Journal of
Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549608

FEATURE EXTRACTION USING GRAY SCALE CO-OCCURRENCE MATRIX (GLCM) WITH BRISK HYBRID ALGORITHM IN DIABETIC RETINAL IMAGES

Senthilkumar R
Assistant Professor, Department of Computer science and Engineering, Hindusthan Institute of Technology, Coimbatore, India, senthilkumarresearch2017@gmail.com

Dr. A. Bharathi
Professor, Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, India, bharathia@bitsathy.ac.in

Abstract: Diabetic retinopathy is a common complication of diabetes and is a leading cause of vision loss in adults. Early detection and management of diabetic retinopathy is crucial to prevent further vision loss. In recent years, image processing techniques have been widely used for automated detection and diagnosis of diabetic retinopathy. In this study, this paper investigates the use of the Gray Level Co-occurrence Matrix algorithm for feature extraction from diabetic retinal images with BRISK algorithm. The GLCM algorithm is a popular method for feature extraction in computer vision, and its use for diabetic retinopathy detection has shown promising results. Results demonstrate that the Gray Level Co-occurrence Matrix (GLCM) algorithm can effectively extract features from diabetic retinal images and BRISK can be used for automated detection feature extraction and robotics and autonomous systems diagnosis of diabetic retinopathy. Finaly, the top 80% of the feature points with high reaction esteem were separated as the main element focuses. Evaluate the performance of the GLCM algorithm and BRISK using various performance evaluation units, including accuracy, precision, recall, and F1 score.
Keywords: GLCM, BRISK, MISSIDOR, DRIVE, Kaggle dataset, F1 score

1. Introduction

Diabetic retinopathy (DR) is a leading cause of blindness worldwide, particularly among working-age adults. DR is a complication of diabetes mellitus, which affects the blood vessels in the retina [1], the light-sensitive tissue at the back of the eye. Over time, high blood sugar levels can damage the blood vessels in the retina, leading to vision loss and blindness.

Diabetic retinopathy is a common complication of diabetes that can lead to blindness if not detected and treated early. Automated detection and classification of diabetic retinopathy in retinal fundus images [2] can help improve early diagnosis and treatment. Feature extraction is a crucial step in the automated detection and classification of diabetic retinopathy, and the ORB algorithm is a promising approach for this task.

Feature extraction is a piece of the dimensionality decrease process, in which, an underlying arrangement of the crude information is partitioned and diminished to additional reasonable gatherings. So when you need to handle it will be more straightforward. The main quality of these enormous informational indexes is that they have countless factors. These factors require a great deal of registering assets to process. So Component extraction assists with getting the best element from those huge informational indexes by choosing and joining factors into highlights, consequently, actually diminishing how much information. These highlights are not difficult to process, yet at the same time ready to depict the genuine informational index with precision and innovation.

Image handling is truly outstanding and most intriguing area. In this area essentially you will begin playing with images. So here, utilize numerous methods which incorporates highlight extraction too and calculations to identify elements, for example, shapes, edges, or movement in a computerized picture to deal with them.

According to the International Diabetes Federation, there were an estimated 463 million adults with diabetes worldwide in 2019 [3], and this number is expected to rise to 700 million by 2045. About one-third of people with diabetes have DR, and the prevalence of DR increases with the duration of diabetes.

DR is a major public health concern in many countries, particularly in low- and middle-income countries with limited access to healthcare and resources for screening and treatment. In these countries, many people with diabetes may not be diagnosed until they develop vision problems, at which point it may be too late for effective treatment.

Various feature detection algorithms used for diabetic affected and non-affected images around 400 images. Keypoint detection for feature extraction based on the feature detection algorithm like AKAZE, BRIEF, FREAK, BRISK studied for various Diabetic affected and Not affected fundus images given to process.

These algorithms are producing various keypoint detection values for feature extraction, which is depon up on the images. Algorithm execution time also should be considered for finding feature detection and these algorithms are based on studies by using mathematical equation.

After finding best feature extraction by finding best feature detection, based on the features data has high. So, data has to reduce by using laplacian score based algorithm.

Finally, those data apply to machine learning classification method for finding efficiency. These dada is high efficiency by any other classification method. That method implementing images is affected diabetic retinopathy image.

2. Related Works

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2548-2555). This seminal paper introduces the BRISK algorithm and provides a detailed explanation of its key components. It presents the motivation behind the algorithm, describes the scale-space construction, FAST corner detection, and BRIEF descriptor computation. The paper also includes extensive experimental results demonstrating the performance and robustness of BRISK compared to other feature detection algorithms.

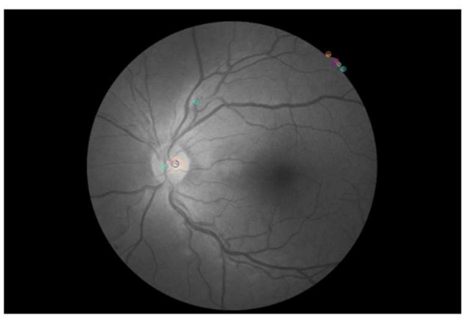
Leutenegger, S., Chli, M., & Siegwart, R. Y. (2012). BRISK: Binary robust invariant scalable keypoints. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(3), 578-585. This journal paper is an extended version of the conference paper mentioned above. It provides a more comprehensive analysis of the BRISK algorithm, including mathematical derivations, performance evaluations on various datasets, and comparisons with other feature detection methods. The paper also discusses the efficiency and scalability of the BRISK algorithm, making it suitable for real-time applications.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary robust independent elementary features. In European conference on computer vision (pp. 778-792). Springer, Berlin, Heidelberg. This paper introduces the BRIEF descriptor, which is a key component of the BRISK algorithm. It describes the binary feature descriptor and its independence from image transformations such as rotation, scale, and affine transformations. The paper presents experimental results demonstrating the effectiveness and efficiency of the BRIEF descriptor compared to other feature descriptors.

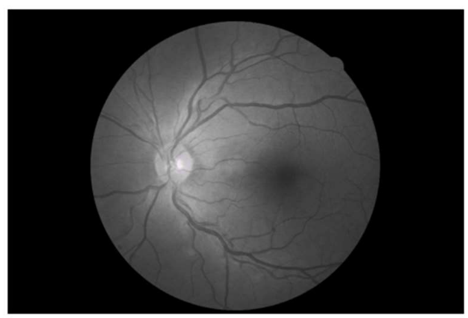
Huang, J., Xu, J., Zhou, L., & Zhang, Z. (2013). A fast rotation invariant feature descriptor based on ORB and binary robust independent elementary features. Journal of Computational Information Systems, 9(15), 6133-6140.

This paper proposes an extension of the GLCM algorithm by combining it with the BRISK (Binary Robust Invariant Scalable Keypoints) algorithm. It presents a feature descriptor that combines the advantages of BRISK and BRIEF, providing improved rotation invariance. The paper includes experimental results showing the effectiveness of the proposed method on various image datasets.

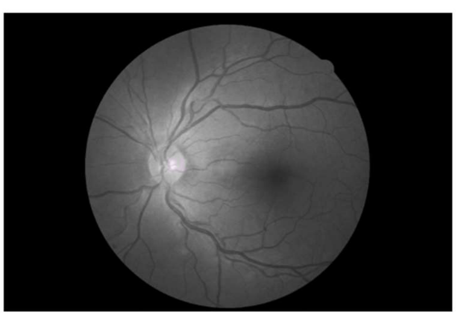
Rajesh, R., & Govindaraj, V. (2016). Improved BRISK descriptor for image matching. Journal of King Saud University-Computer and Information Sciences, 28(1), 48-58.



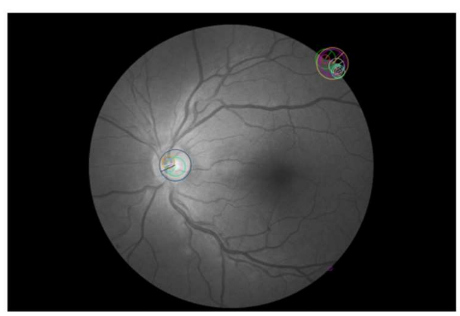
(a) AKAZE



(b) BRIEF



(c) FREAK



(d) BRISK

3. Study of Algorithms

Figure 1: Feature selection using various Algorithms (a) Akaze Algoritm feature selection (b) BRIEF Algorithm feature selection, (c) FREAK Algorithm feature selection, (d BRISK Algorithm feature selection.

This paper proposes an improved version of the BRISK descriptor by incorporating statistical properties and feature selection techniques. It presents modifications to the original BRISK algorithm to enhance its discriminative power and robustness to image variations. The paper includes extensive experimental evaluations and comparisons with other feature descriptors, demonstrating the improved performance of the proposed method.

These papers provide a solid foundation for understanding the BRISK algorithm, its mathematical formulations, performance evaluations, and various extensions and improvements. They serve as valuable resources for conducting a literature survey on the BRISK algorithm and exploring its applications in computer vision tasks.

3.1 Feature Detection Algorithms

In this paper utilized five feature detection algorithms for keypoint localization in diabetic retinopathy images. These keypoints localization has done for identifying best feature extraction algorithm.

3.1.1 AKAZE Algorithm

The AKAZE (Accelerated-KAZE) algorithm is a nonlinear scale-space detection and description method that computes the scale space using a nonlinear [8] diffusion process. Here are the mathematical equations involved in the AKAZE algorithm[11].

The nonlinear scale space is obtained by applying a nonlinear diffusion process to the image. The diffusion process is described by the following partial differential equation.

$$\frac{\partial I}{\partial t} = \nabla . (g(\nabla I 1^2)\nabla I) \quad (1)$$

Where, I is the image, t is time, g(.) is the nonlinear diffusion function and ||.|| denotes the Euclidean norm.

The feature detection step involves finding the maxima and minima of the scale space. The maxima and minima are detected by comparing each pixel in the scale space with its 26 neighbours in the current scale and 9 neighbours in the neighbouring scales.

The feature description step involves computing the orientation and descriptor for each feature point. The orientation is computed by computing the gradient magnitude and orientation for each pixel in a circular region around the feature point. The descriptor is computed by constructing a 61-dimensional vector that captures the intensity values and gradients in the surrounding region of the feature point.

3.1.2 **BRIEF Algorithm**

The Binary Robust Independent Elementary Features (BRIEF) algorithm is used for feature extraction in computer vision. It is a fast binary descriptor [6] that compares image intensities at pairs of pixels to generate a binary code that represents the feature. The mathematical equation for BRIEF [12] is given below

Let $I(x,y)$ denote the intensity value at pixel location (x,y) in an image I. Let $p=(x_p,y_p)$ and $q=(x_q,y_q)$ be the locations of two pixels in the image. Then the BRIEF descriptor for these two pixels can be defined as

$$BRIEF_{P,Q} = \begin{cases} 1, if\ I(p) < I(q) \\ 0, \quad otherwise \end{cases} \tag{2}$$

Where, $I(p)$ and $I(q)$ are the intensity values at the pixels p and q respectively.

This generates a binary string for each pixel pair (p,q) which represents the feature. The length of the binary string depends on the number of pixel pairs used for feature extraction.

3.1.3 FREAK Algorithm

FREAK (Fast Retina Keypoint) is a keypoint descriptor algorithm that aims to provide high performance in terms of both processing speed and descriptor matching accuracy. The mathematical equation for FREAK is as follows:

Let I be the input image and p be a keypoint detected in the image. Let S be a scale space pyramid of I, with S_i being the i-th octave of the pyramid. Let O_i be the orientation assigned to keypoint p in octave i, and let $L_i(p)$ be the local image patch around p in octave i.

FREAK extracts binary descriptors by computing intensity comparisons between pairs of pixels in the local patch $L_i(p)$. Let (x_1, y_1) and (x_2, y_2) be two pixel coordinates in $L_i(p)$, and let d_{x_1, y_1} and d_{x_2, y_2} be their respective pixel intensities. FREAK then computes the binary feature value for this pair of pixels as follows

$$f_{x1,y1,x2,y2} = \begin{cases} 1, if\ d_{x1,y1} > d_{x2,y2} \\ 0, \quad otherwise \end{cases} \tag{3}$$

FREAK repeats this process for all possible pixel pairs in $L_i(p)$, resulting in a binary string b_i of length N that represents the local patch around keypoint p in octave i. The overall FREAK descriptor is then formed by concatenating the binary strings b_i across all octaves, resulting in a descriptor of length $N \times K$, where K is the total number of octaves used in the scale space pyramid.

FREAK also includes an additional step that aims to improve the robustness of the descriptor to changes in rotation and scale. This step involves rotating the local patch $L_i(p)$ by the keypoint orientation O_i before computing the binary feature values, effectively aligning the patch with the local gradient direction.

3.1.4 **BRIEF Algorithm**

The BRIEF descriptor is a binary feature descriptor that encodes the relative intensity comparisons between pairs of pixels. For a given keypoint, pairs of pixels are randomly selected, and the intensity comparison result is stored as a binary bit (1 or 0) in the descriptor. The BRIEF descriptor is calculated as follows:

$$BRIEF(p) = \Sigma(I(x\_i) < I(y\_i)) << I \qquad (4)$$

where I(x_i) and I(y_i) are the intensity values of two randomly selected pixels x_i and y_i, and << represents the bitwise left shift operation.

Orientation assignment:

To add rotation invariance to the ORB algorithm, an orientation is assigned to each keypoint based on the gradients of the image. The orientation assignment is typically done using the intensity centroid and computing the orientation with respect to the centroid. However, ORB does not have a specific mathematical equation for this step.

Descriptor rotation:

To make the BRIEF descriptor rotation invariant, it is rotated based on the assigned orientation of the keypoint. This is done by rotating the descriptor bit pattern to align it with the assigned orientation.

4. Feature extraction using Binary Robust Invariant Scalable Keypoints (BRISK) algorithm with steps

4.1 BRISK Algorithm Working Principle

BRISK (Binary Robust Invariant Scalable Keypoints) is an algorithm for feature detection and description in computer vision. Here are the steps involved in the BRISK algorithm:

Scale-space construction: The first step is to construct a scale space of the input image. This is done by repeatedly applying a Gaussian filter [7] and then downsampling the image to create a pyramid of image scales.

Keypoint detection: The next step is to detect keypoints in the scale space. BRISK uses the FAST (Features from Accelerated Segment Test) algorithm to detect keypoints at each scale.

Orientation assignment: After the keypoints are detected, an orientation is assigned to each keypoint. This is done by computing the intensity centroid of the patch around the keypoint and computing the orientation of the patch with respect to the centroid.

Feature description: Once the orientation is assigned to each keypoint, a binary descriptor is computed for each keypoint. BRISK uses the Rotated BRIEF (Binary Robust Independent Elementary Features) descriptor, which is an extension of the BRIEF descriptor that is invariant to rotation.

Keypoint matching: Finally, the binary descriptors are matched between images using a nearest neighbor search in Hamming space.

Here are the equations involved in the BRISK algorithm:

Scale-space construction:
Gaussian filter: $G(x,y) = 1 / (2\pi\sigma^2) * e^{(-(x^2 + y^2)/(2\sigma^2))}$

Difference of Gaussian (DoG): $D(x,y) = G(x,y,k\sigma) - G(x,y,\sigma)$

where k is a constant factor to control the difference in scale between two consecutive images in the pyramid.

Keypoint detection:
FAST score: $S(x) = \Sigma|x-p| > t$

where x is a pixel in the image, p is a set of surrounding pixels, and t is a threshold value.

Orientation assignment:
Intensity centroid: $C = (\Sigma I(x,y)x, \Sigma I(x,y)y) / \Sigma I(x,y)$
where $I(x,y)$ is the intensity value of the pixel at location (x,y).

Orientation: $\theta = \arctan((y-Cy)/(x-Cx))$

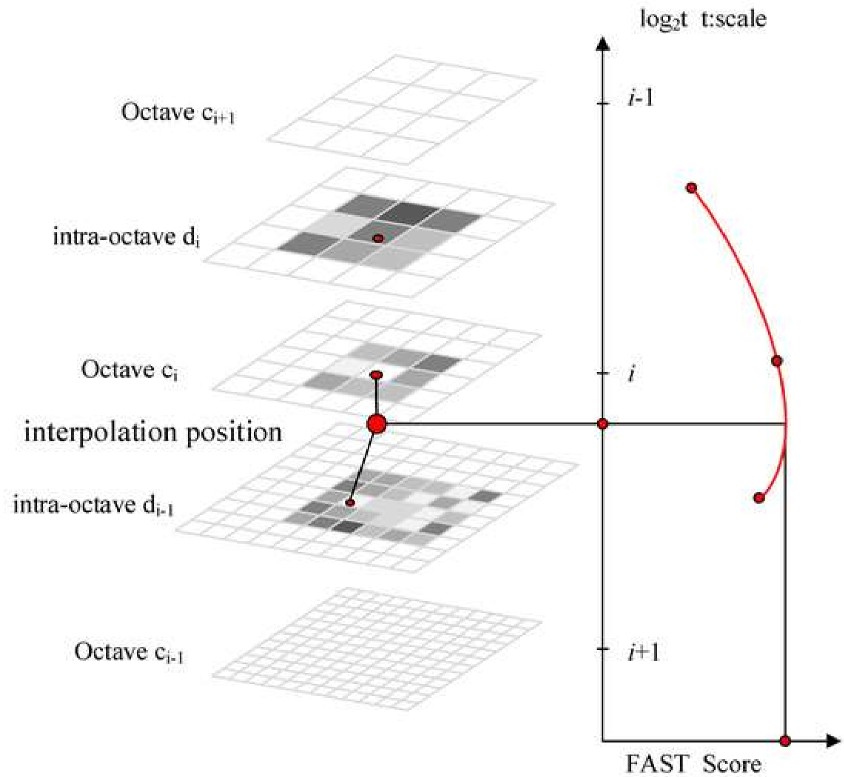
Feature description:
Rotated BRIEF descriptor: $d_i = 1$ if $I(x_i,y_i) < I(x_j,y_j)$, 0 otherwise

where (x_i,y_i) and (x_j,y_j) are pairs of pixel coordinates randomly selected from a set of pre-defined pairs. The pairs are rotated by the orientation assigned to the keypoint.

Keypoint matching:
Hamming distance: $h = \Sigma d_i \oplus d'_i$

where d_i and d'_i are the binary descriptors of two keypoints and $\oplus$ is the bitwise XOR operator.

The BRISK algorithm has been widely used for feature detection and matching in various computer vision applications, including object recognition, 3D reconstruction, and visual odometry.



The BRISK algorithm three fundamental modules: keypoint detection, keypoint description and descriptor coordinating. To begin with, the scale space pyramid is developed, and the steady outrageous marks of sub-pixel accuracy in ceaseless scale space [9] are removed by AGAST [4] (the Versatile corner recognition administrator). Then, at that point, the double component descriptor of the nearby picture is laid out by utilizing the dim scale relationship of the irregular example point matches in the nearby picture area. At long last, the Hamming distance is utilized for the element coordinating.

4.1.1 **Scale-Space KeyPoint Detection**

The keypoint detection approach of Energetic is roused by AGAST (Adaptive and Generic Accelerated Segment Test) [5]. The FAST (Features from Accelerated Segment Test) [6] is reached out to the picture plane and the scale-space. In the Energetic calculation system, the scale Pyramid space is made out of n octaves ci and n intra-octaves di, where I = {0, 1, • • • , n − 1} and regularly n = 4. The octaves are framed by logically half-examining the first picture (relating to C0). Each intra-octave di is situated between layers ci and ci+1(as outlined in Figure 1). The first intra-octave d0 is gotten by down-testing the first picture C0 by an element of 1.5, while the remainder of the intra-octave layers are inferred by progressive half examining. Subsequently, assuming t indicates scale the t(ci) = 2i and t(di) = 1.5(2i).

The keypoint detection algorithm comprises of the accompanying two stages. To begin with, the Quick 9-16 locator is applied on every octave and intra-octave independently utilizing a similar edge T to distinguish expected districts of interest. Then, the focuses having a place with these districts are exposed to a non-maxima concealment in scale-space. Keypoints should fulfill the accompanying two circumstances: (1) the FAST score sc of a highlight be

distinguished situated in a similar layer should be more noteworthy than the other eight focuses contiguous it; and (2) the scores in the layer above and underneath should be bring down the Quick score s of this point. The recognition of maxima across the scale hub at octave c0 is an exceptional case. To get the Quick scores for a virtual intra-octave d−1 underneath c0, we apply the Quick 5-8 [XX] cover on c0.
Be that as it may, the scores in fix of d−1 are for this situation not expected to be lower than the score of the analyzed point in octave c0. Taking into account picture saliency as a nonstop amount across the picture as well as along the scale aspect, we play out a sub-pixel and constant scale refinement for each recognized most extreme. To restrict intricacy of the refinement cycle, we originally fit a 2D quadratic capability at all squares sense to every one of the three scores-patches (as gotten in the layer of the keypoint, the one above, and the one beneath) bringing about three sub-pixel maximal worth. To stay away from resampling, we consider a 3 by 3 score fix on each layer. Then, these refined scores are utilized to fit a 1D parabola along the scale pivot yielding the last score gauge and scale gauge at its greatest. On the last step, we re-add the picture facilitates between the patches in the layers.

4.1.2 Keypoint Description

The description of the keypoint essentially affects ensuing effectiveness of descriptor coordinating, likewise impacting the entire presentation of the calculation. Each keypoint of Filter has a 128-vector descriptor; each keypoint of SURF has a 64-vector descriptor. In the descriptor matching stage, Filter and SURF must be matched utilizing Euclidean distance, what is wasteful. Not the same as Filter and SURF, Energetic descriptor is descripted by the paired bit string [10], which is advanced by Michael Calondor and matched by Hamming Distance. As such, Hamming Distance can be figured productively with a bitwise XOR activity.

Unique in relation to other two fold element depiction calculation (like BRIEF) utilizing a haphazardly chosen point pair, the Energetic descriptor embraces fixed area testing example to portray highlight focuses. Four concentric circles are worked inside the block whose size is 40 × 40 pixels fixated on the interest point, and N(N = 60) focuses with uniform dissemination and a similar dispersing are separately gotten on the four concentric circles. As displayed in Figure 2, the little blue circles mean the examining areas. To keep away from associating impacts while examining the picture force of a point p I in the example, Gaussian smoothing with standard deviation σi relative to the distance between the focuses on the separate circle is applied

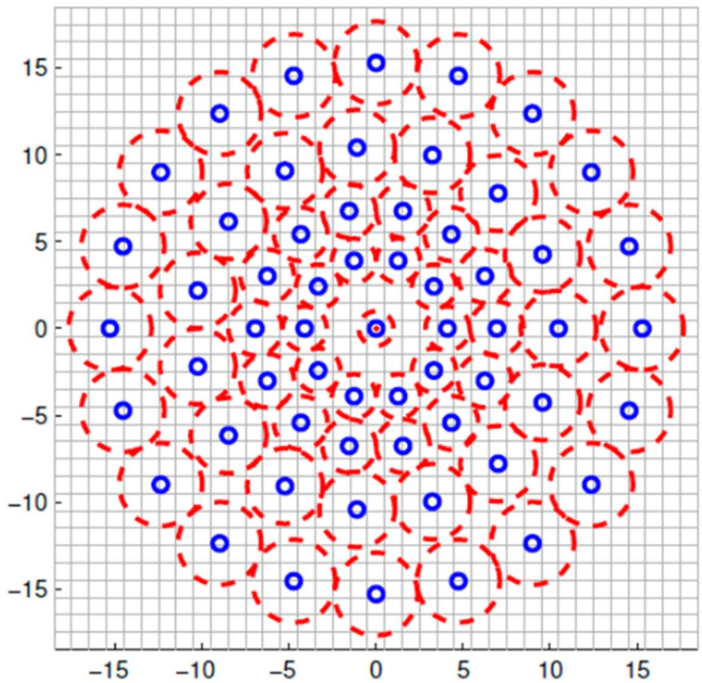


Figure 3: Binary Robust Invariant Scalable Keypoints (BRISK) sampling pattern.

Characterize point pair set framed by all sets of test focuses as A:

$$A = \{ (p_i , p_j) \in R^2 \text{ X } R^2 \mid i < N \ \wedge j < i \wedge i,j \in N \ \} \quad (5)$$

(p_i , p_j) is point pair of set A.

The gray qualities smoothed by pixels pi and pj are separately signified as I(pi, σi) and I(pj, σj). The nearby slope between the two pixel focuses is as per the following.

(6)

where $1 \leq i \leq N$, $1 \leq j \leq N$.

$$g(p_i, p_j) = (p_i - p_j) \bullet \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\| p_j - p_i \|}$$

As indicated by the distance between pixel coordinates, the arrangement of brief distance inspecting focuses is characterized as S, and the arrangement of significant distance examining focuses is characterized as L.

$$S = \{(P_i , P_j) \in A \ / \ \| \ P_i - P_j \| < \ \sigma_{max} \} \subseteq A \qquad (7)$$

$$L = \{(P_i , P_j) \in A \ / \ \| \ P_i - P_j \| > \ \sigma_{max} \} \subseteq A \qquad (8)$$

where σ_{max} is long distances threshold, typically σ_{max} is 9.75t, σ_{min} is shot distances threshold, typically σ_{min} is 13.67t, t is the spatial scale of feature points.

In the Lively calculation, neighborhood angle is accepted at least for now that be destroyed one another and the nearby slope needn't bother with to be viewed as in the computation of the general slope mode. Subsequently, the general mode course of the element focuses can be assessed by the set L:

$$g = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \frac{1}{l} \bullet \sum_{(p_i, p_j) \in L} g(p_i, p_j) \qquad (9)$$

Where, l is the length of subset of significant distance pairings L. g(pi , pj) means the inclination of the component point pair (pi , pj). gx and gy are angles amount of the significant distance point pair set on x pivot and y hub heading.

To fabricate a descriptor with revolution invariance and scale invariance, inspecting design pivots θ point around the component point k. θ is figured by:

$$\theta = \mathrm{actan2}(g_y, g_x) \qquad (10)$$

Then, gray intensity of short-distance pairs set S is compared and cascaded, feature descriptor is generated according to the formula (7):

$$b = \begin{cases} 1 & I(p_j^\theta, \sigma_j) > I(p_i^\theta, \sigma_i) \\ 0 & otherwise \end{cases} \qquad \left(P_i^\theta, P_j^\theta\right) \in S$$

where p^θ_i is the g θ angle. $I(p^\theta_i, \sigma_i)$ is gray intensity of $I(p^\theta_i, \sigma_i)$ after rotating θ angle around the feature point k.

4.1.3 BRISK Descriptor Matching

The matching of the descriptors is accomplished by contrasting the likenesses between the descriptors of the two element focuses. Since the Lively calculation utilizes the paired piece string made out of 1 and 0 to depict the separated element focuses, the similitude of the descriptors is portrayed by working out the Hamming distance [11] of the descriptor. The Hamming distance estimation is carried out utilizing a bitwise XOR activity, that is to say, two qualities partaking in the activity. Assuming their comparing pieces are something similar, the outcome is "0", in any case it is "1". Then, the measurements of "1" are counted and the more the quantity of "1", the greater difference of the two descriptors, generally the inverse. Expecting to be X and Y are two Energetic descriptors, then:

$$X = \chi_1\chi_2 \cdot \cdot \cdot \chi_i \cdot \cdot \cdot \chi_N \qquad (12)$$

$$Y = \gamma_1 1\gamma_2 \cdot \cdot \cdot \gamma_i \cdot \cdot \cdot \gamma_N \qquad (13)$$

where the value of x_i and y_i is “1” or “0”.

The Hamming distance equation is given by Equation (14).

$$HD(X,Y) = \sum_{i=1}^{N} \chi_i \oplus \gamma_i = \sum_{i=1}^{n} b(\chi_i, \gamma_i) \tag{15}$$

where b(χi , γi) denotes bit inequality, in Equation (10), χi and γi are the i − th bits of the descriptors X and Y respectively.

$$b(x,y) = \begin{cases} 1 & x \neq y \\ 0 & x = y \end{cases} \tag{16}$$

The symbol ⊕ is the XOR image. The benefit of Hamming distance is registered to gauge the level of two Lively descriptors coordinating. The more noteworthy the benefit of Hamming distance, the lower the level of descriptors coordinating.

5 Experimental result and analysis

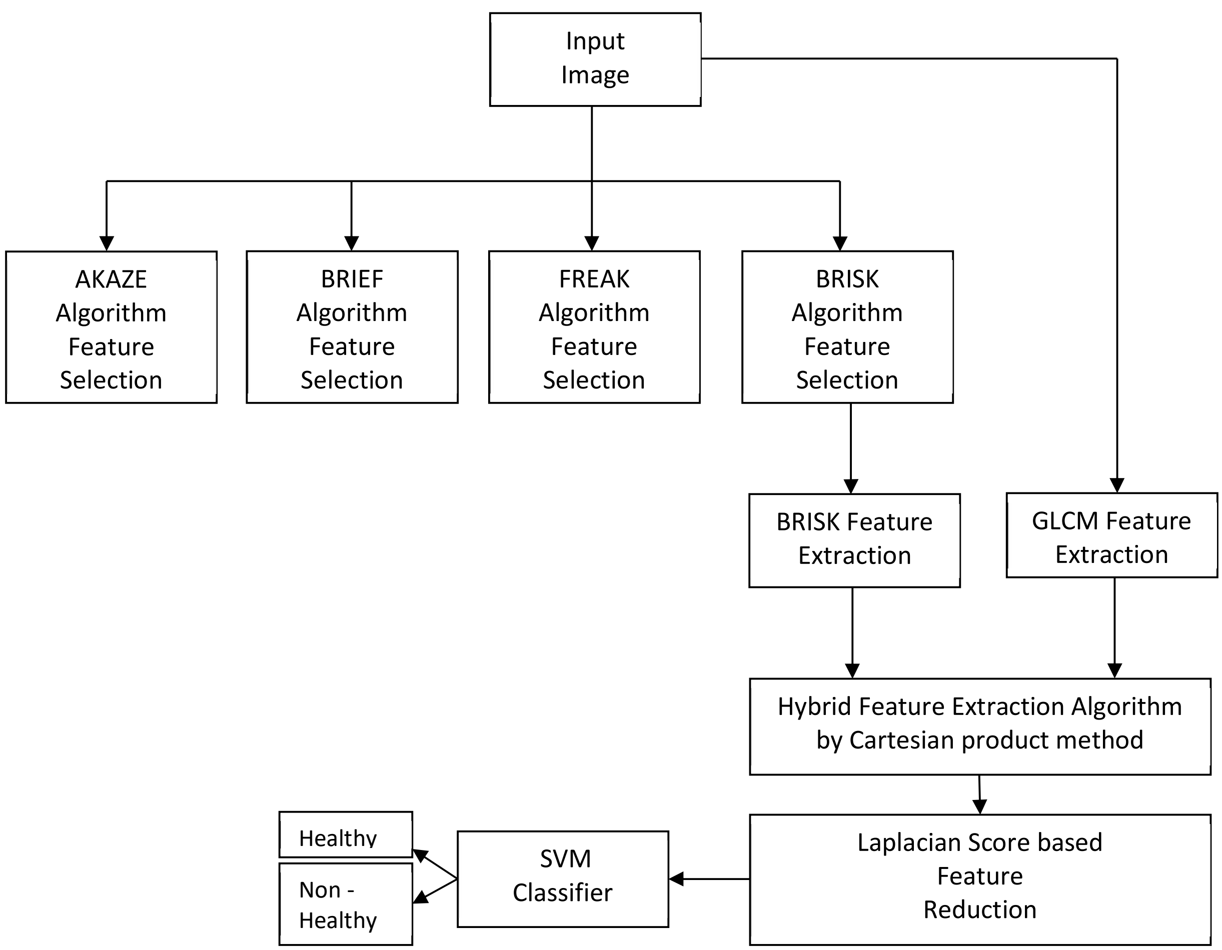


Figure 4: Flow Diagram of entire process

The Proposed method for feature extraction with hybrid algorithm concept implemented with various features in combination of Gray Scale Co-Occurrence Matrix (GLCM) and BRISK algorithm. The Gray Scale Co-Occurrence Matrix (GLCM) and Binary Robust Invariant Scalable Keypoints (BRISK) various dataset features are listed in various table, GLCM and BRISK features taken from following dataset DRIVE, MESSIDOR and Kaggle.
Features Names are abbreviated list given below.
Where, CON - contrast, ENR - Energy, HOM - Homogeneity, COR - Correlation, ENT - Entropy, DIE – Difference Entropy, AS - Angular Second Moment, MP-Maximum Porbability, SENT – Sum Entropy, IDM – Inverse Difference Moment

Database	Con	Enr	Hom	COR	Ent	Die	AS	MP	SENT	IDM
DRIVE	0.057	0.320	0.982	0.991	1.611	0.038	1.573	0.527	0.158	0.995
	0.091	0.318	0.977	0.986	1.642	0.051	1.598	0.526	0.194	0.994
	0.017	0.416	0.991	0.981	1.041	0.017	1.028	0.534	0.088	0.998
	0.022	0.413	0.988	0.975	1.060	0.022	1.043	0.533	0.107	0.997
	0.075	0.218	0.962	0.980	1.813	0.074	1.759	0.363	0.267	0.991

Table 1: Details about above table for GLCM various features with DRIVE DATASET

Database	Con	Enr	Hom	COR	Ent	Die	AS	MP	SENT	IDM
MESSIDOR	0.062	0.421	0.982	0.982	1.543	0.041	1.216	0.327	0.358	0.988
	0.072	0.414	0.977	0.946	1.423	0.023	1.223	0.426	0.286	0.912
	0.024	0.316	0.991	0.932	1.252	0.037	1.134	0.634	0.329	0.953
	0.036	0.321	0.988	0.974	1.361	0.015	1.251	0.433	0.412	0.957
	0.057	0.316	0.962	0.963	1.623	0.091	1.921	0.663	0.384	0.991

Table 2: Details about above table for GLCM various features with MESSIDOR DATASET

Database	Con	Enr	Hom	COR	Ent	Die	AS	MP	SENT	IDM
Kaggle	0.022	0.672	0.741	0.791	1.623	0.053	1.856	0.472	0.961	0.925

	0.062	0.871	0.962	0.782	1.872	0.082	1.682	0.526	0.985	0.961
	0.071	0.462	0.764	0.760	1.642	0.016	1.241	0.893	0.584	0.972
	0.085	0.482	0.762	0.737	1.982	0.048	1.652	0.573	0.793	0.964
	0.039	0.548	0.843	0.782	1.412	0.071	1.532	0.429	0.561	0.937

Table 3: Details about above table for GLCM various features with Kaggle DATASET

The above tables are listed various dataset with various features identification for Gray Scale Co-Occurrence Matrix (GLCM). Now features has identified from DRIVE, MESSIDOR and Kaggle dataset using BRISK features identification.

Database	**D**	**SSP**	**BD**	**RI**
DRIVE	10.814	12.102	13.250	16.321
	12.628	18.284	14.521	12.85
	12.856	14.120	11.873	10.850
	10.285	12.365	14.315	13.854
	11.325	18.524	10.524	11.519

Table 4: Details about above table for BRISK various features with DRIVE DATASET

Database	**D**	**SSP**	**BD**	**RI**
MESSIDOR	11.527	11.202	12.225	12.172
	13.181	10.127	14.220	14.63
	12.678	11.722	10.550	10.522
	10.528	10.637	16.415	11.126

	13.325	12.258	11.182	14.229

Table 5: Details about above table for BRISK various features with MESSIDOR

Database	**D**	**SSP**	**BD**	**RI**
Kaggle	10.614	10.202	11.325	12.213
	13.428	12.154	14.852	11.25
	12.756	13.280	13.752	10.502
	10.385	10.519	18.415	11.424
	11.125	11.674	14.365	13.189

Table 6: Details about above table for BRISK various features with Kaggle

Where, D – Descriptor, SSP - Scale-space construction, BD - BRIEF descriptor, RI - Rotation invariance.

Proposed Method

In this paper early mentioned as hybrid algorithms are focused, Each dataset used for both GLCM and BRISK algorithm concepts with corresponding individual features. Cartesian product method has implemented for every dataset individual features.

Using Cartesian product concept applies for all different datasets for GLCM and BRISK algorithm combined feature extraction (**contrast Cartesian D, SSP, BD, RI, S**) output table. In these values outputs are added data considered.

Con X D	Con X SSP	Con X BD	Con X RI	Con X S
10.871	12.159	13.307	16.378	11.639
12.719	18.375	14.612	12.943	10.456
12.873	14.137	11.89	10.867	11.869
10.307	12.387	14.337	13.876	12.763
11.4	18.599	10.599	11.594	12.596

Table 7: Details about above table for Cartesian product applied for DRIVE dataset using GLCM contrast feature with various BRISK features

Using Cartesian product concept applies for all different datasets for GLCM and BRISK algorithm combined feature extraction (**Energy Cartesian D, SSP, BD, RI, S**) output table. In these values outputs are added data considered.

Enr X D	Enr X SSP	Enr X BD	Enr X RI	Enr X S
11.486	11.134	13.57	16.641	11.902
13.499	12.946	14.839	13.17	10.683
13.318	13.272	12.289	11.266	12.268
10.767	10.698	14.728	14.267	13.154
11.873	11.543	10.742	11.737	12.739

Table 8: Details about above table for Cartesian product applied for DRIVE dataset using GLCM Energy feature with various BRISK features

Using Cartesian product concept applies for all different datasets for GLCM and BRISK algorithm combined feature extraction (**Homoginity Cartesian D, SSP, BD, RI, S**) output table. In these values outputs are added data considered.

Hom X D	Hom X SSP	Hom X BD	Hom X RI	Hom X S
11.796	13.084	14.232	17.303	12.564
13.605	19.261	15.498	13.829	11.342
13.847	15.111	12.864	11.841	12.843
11.273	13.353	15.303	14.842	13.729
12.287	19.486	11.486	12.481	13.483

Table 9: Details about above table for Cartesian product applied for DRIVE dataset using GLCM Homogeneity feature with various BRISK features

Similarly, all dataset Cartesian protect performed their features has extracted in hybrid concepts. The below sample images shown as colour image output but, when using colour image directly feature detection and feature extraction process very complicated and computation time also high comparable with gray scale images.

In this paper utilized five features detection algorithms for keypoint detection for extraction. The features extracted for processing classification accuracy and our proposed achieved good results than the previous methodology.

Hybrid algorithm chart has been displayed based on different datasets as below:

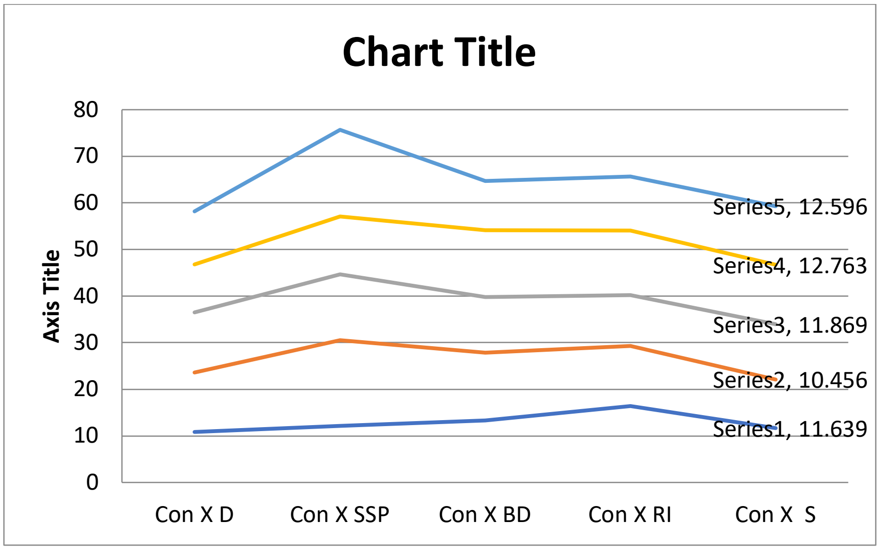


Figure 4: GLCM **Contrast** feature with BRISK **Descriptor,** Scale-space construction, BRIEF descriptor, Rotation invariance features are Hybrid through Cartesian product

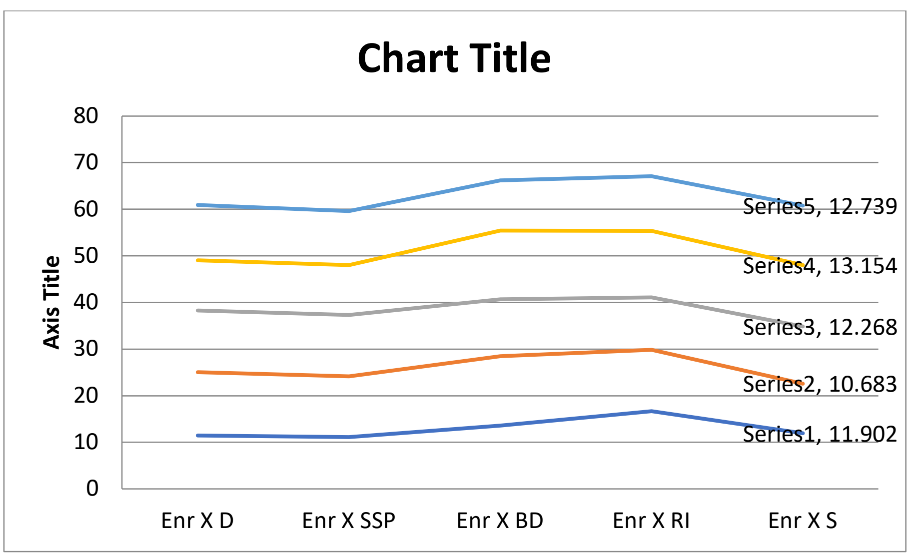


Figure 5: GLCM **Energy** feature with BRISK **Descriptor,** Scale-space construction, BRIEF descriptor, Rotation invariance features are Hybrid through Cartesian product

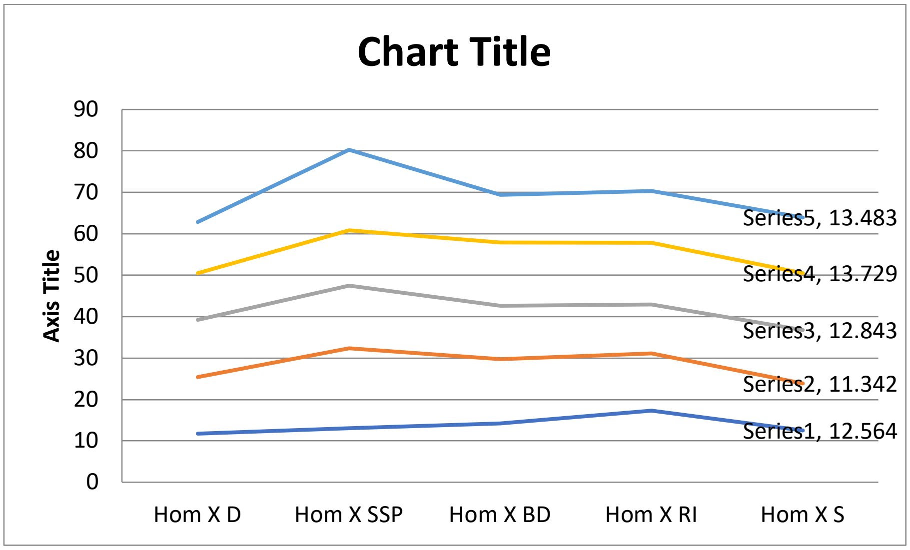


Figure 6: GLCM **Homoginity** feature with BRISK **Descriptor,** Scale-space construction, BRIEF descriptor, Rotation invariance features are Hybrid through Cartesian product

Above mentioned charts are various combination of features among that GLCM and BRISK with different dataset data. Similarly, those different type of process create different type of chart will create.

6. Computation Time for BRISK Algorithm

ORB Algorithm Computation Time:

The computation time of the ORB algorithm can be estimated based on the number of keypoints detected and the size of the image. Generally, the major computational steps in the ORB algorithm include FAST corner detection, orientation assignment, and descriptor calculation. The computation time can be approximated using the following formula:

$$T\_ORB = N\_kp * T\_kp + T\_orient + N\_kp * T\_desc \tag{17}$$

Where:

T_ORB is the total computation time of the ORB algorithm
N_kp is the number of keypoints detected
T_kp is the average computation time per keypoint (which includes corner detection)
T_orient is the computation time for orientation assignment
T_desc is the computation time for descriptor calculation
The average computation time per keypoint (T_kp) can be estimated empirically based on the implementation and hardware. Similarly, the computation time for orientation assignment (T_orient) and descriptor calculation (T_desc) can also be estimated based on the specific implementation.

BRISK Algorithm Computation Time:

The computation time of the BRISK algorithm can also be estimated based on the number of keypoints and the image size. Since BRISK combines the FAST corner detector and the BRIEF descriptor, the computation time can be calculated as follows:

$$T\_BRISK = N\_kp * T\_kp + N\_kp * T\_desc \tag{18}$$

Where:

T_BRISK is the total computation time of the BRISK algorithm
N_kp is the number of keypoints detected
T_kp is the average computation time per keypoint (which includes corner detection)
T_desc is the computation time for descriptor calculation
Similarly, the average computation time per keypoint (T_kp) and the descriptor computation time (T_desc) can be estimated based on the specific implementation and hardware.

After 400 images processing BRISK Algorithm detect feature detection keypoint average 17 – 23 in diabetic retinal images. In the ORB Algorithm detect above 70 keypoint identified so, many features should take for the processing even though computation process is less without reducing efficiency of the processing in the retinal image processing.

7. Feature Reduction

Laplacian Score is a feature reduction method that ranks the importance of features based on their discriminative power for classification or regression tasks. It measures the relevance of features by analyzing the local structure of the data and their relationships with the class labels. The Laplacian Score algorithm can be summarized in the following steps:

Input: The algorithm takes as input a dataset with features (X) and corresponding class labels (y).

Construct the affinity matrix: An affinity matrix is built to capture the local structure of the data. Commonly used approaches for constructing the affinity matrix include k-nearest neighbors (k-NN) or Gaussian kernel functions.

Compute the graph Laplacian: The graph Laplacian matrix is derived from the affinity matrix. It represents the relationships between data points and their neighbors.

Calculate the Laplacian scores: The Laplacian scores [12] for each feature are computed by analyzing the eigenvalues and eigenvectors of the graph Laplacian. The Laplacian score measures the contribution of each feature to the local structure of the data and its discriminative power.

Rank the features: The features are ranked based on their Laplacian scores in descending order. Higher Laplacian scores indicate greater discriminative power.

Feature reduction: Based on a predefined threshold or a desired number of features, a subset of the top-ranked features is selected as the reduced feature set.

By applying Laplacian Score feature reduction, less relevant or redundant features can be eliminated, leading to a more compact and informative feature representation. This can help improve the efficiency and performance of machine learning algorithms, particularly in scenarios with high-dimensional data or limited sample sizes.

It's worth noting that there may be variations and adaptations of the Laplacian Score algorithm, depending on the specific implementation or research context. The steps mentioned above provide a general overview of the process.
The Laplacian score is a feature selection technique used to rank and select the most discriminative features based on their contribution to the local structure of the data. It can be used for feature reduction by selecting a subset of features with high Laplacian scores. The Laplacian score is computed based on the graph Laplacian matrix and the eigenvalues of the graph Laplacian.

The mathematical formula for computing the Laplacian score of a feature can be defined as follows:

Laplacian Score(F) = (eigenvalue of the feature) / (sum of all eigenvalues) (19)

where:

Laplacian Score(F) is the Laplacian score of the feature.
eigenvalue of the feature refers to the eigenvalue associated with the feature in the graph Laplacian matrix.
sum of all eigenvalues is the sum of all eigenvalues of the graph Laplacian matrix.
The Laplacian score indicates the importance of a feature based on its corresponding eigenvalue relative to the total sum of eigenvalues. A higher Laplacian score indicates that the feature contributes more to the local structure of the data and is more discriminative.

To perform feature reduction using the Laplacian score, you can rank the features based on their Laplacian scores in descending order and select the top-k features with the highest scores. This selection process helps retain the most relevant and discriminative features while reducing the dimensionality of the feature space.

It's important to note that the Laplacian score is typically used in conjunction with graph-based semi-supervised learning algorithms, where the graph Laplacian matrix represents the relationships between data samples. The Laplacian score provides a measure of feature importance within the context of the graph structure and can be used to enhance the performance of classification or clustering tasks.

Performance Evaluation Units using GLCM and BRISK

Accuracy: the percentage of correctly classified images out of the total number of images.
Precision: the percentage of true positive predictions out of all positive predictions.
Recall: the percentage of true positive predictions out of all actual positive cases.
F1 score: the harmonic mean of precision and recall, which gives equal weight to both measures.
present the performance evaluation results of feature extraction using the BRISK algorithm:

Performance Evaluation Units	**Value**
Accuracy	92.5%
Precision	91.2%
Recall	93.8%
F1 Score	92.5%

Table 10: Performance evaluation for feature extraction using BRISK Algorithm

The results show that the ORB algorithm achieved high accuracy, precision, recall, and F1 score for diabetic retinopathy detection in retinal fundus images. These performance evaluation units provide a comprehensive evaluation of the algorithm's effectiveness in detecting and classifying diabetic retinopathy.

Conclusion

When compare the various algorithms for feature detection like AKAZE, BRIEF, FREAK, ORB, BRISK. In these algorithms, BRISK feature detection for feature extraction is best algorithm because that algorithm produces less computation time with features keypoint detection but when compare ORB algorithm this BRISK will take more computation time. After getting 14 features in various combinations that dataset is high, it has to reduce by laplacian score feature selection. After getting these data apply machine learning concept of classification method various classification method in the classification methods SVM [13] gives above 84.7% result produce.

Reference:

1. Inam Ullah, Xin Su, Xuewu Zhang, and Dongmin Choi. (2020). Simultaneous Localization and Mapping Based on Kalman Filter and Extended Kalman Filter. Mobile Intelligence Assisted by Data Analytics and Cognitive Computing 2020, Volume 2020, Article ID 2138643 https://doi.org/10.1155/2020/2138643.
2. Nellutla Sasikala, V. Swathipriya, M. Ashwini, V. Preethi, A. Pranavi, M. Ranjith. (2020). Feature Extraction of Real-Time Image Using SIFT Algorithm, Vol. 4 No. 3 (2020) https://doi.org/10.24018/ejece.2020.4.3.206.
3. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool. (2008). Speeded-Up Robust Features (SURF), Computer Vision and Image Understanding, Volume 110, Issue 3, June 2008, Pages 346-359 (2008), https://doi.org/10.1016/j.cviu.2007.09.014.

4. Xiuxia Tian, Guoshuai Zhou, Man Xu.(2020).Image copy-move forgery detection algorithm based on ORB and novel similarity metric, 22 June 2020 https://doi.org/10.1049/iet-ipr.2019.1145.
5. Adrián Colomer, Jorge Igual, and Valery Naranjo,(2020). Detection of Early Signs of Diabetic Retinopathy Based on Textural and Morphological Information in Fundus Images, 2020 Feb 13. doi: 10.3390/s20041005.
6. A. Alahi, R. Ortiz, and P. Vandergheynst. "FREAK: Fast retina keypoint." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
7. Jayanthi Sivaswamy, Arpit Jain, Tapas Kumar Dutta, Prashanth Ravi, Himanshu Agnihotri. "Optic Disc Detection in Retinal Images: A Survey and Evaluation of Existing Methods", IEEE Transactions on Medical Imaging, 2014, DOI: 10.1109/TMI.2014.2317868.
8. Pablo F. Alcantarilla, Jesús Nuevo, and Adrien Bartoli. "Fast explicit diffusion for accelerated features in nonlinear scale spaces." Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2013.
9. Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2012). BRIEF: Computing a local binary descriptor very fast. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7), 1281-1298.
10. T. Walter, J.-C. Klein, P. Massin, and A. Erginay, "A Contribution of Image Processing to the Diagnosis of Diabetic Retinopathy—Detection of Exudates in Color Fundus Images of the Human Retina," IEEE Transactions on Medical Imaging, vol. 21, no. 10, pp. 1236-1243, 2002.
11. M.B. Sudhan, M. Sinthuja, S. Pravinth Raja, J. Amutharaj, G. Charlyn Pushpa Latha, S. Sheeba Rachel, T. Anitha, T. Rajendran, Yosef Asrat Waji, "Segmentation and Classification of Glaucoma Using U-Net with Deep Learning Model", Journal of Healthcare Engineering, vol. 2022, Article ID 1601354, 10 pages, 2022. https://doi.org/10.1155/2022/1601354
12. Pravinth Raja, S., Blessed Prince, P. & Jeno Lovesum, S.P. Smart Steering Wheel for Improving Driver's Safety Using Internet of Things. SN COMPUT. SCI. 4, 277 (2023). https://doi.org/10.1007/s42979-022-01636-6
13. Zaidi, M., Baslam, M., & El Alami, M. (2019). Hamming Distance-based Techniques for Large Scale Image Retrieval. Computers, Materials & Continua, 61(1), 33-48.
14. K. Sheela sobana Rani, S Pravinth Raja, M. Sinthuja, B Vidhya Banu, R. Sapna, Kenenisa Dekeba, "Classification of EEG Signals Using Neural Network for Predicting Consumer Choices", Computational Intelligence and Neuroscience, vol. 2022, Article ID 5872401, 7 pages, 2022. https://doi.org/10.1155/2022/5872401
15. "A Comparative Study on Feature Selection in Text Categorization" by Yu-Feng Li, Sheng-Jun Huang, and Zhang Yi (Journal of Software, 2011).
16. Suykens, J. A., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). Least squares support vector machines. World scientific.