Feature Selection Using Adaptive Mutation Probability Based Starfish Optimization Algorithm in Diabetic Retinopathy Classification

B. R. Raghu [1,*], Janapati Venkata Krishna [2], and N. Pradeep [3]

[1] Department of Computer Science and Engineering, Srinivasa University, Mangalore, India
[2] Department of Computer Science, Institute of Engineering and Technology, Mangalore, India
[3] Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davangere, India
Email: br.raghu585@gmail.com (B.R.R.); venkatakrishna.janapati@gmail.com (J.V.K.);
nmnpradeep@gmail.com (N.P.)
*Corresponding author

***Abstract*—Diabetic Retinopathy (DR) is an ophthalmic complication of diabetes, which is one of the leading causes of vision impairment. Accurate grading of DR severity is essential for early diagnosis and timely intervention. However, manual diagnostic methods are time-consuming, resource-intensive, and error-prone. Contrarily, Computer-Aided Diagnosis (CAD) systems offer a faster, more consistent, and potentially more accurate alternative, while facilitating effective screening and treatment planning. This research proposes the feature selection approach of Adaptive Mutation Probability based Starfish Optimization Algorithm (AMP-SFOA) for automatic diagnosis and classification of DR into different categories. The proposed method is evaluated using two standard dataset, Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (ATPOS-2019) and Diabetic Retinopathy Detection (DDR) based on training and testing. Preprocessing steps, including image resizing, Contrast Limited Adaptive Histogram Equalization (CLAHE), and data augmentation, are performed to enhance the input data. Finally, the Long Short-Term Memory (LSTM) is employed for classifying DR into multi-classes. Experimental results demonstrate that the proposed AMP-SFOA approach attains better accuracies of 99.476% and 96.958% on the APTOS-2019 dataset and the DDR dataset respectively, compared to existing methods such as RT2Net and Swin-TransformerV2 Multi-Branch Fine-grained Diabetic Retinopathy Grade Classification (STMF-DRNet).**

***Keywords*—adaptive mutation probability, contrast limited adaptive histogram equalization, diabetic retinopathy, long short-term memory, starfish optimization algorithm**

I. Introduction

Diabetic Retinopathy (DR) is a retinal vascular disease that occurs at progressive levels of diabetes. Persistently greater blood sugar levels damage small blood vessels all over the body, including the eyes, leading to blood and fluid leakage into the retina and ultimately the development of diabetic retinopathy [1]. The disease initially causes swollen vessels and fatty deposits in the retina and progresses to more severe changes, such as the growth of new and abnormal blood vessels. These changes lead to symptoms like floating spots in vision, blurry vision, and enduring vision impairment [2]. Proliferative Diabetic Retinopathy (PDR) and Non-PDR (NPDR) are the two major types of DR [3, 4]. NPDR is the early stage and is categorized as mild, moderate, and severe stages based on the extent of blood vessel damage [5]. PDR is the advanced stage, categorized through abnormal new blood vessel development on the retinal surface, which can bleed and lead to vision problems [6]. Diagnosis relies on identifying characteristic retinal lesions across different severity levels. These lesions include soft exudates, hemorrhages, macular edema, microaneurysms, hard exudates, and neovascularization, which are primary indicators used to assess disease progression and severity [7, 8].

Accurate detection and grading of diabetic retinopathy are crucial for preventing the progression of blindness [9], emphasizing the need for reliable and effective diagnostic methods [10, 11]. With advancements and applications of Artificial Intelligence (AI) technology, Deep Learning (DL) has become increasingly effective in the medical image processing field [12]. Traditional Machine Learning (ML) approaches for diabetic retinopathy diagnosis rely on manually designed features like color and texture patterns [13]. However, these handcrafted features fail to capture the complexity and variability in retinal images, thereby limiting their effectiveness, especially when applied to large-scale clinical datasets [14–16]. Conversely, DL methods are effective to automatically learn spatial features directly from retinal images, which enables the identification of pathological structures such as hemorrhages, microaneurysms, and exudates [17–20]. DL also handles large data volumes effectively, thereby enhancing model effectiveness and supporting end-to-end

Manuscript received October 25, 2025; revised November 17, 2025; accepted January 26, 2026; published April 28, 2026.

learning that reduces error and improves diagnosis consistency.

II. Literature Survey

Xu *et al*. [21] introduced a multi-view joint learning framework for diabetic retinopathy called RT2Net, which integrated local and global features of fundus images. Primarily, the obtained images were pre-processed through Contrast Limited Adaptive Histogram Equalization (CLAHE), and the vascular structures were segmented. Subsequently, fundus images and vascular images were fed into different branch networks of RT2Net feature extraction and fusion. Eventually, an optimized classification approach was employed to determine five classes of diabetic retinopathy accurately. However, RT2Net struggled with accurately aligning local features because of variations in position and scale over different views, thereby making feature alignment difficult.

Nazir *et al*. [22] presented a weighted average ensemble method by integrating Convolutional Neural Network (CNN), Inception V3, and VGG16. CNN extracted features and ensured layer-by-layer subtraction by stacking pooling, convolution, and a non-linear activation function. The output was passed through a J48 classifier for final prediction. VGG16 and Inception V3 were used as the second and third models, and their extracted features were fed to SoftMax for each model. Finally, a weighted average ensemble was trained based on weights and was given to three models for classification. Nevertheless, a weighted average ensemble failed to capture feature interactions, because of which an independent decision-level fusion relied heavily on optimal weight selection, which led to suboptimal performance.

Alam *et al*. [23] developed DiabSense, which incorporated Human Activity Recognition (HAR) and diabetic retinopathy by leveraging two different Graph Neural Networks (GNNs). A Graph Attention Network (GAT) was utilized to capture internal connections in a time-series data, which offered computational efficiency and handled node significance while processing on a varied neighborhood size without the prior knowledge of a graph structure. However, GNN struggled with over-smoothing, which impacted the node representations to become vague and reduced the approach's ability to capture the subtle and fine-grained DR lesion.

Butt *et al*. [24] established a memory-efficient method for classifying diabetic retinopathy using a CNN named Memory-Efficient Diabetic Retinopathy Classification Network (MEDCNet). Initially, CLAHE was used for the improvement of image quality, and then high-resolution features were extracted using the 'divide and conquer' method from fundus images. Subsequently, these features were fed into diverse ML approaches like Support Vector Machine (SVM), Random Forest (RF) and Multi-Layer Perceptron (MLP) to perform classification, which achieved an accurate and reliable diagnosis. Nevertheless, MEDCNet faced challenges in feature richness due to an aggressive parameter reduction that led to less sensitivity for detecting DR lesions.

Herrero-Tudela *et al*. [25] suggested a visual Explainable AI (XAI) method using Shapley Additive exPlanations (SHAP) for classifying diabetic retinopathy. Initially, cropping was performed to eliminate external background borders, which ensured that the retinal diameter became equal to the width of the image and terminated unwanted information. Subsequently, data augmentation was applied to enhance the dataset, and then different CNNs with backbone architecture was also used for classifying diabetic retinopathy. However, SHAP provided pixel-level attributions without capturing spatial dependencies, which limited the model's ability to fully determine lesion-based decisions.

Liu *et al*. [26] introduced a Swin-TransformerV2 Multi-Branch Fine grained Diabetic Retinopathy Grade Classification (STMF-DRNet). Initially, Swin transformer V2 was integrated with a hybrid attention mechanism to extract lesion features. Subsequently, a multi-branch cascading model was used to integrate features at different scales that optimized the retention of lesion-related data and enhanced the accuracy. Also, a category attention mechanism was employed to expose large discrete regional features, which enhanced the network's ability to detect lesions accurately. Nevertheless, STMF-DRNet demanded a large annotated dataset for stable training because of the heavy parameterization of a transformer model that led to overfitting.

III. Proposed Methodology

This research concentrates on the automatic DR detection by proposing Adaptive Mutation Probability based Starfish Optimization Algorithm (AMP-SFOA) with Long Short-Term Memory (LSTM) approach, which comprises of different important stages like data collection, image pre-processing, and classification. The LSTM approach classifies an input image into multiple categories. Fig. 1 demonstrates the complete framework of the proposed method.

A. Dataset

The proposed methodology is evaluated using two various phases, which are the Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (ATPOS-2019) [1] and Diabetic Retinopathy Detection (DDR) [2] dataset to assess effectiveness on the varied quality fundus images. Detailed information about the dataset is explained below.

1) APTOS2019

It has 3662 training images and 1992 testing images that cover different stages of diabetic retinopathy. The dataset is annotated by professional experts based on the International Clinical DR Disease Severity Scale (ICDRSS) by categorizing images into five grades between 0 and 4.

2) DDR

It contains 13,673 fundus images gathered from 9598

[1] https://www.kaggle.com/datasets/mariaherrerot/aptos2019.

[2] https://www.kaggle.com/datasets/mariaherrerot/ddrdataset

patients from 147 hospitals in China, covering 23 provinces. The images are split into five classes as per the severity scale, such as no DR, mild DR, moderate DR, severe DR, and PDR. The dataset is split into 80% training and 20% testing, respectively. Table I represents the dataset description, and Fig. 2 depicts sample fundus images with different grades. To improve clarity, representative images from both APTOS-2019 and DDR datasets have been included in Fig. 2. These images illustrate the visual appearance of different DR grades and highlight the variability in illumination, lesion density, and image quality present in both the dataset. The obtained images are passed through the pre-processing stage for further processing.

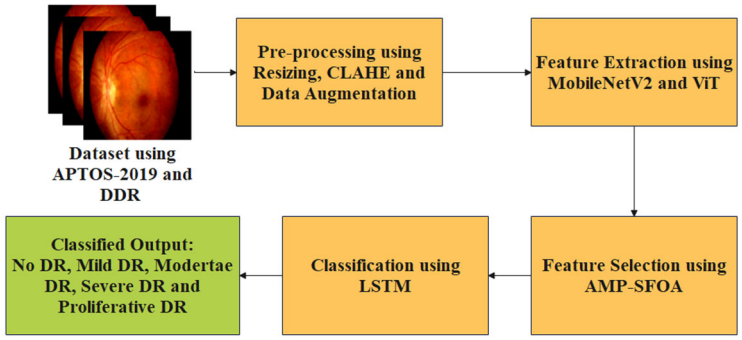


Fig. 1. Workflow of the proposed method.

TABLE I. DISTRIBUTION OF IMAGE PER CLASS IN APTOS 2019 AND DDR

Class	Severity	No. of Samples	
		APTOS 2019	DDR
0	No DR	1805	6266
1	Moderate DR	999	4477
2	Mild DR	1857	6256
3	PDR	295	913
4	Severe DR	193	236

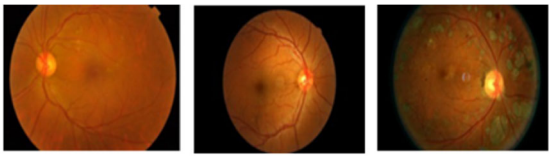


Fig. 2. Sample fundus images.

B. Pre-processing

After obtaining images, resizing, CLAHE and data augmentation are used to ensure uniform input dimension, enhance local contrast, and increase training diversity. The brief description of these methods is explained below in detail.

1) Resizing and CLAHE

The fundus images from the APTOS 2019 dataset and the DDR dataset are resized to 256×256 to ensure a uniform input dimension for the proposed method. After resizing, CLAHE is used to enhance local contrast in fundus images, which increases the visibility of subtle lesions through amplifying the differences in lighter and darker regions without over-amplifying noise. It interpolates the enhancement results for smoothing the boundaries among local blocks and limits noise amplification. Finally, each local blocks are integrated into a new image. Thus, the CLAHE preserves more local information than other histogram equalization processes.

2) Data augmentation

Data augmentation becomes an important part in improving the effectiveness of the model, especially in deep learning-based image analysis. This technique involves creating modified versions of the existing training samples through a variety of transformations, thereby expanding dataset diversity and enhancing model generalization. To this end, several augmentation operations were applied, including rescaling, rotation, translation, shearing, zooming, horizontal flipping, and fill mode adjustments. Specifically, a rotation limit of 45 degrees, width and height shifts up to 30%, shear and zoom ranges set at 0.3, and horizontal flipping were employed. The 'reflect' fill mode was used to seamlessly manage pixel interpolation of a pixel during geometric changes. In addition, brightness levels were randomly varied between 0.5 and 1.5 to simulate real-world lighting inconsistencies and enhance visual robustness.

To prevent data leakage, the dataset was first divided into training (80%) and testing (20%) partitions. All augmentation operations were used significantly for the training subset. No augmented image corresponding to a training sample appears in the testing set, which ensures unbiased estimation.

C. Feature Extraction

This research considers two different feature extraction methods, known as MobileNetV2 and Vision Transformer (ViT) for extracting relevant features. Between them, the MobileNetV2 is utilized for the extraction of local features, while the ViT is utilized for global feature extraction. The details of these methods are provided below.

1) MobileNetV2

MobileNetV2 is designed with efficiency and accuracy in mind, featuring core components such as depth-wise separable convolutions, inverted residuals, and linear bottlenecks. These structural innovations reduce computational complexity and preserve learning capacity. In this work, MobileNetV2 is adapted to serve as a lightweight yet effective local feature extractor. The model's original output layer is replaced with a custom dense layer that is configured to produce a 128-

dimensional feature vector. This fixed-length embedding provides a uniform, information-rich representation that supports consistent similarity computation and facilitates deployment in real-time biomedical imaging systems.

2) Vision transformer

ViT represents a shift from traditional convolution-based models by leveraging self-attention mechanisms to model spatial dependencies across entire images. In this method, each input image is partitioned into equal-sized patches, which are then flattened and projected into a latent space through linear embedding. Positional encodings are added to retain spatial information, and the resulting sequence of tokens is passed through a transformer encoder. This architecture enables ViT to capture long-range interactions and semantic patterns, making it most appropriate for tasks that require a global understanding of visual content.

3) Feature fusion

When both networks process input images, they independently generate feature representations in the form of vector embeddings. These feature vectors—128-dimensional from MobileNetV2 and high-dimensional tokens from ViT—are integrated using a concatenation strategy. This fusion approach aggregates local and global information into a unified feature space, thereby creating a comprehensive representation of the input. The resulting fused vector enables the downstream classification or retrieval modules to leverage both fine-grained textures and broader structural cues for improved accuracy. In this work, the MobileNetV2 CNN extracts 1280 local texture-based features, while the ViT generates 192 global contextual embeddings. These two sets are concatenated to form a fused feature vector of 1742 dimensions. The AMP-SFOA algorithm evaluates each feature using its fitness function that balances classification error and feature-reduction ratio to retain only the most discriminative descriptors. As a result, 563 highly informative features are selected, while 1179 low-variance, redundant, or noise-prone features are automatically discarded. This optimized subset preserves the strongest lesion-related information required for accurate DR severity prediction.

D. Feature Selection Using SFOA

The innovation of the proposed AMP-SFOA lies in the incorporation of an adaptive mutation probability mechanism and a Gbest-guided refinement strategy, both of which are absent in traditional SFOA and other swarm-based optimizers. As compared to traditional SFOA, which utilizes the fixed exploration and exploitation behavior, AMP-SFOA dynamically changes mutation intensity in terms of iteration count, while enabling early wide exploration and late-stage fine exploitation. Additionally, the Gbest-guided directional search improves the convergence stability and minimizes local stagnation. These incorporated mechanisms variate the AMP-SFOA from existing metaheuristics like Grey Wolf Optimization (GWO), Whale Optimization Algorithm (WOA), Penguin Optimization Algorithm (POA), and classical SFOA. Based on the unique foraging and adaptive behaviors of a starfish, this research introduces a novel swarm intelligence algorithm, termed the Starfish Optimization Algorithm (SFOA), which is designed to address complex global and engineering optimization challenges. This section presents biological motivation, mathematical formulation, algorithmic workflow, and computational complexity analysis of SFOA.

1) Initialization

The initialization phase of SFOA begins by generating initial positions of all starfish individuals within the predefined search space. Each starfish's position is initialized randomly, bounded by the minimum and maximum limits of the respective design variables. The position of each starfish is shown in Eq. (1) as trails:

$$X_{ij} = l_j + r\left(u_j - l_j\right),\ \ i = 1, 2, ..., N,\ \ j = 1, 2, ..., D \tag{1}$$

where, X_{ij} denotes the j th dimension position of i th starfish; r specifies the arbitrary number between 0 and 1; u_j and l_j specify the upper and lower bounds of the design variables in j th dimension, respectively.

After generating an initialized matrix of positions, the fitness values of all starfish are obtained through the estimation from an objective function, as in Eq. (2):

$$F = \begin{bmatrix} F(X_1) \\ F(X_2) \\ \cdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{2}$$

where F denotes a matrix to store and update the acquired fitness values, through the size of $N\times 1$.

After an initialization, SFOA arrives to the main loop, and begins exploration and exploitation phases.

2) Exploration

Following the initialization of positions, the algorithm evaluates the value of G_p against a randomly generated number within the interval (0, 1). If this number is less than or equal to G_p, then the algorithm proceeds into the exploration phase. At this stage, a parameter d is computed to guide the exploration behavior. If $d > 5$, the starfish activates its five arms to probe the surrounding search space, thereby mimicking its biological foraging strategy. Subsequently, the position of each starfish is updated based on the best-known information within an agent's memory. This exploratory update mechanism is mathematically formulated in Eq. (3).

$$\begin{cases} P_{i,p}^t = x_{i,p}^t + a_1\left(x_{best,p}^t - x_{i,p}^t\right)cos\theta,\ rand < 0.5 \\ P_{i,p}^t = x_{i,p}^t + a_1\left(x_{best,p}^t - x_{i,p}^t\right)sin\theta,\ rand > 0.5 \end{cases} \tag{3}$$

where, $P_{i,p}^t$ specifies the updated position; $x_{i,p}^t$ and $x_{best,p}^t$ denote the present position and the best position of dimension p correspondingly; p defines five arbitrarily chosen dimensions of dimension d. a_1 and θ are formulated

in Eqs. (4) and (5).

$$a_1 = (2r-1)\pi \tag{4}$$

$$\theta = \frac{\pi}{2} \cdot \frac{t}{t_{\max}} \tag{5}$$

where, t and t_{max} specifies the present and maximum iteration count correspondingly.

At this stage, a_1 is arbitrarily produced and utilized to update the position, while θ differs by the number of iterations. These two parameters jointly evaluate the influence of the distance between the current position and the optimal position along with the selected update dimension. If the updated position exceeds defined boundary constraints, then the algorithm retains previous position instead of applying out-of-bounds the update. This boundary-handling mechanism ensures solution feasibility and is mathematically represented in Eq. (6).

$$x_{i,p}^{t+1} = \begin{cases} P_{i,p}^t & l_b \le P_{i,p}^t \le l_b \\ x_{i,p}^t & else \end{cases} \tag{6}$$

where, p means update dimension.

If $d \le 5$, one arm of the starfish will move to search for food sources. The updated position is formulated in Eq. (7).

$$P_{i,q}^t = E_t x_{i,p}^t + b_1 \left(x_{k1,p}^t - x_{i,p}^t\right) + b_2 \left(x_{k2,p}^t - x_{i,p}^t\right) \tag{7}$$

where, $x_{k1,p}^t$ and $x_{k2,p}^t$ denote dimensional positions of two selected starfishes, b_1 and b_2 as numbers in the range (1, −1), and p means the number of randomly selected in d dimensions. E_t denotes the energy of a starfish, which is acquired through Eq. (8).

$$E_t = \frac{t_{\max} - t}{t_{\max}} \cos\theta \tag{8}$$

3) Exploitation

At the development phase, binary update mechanisms were integrated into the SFOA framework, which is one of the key mechanisms that involves a parallel bidirectional search strategy that leverages both the current optimal position and the positional information from other individuals of the population. Initially, five distances are computed between the current best position and the positions of five randomly selected starfish. From these, two distances are randomly chosen to serve as guiding references. Using this information, the position of each starfish is updated through a parallel bidirectional adjustment process. The calculation of these distances is defined in Eq. (9) as follows:

$$d_o = \left(x_{best}^t - x_{o_p}^t\right), \quad o = 1,2,3,4,5 \tag{9}$$

where, d_o denotes a distance among the best location and other starfish; o_p denotes five arbitrarily selected starfish.

Therefore, a location update rule for each starfish for the predation behavior is acquired through Eq. (10).

$$P_i^t = x_i^t + r_1 d_{o1} + r_2 d_{o2} \tag{10}$$

where r_1 and r_2 demonstrate arbitrary numbers between 0 and 1; d_{o1} and d_{o2} illustrate the arbitrarily chosen variables from d_o.

Naturally, starfish are susceptible to predator attacks during foraging. As a defense mechanism, a starfish may detach one of its arms to escape, later regenerating it over time. This biological behavior is modeled in the SFOA in the regeneration phase, which is applied exclusively to the last-ranked starfish in the population. This strategy promotes diversity and prevents stagnation by allowing poorly performing individuals to reposition based on adaptive regeneration. The positional update during this phase is governed by the formulation presented in Eq. (11).

$$P_i^t = \exp\left(-t \times n / t_{\max}\right) x_i^t \tag{11}$$

For out-of-bounds positions, the update rule is utilized, as shown in Eq. (12).

$$x_i^{t+1} = \begin{cases} P_i^t, & l_b \le P_i^t \le u_b \\ l_b, & P_i^t < l_b \\ u_b, & P_i^t > u_b \end{cases} \tag{12}$$

4) Gbest-guided mechanism

In a conventional SFOA approach, the unknown location of the candidate is generated by moving from an arbitrarily selected location within the population towards another arbitrarily selected position. It is important to note that the likelihood of a randomly selected position being suboptimal is equal to that of it being beneficial. Consequently, the new candidate position, which is generated via random sampling, may not necessarily outperform the previous one. However, the revised search equation introduces two significant improvements. First, the inclusion of a guiding term that enhances the algorithm's exploitation capability, thereby increasing the probability of identifying a superior solution. Second, since the vector Z_K is used to generate the candidate position, which is randomly chosen from the entire population, it introduces directionless exploration while maintaining a broad search capability. Taken together, these modifications strike a balance between exploration and exploitation. The complete updated search strategy is mathematically defined in Eq. (13) as follows:

$$\overrightarrow{Z(t+1)} = \begin{cases} Z_{k1} + F \times \left(Z_{k2} - Z_{k3}\right), & rand < z \\ \overrightarrow{Z_K(t)} + \overrightarrow{c_1} & \\ \left(\vec{G} \cdot \overrightarrow{Z_{best}(t)} - \overrightarrow{Z_K(t)}\right), & r < p \\ \overrightarrow{c_2} \cdot \overrightarrow{Z(t)}, & r \ge p \end{cases} \tag{13}$$

where, K denotes the uniform integer selected in a range $[1, N]$, $\overrightarrow{Z_{best}(t)}$ specifies the present optimal candidates at iteration t.

5) Adaptive mutation probability

When the dynamic random search technique exhibits strong capability in exploiting high-quality solutions within the search space, it often becomes susceptible to local optima while dealing with complex, multimodal, or high-dimensional problems. To enhance the approach's robustness and maintain population diversity, this research presents an adaptive mutation probability mechanism. By leveraging the nonlinear characteristics of the exponential function, a low mutation probability is applied in the early iterations to encourage broad exploration, thereby enabling the algorithm to discover promising regions of the solution space. In contrast, a higher mutation probability is employed in later iterations to facilitate an effective escape from local optima. Additionally, to constrain the mutation probability within a reasonable range, a ratio of 1/3 is adopted for dynamic control. Then, the mutation strategy is designed, as shown in Eqs. (14) and (15) as follows:

$$\overrightarrow{Z_{temp}(t+1)} = \begin{cases} rand() \cdot (u_b - l_b) + l_b, & rand < P \\ \overrightarrow{Z(t+1)}, & rand \geq P \end{cases} \tag{14}$$

$$\overrightarrow{Z(t+1)} = \begin{cases} \overrightarrow{Z_{temp}(t+1)}, if\ f\left(\overrightarrow{Z_{temp}(t+1)}\right) < f\left(\overrightarrow{Z(t+1)}\right) \\ \overrightarrow{Z(t+1)}, \quad otherwise \end{cases} \tag{15}$$

where, P denotes the mutation probability.

Now, a greedy selection is obtained to decide whether the newly generated solution is better than the prior one. Hence, as the probability improves, the excellence of the solution is improved. In the proposed AMP-SFOA, the fitness function is illustrated in Eq. (16).

$$F = w_1 \times E_{cls} + w_2 \times \frac{K_{selected}}{K_{total}} \tag{16}$$

where, E_{cls} is a classification error; $K_{selected}$ is the number of selected features, and K_{total} illustrates the total count of extracted features. The weighting parameters w_1 and w_2 are determined through an ablation-based empirical analysis.

Several combinations (0.5/0.5, 0.6/0.4, 0.7/0.3 and 0.8/0.2) are evaluated on a validation subset. Based on empirical tuning, $w_1 = 0.7$ and $w_2 = 0.3$ are selected, which achieved the best trade-off between classification accuracy and feature reduction, thereby yielding superior performance and generalization. Therefore, this configuration is adopted for all experiments. The AMP-SFOA algorithm evaluates each feature by using its fitness function that balances classification error and feature-reduction ratio to retain only the most discriminative descriptors. As a result, 563 highly informative features are selected, while 1179 low-variance, redundant, or noise-prone features are automatically discarded, because they show low variance, greater redundancy, or noise contamination. Recollecting these features will decrease classifier performance, enhance computation, and cause overfitting. Only 563 features that contribute expressive discriminatory information are preserved. This optimized subset preserves the strongest lesion-related information required for accurate DR severity prediction. This vector is reshaped into a sequence of length 28 with feature width 32, while also enabling temporal modeling using LSTM.

E. Classification Using Long Short-Term Memory

Although LSTM networks are traditionally applied to sequential or temporal data, their gating mechanism is highly effective for modeling structured feature dependencies. After AMP-SFOA selects the most discriminative 563 features, the resulting vector is reshaped into a sequence of length 28 with a feature width of 32. This structured sequential representation allows the LSTM to understand long-range dependencies among feature groups, model the hierarchical correlations, and reduce sensitivity to noise and feature redundancy. This temporal modeling capability enables LSTM to outperform conventional fully connected classifiers by understanding large relationships among lesion-related feature clusters, while improving DR grade discrimination. As compared to CNNs, which treat the 1-D feature vector as independent local neighborhoods and lack memory over feature positions, LSTMs maintain contextual memory across the complete sequence. This enables LSTM to model cross-feature dependencies more effectively than CNNs, resulting in greater DR grade discrimination and more consistent classification performance. Compared to a traditional Multi-Layer Perceptron (MLP), which considers features as independent inputs, LSTM utilizes its gated memory mechanism for capturing contextual relationships among feature groups. This architectural property designs LSTM more appropriate for learning from reshaped AMP-SFOA-selected feature sequences, thereby offering a principled basis for its adoption in the proposed framework. The LSTM network comprises three significant layers such as input, hidden, and output. The input layer's size corresponds to the number of features in each data sample, while the output layer's configuration is defined by the number of target variables. This flexibility makes LSTM effective, particularly for multivariate time-series forecasting tasks. The key innovation of LSTM lies in its memory cells, embedded within the hidden layers, which enable the network to retain and manipulate the information over time. Each memory cell includes three essential gates namely forget f_t, input i_t and output o_t, where t refers to timestep. These gates regulate the data flow by updating the internal cell states s_t, thereby allowing an approach to learn long-term dependencies in sequential data. This mechanism allows the LSTM approach to get adapted to temporal changes, while making them highly suitable for time-dependent applications. At each time step t, the LSTM network follows a series of operations to update its internal state.

Step 1: At this stage, the cell evaluates the information from the previous state s_{t-1} to determine which elements should be discarded. Inputs are passed through a sigmoid

activation function, which scales the values between 0 and 1. These values serve as control signals, thereby guiding the extent to which previous information is retained or forgotten. The activation of the forget gate f_t is computed using Eq. (17) as follows:

$$f_t = \sigma\left(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f\right) \tag{17}$$

where, σ means sigmoid activation function, $W_{f,x}$ and $W_{f,h}$ denote weight matrices, the input vector at timestep t is specified as x_t, the output of the previous timestep ($t-1$) is represented using $h+t-1$ and b_f means forget gate bias vector.

Step 2: Inputs are evaluated using evaluations, and data is added to the cell state s_t using a *sigmoid* function. Furthermore, input data is projected to a [−1, 1] range via the *tanh* function. Different variables are the needs to be computed in this stage, candidate values $\tilde{s}_t$ and activation value i_t for every input gate. These are identified through Eqs. (18) and (19) as follows:

$$\tilde{s} = \tanh\left(W_{\tilde{s},x}x_t + W_{\tilde{s},h}x_t + b_{\tilde{s}}\right) \tag{18}$$

$$i_t = \tanh\left(W_{i,x}x_t + W_{i,h}x_t + b_i\right) \tag{19}$$

where, $b_{\tilde{s}}$ and b_i demonstrates bias vectors. $W_{\tilde{s},x}x_t$, $W_{\tilde{s},h}x_t$, $W_{i,x}x_t$ and $W_{i,h}x_t$ denotes weight indices.

Step 3: New cell states s_t are estimated through Eq. (20).

$$s_t = f_t \circ s_{t-1} + i_t \circ \tilde{s}_t \tag{20}$$

where, ° specifies the Hadamard product.

Step 4: Outputs of h_t are identified by *sigmoid* and *tanh* activation functions paired based on Eqs. (21) and (22).

$$o_t = \sigma\left(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o\right) \tag{21}$$

$$h_t = o_t \circ \tanh\left(s_t\right) \tag{22}$$

where, $W_{o,x}$ and $W_{o,h}$ denote weight matrices.

By integrating these mechanisms, the LSTM network effectively retains short-term memory through its three-gate architecture. Additionally, the internal cell state enables past information to stimulate future predictions, thereby making LSTM particularly well-suited for modeling and forecasting multivariate time-series data. Algorithm 1 demonstrates the pseudocode of the complete procedure of proposed the DR classification approach for better reproducibility. To ensure reproducibility, the comprehensive pseudocode has been standardized to match each algorithmic step utilized in AMP-SFOA. All variable names, operators, and procedures have been arranged through the final implementation.

Pseudo code

Input: Retinal Fundus Dataset (APTOS-2019, DDR)
Output: DR Class Labels (0: No_DR, 1: Mild, 2: Moderate, 3: Severe, 4: Proliferative_DR)
1. Load Dataset:
  Load images and labels from APTOS-2019 and DDR dataset
  For each dataset:
    Read CSV metadata
    Map image filenames to DR severity labels (0 to 4)
2. Preprocessing:
  For each image:
    Resize image to 256×256 pixels
    For each RGB channel:
      Apply CLAHE for contrast enhancement
    Merge enhanced channels into final image
    Save image to corresponding class folder
3. Data Augmentation:
  For each preprocessed image:
    Apply random rotation (−10° to +10°)
    Apply horizontal and vertical flipping
    Apply translation (±5 pixels)
    Resize image to 256×256 pixels
    Save augmented image to class folder
4. Feature Extraction:
  For each image:
    Extract features from MobileNetV2 (224×224 input)
      → Use Global Average Pooling layer output
    Extract features from ViT (384×384 input)
      → Use [CLS] token embedding
    Concatenate MobileNetV2 and ViT features
    Save combined feature vector
5. Feature Selection using AMP-SFOA:
  Initialize population of starfish agents using random positions within the feature search space
  For iter = 1 to max_iterations:
    For each agent:
Apply AMP-SFOA exploration strategy to generate candidate feature subsets
      Perform exploration to find new feature subset
      Perform exploitation (hiding) for local refinement
      Evaluate fitness:
        fitness = w_1×Classification_Error + w_2×(Selected_Features / Total_Features)
    Update agents based on best fitness
  Return optimal feature subset
6. Classification with Deep Models (5-Fold CV):
  For each fold in 5-fold cross-validation:
    Train classifiers on selected features:
      - LSTM
      - CNN
      - DNN
      - Auto-encoder
    Format feature input as required (1D or 2D)
    Train on training split
    Predict DR class labels on test split
    Save predictions
7. Evaluation:
  For each model:
    Compute Accuracy, Precision, Recall, Specificity, F1-Score
    Compute False Discovery Rate (FDR), False Omission Rate (FOR)
    Generate Confusion Matrix
  Average metrics across all folds
8. Statistical Validation:
  Perform statistical tests:
    - T-Test to compare model means
    - P-Test for significance
    - ANOVA for variance across models
End

IV. Experimental Results

The significance of the proposed method is tested and experimented by using three standard dataset and 5-fold

cross-validation. The data for each dataset is partitioned as binary partitions such as 80% for training and 20% for testing. The success of the proposed AMP-SFOA with the LSTM method is performed on the MATLAB 2021a platform, and the system specifications are Windows 10, 64-bit OS, 16GB RAM, and an Intel Core i7 processor. The proposed method is hyperparameter-sensitive and has been carefully optimized using a Bayesian optimization approach. The primary hyperparameters involves a dimension of the hidden layer that is fixed to 256 through an Adam optimizer and a learning rate, which decays from 0.001. A batch size of 32 and a maximum of 100 iterations are used, with early stopping applied using validation error. To address the potential overfitting issue, a dropout mechanism through the rate of 0.5 is integrated in fully connected layers. These hyperparameters were methodically tuned to balance the model's complexity, learning stability, and convergence speed.

The proposed approach is validated through different assessment indices of accuracy, precision, sensitivity/recall, F1-Score, specificity, FOR and FDR. Accuracy reflects overall correctness, while precision and recall measure the network's capacity for detecting positive emotion classes accurately and avoiding false detections. The F1-Score balances precision as well as recall, thereby offering more reliable measures in cases of class imbalance, which is common in the DR dataset. The mathematical representations of each metric are given in Eqs. (23)–(27).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

$$Specificity = \frac{TN}{TN + FP} \tag{26}$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN} \tag{27}$$

where, *TP* is True Positive, *TN* is True Negative, *FP* is False positive, and *FN* is False Negative.

A. Performance Analysis

This analysis performs a complete effectiveness validation of the proposed method in DR classification operations on two benchmarks. This paper performs an estimation mechanism utilized through various classifiers to comparatively analyze the proposed method against the state-of-the-art techniques examined in this research. The existing methods, such as Autoencoder (AE), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), and Deep Neural Network (DNN), are estimated and compared with the proposed LSTM approach.

Table II specifies the performance estimation of different feature selection methods using each dataset. The different feature selection methods, such as Grey Wolf Optimization (GWO), Whale Optimization Algorithm (WOA), Pelican Optimization Algorithm (POA), and SFOA, are estimated and compared using AMP-SFOA approach. The proposed AMP-SFOA outperforms others in terms of accuracy, precision, recall, and F1-Score, while indicating its superior capability in selecting relevant features for DR classification.

TABLE II. PERFORMANCE ESTIMATION OF FEATURE SELECTION METHODS

Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	FOR	FDR
APTOS-2019	GWO	97.468	95.877	95.044	95.425	95.358	0.627	6.9829
	WOA	97.044	94.343	94.833	94.548	99.271	0.733	5.6572
	POA	96.382	95.593	93.584	92.492	91.483	0.746	5.683
	SFOA	97.441	96.568	94.433	95.452	99.317	0.681	4.2733
	AMP-SFOA	98.764	98.975	98.644	98.356	98.975	0.526	4.1289
DDR	GWO	90.456	90.955	90.436	90.352	91.543	1.946	8.495
	WOA	93.564	93.563	93.564	93.574	94.574	1.043	6.476
	POA	94.573	93.593	92.403	94.202	94.208	1.138	7.303
	SFOA	95.345	95.644	94.645	95.754	95.948	0.956	4.564
	AMP-SFOA	95.846	96.460	96.235	96.347	96.873	1.486	3.824

Table III provides the baseline classification results using models such as AE, GRU, RNN, DNN, and LSTM without any feature selection. While LSTM shows the highest accuracy on both the dataset, the overall performance is comparatively lower when the feature selection is applied, while emphasizing the importance of dimensionality reduction. Table IV shows the improved performance of classification models when combined with AMP-SFOA for feature selection. LSTM with AMP-SFOA achieves the highest accuracy 99.476% and 96.958% for APTOS-2019 and DDR dataset, thereby proving the effectiveness of the selected features in enhancing the model outcomes. Although multiple classifiers such as AE, RNN, GRU, DNN, and LSTM are estimated for comparative analysis, the AMP-SFOA framework is primarily designed to optimize feature subsets independent of the classifier. All classifiers are trained utilizing the similar AMP-SFOA-selected features to ensure fairness. Between them, LSTM is emphasized as the final model because of its superior capability of modeling structured feature dependencies and it has reliably greater effectiveness across the datasets.

TABLE III. PERFORMANCE ESTIMATION OF CLASSIFICATION RESULTS WITHOUT FEATURE SELECTION

Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	FOR	FDR
APTOS-2019	AE	95.044	94.343	94.833	94.587	96.271	0.734	5.657
	GRU	96.468	95.877	95.044	95.425	95.358	0.627	6.983
	RNN	96.441	96.568	94.433	95.452	96.352	0.681	5.783
	DNN	97.846	96.547	96.577	96.580	96.567	0.626	4.273
	LSTM	98.764	98.975	98.644	98.809	98.975	0.526	4.129
DDR	AE	90.235	91.362	91.365	91.364	91.476	4.235	9.163
	GRU	91.894	92.235	92.251	92.243	92.128	3.152	7.365
	RNN	92.275	92.896	92.963	92.930	92.836	2.736	5.428
	DNN	93.286	93.388	93.487	93.437	93.984	2.237	4.284
	LSTM	95.846	96.460	96.235	96.347	96.873	1.486	3.824

TABLE IV. PERFORMANCE ESTIMATION OF CLASSIFICATION RESULTS WITH FEATURE SELECTION

Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	FOR	FDR
APTOS-2019	AE	96.367	95.474	95.786	95.630	97.436	0.527	5.012
	GRU	97.934	96.376	96.748	96.562	96.486	0.437	0.623
	RNN	98.376	97.578	96.547	97.060	97.757	0.325	0.735
	DNN	98.665	98.870	97.365	98.112	98.643	0.245	4.041
	LSTM	99.476	99.077	99.161	99.119	99.865	0.129	0.923
DDR	AE	93.826	93.764	93.863	93.813	93.973	3.645	7.047
	GRU	94.587	94.982	93.948	94.462	94.732	2.570	6.473
	RNN	94.836	95.387	94.876	95.131	95.487	1.984	5.387
	DNN	95.893	96.843	95.387	96.109	96.376	0.937	4.272
	LSTM	96.958	97.547	96.373	96.956	97.437	0.672	2.472

Table V evaluates the performance of the proposed model under various k-fold values (k = 3, 5, 7, 9) for both the dataset. The results demonstrate that k = 5 yields the best performance in terms of all key metrics, suggesting that it offers an optimal balance among bias and variance.

Table VI illustrates the cross-dataset analysis of the proposed method. To further validate generalization capability, cross-dataset evaluation was performed by training on APTOS-2019 and testing on DDR, and vice-versa. The proposed AMP-SFOA + LSTM model maintained stable performance with variations within ±2.3%. In addition, 95% confidence intervals were computed through 10-run repeated sampling. The narrow CIs confirm low performance variability, complementing the statistical significance outcomes reported using t-tests and ANOVA.

TABLE V. PERFORMANCE ESTIMATION OF CROSS-FOLD ANALYSIS OF PROPOSED METHOD

Dataset	K Values	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	FOR	FDR
APTOS-2019	K = 3	95.367	95.325	95.784	95.553	95.487	0.635	4.326
	K = 5	99.476	99.077	99.161	99.119	99.865	0.129	0.923
	K = 7	97.264	96.236	97.258	96.744	97.365	0.253	3.327
	K = 9	96.349	95.436	96.065	95.749	96.461	0.532	2.216
DDR	K = 3	93.365	93.363	93.846	93.603	93.152	5.287	6.328
	K = 5	96.958	97.547	96.373	96.956	97.437	0.672	2.472
	K = 7	95.372	96.253	95.437	95.386	96.360	1.326	3.362
	K = 9	94.263	94.436	94.361	94.398	95.939	2.462	2.848

TABLE VI. CROSS-DATA ANALYSIS OF PROPOSED METHOD WITH EXISTING METHODS

Train Dataset	Test Dataset	Accuracy (%)	Confidence Intervals	Precision (%)	Recall (%)	F1-Score (%)
APTOS-2019 →	DDR	90.12	89.42–90.82	89.03	87.87	98.45
DDR →	APTOS-2019	89.72	86.21–89.23	84.18	86.50	85.34

Table VII demonstrates the performance estimation of the computational cost of the proposed model through previous approaches. The computational metrics such as training time, memory usage, and inference time. While attaining great classification effectiveness, the proposed model requires less memory and provides rapid training and inference times compared to other large architectures.

Table VIII signifies an ablation study of the proposed method using a different dataset. A progressive enhancement from different learning rates confirms the balancing nature of chosen architectural constituents. Every addition result in measurable attains overall performance metrics, with the complete approach attaining superior performance.

Fig. 3 specifies the confusion matrix of the proposed LSTM approach based on various datasets. This figure provides insight into the per-class classification accuracy, while visualizing the distribution of true vs. predicted labels.

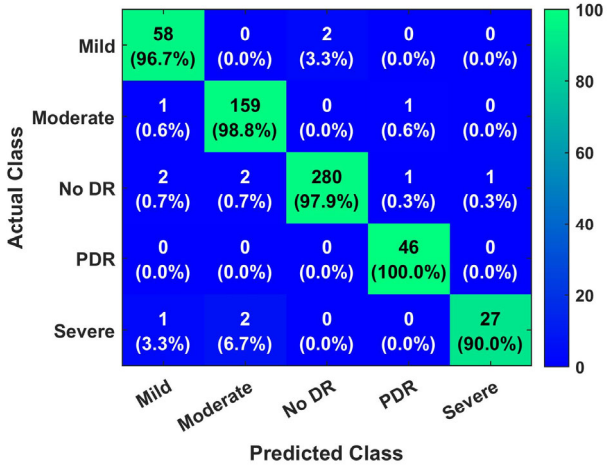
Fig. 4 specifies the ROC curve of the proposed LSTM approach using various datasets. The ROC curves illustrate the approach's ability to distinguish between disease and non-disease cases at different thresholds.

TABLE VII. COMPUTATIONAL COMPLEXITY OF PROPOSED METHOD WITH EXISTING METHODS

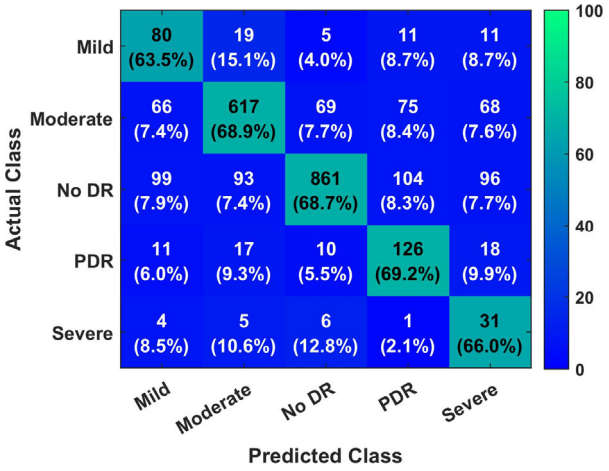
Method	Dataset	Training Time (s)	Memory Usage (MB)	Inference Time (s)
AE	APTOS-2019	1025.4503	390.2507	17.4205
	DDR	1985.3712	575.1216	16.1809
GRU	APTOS-2019	987.6412	372.8845	15.8912
	DDR	1860.2248	552.3107	14.8719
RNN	APTOS-2019	1153.2321	415.771	18.6348
	DDR	2050.5523	600.9803	17.5432
DNN	APTOS-2019	832.1123	335.9456	14.2199
	DDR	1655.8827	515.0491	13.5644
LSTM	APTOS-2019	748.1887	318.1131	13.0323
	DDR	1493.6205	492.5921	11.3646

TABLE VIII. ABLATION STUDY OF THE PROPOSED METHOD WITH DIFFERENT LEARNING RATE

Dataset	Learning Rate	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	FOR	FDR
APTOS-2019	0.400	96.433	96.487	96.267	96.346	96.709	4.375	0.527
	0.450	97.439	97.543	97.476	97.509	97.486	3.476	0.438
	0.500	98.145	97.007	98.371	97.684	97.909	6.749	0.315
	0.550	99.476	99.077	99.161	99.119	99.865	0.129	0.923
DDR	0.400	94.326	94.617	94.836	94.726	93.241	4.126	6.439
	0.450	94.836	94.867	94.972	94.919	94.271	3.438	4.329
	0.500	95.272	96.433	95.326	95.363	96.436	1.273	3.486
	0.550	96.958	97.547	96.373	96.956	97.437	0.672	2.472

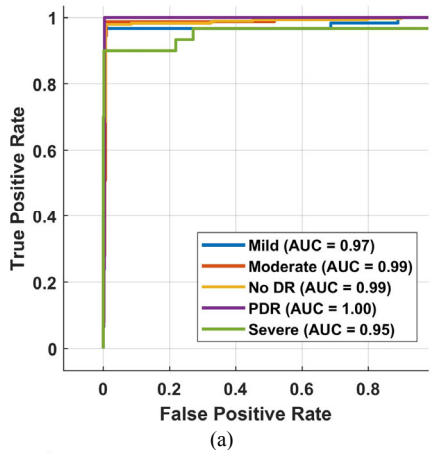


(a)

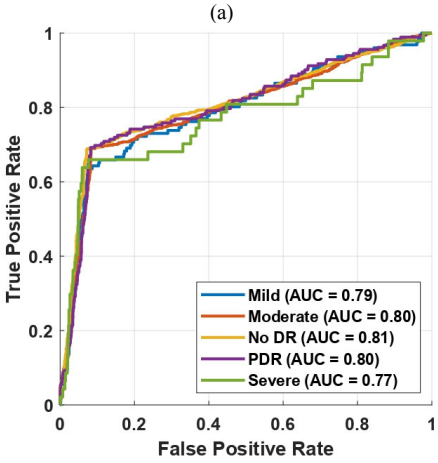


(b)

Fig. 3. Confusion matrix of the proposed method: (a) APTOS-2019 dataset, (b) DDR dataset.



(a)

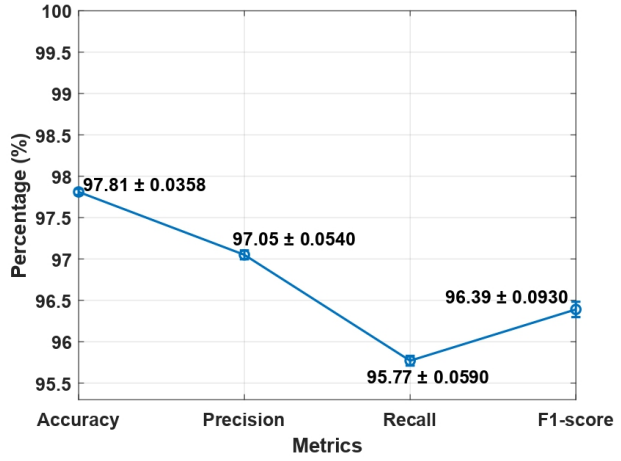


(b)

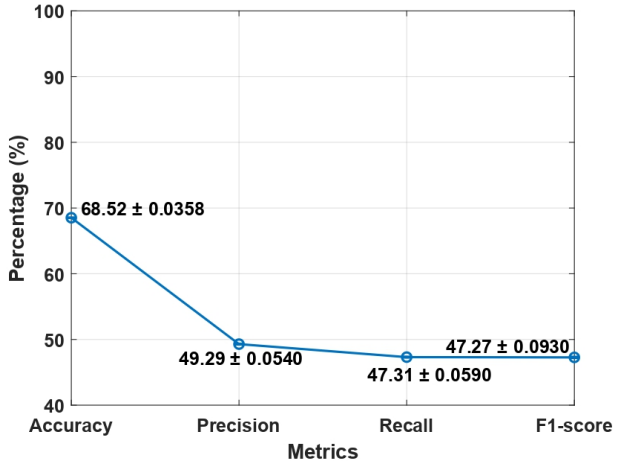
Fig. 4. ROC curve of the proposed method: (a) APTOS-2019 dataset, (b) DDR dataset.

Fig. 5 demonstrates the graphical representation of the standard deviation of the proposed AMP-SFOA with the LSTM approach. The low standard deviation values across all metrics while confirming that the proposed AMP-SFOA with the LSTM method delivers consistent and

reliable performance. This indicates the model's robustness in producing stable results regardless of the dataset variation. The minimal fluctuations across dataset further validate the method's resilience against class imbalance and noise. Hence, the figure supports the statistical consistency of the proposed classification approach.



(a)



(b)

Fig. 5. Standard deviation curve of the proposed method: (a) APTOS-2019 dataset, (b) DDR dataset.

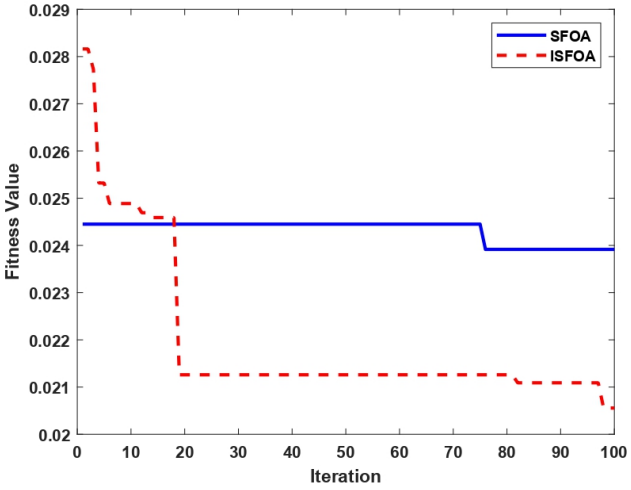


Fig. 6. Fitness vs iteration graph for AMP-SFOA approach.

Fig. 6 demonstrates the fitness vs iteration graph for the AMP-SFOA approach. The graph demonstrates that the ISFOA (red dashed line) achieves faster and more effective convergence than the standard SFOA (blue solid line). ISFOA quickly reduces the fitness value within the first 20 iterations and continues to improve, thereby reaching a lower final value. This indicates better exploration and optimization capability, while making ISFOA more efficient and accurate in finding optimal solutions.

B. Statistical Analysis

To evaluate the statistical significance of the proposed LSTM-based classifier for diabetic retinopathy detection, independent T-tests were conducted against a baseline DNN across two benchmark datasets: DDR and APTOS-2019. On the DDR dataset, the LSTM model achieved a mean accuracy of 95.54%, outperforming the DNN, which achieved 93.39%. The T-test resulted in a p-value of 0.0001 and a t-statistic of 8.5813, indicating a highly significant difference ($p < 0.05$). Similarly, on the APTOS-2019 dataset, the LSTM attained a mean accuracy of 99.29%, while the DNN achieved 98.39%. The corresponding p-value was 0.0386 with a t-statistic of 2.6380, again confirming the statistical significance. These results validate that the proposed LSTM classifier offers a consistent and statistically significant improvement over the baseline, thereby ensuring robust performance across diverse diabetic retinopathy datasets.

C. Comparative Analysis

Table IX specifies the comparative analysis of the proposed method with existing methods based on various dataset. Through analyzing data in the subsequent table, it is evident that the proposed method attains greater results for DR classification across all dataset. For better comparison, the decimal values of existing methods like RT2Net [21] and STMF-DRNet [26] have been converted into percentages to be arranged with the proposed method's reported values.

D. Discussion

Training-validation accuracy curves, class-specific scores, and confusion matrices were analyzed, ensuring consistent learning without overfitting. A cross-dataset experiment (training on APTOS2019, testing on DDR, and vice versa) showed accuracy variation within ±2.3%, indicating strong generalization. Additionally, 95% confidence intervals are computed over 10 independent runs and showed narrow bounds (±0.41% for APTOS; ±0.58% for DDR), demonstrating stable model performance. Although the proposed model attains greater accuracy, potential risks of overfitting and dataset bias are carefully considered. To mitigate these risks, the cross-fold cross-validation, cross-dataset evaluation, and repeated-run confidence interval analysis are performed. The consistent effectiveness of APTOS-2019 and DDR datasets, along with narrow confidence intervals, illustrates better generalization capability and minimizes the probability of dataset-specific bias.

TABLE IX. COMPARATIVE ANALYSIS OF PROPOSED METHOD

Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-Score (%)
APTOS-2019	RT2Net [21]	85.4	82.4	78.2	80.3	NA
	WAE [22]	95.06	87.88	NA	NA	85.69
	ViG+EfficientNet-B5(ensemble) [23]	84.48	84.07	84.55	NA	84.31
	Transfer learning [24]	95.92	95.92	96.02	NA	96.02
	DiabSense [25]	94.64	NA	86.61	96.98	NA
	STMF-DRNet [26]	85.5	NA	81.0	90.4	85.9
	Proposed AMP-SFOA	99.476	99.077	99.161	99.119	99.865
DDR	Transfer learning [24]	97.39	97.40	97.40	NA	97.40
	STMF-DRNet [26]	87.9	[1] NA	81.6	84.9	89.1
	Proposed AMP-SFOA	96.958	97.547	96.373	96.956	97.437

[1] NA represents Not Available.

E. Limitation and Future Scope

In spite of attaining better performance, the proposed AMP-SFOA framework involves some limitations. The feature selection process is integrated through deep feature extraction and LSTM classification, which introduces computational overhead during training. Although appropriate for offline analysis and large-scale screening, real-time deployment in clinical environments needs further optimization. Future work will concentrate on minimizing computational complexity by using lightweight backbone networks, parallel processing strategies, and model compression techniques to improve scalability and real-time applicability.

V. CONCLUSION

Visual impairment is a serious global concern as it severely affects the person's eyesight and quality of life. Diabetic retinopathy is receiving a lot of attention from researchers because it causes irreversible vision loss. The lack of early warning signs or minor visual issues that eventually lead to blindness makes it a difficult disease. AMP-SFOA is proposed for automatic diagnosis and classification of DR into multiple classes. Primarily, this research considers the two-standard dataset, which are ATPOS-2019 and DDR to implement the proposed method based on training and testing. Then, image resizing, CLAHE and data augmentation are performed to improve the input data. Finally, LSTM is used for classification of DR into multi-classes. The experimental results demonstrates that the proposed AMP-SFOA approach reaches better accuracies of 99.476% and 96.958% on APTOS-2019 and DDR dataset, respectively as compared to existing methods such as RT2Net and STMF-DRNet. From a clinical perspective, the proposed AMP-SFOA-assisted DR classification model provides rapid, consistent automated screening, which supports ophthalmologists in early detection, especially in the regions, where there is a limited availability of specialists. By enhancing the identification of mild and moderate DR, the system reduces the delay of diagnosis and prevent the progression of vision-threatening stages. Hence, the proposed approach has a significant potential to be integrated into real-world tele-ophthalmology and mass-screening workflows. Future work will investigate by stimulating the network's flexibility in large real-time conditions, thereby integrating mechanisms to attain overall performance.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

B. R. Raghu conceptualized and designed the research study, developed the methodology, and prepared the initial manuscript draft. Janapati Venkata Krishna contributed to data analysis, validation of results, and performed critical revisions to the manuscript. N. Pradeep supported the investigation, resource management, and technical guidance throughout the research work. All authors have read and approved the final version of the manuscript for publication.

ACKNOWLEDGMENT

The authors sincerely acknowledge the support and facilities provided by Srinivasa University, Institute of Engineering and Technology, and Bapuji Institute of Engineering and Technology for the successful completion of this research work.

REFERENCES

[1] N. Sharma and P. Lalwani, "A multi model deep net with an explainable AI based framework for diabetic retinopathy segmentation and classification," *Scientific Reports*, vol. 15, no. 1, 8777, 2025.

[2] M. S. B. Phridviraj, R. Bhukya, S. Madugula, A. Manjula, S. Vodithala, and M. S. Waseem, "A bi-directional long short-term memory-based diabetic retinopathy detection model using retinal fundus images," *Healthcare Anal.*, vol. 3, 100174, 2023.

[3] M. Akram, M. Adnan, S. F. Ali *et al.*, "Uncertainty-aware diabetic retinopathy detection using deep learning enhanced by Bayesian approaches," *Sci. Rep.*, vol. 15, no. 1, 1342, 2025.

[4] P. Zhang, J. Zhao, Q. Liu, X. Liu, X. Li, Y. Gao, and W. Li, "Fundus image generation and classification of diabetic retinopathy based on convolutional neural network," *Electronics*, vol. 13, no. 18, 3603, 2024.

[5] M. A. K. Raiaan, K. Fatema, I. U. Khan *et al*., "A lightweight robust deep learning model gained high accuracy in classifying a wide range of diabetic retinopathy images," *IEEE Access*, vol. 11, pp. 42361–42388, 2023.

[6] M. Tariq, V. Palade, and Y. Ma, "Effective diabetic retinopathy classification with siamese neural network: A strategy for small dataset challenges," *IEEE Access*, vol. 12, pp. 182814–182827, 2024.

[7] K. V. Naveen, B. N. Anoop, K. S. Siju, M. K. Kar, and V. Venugopal, "EffNet-SVM: A hybrid model for diabetic retinopathy

classification using retinal fundus images," *IEEE Access*, vol. 13, pp. 79793–79804, 2025.
[8] C. Xu, X. Guo, G. Yang, Y. Cui, L. Su, H. Dong, X. Hu, and S. Che, "Prior-guided attention fusion transformer for multi-lesion segmentation of diabetic retinopathy," *Sci. Rep.*, vol. 14, no. 1, 20892, 2024.
[9] J. Zhang, R. Jin, and W. Liu, "Enhancing semi-supervised contrastive learning through saliency map for diabetic retinopathy grading," *IET Comput. Vision*, vol. 18, no. 8, pp. 1127–1137, 2024.
[10] P. Macsik, J. Pavlovicova, S. Kajan, J. Goga, and V. Kurilova, "Image preprocessing-based ensemble deep learning classification of diabetic retinopathy," *IET Image Proc.,* vol. 18, pp. 807–828, 2023.
[11] S. Guefrachi, A. Echtioui, and H. Hamam, "Diabetic retinopathy detection using deep learning multistage training method," *Arabian Journal for Science and Engineering*, vol. 50, no. 2, pp.1079–1096, 2024.
[12] Z. Gu, Y. Li, Z. Wang, J. Kan, J. Shu, and Q. Wang, "Classification of diabetic retinopathy severity in fundus images using the vision transformer and residual attention," *Comput. Intell. Neurosci*., vol. 2023, no. 1, 1305583, 2023.
[13] Y. Fu, M. Liu, G. Zhang, and J. Peng, "Lightweight frequency recalibration network for diabetic retinopathy multi-lesion segmentation," *Applied Sciences*, vol. 14, no. 16, 6941, 2024.
[14] R. Ramesh and S. Sathiamoorthy, "Blood vessel segmentation and classification for diabetic retinopathy grading using dandelion optimization algorithm with deep learning model," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 5, pp. 11–20, 2023.
[15] S. Tang and Q. Wu, "Research on recognition of diabetic retinopathy hemorrhage lesions based on fine tuning of segment anything model," *Scientific Reports*, vol. 15, no. 1, 10292, 2025.
[16] M. H. Ashraf and H. Alghamdi, "HFF-Net: A hybrid convolutional neural network for diabetic retinopathy screening and grading," *Biomed. Technol*., vol. 8, pp. 50–64, 2024.
[17] M. Sushith, A. Lakkshmanan, M. Saravanan, and S. Castro, "Attention dual transformer with adaptive temporal convolutional for diabetic retinopathy detection," *Sci. Rep.*, vol. 15, no. 1, 7694, 2025.
[18] I. A. Taifa, D. M. Setu, T. Islam, S. K. Dey, and T. Rahman, "A hybrid approach with customized machine learning classifiers and multiple feature extractors for enhancing diabetic retinopathy detection," *Healthcare Anal*., vol. 5, 100346, 2024.
[19] J. M. N. D. Rio, P. Nderitu, R. Raman *et al*., "Using deep learning to detect diabetic retinopathy on handheld non-mydriatic retinal images acquired by field workers in community settings," *Sci. Rep.*, vol. 13, no. 1, 1392, 2023.
[20] S. Rajeshwar *et al.*, "Diabetic retinopathy detection using DL-based feature extraction and a hybrid attention-based stacking ensemble," *Advances in Public Health*, vol. 2025, no. 1, 8863096, 2025.
[21] X. Xu, D. Liu, G. Huang, M. Wang, M. Lei, and Y. Jia, "Computer aided diagnosis of diabetic retinopathy based on multi-view joint learning," *Comput. Biol. Med*., vol. 174, 108428, 2024.
[22] K. Nazir, J. Kim, and Y. C. Byun, "Enhancing early-stage diabetic retinopathy detection using a weighted ensemble of deep neural networks," *IEEE Access*, vol. 12, pp. 113565–113579, 2024.
[23] M. N. U. Alam, I. Hasnine, E. H. Bahadur *et al*., "DiabSense: Early diagnosis of non-insulin-dependent diabetes mellitus using smartphone-based human activity recognition and diabetic retinopathy analysis with graph neural network," *Journal of Big Data*, vol. 11, no. 1, 103, 2024.
[24] M. Butt, D. N. F. Awang-Iskandar, M. A. Khan, G. Latif, and A. Bashar, "MEDCnet: A memory efficient approach for processing high-resolution fundus images for diabetic retinopathy classification using CNN," *Int. J. Imaging Syst. Technol*, vol. 35, no. 2, e70063, 2025.
[25] M. Herrero-Tudela, R. Romero-Oraá, R. Hornero, G. C. G. Tobal, M. I. López, and M. García, "An explainable deep-learning model reveals clinical clues in diabetic retinopathy through SHAP," *Biomed. Signal Process. Control*, vol. 102, 107328, 2025.
[26] Y. Liu, D. Yao, Y. Ma, H. Wang, J. Wang, X. Bai, G. Zeng, and Y. Liu, "STMF-DRNet: A multi-branch fine-grained classification model for diabetic retinopathy using Swin-TransformerV2," *Biomed. Signal Process. Control*, vol. 103, 107352, 2025.