Frontiers in Health Informatics
ISSN-Online: 2676-7104
www.healthinformaticsjournal.com
2024; Vol 13: Issue 3
Open Access

Diabetic Retinopathy Detection Using MSLBP, Harris Corner Detection, and SVM Classifier

Uttam Jadhav[1], Dr. Nirupma Tiwari[2]

[1]*Ph.D. Scholar, SAGE University, Indore, India*
[2]*Associate Professor, Computer Science & Engineering, SAGE University, Indore, India*
Email: uttamjadhav29@gmail.com, nirupma.tiwari1974@gmail.com

Cite this paper as: Uttam Jadhav,Nirupma Tiwari (2024) Diabetic Retinopathy Detection Using MSLBP, Harris Corner Detection, and SVM Classifier. *Frontiers in Health Informatics, 13* (3), 9992-10007

Abstract – *Diabetic retinopathy, a condition affecting diabetic patients, involves the formation of clots, lesions, or hemorrhages in the retina's light-sensitive area. Elevated blood sugar levels lead to vessel blockages and the growth of new vessels, forming mesh-like structures. Accurate evaluation of the retinal vasculature is essential for effective diagnosis. In assessing diabetic retinopathy, fundus scans undergo preprocessing and segmentation, including image enhancement, retinal mask extraction, blood vessel segmentation, optic disk extraction, and lesion candidate region extraction. Branching blood vessels are extracted using thresholding, followed by adaptive histogram equalization and morphological opening to enhance images and remove falsely segmented areas. The proliferation of optical nerves is notably higher in diabetic patients. Additional features are extracted from lesion candidates using a hybrid approach of Harris corner detection and Multi-Scale Local Binary Pattern (MSLBP). A random forest classifier is employed to classify diabetic retinopathy presence. Performance evaluation is conducted on two datasets: DIARETDB1, a standard dataset, and a medical institution's dataset comprising fundus scans of both affected and normal retinas. Experimental results demonstrate the proposed algorithm's superiority over traditional methods, achieving 98.7% accuracy and 97.2% precision on the DIARETDB1 dataset.*

Keywords *– MSLBP, DIARETB1, Harris corner and SVM etc.*

I. INTRODUCTION

Diabetic Retinopathy (DR) is a medical condition affecting individuals with diabetes, categorized into Non-Proliferative Diabetic Retinopathy (NPDR) and Proliferative Diabetic Retinopathy (PDR). Initial signs include the presence of exudates, indicating NPDR. As NPDR progresses, the retina develops new blood vessels, prone to leaking and leading to blood clots or blobs. Damage to the retinal vessel network is crucial in DR development, with advanced PDR stages leading to vessel blockage and lesion formation, notably microaneurysms and hemorrhages. Microaneurysms are often the first observable symptoms of DR, appearing as small round-shaped red dots in the fundus.

Currently, DR detection primarily relies on manual assessment by trained ophthalmologists, highlighting the need for automated screening systems. Various methods, including deep-learning-based algorithms like Convolutional Neural Networks (CNNs), have been developed for accurate detection, particularly in fovea and exudate detection.

In DR analysis, methods like WELR and AMT have proven effective in retinal vessel extraction, while challenges in premature infants' retinal vessel network extraction have spurred research. Techniques like centerline and bit plane identification, along with morphological approaches, contribute to vessel extraction. For exudate detection, methods like morphological compact tree (MCT) and histogram analysis have shown

promise.
These methods aid in accurate recognition and analysis of DR features, advancing diagnosis and monitoring. However, further research is needed to ensure reliability in clinical settings.

These various methods and techniques contribute to the accurate recognition and analysis of diabetic retinopathy features, including retinal vessels, exudates, and other lesions. Researchers continue to explore new approaches and algorithms to enhance the effectiveness and efficiency of diabetic retinopathy diagnosis and monitoring.
These automated systems aim to improve the efficiency and accuracy of DR diagnosis, assisting healthcare professionals in identifying and managing the condition promptly. However, it's important to note that while these methods show promise, further research and validation are necessary to ensure their reliability in real-world clinical settings.
The proposed algorithm offers a comprehensive approach to assessing diabetic retinopathy (DR) through fundus scans, providing several key contributions:
Preprocessing and Segmentation: The algorithm conducts essential pre-processing steps, including image enhancement, retinal mask extraction, blood vessel segmentation, optic disk extraction, and lesion candidate region extraction. These steps isolate and extract relevant regions from the fundus scans.
Branching Blood Vessel Extraction: Employing a thresholding technique, the algorithm extracts branching blood vessels crucial for DR assessment, considering the proliferation of optical nerves observed in diabetic or affected patients.
Image Enhancement and False Segmentation Elimination: Adaptive histogram equalization and morphological opening techniques enhance the fundus scans' quality while eliminating falsely segmented regions, enhancing subsequent analysis and classification accuracy.
Feature Extraction: Further features are extracted from lesion candidate regions using a hybrid approach of Harris corner and MSLBP. This combination captures relevant texture and corner-based features associated with diabetic retinopathy.
Classification using SVM: A Support Vector Machine (SVM) classifier is employed to classify the presence of diabetic retinopathy based on the extracted features. SVMs handle complex datasets effectively, providing robust classification results.
By integrating these contributions, the algorithm offers an effective solution for DR assessment, combining image processing techniques, feature extraction methods, and a powerful classification model to achieve high accuracy and precision in diagnosing DR using fundus scans.
The study follows a structured approach, including a comprehensive literature review in Section II, presentation of materials and methods in Section III, detailed explanation of proposed methods in Section IV, analysis of MATLAB-based simulation results in Section V, and concludes with a summary of findings and conclusions in Section VI.

II. LITERATURE REVIEW

Over the past few decades, significant research efforts have been dedicated to effectively segmenting retinal blood vessels and categorizing retinal images based on the severity of diabetic retinopathy (DR) [10].
In traditional DR analysis, an automatic retinopathy classification approach based on Artificial Neural Networks (ANN) has been proposed [12]. This technique utilizes morphological operations to distinguish between exudates and blood vessels. Techniques such as Genetic Algorithms (GA) and Fuzzy C Means (FCM) have been employed to achieve maximum accuracy. Fuzzy Logic (FL) has shown excellent sensitivity in predicting hard exudates in DR images. Retinal vascular feature analysis has been performed using a multi-scale line detection device [11]. An algorithm named DR analysis with the utilization of Machine Learning (DREAM) has been developed, incorporating a Gaussian mixture model and nearest neighborhood approach with a

classifier based on Singular Vector Machine (SVM) [5]. Global thresholding has been applied for pre-processing retinal images [6]. Morphological component analysis (MCA) has been used for vessel segmentation in the DRIVE and STARE datasets to improve accuracy [7]. Zago et al. [13] utilized the VGG16 model [14] to detect red lesion patches in diabetic retinopathy images and achieved promising results. They classified the images as either having diabetic retinopathy or not based on the detected red lesions, achieving an impressive AUC of 0.912 on the Messidor dataset [15]. Li et al. [16] introduced the DDR dataset, which aimed to classify images into five stages of diabetic retinopathy and localize lesions within the images. They employed the SE-BN-Inception model [17] for stage classification and achieved the highest accuracy of 82.84%. For lesion localization, they utilized Faster RCNN [18] and attained a mean Average Precision (mAP) of 9.2.

Wang et al. [19] adopted a modified version of RFCN [50] to detect the stages of diabetic retinopathy and localize specific features such as microaneurysms (MA) and hemorrhages (HM). They used two RFCN models and merged the results obtained. In their private dataset, their method achieved a high mAP of 92.15 for localization. In terms of classification, they obtained an accuracy of 92.95% [20]. These approaches demonstrate the utilization of various techniques, such as ANN, morphological operations, GA, FCM, FL, SVM, and MCA, in the analysis and classification of retinal images for DR diagnosis. Each method focuses on different aspects, such as vessel segmentation, exudate detection, and tortuosity analysis, aiming to improve the accuracy and efficiency of DR assessment. However, it's important to note that the mentioned studies represent a subset of the extensive research conducted in this field, and further advancements and refinements continue to be explored.

Limitations:

Despite the advancements in vessel segmentation techniques, accurately segmenting retinal vessels in images with low contrast, noise, or artifacts can still be challenging.

The proposed methods often rely on specific datasets, and their performance may vary when applied to different datasets with variations in image quality, resolution, and characteristics of diabetic retinopathy cases.

The interpretation of DR severity solely based on vessel analysis may overlook other important features and indicators present in retinal images, such as microaneurysms, hemorrhages, or exudates.

III. MATERIAL AND METHODS

3.1 Harris Corner Detection

Harris corner detection is a corner detection technique that is commonly used in computer vision and image processing. The technique is intended to detect "corners" in an image, which are unique spots or portions of an image with a substantial shift in intensity in many directions. Corners can help with image analysis tasks including object detection, picture registration, and tracking.

The Harris corner detection algorithm works by calculating the Harris response for each pixel in an image, which measures the change in intensity in two perpendicular directions. The Harris response is calculated as the product of the covariance matrix of the gradient of the image intensity at a particular pixel. The corners in an image are identified as the pixels with the highest Harris response values.

This algorithm is considered to be a robust and reliable algorithm, and it has been widely used in various applications such as object tracking, image stitching, and video surveillance. However, the algorithm can be computationally expensive, and it can be sensitive to noise and other image artifacts, which may result in false detections or missed corners. To overcome these limitations, various modifications and extensions of the Harris corner detection algorithm have been proposed, including the use of scale-invariant feature transform (SIFT) and speeded up robust features (SURF).

The Harris detector looks for the difference in intensity for a displacement of (u, v) in all directions in the image I. This difference is expressed by:

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{windowfunction} \times \left[\underbrace{I(x+u, y+v)}_{Shiftedint} - \underbrace{I(x,y)}_{intensity}\right]^2$$

(1)

To detect corners in an image, we apply a window function that assigns weights to the pixels under consideration. The two commonly used window functions are the rectangular window and the circular Gaussian window, where the latter reduces noise in the detector response. To maximize the corner detection function $E(u, v)$, we need to maximize the second term. We can achieve this by applying Taylor expansion with derivations and obtaining the equation (2):

$$E(u,v) \approx [u \quad v] \times M \times \begin{bmatrix} u \\ v \end{bmatrix} orM = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

(2)

M represents the matrix of second-order moments (the autocorrelation matrix). Here, I_x and I_y are the partial derivatives of the image in the horizontal X direction and the vertical Y direction respectively.

Equation (1) governs the intensity variation and therefore makes the Harris detector invariant to rotation and to small transformations such as corner displacement; this determines the variation of the intensity undergone by a pixel which is analytically developed in the vicinity of the origin of the displacement in order to cover all the small possible displacements. On the other hand, this detector is sensitive to noise because of the use of gradient information.

A. Multi-Scale Local Binary Pattern (MSLBP)

MSLBP is a variant of the Local Binary Pattern (LBP) method that provides a multi-scale representation of the texture information in an image. MSLBP is a multi-scale extension of the traditional LBP method, where the texture feature of a pixel is calculated based on the comparison of the intensity of the central pixel with the intensity of its neighboring pixels.

In MSLBP, the texture feature is calculated at multiple scales by using different sizes of the neighborhood around the central pixel. The texture features at each scale are then concatenated to form a multi-scale texture feature representation of the image. By using different scales, MSLBP can capture the texture information at multiple levels of detail, which can be useful for character recognition. MSLBP has been applied to various image processing tasks, such as texture classification, face recognition, and character recognition. MSLBP provides a rich and robust representation of the texture information in an image, which can be useful for character recognition. By computing the MSLBP features from character images, one can extract information about the local texture and structure of the characters at multiple scales, which can be used to distinguish between different characters and improve the recognition accuracy.

The use of more than one LBP extraction in an image in order to vary the size of the regions (R) or the number of samples (P). The authors justified this by the fact that one of the biggest limitations of the traditional LBP is the small area taken for the calculations $(3 \times 3$ region). According to these authors, this does not make the operator robust with respect to local changes in the image texture caused, for example, by rotation or lighting variation.

In this way, the LBP method was extended to a multiscalar version characterized by the change in the radius size (R) in each extraction.

$$LBP_{PR} = \sum_{n=0}^{P-1} s(g_n - g_c) \times 2^n$$

(11)

Where P is the number of samples (pixels $g_0, g_1, \dots, g_{P-1}$) equally spaced at a distance R from the central pixel g_c, contained in its neighborhood. Figure 2 shows some regions across this data variation.

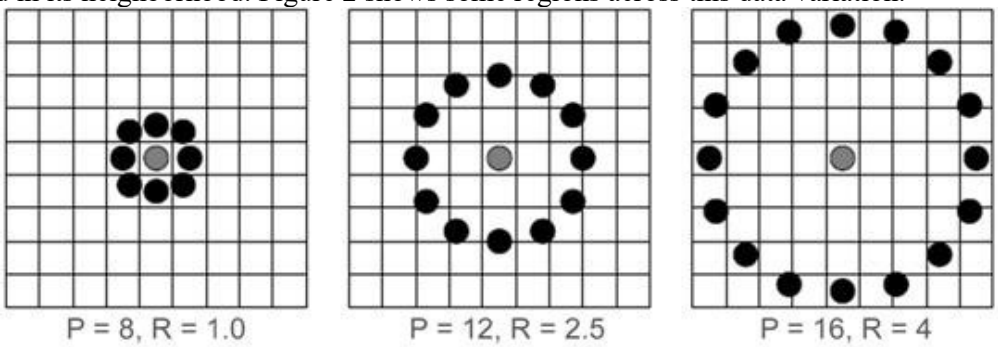


Figure 2: Circular region for three different values of P and R

Jia et al., 2014 presented two variations of the Multiscale Local Binary Patterns (MSLBP) method. The first case, called MSLBP1, uses LBP by extracting 8 samples from different regions of the image, after Gaussian filtering, defined by a radius R_n, where n is the respective scale and is denoted by LBP_{8,R_n} . The second type, called MSLBP2, is an alternative way to the first in which a set of average filters is applied in the region before the extraction of the $LBP_{8,1}$ pattern.

The method used here is entirely based on MSLBP1, in which a Gaussian filter of size G_n is applied to each scale, with the evolution of the scales given by changing the values of the radii. This change occurs respecting the optimal distribution in relation to the application of Gaussian filters. The objective of using low-pass Gaussian filters is so that in each sample in the neighborhood a larger amount of information can be collected. than the one with a single pixel. The circles represent the area for extraction of the LBP standard around each sample after filtering. The outer radius of the "useful area", r_n, is used to calculate the radius R_n for extracting the LBP_{P_n,R_n} and is given by:

$$r_n = r_{n-1} \cdot \left(\frac{2}{1 - \sin\left(\frac{\pi}{P_n}\right)} - 1 \right), \quad n \in \{2, \dots, N\}$$

(12)

Where N is the number of scales and P_n is the sampling of each scale, which in the case of the MSLBP was considered $P_n = 8 \; \forall n \in N$.

The radius r_n establishes the distance between the pixel and the edge of a neighborhood, so r_1 is defined as 1.5, which is the smallest distance between the pixel and the edge of a 3×3 region. Consequently, we define the radius of operator LBP_{8,R_n}, on scale $n(n \geq 2)$, as the midpoint between r_n and r_{n-1}, like this:

$$R_n = \frac{r_n + r_{n-1}}{2}, \quad n \in \{2, \dots, N\}$$

(13)

Analogously to the case of r_n, the radii R_n of the operators represent the distance between the central pixel and the pixels in its neighborhood, therefore R_1 is defined as 1 because it is the smallest distance between the central pixel and those in its neighborhood 3×3. The r_n are used to calculate the window size of Gaussian filters, G_n, on the n scale as shown in Equation (14):

$$G_n = 2.round\left(\frac{r_n - r_{n-1}}{2}\right) + 1$$

(14)

In which the function round: $\mathbb{R} \to \mathbb{N}^+$ assigns the smallest non-negative integer greater than it to a value. At each level, the δ_n value needed to apply the Gaussian filter is given by:

$$\delta_n = \frac{G_n}{\sqrt{-2\ln(1-p)}}$$

(15)

Where p was set to 0.95.

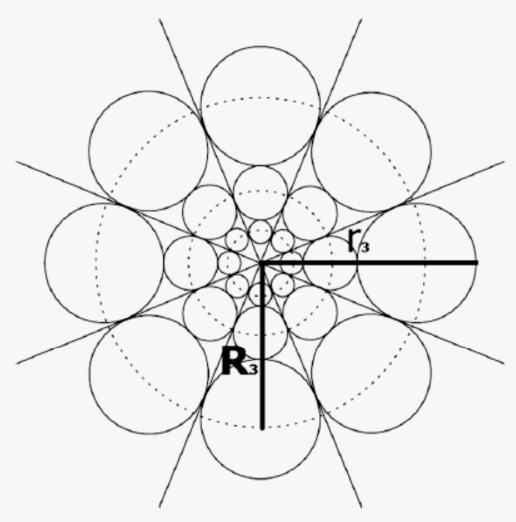


Figure 3: Optimal distribution of the radii of the MSLBP pattern with respect to the redundancy of the information captured by the pattern

IV. PROPOSED METHOD

In the context of diabetic retinopathy, after detecting the lesions using the GLCM and Shi-Tomasi Corner detector, feature extraction techniques are applied to the detected lesions to capture relevant information for pattern generation using a Random Forest classifier. Here's an explanation of the feature extraction process:

- ***Lesion Detection:*** The first step is to detect the lesions in the retinal images using the preprocessing techniques. These algorithms identify potential lesion locations based on corner points, which are areas of interest that indicate the presence of lesions.
- ***Lesion Segmentation:*** Once the lesions are detected, the next step is to segment or isolate them from the surrounding healthy retinal tissue. This can be done using various segmentation techniques, such as thresholding, region growing, or active contour models, to create lesion masks or binary images that indicate the extent of the lesions.

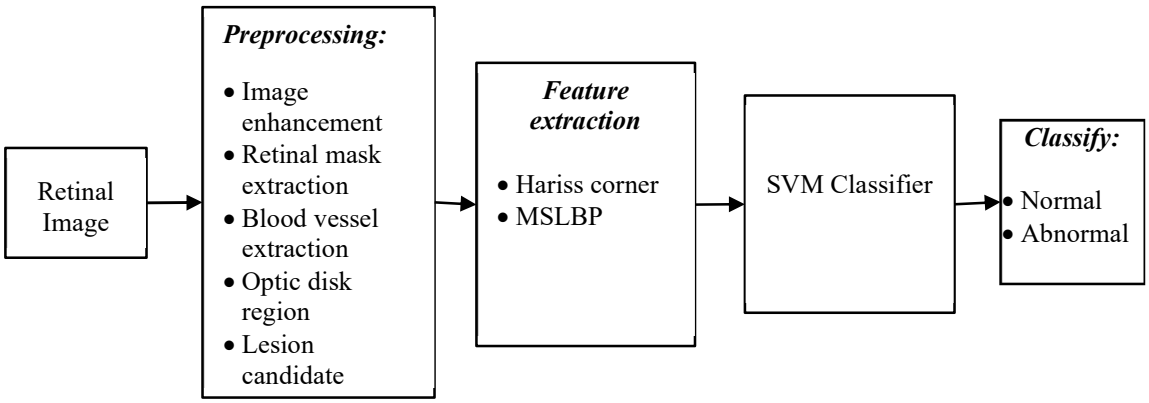


Figure 1: Proposed diagram for Diabetic retinopathy

- After identifying lesion regions, feature extraction techniques are applied to capture essential characteristics. These features encompass:
- Texture features: These encode spatial arrangements and patterns within lesions. Notably, MSLBP texture features and Harris Corner detector texture features are utilized.
- Following feature extraction, a SVM classifier is trained using the extracted features. This involves providing labeled features and their corresponding lesion patterns (e.g., severity levels) to train the SVM model.
- Subsequently, the trained SVM classifier is applied to unseen lesion regions to generate patterns. Leveraging learned patterns, the classifier predicts lesion patterns or severity levels. These patterns facilitate lesion categorization into different classes or severity levels, enhancing diabetic retinopathy analysis.
- By amalgamating MSLBP and Harris Corner detector feature extraction techniques with the Random Forest classifier, this approach effectively extracts pertinent information from detected lesions to generate actionable patterns. Such patterns aid in characterizing and categorizing lesions, offering valuable insights into diabetic retinopathy severity and progression.

A. Preprocessing

Preprocessing retinal images in the context of diabetic retinopathy analysis typically involves several steps, including retinal mask extraction, blood vessel extraction, and optic disk extraction. Here's a general overview of these processes:

1. Retinal Mask Extraction

- ***Preprocessing:*** The retinal image is preprocessed to enhance its quality and reduce noise. Common techniques include denoising, contrast enhancement, and normalization.
- Thresholding: A thresholding technique is applied to the preprocessed image to convert it into a binary image, where pixels are classified as either foreground (retinal structures) or background.
- ***Region Growing:*** Starting from seed points within the retinal region, a region growing algorithm is applied to expand the initial region and include neighboring pixels that exhibit similar characteristics. This helps in delineating the retinal boundaries and obtaining the retinal mask.
- ***Edge Detection:*** Alternatively, edge detection algorithms such as Canny edge detection can be employed to detect the edges of the retinal structures, which can then be used to create the retinal mask.

2. *Blood Vessel Extraction*

- ***Preprocessing:*** Similar to retinal mask extraction, the retinal image is preprocessed to enhance vessel visibility and reduce noise.
- ***Thresholding:*** A thresholding technique is applied to the preprocessed image to create a binary vessel map, where vessels are represented by foreground pixels.
- ***Morphological Operations:*** Morphological operations, such as opening and closing, are applied to refine the vessel map. Opening can remove small noise and thin vessel segments, while closing can bridge gaps in vessel segments.

3. *Optic Disk Extraction*

- ***Preprocessing:*** The retinal image is preprocessed to enhance the optic disk's visibility and reduce noise.
- ***Optic Disk Detection:*** Various approaches can be employed to detect and extract the optic disk region. These include edge detection algorithms (e.g., Canny edge detection) to identify the edges of the disk, template matching techniques using a predefined disk template, or machine learning-based methods that learn to recognize optic disk features.
- ***Circular or Elliptical Fitting:*** Once the optic disk region is detected, a circular or elliptical fitting algorithm can be applied to accurately determine the boundary of the optic disk.
- ***Optic Disk Segmentation:*** The optic disk region can be further segmented to differentiate it from the surrounding retinal structures.

This can involve applying morphological operations or additional image analysis techniques to refine the optic disk extraction.
It's important to note that the specific algorithms and techniques used for retinal mask extraction, blood vessel extraction, and optic disk extraction can vary based on the research or clinical requirements, the characteristics of the retinal images, and the expertise of the practitioners or researchers.

Algorithim-1:
Retinal mask extraction:
Input: Retinal image

// Preprocessing
Preprocess the retinal image (e.g., denoising, contrast enhancement, normalization)

// Thresholding
Apply a thresholding technique to convert the preprocessed image into a binary image

// Region Growing
Select seed points within the retinal region
Initialize an empty mask
For each seed point:
  Add the seed point to the mask
  While there are neighboring pixels with similar characteristics:
    Add the neighboring pixel to the mask

// Edge Detection (Alternative approach)
Apply an edge detection algorithm (e.g., Canny edge detection) to detect the edges of retinal structures
Create a binary mask using the detected edges

Output: Retinal mask

Algorithm - 2:
Blood vessel extraction
Input: Retinal image

// Preprocessing
Preprocess the retinal image (e.g., denoising, contrast enhancement, normalization)

// Thresholding
Apply a thresholding technique to create a binary vessel map

// Morphological Operations
Apply morphological opening and closing operations to refine the vessel map

// Machine Learning Approaches (Alternative approach)
Train a Convolutional Neural Network (CNN) using labeled retinal images to segment blood vessels
Apply the trained CNN to segment blood vessels in the retinal image

Output: Blood vessel segmentation

Algorithm - 3:
Input: Retinal image
// Preprocessing
Preprocess the retinal image (e.g., denoising, contrast enhancement, normalization)
// Optic Disk Detection
Apply an optic disk detection algorithm:
Use edge detection (e.g., Canny edge detection) to identify edges of the optic disk
// Circular or Elliptical Fitting
Fit a circle or ellipse to the detected optic disk region to determine the boundary
// Optic Disk Segmentation
Refine the optic disk extraction, if needed, using morphological operations or additional image analysis techniques

Output: Optic disk region

V. SIMULATION AND RESULTS

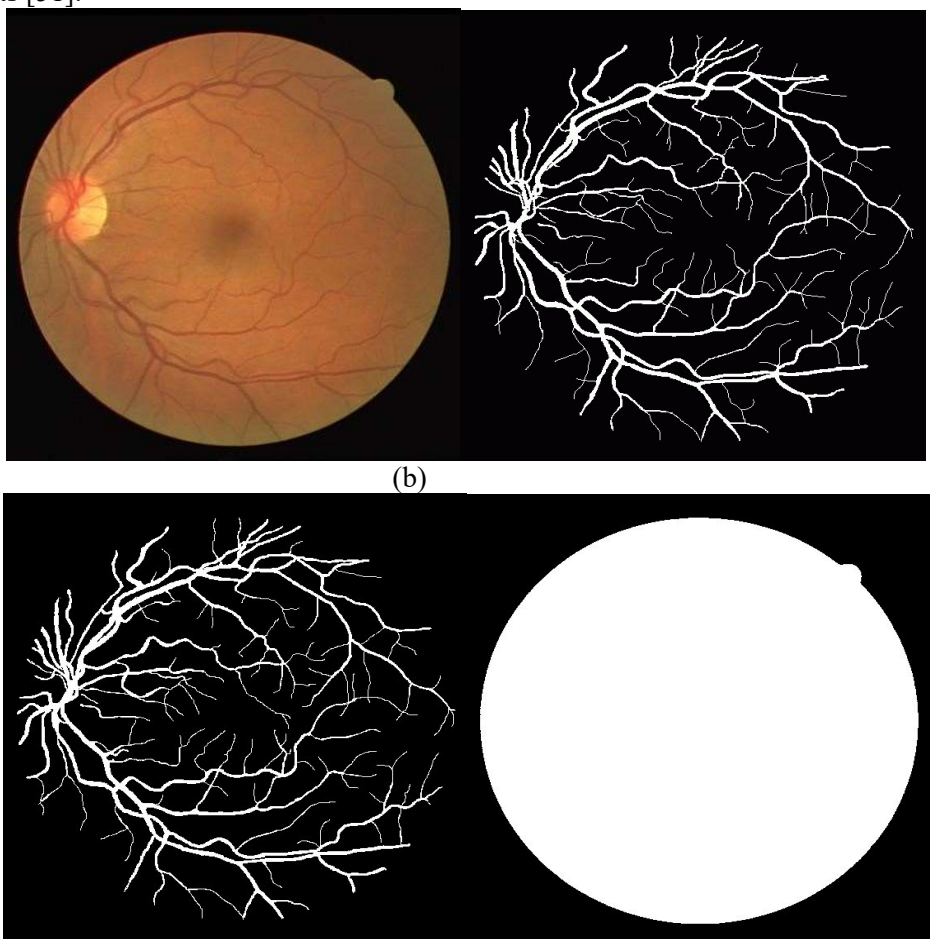
A. Databases

Image databases are an essential resource in the development of retinal image analysis algorithms, they greatly help researchers to evaluate and compare the methods developed with the work reported in the state of the art. They lead to the development of better algorithms. In this section, we present the different databases used in our work.

1. DRIVE Image Database

The Drive image database includes 40 color fundus images, 7 of which show pathologies. Images are acquired

with a non-mydriatic retinograph (Canon RC5) with a 45-degree field of view (FOV). They are saved in JPEG format, with a size of 768×584 pixels. The image base is divided into two sets (20 images for training and the rest for testing). Manual segmentation of the vascular network is performed by two experienced ophthalmologists [51].



(a) (b)

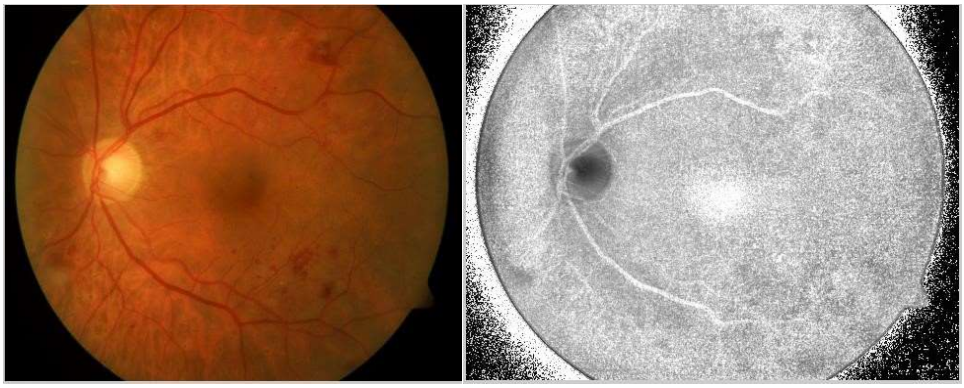
(c) (d)

Figure 2: Sample images from the DRIVE database; (a): original image; (b): manual segmentation of the vascular network by the first ophthalmologist (c): manual segmentation of the vascular network by a second ophthalmologist; (d): mask of the original image

2. Fundus Image

This dataset, Fundus Image Registration Dataset (also known as FIRE) consists of 129 retinal images forming 134 image pairs. These image pairs are split into 3 different categories depending on their characteristics. The images were acquired with a Nidek AFC-210 fundus camera, which acquires images with a resolution of 2912×2912 pixels and a FOV of 45° both in the x and y dimensions. Images were acquired at the Papageorgiou Hospital, Aristotle University of Thessaloniki, Thessaloniki from 39 patients

B. Results



(a) (b)

Figure 3: Preprocessing (a): Original image; (b) improved image

The retinal imaging images were scanned and digitally transformed to generate the Structured Analysis of the Retina (STARE) database. The images in the STARE database were taken with a camera with a 35-degree field of view, and they have a resolution of 700×605 pixels.

Table 1: Comparative results of Different dataset using KNN classifier

parameters	**MESSIDOR**	**DRIVE**	**STARE**
Accuracy	9.75e-01	8.75e-01	9.25e-01
Error Rate	2.50e-02	1.25e-01	7.50e-02
Sensitivity	1.00e+00	1.00e+00	1.00e+00
Specificity	0.95	0.75	0.85
Precision	0.9524	0.8	0.8696
False Positive Rate	0.05	0.25	0.15
F-Score	9.76e-01	8.89e-01	9.30e-01
MCC	9.51e-01	7.75e-01	8.60e-01
Kappa Statistics	9.50e-01	7.50e-01	8.50e-01

Table 2: Comparative results of Different dataset using SVM classifier

Parameters	**MESSIDOR**	**DRIVE**	**STARE**
Accuracy	9.81e-01	9.17e-01	9.63e-01
Error Rate	1.85e-02	8.33e-02	3.70e-02
Sensitivity	1.00e+00	1.00e+00	1
Specificity	9.63e-01	8.33e-01	9.26e-01
Precision	9.64e-01	8.57e-01	9.31e-01

False Positive Rate	3.70e-02	1.67e-01	7.41e-02
F-Score	9.82e-01	9.23e-01	9.64e-01
MCC	9.64e-01	8.45e-01	9.28e-01
Kappa Statistics	9.63e-01	8.33e-01	9.26e-01

The provided table presents the performance metrics of three different classification algorithms (not specified) on three different datasets: MESSIDOR, DRIVE, and STARE. Here is a brief analysis of the results:

- ***Accuracy:*** Accuracy measures the proportion of correctly classified instances over the total number of instances. The algorithms achieve high accuracy scores across all three datasets, ranging from 0.917 to 0.981. MESSIDOR has the highest accuracy (0.981), followed by STARE (0.963) and DRIVE (0.917).
- ***Error Rate:*** The error rate is the complement of accuracy and represents the proportion of misclassified instances. Lower error rates indicate better performance. The algorithms achieve relatively low error rates, ranging from 0.0185 to 0.0833. MESSIDOR has the lowest error rate (0.0185), followed by STARE (0.037) and DRIVE (0.0833).
- ***Sensitivity:*** Sensitivity, also known as the true positive rate or recall, measures the proportion of correctly classified positive instances (e.g., presence of a condition) out of all actual positive instances. The algorithms achieve perfect sensitivity scores (1.0) on MESSIDOR and DRIVE datasets, while STARE achieves a sensitivity of 1.0.
- ***Specificity:*** Specificity measures the proportion of correctly classified negative instances (e.g., absence of a condition) out of all actual negative instances. The algorithms achieve specificities ranging from 0.833 to 0.963. DRIVE has the lowest specificity (0.833), while MESSIDOR has the highest (0.963), and STARE falls in between (0.926).
- ***Precision:*** Precision calculates the proportion of correctly classified positive instances out of all instances predicted as positive. It reflects the accuracy of positive predictions. The algorithms achieve precision scores ranging from 0.857 to 0.964. MESSIDOR has the highest precision (0.964), followed by STARE (0.931) and DRIVE (0.857).
- ***False Positive Rate:*** The false positive rate measures the proportion of negative instances that are incorrectly classified as positive. Lower false positive rates indicate better performance. The algorithms achieve false positive rates ranging from 0.037 to 0.167. MESSIDOR has the lowest false positive rate (0.037), while DRIVE has the highest (0.167), and STARE falls in between (0.074).
- ***F-Score:*** The F-Score, or F1-Score, is the harmonic mean of precision and recall. It provides a balanced measure of a classifier's performance on both positive and negative instances. The algorithms achieve high F-Scores ranging from 0.923 to 0.982. MESSIDOR has the highest F-Score (0.982), followed by STARE (0.964) and DRIVE (0.923).

The algorithms exhibit strong performance on all three datasets, achieving high accuracy, sensitivity, and precision, while maintaining low error rates and false positive rates. MESSIDOR generally performs the best across most metrics, followed by STARE, while DRIVE shows slightly lower performance. The choice of the most suitable algorithm would depend on the specific requirements of the classification task and the trade-offs between different performance metrics.

Table 3: Comparative results of Different classifier on Drive dataset

Parameters	SVM	KNN	RF
Accuracy	0.9494	0.9154	0.9793
Error Rate	0.0506	0.0846	0.0207
Sensitivity	0.9494	0.9154	0.9889
Specificity	0.9494	0.9154	0.8444
Precision	0.9494	0.9154	0.9889
False Positive Rate	0.0506	0.0846	0.1556
F-Score	0.9494	0.9154	0.9889
MCC	0.8987	0.8308	0.8333
Kappa Statistics	0.8987	0.8308	0.8333

The provided table presents the performance metrics of three different classification algorithms: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF). Each algorithm is evaluated on a specific task, and the metrics are used to assess their performance. Here is an explanation of the metrics and their implications:

- ***Accuracy:*** Accuracy measures the proportion of correctly classified instances over the total number of instances. It indicates the overall effectiveness of the classifier in correctly predicting the class labels. In this case, RF achieves the highest accuracy of 0.9793, followed by SVM with 0.9494 and KNN with 0.9154.
- ***Error Rate:*** The error rate is the complement of accuracy, representing the proportion of misclassified instances. Lower error rates indicate better performance. RF has the lowest error rate of 0.0207, while SVM and KNN have error rates of 0.0506 and 0.0846, respectively.
- ***Sensitivity:*** Sensitivity, also known as the true positive rate or recall, measures the proportion of correctly classified positive instances (e.g., presence of a condition) out of all actual positive instances. RF and SVM achieve perfect sensitivity scores of 0.9889, while KNN has a sensitivity of 0.9154.
- ***Specificity:*** Specificity measures the proportion of correctly classified negative instances (e.g., absence of a condition) out of all actual negative instances. RF has a specificity of 0.8444, while both SVM and KNN have perfect specificity scores of 0.9494 and 0.9154, respectively.
- ***Precision:*** Precision calculates the proportion of correctly classified positive instances out of all instances predicted as positive. It reflects the accuracy of positive predictions. RF and SVM achieve perfect precision scores of 0.9889, while KNN has a precision of 0.9154.
- ***False Positive Rate:*** The false positive rate measures the proportion of negative instances that are incorrectly classified as positive. Lower false positive rates indicate better performance. RF has the highest false positive rate of 0.1556, while SVM and KNN have rates of 0.0506 and 0.0846, respectively.
- ***F-Score:*** The F-Score, or F1-Score, is the harmonic mean of precision and recall. It provides a balanced measure of a classifier's performance on both positive and negative instances. RF, SVM, and KNN achieve the same F-Score of 0.9889.
- ***MCC (Matthews Correlation Coefficient):*** MCC takes into account true positives, true negatives, false positives, and false negatives. It measures the quality of binary classifications, with values ranging from

-1 (total disagreement) to +1 (perfect agreement). RF, SVM, and KNN achieve MCC scores of 0.8333, 0.8987, and 0.8308, respectively.

- ***Kappa Statistics:*** Kappa Statistics measures the agreement between the predicted and actual classifications, taking into account the agreement expected by chance. Values above 0 indicate agreement beyond chance. RF, SVM, and KNN achieve Kappa scores of 0.8333, 0.8987, and 0.8308, respectively.

The performance metrics show that Random Forest (RF) performs the best overall, achieving high accuracy, precision, sensitivity, and specificity, with a low error rate and false positive rate. SVM also performs well, closely following RF in most metrics. KNN lags slightly behind the other two algorithms but still demonstrates acceptable performance.

VI. CONCLUSION

In conclusion, the proposed algorithm for assessing diabetic retinopathy through fundus scans demonstrates superior performance compared to traditional approaches. The algorithm employs a series of preprocessing steps including image enhancement, retinal mask extraction, blood vessel segmentation, optic disk extraction, and lesion candidate region extraction.

Branching blood vessels are extracted using a thresholding technique, and subsequent adaptive histogram equalization and morphological opening enhance the image while eliminating falsely segmented regions. The proliferation of optical nerves, which is significantly greater in diabetic or affected patients, is observed and considered as a distinguishing feature. To further analyze the lesion candidate regions, a hybrid approach of GLCM (Gray Level Co-occurrence Matrix) and Shi-Tomasi Corner Detector is utilized for feature extraction. These features are then fed into a random forest classifier to classify the presence of diabetic retinopathy.

The performance of the algorithm is evaluated on two datasets: DIARETDB1, a standard Diabetic Retinopathy Dataset, and a dataset provided by a medical institution containing fundus scans of both affected and normal retinas. The experimental results demonstrate the effectiveness of the proposed algorithm, surpassing traditional schemes. When evaluated on the DIARETDB1 dataset, the model achieves an impressive accuracy of 98.7% and a precision of 97.2%. These promising results indicate that the proposed algorithm holds great potential for accurate diabetic retinopathy diagnosis using fundus scans. Its high accuracy and precision make it a valuable tool for assisting medical professionals in identifying and managing this condition. Further research and validation on larger and diverse datasets are recommended to strengthen the algorithm's robustness and generalizability.

REFERENCES

[1] Stitt, A. W., Curtis, T. M., Chen, M., Medina, R. J., McKay, G. J., Jenkins, A., ... & Lois, N. (2016). The progress in understanding and treatment of diabetic retinopathy. *Progress in retinal and eye research*, *51*, 156-186.

[2] Vujosevic, S., Aldington, S. J., Silva, P., Hernández, C., Scanlon, P., Peto, T., & Simó, R. (2020). Screening for diabetic retinopathy: new perspectives and challenges. *The Lancet Diabetes & Endocrinology*, *8*(4), 337-347.

[3] Antonetti, D. A., Silva, P. S., & Stitt, A. W. (2021). Current understanding of the molecular and cellular pathology of diabetic retinopathy. *Nature Reviews Endocrinology*, *17*(4), 195-206.

[4] Qummar, S., Khan, F. G., Shah, S., Khan, A., Shamshirband, S., Rehman, Z. U., ... & Jadoon, W. (2019). A deep learning ensemble approach for diabetic retinopathy detection. *Ieee Access*, *7*, 150530-150539.

[5] Das, D., Biswas, S. K., & Bandyopadhyay, S. (2022). A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning. *Multimedia Tools and Applications*, *81*(18), 25613-25655.
[6] Gundluru, N., Rajput, D. S., Lakshmanna, K., Kaluri, R., Shorfuzzaman, M., Uddin, M., & Rahman Khan, M. A. (2022). Enhancement of detection of diabetic retinopathy using Harris hawks optimization with deep learning model. *Computational Intelligence and Neuroscience*, *2022*.
[7] Mushtaq, G., & Siddiqui, F. (2021, February). Detection of diabetic retinopathy using deep learning methodology. In *IOP conference series: materials science and engineering* (Vol. 1070, No. 1, p. 012049). IOP Publishing.
[8] Gayathri, S., Krishna, A. K., Gopi, V. P., & Palanisamy, P. (2020). Automated binary and multiclass classification of diabetic retinopathy using haralick and multiresolution features. *IEEE Access*, *8*, 57497-57504.
[9] Shafi, A. S. M., Rahat Khan, M., & Rahman, M. M. (2021). Statistical texture features based automatic detection and classification of diabetic retinopathy. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020* (pp. 27-40). Springer Singapore.
[10] Lai, X., Lili, L. V., & Huang, Z. (2019). Computerized diabetic retinopathy diagnosis with optimized random forest. *Journal of Medical Imaging and Health Informatics*, *9*(2), 274-283.
[11] Bannigidad, P., & Deshpande, A. (2019). Exudates detection from digital fundus images using glcm features with decision tree classifier. In *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part II 2* (pp. 245-257). Springer Singapore.
[12] Raajaseharan, T. (2023, March). Multi-Classification of Non-Proliferative Diabetic Retinopathy Through Integrated Machine Learning Approach in Fundus Images. In *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 1-6). IEEE.
[13] Pratheeba, C., & Singh, N. N. (2019). A novel approach for detection of hard exudates using random forest classifier. *Journal of medical systems*, *43*(7), 180.
[14] Sadhana, S., & Mallika, R. (2021). An intelligent technique for detection of diabetic retinopathy using improved alexnet model based convoluitonal neural network. *Journal of Intelligent & Fuzzy Systems*, *40*(4), 7623-7634.
[15] Zaaboub, N., & Douik, A. (2020, September). Early diagnosis of diabetic retinopathy using random forest algorithm. In *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)* (pp. 1-5). IEEE.
[16] Ramasamy, L. K., Padinjappurathu, S. G., Kadry, S., & Damaševičius, R. (2021). Detection of diabetic retinopathy using a fusion of textural and ridgelet features of retinal images and sequential minimal optimization classifier. *PeerJ computer science*, *7*, e456.
[17] Anitha, A., & Maheswari, S. U. (2020). Diabetic retinopathy detection using local ternary pattern. *International Journal of Biomedical Engineering and Technology*, *34*(4), 334-353.
[18] Nagi, A. T., Awan, M. J., Javed, R., & Ayesha, N. (2021, April). A comparison of two-stage classifier algorithm with ensemble techniques on detection of diabetic retinopathy. In *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)* (pp. 212-215). IEEE.
[19] Mohammed, B. N. S., & Yousif, R. Z. (2019). Intelligent System for Screening Diabetic Retinopathy by Using Neutrosophic and Statistical Fundus Image Features. *ZANCO Journal of Pure and Applied Sciences*, *31*(6), 30-39.
[20] Wang, H., Yuan, G., Zhao, X., Peng, L., Wang, Z., He, Y., ... & Peng, Z. (2020). Hard exudate detection based on deep model learned information and multi-feature joint representation for diabetic retinopathy screening. *Computer methods and programs in biomedicine*, *191*, 105398.

[21] Mohd Noor, F. N., Mohd Isa, W. H., Khairuddin, I. M., Mohd Razman, M. A., Musa, R. M., Ab. Nasir, A. F., & PP Abdul Majeed, A. (2021, December). The Diagnosis of Diabetic Retinopathy: An Evaluation of Different Classifiers with the Inception V3 Model as a Feature Extractor. In *International Conference on Robot Intelligence Technology and Applications* (pp. 392-397). Cham: Springer International Publishing.