**Performance Comparison of Feature Descriptors in Offline Signature Verification**

A.B.M. Ashikur Rahman, Md. Redwan Karim Sony, Md. Hasanul Kabir and Rafsanjany Kushol

Islamic University of Technology (IUT), Dhaka, Bangladesh

Email: [ashikiut@iut-dhaka.edu](mailto:ashikiut@iut-dhaka.edu)

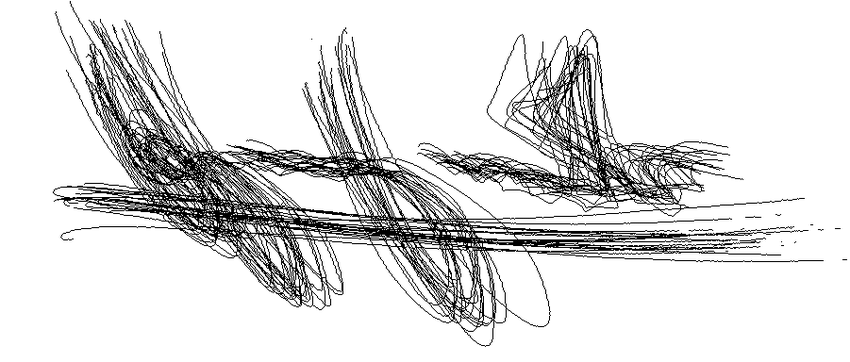
**Abstract – Handwritten Signature is a widely used biometric in or daily life as a mean of identity verification of an individual. For offline signature verification both speed and accuracy are important parameters. Accuracy may fall as signature datasets shows a high intra-class variability. As these properties depend on the feature descriptors used to represent the signature image, this choice is very crucial. In this study we provided a comparative performance evaluation of some well-known histogram based feature descriptors like SIFT and SURF and a wide variety of binary descriptors like FAST+BRIEF, ORB, BRISK and FREAK in the application of handwritten signature verification. We compared the performance of these image descriptors for their speed and accuracy. After the investigation we observed that binary feature like ORB is faster with moderate accuracy but SIFT-like descriptors give better accuracy. Among them the combination of FAST and BRIEF is the fastest one but with lowest accuracy.**

**Keywords- handwritten signature verification, feature detection, feature descriptors, SIFT, SURF, FAST, BRIEF, ORB, BRISK, FREAK, performance analysis.**

1. **INTRODUCTION**

With the rise of new technologies, machines are contributing more to human civilization now than any other time in history. Once computers was used to ease the calculation whereas now a days every single aspect of life is benefited from it. Even computer is being used to verify biometrics. 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 later one is concerned with behavioral traits such as voice and the handwritten signature. Offline handwritten signature has been one of the most used method because of simplicity and ease of the user. But it is the most challenging one as a very small amount of information can be extracted from static signature images compared to the online signature systems which contains the gestures, speed and mode of the signatory etc. Signature images are often prone to noises which distorts the image. Moreover, authentic signature of a person may vary from one another. To overcome those 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 and robust to scale, rotation and noise.

**Problem formulation: [c:hafeman]**



Over the years, many features descriptors have been proposed to serve the purpose of matching. 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 combination of popular feature detection and descriptor methods such as SIFT, SURF, BRIEF, ORB, BRISK and FREAK. For testing we have used ICDAR 2011 dataset. [c:dataset] 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. FAST + BRIEF is the fastest one but it lacks at performance. On the other hand, SIFT-like descriptors consumes more time and space but results in better accuracy.

In section 2 we introduce an overview of the feature detection and descriptor methods. Following section 3 has details about 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.

1. **LITERATURE REVIEW**

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 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 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 in 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 image can be classified into two groups.

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

**Histogram of Oriented Gradients (HOG) based Descriptors**

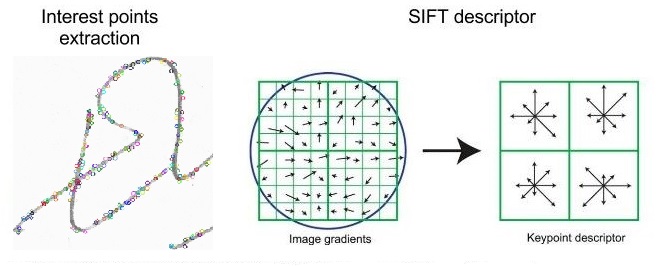
Histograms of this class are prominent for their performance in accuracy. They calculate the features in float points. Members of the family are- SIFT, SURF and GLOH. Most popular member of them is SIFT.

**Scale Invariant Feature Transform (SIFT)[1999]** [c]

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

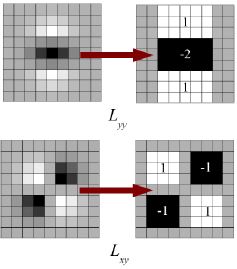
First the scale –space extrema is calculated using Difference of Gaussian (DoG) for potential Keypoints for the image. Then these keypoints are refined for better accuracy. Low contrast points are eliminated by thresholding. The edge response which is resulted from DoG are removed using the concept of Harris Corner Detector and Hessian Matrix. Thirdly, an orientation is assigned to achieve rotation invariance. 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 of the keypoints.



**Speeded Up Robust Feature (SURF)[2006]** [c]

Though SIFT performs well, still there was question about the speed of the processing. To answer that question SURF was proposed on 2006 by Bay et al.[c]. 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 Box-filter. Convolution with Box filter can be calculated very efficiently using integral images. Figure # shows a box filter. 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 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 of form

Equation { v =( dx, dy, |dx|, |dy| )}



In total 16 regions contributes to create a feature descriptor of dimension 64. Another variation of SURF uses descriptor of dimension 128 to provide better distinctiveness.

**Binary Image Descriptors**

Though Histogram based descriptors are very good at performance, the question comes about efficiency. SIFT[c] and SURF[c] are based on the Histogram of gradients. So, 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.

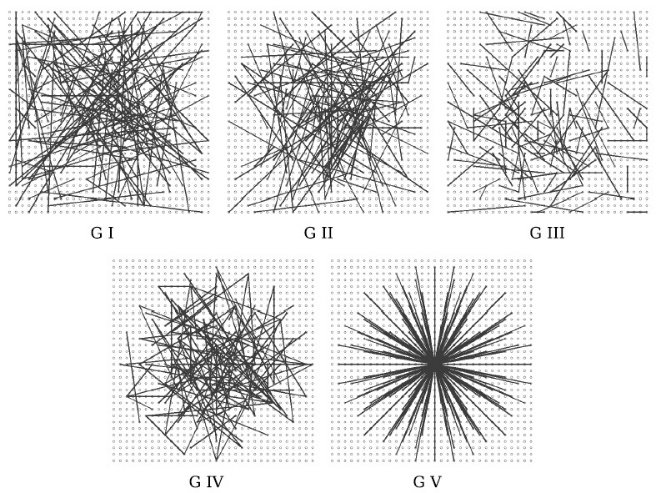
Binary descriptors encodes the information of a patch in binary strings. It compares the intensity of the reference points in the patches and stores the result in either 0 or 1. This operations are fast and can be stored using very less amount of space. Distance measure between two binary strings is computed using Hamming distance. Then matching of binary strings can be done using a single XOR operation of 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: [*gillevi*] [c]

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

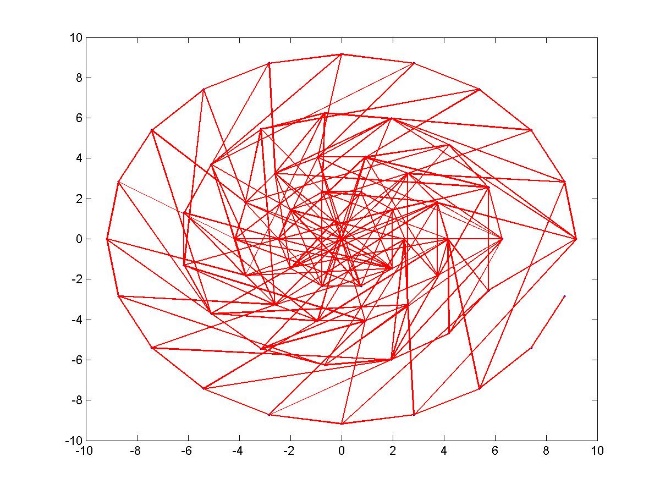
**Binary Robust Independent Elementary Features (BRIEF)[2010]**

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. First the patch is smoothed using Gaussian filter to make it less sensitive to noise. Then to make a length n BRIEF descriptor, n pairs are determined using any of the five methods shown in Figure #.[2010] 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.



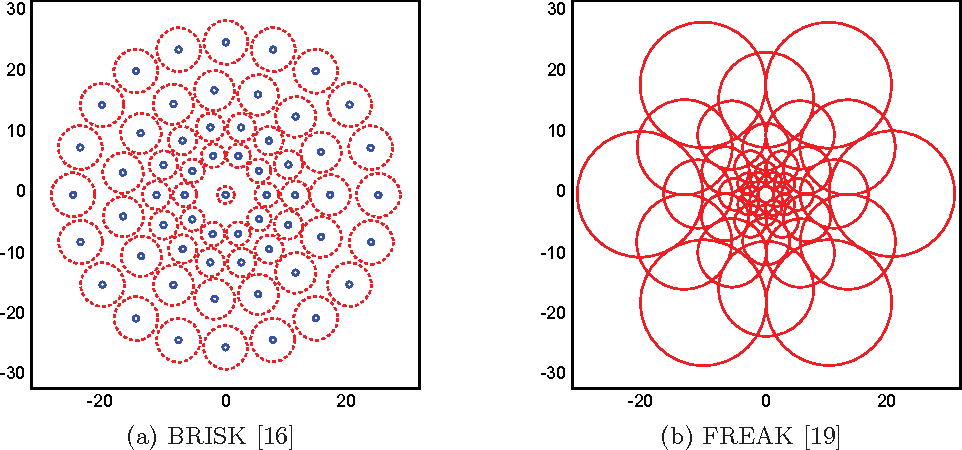
**Oriented FAST and Rotated BRIEF (ORB) [2011]**

ORB is a combination of two popular Algorithms (FAST [c] and BRIEF[c]). 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 [cite] is used. After detecting 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. Figure # shows all the pairs for ORB.



**Binary Robust Invariant Scalable Keypoints (BRISK)** [2011] [c]

Another member of binary descriptors family is BRISK which has same structure as BRIEF**[c]** and ORB. Unlike its ancestors using random sampling or unsupervised learning of pairs, BRISK uses a specially crafted concentric-rings sampling patterns shown in Figure #. 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 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 a dissimilarity metric.



**Fast REtinA Keypoint (FREAK) [c]**

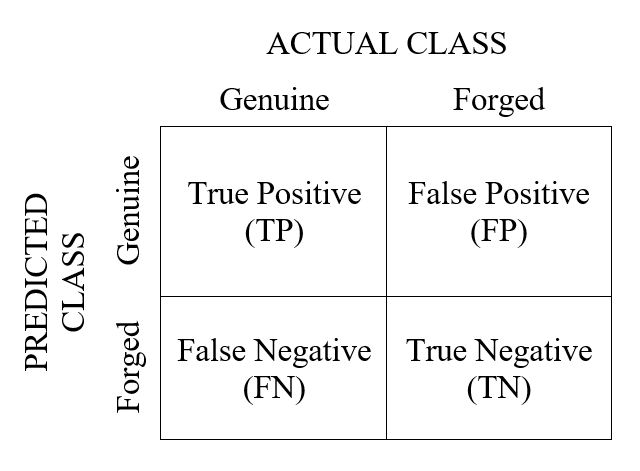
FREAK is another binary descriptor which uses methods similar to BRISK. Instead of **using** hand-crafted concentric-ring sampling pattern, it uses retinal sampling grid. Retinal sampling grid is also circular but the density of points is higher near the center of the patch and decreases exponentially 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**[c]** 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.

1. **RESULT ANALYSIS**

**Experimental Setup and Data Set**

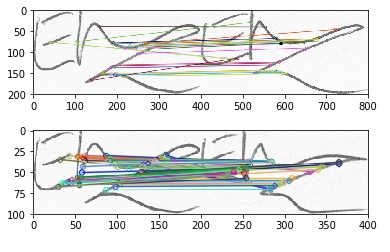
Our System for evaluation was implemented on openCV which has been run on a personal computer of 3.6 GHz processors with 16 GB main memory with windows 10 operating system. There is a number of signature datasets available to test the performance of verification system. We have chosen a very well-known dataset provided by [c:dataset]. This dataset contains both offline and online signatures. Offline signatures comprises 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. [c:statistical]. 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 used FLANN based matching and for Binary descriptors BF matcher was used.

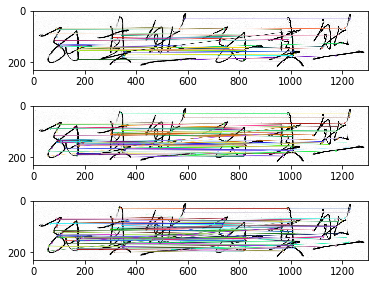
**Evaluation Criterion**

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. Fig - # shows a confusion matrix. 

From this matrix many performance evaluation metrics can be derived. [c:ROC] 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 F-measure. These metrics can be estimated as the following equations:

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 an ROC graph. Varying the threshold value we can get a ROC curve which gives an important evaluation metric named Area Under ROC Curve (AUC). AUC represent higher value for better performance. These metrics evaluates the quality of the descriptors when we perform image matching. Computation time is another form of metric which evaluates the speed of the algorithm.



****

**Comparison of Computation Time**

To compare the speed of the algorithms we 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 # gives us the speed comparison.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature detection method | Feature descriptor | Time (ms) | # 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 | 965 | 0.04041 |

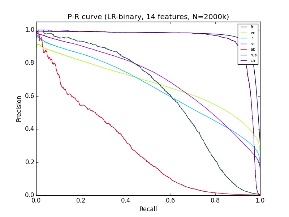
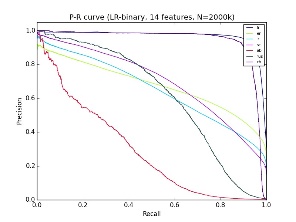
**Feature descriptor Performance comparison**

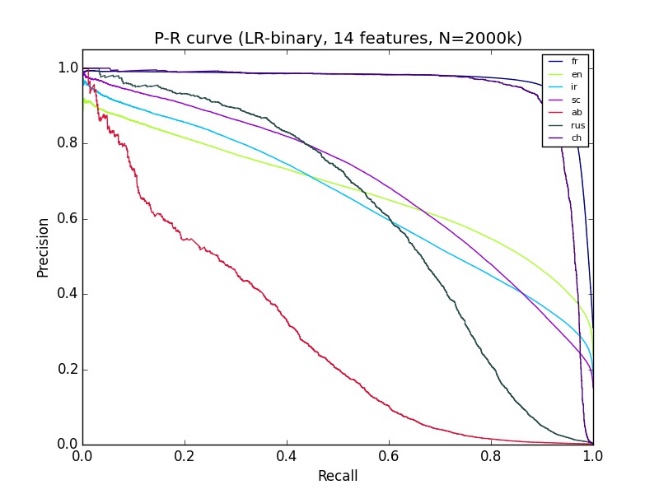
Then we ran the methods for matching the signatures to verify them. The performance for the Dutch and Chinese dataset is given in Table #.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature detection method | Feature descriptor | Accuracy | AUC | F - Score |
| SIFT | SIFT |  |  |  |
| SURF | SURF |  |  |  |
| FAST | BRIEF |  |  |  |
| ORB | ORB |  |  |  |
| BRISK | BRISK |  |  |  |
| FREAK | FREAK |  |  |  |

From the values of table # we can see that -----\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*………………..

We can also compare their Precision and Recall. Figure # a shows the comparison of Precision and Figure # b gives visual about their recall.





From these curves we can see that SURF out performs the other ones where FAST+BRIEF performs worst. The reason behind this is float point descriptors have higher resolution to represent the keypoints. More information leads to better performance but sows 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 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 these kind of signatures causing poor performance.

1. **CONCLUSION**

From the performance evaluation we see that the performance metrics provide a trade-off between accuracy and time. Float features like SIFT and SURF gives better accuracy but takes longer to finish the task. Binary descriptor like ORB can be a quick one with a moderate performance nearly catching the performance of histogram based ones. If the application needs real time action where accuracy is not the prime concern, ORB can be a very good choice. But for application like banking where customer and teller can go for extra minutes rather than giving money to a fraud, float point descriptors are still the top choice.

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 combine the performance both in accuracy and speed.

1. **REFERENCES**

**SIFT -** *Lowe, David G. “Object recognition from local scale-invariant features.” Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.‏*

**SURF -** *Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features.” Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. 404-417*

**BRIEF -** *Calonder, Michael, et al. “Brief: Binary robust independent elementary features.” Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010. 778-792.‏*

**ORB -** *Rublee, Ethan, et al. “ORB: an efficient alternative to SIFT or SURF.” Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.‏*

**BRISK -** *Leutenegger, Stefan, Margarita Chli, and Roland Y. Siegwart. “BRISK: Binary robust invariant scalable keypoints.” Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.‏*

**FREAK -** *Alahi, Alexandre, Raphael Ortiz, and Pierre Vandergheynst. “Freak: Fast retina keypoint.” Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.*

**FAST -** *Rosten E. Porter R. and Drummond, T. Faster and better: A machine learning approach to corner detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 32, Number 1, 2010, pp. 105-119.*

**Gil Levi -***https://gilscvblog.com/2013/08/26/tutorial-on-binary-descriptors-part-1/*

**Dataset -** *Marcus Liwicki, Michael Blumenstein, Elisa van den Heuvel, Charles E.H. Berger, Reinoud D. Stoel, Bryan Found, Xiaohong Chen, Muhammad Imran Malik. "SigComp11: Signature Verification Competition for On- and Offline Skilled Forgeries", Proc. 11th Int. Conference on Document Analysis and Recognition, 2011*

**Statistical -** *Rahman, ABM Ashikur, Golam Mostaeen, and Md Hasanul Kabir. "A statistical approach for offline signature verification using local gradient features." Electrical, Computer & Telecommunication Engineering (ICECTE), International Conference on. IEEE, 2016.*

**Confusion\_matrix\_image** *: rasbt.github.io*

**ROC:** *Fawcett, Tom, An introduction to ROC analysis*

**Hafeman -** *Offline Handwritten Signature Verification -Literature Review 2017*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | ACTUAL CLASS | |
|  |  | Genuine | Forged |
| PREDICTED CLASS | Genuine | True Positive (TP) | False Positive (FP) |
| Forged | False Negative (FN) | True Negative (TN) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature detection method | Feature descriptor | Accuracy  (chinese)  % | AUC | F1 - Score |
| SIFT | SIFT | 0.8333 |  | 0.8462 |
| SURF | SURF | 0.875 |  | 0.8800 |
| FAST | BRIEF | 0.75 |  | 0.7000 |
| ORB | ORB | 0.8333 |  | 0.8333 |
| BRISK | BRISK | 0.7916 |  | 0.7826 |
| FREAK | FREAK | 0.7916 |  | 0.8000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature detection method | Feature descriptor | Accuracy  (Dutch)  % | AUC | F - Score |
| SIFT | SIFT | 0.7917 |  | 0.8000 |
| SURF | SURF | 0.8333 |  | 0.8462 |
| FAST | BRIEF | 0.6250 |  | 0.5714 |
| ORB | ORB | 0.8333 |  | 0.8182 |
| BRISK | BRISK | 0.6667 |  | 0.6923 |
| FREAK | FREAK | 0.7083 |  | 0.6957 |