

ORIGINAL RESEARCH ARTICLE

Particle swarm optimization design of convolutional neural networks with a fuzzy learning rate estimation for diabetic retinopathy detection and classification

Daniela Sánchez^{id}, Patricia Melin^{id}, Rodrigo Cordero-Martínez^{id}, and Oscar Castillo*^{id}

Division of Graduate Studies and Research, Tijuana Institute of Technology, National Technological Institute of Mexico, Tijuana, Baja California, Mexico

Abstract

One of the leading causes of blindness in young adults is diabetic retinopathy, a chronic eye complication that arises from diabetes mellitus. It occurs due to progressive damage caused by persistently high blood glucose levels. Convolutional neural networks are becoming a fundamental tool for analyzing and classifying medical images, making them a valuable aid in medical diagnosis. This work proposes a particle swarm optimization-based design of convolutional neural networks. During training, fuzzy logic estimates the learning rate based on the accuracy and loss of both the training and validation sets. The Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection dataset was used to test the proposed method. In the binary case, the best accuracy was 96.73%; in the multiclass case, it was 77.49%. The results demonstrate that the proposed method, when fuzzy logic is applied, improves diagnostic accuracy for diabetic retinopathy compared with a design that excludes fuzzy logic from the learning process and applies the adaptive moment estimation algorithm.

Keywords: Particle swarm optimization; Convolutional neural networks; Swarm intelligence; Learning rate; Type-1 fuzzy logic; Adam algorithm; Diabetic retinopathy

***Corresponding author:**
Oscar Castillo
(ocastillo@tectijuana.mx)

Citation: Sánchez D, Melin P, Cordero-Martínez R, Castillo O. Particle swarm optimization design of convolutional neural networks with a fuzzy learning rate estimation for diabetic retinopathy detection and classification. *Artif Intell Health*. 2026;3(4):026080019. doi: 10.36922/AIH026080019

Received: February 18, 2026

Revised: March 31, 2026

Accepted: April 2, 2026

Published online: April 24, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Diabetic retinopathy (DR) is one of the most frequent causes of vision loss worldwide and is a retinal condition caused by diabetes. Preserving vision requires the early diagnosis of DR. The two main types of DR are non-proliferative and proliferative, depending on whether neovascularization is present. Early identification is crucial for preventing retinal damage, as most people with diabetes do not exhibit ophthalmologic symptoms. A variety of applications using deep learning techniques have been developed to support specialists in diagnosis and improve the quality of life.^{1,2}

Models based on artificial neural networks have become a powerful tool for solving complex problems across many fields, such as human recognition, image classification, medical diagnosis, and forecasting.³⁻⁶ These applications leverage the learning, memory, fault tolerance, and distributed processing capabilities of artificial neural networks,

allowing them to generate outcomes based on data provided during the learning phase and their generalization ability.⁷⁻¹⁰ Several publications have conducted in-depth analyses of existing literature on neural networks, as well as innovative proposals that address specific problems in real-world environments and demonstrate the benefits of these techniques.¹¹⁻¹⁴

Another important intelligent technique is fuzzy logic (FL). This technique has been the subject of many practical applications and theoretical studies. Over the years, several extensions of FL have been developed to better model environments with uncertainty.^{15,16} These studies demonstrate that FL is a powerful tool for handling imprecise or incomplete information, enabling its use both individually and in conjunction with other intelligent techniques.^{17,18} The combination of convolutional neural networks (CNNs) and FL has enabled the development of methods that improve CNN performance.^{19,20}

In recent research, artificial neural networks have demonstrated significant progress in solving problems related to image detection, segmentation, and classification, particularly CNNs,²¹⁻²⁴ whose performance depends heavily on the selected parameters. One advantage of CNNs is their built-in feature extraction and dimensionality reduction performed by convolutional and pooling layers. However, manually searching for an optimal architecture is time-consuming. Consequently, CNN architecture design remains a broad field of study, with the main goal of generating a neural network architecture suited to the problem being addressed.²⁵⁻²⁷

Among the appropriate techniques for finding such architectures are particle swarm optimization (PSO), bat optimization, and the grey wolf optimizer, among others, which have demonstrated advantages for optimal CNN architecture search.²⁸⁻³¹ The combination of techniques such as FL and optimization methods has been shown to improve existing results in the literature, demonstrating an effective approach for creating robust intelligent hybrid systems.³²⁻³⁶ Furthermore, combining CNNs, FL, and optimization methods has led to performance improvements across diverse application domains.³⁷⁻⁴⁰

In this work, we propose designing CNN architectures and parameters using PSO, along with a type-1 fuzzy system (T1FS) to estimate the learning rate, and applying this framework to the detection and classification of DR. The selection of these techniques is based on evidence demonstrating the effectiveness of intelligent hybrid systems, which improve the performance of individual techniques when combined.

The main contribution of this work is the generation of

optimal CNN architectures using a PSO-based optimization scheme. Furthermore, during the training phase, an adaptive mechanism based on a T1FS dynamically adjusts the learning rate of the adaptive moment estimation (Adam) optimizer based on accuracy and loss values. In each training epoch, a new fuzzy inference system is generated, allowing continuous adaptation to training dynamics. As neural network behavior changes over time, a fixed fuzzy system cannot capture these variations; in contrast, generating a new fuzzy system based on current behavior enables more effective adaptation. The proposed method seeks to avoid overfitting and ensure stable convergence, thereby improving model generalization. The main objective is to improve accuracy in both the detection and multiclass classification tasks.

This paper is structured as follows: Section 2 presents related works. Section 3 describes the basic concepts required to understand the proposed method. Section 4 presents the proposed methodology and dataset. Section 5 reports the experimental results, and Section 6 presents the conclusions.

2. Related work

Convolutional neural networks have proven their efficiency for classification tasks, particularly in medical diagnosis. Luo *et al.*⁴¹ proposed a deep learning model to improve DR detection. Experiments conducted on two existing datasets (Methods to Evaluate Segmentation and Indexing in Digital Ophthalmology and Eye Picture Archive Communication System) achieved greater accuracy than existing models. Sorgente *et al.*⁴² proposed a method for identifying and localizing retinal abnormalities, including DR, using deep learning. They compared their model against different fine-tuned models, and analysis of the results showed that the proposed model outperforms the others, highlighting its ability to detect pathological characteristics. El-Hag *et al.*⁴³ proposed a two-part method for retinal image classification. The first part applies fuzzy preprocessing to improve contrast between objects and the background. Image binarization is performed for segmentation to remove blood vessels and the optic disc, followed by gradient processing to distinguish between normal and abnormal cases. In the second part, a CNN is used to classify images into two classes by automatically detecting maculopathy. Kansal *et al.*⁴⁴ proposed four deep learning models and compared them with state-of-the-art models, including the EfficientNet-B0. The results showed that the bidirectional long short-term memory optimized with PSO demonstrated strong generalization capability.

Elhani *et al.*⁴⁵ proposed a PSO variant to reduce the number of iterations required to determine optimal CNN

parameters. The algorithm was evaluated against several recent algorithms in the literature using nine classification datasets. Compared with 27 algorithms, the study showed that its variant can rapidly identify effective convolutional network architectures that deliver performance comparable to the best existing designs, indicating significant potential. Melin *et al.*⁴⁶ proposed a PSO-based approach to design CNN architectures for DR detection and classification. The optimization includes main CNN parameters as well as additional parameters such as batch size, epochs, and training algorithm. The PSO selects among six training algorithms during training. The best result for the binary case was achieved with the Nesterov accelerated adaptive moment estimation (Nadam) algorithm, while the best result for the multiclass case was achieved using the Adam algorithm. Several studies have proposed improvements to the Adam optimizer to enhance convergence and generalization performance on complex datasets. Ibrahim *et al.*⁴⁷ incorporated a projection-based gradient correction mechanism inspired by quasi-Newton and conjugate gradient methods and evaluated the proposed methods on the Modified National Institute of Standards and Technology dataset (MNIST), Canadian Institute for Advanced Research 10-class dataset, and Canadian Institute for Advanced Research 100-class dataset, demonstrating improvements over the standard Adam optimizer. Cheng *et al.*⁴⁸ proposed a modification of ADAM that uses the Frobenius norm to measure the model's complexity and adjusts the learning rate accordingly. The proposed method was tested on datasets such as MNIST and Fashion-MNIST, showing improved efficiency and stability. Wu *et al.*⁴⁹ proposed a learning rate scheduler that automatically adjusts the learning rate according to the stages of neural network training. The method is based on a unified formulation that considers the evolution of the loss. The proposed method was tested on the ImageNet dataset, showing improvements in performance and robustness on complex tasks. Recently, the combination of neural networks, PSO, and fuzzy logic has also been explored. Feng *et al.*⁵⁰ proposed a deep learning framework, a three-dimensional fully convolutional neural network-hybrid tuned swarm optimization-PSO, for brain tumor detection and classification. An interval type-2 fuzzy weighting system was applied to generate the initial convolution filter weights, combined with a hybrid optimization algorithm that integrates the exploratory strengths of transit search optimization and the fast convergence of PSO. The proposed models significantly reduced the number of trainable parameters while enhancing accuracy and training efficiency. Sobhanifard⁵¹ proposed a hybrid model combining Bayesian networks, fuzzy logic, and PSO to optimize neural network weights and structure, thereby

reducing errors and improving performance. The model is applied to the analysis and prediction of economic growth using time series on gross domestic product. The method is evaluated and compared with other time series techniques, achieving higher precision and correlation.

The studies conducted so far have enabled the development of new retinal image processing methods for detecting key information, which is then primarily used in deep learning models. Prior work has also focused on the use and modification of existing models reported in the literature, and on the creation of new deep learning models using conventional metaheuristics or modifications to accelerate convergence, with the aim of solving problems related to the retina and supporting medical diagnosis of conditions affecting it. Modifications to the ADAM optimizer have recently been proposed to improve its convergence and convergence speed. The combination of neural networks, fuzzy logic, and PSO has also been addressed, where metaheuristics are responsible for finding the optimal hyperparameters of the neural networks. Fuzzy logic determines the neural network's weights. The analyzed works highlight the importance of the learning rate during the training phase and motivate the pursuit of learning rate optimization strategies.

3. Basic concepts

The intelligence techniques analyzed and utilized in this work are presented in this section.

3.1. Convolutional neural networks

Artificial neural networks function similarly to the nervous system in the human brain, emulating the learning and information processing. They serve as digital representations of biological neurons and the synapses that connect them, enabling highly accurate predictions.^{4,5} The ability of a CNN to extract features is highlighted in image processing. In the feature extraction phase, convolutional and pooling layers are particularly crucial.⁵² A CNN consists of one or more convolutional layers (CLs) that apply a predetermined number of filters to the input image. The first task of these layers is to identify basic features such as lines and edges by convolving the input data with an $m \times m$ filter. As the filter passes through the neurons, a new matrix is generated and passed to subsequent layers. The rectified linear unit function is the most commonly used activation function and is typically applied after CL.^{5,53,54} The pooling layer also employs a filter of size $n \times n$, but its purpose is to reduce the information of a specific region. Common operations performed by this layer include computing the maximum, average, or minimum values within a region. This process allows the information produced by the CL to be grouped and reduces the amount of data passed to the

fully connected layers. Finally, the output layer produces a probability distribution over a predefined set of classes.^{55,56}

3.2. Fuzzy logic

Zadeh⁵⁷ introduced the theory of type-1 fuzzy sets. Unlike traditional binary logic, which operates on discrete values, FL uses degrees of truth that can take any value within the continuous range (0, 1). This logic is inspired by the way humans manage uncertainty, enabling the representation and processing of imprecise information, and has found applications in medical, robotic, financial, and automatic control systems.⁵⁸ FL is characterized by its ability to mimic human reasoning by using approximate rather than exact approaches. A fuzzy inference system comprises three fundamental elements: a rule base, a database (or dictionary), and a reasoning mechanism that performs the inference process.⁵⁹ **Equation 1** is used to determine the membership value.

$$A = \{(x, \mu_A(x)) | \forall x \in X\} \quad (1)$$

where the type-1 membership function (MF) is represented by $\mu_A(x)$, $x \in X$ in a universe of discourse X . The main type-1 membership functions are shown in [Figure 1](#).

3.3. Particle swarm optimization

Particle swarm optimization is inspired by the collective behavior of flocks of birds or schools of fish, particularly

their social behavior and movement patterns.⁶⁰ In PSO, candidate solutions are referred to as particles that move through a multidimensional search space, forming a swarm.^{61,62} This algorithm uses two vectors: the current position and the velocity of each particle, represented by x and v , respectively. The next position of particle i is calculated with **Equation 2**.⁶³

$$x_{id}(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where $x_{id}(t)$ represents the current position of particle i in dimension d at time t . To calculate the next position, the velocity $v_i(t+1)$ is used. Eberhart and Shi⁶⁴ proposed an improvement to the exploration and exploitation processes by introducing a new parameter: inertia weight. This inertia weight is reduced during execution, allowing global exploration in the initial stages and promoting local exploitation as the algorithm progresses. Several proposals in the literature exist for decrementing the inertia weight. **Equation 3** presents one of the main: the linear decrement.

$$w_t = (w_s - w_e) \times \frac{(t_{max} - t)}{t_{max}} + w_e \quad (3)$$

where the number of maximum iterations is represented by t_{max} , the initial inertial weight is represented by w_s , and w_e is the expected final value for w .⁶⁵ **Equation 4** allows the determination of the particle's velocity, including the

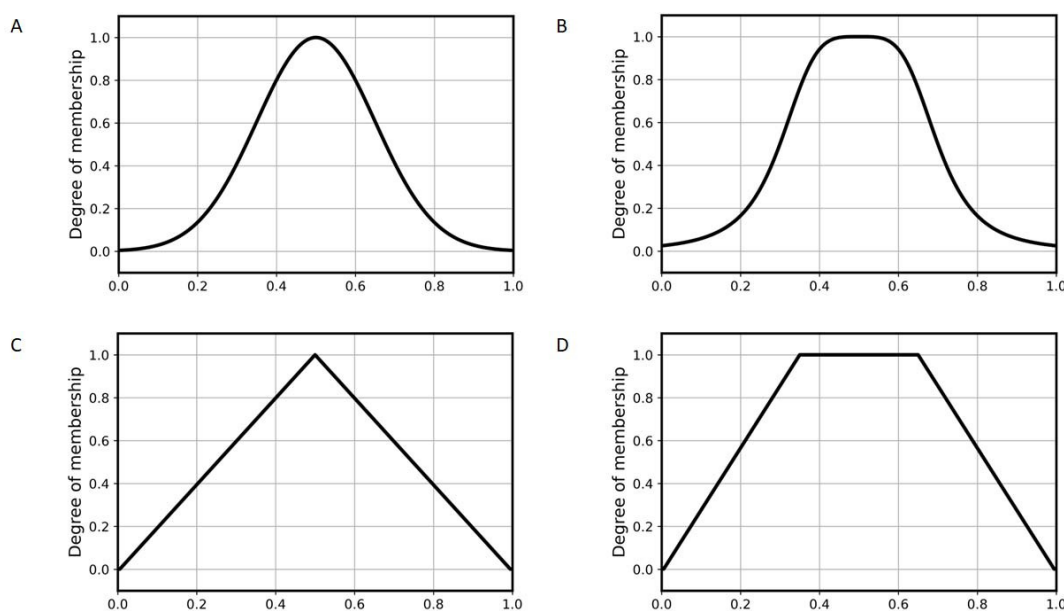


Figure 1. Type-1 (A) Gaussian, (B) generalized bell, (C) triangular, and (D) trapezoidal membership function

inertial weight.

$$v_{id}(t+1) = wv_{id}(t) + c_1r_{1d}(t)*[y_{id}(t) - x_{id}(t)] + c_2r_{2d}(t)*[\hat{y}_d(t) - x_{id}(t)] \quad (4)$$

where c_1 and c_2 are cognitive and social components related to the acceleration of the particle, r_1 and r_2 are random values between 0 and 1. The cognitive component represents the particle's experience, and the social component represents the swarm's collective information. The best position at time t of particle i in dimension d is represented by $y_{id}(t)$, while $\hat{y}_d(t)$ represents the best position in dimension d .

4. Proposed method

The proposed method seeks to find the optimal CNN architecture by considering parameters such as the number of CLs, the number of filters, and the filter size ($m \times m$). After each convolutional layer, a max pooling layer is implemented. The proposed method also determines the pool size ($n \times n$), the number of fully connected layers, and the number of neurons, epochs, and batch sizes for each layer. The rectified linear activation function and dropout are utilized. Figure 2 illustrates the proposed method, in which the PSO searches for an optimal CNN architecture for the DR application.

During the training phase, the Adam algorithm is used; at each training step, a T1FS is implemented to estimate the learning rate, which is usually set at the beginning of training and used by Adam throughout the process. Kingma and Ba⁶⁶ determined α as the learning rate that allows the Adam algorithm to adapt the weights. In this work, the estimation of this value is proposed using a T1FS that is adapted based on the current accuracy and loss values for the training and validation sets. That is, in each epoch, a new fuzzy system is generated, which establishes

the ranges of the fuzzy variables and the membership values based on the behavior of the precision and the loss. An example of a T1FS is shown in Figure 3, where the learning rate is determined based on the accuracy and loss values in each epoch. The T1FS is a Mamdani model with four inputs and one output. Each fuzzy variable has three membership functions, all of which are generalized bell (Gbell). These MFs were selected because they achieved better results in another previously developed application, where their performance was compared with that of trapezoidal membership functions.⁶⁷

The Gbell MFs are determined using Equation 5, where, for all Gbell MFs, the value of B is set to 1.5. The maximum and minimum values for accuracy, loss, and learning rate are determined using Equations 6–11, where values from the current epoch are used. The ranges for accuracy, loss (training and validation), and learning rate fuzzy variables are established using Equations 12–17.

$$gbellmf(x, [A, B, C]) = \frac{1}{1 + \left[\left(\frac{x - C}{A} \right)^2 \right]^B} \quad (5)$$

$$Accuracy_{min} = \min(Accuracy, Val_Accuracy) \quad (6)$$

$$Accuracy_{max} = \max(Accuracy, Val_Accuracy) \quad (7)$$

$$Loss_{min} = \min(Loss, Val_Loss) \quad (8)$$

$$Loss_{max} = \max(Loss, Val_Loss) \quad (9)$$

$$LR_{min} = \min(LR, 0.00001) \quad (10)$$

$$LR_{max} = \max(LR, 0.00001) \quad (11)$$

$$R_Accuracy_{min} = Accuracy_{min} - 0.1(Accuracy_{max} - Accuracy_{min}) \quad (12)$$

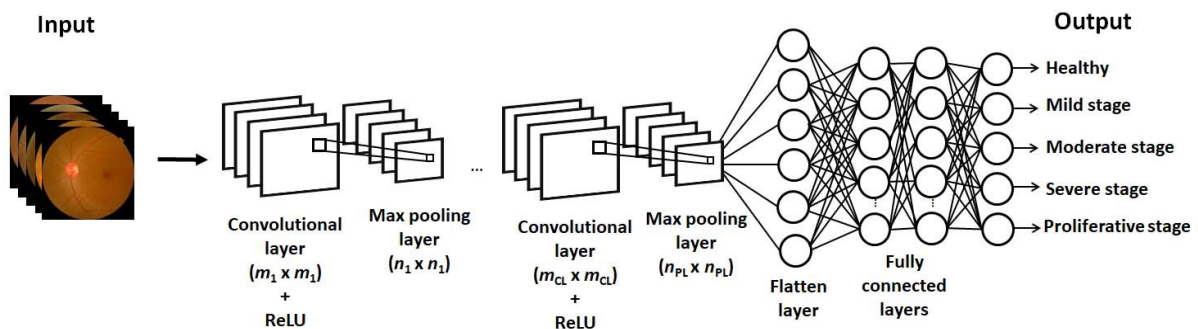


Figure 2. The proposed method for the convolutional neural network design
Abbreviation: ReLU: Rectified linear unit.

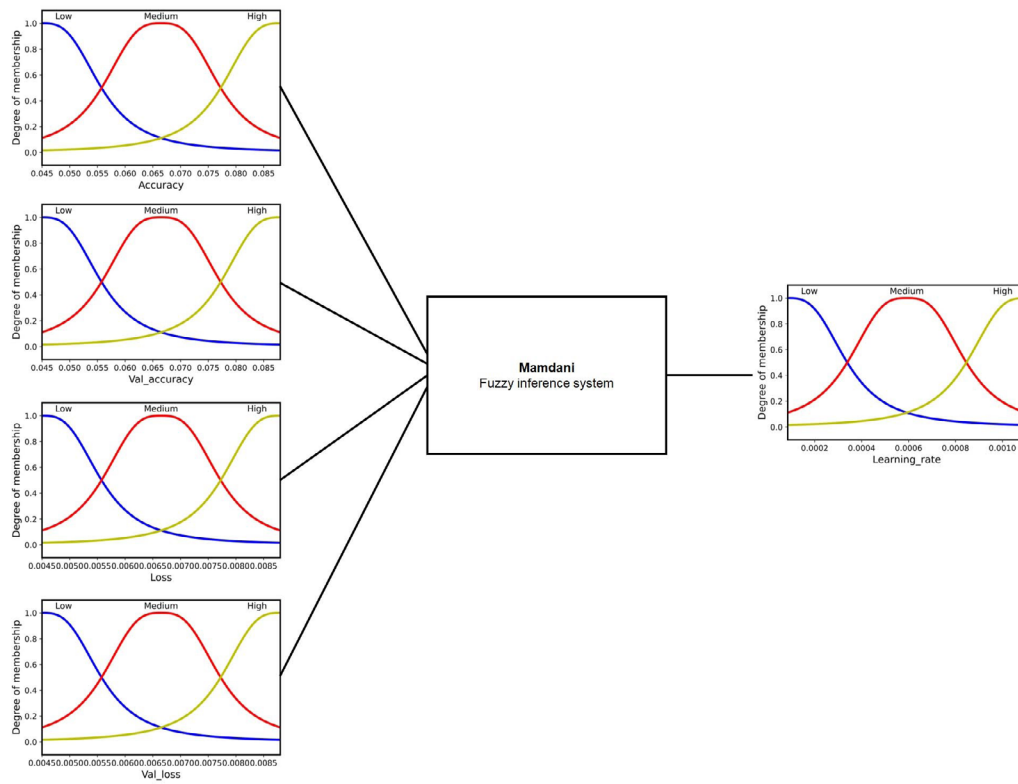


Figure 3. Type-1 fuzzy system for the learning rate estimation

$$R_Accuracy_{max} = Accuracy_{max} + 0.1(Accuracy_{max} - Accuracy_{min}) \quad (13)$$

$$R_Loss_{min} = Loss_{min} - 0.1(Loss_{max} - Loss_{min}) \quad (14)$$

$$R_Loss_{max} = Loss_{max} + 0.1(Loss_{max} - Loss_{min}) \quad (15)$$

$$R_LR_{min} = LR_{min} - 0.1(LR_{max} - LR_{min}) \quad (16)$$

$$R_LR_{max} = LR_{max} + 0.1(LR_{max} - LR_{min}) \quad (17)$$

An example of the fuzzy variable is shown in Figure 4. For the accuracy variables, the values of C_1 and C_3 are equal to $R_Accuracy_{min}$ and $R_Accuracy_{max}$, respectively. In the same way, the values for C_1 and C_3 for the loss and learning rate are determined.

To establish C_2 and $A_{1,2,3}$ for the accuracy, loss, and learning rate fuzzy variables, Equations 18–23 are applied, respectively. The fuzzy if-then rules defined for the T1FS are shown in Table 1, where MF_L denotes low, MF_M denotes medium, and MF_H denotes high. These rules were designed to improve the models' generalization and to prevent

overfitting during training. It is important to note that the T1FS could have up to 81 fuzzy if-then rules; however, based on the possible behaviors of accuracy and loss, only eight rules were selected to achieve better performance.

$$C_2 = (R_Accuracy_{min} + R_Accuracy_{max}) / 2 \quad (18)$$

$$A_{1,2,3} = (C_2 - R_Accuracy_{max}) / 2 \quad (19)$$

$$C_2 = (R_Loss_{min} + R_Loss_{max}) / 2 \quad (20)$$

$$A_{1,2,3} = (C_2 - R_Loss_{max}) / 2 \quad (21)$$

$$C_2 = (R_LR_{min} + R_LR_{max}) / 2 \quad (22)$$

$$A_{1,2,3} = (C_2 - R_LR_{max}) / 2 \quad (23)$$

4.1. Optimization description

The PSO seeks to reduce the accuracy error. Each particle is evaluated with the fitness function determined by Equation 24, and the accuracy percentage is determined by Equation 25. Other formulas were also used to measure

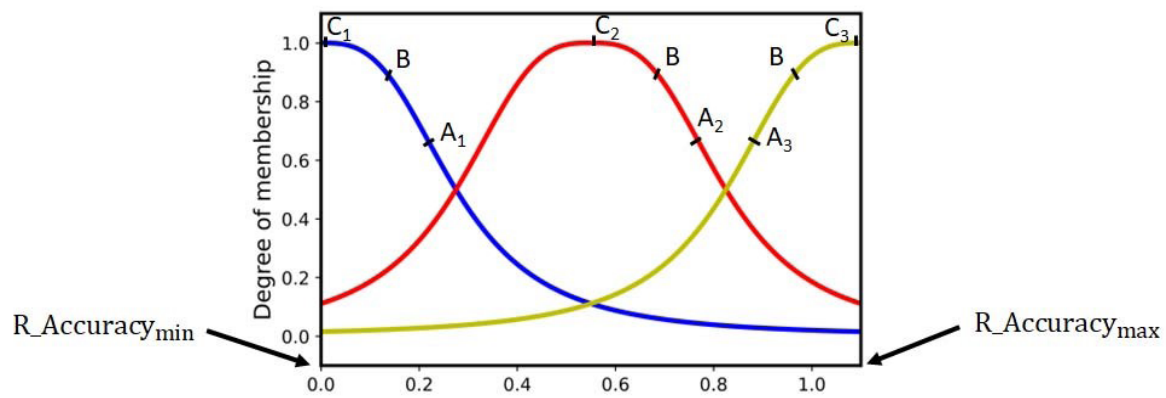


Figure 4. Example of the accuracy of the fuzzy variable of the type-1 fuzzy system

Table 1. Fuzzy if-then rules for the type-1 fuzzy system

Rule	Inputs				Learning rate
	Accuracy	Validation accuracy	Loss	Validation loss	
1	MF_L	MF_L	MF_H	MF_M	MF_H
2	MF_L	MF_M	MF_L	MF_M	MF_H
3	MF_M	MF_L	MF_M	MF_H	MF_L
4	MF_M	MF_H	MF_M	MF_L	MF_L
5	MF_M	MF_M	MF_M	MF_M	MF_M
6	MF_H	MF_H	MF_L	MF_L	MF_M
7	MF_H	MF_L	MF_L	MF_H	MF_L
8	MF_H	MF_M	MF_L	MF_M	MF_M

Note: MF_L denotes low, MF_M denotes medium, and MF_H denotes high.

the performance of the proposed method (Equations 26–29).

$$f = 1 - \frac{TP + TN}{TP + FP + TN + FN} \quad (24)$$

$$\%Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100 \quad (25)$$

$$recall(sensitivity) = \frac{TP}{TP + FN} \quad (26)$$

$$precision = \frac{TPN}{TP + FP} \quad (27)$$

$$F1-score = \frac{2 * precision * recall}{precision + recall} \quad (28)$$

$$specificity = \frac{TN}{TN + FP} \quad (29)$$

where false positive is FP, false negative is FN, true negative is TN, and true positive is TP. Table 2 shows the PSO parameters. These values are based on previous work in which the CNN design was also proposed.⁴⁶

Each particle dimension corresponds to a design parameter of a particular CNN. Each particle has a total of 16 dimensions, and an illustration of a particle is shown in

Figure 5. Table 3 presents the ranges that define the search space. The batch size can take six possible values (8, 16, 32, 64, 128, and 256). Figure 6 shows a flow diagram of the proposed method. In summary, PSO searches for the optimal CNN architecture, training each particle while applying T1FS at each epoch to estimate the learning rate.

4.2. Dataset description

The Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection Challenge (APTOS 2019) database⁶⁸ includes 5,590 samples under different conditions (size and noise), and only 3,662 have their classification labels.

Table 2. Particle swarm optimization parameters

Parameter	Particles	Maximum iterations	C_1	C_2	Inertia weight (initial)
Value	10	30	1.5	2	0.8

Table 3. Ranges of the search space

Parameters	Convolutional layers	Number of filters for the convolutional layers			Filter size	Pool size	Fully connected layers	Neurons for fully connected layers	Epochs	Batch size
		#1	#2	#3						
Min	1	8	16	32	3	3	1	10	10	1
Max	3	16	32	64	5	5	3	300	30	6

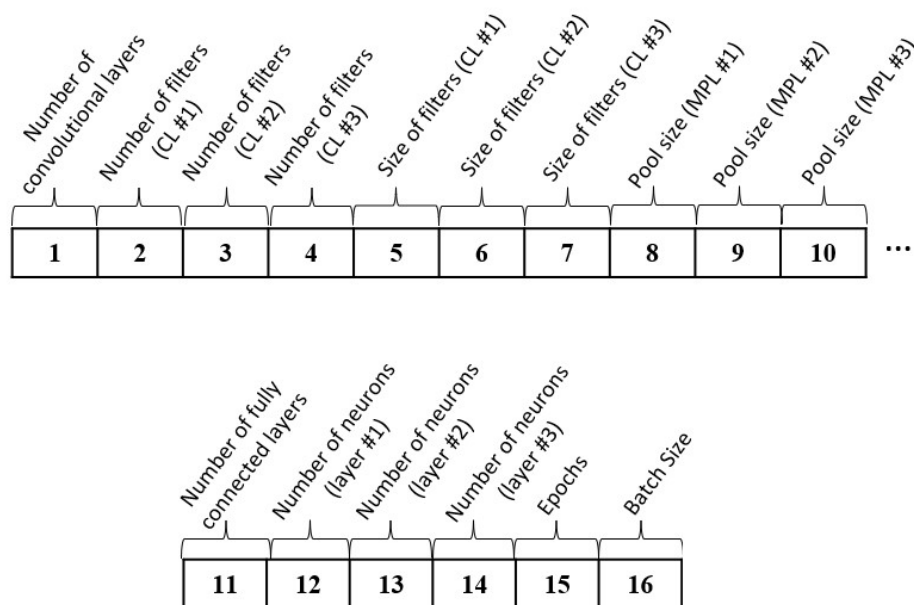


Figure 5. Representation of the particle for the convolutional neural network design
Abbreviations: CL: Convolutional layer; MPL: Max pooling layer.

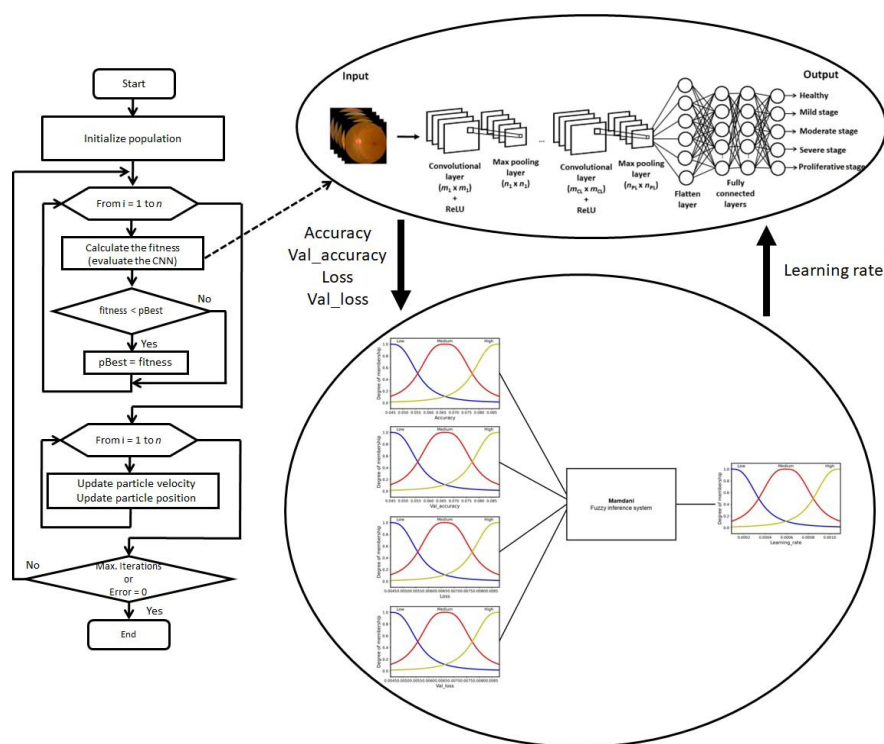


Figure 6. Flow diagram of the proposed convolutional neural network design
Abbreviation: CNN: Convolutional neural network.

The labeled images are used for training (65%), validation (15%), and testing phases (20%), using a size of 200×200 pixels. As preprocessing, each image initially has pixel values between 0 and 255. A binary mask is created to determine the central pixel of the retina and its size, along with its height and width. This process is used to remove noise or unnecessary information.⁶⁹ An example of the APTOS 2019 database is shown in Figure 7. In Table 4, the distribution among each class is defined. In this work, two cases are conducted: binary (to detect DR) and multiclass (to classify DR).

5. Experimental results

This section presents the results of the proposed CNN design: detection and classification. The results achieved are compared with those of other CNN designs trained by PSO without FL. For the results, 10 experiments were performed for each case (detection and classification). These experiments have an early stopping with a patience value of 4. To test the effectiveness of the proposed method, two validations were generated for each case (binary and multiclass). Sections 5.1 and 5.2 show the results of validation #1, while a summary of the results of validation #2 is shown in Section 5.3.

5.1. Detection results

In these experiments, the CNN determines whether a retina is healthy, addressing a binary classification problem. The best architecture for detection (binary case) is shown in Table 5, where the CNN achieved 96.73% accuracy in experiment #6. As shown, the architecture consists of three CLs and two fully connected layers. This model was trained with 20 epochs.

The behavior of accuracy and loss for both sets (training and validation) for the binary case is shown in Figure 8. The resulting model exhibits rapid convergence in the early epochs, with training and validation accuracy and losses evolving in a very similar fashion. The gap between training and validation is minimal (less than 0.5%), indicating good generalizability and no overfitting. Using up to 20 epochs allowed the model to train correctly and learn efficiently.

The learning rate estimated by the TIFS in this experiment is shown in Figure 9, where it can be seen that, from epoch 12 onwards, the learning rate changed very slightly, allowing for better training.

The best result for the multiclass case (experiment #6) is shown in Figure 10, where it can be observed that the experiment shows good convergence, with a stagnation

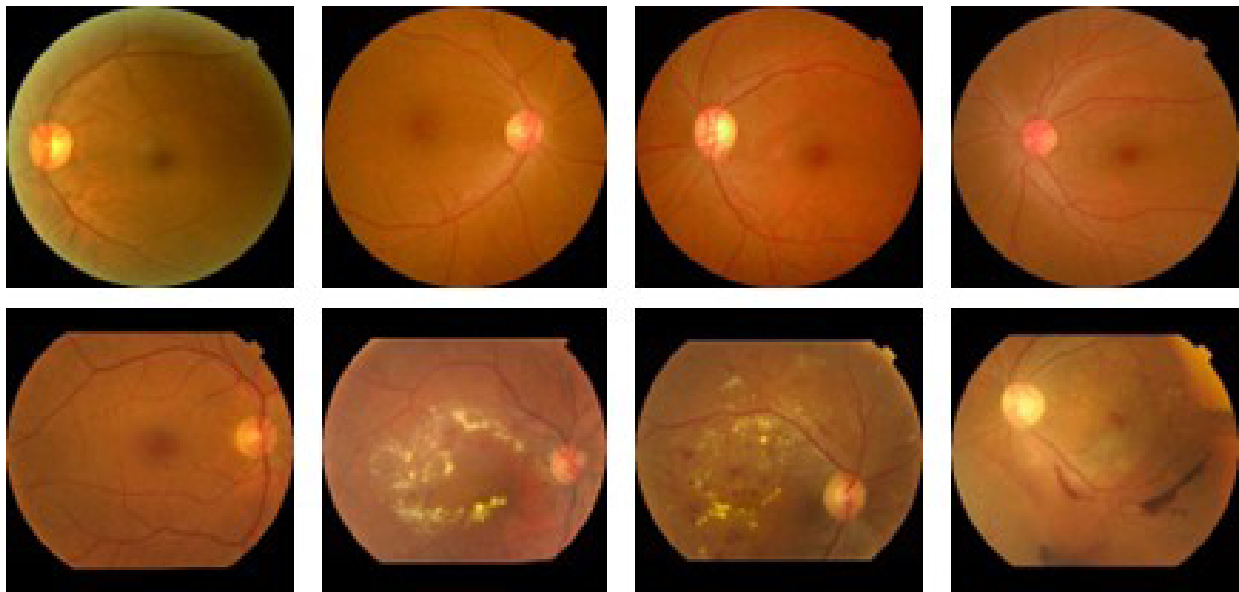


Figure 7. Example of the Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection Challenge database

Table 4. Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection Challenge database description

Class	Healthy retina	Retina with damage			
		Mild stage	Moderate stage	Severe stage	Proliferative stage
Number of images	1,805	370	999	193	295

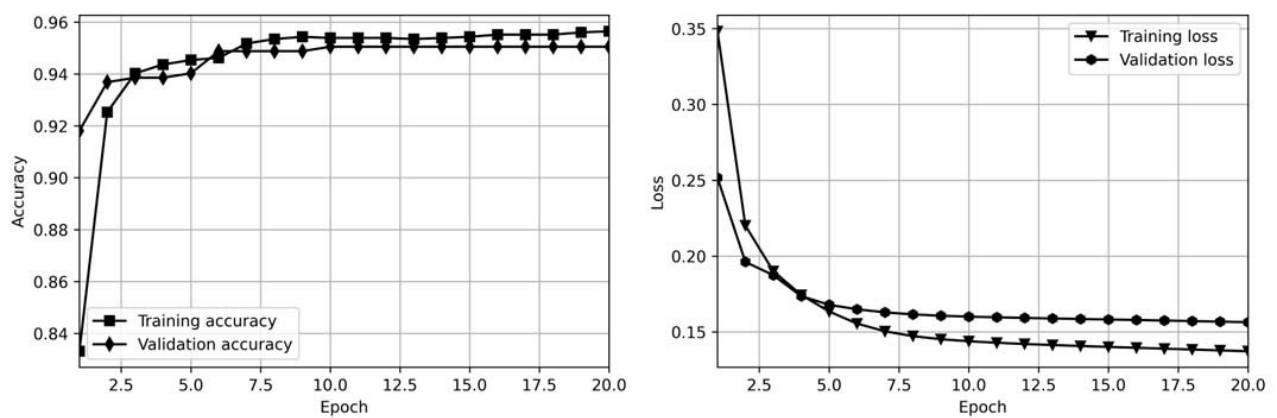


Figure 8. Accuracy and loss behavior of the best architecture (binary case)

between iterations 6 and 15. The summary of results in the binary case is shown in Table 6, which presents the accuracy percentage.

Table 5. The best architecture (binary case)

Architecture details	Performance metrics
Convolutional layers	3
Filters/size (Convolutional layers)	16, 5 × 5 25, 4 × 4 42, 5 × 5
Pool size	4 × 4 5 × 5 3 × 3
Fully connected layers (neurons)	2 (56, 157)
Epoch	20
Batch size	8
Recognition rate (%)	96.73
Error	0.0327

Table 6. Results of the binary case

Best (%)	Average (%)	Worst (%)
96.73	96.29	96.04

5.2. Classification results

In these experiments, the CNN must determine whether a retina is healthy or exhibits a specific type of damage; thus, it addresses a multiclass classification problem. The best architecture for this multiclass case is shown in Table 7, where the CNN achieved an error of 0.2251. This result was obtained in experiment #10. As shown, the architecture consists of two CLs and two fully connected layers. This model was trained for 29 epochs.

The behavior of accuracy and loss for both sets (training and validation) for the multiclass case is shown in Figure 11. The model exhibits adequate initial convergence in the early stages, with training and validation accuracy showing similar behavior. However, from epoch 5 onwards, the training accuracy continues to grow to approximately 0.77, while the validation accuracy stagnates at around 0.75.

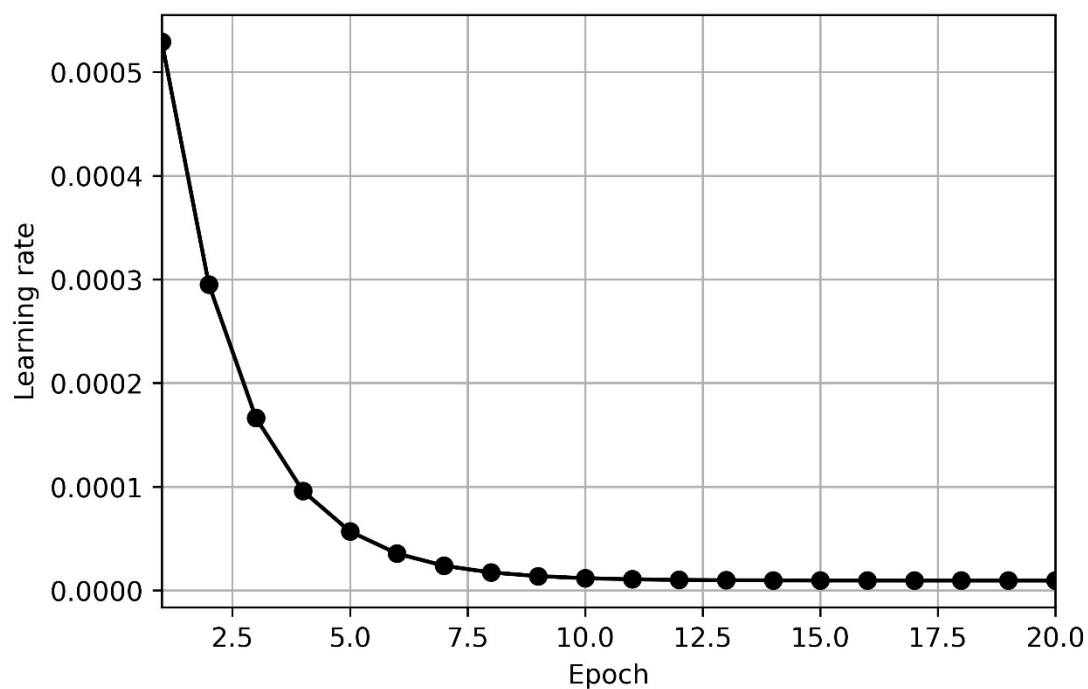


Figure 9. Learning rate estimation for the best architecture (binary case)

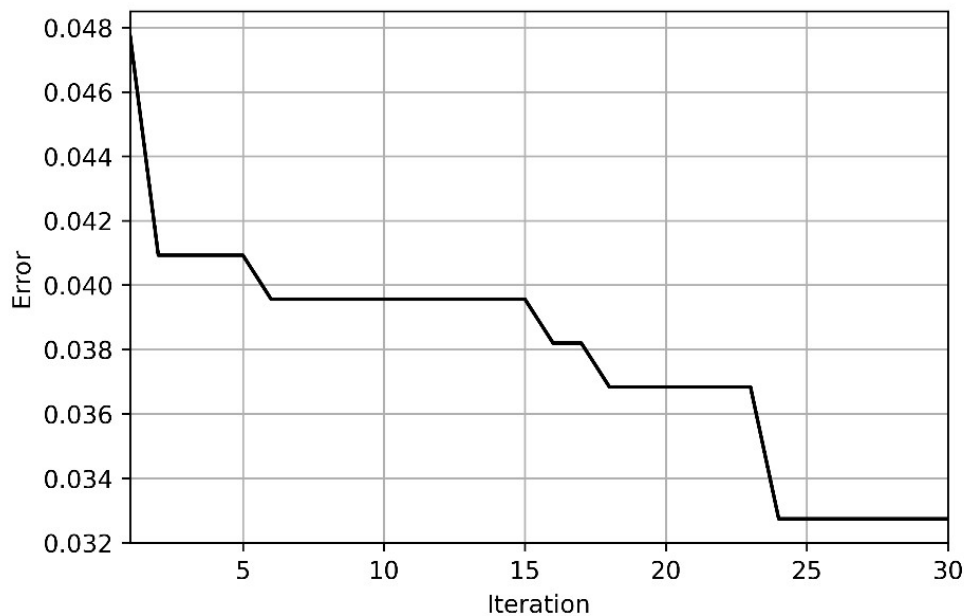


Figure 10. Convergence of experiment #10 (binary case)

This difference results in a separation of approximately 2%, indicating slight overfitting. This behavior is reinforced by the loss values, which differ by approximately 0.10 between the training and validation sets, indicating slight to moderate overfitting. Furthermore, this problem may be associated not only with overfitting but also with class imbalance in the dataset, leading the model to learn in a biased manner towards the majority classes and limiting its generalization capacity.

Table 7. The best architecture (multiclass case)

Architecture details	Performance metrics
Convolutional layers	2
Filters/size (convolutional layers)	15, 4×4 23, 3×3
Pool size	4×4 3×3
Fully connected layers (neurons)	2 (181, 42)
Epoch	29
Batch size	16
Recognition rate (%)	77.49
Error	0.2251

The learning rate estimated by the T1FS in experiment #10 is shown in Figure 12. It can be observed that the learning rate decreased significantly in the first five epochs, as in the binary case, and then decreased gradually thereafter. Finally, in the last epochs, the learning rate changed minimally.

The best result for the multiclass case (experiment #10) is shown in Figure 13. The experiment shows good convergence, with a stagnation between iterations 10 and 19. The summary of results in the multiclass case is shown in Table 8, which presents the accuracy percentage.

Table 8. Best, average, and worst accuracies for the multiclass case

Metric	Best (%)	Average (%)	Worst (%)
Accuracy	77.49	77.05	76.67

5.3. Comparison of results

Table 9 summarizes the results for validation #1, comparing the proposed PSO with T1FS for learning rate estimation, and a PSO that optimizes the CNN using Adam without FL. The results show that the proposed method

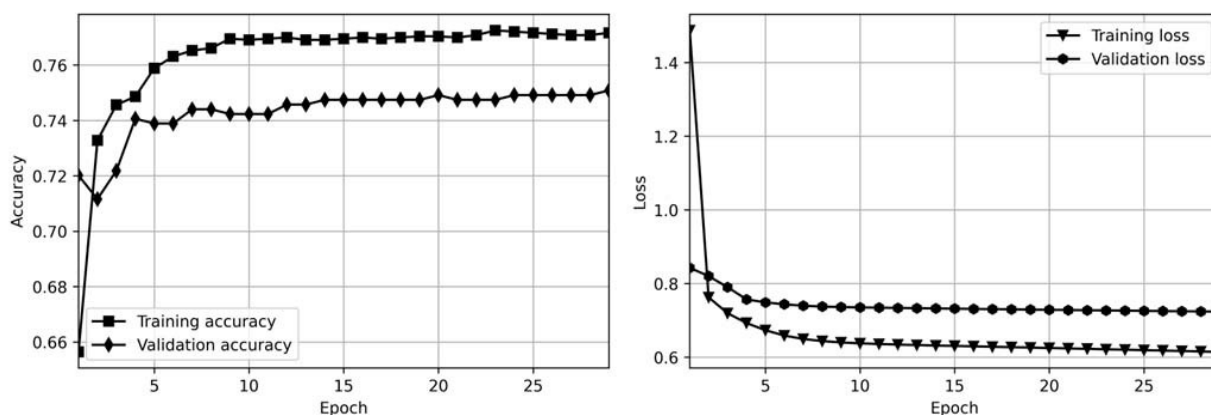


Figure 11. Accuracy and loss behavior of the best architecture (multiclass case)

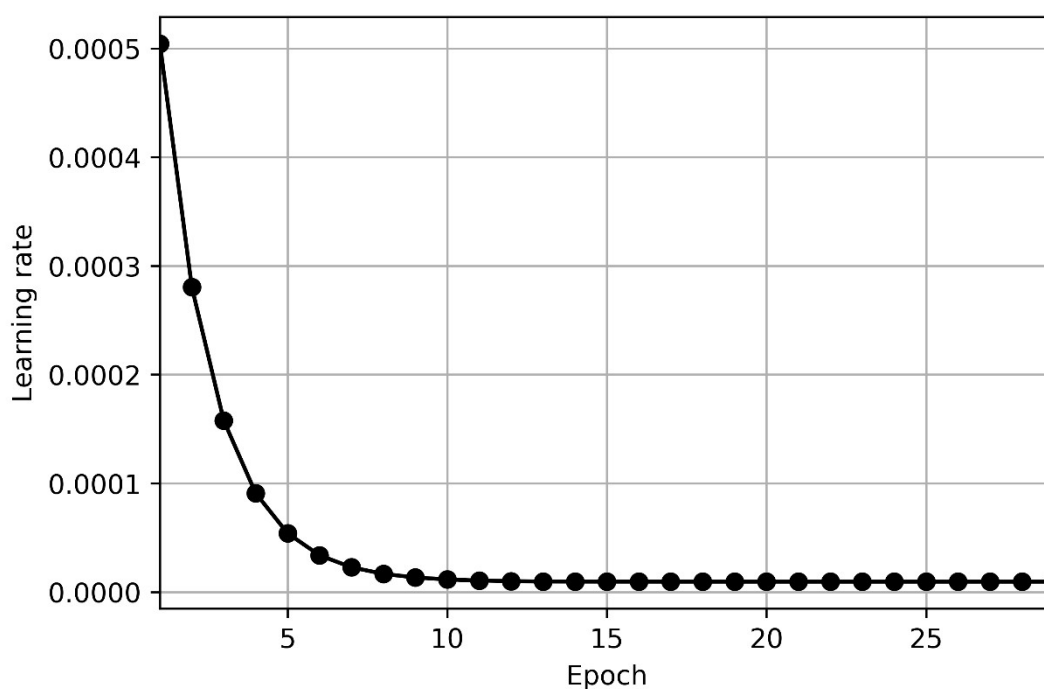


Figure 12. Learning rate estimation of the best architecture (multiclass case)

achieves higher average accuracy in both cases (binary and multiclass), and the worst-case value in each case is improved using the proposed method. The results reported by Melin *et al.*⁴⁶ were obtained using labeled images for training (72%), validation (8%), and testing (20%).

Table 10 shows the average of the different metrics obtained.⁷⁰ It also reports the average floating-point operations (FLOPs), which represent the number of mathematical operations, along with the number of parameters, to provide a more detailed analysis of the

resulting models. The results indicate that the proposed method performs better across the presented metrics. The FLOPs of the proposed method are higher, indicating that it performs more operations while using fewer parameters on average. This suggests that, for the binary case in validation #1, the proposed method can produce simpler models.

Table 11 shows that, as in the binary case, the proposed method achieves better metric values. It is important to note that, although the metrics are higher, sensitivity

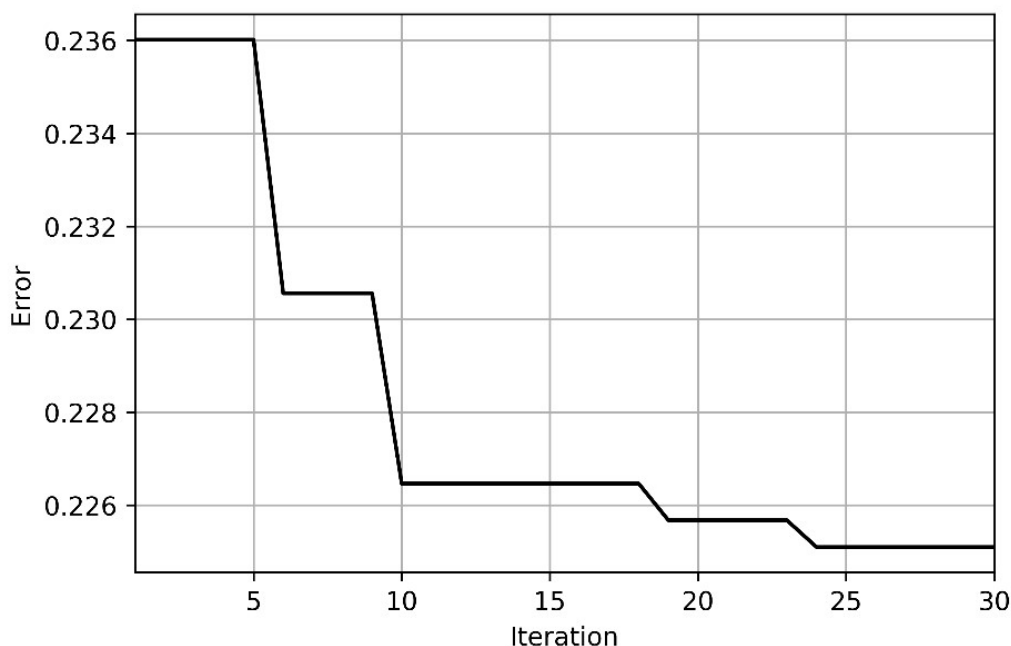


Figure 13. Convergence of experiment #10 (multiclass case)

Table 9. Summary of results (Validation #1)

Method	Case					
	Binary			Multiclass		
	Best (%)	Average (%)	Worst (%)	Best (%)	Average (%)	Worst (%)
PSO ⁴⁶	96.59	95.33	94.68	77.35	76.03	74.90
PSO (Adam)	96.18	96.02	95.77	76.81	76.13	75.72
PSO (Adam + T1FS)	96.73	96.29	96.04	77.49	77.05	76.67

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

remains low in both methods due to the imbalance in the number of images across classes with some degree of retinopathy, particularly in mild, severe, and proliferative stages. In terms of complexity and parameter count, the proposed method produces architectures with, on average, more operations and parameters. Consequently, the average training time increases due to more complex architectures.

The average accuracy achieved by the proposed method

and by the PSO using ADAM without FL as the training algorithm for validation #1 is shown in Figures 14 and 15, respectively. The proposed method converges faster in both the binary and multiclass cases.

The results shown in Table 12 indicate that, as in validation #1, the proposed method achieves higher average scores across the metrics, with only a slight increase in time for validation #2 (binary case). Importantly, for validation #2, the number of operations increased for models trained

without fuzzy logic, but the resulting architectures used fewer parameters on average.

Table 10. Metrics and complexity average results (validation #1, binary)

Metric	PSO (Adam)	PSO (Adam + T1FS)
Accuracy	0.960164	0.962892
ROC-AUC	0.984781	0.985517
F1-Score	0.960162	0.962893
Sensitivity	0.94543	0.952957
Specificity	0.97313	0.975346
Training time	80 s	55s
FLOPs	31.81 M	43.5 M
Parameters	9.24 M	4.43 M

Abbreviations: Adam: Adaptive moment estimation; AUC: Area under the curve; FLOPs: Floating-point operations; PSO: Particle swarm optimization; ROC: Receiver operating characteristic; T1FS: Type-1 fuzzy system.

Table 11. Metrics and complexity average results (validation #1, multiclass)

Metric	PSO (Adam)	PSO (Adam + T1FS)
Accuracy	0.761255	0.770532
ROC-AUC	0.874627	0.88006
F1-Score	0.733582	0.734886
Sensitivity	0.496499	0.508479
Specificity	0.933518	0.934646
Training time	25 s	38 s
FLOPs	32.29 M	32.40 M
Parameters	6.17 M	7.05 M

Abbreviations: Adam: Adaptive moment estimation; AUC: Area under the curve; FLOPs: Floating-point operations; PSO: Particle swarm optimization; ROC: Receiver operating characteristic; T1FS: Type-1 fuzzy system.

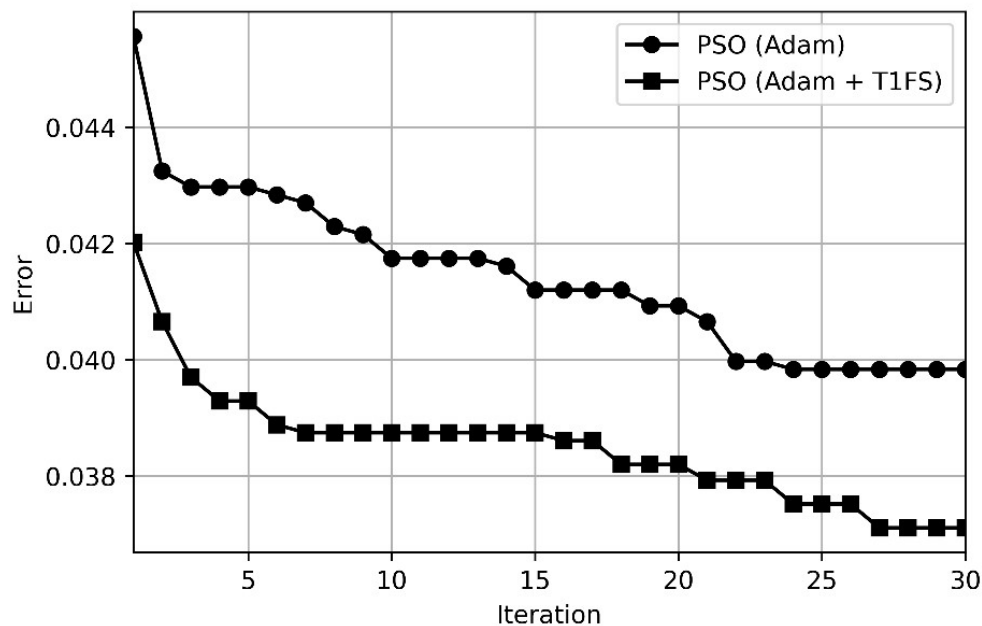


Figure 14. Average convergence for the binary case

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

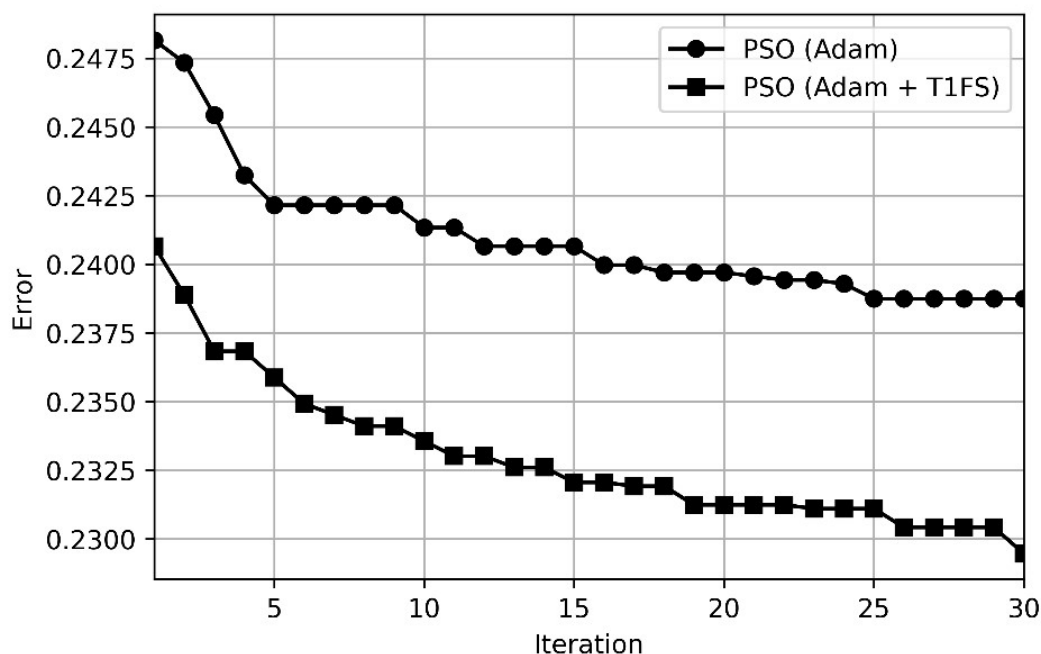


Figure 15. Average convergence for the multiclass case

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

Table 12. Metrics and complexity average results (validation #2, binary)

Metric	PSO (Adam)	PSO (Adam + T1FS)
Accuracy	0.95239	0.95539
ROC-AUC	0.98236	0.98237
F1-Score	0.95239	0.95539
Sensitivity	0.95000	0.95457
Specificity	0.95485	0.95623
Training time	45 s	50 s
FLOPs	66.33 M	39.23 M
Parameters	3.85 M	7.39 M

Abbreviations: Adam: Adaptive moment estimation; AUC: Area under the curve; FLOPs: Floating-point operations; PSO: Particle swarm optimization; ROC: Receiver operating characteristic; T1FS: Type-1 fuzzy system.

Table 13 shows the averages across the different metrics, indicating that, for validation #2 in the multiclass case, the proposed method achieved better performance. In this validation, the proposed method achieved a slightly higher average time. Regarding the number of operations and parameters when fuzzy logic is not used, the values are lower. However, it is important to note that the values (FLOPs and parameters) obtained by the proposed method are not superior.

The average accuracy achieved by the proposed method and by the PSO using ADAM without FL as the training algorithm for validation #2 are shown in Figures 16 and 17, respectively. The proposed method converges faster in both the binary and multiclass cases.

Table 14 shows the results obtained by modern architectures without modification, using only labeled images, as reported by Kassani *et al.*⁷¹ All models were trained with 72%, validated with 18%, and tested with 10%. The best accuracy achieved by our proposed method was 77.49%, while an average accuracy of 77.05% (accuracy) was obtained when using more images for the training phase. Our results can outperform those of some modern and complex architectures.

Table 13. Metrics and complexity average results (validation #2, multiclass)

Metric	PSO (Adam)	PSO (Adam + T1FS)
Accuracy	0.73997	0.75034
ROC-AUC	0.85592	0.86830
F1-Score	0.70056	0.71387
Sensitivity	0.46231	0.47749
Specificity	0.92436	0.92794
Training time	35 s	47 s
FLOPs	31.88 M	35.15 M
Parameters	8.52 M	9.42 M

Abbreviations: Adam: Adaptive moment estimation; AUC: Area under the curve; FLOPs: Floating-point operations; PSO: Particle swarm optimization; ROC: Receiver operating characteristic; T1FS: Type-1 fuzzy system.

Table 14. Comparison with other architectures (multiclass)

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)
Xception	79.59	82.35	86.32
InceptionV3	78.72	63.64	85.37
MobileNet	79.01	76.47	84.62
ResNet50	74.64	56.52	85.71
Our proposed model	77.05	50.85	93.46

5.4. Statistical comparison

The results obtained in both validations by the proposed method were used to compare it statistically with the PSO using the Adam algorithm without FL, which, in addition to CNN architecture optimization, seeks the best algorithm for the training phase in validation #1. A two-tailed *T*-test was conducted at 95% significance level, with a critical value of 2.1. The null hypothesis states that the means of the two methods are equal, whereas the alternative

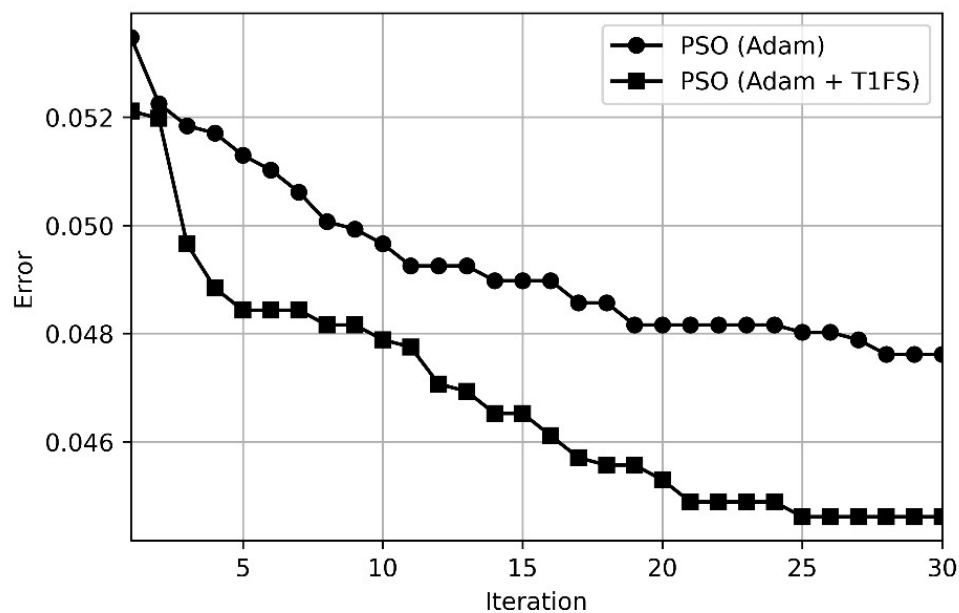


Figure 16. Average convergence for the binary case

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

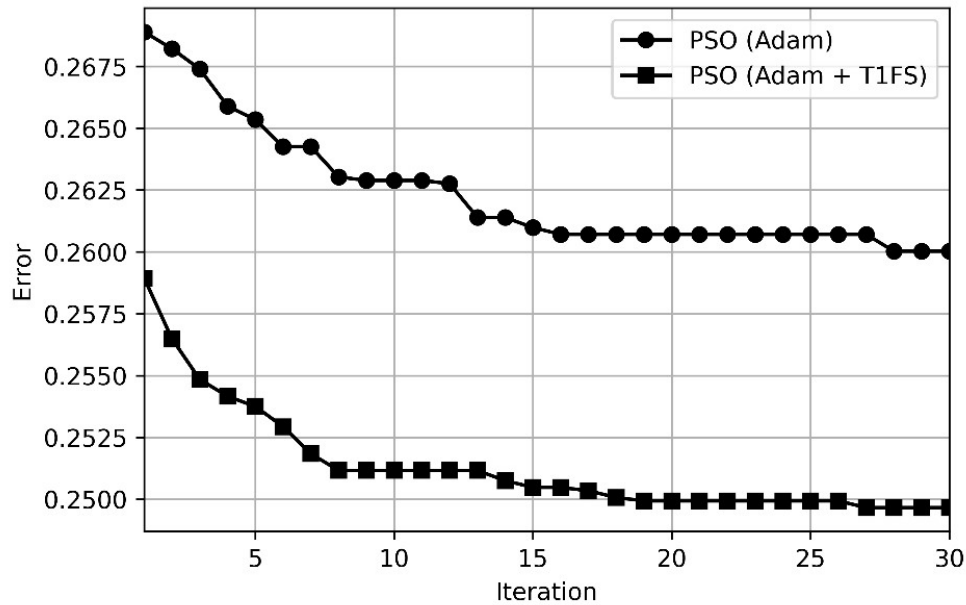


Figure 17. Average convergence for the multiclass case

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

hypothesis indicates a significant difference between them. The T -test values for the binary case are presented in Table 15, which shows that, based on the mean values, the proposed PSO with T1FS for learning rate estimation achieves a significant improvement over the simple PSO.

The T -test values for the multiclass case are presented in Table 16, where results based on the mean values demonstrate that the proposed PSO with T1FS for learning rate estimation also achieves a significant improvement over simple PSO methods.

Table 15. T -test values for detection (binary case)

Validation	Method	Mean	SD	t -value	p -value
1	PSO (Adam + T1FS)	96.29	0.191	3.78	0.00137
	PSO (Adam)	96.02	0.125		
2	PSO (Adam + T1FS)	95.54	0.214	2.44	0.0251
	PSO (Adam)	95.24	0.324		

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

Table 16. *T*-test values for classification (multiclass case)

Validation	Method	Mean	SD	<i>t</i> -value	<i>p</i> -value
1	PSO (Adam + T1FS)	77.05	0.264	7.23	1.00×10^{-6}
	PSO (Adam)	76.13	0.308		
2	PSO (Adam + T1FS)	75.03	0.340	6.86	2.02×10^{-6}
	PSO (Adam)	74.00	0.335		

Abbreviations: Adam: Adaptive moment estimation; PSO: Particle swarm optimization; T1FS: Type-1 fuzzy system.

6. Conclusion

In this work, a CNN design is proposed that uses PSO and T1FS during the training phase to determine the learning rate utilized by the Adam algorithm. The proposed T1FS is adapted based on the accuracy and loss values of the training and validation sets; that is, it uses these values to establish the ranges of the fuzzy variables. This process allows the fuzzy system to be updated at each epoch during training, thereby improving the average accuracy. The results obtained with the proposed method were compared with other CNN optimizations using PSO. The *T*-tests showed that the proposed method achieves higher average accuracy, improves the worst results, and outperforms the CNN design using Adam without FL, achieving the best result in both the binary and multiclass cases. The experiments conducted in this work were performed using a search space of 10 to 30 epochs, as preliminary experiments indicated that using very few epochs (5 to 10) leads to overfitting and prevents the models from reaching their full potential. As future work, learning rate estimation using other metaheuristics and the design of fuzzy if-then rules will be explored. Also, the implementation of type-2 FL in learning rate estimation will be considered to evaluate whether it can further improve the results obtained in this work. Experiments will also be performed to expand the search space and compare the proposed method with other modern architectures. Another aim is also to address the class imbalance using data augmentation techniques, which will also help improve generalization in the multiclass case.

Acknowledgments

We thank the Tijuana Institute of Technology/TecNM and SECIHTI for their support during the realization of this research.

Funding

None.

Conflict of interest

Oscar Castillo is an Editorial Board Member of this journal, but was not in any way involved in the editorial and peer-review process conducted for this paper, directly or indirectly. Separately, other authors declared that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Author contributions

Conceptualization: Daniela Sánchez, Patricia Melin

Formal analysis: Daniela Sánchez, Rodrigo Cordero-Martínez

Investigation: Daniela Sánchez, Rodrigo Cordero-Martínez

Methodology: Patricia Melin, Oscar Castillo

Writing—original draft: Daniela Sánchez

Writing—review & editing: Patricia Melin, Oscar Castillo

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

A publicly available dataset was analyzed in this study. The data can be accessed at <https://www.kaggle.com/c/aptos2019-blindness-detection> (accessed on October 11, 2025).

References

- Wang J, Xu L, Shao J, Lv J, Zeng X, Li W. SODAS: Second-order optimization differential architecture search for diabetic retinopathy prediction. *Neural Networks*. 2026;196:108419.
doi: 10.1016/j.neunet.2025.108419
- Pedrini A, Nowosielski Y, Rehak M. Diabetic retinopathy—recommendations for screening and treatment. *Wien Med Wochenschr*. 2025;175:253–263.
doi: 10.1007/s10354-025-01088-6
- Hayat M, Allawi Y, Alamro W, *et al.* A Hybrid Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM)–Attention Model Architecture for Precise Medical Image Analysis and Disease Diagnosis. *Diagnostics*. 2025;15(21):2673.
doi: 10.3390/diagnostics15212673
- Rivero-Cacho A, Sánchez-Barroso G, Pastor-Pérez P, García-Sanz-Calcedo J. Forecasting indoor air quality in high-risk operating theatres in hospitals using Artificial Neural Networks. *J Build Eng*. 2025;111:113339.
doi: 10.1016/j.jobe.2025.113339
- Castillo-Girone S, Munera S, Martínez-Sober M, Blasco J, Cubero S, Gómez-Sanchis J. Artificial Neural Networks in Agriculture, the core of artificial intelligence: What, When, and Why. *Comput Electron Agric*. 2025;230:109938.
doi: 10.1016/j.compag.2025.109938
- Castillo O, Melin P, Valdez F, *et al.* A Review on the Role of Fuzzy Logic in Hybrid Intelligent Systems. *Comput Sist*. 2025;29(3):1723–1740.
doi: 10.13053/cys-29-3-5897
- Liawrungrueang W, Cho S, Kotheeranurak V, Jitpakdee K, Kim P, Sarasombath P. Osteoporotic vertebral compression fracture (OVCF) detection using artificial neural networks model based on the AO spine-DGOU osteoporotic fracture classification system. *N Am Spine Soc J*. 2024;19:100515.
doi: 10.1016/j.xnsj.2024.100515
- da Silva G, Roque R, Arreguín Sámano M, Dias N, Golzio A, Bonini Neto A. Multilayer Perceptron, Radial Basis Function, and Generalized Regression Networks Applied to the Estimation of Total Power Losses in Electrical Systems. *Mach Learn Knowl Extr*. 2026;8(1):4.
doi: 10.3390/make8010004
- Karimov S, Sotvoldiyeva D, Khalilov D, Mamadaliyev N. Deep neural network for semantic segmentation of satellite images. In: *International Scientific Conference on Green Energy*. Vol 587. 2024:03006.
doi: 10.1051/e3sconf/202458703006
- Melin P, Ramirez M, Castillo O. *Clustering, Classification, and Time Series Prediction by Using Artificial Neural Networks*. Springer; 2024.
doi: 10.1007/978-3-031-71101-5
- Sayeh K, Tamalouzt S, Sahri Y, Belaid S, Bekhiti A. Artificial intelligence-based direct power control for power quality improvement in a WT-DFIG system via neural networks: Prediction and classification techniques. *J Franklin Inst*. 2025;362(1):107401.
doi: 10.1016/j.jfranklin.2024.107401
- Semmad A, Bahoura M. Comparative study of respiratory sounds classification methods based on cepstral analysis and artificial neural networks. *Comput Biol Med*. 2024;171:108190.
doi: 10.1016/j.compbiomed.2024.108190
- Conforti A, Ruggiero M, Lucchetti L, *et al.* Artificial Intelligence and Machine Learning in the Diagnosis and Management of Osteoporosis: A Comprehensive Review. *Medicina*. 2026;62(1):27.
doi: 10.3390/medicina62010027
- Sabarish J. Artificial Neural Networks in Medical Diagnosis. *Int J Res Publ Rev*. 2024;5(6):2576–2579.
doi: 10.55248/gengpi.5.0624.1505
- Castillo O. Review on Quantum Optimization of Type-1, Type-2 and Type-3 Fuzzy Systems. In: *Artificial Intelligence and Quantum Computing: Early Innovations*. Vol 1. Springer; 2025:235–244.
doi: 10.1007/978-3-031-85614-3_12
- Valdez F, Castillo O, Melin P. A Bibliometric Review of Type-3 Fuzzy Logic Applications. *Mathematics*. 2025;13(3):375.
doi: 10.3390/math13030375
- Rodrigues Lima-Junior F. Advances in Fuzzy Logic and Artificial Neural Networks. *Mathematics*. 2024;12(24):3949.
doi: 10.3390/math12243949
- Liu J, Lv W, Sun Z, Song Q. Modified correlation coefficients of hesitant fuzzy sets based on proximity and their application in medical diagnosis. *Comp Appl Math*. 2026;45(1).
doi: 10.1007/s40314-025-03383-9
- Sheth F, Mathur P, Gupta A, Chaurasia S. An advanced artificial intelligence framework integrating ensembled convolutional neural networks and Vision Transformers for precise soil classification with adaptive fuzzy logic-based crop recommendations. *Eng Appl Artif Intell*. 2025;158:111425.
doi: 10.1016/j.engappai.2025.111425
- Park S, Kim T. Fuzzy Inference System for Interpretable Classification of Wafer Map Defect Patterns. *Electronics*. 2025;15(1):130.
doi: 10.3390/electronics15010130

21. Carreras J, Roncador G, Hamoudi R. Ulcerative Colitis, LAIR1 and TOX2 Expression, and Colorectal Cancer Deep Learning Image Classification Using Convolutional Neural Networks. *Cancers*. 2024;16(24):4230.
doi: 10.3390/cancers16244230
22. Giri L, Regmi P, Gurung G, *et al.* Diagnostic Accuracy of an Offline CNN Framework Utilizing Multi-View Chest X-Rays for Screening 14 Co-Occurring Communicable and Non-Communicable Diseases. *Diagnostics*. 2025;16(1):66.
doi: 10.3390/diagnostics16010066
23. Poma Y, Melin P. Momentum Adaptation in Convolutional Neural Networks Using a Fuzzy Gravitational Algorithm. In: *Intelligent System Design Based on Soft Computing Models*. Vol 1223. Springer; 2025:91-101.
doi: 10.1007/978-3-031-97309-3_8
24. Emraha A, Yildirim Ö. Comparison of machine learning algorithms for automatic prediction of Alzheimer disease. *J Chin Med Assoc*. 2025;88(2):98-107.
doi: 10.1097/JCMA.0000000000001188
25. Srikanth M, Sunitha P, Puppala R, Asileti S, Akshaykranth A. A novel framework for intrusion detection in IOT networks using hybrid optimization algorithm and convolutional neural networks. *Franklin Open*. 2026;14:100461.
doi: 10.1016/j.fraope.2025.100461
26. Revathi K, Shirley C, Sreethar S, Ezhilarasi P. Brain tumor segmentation using optimized depth wise separable convolutional neural network with dense U-Net. *Knowl Based Syst*. 2025;324:113678.
doi: 10.1016/j.knosys.2025.113678
27. Rehman M, Aamir M, Nawi N, Khan A, Lashari S, Khan S. An Optimized Neural Network with Bat Algorithm for DNA Sequence Classification. *Comput Mater Contin*. 2022;73(1):493-511.
doi: 10.32604/cmc.2022.021787
28. Antonius F, Sekhar J, Rao V, *et al.* Unleashing the power of Bat optimized CNN-BiLSTM model for advanced network anomaly detection: Enhancing security and performance in IoT environments. *Alex Eng J*. 2023;84:333-342.
doi: 10.1016/j.aej.2023.11.015
29. Rashno A, Fadaei S. Convolutional neural networks optimization using multi-objective particle swarm optimization algorithm. *Inf Sci*. 2025;689:121443.
doi: 10.1016/j.ins.2024.121443
30. Jain S, Jain A, Jangid M. MGWO-CNN: hyperparameter optimization of CNN classifier for cervical cancer detection using Modified Grey Wolf Optimizer. *Sci Rep*. 2025;15(1):43514.
doi: 10.1038/s41598-025-26047-4
31. Ye J, Zhao Z, Ghafourian E, Tajally A, Alkhazaleh H, Lee S. Optimizing the topology of convolutional neural network (CNN) and artificial neural network (ANN) for brain tumor diagnosis (BTD) through MRIs. *Heliyon*. 2024;10(16):e35083.
doi: 10.1016/j.heliyon.2024.e35083
32. Naji R, Dulaimi H, Al-Khazraji H. An Optimized PID Controller Using Enhanced Bat Algorithm in Drilling Processes. *J Eur Syst Autom*. 2024;57(3):767-772.
doi: 10.18280/jesa.570314
33. Díaz-Bello D, Vargas-Salgado C, Montuori L, Alcázar-Ortega M. Cluster-based demand prediction: a hybrid approach with grey wolf optimizer and multilayer perceptron network. *Clean Technol Environ Policy*. 2025;27(12):8411-8429.
doi: 10.1007/s10098-025-03299-2
34. Mohakud R, Dash R. Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection. *J King Saud Univ Comput Inf Sci*. 2022;34(8):6280-6291.
doi: 10.1016/j.jksuci.2021.05.012
35. Pulido M, Melin P, Castillo O. Design of Ensemble Neural Networks with Type-3 Fuzzy Aggregation using Particle Swarm Optimization and Genetic Algorithms for Ethereum Prediction. *Comput Syst*. 2025;29(3):1765-1783.
doi: 10.13053/cys-29-3-5910
36. Sánchez D, Melin P, Castillo O, Castro J. Optimal Genetic Design of Interval Type-3 Fuzzy Aggregators for Modular Neural Networks Applied to Human Recognition. *Int J Fuzzy Syst*. 2025;28:85-104.
doi: 10.1007/s40815-024-01932-8
37. Mazouzi A, Hadroug N, Hafaifa A, Iratni A, Colak I. Particle swarm optimization of fuzzy logic-based energy management system for enhanced efficiency in fuel cell hybrid electric vehicles. *Sustain Comput Inform Syst*. 2025;48:101239.
doi: 10.1016/j.suscom.2025.101239
38. Hong TY, Chen CC. Hyperparameter optimization for convolutional neural network by opposite-based particle swarm optimization and an empirical study of photomask defect classification. *Appl Soft Comput*. 2023;148:110904.
doi: 10.1016/j.asoc.2023.110904
39. Mumali F, Kalkowska J. Intelligent support in manufacturing process selection based on artificial neural networks, fuzzy logic, and genetic algorithms: Current state and future perspectives. *Comput Ind Eng*. 2024;193:110272.
doi: 10.1016/j.cie.2024.110272
40. Xu XF, Wang K, Ma WH, *et al.* Multi-objective particle swarm optimization algorithm based on multi-strategy

- p>improvement for hybrid energy storage optimization configuration.
- Renew Energy*
- . 2024;223:120086.
-
- doi: 10.1016/j.renene.2024.120086
41. Luo X, Wang W, Xu Y, *et al.* A deep convolutional neural network for diabetic retinopathy detection via mining local and long-range dependence. *CAAI Trans Intell Technol*. 2024;9:153-166.
doi: 10.1049/cit2.12155
42. Sorgente V, Correra S, Verrillo I, *et al.* NeoNet: A Novel Deep Learning Model for Retinal Disease Diagnosis and Localization. *Sensors*. 2025;25(19):6147.
doi: 10.3390/s25196147
43. El-Hag N, Sedik A, El-Shafai W, *et al.* Classification of retinal images based on convolutional neural network. *Microsc Res Tech*. 2021;84(3):394-414.
doi: 10.1002/jemt.23596
44. Kansal S, Mishra B, Sethi S, Vinayak K, Kansal P, Narayan J. RetinoDeep: Leveraging Deep Learning Models for Advanced Retinopathy Diagnostics. *Sensors*. 2025;25(16):5019.
doi: 10.3390/s25165019
45. Elhani D, Megherbi A, Zitouni A, Dornaika F, Sbaa S, Taleb-Ahmed A. Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification. *Expert Syst Appl*. 2023;229:120411.
doi: 10.1016/j.eswa.2023.120411
46. Melin P, Sánchez D, Cordero-Martínez R. Particle Swarm Optimization of Convolutional Neural Networks for Diabetic Retinopathy Classification. In: *Fuzzy Logic and Neural Networks for Hybrid Intelligent System Design. Studies in Computational Intelligence*. Vol 1061. Springer; 2023:237-252.
doi: 10.1007/978-3-031-22042-5_14
47. Ibrahim AL, Fathi BG, Abdulrazzaq MB. Improving the Adam optimizer via projection-based gradient correction in deep learning. *Knowl Based Syst*. 2026;336:115267.
doi: 10.1016/j.knosys.2026.115267
48. Cheng W, Pu R, Wang B. AMC: Adaptive Learning Rate Adjustment Based on Model Complexity. *Mathematics*. 2025;13(4):650.
doi: 10.3390/math13040650
49. Wu T, Dong Q, Yu W, Li Z, Guo M. A general learning rate improvement strategy for deep neural networks training. *Complex Intell Syst*. 2026;12:106.
doi: 10.1007/s40747-026-02226-2
50. Feng L, Hao G, Jian W. Efficient hybrid fuzzy weighted 3D FCNN with TSO PSO optimization for accurate multi modal MRI brain tumor classification. *Sci Rep*. 2025;15:36037.
doi: 10.1038/s41598-025-19889-5
51. Sobhanifard F. Optimization of Bayesian Neural Networks using hybrid PSO and fuzzy logic approach for time series forecasting. *Discov Artif Intell*. 2025;5:132.
doi: 10.1007/s44163-025-00322-9
52. Aggarwal C. *Neural Networks and Deep Learning: A Textbook*. Springer; 2023.
doi: 10.1007/978-3-031-29642-0
53. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016. Available from: <https://mitpress.mit.edu/9780262035613/deep-learning/> [Last accessed on October 11, 2025].
54. Zafar A, Aamir M, Nawi N, *et al.* A Comparison of Pooling Methods for Convolutional Neural Networks. *Appl Sci*. 2022;12(17):8643.
doi: 10.3390/app12178643
55. Linke M, Meßmer T, Micard G, Schubert G. Power grid operation in distribution grids with convolutional neural networks. *Smart Energy*. 2025;17:100169.
doi: 10.1016/j.segy.2024.100169
56. Ozturk S. *Convolutional Neural Networks for Medical Image Processing Applications*. 1st ed. CRC Press; 2022.
doi: 10.1201/9781003215141
57. Zadeh L. Fuzzy sets. *Inf Control*. 1965;8(3):338-353.
doi: 10.1016/s0019-9958(65)90241-x
58. Masjedy H, Reza Adel S. Fuzzy logic theory: A unifying pathway to intelligent accountability in addressing methodological challenges of applied linguistics. *Res Methods Appl Linguist*. 2025;4(3):100278.
doi: 10.1016/j.rmal.2025.100278
59. Saatchi R. Fuzzy Logic Concepts, Developments and Implementation. *Information*. 2024;15(10):656.
doi: 10.3390/info15100656
60. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE; 1995:39-43.
doi: 10.1109/mhs.1995.494215
61. Liu W, Bai Y, Zhang C, Wang Z, Yang A, Wu M. PSO-DFNN: A particle swarm optimization enabled deep fuzzy neural network for predicting the pellet strength. *Alex Eng J*. 2024;106:505-516.
doi: 10.1016/j.aej.2024.08.069
62. Kowalski P, Słoczyński T. A modified particle swarm optimization procedure for triggering fuzzy flip-flop neural networks. *Int J Appl Math Comput Sci*. 2021;31(4):577-586.
doi: 10.34768/amcs-2021-0039

63. Engelbrecht AP. *Fundamentals of Computational Swarm Intelligence*. Wiley; 2005.
64. Eberhart R, Shi Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. Vol 1. 2000:84-88.
doi: 10.1109/cec.2000.870279
65. Xin J, Chen G, Hai Y. A Particle Swarm Optimizer with Multi-stage Linearly-Decreasing Inertia Weight. In: *2009 International Joint Conference on Computational Sciences and Optimization*. 2009:505-508.
doi: 10.1109/cso.2009.420
66. Kingma D, Ba J. Adam: A Method for Stochastic Optimization. *arXiv*. Preprint posted online 2014.
doi: 10.48550/arXiv.1412.6980
67. Sánchez D. Optimization of Interval Type-3 Fuzzy Inference Systems Using a Genetic Algorithm Applied to Medical Classification. In: *Modern Artificial Intelligence Based on Soft Computing Techniques*. Vol 1204. Springer; 2025:381-394.
doi: 10.1007/978-3-031-88279-1_24
68. Asia Pacific Tele-Ophthalmology Society. APTOS 2019 Blindness Detection. Kaggle. Available from: <https://www.kaggle.com/c/aptos2019-blindness-detection> [Last accessed on October 11, 2025].
69. Gangwar A, Ravi V. Diabetic retinopathy detection using transfer learning and Deep Learning. In: *Evolution in Computational Intelligence*. Vol 1176. Springer; 2021:679-689.
doi: 10.1007/978-981-15-5788-0_64
70. Aslan E, Özüpak Y, Alpsalaz F, Elbarbary ZMS. A Hybrid Machine Learning Approach for Predicting Power Transformer Failures Using Internet of Things-Based Monitoring and Explainable Artificial Intelligence. *IEEE Access*. 2025;13:113618-113633.
doi: 10.1109/ACCESS.2025.3583773
71. Kassani SH, Kassani PH, Khazaeinezhad R, Wesolowski MJ, Schneider KA, Deters R. Diabetic Retinopathy Classification Using a Modified Xception Architecture. In: *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE; 2019:1-6.
doi: 10.1109/isspit47144.2019.9001846