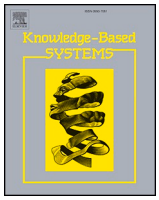
Adaptive feature selection for active trachoma image classification

Knowledge-Based Systems 294 (2024) 111764

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Adaptive feature selection for active trachoma image classification

Mulugeta Shitie Zewudie [a], Shengwu Xiong [a,b,c,d], Xiaohan Yu [e,*], Xiaoyu Wu [f], Moges Ahmed Mehamed [a]

[a] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, 430070, China
[b] Sanya Science and Education Innovation Park, Wuhan University of Technology, Sanya, 572000, China
[c] Shanghai Artificial Intelligence Laboratory, Shanghai, 200232, China
[d] School of Information Science and Technology, Qiongtai Normal University, Haikou, 571127, China
[e] School of Computing, Macquarie University, NSW, Australia
[f] Institute for Integrated and Intelligent Systems, Griffith University, QLD, Australia

ARTICLE INFO

Keywords:
Active trachoma
Attention mechanism
Deep learning
Feature selection

ABSTRACT

Trachoma is a neglected tropical eye disease caused by ocular strains of Chlamydia trachomatis, which affects millions of people worldwide. To examine the eye for signs of active trachoma, healthcare providers typically look for clusters of five or more follicles on the conjunctiva of the upper eyelid for the follicular inflammatory trachoma stage. However, it is also possible to find individual follicles scattered throughout the conjunctiva, particularly in mild or early-stage trachoma cases. Additionally, the datasets are photographic images collected in the field that can be high-dimensional and may contain large amounts of redundant information. We propose integrating novel attention-based feature extraction and feature selection techniques to address these challenges. First, we present the Lambda layer within the Convolutional Block Attention Module (L-CBAM) to normalize attention weights and improve the feature extraction process. Second, we introduce an adaptive mechanism, Adaptive Beta Hill Climbing (AβHC) with Social Ski-Driver (SSD), which adjusts the exploration-exploitation trade-off during the search process, allowing for better exploration of the search space and more efficient convergence toward an optimal feature subset. We then use the multilayer perceptron (MLP) classifier to produce final classification results using selected subsets. We evaluated the proposed approach on active trachoma inverted eyelid images and obtained accuracy scores of 93.3% with only 19.7% of the selected features, surpassing many of the algorithms used for comparison. Our proposed method has demonstrated excellent performance compared to recent works utilizing the same datasets. The source code of this work is available at https://github.com/mshitie2/Active_Trachoma.

1. Introduction

Trachoma is a neglected tropical eye disease caused by ocular strains of Chlamydia trachomatis and affects millions of people worldwide [1,2]. It is transmitted through infected ocular and nasal secretions, often through contact with fingers, clothing, or germ carriers, and can also be spread by eye-seeking flies [1]. Trachoma is acknowledged by the World Health Organization (WHO) as a significant public health issue in 44 countries, affecting a population of 137 million individuals residing in endemic areas [2]. The WHO classifies active trachoma into two grades: follicular inflammatory trachoma (TF) and intense inflammatory trachoma (TI) [2]. TF manifests as five or more small bumps or follicles on the conjunctiva of the upper eyelid [3]. Inflammation caused by bacterial infection gives rise to these bumps, typically observed in the early stages of the disease.

Conversely, TI refers to the thickening of the tarsal conjunctival tissue, which lines the inner surface of the eyelid [3]. This thickening occurs due to scarring and fibrosis developed from repeated infections and inflammation. When examining the eye for signs of trachoma, healthcare providers typically look for clusters of five or more follicles on the conjunctiva of the upper eyelid. However, it is also possible to find individual follicles scattered throughout the conjunctiva, particularly in mild or early-stage trachoma cases. Thickening of the tarsal conjunctiva tissue is typically diffuse and affects the entire inner surface of the eyelid. Eye diseases can manifest in various forms and textures that can be difficult for optometrists to detect, recognize, and analyze

* Corresponding author.
E-mail address: xiaohan.yu@griffith.edu.au (X. Yu).

https://doi.org/10.1016/j.knosys.2024.111764
Received 7 December 2023; Received in revised form 19 March 2024; Accepted 3 April 2024
Available online 4 April 2024

[4]. Manual grading of retinal images is a time-consuming and expensive process that requires highly skilled staff and is often prone to inaccuracies due to grader fatigue [5]. Studies have shown inconsistent grading among different graders and even for the same grader [1]. Therefore, leveraging deep learning technologies to improve the current system and offer maximum comfort to patients and ophthalmologists is essential.

The use of deep learning (DL) in developing models for automated diagnosis of eye diseases, such as age-related macular degeneration [4,6], diabetic retinopathy [7–9], and glaucoma [10,11], is a popular area of research. Fundus photography or optical coherence tomography images serve as input sources. However, there has been limited research on active trachoma classification. Study [1] utilized a convolutional neural network, and Study [2] proposed automated machine learning (AutoML) for classifying the two stages of active trachoma, while another study [12] focused solely on a single stage. Researchers have demonstrated that deep learning applications can provide reliable predictions for detecting and classifying active trachoma [1,12]. Despite the effectiveness of deep learning in addressing research challenges, incorporating machine learning techniques such as attention mechanisms and feature selection into deep learning models has emerged as a prevalent strategy to improve model performance further [13].

Attention mechanisms help to identify crucial regions in an image by giving those areas more attention [14,15]. This capability is valuable in mild or early-stage trachoma cases, where individual follicles are scattered across the conjunctiva. The Convolutional Block Attention Module (CBAM) is a widely used attention mechanism that enhances conventional convolutional blocks (CB) by incorporating attention mechanisms [16,17]. CBAM attention modules have been extensively integrated into Convolutional Neural Network (CNN) models to improve their feature extraction capabilities [14,18]. However, CBAM does suffer from specific issues, such as outliers or extreme values and the loss of relevant regions. These limitations can impact the performance of the attention mechanism.

Like other medical datasets, photographic images collected in the field can be considered high dimensional, negatively affecting accuracy, cost, and speed [19]. Feature selection techniques are employed in machine learning models to enhance performance by removing irrelevant and redundant features [20,21], addressing such problems. This process selects the most informative features to improve accuracy while reducing processing time and storage space [19,20]. Indeed, many features within input datasets are redundant or irrelevant, leading to increased training times and posing challenges for classification methods [22,23]. Feature selection is a crucial preprocessing technique to address these issues by identifying a minimal subset of pertinent features from the original expansive feature set. This reduction in features accelerates training and enhances learning performance [24]. Therefore, the objective of feature selection extends beyond merely improving classification accuracy [25]. In FS methods, evaluation techniques delineate three principal categories: filter, wrapper, and embedded approaches [22,26]. Filter approaches assess features independently of any specific classification methodology, potentially constraining overall performance. Despite their computational efficiency relative to wrapper approaches, filter methods typically exhibit inferior efficacy.

Conversely, wrapper approaches integrate a classification method within the evaluation function, rendering them more time-intensive. Nonetheless, they often yield superior outcomes compared to filter approaches. Embedded approaches seamlessly incorporate the FS process into the classifier training procedure, effectuating simultaneous optimization. Within embedded methods, convergence is frequently facilitated by incorporating appropriate regularization techniques to minimize feature weights.

Meta-heuristic wrapper-based feature selection algorithms are widely used to solve various optimization problems due to their nonderivative nature, flexibility, and ability to avoid local optima [27]. These algorithms have two main components: exploration and exploitation [27]. Exploration allows the algorithm to explore different regions of the search space and achieve a global search for a given problem. At the same time, exploitation ensures the identification of optimal individuals within a particular region, facilitating local search. Striking a balance between these two components is crucial in assessing the effectiveness of a meta-heuristic algorithm, but it can be challenging due to the stochastic properties involved.

Implementing an ensemble of local searches can enhance the ability to exploit feature selection techniques, thereby improving the overall performance of the learning model. As demonstrated in studies on prostate cancer detection [28] and COVID-19 detection [29], meta-heuristic algorithms have received significant scientific interest in medical image classification and analysis. Furthermore, researchers have explored local search to enhance the exploitability of optimization algorithms, which frequently converge prematurely to a local optimum [20,29].

Based on the abovementioned problem, we propose a two-stage classification model that integrates attention-based feature extraction and adaptive feature selection techniques. Existing CBAM feature extraction and SSD feature selection methods typically perform poorly in extracting informative features, leading to poor model generalization. CBAM suffers from issues such as outliers or extreme values and the loss of relevant regions. We propose the Lambda layer within the CBAM module to address these limitations. This layer serves to normalize the attention weights within an attention model.

On the other hand, SSD often faces limitations like inadequate exploration of local search spaces, resulting in suboptimal feature subsets and reduced performance. To overcome this limitation, we propose an adaptive mechanism called Adaptive Beta Hill Climbing (AβHC) within SSD. This mechanism adjusts the exploration-exploitation tradeoff during the search process, allowing for better search space exploration and more efficient convergence towards an optimal feature subset. The selected feature subset is input to an MLP classifier for final classification.

The following are the primary contributions of this research:

1. We propose a novel approach that combines attention-based feature extraction with adaptive feature selection, effectively closing the gap between feature extraction and selection in active trachoma image classification.
2. We have introduced a Lambda layer within the CBAM module, called L-CBAM, to address the problem of losing relevant regions in CBAM. L-CBAM enables the normalization of attention weights within the attention model.
3. We propose an adaptive mechanism called Adaptive Beta Hill Climbing (AβHC) within SSD for feature selection. This mechanism addresses the inadequate exploration of local search spaces and facilitates efficient convergence toward an optimal feature subset.
4. Our proposed model exhibits superior (or comparable) performance to existing approaches, achieved by eliminating 80.3% of irrelevant and redundant features.

The paper follows the following format: Section 2 presents the related work, while Section 3 describes the preliminaries. Section 4 provides a detailed explanation of the proposed method. The dataset description and parameter setting are presented in Section 5, followed by the study findings and discussion in Section 6. Finally, Section 7 concludes the paper.

2. Related works

DL has been proven successful in various medical image classification tasks, surpassing clinical classification performance. DL-based image detection and classification systems have also been implemented in trachoma, explicitly focusing on active trachoma image

classification. For instance, Kim et al. [1] proposed CNN to detect clinical signs of trachoma. They focused on two stages of active trachoma, TF, and TI, achieving 70% accuracy for TF cases and 85% for TI cases. Milad et al. [2] studied the performance of automated machine learning (AutoML) for diagnosing active trachoma by analyzing conjunctiva images. The findings revealed that the TF model achieved an accuracy rate of 88%, while the TI model achieved an accuracy rate of 93%. Yenegeta and Assabie [30] used a texture feature-based CNN approach for trachoma detection and grading, achieving an accuracy rate of 97.9%. However, it is essential to note that their method only categorized three stages of trachoma (TS, TT, and CO) and did not encompass active trachoma. Socia et al. [12] proposed trachoma classifiers using ResNet101 and VGG16 CNN models and employed oversampling techniques on positive images to balance the data. This study focused solely on one of the five stages of trachoma, namely TF, achieving an 88% accuracy rate using the ResNet101 model.

After examining the studies mentioned above, it becomes clear that researchers widely utilize CNNs for feature extraction in active trachoma image classification. Transfer learning is also valuable for addressing data scarcity, particularly in emerging domains [31–33]. Nonetheless, CNNs have specific limitations, such as potential redundancy in the extracted features. To address these challenges, researchers have proposed numerous approaches: i) Attention mechanisms: This architectural element aims to enhance representation capabilities by prioritizing important features while suppressing irrelevant ones [34]. Recently, researchers have been increasingly interested in investigating attention mechanisms and incorporating them into deep learning frameworks. For instance, in [18], a novel attention-based feature learner was devised, integrating a lightweight CBAM into the ResNet12 architecture (CBAM-ResNet12). This integration enabled optimal learning of informative features in an end-to-end manner. Similarly, in [14], a computer-generated image detection algorithm was proposed, leveraging transfer learning and the CBAM. The main difference between the proposed method and the existing methods in the literature is that existing CBAM suffers from specific issues, such as outliers, extreme values, and the loss of relevant regions. These limitations can impact the performance of the attention mechanism. The proposed method addresses these limitations by utilizing a Lambda layer in the CBAM to normalize the attention weights within an attention model. ii) Meta-heuristic feature selection algorithms: Various studies have employed meta-heuristic feature selection algorithms to improve the performance of DL models. Basu et al. [29] proposed a two-stage framework for detecting COVID-19 from CT scan images, comprising feature extraction and selection stages. They employed CNN architectures such as DenseNet, ResNet, and Xception during the feature extraction stage to generate a feature vector from the input images. For feature selection, they used the Harmony Search (HS) [35] algorithm and AβHC. These algorithms effectively select the most relevant features and enhance the performance of the feature extractor CNNs. Additionally, incorporating a local search using AβHC generally improves classification accuracy. The proposed method achieves impressive accuracy rates of 97.30% and 98.87% on two datasets. Xue et al. [13] proposed the External Attention-Based Feature Ranker for Large-Scale Feature Selection (EAR-FS). This methodology effectively combines neural network architectures with feature selection methodologies to pinpoint essential features within datasets of high dimensionality. This method assigns scores to individual features by utilizing attention units, preserving those with high scores while discarding those with low ones. Following the ranking process, EAR-FS employs a hybrid heuristic search strategy, which amalgamates global and local searches, to pinpoint a compact feature subset while preserving high levels of accuracy. In another study by Chatterjee et al. [20], they proposed a new meta-heuristic algorithm for feature selection based on the SSD optimization technique. The authors apply S-shaped and V-shaped transfer functions to convert the continuous search space of SSD to binary and utilize the LAHC (Late Acceptance Hill Climbing) algorithm to enhance the exploitation ability of SSD. The study demonstrates that incorporating LAHC with SSD significantly improves the results in terms of classification accuracy and the number of selected features. Existing SSDs often suffer from limitations such as inadequate exploration of local search spaces, leading to suboptimal feature subsets and reduced performance. The proposed method uses an adaptive mechanism called AβHC that adjusts the exploration-exploitation trade-off during the search process, allowing for better search space exploration and more efficient convergence towards an optimal feature subset. As far as we know, attention-based feature extraction and the SSD optimization technique have not been applied in active trachoma detection and classification.

3. Preliminaries

This section covers the fundamental concepts of attention mechanisms to understand how to efficiently extract features and the SSD and AβHC algorithms for selecting relevant features.

3.1. Attention mechanism

Deep learning has succeeded in prediction and classification tasks, with CNNs particularly suitable for analyzing 2D or 3D images [17]. CNNs have a solid ability to represent data, enhancing the performance of visual tasks. Researchers have explored depth, width, and cardinality to improve CNN quality and capacity. Attention mechanisms have also gained significant attention, enabling CNNs to focus on essential features and improving performance and efficiency [34]. The attention mechanism operates as an adaptive weighting operation on the input, allowing the model to focus on crucial information. Depending on the dimensions involved, attention mechanisms can be categorized into channel, spatial, or mixed attention [18].

In recent years, there has been a growing interest among researchers in investigating attention mechanisms and integrating them into deep learning frameworks. For example, Hu et al. [36] introduced the squeeze-and-excitation (SE) network, which incorporates a channel attention module that assigns varying weights to individual channels within feature maps. Similarly, Wang et al. [37] proposed a parameter-free spatial attention (SA) layer that evaluates pixel values within each feature map to determine their relative importance. This SA layer belongs to the spatial attention module. In 2018, Woo et al. [16] introduced a novel convolutional block attention module (CBAM) that enhances the traditional convolutional block (CB) by integrating an attention mechanism. CBAM combines channel and spatial attention, allowing the model to calculate weights for each channel and pixel. This hybrid attention approach overcomes the limitations of individual attention methods by guiding the model on both “what” to learn and “where” to focus [18].

Additionally, studies attempt to incorporate these attention mechanisms into Neural Architecture Search (NAS). For instance, [38] introduced progressive partial channel connections based on channel attention for differentiable architecture search (PA-DARTS), and [39] proposed self-adaptive weights based on dual attention for differentiable neural architecture search (SWD-NAS), which is a channel attention-based approach. These two approaches aim to tackle performance collapse in DARTS because non-parametric operations tend to accumulate more advantages than parametric operations during the incomplete training period.

3.2. Social ski-driver optimization algorithm

The SSD optimization algorithm is a population-based metaheuristic optimization algorithm inspired by the behavior of a group of skiers who ski downhill together [40,41]. The SSD algorithm mimics the social behavior of skiers following each other down a slope to solve complex optimization problems. Below is a brief overview of these parameters

Location of the agents: The location of the agents is where we calculate the fitness function ($L_K^{R^n}$), with n representing the dimension of the search space.

Best personal location: The fitness function computes all agents' fitness values and compares them with their current location. Based on this comparison, the agent's best location (PBk) is recorded.

Best mean global location (MGB): The global point is the mean of the top three best solutions, calculated per Eq. (1), and the agents move towards this point.

$$MGB = \frac{L_X + L_y + L_z}{3} \tag{1}$$

The top three solutions are L_x, L_y, and L_z, respectively.

Velocity and location updating: Eq. (2) and (3) are used to adjust the position and velocity of the agents, respectively.

$$L_K^{T+1} = V_K^T + L_K^T \tag{2}$$

$$V_K^{T+1} = \begin{cases} h * \sin(\text{rand}(0,\ 1))\left(\text{PB}_K^T - L_K^T\right) \\ +\sin(\text{rand}(0,\ 1))\left(\text{MGB}_K^T - L_K^T\right) \quad if\ rand(0,\ 1) \le 0.5 \\ h * \cos(\text{rand}(0,\ 1))\left(\text{PB}_K^T - L_K^T\right) \\ +\cos(\text{rand}(0,\ 1))\left(\text{MGB}_K^T - L_K^T\right) \qquad else \end{cases} \tag{3}$$

Eqs. (2) and (3) represent the velocity V_K^T, mean global best position MGB_K^T, and current position L_K^T of the particle at the kth dimension and Tth iteration. PBk denotes the personal best position of the particle at the kth dimension. The sine and cosine functions are conventionally denoted as $Sin(x)$ and $Cos(x)$, respectively. The $Rand(0,\ 1)$ function randomly selects a number between 0 and 1. The variable 'h' balances the two crucial aspects of exploitation and exploration and is calculated according to Eq. (4).

$$h^{T+1} = r \times h^T \tag{4}$$

The value of 'T' in Eq. (4) represents the current iteration, and 'r' is used to decrease the value of h. The sine and cosine functions derive V_K^{T+1}; in Eq. (3), it ensures that the agent movement direction is not entirely straightforward due to these functions, allowing the algorithm to explore and diversify the search space systematically. Moreover, SSD is more social than other meta-heuristics. The agents in SSD strive to move towards the mean of the best three options, allowing the algorithm to escape from local minima if the global best solution is present there [28]. Compared to the Particle Swarm Optimization (PSO) [42] algorithm, SSD is also faster in discovering optimal solutions.

Adaptive Beta Hill Climbing (AβHC): Hill climbing may sometimes encounter difficulties in local optima. To address this issue, researchers proposed Beta Hill Climbing (βHC) [27,28]. However, in βHC, parameter tuning can be a significant challenge, and it requires extensive experimentation for each problem under consideration. To avoid these exhaustive experiments for setting optimal parameter values, an adaptive version called AβHC was proposed [27]. AβHC is a local search algorithm used for solving optimization problems. The algorithm works by iteratively modifying the current solution by slightly changing one or more variables [43]. If the modified solution is better than the current one, it is accepted and becomes the new one. Otherwise, the algorithm continues exploring the neighborhood until it finds a better solution or meets a stopping criterion. The "adaptive" part of AβHC refers to its ability to adjust the search parameters during the optimization process. AβHC uses a "hill-climbing" approach, which means that it always tries to move toward the direction of the best solution found so far. Integrating AβHC into the SSD algorithm enhances its ability to find high-quality solutions efficiently, making it a popular choice for solving complex optimization problems. AβHC iteratively uses N and β operators to generate a better solution $L^{'}(L^{'}_1,\ L^{'}_2, L^{'}_3, \ldots L^{'}_K)$ with the help of a neighbor solution $L(\ L_1,\ L_2,\ L3, \ldots L_K)$ as follows:

$$L_j^{'} = L_j \pm rand(0,\ 1) \times N \quad where\ j = 1,\ 2,\ \ldots k \tag{5}$$

Eq. (5) defines N as the maximum anticipated distance between the current solution and its neighboring solutions. The function $rand(0,\ 1)$ generates random numbers ranging from 0 to 1. The Beta operator draws inspiration from the mutation operator utilized in the Genetic Algorithm (GA) [44]. When assigning values to new solutions, we have two options: randomly selecting values from the same domain with a probability of $Beta = rand(0,\ 1)$, or utilizing the current solution. This process is illustrated as follows:

$$L_j^{''} = \begin{cases} L_j & if\ Beta > rang(0,\ 1) \\ L_j^{'} & otherwise \end{cases} \tag{6}$$

In Eq. (6) $L_j^{''}$ represents the *jth* updated location dimension in the solution while L_j and $L_j^{'}$ represent the previous solution and the neighborhood of the last solution, respectively. This version of hill climbing is effective, but its performance heavily relies on Beta and N. Determining the optimal values of these two parameters requires extensive experimentation. To avoid this limitation, AβHC was introduced. In AβHC, Beta and N are functions of the number of iterations. $N(z)$ represents the functional measure of N in the *zth* iteration and can be determined using Eq. (7).

$$N(z) = 1 - \frac{z^{\frac{1}{c}}}{MT^{\frac{1}{c}}} \ where\ c = constant \tag{7}$$

In this context, "MT" refers to the maximum number of iterations, and "z" represents the current iteration number. The value of Beta in the *zth* iteration is denoted as $Beta(z)$, as shown below:

$$Beta(z) = \frac{(\text{Ma} - \text{Mi}) \times z}{MT} + Mi \tag{8}$$

In Eq. (8), "Ma" and "Mi" correspond to Beta's upper and lower limits, while "z" represents the current iteration number. If the newly generated neighbor $L^{''}$ outperforms the current best solution L, it replaces L.

4. Proposed method

This section presents our proposed method for classifying inverted eyelid images, as depicted in Fig. 1. The classification process involves multiple stages. First, the training datasets undergo preprocessing. Next, design attention-based feature extraction that integrates the Lambda layer with CBAM into the pretrained VGG16 model. Second, we propose AβHC with SSD to adjust the exploration-exploitation trade-off during the search process for FS to identify the most relevant features from the feature extraction model. Finally, we utilize the relevant selected features to train an MLP classifier, predicting whether the input inverted eyelid image is normal, TF, or TI. In the following subsections, we discuss each primary pipeline stage in detail.

4.1. Feature extraction model

In this study, we introduced an attention-based deep CNN model for extracting features from inverted eyelid images. We preprocessed the images before employing feature extraction models. The preprocessing typically involves cropping and resizing the original dataset images. To emphasize the central part of each image and eliminate irrelevant elements, we implemented central image cropping using OpenCV. This technique selects a submatrix of the original image matrix that includes only the pixels within a rectangular region defined by the desired output size and the top-left corner coordinates. After cropping, all images were resized to a standardized size of 224 × 224 pixels.

Our approach integrates an attention mechanism into the pretrained VGG16 model, omitting its fully connected layers and keeping the layer

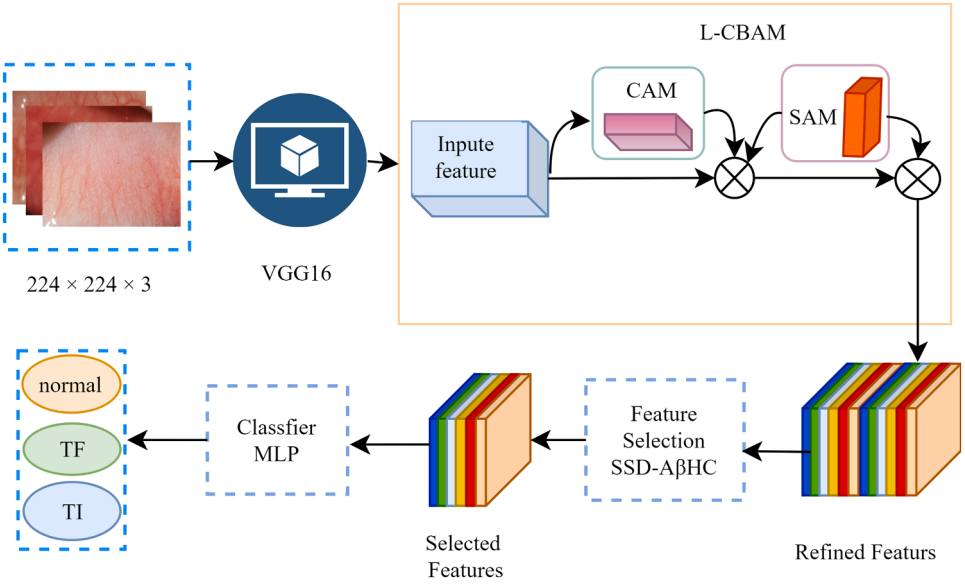


Fig. 1. A pipeline of the proposed model.

weights frozen to prevent new information acquisition during training. In place of the fully connected layers, we incorporate CBAM, as shown in Fig. 1. In a VGG16 network, the input image data X is transformed into a feature map F, which serves as the input for CBAM. CBAM consists of two independent sub-modules: the channel attention module (CAM) and the spatial attention module (SAM). These modules apply the attention mechanism to the channel and spatial dimensions, leading to parameter and computational efficiency [45].

4.1.1. Channel attention module

The channel attention module employs the global average pooling layer and the global maximum pooling layer to compress the feature maps along the spatial dimension to facilitate feature extraction and minimize information loss. Fig. 2 illustrates the channel attention module. The global average pooling layer captures comprehensive information, while the global maximum pooling layer captures feature variance information. The combination of these two layers outperforms individual layers [45]. The VGG16 convolutional layer generates a feature vector, a channel attention module process. This module assigns weights to the input feature based on the relationships between different channels [14]. Initially, the input feature $F \in R^{H\times W\times C}$ is considered as a collection of feature maps $Mch \in R^{c\times 1\times 1}$, which are then aggregated spatially. Global average pooling and global max pooling operations are applied to these feature maps, resulting in two-channel attention vectors: AvgPool $(F) \in R^{1\times 1\times c}$ and MaxPool $(F) \in R^{1\times 1\times c}$. An MLP with shared parameters then processes these vectors, and the outputs of the MLP are combined element-wise. Finally, the Sigmoid function maps the values to a range between 0 and 1. The resulting channel attention feature is obtained using Eq. (9).

$$M_{Ch}(F) = \sigma(MLP(AvgPool(F)) \oplus MLP(MaxPool(F))) \tag{9}$$

The element-wise summation is denoted by $\oplus$. AvgPool and MaxPool represent global average pooling and global max pooling operations, respectively, while σ represents the Sigmoid function defined in Eq. (10).

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{10}$$

L-CBAM: L-CBAM incorporates the Lambda layer into the channel attention sub-module of CBAM, as illustrated in Fig. 2. The Lambda layer normalizes the attention weights within the attention model. It applies after computing the attention scores, ensuring that the weights are normalized across different elements of the attention tensor. This normalization is achieved by dividing each element of the attention tensor by its maximum value, as shown in Eq. (11). By normalizing the attention weights; the model can prioritize the most relevant features and mitigate the impact of outliers or extreme values in the attention mechanism. To mathematically express the incorporation of the Lambda layer and the normalization of attention weights, let's consider the attention tensor $MLP(AvgPool(F)) \oplus MLP(MaxPool(F))$, which represents the computed attention scores. The normalization of the attention weights is as follows:

$$N = \frac{MLP(AvgPool(F)) \oplus MLP(MaxPool(F))}{\max(MLP(AvgPool(F)) \oplus MLP(MaxPool(F)))} \tag{11}$$

$$M_{Ch}(F) = \sigma(N) \tag{12}$$

where N represents the attention tensor after normalization, we obtain the normalized channel attention feature using Eq. (12). This normalization process ensures that the attention weights are within a consistent range and helps the model focus on the most relevant features. It also reduces the impact of outliers or extreme values in the attention mechanism, allowing for more reliable and robust feature selection.

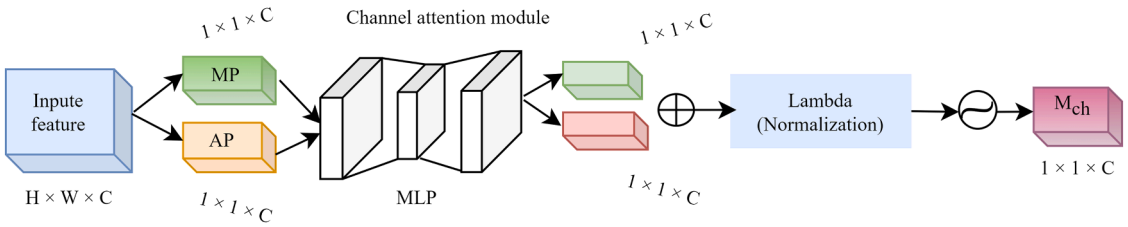


Fig. 2. Channel attention module.

4.1.2. Spatial attention module

The feature map F' generated by the channel attention module serves as the input feature map for this module [45]. Initially, the feature map $F^{'} \in R^{H\times W\times C}$ is aggregated along the channel axis using global max pooling and global average pooling operations, resulting in two spatial attention vectors: AvgPool $(F^{'}) \in R^{H\times W\times 1}$ and MaxPool $(F^{'}) \in R^{H\times W\times 1}$. These two vectors are then concatenated along the channel dimension and convolved using a standard convolution layer. Finally, the Sigmoid function is applied to generate the spatial attention feature, defined in Eq. (13).

$$M_{sp}(F^{'}) = \sigma\left(f^{7\times 7}(\text{AvgPool }(F^{'});\ \text{MaxPool }(F^{'}))\right) \tag{13}$$

The spatial attention module utilizes a convolution kernel f with dimensions of 7×7. The Sigmoid function is denoted by σ. Fig. 3 illustrates the structure of the spatial attention module.

After passing through the CAM and SAM modules, the feature map F undergoes refinement and transforms into $F^{''}$, representing the final feature extracted by our feature extraction network. Given a feature map $F \in R^{C\times H\times W}$, L-CBAM generates a 1D channel attention map $Mch \in R^{C\times 1\times 1}$ and a 2D spatial attention map $Msp \in R^{1\times H\times W}$ sequentially. We then utilize these attention maps to obtain the refined feature map F''. To broadcast channel attention values across the spatial dimension, element-wise multiplication (denoted as $\otimes$) is employed as described in Eqs. (14) and (15).

$$F^{'} = M_{ch}(F) \otimes F, \tag{14}$$

$$F^{''} = M_{sp}(F^{'}) \otimes F^{'}, \tag{15}$$

This refined feature, $F^{''}$, is then utilized in our feature selection model to identify and select the most relevant features. These selected features are subsequently fed into an MLP for the final classification task.

4.2. Feature selection and classification

FS techniques aim to improve the accuracy of learning models by identifying the optimal subset of features from a primary set and reducing the dimensionality by eliminating redundant and irrelevant features. However, since the problem is binary, most meta-heuristics cannot solve it, as they are designed for continuous variables. The literature has proposed various strategies to adapt meta-heuristics to binary problems, including utilizing transfer functions. The Particle Swarm Optimization (PSO) [42] algorithm updates solutions using a velocity vector. A transfer function is often employed to assess the chances of modifying a position in binary space. This function typically maps input values to output values and adjusts the agent's behavior in response to its surroundings. In the case of binary space, the transfer function determines the likelihood of a particular position being updated or not. In this study, we employed the transfer function [20] illustrated in Eq. (16) to investigate its effectiveness.

$$V(x) = \frac{|x|}{\sqrt{1+x^2}} \tag{16}$$

The position of an agent is modified based on Eq. (17) using the V-shaped transformation function.

$$LB_j^{k+1} = \begin{cases} c\left(\text{LB}_j^k\right) \ if\ V\left(\text{LB}_j^{k+1}\right) > rand(0,\ 1) \\ LB_j^{k+1} \qquad otherwise \end{cases} \tag{17}$$

Eq. (17) defines the modified location of an agent as LB_j^{k+1}, where LB_j^k represents the agent's location at a specific time (with k denoting the iteration number and j indicating the number of dimensions). The function $rand(0,\ 1)$ generates random numbers between 0 and 1. The complement function for all binary x, denoted by $c(x) = 1 - x$. In each iteration, we modify the agent location, utilizing AβHC to achieve a higher fitness value by optimizing the agents' positions. The AβHC local search technique improves the exploitation potential of the SSD algorithm, as shown in Algorithm 1.

Fitness function: The primary goal of any FS method is to minimize the number of features while maximizing the classification accuracy of a classification problem [29]. This strategy is common in wrapper-based, multiobjective FS methodologies [26]. Additionally, other objectives can include obtaining a trade-off between the feature cost and the classification accuracy. For example, Hu et al. [46] proposed a fuzzy multiobjective FS method called PSOMOFS, which utilizes particle swarm optimization to optimize feature sets with fuzzy costs for cost-based feature selection. We aim to minimize the fitness function, reducing the classification error and the number of selected features. This approach is considered multiobjective, integrating the two objectives into the fitness function.

Nonetheless, it is acknowledged that these two objectives often exhibit inherent conflicts. Simultaneously optimizing both objectives is advocated for its capacity to more faithfully capture the nuanced decision-making dynamics inherent in FS problems encountered within practical applications [47]. In this study, we utilize AβHC to identify the optimal feature subset and evaluate its accuracy using MLP classifiers. We calculate each feature subset's fitness value and compare it with others to determine the best subset, with lower fitness values indicating better subsets. In our case, we prioritize selecting the minimum number of features while reducing the classification error.

$$Fitness\ function\ = w \times \propto + (1 - w) \times \frac{|s|}{|d|} \tag{18}$$

In Eq. (18), the defined variables are as crucial to the feature selection process. The symbol $|d|$ denotes the total number of features in the dataset, while $|s|$ represents the number of features included in the candidate solution (i.e., the selected feature set). The variable $\propto$ signifies the classification error incurred when using the particular feature subset. Lastly, the variable w, which falls within the range of [0, 1], is utilized to assign a weight to the number of features and the classification error, indicating their respective importance.

5. Dataset description and parameter setting

5.1. Dataset description

We used an open dataset of field-collected conjunctival images from clinical trial participants in Niger and Ethiopia, as utilized by Kim et al. [1]. Trained field workers followed a standardized protocol to capture the images. Three experts independently classified each image using the WHO simplified system for trachomatous inflammation. The original dataset is publicly hosted on the Figshare[1] platform for easy access and use. It comprises photographs collected under field conditions, varying lighting conditions, camera angles, and distances. The original raw images are in color JPEG format, with dimensions ranging from 4288 × 2848 to 3008 × 2000 pixels. We preprocessed the images as discussed in Section 4.1, resizing them to a fixed dimension of 224 × 224 before inputting them into the VGG16 model. The dataset comprised 1656 labeled conjunctival images, with 39% showing trachomatous changes. Among them, 22% had TF, 7% had TI, and 10% had both TF and TI, as shown in Table 1. The original authors obtained ethical approval. The dataset was randomly divided into three sets to evaluate the model's performance: 80% for training, 10% for validation, and 10% for testing.

5.2. Parameter setting for feature extraction

We conducted all experiments on Google Colab, utilizing its GPU for

[1] https://doi.org/10.6084/m9.figshare.7551053.v1.

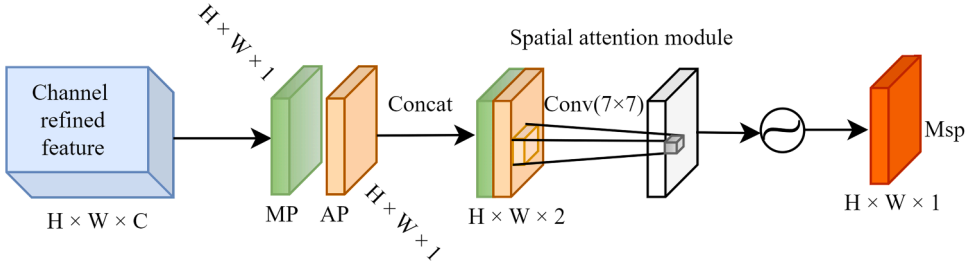


Fig. 3. Spatial attention module.

Algorithm 1
Pseudocode of the SSD-AβHC algorithm.

Set the values of population size and Max-iteration.
Randomly initialize velocity, position, personal best (pbest), and global best (gbest).
for $z = 1$ to max-iteration do
  for $p = 1$ to population-size do
    Set agent $= L$
    Determine the initial fitness value of agent L by applying the fitness function.
    Update gbest by randomly selecting one of the top 3 sites.
    Update pbest based on the fitness of the current agent.
    Check if ($r1 < 0.5$) then
    set $V_K^{T+1} = h * \sin(rand(0,1))(PB_K^T - L_K^T) + \sin(rand(0,1))(MGB_K^T - L_K^T)$
    else
    set $V_K^{T+1} = h * \cos(rand(0,1))(PB_K^T - L_K^T) + \cos(rand(0,1))(MGB_K^T - L_K^T)$
    End if
    Calculate $V(x) = \frac{|x|}{\sqrt{1 + x^2}}$
    Update the position of LB_j^{k+1} using Eq. (17)
    Set the new location of LB_j^{k+1} = local search (LB_j^{k+1})
    End for
  Set $h^{T+1} = r \times h^T$
End for

Table 1
Clinical categories of conjunctival images, where "n" is the number of images and (%) is the number of images in percentage.

Label	n (%)	Subcategories	
Normal	1019 (61)		
Infected	637 (39)	TF, n (%)	365 (22)
		TI, n (%)	110 (7)
		TF and TI, n (%)	162(10)

computation. The Colab environment used for the experiments had 12 GB of RAM and an NVIDIA Tesla K80 GPU. We conducted experiments using various standard parameters for learning rate and batch size to determine the most effective combination. We explored initial learning rates of {1e-2, 1e-3, 1e-4} and {16, 32, 64} batch sizes. Table 2 presents a detailed overview of the outcomes of these experiments. Importantly, we observed that we achieved the optimal solution with a batch size of 32 and a learning rate of 1e-3 (0.001). The training process encompasses 50 epochs. We employed the Adam optimizer and the commonly used cross-entropy loss function to optimize the deep learner.

Table 2
Comparison of various batch sizes and learning rates in the attention-based feature extraction model.

Learning rate	Batch size	Accuracy in (%)
	16	87.6
1e-2	32	89.1
	64	88.1
	16	88.9
1e-3	**32**	**89.3**
	64	88.2
	16	86.4
1e-4	32	85.9
	64	85.2

5.3. Parameter setting for feature selection

This section highlights the significance of the fitness function and parameter values in the SSD algorithm. The primary objective of the algorithm is to minimize the fitness function, as defined in Eq. (18), which involves reducing the number of features and the classification error. Both factors are crucial in this task, as using too many features can lead to overfitting while using too few features can result in underfitting. To balance these factors, one can adjust the weight parameter "w" in the fitness function to assign more importance to either feature reduction or classification accuracy increase. Through experimentation, we found that setting "w'' to 0.2 prioritizes the reduction of classification error, leading to increased accuracy. To examine how different parameter values affect the algorithm's performance, we conducted a series of experiments in which we modified the values of "h" and "r" within the search space. Our results indicated that increasing the value of "h" led to improved classification accuracy. However, we observed that decreasing the value of "r" initially led to better performance, but eventually, the classification accuracy began to decline after reaching a peak. This decline occurs due to overfitting when the r value becomes too low. After further experiments, we determined that a fixed value of "$h = 100$'' and "$r = 0.9$'' yields the maximum classification accuracy.

The proposed method and all other comparison algorithms were assessed across population sizes of {5, 10, 20, 30, 40, 50}. Based on the findings in Fig. 8(a) and (b), we have set the population size to 30 and the maximum number of iterations to 100.

5.4. Performance evaluation metrics

Our analysis used precision, recall, and F1-score as performance evaluation metrics. Additionally, evaluating the model involves assessing the receiver operating characteristic (ROC) curve and the area under the curve (AUC). The ROC curve illustrates the relationship between the false positive rate (specificity) and the true positive rate (recall) displayed on the horizontal and vertical axes. Apart from accuracy and ROC, the number of selected features is also a crucial metric in our analysis. However, it remains essential to maintain accuracy during predictions, even when working with fewer features. Therefore, the FS algorithm aims to achieve high accuracy while selecting the minimum number of features. The number of selected features influences this trade-off between prediction accuracy and inference time.

6. Result and discussion

6.1. Feature extraction performance

In this section, we assess the effectiveness of an attention-based deep feature extraction model. Initially, we conducted experiments using end-

to-end VGG16 models for deep feature extraction and VGG16 models with various attention modules, as shown in Table 3. The results indicate that utilizing the end-to-end VGG16 pretrained model leads to an accuracy of 86.6%. When employing the CBAM, the accuracy improves to 88.6%. However, we achieved the highest classification accuracy of 89.3% using the proposed Lambda normalization layer within the CBAM. To account for the impact of class imbalance, we reported the precision, recall, and F1-score values using the weighted average.

Fig. 4 illustrates the training accuracy and training loss of different models. Initially, in the earlier epochs, the accuracy score of the training set was lower. However, as the training progresses, the accuracy of the training set gradually improves, indicating better performance. This convergence suggests that the model effectively learns and adapts to the data, producing positive results. The observed pattern in the loss values follows a similar trend to the accuracy values. During the earlier epochs, the loss score of the training set was high. However, as the training progresses, the loss of the training set gradually decreases. These findings highlight the positive impact of transfer learning on the model's performance by mitigating overfitting and facilitating faster convergence.

Fig. 5 illustrates that the ROC curve of the L-CBAM-VGG16 model surpasses the other two models on the testing sets, achieving an impressive AUC value of 0.95, 0.95, and 0.97 for the three classes separately. This comparison further confirms the strong generalization ability of the L-CBAM-VGG16 model when faced with different data sources. These experiments demonstrate that the proposed L-CBAM-VGG16 model outperforms the other two models.

6.2. Feature selection performance

In this section, we present our approach to utilizing a local search-driven FS algorithm to improve the accuracy of the classification model and reduce the number of features. Since this FS is a wrapper method, a machine learning classifier is employed to select the subset of features with the highest classification accuracy. The fitness function in Section 4.2 is utilized in all experiments to balance classification accuracy and the number of selected features. Regardless of the goal to minimize error or maximize accuracy, the solution with the lowest fitness value is chosen as the best solution for predictive purposes, as it represents the optimal compromise between accuracy and the number of selected features. We assess each algorithm according to the fitness function. Ultimately, the number of selected features and classification accuracy are the final criteria for evaluating the model's performance. The local search algorithm explores the feature space, evaluates subsets of features, and updates the current solution based on the evaluation results. It repeats this process until it finds a good subset of features. Our proposed method achieved a high level of accuracy (93.3%), precision (94.4%), recall (93.3%), and F1 score (93.5%) using only 19.7% of selected features, as demonstrated in Table 4.

6.3. Comparison with different feature selection methods

This section compares the experimental results of our proposed method with various FS methods. The methods under comparison include Gravitational Search Algorithm (GSA) [48], GA [44], Whale Optimization Algorithm (WOA) [49], PSO [42], HS algorithm [35], Equilibrium Optimizer (EO) [50], Binary Bat Algorithm (BBA) [51], Gray-wolf Optimization (GWO) [52], and Sine Cosine Algorithm (SCA) [53]. Our experiments also utilized local search methods LAHC [54] and AβHC [55]. These methods are designed to explore the feature space, evaluate subsets of features, and update the current solution based on evaluation results.

Meta-heuristic-based FS algorithms require numerous mathematical operations to identify the optimal feature subset, utilizing sets of equations aided by various parameters critical for controlling the optimization process. In this study, we used the standard parameter values for the different algorithms, as listed in Table 5. Table 6 displays the experimental results of the local search integrated with the proposed and comparison feature selection method on the testing dataset. These results indicate that the local search AβHC method is more effective in finding the optimal feature subset and improving the performance of the classification model.

Acknowledging the involvement of the LAHC and AβHC local search approaches in the feature selection process is essential. These methods rely on the hill-climbing optimization technique. The AβHC-based FS algorithm generally outperforms the LAHC-based FS algorithm in terms of accuracy and the number of selected features. For instance, the SSD+AβHC algorithm achieves 93.3% accuracy with only 101 selected features, while the SSD+LAHC algorithm achieves 91.6% accuracy with 228 selected features. WOA+LAHC and PSO+LAHC perform slightly better regarding selected features but are outperformed by the proposed method in classification accuracy. The potential reason for this is that the solutions generated by WOA and PSO initially converge to regions of the search space with fewer features. In that case, LAHC may not have sufficient opportunity to explore and expand the feature subset since LAHC selects the agent at each iteration. The algorithm with fewer selected features (WOA+LAHC and PSO+LAHC) may have selected features that are more complex or less interpretable compared to the algorithm with more selected features (SSD+LAHC), resulting in a lack of interpretability despite having fewer features. Additionally, the quality and relevance of the selected features also play a crucial role in determining the interpretability or computational cost of the algorithm. For the FS problem, the first and most important indicator or objective is classification accuracy [46].

In the AβHC local search, the proposed method surpasses WOA+LAHC and PSO+LAHC in selected features and classification accuracy. This comparison suggests that the AβHC-based FS algorithm may have advantages over the LAHC-based FS algorithm, such as better convergence properties or more effective feature selection mechanisms, since the AβHC method considers the agent within a particular neighborhood. LAHC may encounter challenges in high-dimensional feature spaces, where exhaustive search becomes computationally burdensome; its exploration-exploitation balance might not be ideal for all datasets, potentially resulting in premature or sluggish convergence. In contrast, the adaptive nature of SSD+AβHC enables it to dynamically adjust its exploration-exploitation balance, potentially leading to quicker convergence and improved solutions.

The simplicity and computational efficiency of the SSD method render it suitable for high-dimensional datasets or problems with limited computational resources. The SSD-based FS technique employs sine and cosine functions to diversify the movement direction of the agents, a crucial feature enhancing exploration. The algorithm maintains stability between exploration and exploitation by integrating the parameter "h" in Eq. (3), facilitating convergence towards superior solutions.

While the comparison method (referring to other optimization techniques like GSA, GA, WOA, PSO, HS, EO, BBA, GWA, and SCA) yields competitive results, it falls short compared to SSD. The comparison method may suffer from suboptimal exploration-exploitation strategies, leading to premature convergence or getting trapped in local optima. Its feature selection mechanism may not be as adaptive or efficient as SSD, resulting in less effective feature subset selection and model training. GA+AβHC yields relatively poor results in accuracy and the number of features due to encountering redundant representation.

Table 3
Illustrates the accuracy comparison of different attention modules integrated with the VGG16 model.

Module	Precision	Recall	F1	Accuracy
VGG16	87	87	86	86.6
CBAM-VGG16	88	89	88	88.6
L-CBAM-VGG16	**89**	**89**	**89**	**89.3**

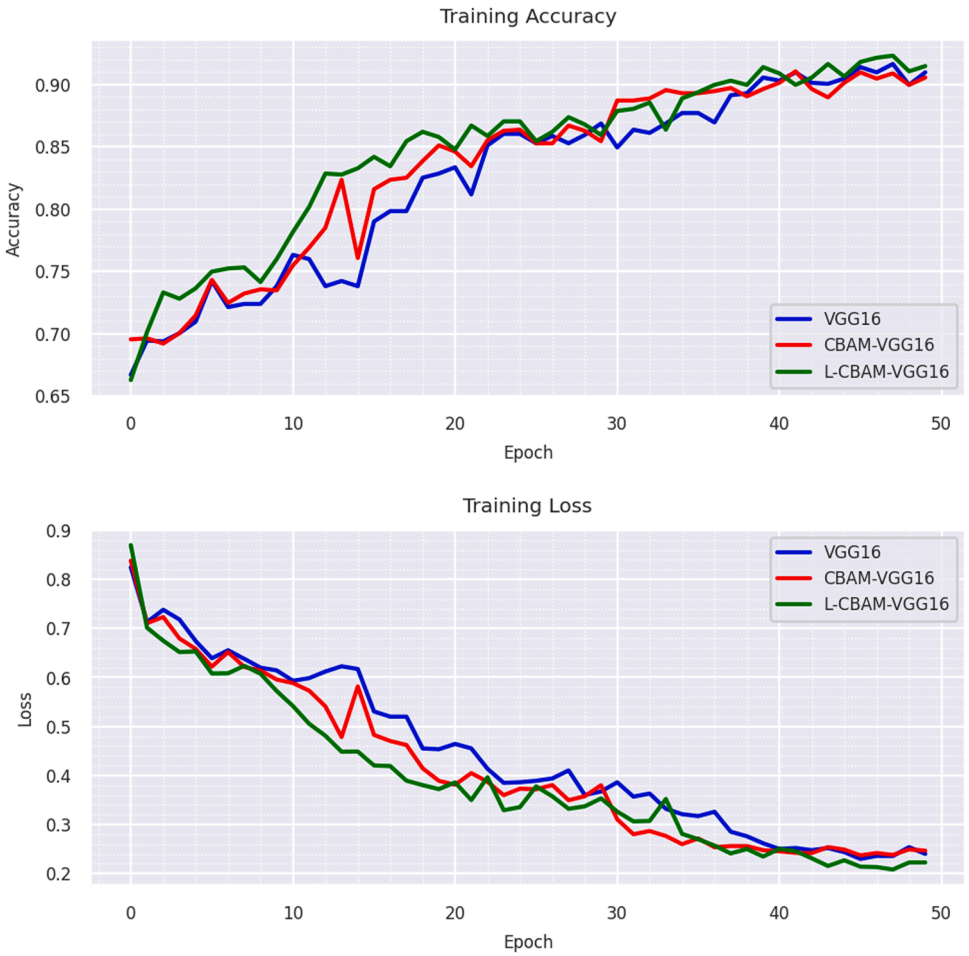


Fig. 4. Shows the training accuracy and loss of the VGG16 model and different attention modules integrated with it based on the active trachoma image dataset. The x-axis represents the number of epochs.

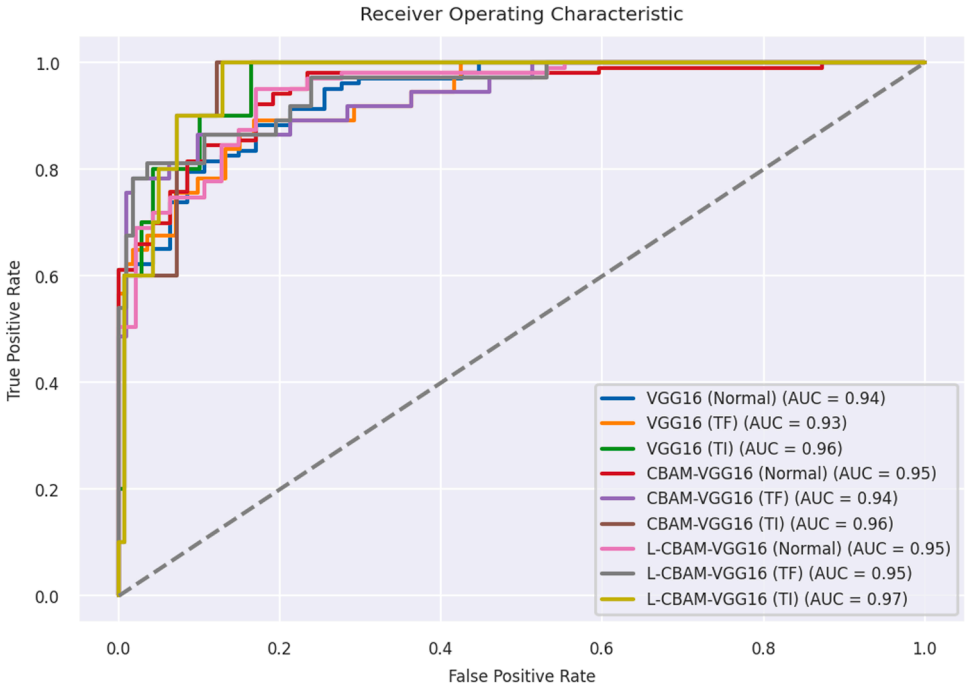


Fig. 5. ROC curves for the testing sets; each class's ROC and AUC values are plotted separately.

Table 4
Accuracy, precision, recall, F1, and number of selected features (SF) of the proposed model.

Proposed model	Accuracy	Precision	Recall	F1	SF
	93.3	94.4	93.3	93.5	19.7

Table 5
Shows the values of the corresponding hyperparameters for different meta-heuristic-based FS algorithms.

FS algorithms	Parameter(s)	Value(s)
shared parameters	Population size	20
	Number of iterations	100
	Weight for accuracy ($\propto$)	$\propto = 0.98$
GSA	Initial gravitational cont. (G_{init})	$G_{init} = 6$
	Constant (ε)	0.00001
WOA	Encircling parameter (a)	a lies in [0 2]
	Shape of spiral (b)	$b = 1$
HS	Harmony memory Considering rate	$HMCR = 0.90$
GA	Gene selection	Roulette wheel
	Crossover probability	0.4
	Mutation probability	0.3
PSO	Inertia weight (I_w)	I_w lies in [0 1]
	Coefficients (r_1, r_2)	r_1,r_2 lies in [0 1]
EO	Pool size	4
	Constants (a_1, a_2)	$a\_1 = 2$, $a\_2 = 1$
	Generation rate (GP)	$GP = 0.5$
BBA	Loudness	$L = 1$
	Pulse Emission Rate	$PER = 0.15$
	loudness decay rate (alpha)	$\alpha = 0.95$
	pulse rate (Beta)	$\beta = 0.5$
GWO	Convergence operator (a)	a lies in [0 2]
SCA	Constant (a)	$a = 3$
	Movement direction (r1)	r1 lies in [0 3]

Redundant representation refers to duplicated or uninformative features, which can increase computational complexity and hinder feature selection efficiency.

Furthermore, Figs. 6 and 7 provide a comparative analysis of different combinations of meta-heuristics and local search algorithms. The study demonstrates that the SSD+AβHC approach outperforms other combinations in classification accuracy, highlighting the superiority of the proposed approach and emphasizing its potential for practical applications in medical image analysis. Integrating the AβHC local search approach aids the algorithm in improving its solutions by overcoming local optima, resulting in superior outcomes.

Fig. 8(a) shows the impact of population size on the classification accuracy of a classifier using the suggested feature selection method and comparison algorithms. Notably, increasing the population size does not yield any further improvements in accuracy beyond a certain threshold. Fig. 8(b) illustrates how the proposed and comparison algorithms' fitness values change as the number of iterations increases. Increasing the value of this parameter results in a proportional increase in the algorithm's execution time while maintaining the fitness value of the optimal search agent unaltered.

Statistical test: We conducted a statistical test, specifically the Wilcoxon rank-sum test [56], to determine the statistical significance of the results obtained using the current method. Table 7 presents the p-values resulting from conducting the Wilcoxon test to compare the performance of different algorithms. The calculated p-values for each algorithm are less than the predetermined significance level of 0.05. Rejecting the null hypothesis based on these p-values suggests sufficient evidence to support a significant difference in accuracy among the compared methods.

Table 8 depicts the effectiveness of various modules of our proposed method. The VGG16 model with attention-based achieved a classification accuracy of 89.3%. Moreover, using the SSD algorithm for feature selection improved accuracy by 90%. The experimental results demonstrate that the accuracy was further enhanced using SSD-based global feature selection and incorporating the AβHC local search method with 93.3%. Thus, based on these findings, we can infer that the individual components of our proposed method, namely utilizing attention-based VGG16 for deep feature extraction and employing the SSD-AβHC method for feature selection, are effective.

6.4. Comparison with state-of-the-art approaches

Table 9 compares the effectiveness of the proposed model to the state of the art using various models and measures. Each model has its approach for determining whether an active trachoma inverted eyelid image is normal, TF, or TI. The first study used a CNN model to classify TF and TI cases separately, achieving 72% accuracy for TF cases and 85% for TI cases. The second study used ResNet101 models to detect TF cases and achieved 88% accuracy. The third study used automated machine learning (AutoML) in diagnosing trachoma using conjunctiva images to classify TF, TI, and (TF and/or TI) cases separately and achieved 88%, 93%, and 83% accuracy, respectively. However, the proposed model utilized attention-based VGG16 feature extraction and SSD, a global search algorithm, combined with AβHC, a local search algorithm, to select the optimal and reduced feature subset. As a result, it achieved increased accuracy scores of 93.3% for multiclass classification, 94.6% for TF classification, 97% for TI classification, and 100% for TF and/or TI classification.

7. Conclusion and feature direction

This study addresses the challenge of accurately classifying active trachoma using photographic images. The proposed approach utilizes a two-stage classification model combining an attention-based deep learning model for feature extraction of salient textures. Additionally, it employs a feature selection technique that combines SSD and AβHC algorithms to select an optimal and reduced feature subset. We obtain

Table 6
Shows the results of applying different FS algorithms with the LAHC and AβHC local search algorithms. The proposed algorithm is highlighted in bold.

FS algorithms	LAHC + FS algorithm				*AβHC* + FS algorithm			
	Accuracy (%)	No. of features		Selected features (%)	Accuracy (%)	No. of features		Selected features (%)
		Initially	Finally			Initially	Finally	
GSA	91.2	512	246	48	91.5	512	246	48
GA	87.7	512	251	49	87.7	512	251	49
WOA	91.3	512	149	29.1	91.3	512	172	33.6
PSO	91.3	512	188	36.7	91.4	512	222	43.3
HS	91.2	512	307	59.3	91.7	512	210	41.0
EO	91	512	249	48.6	91.2	512	242	47.2
BBA	91.2	512	242	47.2	91.4	512	225	43.9
GWO	91.4	512	237	46.3	91.6	512	220	43
SCA	91.1	512	245	47.9	91.3	512	230	44.9
SSD	91.6	512	228	44.5	**93.3**	**512**	**101**	**19.7**

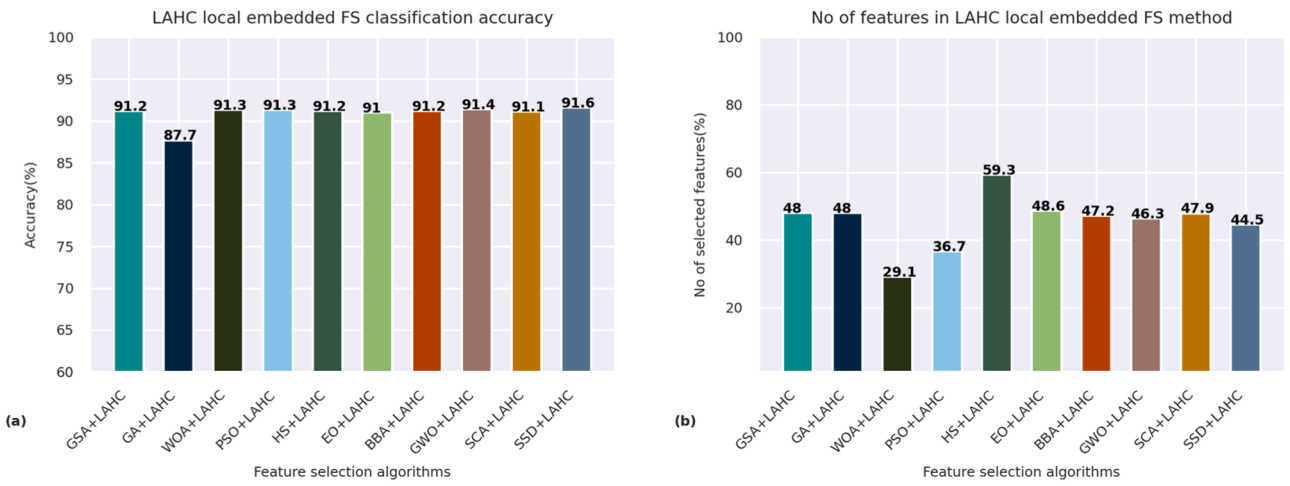


Fig. 6. Results of combining various FS algorithms with LAHC local search regarding two metrics: (a) classification accuracy and (b) the proportion of selected features.

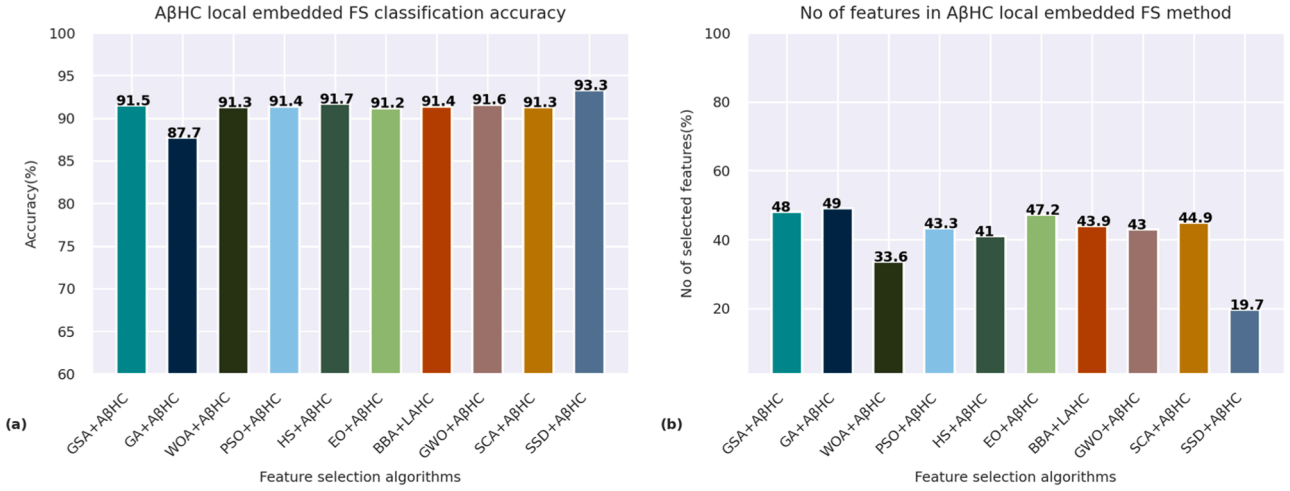


Fig. 7. Results of combining various FS algorithms with AβHC local search in terms of two metrics: (a) classification accuracy and (b) the proportion of selected features.

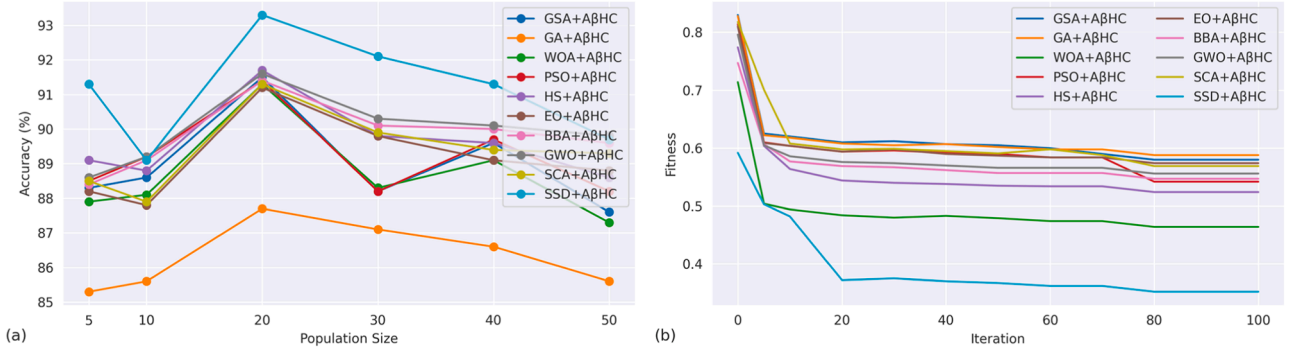


Fig. 8. (a) demonstrates how population size affects the performance of the proposed and comparison algorithms, measured in classification accuracy. Meanwhile, (b) illustrates the changes in fitness values for the proposed and comparison algorithms as the number of iterations increases.

the final classification results using an MLP classifier with this subset. Experimental results indicate that the attention-based VGG16 model and the SSD-AβHC feature selection algorithm outperform other existing methods, achieving an accuracy score of 93.3% with only 19.7% of the extracted features. This performance is superior to many of the algorithms used for comparison. The proposed approach has demonstrated excellent performance compared to recent works utilizing the same datasets. Thus, the key advantage of the proposed method arises from its capability to boost classification performance, strengthen discriminative abilities, efficiently explore local search spaces, and attain superior or comparable performance via the integration of attention-based feature extraction with adaptive feature selection.

Table 7
Presents the p-values resulting from conducting the Wilcoxon test to compare the performance of different algorithms.

FS algorithm with AβHC	p-values
GSA	0.0090
GA	0.0090
WOA	0.0090
PSO	0.0090
HS	0.0090
EO	0.0090
BBA	0.0090
GWO	0.0090
SCA	0.0090

Table 8
Presents the classification accuracy of different module combinations in the proposed model.

Method	Accuracy (%)
$L-CBAM-VGG16$	89.3
$L-CBAM-VGG16+SSD+MLP$	90
$L-CBAM-VGG16+SSD-ABHC+MLP$	93.3

Table 9
Proposed model comparison with state-of-the-art approaches, N/A: Not Available.

Model	Accuracy		
	TF vs. normal	TI vs. normal	TF and/or TI vs. normal
Kim et al. [1]	72	85	N/A
Socia et al. [12]	88	N/A	N/A
Milad et al. [2]	88	93	83
Proposed	94.6	97	100
Proposed	93.3		

Conversely, the main limitations of the proposed model are: first, the use of random initialization in the optimization algorithm, potentially sacrificing accuracy and convergence time, suggesting that exploration of techniques like chaotic maps can address this challenge; and second, the possibility of early convergence for specific inputs. Future research could explore multi-modal cross-fusion analysis, integrating image data with patient clinical data, including patient demographics, medical history, symptom severity scores, and treatment history. This exploration could involve employing cascade operations, weight scaling, screening, filtering, and other methods to establish connections between cross-modal data, thus enhancing the fusion learning ability of the model. Consequently, this approach could significantly improve the performance of active trachoma diagnosis systems by effectively mapping heterogeneous data features and addressing the issues of insufficient feature richness encountered when using single-mode data.

Ethics approval

The original authors obtained ethical approval for the dataset.

CRediT authorship contribution statement

Mulugeta Shitie Zewudie: Writing – review & editing, Writing – original draft, Resources, Methodology, Data curation, Conceptualization. **Shengwu Xiong:** Funding acquisition, Investigation, Methodology, Supervision. **Xiaohan Yu:** Writing – review & editing, Writing – original draft, Validation, Methodology. **Xiaoyu Wu:** Resources, Data curation. **Moges Ahmed Mehamed:** Resources, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The original images for this study are available on Figshare: https://doi.org/10.6084/m9.figshare.7551053.v1. We can provide processed data via email.

Acknowledgment

This work was in part supported by the National Key Research and Development Program of China (Grant No. 2022ZD0160604) and NSFC (Grant No. 62176194), and the Key Research and Development Program of Hubei Province (Grant No. 2023BAB083), the Project of Sanya Yazhou Bay Science and Technology City (Grant No. SCKJ-JYRC-2022-76, SKJC-2022-PTDX-031), and the Project of Sanya Science and Education Innovation Park of Wuhan University of Technology (Grant No. 2021KF0031).

References

[1] M.C. Kim, et al., Sensitivity and specificity of computer vision classification of eyelid photographs for programmatic trachoma assessment, PLoS. One 14 (2) (2019) e0210463.
[2] D. Milad, F. Antaki, M.-C. Robert, R. Duval, Development and deployment of a smartphone application for diagnosing trachoma: leveraging code-free deep learning and edge artificial intelligence, Saudi J. Ophthalmol. (2023).
[3] S.A. Al-Eryani, E.Y.A. Alshamahi, H.A. Al-Shamahy, A.A.M. Al-Ankoshy, Prevalence and risk factors for Trachoma among primary school children in Sana'a city, Yemen, J. Pharmaceut. Res. 6 (4) (2021) 19–25.
[4] S. Kadry, V. Rajinikanth, R.G. Crespo, E. Verdú, Automated detection of age-related macular degeneration using a pre-trained deep-learning scheme, J. Supercomput. (2022) 1–20.
[5] A. Govindaiah, M.A. Hussain, R.T. Smith, A. Bhuiyan, Deep convolutional neural network based screening and assessment of age-related macular degeneration from fundus images, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI2018), IEEE, 2018, pp. 1525–1528.
[6] S. Diao, et al., Classification and segmentation of OCT images for age-related macular degeneration based on dual guidance networks, Biomed. Signal. Process. Control 84 (2023) 104810.
[7] P. Saranya, R. Pranati, S.S. Patro, Detection and classification of red lesions from retinal images for diabetic retinopathy detection using deep learning models, Multimed. Tools. Appl. (2023) 1–21.
[8] X. Qin, D. Chen, Y. Zhan, D. Yin, Classification of diabetic retinopathy based on improved deep forest model, Biomed. Signal. Process. Control 79 (2023) 104020.
[9] X. Li, X. Hu, L. Yu, L. Zhu, C.-W. Fu, P.-A. Heng, CANet: cross-disease attention network for joint diabetic retinopathy and diabetic macular edema grading, IEEe Trans. Med. ImAging 39 (5) (2019) 1483–1493.
[10] N.J. Shyla, W.S. Emmanuel, Glaucoma detection and classification using modified level set segmentation and pattern classification neural network, Multimed. Tools. Appl. 82 (10) (2023) 15797–15815.
[11] P.Y. Kim, et al., Novel fractal feature-based multiclass glaucoma detection and progression prediction, IEEe J. Biomed. Health Inform. 17 (2) (2013) 269–276.
[12] D. Socia, C.J. Brady, S.K. West, R.C. Cockrell, Detection of trachoma using machine learning approaches, PLoS. Negl. Trop. Dis. 16 (12) (2022) e0010943.
[13] Y. Xue, C. Zhang, F. Neri, M. Gabbouj, Y. Zhang, An external attention-based feature ranker for large-scale feature selection, Knowl. Based. Syst. 281 (2023) 111084.
[14] Y. Yao, Z. Zhang, X. Ni, Z. Shen, L. Chen, D. Xu, CGNet: detecting computer-generated images based on transfer learning with attention module, Signal Process.: Image Commun. 105 (2022) 116692.
[15] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, R. Menaka, Attention embedded residual CNN for disease detection in tomato leaves, Appl. Soft. Comput. 86 (2020) 105933.
[16] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.
[17] Y.-D. Zhang, Z. Zhang, X. Zhang, S.-H. Wang, MIDCAN: a multiple input deep convolutional attention network for Covid-19 diagnosis based on chest CT and chest X-ray, Pattern. Recognit. Lett. 150 (2021) 8–16.
[18] C. Li, S. Li, H. Wang, F. Gu, A.D. Ball, Attention-based deep meta-transfer learning for few-shot fine-grained fault diagnosis, Knowl. Based. Syst. 264 (2023) 110345.

[19] R. Bandyopadhyay, A. Basu, E. Cuevas, R. Sarkar, Harris Hawks optimisation with Simulated Annealing as a deep feature selection method for screening of COVID-19 CT-scans, Appl. Soft. Comput. 111 (2021) 107698.
[20] B. Chatterjee, T. Bhattacharyya, K.K. Ghosh, P.K. Singh, Z.W. Geem, R. Sarkar, Late acceptance hill climbing based social ski driver algorithm for feature selection, IEEe Access. 8 (2020) 75393–75408.
[21] G. Nijaguna, J.A. Babu, B. Parameshachari, R.P. de Prado, J. Frnda, Quantum fruit fly algorithm and ResNet50-VGG16 for medical diagnosis, Appl. Soft. Comput. 136 (2023) 110055.
[22] Y. Xue, X. Cai, W. Jia, Particle swarm optimization based on filter-based population initialization method for feature selection in classification, J. Ambient. Intell. Humaniz. Comput. 14 (6) (2023) 7355–7366.
[23] Y. Xue, B. Xue, M. Zhang, Self-adaptive particle swarm optimization for large-scale feature selection in classification, ACM Trans. Knowl. Discov. Data (TKDD) 13 (5) (2019) 1–27.
[24] Y. Xue, Y. Zhao, Structure and weights search for classification with feature selection based on brain storm optimization algorithm, Appl. Intell. 52 (5) (2022) 5857–5866.
[25] Y. Xue, H. Zhu, J. Liang, A. Słowik, Adaptive crossover operator based multiobjective binary genetic algorithm for feature selection in classification, Knowl. Based. Syst. 227 (2021) 107218.
[26] R. Jiao, B.H. Nguyen, B. Xue, M. Zhang, A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges, IEEE Trans. Evolut. Comput. (2023).
[27] S. Ahmed, K.K. Ghosh, L. Garcia-Hernandez, A. Abraham, R. Sarkar, Improved coral reefs optimization with adaptive β-hill climbing for feature selection, Neural Comput. Appl. 33 (12) (2021) 6467–6486.
[28] P. Pramanik, S. Mukhopadhyay, S. Mirjalili, R. Sarkar, Deep feature selection using local search embedded social ski-driver optimization algorithm for breast cancer detection in mammograms, Neural Comput. Appl. 35 (7) (2023) 5479–5499.
[29] A. Basu, K.H. Sheikh, E. Cuevas, R. Sarkar, COVID-19 detection from CT scans using a two-stage framework, Expert. Syst. Appl. 193 (2022) 116377.
[30] B. Yenegeta, Y. Assabie, TrachomaNet: detection and grading of trachoma using texture feature based deep convolutional neural network, Multimed. Tools. Appl. (2022) 1–26.
[31] H.E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M.E. Maros, T. Ganslandt, Transfer learning for medical image classification: a literature review, BMC. Med. ImAging 22 (1) (2022) 69.
[32] M. Shaha, M. Pawar, Transfer learning for image classification, in: 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, 2018, pp. 656–660.
[33] C. Desai, Image classification using transfer learning and deep learning, Int. J. Eng. Comput. Sci. 10 (9) (2021).
[34] H. Ghizlane, R. Jamal, M.A. Mahraz, Y. Ali, T. Hamid, Spam image detection based on convolutional block attention module, in: 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), IEEE, 2022, pp. 1–4.
[35] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation. 76 (2) (2001) 60–68.
[36] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[37] F. Wang, et al., Residual attention network for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164.
[38] Y. Xue, C. Lu, F. Neri, J. Qin, Improved differentiable architecture search with multi-stage progressive partial channel connections, IEEe Trans. Emerg. Top. Comput. Intell. (2023).
[39] Y. Xue, X. Han, Z. Wang, Self-adaptive weight based on dual-attention for differentiable neural architecture search, IEEe Trans. Industr. Inform. (2024).
[40] P. Gunasekhar, S. Vijayalakshmi, Optimal biomarker selection using adaptive social ski-driver optimization for liver cancer detection, Biocybern. Biomed. Eng. 40 (4) (2020) 1611–1625.
[41] A. Tharwat, T. Gabel, Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm, Neural Comput. Appl. 32 (2020) 6925–6938.
[42] M.A. Khanesar, M. Teshnehlab, M.A. Shoorehdeli, A novel binary particle swarm optimization, in: 2007 Mediterranean Conference on Control and Automation, IEEE, 2007, pp. 1–6.
[43] S. Ahmed, K.K. Ghosh, S. Mirjalili, R. Sarkar, AIEOU: automata-based improved equilibrium optimizer with U-shaped transfer function for feature selection, Knowl. Based. Syst. 228 (2021) 107283.
[44] A. Tuson, P. Ross, Adapting operator settings in genetic algorithms, Evol. Comput. 6 (2) (1998) 161–184 [Online]. Available.
[45] Z. Zhang, M. Wang, Convolutional neural network with convolutional block attention module for finger vein recognition, arXiv preprint (2022) arXiv: 2202.06673.
[46] Y. Hu, Y. Zhang, D. Gong, Multiobjective particle swarm optimization for feature selection with fuzzy cost, IEEe Trans. Cybern. 51 (2) (2020) 874–888.
[47] Y. Xue, X. Cai, F. Neri, A multiobjective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification, Appl. Soft. Comput. 127 (2022) 109420.
[48] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[49] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[50] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, Knowl. Based. Syst. 191 (2020) 105190.
[51] S. Mirjalili, S.M. Mirjalili, X.-S. Yang, Binary bat algorithm, Neural Comput. Appl. 25 (2014) 663–681.
[52] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[53] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl. Based. Syst. 96 (2016) 120–133.
[54] E.K. Burke, Y. Bykov, The late acceptance hill-climbing heuristic, Eur. J. Oper. Res. 258 (1) (2017) 70–78.
[55] M.A. Al-Betar, I. Aljarah, M.A. Awadallah, H. Faris, S. Mirjalili, Adaptive β-hill climbing for optimization, Soft. comput. 23 (24) (2019) 13489–13512.
[56] J. Carrasco, S. García, M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review, Swarm. Evol. Comput. 54 (2020) 100665.