structure allows FREAK to increase the speed during the matching of descriptors. First it compares the first 128 bits and further continues the comparison if the distance is lower than a threshold.

III. Result Analysis

A. Experimental Setup and Data Set

Our System for evaluation was implemented on openCV which has been run on a personal computer of 3.6 GHz processors with 16 GB main memory with windows 10 operating system. There is a number of signature datasets available to test the performance of the verification system. We have chosen a very well-known dataset provided by [16]. This dataset contains both offline and online signatures. Offline signatures comprise of PNG images which were scanned at 400 dpi and RGB color coded. The offline dataset has two parts- Chinese Dataset and Dutch Dataset. In Dutch Dataset, there are 362 total signatures of 10 reference for training the system. For testing, Test set contains 1932 signature comprising signature from 54 reference writers and skilled forgeries of these signatures. This Dutch dataset was used for our study. For matching the signature we used the method of Rahman et al. [31]. Instead of using 3 signatures from the reference signature, we used all 12 of them to train the system. Matching for testing was done accordingly. For Histogram based descriptors we have used FLANN based matching and for Binary descriptors BF matcher is used.

B. Evaluation Criterion

A signature verification system verifies a signature image which claims to belong to an individual. This verification process has two results. Either the signature is genuine or it is a forged one. Given a classifier and a signature, there are four possible outcomes. If the signature is genuine and it is classified as genuine, it is counted as a true positive; if it is classified as forged, it is counted as a false negative. If the signature is forged and it is classified as forged, it is counted as a true negative; if it is classified as genuine, it is counted as a false positive. Given a classifier and a set of instances (the test set), a two-by-two confusion matrix (also called a contingency table) can be constructed representing the dispositions of the set of instances. This matrix forms the basis for many common metrics.

From this matrix many performance evaluation metrics can be derived. [32] The true positive rate (tp rate) is also known as hit rate or recall of a classifier whereas false positive rate (fp rate) is known as false alarm rate. Positive predictive value is widely known to be Precision. Another important metric is F1-measure. These metrics can be estimated using the following equations:

$$fprate = FP/(FP + TP) \tag{2}$$

$$Recall = TP/(TP + FN) \tag{3}$$

$$Precission = TP/(TP + FP) \tag{4}$$

$$Accuracy = (TP = TN)/(TP + FP + FN + TN) \tag{5}$$

$$F1 - measure = 2/(1/Precision + 1/Recall) \quad (6)$$

If we draw a two-dimensional graph taking true positive rate (tp rate) in Y-axis and false positive rate (fp rate) in X-asis we get a ROC graph. Varying the threshold value we can get a ROC curve which gives an important evaluation metric named Area Under ROC Curve (AUC). For AUC, higher value represents better performance. These metrics evaluate the quality of the descriptors when we perform image matching. Computation time is another form of metric which evaluates the speed of the algorithm.

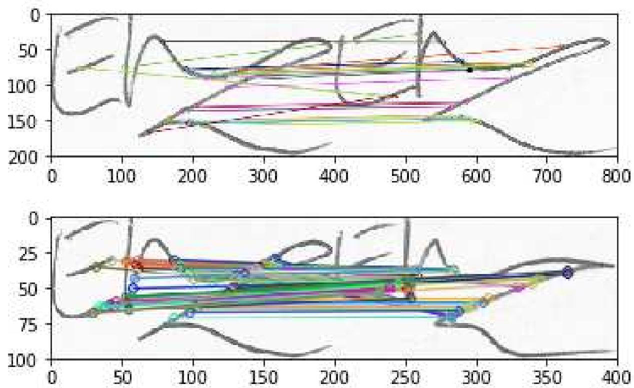


Fig. 8. Matched keypoints in Dutch Dataset

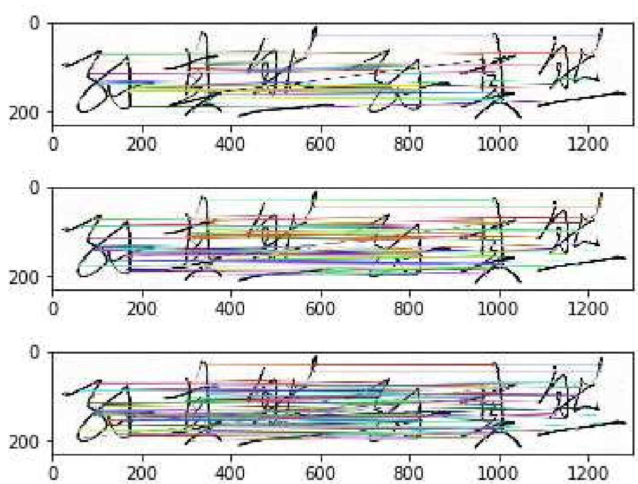


Fig. 9. Matched keypoints in Chinese Dataset

C. Comparison of Computation Time

To compare the speed of the algorithms we have run them in our system and a time log is kept. Evaluation metric of speed is time per keypoints which refer lower value to be better in performance. Table I gives us the speed comparison result.

TABLE I
PERFORMANCE ANALYSIS WITH COMPUTATION TIME

Feature detection method	Feature descriptor	Time (ms)	Number of keypoints	Time / point
SIFT	SIFT	112.98	598	0.1889
SURF	SURF	35.98	855	0.04208
FAST	BRIEF	8.0003	930	0.0086
ORB	ORB	9.003	475	0.01895
BRISK	BRISK	81.984	1307	0.06273
FAST	FREAK	38.996	598	0.04041

TABLE II
PERFORMANCE ANALYSIS WITH CHINESE DATASET

Feature detection method	Feature descriptor	Accuracy	AUC	F1-measure
SIFT	SIFT	0.8333	0.924	0.8462
SURF	SURF	0.875	0.935	0.8800
FAST	BRIEF	0.75	0.854	0.7000
ORB	ORB	0.8333	0.915	0.8333
BRISK	BRISK	0.7916	0.894	0.7826
FAST	FREAK	0.7916	0.889	0.8000

D. Feature descriptor Performance comparison

Then we used the writer dependent method proposed by Rahman et al. [31] the performance evaluation. The performance of the Chinese and Dutch dataset is given in Table II and Table III respectively.

From the values of Table II and Table III, we can see that float point descriptors outperforms binary descriptors almost in all the metrics. Among the binary descriptors, the performance of ORB is better than the others. We can also compare their Precision and Recall. Fig. 11 shows this comparison for Chinese Dataset and Fig. 10 gives visual Dutch Dataset.

From these bar charts, we can see that SURF outperforms the other ones where BRIEF performs worst. The reason

TABLE III
PERFORMANCE ANALYSIS WITH DUTCH DATASET

Feature detection method	Feature descriptor	Accuracy	AUC	F1-measure
SIFT	SIFT	0.7917	0.911	0.8000
SURF	SURF	0.8333	0.928	0.8462
FAST	BRIEF	0.6250	0.823	0.5714
ORB	ORB	0.8263	0.882	0.8182
BRISK	BRISK	0.6667	0.86	0.6923
FAST	FREAK	0.7083	0.876	0.6957

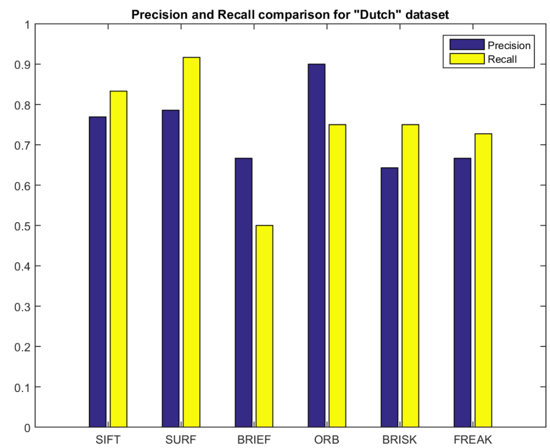


Fig. 10. Precision and Recall comparison for Dutch Dataset

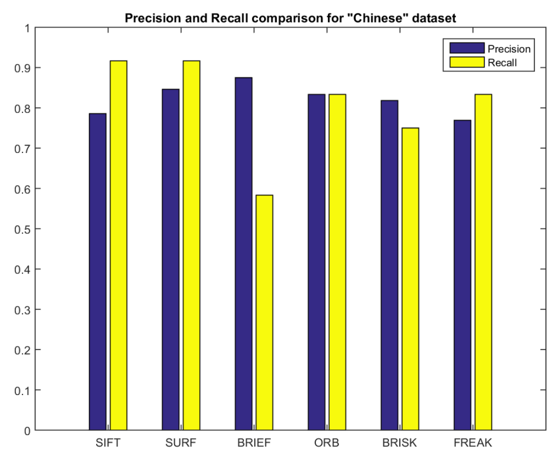


Fig. 11. Precision and Recall comparison for Chinese Dataset

behind this is float point descriptors have higher resolution to represent the keypoints. More information leads to better performance but slows down the process. On the other hand binary descriptors are faster but performance goes a bit down. Among the descriptors BRIEF performance the worst. As we know that BRIEF does not assign an orientation to the keypoints and thus rotation variant. Signature images are often prone to rotation. Even a Genuine signature from a reference person can become slant. BRIEF fails to verify this kind of signatures causing poor performance.

One of the most important application of offline signature verification is banking. As there is financial transaction, authentication accuracy is a bigger concern. More importantly, false positive rate should be as less as possible. The method with lower false positive rate considered to be a better one for banking applications. Table IV illustrates the comparison of performance against false positive rate.

IV. Conclusion

From the performance evaluation we see that the performance metrics provide a trade-off between accuracy and time.

TABLE IV
PERFORMANCE AGAINST FALSE POSITIVE RATE (FPR)

Feature detection method	**Feature descriptor**	**FPR with Chinese Dataset**	**FPR with Dutch Dataset**
SIFT	SIFT	0.25	0.25
SURF	SURF	0.1667	0.25
FAST	BRIEF	0.083	0.2500
ORB	ORB	0.1667	0.0833
BRISK	BRISK	0.1667	0.4167
FAST	FREAK	0.25	0.25

Float features like SIFT and SURF give better accuracy but take longer time to finish the task. Binary descriptors are faster than their ancestors. BRIEF gives the fastest performance but lacks accuracy. ORB can be a quick one nearly catching the performance of histogram based ones. If the application needs real time action where accuracy is not the only concern, ORB can be a very good choice. But for an application where user can go for extra minutes for accuracy, float point descriptors are still the top choice. For banking application, lower false positive rate ensures that forged signatures is not considered as genuine. Thus money is not handed over to a fraud. Considering the performance in both the datasets, ORB again tops the list.

The comparisons in this study give sufficient insight to investigate the choice of a good descriptor. It also shows the demand of feature descriptor which combines the performance both in accuracy and speed. For the future work, more descriptors can be put into test. For offline signature verification, some recently proposed methodologies like neural networks and deep learning will be added to the comparison. However, we believe that this comparison is sufficient to analyze the fastest and robust method.

References

[1] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verificationliterature review," in *Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017, pp. 1–8.

[2] E. J. Justino, A. El Yacoubi, F. Bortolozzi, and R. Sabourin, "An off-line signature verification system using hmm and graphometric features," in *Proc. of the 4th international workshop on document analysis systems*, 2000, pp. 211–222.

[3] R. Plamondon and G. Lorette, "Automatic signature verification and writer identificationthe state of the art," *Pattern recognition*, vol. 22, no. 2, pp. 107–131, 1989.

[4] F. Leclerc and R. Plamondon, "Automatic signature verification: The state of the art1989–1993," *International journal of pattern recognition and artificial intelligence*, vol. 8, no. 03, pp. 643–660, 1994.

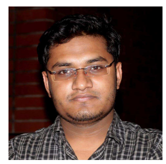
[5] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 609–635, 2008.

[6] D. Impedovo, G. Pirlo, and R. Plamondon, "Handwritten signature verification: New advancements and open issues," in *International conference on Frontiers in handwriting recognition (ICFHR)*. IEEE, 2012, pp. 367–372.

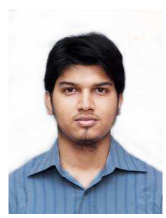
[7] M. Okawa, "From bovw to vlad with kaze features: Offline signature verification considering cognitive processes of forensic experts." Elsevier, 2018.

[8] J. Ruiz-del Solar, C. Devia, P. Loncomilla, and F. Concha, "Offline signature verification using local interest points and descriptors," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2008, pp. 22–29.
[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," vol. 27, no. 10. IEEE, 2005, pp. 1615–1630.
[10] Ş. Işık, "A comparative evaluation of well-known feature detectors and descriptors," vol. 3, no. 1, 2014, pp. 1–6.
[11] M. Kashif, T. M. Deserno, D. Haak, and S. Jonas, "Feature description with sift, surf, brief, brisk, or freak? a general question answered for bone age assessment," vol. 68. Elsevier, 2016, pp. 67–75.
[12] E. Bayraktar and P. Boyraz, "Analysis of feature detector and descriptor combinations with a localization experiment for various performance metrics," 2017.
[13] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: performance comparison for distorted images," 2017.
[14] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," vol. 40, no. 5. IEEE, 2018, pp. 1224–1244.
[15] "Oxford dataset," [Online; accessed 13-September-2018]. [Online]. Available: robots.ox.ac.uk/ vgg/data/data-aff.html
[16] M. Liwicki, M. I. Malik, C. E. Van Den Heuvel, X. Chen, C. Berger, R. Stoel, M. Blumenstein, and B. Found, "Signature verification competition for online and offline skilled forgeries (sigcomp2011)," in *International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2011, pp. 1480–1484.
[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *The proceedings of the seventh IEEE international conference on Computer vision.*, vol. 2. IEEE, 1999, pp. 1150–1157.
[18] J. Ruiz-del Solar, C. Devia, P. Loncomilla, and F. Concha, "Offline signature verification using local interest points and descriptors," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2008, pp. 22–29.
[19] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
[20] M. I. Malik, M. Liwicki, A. Dengel, S. Uchida, and V. Frinken, "Automatic signature stability analysis and verification using local features," in *14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2014, pp. 621–626.
[21] Gil Levi, "Gil's computer vision blog," 2015, [Online; accessed 13-September-2018]. [Online]. Available: https://gilscvblog.com/
[22] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," vol. 32, no. 1. IEEE, 2010, pp. 105–119.
[23] S. M. Smith and J. M. Brady, "Susan: a new approach to low level image processing," in *International journal of computer vision*, vol. 23, no. 1. Springer, 1997, pp. 45–78.
[24] J. Bresenham, "A linear algorithm for incremental digital display of circular arcs," in *Communications of the ACM*, vol. 20, no. 2. ACM, 1977, pp. 100–106.
[25] E. Rosten, "Fast corner detection," [Online; accessed 13-September-2018]. [Online]. Available: http://www.edwardrosten.com/work/fast.html
[26] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.
[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *IEEE international conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2564–2571.
[28] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.
[29] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2548–2555.
[30] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2012, pp. 510–517.
[31] A. A. Rahman, G. Mostaeen, and M. H. Kabir, "A statistical approach for offline signature verification using local gradient features," in *International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*. IEEE, 2016, pp. 1–4.
[32] T. Fawcett, "An introduction to roc analysis," in *Pattern recognition letters*, vol. 27, no. 8. Elsevier, 2006, pp. 861–874.

A.B.M. Ashikur Rahman received his B.Sc. Engg. in Computer Science and Engineering degree from Islamic University of Technology (IUT), Bangladesh in 2014. Currently he is working as a lecturer in the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Bangladesh and also pursuing his M.Sc. Engg. in Computer Science and Engineering at the same university. His research interests include computer vision, machine learning and data mining.



Md Redwan Karim Sony received his B.Sc. Engg. in Computer Science and Engineering degree from Islamic University of Technology (IUT), Bangladesh in 2016. Currently he is working as a lecturer in the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Bangladesh and also pursuing his M.Sc. Engg. in Computer Science and Engineering at the same university. His research interests include image processing, computer vision, pattern recognition, machine learning and data mining.



Rafsanjany Kushol received his B.Sc. Engg. degree and M.Sc. Engg. degree in Computer Science and Engineering from Islamic University of Technology (IUT), Bangladesh in 2013 and 2018 respectively. Currently he is working as a lecturer in the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Bangladesh and also pursuing his M.Sc. Engg. in Computer Science and Engineering at the same university. His research interests include image processing and computer vision.

Md. Hasanul Kabir received his B.Sc. degree from Islamic University of Technology (IUT), Bangladesh and PhD degree in Computer Engineering from the Kyung Hee University, South Korea. Currently he is working as a professor in the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Bangladesh. His research interests include feature extraction, motion estimation, image processing, computer vision and pattern recognition.