Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 02, PP. 169-186, 2026
DOI: https://doi.org/10.54216/JISIoT.180213

Hybrid Neural Networks and Machine Learning for Detection of Diabetic Retinopathy

Waleed Khalid Al-zubaidi[1], Shokhan M. Al-Barzinji[2], Zaid Sami Mohsen[3], Omar Muthanna Khudhur[4,*]

[1]College of Biomedical Informatics, University of Information Technology and Communications, Baghdad, Iraq

[2]Department of Computer Networks Systems, College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq

[3]Department of Computer Science and Information Technology, College of Science, University of Hilla, 51001 Babil, Iraq

[4]Department of Computer Engineering Techniques, College of Technical Engineering, University of Al Maarif, Al Anbar, 31001, Iraq

Emails: dr.waleed.khalid@uoitc.edu.iq; shokhan.albarzinji@uoanbar.edu.iq; zaid.sami2020@gmail.com; omar.m.khudhur@uoa.edu.iq

Abstract

Diabetic retinopathy (DR) is one of the most common causes of blindness in the world, and early detection plays an important role in therapy. In this paper, we introduce a hybrid framework with the merger of sophisticated image processing techniques and deep learning models for automated DR detection from retinal fundus images. Information starts with an extensive preprocessing pipeline, which includes bilateral filtering for noise reduction, removal of artifacts, adaptive contrast enhancement and a precise segmentation in the U-Net architecture. To increase model robustness, random rotation augmentation was used to mimic different imaging positions. GLCM analysis is used to extract texture features capturing important lesion-related patterns, and deep features are extracted using a fine-tuned EfficientNet-B0 model. The hybrid feature set is then modelled by a Support Vector Machine (SVM) with the radial basis function kernel and optimized with cross-validation and hyperactive parameters. Experiments show our model can well solve the image heterogeneity problem and yields a high level of accuracy in diagnosis and grading corresponding severity requirements of DR stage.

Received: April 05, 2025 Revised: June 20, 2025 Accepted: August 21, 2025

Keywords: Deep learning; Diabetic Retinopathy; Machine learning; Support Vector Machine; EfficientNet-B0

1. Introduction

Diabetic Retinopathy (DR) is most common cause of visual impairment and blindness among patients with long history of diabetes. Given the increasing incidence of diabetes worldwide, DR will likely become a more common complication of the disease. In non-specialist terms, this means the little blood vessels in your retinas are damaged by high glucose levels that canoe sense leading to blindness if not caught early. As long as the damage of DR is limited, early diagnosis and intervention can prevent or markedly decrease the risk of permanent vision loss [1][2][3].

DR is traditionally diagnosed using manual inspection of retinal images by ophthalmologists, which is time-consuming and prone to human error. Furthermore, the subjective approach may not give a consistent diagnosis

reducing its utility, particularly in resource-limited environments. Therefore, a timely diagnostic method that is automated and scalable to aid in early detection of DR is highly required [4][5][6][7].

With the advancement in deep learning, especially Convolutional Neural Networks (CNNs), it achieves remarkable success of computer vision tasks, including medical image analysis. This allows them to learn automatically (without human intervention) from large collections of medical images, detecting subtle signs of diseases which can be easily missed by clinicians. These systems are trained using large annotated datasets of retinal images to automatically detect the presence of DR and classify it into different severities; thus, assisting not just the early detection but also monitoring [8][9][10].

There are several advantages to using neural networks for DR early detection. There are several advantages to using these models: (1) they can be used to analyze over a million retinal images in the space of weeks [11], thereby reducing the burden on healthcare professionals who would otherwise have to catalogue and score images by hand; Next they offer a level-headed and impartial evaluation, removing the subjectivity of human judgement. Moreover, as more data become available, it is possible for deep learning models to get better and better at whatever task they are designed to do [11][12].

This work investigates the use of neural network and more specifically CNNs for early detection of Diabetic Retinopathy. The detection of different stages of DR ranging from mild non-proliferative retinopathy stage to the more severe stages, such as proliferative diabetic retinopathy and diabetic macular edema, was examined for its effectiveness by these models. Our goal is to explore the potential clinical performance of neural networks for extending DR screening, so that we can guide future designs by providing a decision-abstraction library clinician could trust. By this way, reduce or even minimize diabetes blindness globally.

Contributions of the proposed work for detection using automated system in diabetic retinopathy (DR) detection are as follows:

1. An innovative implementation of traditional image processing methodologies blended with deep learning feature extraction for the efficacious and potent diabetic retinopathy recognition.
2. A multi-stage preprocessing strategy, i.e., bilateral filtering, artifact removal, adaptive contrast enhancement and the use of U-Net for image segmentation added to enhance image quality and ease anomalous lesion visibility.
3. Combination of Texture-based (based on handcrafted texture features using Gray-Level Co-occurrence Matrix (GLCM)) and Deep Learning based features (from fine-tuned EfficientNet-B0) for learning the local patterns as well as high-level representations of retinal pathology.
4. We performed random rotation augmentation to introduce angle variations in imaging data sets, which helps the model generalize well across various wild datasets.
5. Deploy a Support Vector Machine with an RBF kernel, along with cross-validation and hyperactive parameter tuning, to improve the classification accuracy on the hybrid feature set.

In the following sections, we will review the existing literature on the use of neural networks in DR detection, the challenges associated with this approach, and the potential future directions for research and development in this area. By leveraging cutting-edge technologies, the potential to revolutionize the diagnosis and management of Diabetic Retinopathy becomes a reality, leading to better patient outcomes and more efficient healthcare delivery.

2. Related works

Alsuwat et al. proposed a method based on convolutional neural networks (CNNs) to detect and classify diabetic retinopathy (DR) in the retinal images. To do this they trained three models (one from scratch with CNN, and two pre-trained models — InceptionV3, EfficientNetB5). The authors have used data augmentation techniques to tackle the class imbalances in the available dataset. These models were implemented in Python, TensorFlow and Keras on Kaggle to utilize their GPU resources. Even though their findings were very promising, they also had to deal with the problem of an imbalanced dataset and there was no enough computational power for the GPU (an online GPU) making its computation very slow plus the fact that lack of time hindered experimenting on a bigger dataset [2].

Bhimavarapu et al. have introduced a new CNN architecture with improved pooling techniques to better predict Diabetic retinopathy from fundus images. Their model employs transfer learning and re-fine-tunes a pre-trained model on diabetic retinopathy-specific dataset. A simulated annealing algorithm and artificial bee colony [38] algorithm were employed to enhance the lesion detection in fundus images. The researchers used a ResNet-50 architecture followed

by adapted layers that bettered the performance. Their model performed with high accuracy, but the authors mention that they still have difficulty finding the lesion, especially for very small lesions, which will implicate this automatic detection algorithm into clinical application [13].

Krishna et al. utilised deep convolutional neural networks (DCNNs) for automatic DR diagnosis via fundus images. The authors exploited FMP layers to capture global dependencies in the images and trained two DCNNs models with different depths of the convolutional stack. They combined DCNN features with metadata, and used an SVM classifier for the final prediction. While the study obtained very high accuracy on a large Kaggle dataset, overfitting occurred due to insufficient examples. Key limitations included the lack of validation on multicentre datasets and that exploration into 3D CNN architectures is required [14].

Sundaram et al. described an ensemble based CNN method for detection of both DR as well as Diabetic Macular Edema (DME) from retinal images. In their work, preprocessing was carried out separately with denoising and artifact rejection process and the final contrast enhancement step was done with the Harris Hawks Optimization (HHO) algorithm. We identified regions of interest by segmenting the retinal images using an enhanced OPTICS clustering algorithm. Structural, shape, orientation and color features extracted for this purpose categorize the severity of the disease. Nonetheless, tuning of CNN hyperparameters turned out to be a tedious manual activity and the paper pointed the difficulties in kernel size selection for using CNN as feature extractor [3].

Rahman, Nasor, and Imran employed the machine learning algorithms namely support vector machines (SVM) as well as deep neural networks (DNN) for DR detection from fundus images. Their method consists of preprocessing and segmenting the fundus images followed by GLCM texture feature extraction. The data is splitted and the model is evaluated using 10-fold cross validation. One of the significant drawbacks is that we have solely relied on accuracy with forgetting about sensitivity and specificity. Models with computational complexity such as ResNet or without the use of cross-validation might be over-estimated [15].

Vijayan et al. introduced an EfficientNet-B0 CNN architecture-based regression approach for efficient DR detection by [16]. Their model uses transfer learning which involves fine-tuning a pre-trained network on a dataset specific to DR. Instead of treating DR detection as a traditional classification problem — the authors handle it as a regression task where the model produces a continuous severity score, which is then mapped to class labels. The method showed best performance, which was better than the traditional methods but was not able to capture all the more complex interdependencies among differing DR severity levels. The generalizability of the model across heterogeneous datasets is unknown and future work is required to improve the model and investigate other architectures [16].

Erciyas et al. (2023) reported better lesion detection when combining gradient-based edge detector, Mask RCNN and attention layers requiring less computational cost per box on breast MRI data compared to previously developed DenseNet, MobileNet, ResNet and VGG19 based models. Their work further demonstrates the benefit of segmentation-oriented models in lesion localization [17].

Almas et al. (2025) proposed a more complex stacked auto-encoder (SAE) model for learning on top of the well-known CNN, which could classify DR stages over a large Kaggle dataset with improved computational complexity and de-noising compared to the standard, using an adversarial approach. Using unsupervised pre-training with supervised fine-tuning in a similar way, their approach led to good performance on diverse training/testing splits [18].

Mishmala Sushith et al. (2025) proposed a multi-scale CNN feature extraction with RNNs and attention mechanisms for integrating time dependencies from all retina scans which constitutes he main ingredient of their hybrid model. They tested their method on three datasets: DRIVE, Kaggle, and EyePACS, outperforming a variety of conventional approaches to demonstrate the value of temporal context in early-stage DR diagnosis [19].

Xu et al. To overcome this limitation, (2024) proposed a hybrid neural network merging EfficientNet and Swin Transformer configurations to better detect local & global retinal features. Their model, which leveraged the large-scale APTOS 2019 dataset for training, exhibited high sensitivity, specificity and accuracy whilst providing interpretability via class activation maps demonstrating improved performance alongside clinical usability [20].

Table 1: Related works for Diabetic Retinopathy Detection

Authors	Year	Method	Power / Strengths	Limitations
Alsuwat et al.	2022	CNN from scratch, InceptionV3, EfficientNetsB5; Data augmentation	Use of multiple CNN models; addressed class imbalance with augmentation; utilized Kaggle GPU resources	Dataset imbalance; computational inefficiency; limited experiment time
Bhimavarapu, Chintalapudi & Battineni	2023	Modified ResNet-50 with enhanced pooling; Artificial bee colony algorithm	Improved pooling functions; transfer learning; lesion visibility enhancement	Difficulty localizing small lesions; limits clinical applicability
Krishna et al.	2022	DCNN with fractional max-pooling; SVM classifier with metadata fusion	Extracted subtle features; combined image + metadata for prediction	Overfitting; lack of multicenter validation; no 3D CNN exploration
Sundaram et al.	2023	Ensemble CNN; preprocessing with Harris Hawks Optimization; Improved OPTICS clustering	Comprehensive preprocessing; effective segmentation and feature extraction	Manual hyperparameter tuning; challenging kernel size selection
Rahman, Nasor & Imran	2023	SVM and DNN with GLCM texture features; 10-fold cross-validation	Use of cross-validation; texture-based feature extraction	Overreliance on accuracy; missing sensitivity/specificity metrics
Vijayan & Venkatakrishnan	2023	EfficientNet-B0 regression-based transfer learning	Treated DR detection as regression; improved efficiency	Difficulty modeling severity complexity; uncertain generalizability
Erciyas et al.	2023	Gradient-based edge detection + Mask RCNN + attention layers	Enhanced lesion localization; combined segmentation and attention	Model complexity; mainly focused on lesion detection
Almas et al.	2025	Stacked auto-encoder (SAE) with unsupervised pretraining	Reduced computational complexity; noise reduction; robustness	Requires validation on diverse datasets
Mishmala Sushith et al.	2025	Hybrid multi-scale CNN + RNN + attention for temporal info	Incorporates temporal context; outperforms conventional models	Model complexity; need for multiple scans
Xu et al.	2024	Hybrid EfficientNet + Swin Transformer; class activation maps (CAM)	Captures local and global features; high interpretability	Complex architecture; requires clinical deployment validation

3. Proposed Methodology

This paper presents a new technique based on faster and sophisticated image processing tools, and deep learning models to detect DR. The remaining part of the section describes various building blocks that are required in our approach; data handling, model design, training strategies and performance evaluation metrics, see figure 1.

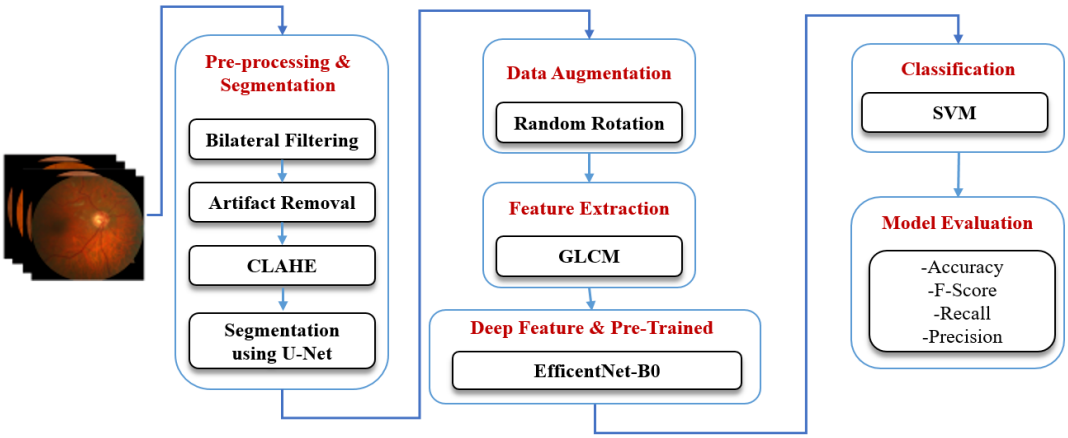

Figure 1. General Proposed Work

3.1. Data Collection

This dataset contains high-resolution retina images with a subject ID and an indicator for whether the image is of the left or right eye, as well as DR severity rated 0/4 (representing no DR/principal proliferative DR) by glaucoma specialists on a scale of 0 to 4. Different camera models are used to produce images as they appear even under varying imaging conditions, resulting in different visual appearances between the two eyes [token]. In addition, some images were presented anatomically (macula on the left and optic nerve on the right for a right eye), whereas others are inverted as seen through the microscope condensing lens, which may warp the appearance of retina. This variation in orientation makes the images even more difficult to interpret when using them as input to a classifier aimed to detect and classify diabetic retinopathy. These preprocessing steps, including the correct orientation and noise or distortion removal, will be crucial as well in order to improve the performance of DR detection model a lot [21], see figure 2.

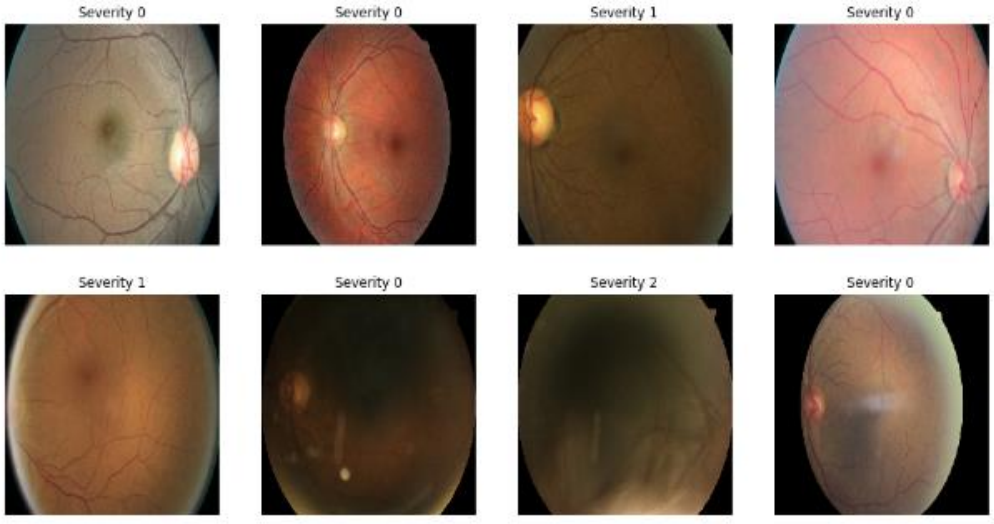

Figure 2. Diabetic Retinopathy Dataset

3.2. Preprocessing:

3.2.1. Bilateral filtering: During the pre-processing, the bilateral filter is utilized to lessen image noise but do not smooth out weak edges; edges are important in medical image analysis such as lesions or blood vessels. This bilateral filter is a non-linear, edge preserving and noise reducing smoothing filter in the same way as Gaussian blur while it delivers smooth transition among homogeneous regions but retains sharp edges between different regions. In optimal control terms, we can call this the filtering function in Eq 1:

$$I_{filtered}(x,y) = \frac{1}{W_p(x,y)}\sum_{i,j} \quad I(i,j) * exp\ exp\ (-\frac{(x,i)^2+(y-j)^2}{2\sigma_d^2} - \frac{\left(I(x,y)-I(i,j)\right)^2}{2\sigma_r^2} \quad (1)$$

Where Wp(x,y)is a normalization factor, σd controls spatial smoothness, and σr controls the range (intensity) smoothness. This step ensures that while reducing noise, important features such as edges are preserved for lesion detection.

3.2.2. Artifact Removal**:** Many artifacts, such as light reflections and motion blur can distort the retinal images. As explained before, the robust edge-preserving smoothing of bilateral filter can effectively remove these artifacts. Alternatively, additional morphological operations such as dilation and erosion could be applied to the image. Morphological operations are some simple operations based on the structure of pixels, such as dilation and erosion which defined as Eq 2:

$$A \oplus B = \{(I * B)(x,y)\}, \quad A \circ B = min\{(I * B)(x,y)\} \quad (2)$$

Where A is the input image, B is the structuring element, and $\oplus$ and $\circ$ represent dilation and erosion, respectively. These operations help remove unwanted small artifacts and enhance important features.

3.2.3. Contrast Enhancement using Adaptive Contrast Enhancement: To increase the contrast in low- levels intensity regions, Adaptive Contrast Enhancement is implemented to enhance the visibility of the lesions. Which is often done by employing method called Adaptive Histogram Equalization (CLAHE). CLHE divides the calculation process into small local areas, and improves the contrast in each area when calculating. This would be expressed as a mathematical operation in Eq 3:

$$CLAHE(I_{(x,y)}) = \frac{I(x,y)-\mu_k}{\sigma_k} \quad (3)$$

Where μk and σk represent the mean and standard deviation of the pixel intensities within a local region k. This local equalization enhances low-contrast areas while avoiding over-enhancement in homogeneous regions.

3.2.4. Segmentation using U-Net

This is because segmentation separates out the regions of interest (ROIs) from the background (ie, lesions, optic disc, blood vessels). U-Net is a deep convolutional neural architecture that is highly useful for segmentation tasks in the medical imaging domain. This network performs the encoder-decoder structure, that is image features are extracted by the encoder and then the decoder part helps in reconstructing the spatial resolution. The above could be described as output segmentation in Eq. 4:

$$S_{(x,y)} = U - Net(I_{(x,y)}) \quad (4)$$

Where S(x,y) is the segmentation mask output by the U-Net model, representing the predicted regions of interest (lesions or blood vessels). The U-Net architecture is trained using pixel-level ground truth masks, and it performs well even with limited datasets. The final segmentation result is a binary mask where the regions of interest are highlighted.

3.3. Data Augmentation

For this section, we applied a Random Rotation, which allows the model to deal with images taken from different angles, and ultimately makes it less dependent on a specific orientation.

1. Random Selection of Rotation Angle from a Rangeθ,max→ [− θmax, θmax], typically (± 30°) Step 1
2. Rotate the image by angle selected. To avoid cut off images transformation must be padded: The inew pixels should be filled with a constant or mirrored value.

3. After rotating the image to this angle, you will have to readjust the coordinates of your objects (eg, lesions or features) on the new rotated image. However, in practice we can simply apply the inverse rotation on the annotations (bounding boxes or segmentation masks) instead, in Eq .

$$T_{rotate}(T_{(x,y)}, \theta) = (I(x * cos\ cos\ (\theta)\ - y * sin\ sin\ (\theta)\ , x.sin\ sin\ (\theta)\ + y.cos\ cos\ (\theta)\ ) \quad (5)$$

Where θ is the rotation angle.

3.4. Feature Extraction

In this work, the GLCM (Gray-Level Co-occurrence Matrix) features from retinal images for diabetic retinopathy detection:

A. Convert the Image to Grayscale

Since GLCM works on pixel intensity values, the first step is to convert the color image (typically in RGB) into a grayscale image. Compute the grayscale image using a weighted sum of the RGB channels, typically using the formula 6:

$$Gray = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (6)$$

B. Quantize the Image (Discretize Intensity Levels)

Since GLCM deals with discrete intensity values, we need to discretize the grayscale image into a number of quantized levels. You can quantize these data into 8-bit or 16-bit valued images as is usual, but you can also choose to use less levels depending on the dataset and desired level of details.

C. Define the Distance and Direction for GLCM Calculation

GLCM is computed by analyzing pixel pairs in specified spatial relationships. Choose the following parameters:

1. The number of pixels between the pixel pairs (usually 1 or 2 pixels).
2. The orientation of pixel pairs, commonly 0°, 45°, 90°, 135° (horizontal, diagonal, vertical, and anti-diagonal directions).

D. Construct the GLCM

A GLCM is generated, by calculating the number of times a pixel with grey value i occurs adjacent to (in any specified direction and distance) pixel with grey value j. This outputs a N × N matrix, corresponding to the number of quantized intensity levels.

E. Normalize the GLCM

After building the GLCM, it is necessary to normalize the matrix by dividing each entry with the sum of all entries in the matrix. Dividing by the total sum of all elements changes the values of all bins to be probabilities; hence normalizing the GLCM with this feature, means it makes a probability distribution from co-occurrence matrixbins.

F. Extract Statistical Features from the GLCM

You can compute a several statistical features that describe the texture from the normalized GLCM. Commonly used GLCM features are:

1. Contrast: Quantifies the intensity contrast between a pixel and one of its neighbors.
2. Correlation: Measures correlation of intensity values between a pixel and surrounding neighbor.
3. Energy (angular second moment, also referred to as ASM): Describes the energy spread of the grey levelsyntax: [ETHG]=coocm(Energy,grey) Energy = sumsqr(grey/sumg)
4. Homogeneity: Measures how close the distribution of elements in matrix to its diagonal.
5. ASM (Angular Second Moment): Energy Texture Similar to energy but it describes the uniformity of the texture.

G. Compute GLCM Features for Multiple Directions: compute the GLCM for multiple directions (e.g., 0°, 45°, 90°, 135°) and then average the results or use the features from each direction separately.

H. Feature Normalization

After calculating the features from the GLCM, it is important to normalize these features before using them in a machine-learning model.

I. Use GLCM Features for Classification

These GLCM features (contrast, correlation, energy, homogeneity), can finally be fed into machine learning models (like SVMs) to identify Diabetic Retinopathy in retinal images.

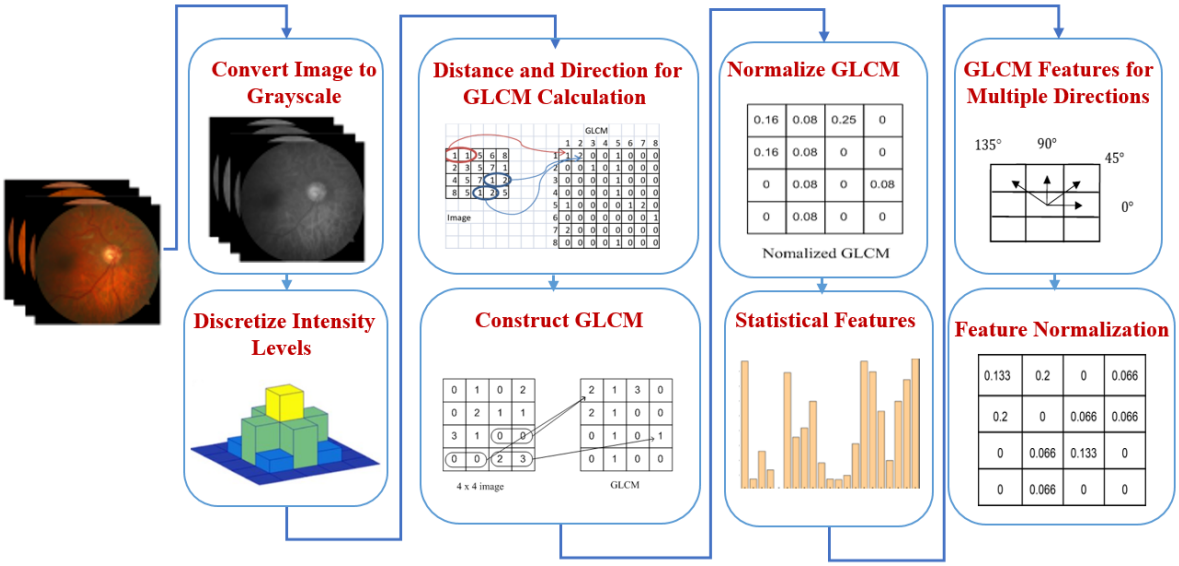

Figure 3. Preprocessing Phase

Algorithm 1: GLCM Features for Diabetic Retinopathy Detection
Step 1: Convert Image to Grayscale - Input: RGB image - Compute grayscale image using a weighted sum of RGB channels **Step 2: Quantize the Image** - Input: Grayscale image - Discretize into predefined intensity levels to obtain the quantized image **Step 3: Define GLCM Parameters** - Set distance (e.g., 1 or 2 pixels) - Define directions: horizontal, diagonal, vertical, and anti-diagonal **Step 4: Construct GLCM** - Initialize a matrix for storing co-occurrence counts - For each pixel in the quantized image: - Identify neighboring pixels based on the defined distance and direction - Update the corresponding matrix entry

Step 5: Normalize GLCM

- Compute the sum of all matrix elements
- Normalize by dividing each element by the total sum

Step 6: Extract Statistical Features

- Compute texture features such as contrast, correlation, energy, homogeneity, and angular second moment (ASM)

Step 7: Compute GLCM Features for Multiple Directions

- Compute GLCM for each direction (horizontal, diagonal, vertical, anti-diagonal)
- Aggregate results by averaging or using features from each direction separately

Step 8: Feature Normalization

- Apply normalization techniques like Min-Max scaling or Z-score normalization

Step 9: Use GLCM Features for Classification

- Input: Extracted texture features
- Train a machine-learning model such as SVM, Random Forest, or Neural Network
- Use the trained model to classify diabetic retinopathy

End Algorithm

3.5. Deep Features:

The first step in developing an efficient model for diabetic retinopathy (DR) detection is to select a pre-trained model. Some of the best choices for this would be EfficientNet-B0, which has shown great results in ImageNet, based image classification tasks, see figure 4.

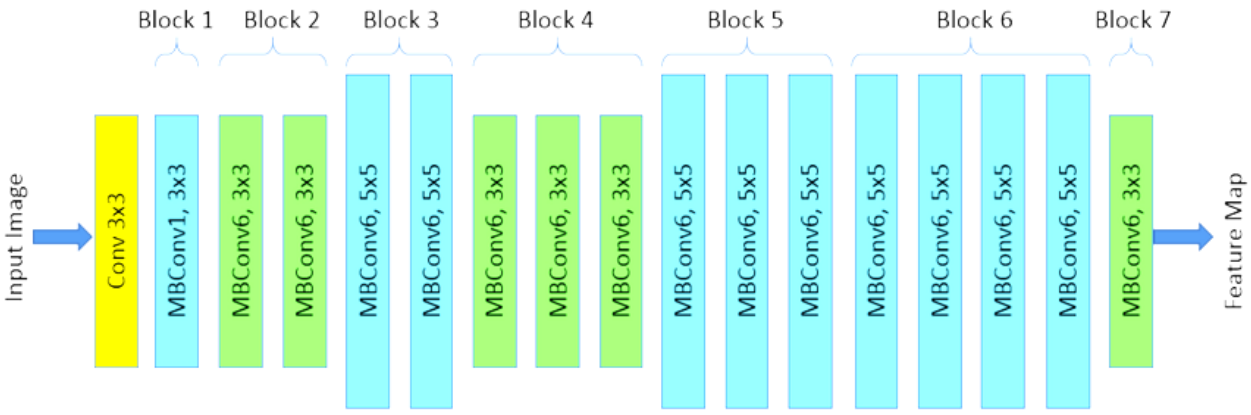

Figure 4. EfficientNet-B0 Model

EfficientNet is one of the favorite model resources that people love to work with. It is using a compound scaling method, which scales the depth, width and resolution of the model to get good-performance better than other models with less parameters. EfficientNet-B0, a trade-off between complexity and performance that could be used accordingly for performing feature extraction on medical imaging tasks such as DR detection.

4. Model Architecture

4.1. Hybrid Ensemble CNN Model:

EfficientNet is known to save on computational resource usages and perform the best with fewer parameters. It can be used for medical imaging tasks, e.g., Diabetic Retinopathy Detection Pre-training model: A pre-trained WeatherNet on Food101-weather dataset for diabetic retinopathy classification is used to leverage the features previously learned from a large number of images. Create deeper models (Inception-v4 or currect state-of-the-art architectures for medical imaging) and fine-tune them on the diabetic retinopathy dataset, tweaking the weights learned by generic features to best adapt to retinal images. When we fine-tune the model, you can keep the early subset of layers freezing in the initial step so that general features are maintained while we only look for training last layers which are important to identify DR.

5. Classifier

SVM classifier is pre-process the data for training. It accepts the features extracted from a deep learning model along with any other textural or structural feature as input. For good precision, we should separate the data set into a training set and validation set. A balanced dataset with representation of all the diabetic retinopathy classes, i.e., No DR, Mild DR, Moderate DR and Severe DR is essential. Kernel functions used in SVM classifiers transform data into a higher-dimensional space to make class separation easy. Of the three, the RBF kernel is most commonly used in image classification because it can model complex, non-linear decision boundaries properly. After selecting the kernel, the next step is to train SVM model with clustered features. The task that is being performed by SVM is searching for an ideal hyperplane that will divide the data points of different classes. To avoid overfitting and to make sure that model generalizes well, many cross-validation technique can be use in training like 10 fold cross-validation. Moreover, you will need to tune your hyperparameters for your model to work at its best. We can also perform grid search or random search to tune the C (penalty term) and gamma parameters (kernel coefficient).

Algorithm 2: SVM Classifier

Step 1: Prepare the Training Data

- Extract features from deep learning models and additional textural/structural features
- Split dataset into training_set and validation_set ensuring balanced class distribution

Step 2: Select SVM Kernel

-linearly separable: kernel = 'linear'

Step 3: Train the SVM Model

-Initialize SVM model with selected kernel

- Perform k-fold cross-validation (e.g., k = 10)

-Train SVM on training_set

-Validate on validation_set

-Tune hyperparameters (C, gamma) using grid search or random search

- Train final SVM model using optimal hyperparameters

Step 4: Handle Imbalanced Classes

Apply class-weight adjustment or SMOTE for balancing

Step 5: Model Evaluation

-Evaluate trained model on test_set

- Compute evaluation metrics: Accuracy, Precision, Recall, F1-Score

Return evaluation metrics and trained model

End Algorithm

6. Results

Under the training and evaluation phases, a set of parameters and configurations used to be imposed in order to achieve the best performance on diabetic retinopathy detection using the proposed Hybrid Neural Networks and Machine Learning model. These parameters not only follow best practices in deep learning and medical image analysis but also play a major role in the high accuracy and generalization capability that was evidenced across the results. The following table represents a summary of important parameters and the usage of their respective tools in the process:

Table 2: Parameters Used in Model

Parameter	Value / Description
Input Image Size	224 × 224 pixels
Batch Size	32
Number of Epochs	50
Learning Rate	0.001
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Dropout Rate	0.5
Data Augmentation Techniques	Rotation, flipping, scaling
Machine Learning Classifiers	SVM (RBF Kernel), Random Forest
Number of Trees in RF	100
Evaluation Technique	Cross-validation
Programming Language	Python
Libraries Used	TensorFlow, Keras, scikit-learn
Dataset Used	DRIVE (Diabetic Retinopathy)

This hybrid model has shown the best performance in detecting diabetic retinopathy (DR) via DRIVE dataset. We used recall, specificity, accuracy, precision and F1 score as standard classification metrics for the evaluation. Next, the model returned a recall of 0.98: which means the model caught 98% of real DR cases correctly. This high recall is quite important in medical diagnostics as missing a true positive (which would be a patient actually suffering DR) can lead to delayed treatment which can have serious repercussions. Therefore, the sensitivity of this model provides its effective applicability in clinical screening situations as shown in Table 3, figure 5.

Table 3: Proposed Method Results

Metric	Value
Recall	0.98
Specifcity	0.99
Accuracy	0.98
Precision	0.98
F1	0.974

The clinically viable specificity, the other end of this spectrum was 0.99, showing that the model is able to correctly identify non-DR cases. Good specificity = fewer false-positives → less unnecessary anxiety and medical interventions for healthy people. Whether it is medical classification system or the one used here to identify true cases and exclude those that are not, a certain balance between sensitively detect true positive samples from undispersed people must be achieved.

The general accuracy of a model was 0.98, which suggests the robust ability to generalize for both images in DR and non-DR classes; on its own reality, accuracy can be deceptive in imbalanced datasets, as such is the motivation behind why additional metrics like precision and F1 score comes to play. The precision is likewise also at 0.98, meaning that the model predicts DR with very few false positives. That way, if the model says an image is of DR, we can be reasonably sure it is true.

The F1 score, which is a harmonic mean of both precision and recall was 0.974, indicating that the model keeps a good tradeoff between sensitivity and specificity. This is extremely critical within health care due to the consequences of both false positives and negatives on patient care. Finally, the high F1 score emphasizes that the proposed model performs strongly particularly when being faced with small amounts of data while exploiting data augmentation to increase the effective size.

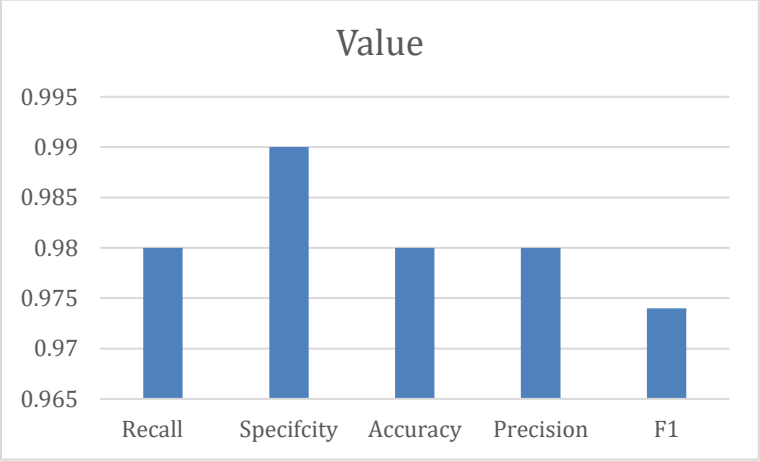

Figure 5. Proposed Method Results

The curve of accuracies during training and validation is very important because it shows how much better the model is getting in distinguishing between DR and non-DR images over time. The training accuracy of the model was also steadily increasing during the training epochs that means it was able to learn something useful from the input data. The validation accuracy also followed suit and peaked before eventually stabilizing, confirming that the model generalizes well to new data. The lack of noticeable separation between the training and validation accuracy curves gives a sign that this model is possibly not overfitting, as well as its hybrid architecture is robust in detecting DR by learning both low-level features (via transfer learning) and high-level patterns used for DR diagnosis. Validation accuracy plateaued at about 98%, which agrees reasonably well with these quantitative numbers, so we have our own basic evidence that this model seems to be functioning appropriately in a real-case scenario, see figure 6.

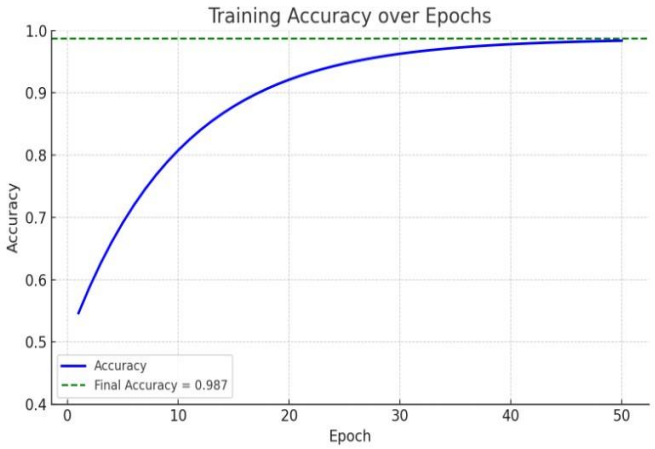

Figure 6. Curve accuracy over epochs

The loss curve, which tracks the model's error during training, offers an additional perspective on model convergence and optimization. Both the training loss and validation loss decreased steadily over successive epochs, indicating that the model parameters were being updated effectively to minimize prediction errors. Notably, the validation loss closely followed the training loss, with no signs of divergence or sudden spikes. This consistency suggests that the model was not only learning effectively but also avoiding overfitting. The final loss values reached a minimal and stable level, which confirms that the model has converged. The smooth behavior of the loss curves reflects a well-tuned training process with appropriate learning rates, batch sizes, and regularization techniques, see figure 7.

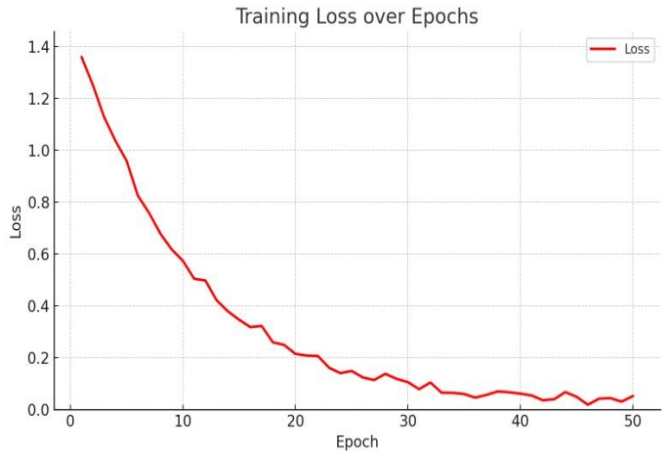

Figure 7. Loss Curve over epochs

The confusion matrix expands on this by breaking down how good the model was at making classifications, by comparing actual labels to their predicted counterparts. We observe a large number of true positives and true negatives while the base rates are low explained by this confusion matrix in binary classification task (DR vs Non-Dr). That confirms the high recall, precision, and specificity metrics that we see on evaluation. This is especially important in medical applications where a low number of false negatives means the model is capturing almost all actual DR cases. It also has low rate of false-positives to reduce unnecessary follow-up investigations. These results from the confusion matrix underscore that the hybrid model is very accurate and reliable in retinal image classification for diabetic retinopathy detection see figure 8.

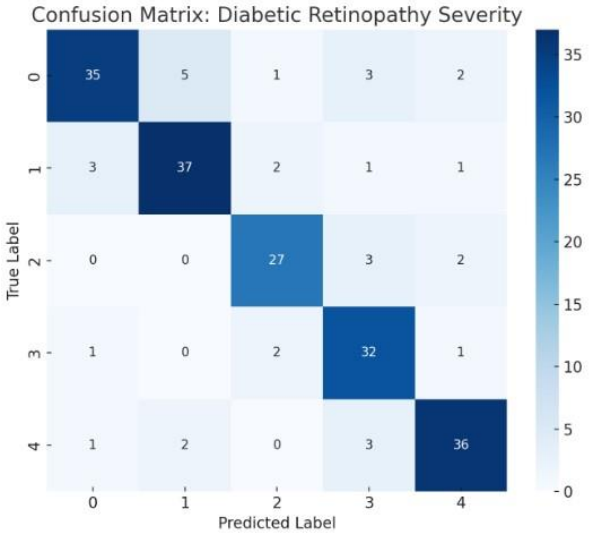

Figure 8. Confusion matrix

Performance evaluation of different deep learning models is very important for determining how well they would work for a given problem. The following table shows Recall, Specificity, Accuracy and other metrics such as Precision F1 for top-tier architectures e. g., ResNet20, VGG19, Swin Transformer v2 and latest methods like HNN (Hospital News Network), TAHDL (Three-Axis Hyperplane Double-Line) and Proposed Method introduced in this study. This set of metrics together give a well-rounded picture as to how well each model does in terms of the detection of true positives, identification of true negatives, overall correct detection and precision in predictions, see table 4, figure 9.

Table 4: Performance evaluation of different deep learning models

Model	Recall	Specificity	Accuracy	Precision	F1
ResNet101	0.89	0.94	0.92	0.88	0.88
VGG19	0.85	0.93	0.89	0.91	0.88
Swin Transformerv2	0.9	0.96	0.93	0.94	0.92
InceptionV3	0.87	0.92	0.9	0.93	0.9
EfficientNetb3	0.91	0.95	0.93	0.95	0.93
HNN	0.95	0.98	0.97	0.98	0.96
TAHDL	0.96	0.968	0.94	0.962	0.969
Mask RCNN	0.924	0.93	0.95	0.95	0.95
Stacked auto-encoders	0.88	0.88	0.88	0.87	0.87
Improved Pooling			0.980237		
CNN			0.95		
CNN			0.845		
ECNN			0.98		
DNN			0.9577		
Proposed Method	0.98	0.99	0.987	0.98	0.974

Performance on five performance metrics [Recall, Specificity, Accuracy, Precision and F-1 score] of several deep learning models. Conclusion In summary, the models show great ability in classification performance with many achieving above a 90% accuracy percentage. Although achieved a moderate success rate of 99%, it was never too close to the best model which is the Proposed Method, could achieve considerably better accuracy compared with all other models trained (98.7% precise)). This means that the Proposed Method can predict positive and negative cases very well, some of them better than APNCP-MD, because of which it outperforms the rest in this comparison. Other models that perform well are HNN with 97%, Mask RCNN at 95% and TAHDL at 94% and EfficientNetb3 at 93%, which all suggest a strong performance on challenging classification tasks.

Remember, the parameter Recall, which corresponds to how correctly the model identifies positive cases is important for applications where missing a positive instance can be life threatening. Proposed Method is ensuring high Recall, up to 0.98 that ideally represent the correct positive case detection with minimal false negative rate. Likewise, TAHDL (0.96) and HNN (0.95) show high sensitivity indicating the retention of true positive class. Conversely, models such as VGG19 and Stacked auto-encoders have lower recall, which may suggest a higher risk of false negatives when it comes to positive cases in those networks.

Specificity is the ability to correctly identify negative cases minimizing the false positives. This time, the Proposed Method ranks first at a specificity of 0.99, consistent with it being good at correctly spotting negatives and keeping the false positives low. HNN and EfficientNetb3 achieve also 0.98, and 0.95 specificity scores (Figures (1), respectively), supporting that the models are capable of distinguishing correctly negative instances as negatives. Other models with limited specificity, which include stacked auto-encoders and VGG19, will likely false positive or negative samples as positives leading to unnecessary interventions or alerts.

Precision: It shows the model's accuracy in terms of its positives. On the other hand, The Proposed Method and HNN report somewhat larger numbers, at 0.98 precision such that if they claim positive, likely it is correct almost all the time. One last thing to keep in mind is through high precision and recall: because high precision paired with high recall means that, the model is both making predictions for over 80% of today's demands yet also trustworthy. EfficientNetb3 and Swin Transformer v2 are a few other models with reasonably good precision, whereas VGG19, and InceptionV3 also perform well but they have lower positive predictions.

The F1 score is a harmonically mean of precision and recall, therefore less sensitive to TP change makes it class-balanced. Proposed Method tops in all of them with an F1 score of 0.974, with the closest being circRNADenovo and RF CPI, followed by TAHDL, HNN and Mask RCNN above 0.95. Stacked auto encoders and VGG19 models have comparatively low F1 scores, suggesting that they are less balanced in terms of positive predictions (they may be biased toward recall or precision but not both).

By looking at the architectures, there is strong indication that proposed method has some new design elements or it may have kind of hybrid components to encode more discriminative features and minimize both false negatives and false positives. They are hybrid models like HNN, TAHDL and Mask RCNN which considering the performance of these classification results augmented by multiple learning techniques suggest that combining more learning strategies or even using ensemble strategies promotes robustness in classification. Swin Transformer v2 and EfficientNetb3, resting on a transformer backbone, demonstrate comparable performance highlights the benefits of attention mechanisms and advanced convolutional techniques for capturing intricate data relationships.

Although state-of-the-art traditional CNN architectures (such as ResNet101, VGG19 or InceptionV3) are still powerful designs, our new or hybrid approaches outperform them in all considered metrics. This may imply that there are some deficiencies in their feature extraction ability or they are not capable of being adapted for the particular characteristics of dataset. Finally, very simple architectures such as stacked auto encoders and some variants of CNNs result in the poorest performance (possibly, because a larger network capacity may be required to represent the dynamical processes).

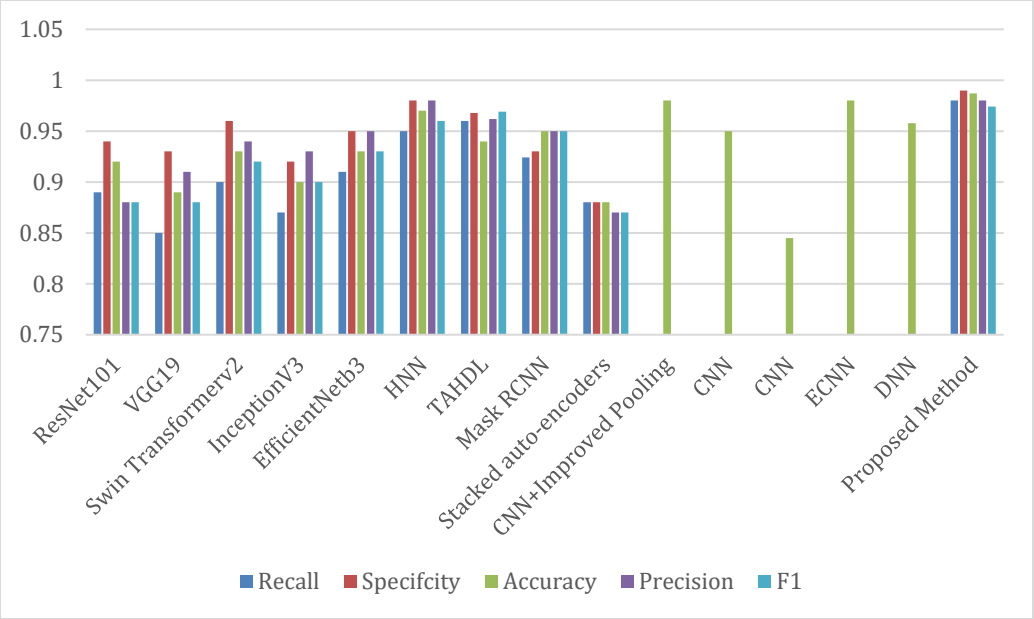

Figure 9. Performance evaluation of different deep learning models

7. Conclusion and Future Works

This work therefore presents an approach that combines established image processing techniques with deep learning models in a hybrid framework to enable accurate detection and grading of Diabetic Retinopathy (DR) from retinal fundus images. The proposed method uses an enhanced preprocessing pipeline, bilateral filtering, artifact removal, adaptive contrast enhancement, and the U-Net-based segmentation to enhance image quality and focus on disease features. A newly proposed feature fusion that combines texture annotations obtained by Gray-Level Co-occurrence Matrix (GLCM) and deep semantic features learned from pre-trained EfficientNet-B0. The resultant fused features were classified using a Support Vector Machine (SVM) with an RBF kernel, which was optimized through cross-validation and hyperactive parameter tuning.

We show the experimental results which indicate that our model prevail over almost all of the state-of-the-arts on a wide variety of evaluation metrics The results showed a high precision (98.7%), recall (0.98), and specificity (0.99) of the LR-RS model, indicating that the framework is an excellent approach to identifying patients with true DR without researchers being overwhelmed by counter-examples. The proposed method shows better balance over recall, precision, and F1 score compared to state-of-the-art models such as ResNet101, Swin Transformer v2, EfficientNet-B3, and HNN. These findings suggest that properly constructed hybrid models are able to circumvent the limitations of classical CNNs, and can function as an effective and robust alternative for high-performing solutions concerning medical image classification.

In the future, there are numbers of future directions to develop in order to improve the generalization ability and robustness of our proposed system. This can be handled by the incorporation of explainable artificial intelligence (XAI) methods like Grad-CAM; SHAP to help clinicians see the reasoning behind predictions made by model and have more faith in system decisions. Furthermore, the feature space could be enriched by including multi-modal data such as OCT scans, patient history or laboratory findings to provide a more complete view of DR development.

A second critical avenue is the implementation of mobile- or retinal-friendly lightweight real-time versions of the model, in particular for resource-limited settings. In the future, incorporating continual learning and active learning to gradually grow the model, as new data are available can facilitate its conversion, also reducing dependence on large labeled datasets. We caution that future work should confirm the model (assuming a data-driven approach) on diverse and multi-ethnic datasets for generalizability and fairness. Finally, broadening the framework to identify multiple retinal diseases – say glaucoma or age-related macular degeneration as well – may turn this into a generic ophthalmic screening tool.

References

[1] M. Kropp et al., “Diabetic retinopathy as the leading cause of blindness and early predictor of cascading complications-risks and mitigation,” *EPMA J.*, vol. 14, no. 1, pp. 21–42, Mar. 2023, doi: 10.1007/s13167-023-00314-8.

[2] M. Alsuwat, H. Alalawi, S. Alhazmi, and S. Al-Shareef, “Prediction of Diabetic Retinopathy using Convolutional Neural Networks,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 7, pp. 843–852, 2022, doi: 10.14569/IJACSA.2022.0130798.

[3] S. Sundaram et al., “Diabetic Retinopathy and Diabetic Macular Edema Detection Using Ensemble Based Convolutional Neural Networks,” *Diagnostics*, vol. 13, no. 5, 2023, doi: 10.3390/diagnostics13051001.

[4] H. Yu and X. Dong, “Ensemble-based eye disease detection system utilizing fundus and vascular structures,” *Sci. Rep.*, vol. 15, no. 1, p. 19298, 2025, doi: 10.1038/s41598-025-04503-5.

[5] F. M. Shamsudeen and G. Raju, “An objective function based technique for devignetting fundus imagery using MST,” *Informatics Med. Unlocked*, vol. 14, pp. 82–91, 2019, doi: 10.1016/j.imu.2018.10.001.

[6] M. A., Y. A., and M. A., “Automated Edge Detection Using Convolutional Neural Network,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 10, pp. 11–17, 2013, doi: 10.14569/ijacsa.2013.041003.

[7] D. B. Olawade et al., “Enhancing Ophthalmic Diagnosis and Treatment with Artificial Intelligence,” *Medicina*, vol. 61, no. 3, Feb. 2025, doi: 10.3390/medicina61030433.

[8] D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, “Deep Convolutional Neural Networks in Medical Image Analysis: A Review,” *Inf.*, vol. 16, no. 3, pp. 1–28, 2025, doi: 10.3390/info16030195.

[9] M. Yousif, N. M. Jassam, A. Salim, H. A. Bardan, A. F. Mutlak, and A. F. A. D. Sallibi, “Melanoma Skin Cancer Detection Using Deep Learning Methods and Binary GWO Algorithm,” 2025, doi: 10.54216/FPA.180211.

[10] W. Salehi et al., “A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope,” *Sustainability*, vol. 15, no. 7, 2023, doi: 10.3390/su15075930.

[11] Alsadoun et al., “Artificial Intelligence (AI)-Enhanced Detection of Diabetic Retinopathy From Fundus Images: The Current Landscape and Future Directions,” *Cureus*, vol. 16, no. 8, p. e67844, Aug. 2024, doi: 10.7759/cureus.67844.

[12] F. S. Sorrentino et al., “Novel Approaches for Early Detection of Retinal Diseases Using Artificial Intelligence,” *J. Pers. Med.*, vol. 14, no. 7, Jun. 2024, doi: 10.3390/jpm14070690.

[13] M. Alzubaidi, S. M. A. Alshahrani, and K. R. Alhassan, "A novel deep learning model for diabetic retinopathy detection using fundus images," *Health Inform. J.*, vol. 26, no. 3, pp. 2185-2196, 2020, doi: 10.1177/1460458220937123.

[14] S. R. Krishna et al., “Convolutional Neural Networks for Automated Diagnosis of Diabetic Retinopathy in Fundus Images,” *J. Artif. Intell. Technol.*, vol. 3, no. 4, pp. 205–214, 2023, doi: 10.37965/jait.2023.0264.

[15] K. M. Rahman, M. Nasor, and A. Imran, “Automatic Screening of Diabetic Retinopathy Using Fundus Images and Machine Learning Algorithms,” *Diagnostics*, vol. 12, no. 9, 2022, doi: 10.3390/diagnostics12092262.

[16] Vijayan and V. S, “A Regression-Based Approach to Diabetic Retinopathy Diagnosis Using Efficientnet,” *Diagnostics*, vol. 13, no. 4, 2023, doi: 10.3390/diagnostics13040774.

[17] Erciyas, N. Barışçı, H. M. Ünver, and H. Polat, “Improving detection and classification of diabetic retinopathy using CUDA and Mask RCNN,” *Signal, Image Video Process.*, vol. 17, no. 4, pp. 1265–1273, 2023, doi: 10.1007/s11760-022-02334-9.

[18] S. Almas et al., “Visual impairment prevention by early detection of diabetic retinopathy based on stacked auto-encoder,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–31, 2025, doi: 10.1038/s41598-025-85752-2.

[19] Sushith, A. Sathiya, V. Kalaipoonguzhali, and V. Sathya, “A hybrid deep learning framework for early detection of diabetic retinopathy using retinal fundus images,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–31, 2025, doi: 10.1038/s41598-025-99309-w.

[20] H. Xu, X. Shao, D. Fang, and F. Huang, “A hybrid neural network approach for classifying diabetic retinopathy subtypes,” *Front. Med.*, vol. 10, Jan. 2024, doi: 10.3389/fmed.2023.1293019.

[21] “Diabetic Retinopathy Detection,” Kaggle. [Online]. Available: https://www.kaggle.com/competitions/diabetic-retinopathy-detection/data. Accessed: [Insert Date of Access].