ISSN 2007-9737

Adaptive Learning Rate Adjustment via Type-2 Fuzzy Logic in Convolutional Neural Networks for Diabetic Retinopathy Detection and Classification

Rodrigo Cordero-Martínez, Daniela Sánchez, Patricia Melin*

Tijuana Institute of Technology / TecNM, Tijuana, Mexico

{rodrigo.cordero201, daniela.sanchez, pmelin}@tectijuana.mx

Abstract. One problem that the medical field has faced is the early detection of various existing diseases. Patients with diabetes mellitus are prone to additional conditions, one of which is diabetic retinopathy. Due to the increasing number of people with diabetes mellitus, the number of expert technicians is insufficient to adequately treat them. To solve this problem, computer tools have been used to automate the detection of diseases. One of these tools is the use of artificial neural networks. These networks have the characteristic that they can be adapted to a specific disease, which allows for the creation of different neural network models. Each model has parameters that adjust the weights of its neurons. These parameters are assigned by the designer of network architecture before training. This requires time for testing and fine-tuning the parameters until the desired result is obtained. One of these parameters is the learning rate of the training algorithm. This value can only be modified before training, so selecting the most appropriate one may require a significant investment in time and analysis. This work proposes a method that adjusts the learning rate of the Adaptive Moment Estimation training algorithm between each epoch using interval Type-2 and generalized Type-2 fuzzy inference systems, taking as input the average training and validation loss values, as well as the epoch number. This reduces analysis time, allowing for a focus on other network parameters. The proposed method is applied to two different convolutional neural network architectures: disease detection and classifying disease severity.

Keywords. Convolutional neural networks, type-2 fuzzy logic, learning rate, image classification, diabetic retinopathy, adaptive moment estimation, stochastic gradient descent.

1 Introduction

In the medical field, expert technicians face the constant challenge of detecting potential diseases in their patients [17, 46]. To achieve this, there are different tools to use depending on the symptoms presented. Although the expert technician is responsible for providing an accurate response, doing so requires time, which prevents many detections, limiting the number of people treated and increasing the risks for those who were not diagnosed in time. This situation is common in patients with both type 1 and type 2 diabetes mellitus [4, 7, 29, 33], as these patients may have additional consequences that require their own care, with diabetic retinopathy (DR) being one of the most affected patients [14, 25].

The importance of detecting and classifying DR lies in the timing of treatment. In the early stages, the symptoms are almost imperceptible; however, when the patient begins to see the consequences, it means the stage of DR is advanced [9]. A late diagnosis can affect partial or total vision of the patient [21]. Due to the seriousness of early detection of DR, expert technicians must turn to tools that automate disease detection and classification, finding artificial neural networks a practical and reliable solution for delegating this responsibility [16]. There are different types of networks, with convolutional neural networks (CNN) being the most appropriate for working with images [15, 19, 24, 30, 42]. These networks produce network models that allow

disease detection or classification [8, 26], but developing the architecture requires significant knowledge on the part of the developer. To streamline the training process and improve the results obtained by the network, it can be combined with other computing tools such as fuzzy logic [1, 18] to create an intelligent hybrid system.

This work begins with the application of preprocessing to the database to improve neural network training by facilitating the extraction of key features of the consequences of DR in retinal photographs. Following this, convolutional neural network architectures focused on detection and another for classification are developed. Finally, a fuzzy inference system (FIS) is designed to update certain parameters of the training algorithms of the neural network.

Before training any neural network, the parameters of the functions that make up the network must be specified, one of these parameters being the training algorithm. There are different training algorithms that, depending on the network's objective, adjust the weights of the neurons using the learning rate hyperparameter. Obtaining the value of the hyperparameters using fuzzy logic allows for stability in training by adjusting these values according to the input information.

The structure of this work is presented below: Section 2 presents a review of previous work developed by various researchers in the field. Section 3 details the essential theoretical concepts, accompanied by examples that contribute to a better understanding of the study. Section 5 compiles the results obtained from the experiments carried out using the proposed method, while Section 6 highlights the most relevant conclusions according to the hypothesis tests and the observation of boxplots derived from the analysis.

2 Related Work

In the study conducted by Sarwo [41], the performance of the Adaptive Moment Estimation (Adam) and Stochastic Gradient Descent (SGD) training algorithms in training CNNs applied to offline handwritten text recognition (OHTR) is analyzed. Both methods are evaluated considering their influence on convergence speed, training stability, and final model accuracy.

Experimental results indicate that SGD outperforms Adam in most cases, offering faster convergence and higher accuracy in the tests performed. In contrast, Adam demonstrates better generalization capabilities under certain dataset conditions, suggesting that the choice of optimizer should be tailored to the specific characteristics of the problem and the model.

From the experiments performed, the model trained with SGD obtained five correct results and one incorrect result, while the model trained with ADAM obtained two correct results and three incorrect results. In conclusion, the use of SGD was more effective for the analyzed case, although it is recognized that the behavior of the training algorithms may vary depending on the CNN parameters and conditions. This work provides a deeper understanding of the impact on the performance of training algorithms in a scenario where the use of CNNs is applied.

The recent study [22] focuses on the development of an ocular disease detection model using CNN-based deep learning techniques. The main objective is to identify the most effective training algorithm among Root Mean Square Propagation (RMSProp), SGD, and Adam to improve the model's performance in classifying various ocular pathologies, such as cataracts, glaucoma, and DR, in addition to the normal class.

According to their results, the SGD and Adam training algorithms performed worse, demonstrating that RMSProp favors more stable convergence and better adaptation to the data type used. In conclusion, the use of RMSProp allowed the learning of the CNN to be trained, offering an adequate balance between convergence speed and accuracy, and constituting a solid alternative for the development of automatic eye disease detection systems.

3 Basic Concepts

The second section introduced certain terms that may be unfamiliar to readers without prior experience in the use of hybrid intelligent systems. Therefore, this section offers the

necessary information to ensure a comprehensive understanding of the study.

3.1 Fuzzy Logic

Type-1 fuzzy logic is a reasoning system that handles imprecise or gradual concepts, such as ”tall,” ”hot,” or ”fast,” using fuzzy sets instead of absolute values (true/false). In this system, inputs (which can be precise values or fuzzy sets) are converted into degrees of fuzzy set membership using membership functions. These memberships represent how well a value fits a linguistic category [2, 28, 31].

The process follows a rule-based structure of the ”IF (antecedent) THEN (consequent)” type, where the antecedent combines one or more fuzzy conditions and the consequent defines a fuzzy output. The ”firing strength” of each rule is determined by how closely the inputs fit the conditions of each rule. The outputs of all active rules are combined to form an aggregate fuzzy output, which is then converted into a concrete numerical value through a process called defuzzification. This final value is the answer of the system and is used for decision-making or control [38, 43].

In short, Type-1 fuzzy logic models the imprecision and approximate reasoning inherent in human language, allowing automated systems to make decisions based on qualitative descriptions rather than exact measurements.

3.1.1 Interval Type-2 Fuzzy Logic

Interval Type-2 (IT2) fuzzy logic is an extension of traditional fuzzy logic used when there is a greater degree of uncertainty in the definition of the rules or input data. Unlike Type-1 logic, where each value has a unique degree of membership in a set, in Interval Type-2, the degree of membership is itself an interval, that is, a range of values between a lower and upper bound. This allows uncertainty to be captured and preserved throughout the inference process [20, 23].

The system operates by transforming numerical inputs into Type-2 fuzzy sets, applying fuzzy rules that relate antecedents and consequents, and combining the results of multiple rules. Then, through a process called type reduction, the resulting Type-2 fuzzy set is converted into a Type-1 fuzzy set, which is an interval. Finally, this interval is transformed into a concrete numerical value through defuzzification, typically by taking the midpoint of the interval [35, 37].

The main advantage of this approach is its ability to explicitly handle uncertainty within the system, making it especially useful in applications where definitions are vague or the data exhibits significant variability.

3.1.2 Generalized Type-2 Fuzzy Logic

Generalized Type-2 (GT2) fuzzy logic is an advanced extension of fuzzy logic that allows for handling a higher degree of non-uniform uncertainty in inference systems. Unlike IT2 fuzzy logic, where uncertainty is represented homogeneously by an interval, in GT2 fuzzy logic, uncertainty can vary non-uniformly across membership functions [32].

In this system, the degree of membership is not a single value or a fixed interval but is represented by a secondary membership function that is itself a Type-1 fuzzy set. This means that, for each input value, there is not only a range of possible membership degrees, but within that range, each possible degree has its own level of truth or membership. This creates a three-dimensional structure that captures more complex and detailed variations in uncertainty [36].

The system operates by transforming inputs into GT2 fuzzy sets, applying rules with Type-2 antecedents and consequents, and combining the results using specialized operations such as join and meet tailored for this type of set. Finally, the resulting Type-2 output set is processed to obtain a specific output [6].

The primary advantage of this approach is its ability to model and preserve more complex and non-uniform uncertainties, making it suitable for applications where the sources of inaccuracy are varied or where uncertainty is not homogeneously distributed.

3.1.3 Convolutional Neural Networks

Among the most widely used machine learning tools for disease detection are supervised artificial neural networks, which allow specialized technicians to train the model using a labeled database [34]. When images are utilized during the training phase, it is crucial to define the relevant information to guarantee precise classification in later stages. This highlights the importance of expert supervision in image management and analysis. A CNN is a type of network capable of performing this task automatically through the use of convolution functions and filters [5].

The design of a CNN model shares similarities with that of a feedforward neural network, differing mainly in the way it processes information after the input stage [10]. In CNNs, the convolution layer constitutes the first stage of the architecture and its function is to identify the most relevant features of the input images [40]. Thanks to this capability, the network eliminates the need for manual preprocessing methods, as the convolution process uses a kernel or filter that generates a new feature matrix.

Subsequently, the Rectified Linear Unit (ReLU) activation function takes a fundamental role in the CNN architecture. This widely used function is applied after each neuron and allows only positive values to be activated, significantly speeding up training time and contributing to the model's efficiency [44]. Finally, although the activation function speeds up the learning process, the size of the images remains constant, and not all pixels have the same relevance. To optimize this aspect, a reduction method known as pooling is used, with MaxPooling being one of the most common techniques in the field of CNNs [45]. This procedure allows image dimensions to be reduced while preserving the most representative features, which improves the performance of the model.

4 Proposed Method

This section addresses the concepts presented previously. First, the architectures of the CNN models employed in other researches, to which the proposed method will be applied, are described in detail.

Subsequently, the APTOS 2019 database is presented, along with its main aspects and the two ways employed in the case studies of this research. The preprocessing process applied to the database is then explained in detail. Finally, the creation and implementation of an FIS designed to tune the hyperparameters of the training algorithm are described.

4.1 CNN Models

Two CNN models derived from the research on DR were used [12, 13]. The first model, aimed at disease detection, uses input images with a size of 256 × 256 × 3 (width, height, and depth). It is composed of five convolutional layers, some of which include a subsequent MaxPooling layer, but not all. It also incorporates three hidden layers with different numbers of neurons and uses the Sigmoid activation function at the end of the model.

The second model, focused on disease classification, maintains the same input size and also has five convolutional layers, although with different hyperparameter values (such as the MaxPooling size or the use of Dropout, when applicable). This model includes a single hidden layer and uses the Softmax activation function on the output. Both used the Adam training algorithm with a fixed learning rate of 0.001.

4.2 APTOS 2019 Database

The APTOS 2019 database on DR contains a total of 3,662 labeled images, used for the training and validation phases [3]. This database is composed of five classes representing different levels of damage caused by the disease and is used in multiclass/classification case studies [27]. However, it is also possible to group four of these classes to form only two categories in binary/detection cases: images of healthy retinas and images of retinas with DR [39].

4.3 APTOS 2019 Preprocessing Method

The proposed method involves starts removing the problematic pixels from the image background to fully isolate the retina. To accomplish this, the images are first converted to grayscale.

Once the grayscale version is obtained, it can be transformed into a binary image, a process that involves selecting each pixel to refine the contours. At this stage, the brightness level of each pixel must surpass a predefined threshold to prevent it from being classified as noise and, instead, allow its use for accurate retinal extraction.

Using the resulting binary image, the region corresponding to the retina is located and identified. This task is simplified because, in the binary image, it is sufficient to recognize the most prominent shape. Once the retina is detected, its position is determined, and this information is utilized to isolate it from the original image.

To complete the preprocessing method, black pixels were added where necessary to achieve a width equal to the height, thus preventing image distortion. This method has demonstrated superior performance compared to other preprocessing techniques focused on retinal images [11].

4.4 Description of the FIS

For the development of both case studies based on the APTOS 2019 database, a Mamdani Interval or GT2 FIS is required. All membership functions used are triangular.

Two FIS systems with similar characteristics are used. Both have two inputs: the mean loss and the current training epoch. The systems generate two outputs: the first corresponds to the learning rate used in the Adam and SGD training algorithms, while the second output, presented only in the FIS associated with SGD, represents the momentum. Both the inputs and the momentum output have three membership functions, while the learning rate output has five. The graphical representation of the FIS used for the training algorithms is shown in Figure 1. The mean loss value is normalized to a range of 0 to 0.001 for Adam and 0 to 0.01 for SGD, while the current epoch is set to the interval 0 to 1.0. The Centroid defuzzification method selected is the FIS.

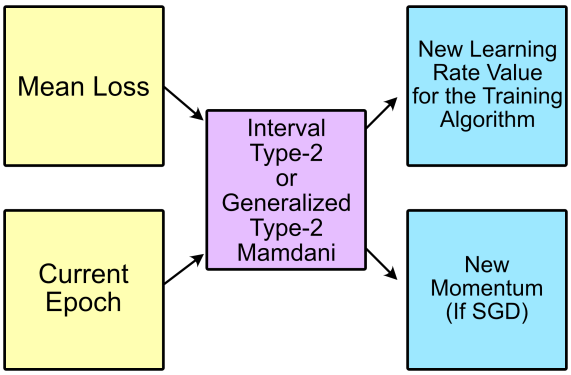

Fig. 1. Graphic representation of the FIS

Within the context of the experiments conducted, the FIS incorporates a total of nine fuzzy "if-then" rules, obtained through trial and error, which are detailed in Table 1. The rules are identical for both systems, with the only difference being that the FIS corresponding to Adam does not include momentum-related rules. The equations corresponding to the IT2 FIS are presented in:

$$\tilde{\mu}(x) = [\underline{\mu}(x), \overline{\mu}(x)] : \textbf{triScaleIntervalType2MF}(x, \{\{[a\, b\, c]\}, \lambda, [\ell_1, \ell_2]\}), \tag{1}$$

$$\overline{\mu}(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b < x \leq c \\ 0 & x > c. \end{cases} \tag{2}$$

While those for the GT2 FIS are shown in:

$$\mu_{\tilde{A}}(x, u) = \textbf{ScaleTriGaussT2MF}(x, \{\{[a\, b\, c]\}, \lambda, [\ell_1, \ell_2]\}), \tag{3}$$

$$\overline{\mu}(x) = \begin{cases} 0 & x < a_1 \\ \frac{x-a_1}{b_1-a_1} & a_1 \leq x \leq b_1 \\ \frac{c_1-x}{c_1-b_1} & b_1 < x \leq c_1 \\ 0 & x > c_1, \end{cases} \tag{4}$$

$$a_2 = b_1 - (b_1 - a_1)(1 - \ell_1), \tag{5}$$

$$c_2 = b_2 - (c_1 - b_1)(1 - \ell_2), \tag{6}$$

$$\mu(x) = \begin{cases} 0 & x < a_2 \\ \frac{x-a_2}{b_1-a_2} & a_2 \leq x \leq b_1 \\ \frac{c_2-x}{c_2-b_1} & b_1 < x \leq c_2 \\ 0 & x > c_2. \end{cases} \quad (7)$$

The parameters used in this work for the equations can be observed in Table 2.

5 Experimental Results

This section integrates all the concepts and proposed methods to conduct distributed experimentation across two case studies: disease detection and classification.

5.1 Detection Experiments

Five experiments were conducted; the initial one involved employing the CNN model mentioned in Section 4.1 for disease detection. The second and third experiments utilized the CNN model with the proposed method using IT2 FIS with Adam for the second experiment and SGD for the third experiment. Finally, the fourth and fifth experiments utilized the CNN model with the proposed method using GT2 FIS with Adam for the fourth experiment and SGD for the fifth experiment.

Each experiment was iterated 30 times with the objective to get the mean accuracy (MA), standard deviation (SD), and maximum accuracy (MxA), maintaining consistent hyperparameters: 100 epochs and the use of the APTOS 2019 database with detection distribution. The complete results can be observed in Table 3 and the MA, SD and MxA values of the experiments can be observed in Table 4.

5.1.1 Boxplot for the Detection Experiments

A boxplot was created to compare the obtained values. The boxplot for the detection experiments is shown in Figure 2.

5.1.2 Hypothesis Testing for Detection Experiments

Based on the results presented in Table 3 and 4, different hypothesis tests were performed using the MA and SD obtained from the proposed method applied to Adam and SGD using IT2 against the results of the traditional CNN model. Using an Alpha value of 0.05, it is necessary to exceed the critical z-value of 1.645. With the proposed method in Adam, a z-value of 6.67570117 was obtained, and applied to SGD, a z-value of 4.927779759 was obtained. In both scenarios, the null hypothesis was rejected, with sufficient evidence to support our claim that the proposed method using IT2 offers better results.

Then, two hypothesis tests were performed comparing the proposed method applying GT2 against the proposed method applying IT2 in the Adam and SGD training algorithms. The proposed method in Adam obtained a z-value of -0.29879285, and when applied to SGD, a z-value of 0.269858841 was obtained. In both scenarios, the null hypothesis was rejected, with insufficient evidence to support our claim that the proposed method using GT2 offers better results than the proposed method using IT2.

Finally, a hypothesis test was performed comparing the best results obtained according to the previous hypothesis tests. The best results were obtained by the proposed method using IT2 in SGD against the proposed method using IT2 in Adam. The result of the test gave a z-value of 2.10278593, concluding that the null hypothesis was rejected, with sufficient evidence to support our claim that the proposed method using IT2 applied to SGD offers better results than the same implementation in Adam.

5.2 Classification Experiments

Five experiments were conducted; the initial one involved employing the CNN model mentioned in Section 4.1 for disease classification. The second and third experiments utilized the CNN model with the proposed method using IT2 FIS with Adam for the second experiment and SGD for the third experiment. Finally, the fourth and fifth experiments utilized the CNN model with the

Table 1. Fuzzy Rules

Rule number	Inputs variables		Outputs variables	
	Mean Loss	Current Epoch	Learning Rate	Momentum
1	Low	Start	Very Low	Low
2	Low	Medium	Low	Low
3	Low	Finish	Low	Low
4	Medium	Start	Medium	Medium
5	Medium	Medium	Medium	Medium
6	Medium	Finish	Medium	Medium
7	High	Start	High	High
8	High	Medium	High	High
9	High	Finish	Very High	High

Table 2. Parameters used to generate the membership functions

Variable Name	Type of variable	a	b	c	Scale	Lower Lag_1	Lower Lag_2
Mean Loss	Input	-0.0833	0	0.0833	0.95	0.2	0.2
		0.0167	0.1	0.1833			
		0.1167	0.2	0.2833			
Current Epoch		-0.4167	0	0.4167	0.9	0.25	0.25
		0.0833	0.5	0.9167			
		0.5833	1	1.4167			
Learning Rate	Output	-0.0021	0	0.0021	0.95	0.3	0.3
		0.0004	0.0025	0.0046			
		0.0029	0.0.005	0.0071			
		0.0054	0.0075	0.0096			
		0.0079	0.1	0.0121			
Momentum		-0.4167	0	0.4167	0.95	0.3	0.3
		0.0833	0.5	0.9167			
		0.5833	1	1.4167			

proposed method using GT2 FIS with Adam for the fourth experiment and SGD for the fifth experiment.

In the same way as detection experiments, each experiment for classification was iterated the same times as detection experimentations, maintaining the consistent of hyperparameters just changing the distribution to classification. The complete results can be observed in Table 5 and the MA, SD and MxA of the experiments can be observed in Table 6.

5.2.1 Boxplot for the Classification Experiments

A boxplot was created to compare the obtained values. The main results are presented in the

Table 3. Results obtained for the detection experiments

Experiment	Traditional	IT2 ADAM	IT2 SGD	GT2 ADAM	GT2 SGD
1	0.952251017	0.972714841	0.97135061	0.95634377	0.952251017
2	0.938608468	0.972714841	0.964529335	0.968622088	0.97135061
3	0.959072292	0.957708061	0.95634377	0.963165104	0.967257857
4	0.953615308	0.954979539	0.963165104	0.965893567	0.965893567
5	0.953615308	0.949522495	0.965893567	0.963165104	0.952251017
6	0.954979539	0.952251017	0.968622088	0.964529335	0.964529335
7	0.948158264	0.952251017	0.964529335	0.976807654	0.957708061
8	0.957708061	0.959072292	0.969986379	0.961800814	0.969986379
9	0.920873106	0.961800814	0.965893567	0.945429742	0.969986379
10	0.95634377	0.965893567	0.969986379	0.968622088	0.968622088
11	0.948158264	0.965893567	0.964529335	0.945429742	0.959072292
12	0.959072292	0.942701221	0.946793973	0.978171885	0.964529335
13	0.953615308	0.964529335	0.953615308	0.960436583	0.965893567
14	0.941336989	0.953615308	0.963165104	0.967257857	0.964529335
15	0.953615308	0.959072292	0.967257857	0.957708061	0.965893567
16	0.938608468	0.961800814	0.95634377	0.959072292	0.964529335
17	0.967257857	0.952251017	0.965893567	0.944065511	0.969986379
18	0.944065511	0.946793973	0.97135061	0.961800814	0.953615308
19	0.949522495	0.95634377	0.915416121	0.974079132	0.963165104
20	0.953615308	0.950886786	0.950886786	0.961800814	0.967257857
21	0.95634377	0.954979539	0.967257857	0.961800814	0.968622088
22	0.952251017	0.965893567	0.964529335	0.963165104	0.95634377
23	0.959072292	0.960436583	0.963165104	0.964529335	0.954979539
24	0.931787193	0.960436583	0.968622088	0.960436583	0.968622088
25	0.949522495	0.957708061	0.968622088	0.959072292	0.953615308
26	0.942701221	0.953615308	0.968622088	0.967257857	0.963165104
27	0.953615308	0.960436583	0.972714841	0.965893567	0.967257857
28	0.95634377	0.948158264	0.978171885	0.974079132	0.968622088
29	0.949522495	0.972714841	0.972714841	0.967257857	0.972714841
30	0.95634377	0.961800814	0.965893567	0.967257857	0.972714841

Table 4. Summary of the results for detection experiments

Experiment	Traditional	IT2 ADAM	IT2 SGD	GT2 ADAM	GT2 SGD
Average	0.950386542	0.958299224	0.963528875	0.963165079	0.96416553
Std. dev.	0.009256188	0.007605843	0.01130079	0.008138358	0.006266501
Max	0.967257857	0.972714841	0.978171885	0.978171885	0.972714841

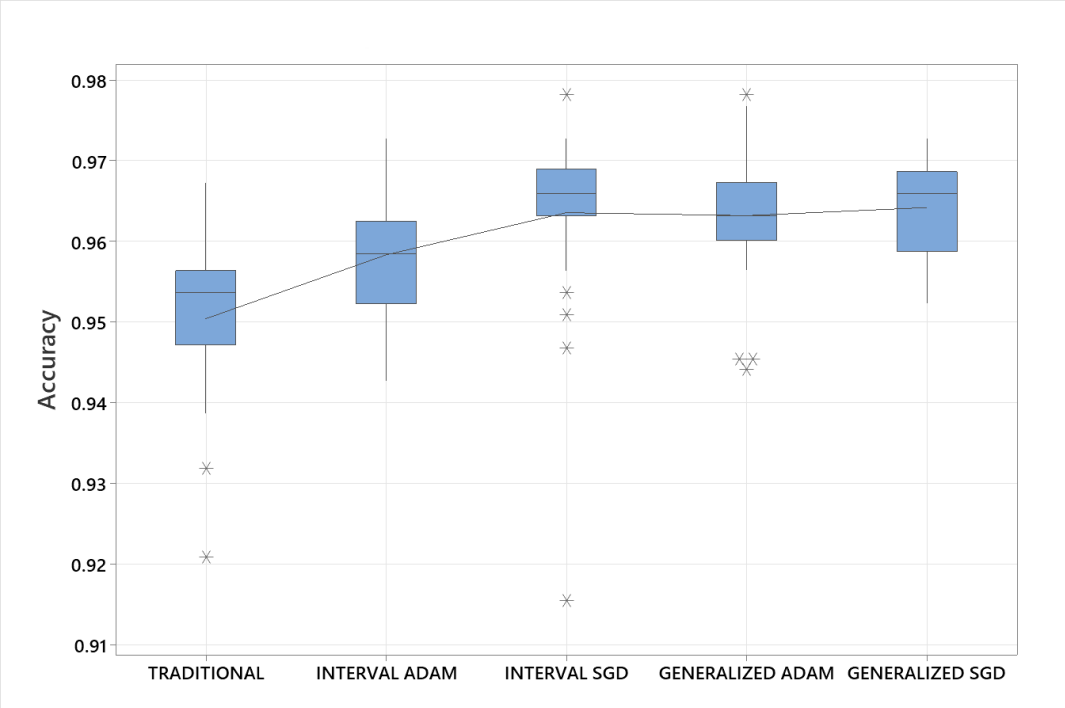

Fig. 2. Boxplot of detection experiments

boxplot obtained after the experiments for the classification, see Figure 3.

5.2.2 Hypothesis Testing for Classification Experiments

Based on the results presented in Table 5 and Table 6, different hypothesis tests were performed using the MA and SD obtained from the proposed method applied to Adam and SGD using IT2 against the results of the traditional CNN model. Using an Alpha value of 0.05, it is necessary to exceed the critical z-value of 1.645. With the proposed method in Adam, a z value of 2.54870324 was obtained, and applied to SGD, a z-value of 2.637427297 was obtained. In both scenarios, the null hypothesis was rejected, with sufficient evidence to support our claim that the proposed method using IT2 offers better results.

Then, two hypothesis tests were performed comparing the proposed method applying GT2 against the proposed method applying IT2 in the Adam and SGD training algorithms. The proposed method in Adam obtained a z-value of -0.384819951, and when applied to SGD, a z-value of -0.730549592 was obtained. In both scenarios, the null hypothesis was rejected, with insufficient evidence to support our claim that the proposed

Table 5. Results obtained for the detection experiments

Experiment	Traditional	IT2 ADAM	IT2 SGD	GT2 ADAM	GT2 SGD
1	0.732605755	0.735334218	0.725784421	0.720327437	0.750341058
2	0.721691668	0.732605755	0.713506162	0.731241465	0.727148712
3	0.717598915	0.718963146	0.723055959	0.744884014	0.731241465
4	0.716234624	0.731241465	0.740791261	0.736698508	0.723055959
5	0.731241465	0.72442019	0.736698508	0.728512943	0.723055959
6	0.729877234	0.728512943	0.725784421	0.732605755	0.732605755
7	0.731241465	0.728512943	0.743519783	0.740791261	0.738062739
8	0.735334218	0.736698508	0.738062739	0.748976827	0.744884014
9	0.725784421	0.725784421	0.747612536	0.740791261	0.721691668
10	0.701227844	0.733969986	0.740791261	0.750341058	0.731241465
11	0.712141871	0.708049119	0.742155552	0.743519783	0.735334218
12	0.738062739	0.735334218	0.725784421	0.723055959	0.732605755
13	0.733969986	0.748976827	0.732605755	0.727148712	0.729877234
14	0.714870393	0.732605755	0.736698508	0.735334218	0.75306958
15	0.723055959	0.717598915	0.740791261	0.713506162	0.750341058
16	0.735334218	0.716234624	0.718963146	0.728512943	0.735334218
17	0.720327437	0.714870393	0.731241465	0.720327437	0.740791261
18	0.731241465	0.723055959	0.725784421	0.723055959	0.738062739
19	0.742155552	0.742155552	0.731241465	0.712141871	0.727148712
20	0.748976827	0.720327437	0.727148712	0.729877234	0.731241465
21	0.731241465	0.725784421	0.733969986	0.721691668	0.731241465
22	0.721691668	0.746248305	0.727148712	0.742155552	0.731241465
23	0.727148712	0.720327437	0.729877234	0.728512943	0.729877234
24	0.727148712	0.712141871	0.736698508	0.733969986	0.735334218
25	0.736698508	0.716234624	0.735334218	0.728512943	0.750341058
26	0.720327437	0.713506162	0.733969986	0.732605755	0.727148712
27	0.725784421	0.735334218	0.731241465	0.758526623	0.735334218
28	0.723055959	0.705320597	0.738062739	0.744884014	0.49249658
29	0.731241465	0.728512943	0.742155552	0.75306958	0.738062739
30	0.721691668	0.731241465	0.731241465	0.723055959	0.736698508

Table 6. Summary of the results for detection experiments

Experiment	Traditional	IT2 ADAM	IT2 SGD	GT2 ADAM	GT2 SGD
Average	0.726966802	0.726330147	0.732924054	0.733287861	0.726830374
Std. dev.	0.009631659	0.010834971	0.007764521	0.011618985	0.045022151
Max	0.748976827	0.748976827	0.747612536	0.758526623	0.75306958

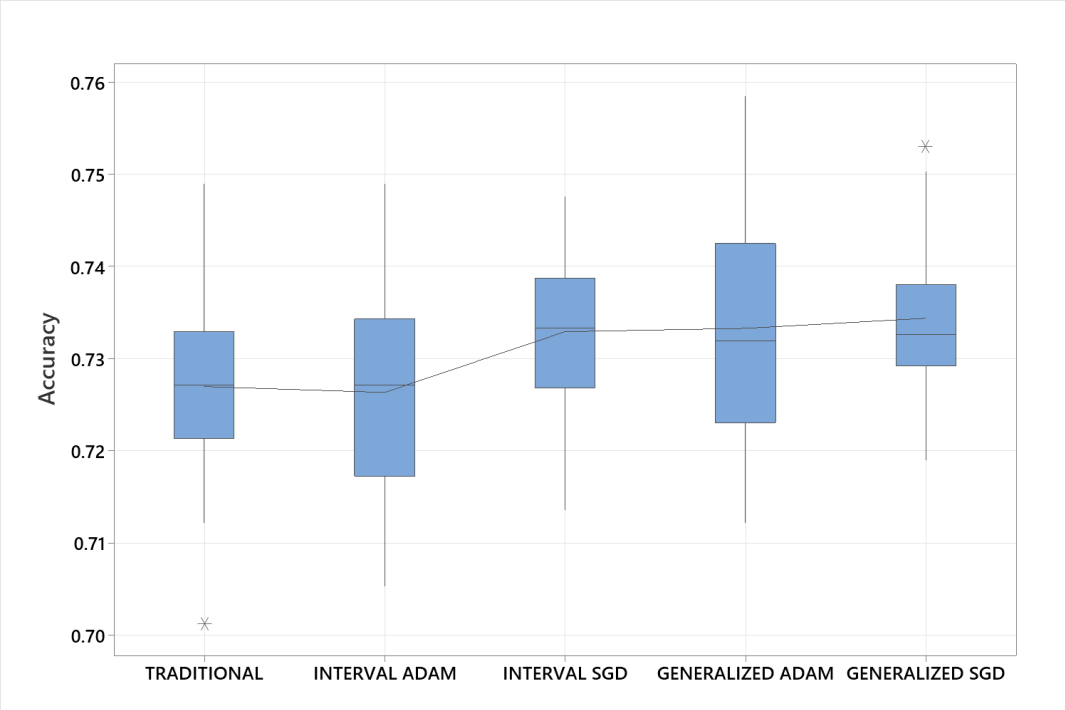

Fig. 3. Boxplot of classification experiments

method using GT2 offers better results than the proposed method using IT2.

Finally, a hypothesis test was performed comparing the best results obtained according to the previous hypothesis tests. The best results were obtained by the proposed method using IT2 in SGD against the proposed method using IT2 in Adam.

The result of the test gave a z-value of 2.70943556, concluding that the null hypothesis was rejected, with sufficient evidence to support our claim that the proposed method using IT2 applied to SGD offers better results than the same implementation in Adam.

6 Conclusions

In this present study, the focus was on using four FISs: Mamdani IT2 and GT2 to determine the learning rate for the Adam and SGD training algorithms, and momentum only for SGD. Prior to implementing the proposed method, MA and SD of the base CNN model were calculated for comparative purposes. Subsequently, the proposed methods were incorporated into two different pre-existing CNN models.

After implementing the proposed method in the CNN models, calculations were performed to obtain the SD, MA, and MxA values. Hypothesis

tests were then performed, concluding that the implementation of the proposed method using IT2 FIS yields the best results compared to the other experiments.

It should be noted that there is still room for improvement in the design of the FIS, particularly in aspects such as the number of variables, the number of rules, and the membership functions used. Therefore, as future work, we propose implementing these improvements based on the present work, with the goal of increasing mean accuracy and reducing the standard deviation.

Additionally, according to related work, there is a possibility of implementing the proposed method in the RMSProp training algorithm by updating the FIS inputs and/or outputs to improve the results obtained by Adam and SGD.

Finally, the APTOS 2019 database is established as a valuable resource for real DR studies; however, it is not the only disease that doctors are concerned about detecting in time and the proposed method could be successfully applied in more medical datasets.

Acknowledgments

We would like to express our gratitude to the SECIHTI, Tijuana Institute of Technology and TecnNM for the facilities and resources granted for the development of this research.

References

1. **Alateeq, M., Pedrycz, W. (2024).** Logic-oriented fuzzy neural networks: A survey. Expert Systems with Applications, Vol. 257, pp. 125120.
2. **Arun, M., Efremov, C., Nguyen, V. N., Barik, D., Sharma, P., Bora, B. J., Kowalski, J., Le, H. C., Truong, T. H., Cao, D. N. (2024).** Fuzzy logic-supported building design for low-energy consumption in urban environments. Case Studies in Thermal Engineering, Vol. 64, pp. 105384.
3. **Aruna Kumari, A., Bhagat, A., Kumar Henge, S. (2024).** Classification of diabetic retinopathy severity using deep learning techniques on retinal images. Cybernetics and Systems, pp. 1–25.
4. **Barroso, E., Jurado-Aguilar, J., Wahli, W., Palomer, X., Vázquez-Carrera, M. (2024).** Increased hepatic gluconeogenesis and type 2 diabetes mellitus. Trends in Endocrinology and Metabolism, Vol. 35, No. 12, pp. 1062–1077.
5. **Bashir, T., Wang, H., Tahir, M., Zhang, Y. (2025).** Wind and solar power forecasting based on hybrid cnn-abilstm, cnn-transformer-mlp models. Renewable Energy, Vol. 239, pp. 122055.
6. **Castillo, O., Melin, P., Valdez, F., Gonzalez, C., Garcia, M., Mancilla, A., Cortes-Antonio, P., J., Soria (2025).** A review on the role of fuzzy logic in hybrid intelligent systems. Computación y Sistemas, Vol. 29, No. 3, pp. 1723–1740.
7. **Chandrasekaran, P., Weiskirchen, R. (2024).** The role of obesity in type 2 diabetes mellitus—an overview. International Journal of Molecular Sciences, Vol. 25, No. 3, pp. 1882.
8. **Chen, C., Mat Isa, N. A., Liu, X. (2025).** A review of convolutional neural network based methods for medical image classification. Computers in Biology and Medicine, Vol. 185, pp. 109507.
9. **Chong, D. D., Das, N., Singh, R. P. (2024).** Diabetic retinopathy: Screening, prevention and treatment. Cleveland Clinic Journal of Medicine, Vol. 91, No. 8, pp. 503–510.
10. **Chuquimarca, L. E., Vintimilla, B. X., Velastin, S. A. (2024).** A review of external quality inspection for fruit grading using cnn models. Artificial Intelligence in Agriculture, Vol. 14, pp. 1–20.
11. **Cordero-Martínez, R., Sánchez, D., Castillo, O., Melin, P. (2025).** Estimation of filter number for convolutional neural networks with fuzzy logic for diabetic retinopathy classification. In Hybrid Intelligent Systems. HIS 2023. Lecture Notes in Networks and Systems, volume 1223. Springer, pp. 90–100.
12. **Cordero-Martínez, R., Sánchez, D., Melin, P. (2022).** Hierarchical genetic optimization of convolutional neural models for diabetic retinopathy classification. International Journal of Hybrid Intelligent Systems, Vol. 18, No. 1–2, pp. 97–109.
13. **Cordero-Martínez, R., Sánchez, D., Melin, P. (2023).** Optimizing a convolutional neural network with a hierarchical genetic algorithm for diabetic retinopathy detection. In Studies in Computational Intelligence, volume 1061. Springer, pp. 199–208.

14. **Flindris, K., Lagkada, V., Christodoulou, A., Gazouli, M., Moschos, M., Markozannes, G., Kitsos, G. (2025).** Investigation of uchl3 and hnmt gene polymorphisms in greek patients with type 2 diabetes mellitus and diabetic retinopathy. Biomedicines, Vol. 13, No. 2, pp. 341.

15. **Gonzalez, C. I. (2025).** Designing optimal cnns architectures using metaheuristic algorithms applied to the classification of alzheimer's disease. Computación y Sistemas, Vol. 29, No. 1, pp. 179–189.

16. **Haq, N. U., Waheed, T., Ishaq, K., Hassan, M. A., Safie, N., Elias, N. F., Shoaib, M. (2024).** Computationally efficient deep learning models for diabetic retinopathy detection: a systematic literature review. Artificial Intelligence Review, Vol. 57, No. 11, pp. 1–44.

17. **Hermosilla, P., Soto, R., Vega, E., Suazo, C., Ponce, J. (2024).** Skin Cancer Detection and Classification using neural network algorithms: A systematic review. Diagnostics, Vol. 14, No. 4, pp. 454.

18. **Hidalgo, D., Cervantes, L., Castillo, O., Melin, P., Soto, R. M. (2020).** Fuzzy parameter adaptation in genetic algorithms for the optimization of fuzzy integrators in modular neural networks for multimodal biometry. Computación y Sistemas, Vol. 24, No. 3, pp. 1093–1105.

19. **Jiang, K., Wang, Q., An, Z., Wang, Z., Zhang, C., Lin, C. W. (2024).** Mutual retinex: Combining transformer and cnn for image enhancement. IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 8, No. 3, pp. 2240–2252.

20. **Koklu, A., Guven, Y., Kumbasar, T. (2025).** Odyssey of interval type-2 fuzzy logic systems: Learning strategies for uncertainty quantification. IEEE Transactions on Fuzzy Systems, Vol. 33, pp. 468–478.

21. **Kommaraju, R., Anbarasi, M. S. (2024).** Diabetic retinopathy detection using convolutional neural network with residual blocks. Biomedical Signal Processing and Control, Vol. 87, pp. 105494.

22. **Kurniadi, D., Mulyani, A., Rafliana, R., Suwandy, M. (2024).** Eye disease detection model using convolutional neural network with rmsprop, sgd and adam optimizers. 11th International Conference on ICT for Smart Society (ICISS 2024).

23. **Köklü, A., Güven, Y., Kumbasar, T. (2024).** Enhancing interval type-2 fuzzy logic systems: Learning for precision and prediction intervals. IEEE International Conference on Fuzzy Systems, pp. 1–7.

24. **Li, H., Ju, W., Song, Y., Cao, Y., Yang, W., Li, M. (2024).** Soil organic matter content prediction based on two-branch convolutional neural network combining image and spectral features. Computers and Electronics in Agriculture, Vol. 217, pp. 108561.

25. **Liang, Y., Zhang, X., Mei, W., Li, Y., Du, Z., Wang, Y., Huang, Y., Zeng, X., Lai, C., Wang, S., Fang, Y., Zhang, F., Zang, S., Sun, W., Yu, H., Hu, Y. (2024).** Predicting vision-threatening diabetic retinopathy in patients with type 2 diabetes mellitus: Systematic review, meta-analysis and prospective validation study. Journal of Global Health, Vol. 14.

26. **Liu, S., Wang, L., Yue, W. (2024).** An efficient medical image classification network based on multi-branch cnn, token grouping transformer and mixer mlp. Applied Soft Computing, Vol. 153, pp. 111323.

27. **Long, F., Xiong, H., Sang, J. (2024).** A classification method for diabetic retinopathy based on self-supervised learning. In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 14881. Springer, pp. 347–357.

28. **Lowe, J., Kuru, K. (2024).** Design & development of a smart blind system using fuzzy logic. MESA 2024 - 20th International Conference on Mechatronic, Embedded Systems and Applications, Proceedings, pp. 1–8.

29. **Lu, X., Xie, Q., Pan, X., Zhang, R., Zhang, X., Peng, G., Zhang, Y., Shen, S., Tong, N. (2024).** Type 2 diabetes mellitus in adults: pathogenesis, prevention and therapy. Signal Transduction and Targeted Therapy, Vol. 9, No. 1, pp. 1–25.

30. **López-Miguel, N., Diaz-Hernández, R., Altamirano-Robles, L. (2025).** Confidence calibration of cnns in medical image databases. Computación y Sistemas, Vol. 29, No. 1, pp. 7–14.

31. **Małolepsza, O., Mikołajewski, D., Prokopowicz, P. (2025).** Using fuzzy logic to analyse weather conditions. Electronics, Vol. 14, pp. 85.

32. **Mendel, J. M. (2024).** General type-2 fuzzy systems. In Explainable Uncertain Rule-Based Fuzzy Systems. Springer, pp. 519–568.

33. **Mlynarska, E., Czarnik, W., Dzieza, N., Jedraszak, W., Majchrowicz, G., Prusinowski, F., Stabrawa, M., Rysz, J., Franczyk, B. (2025).** Type 2 diabetes mellitus: New pathogenetic mechanisms, treatment and the most important complications. International Journal of Molecular Sciences, Vol. 26, No. 3, pp. 1094.

34. **Poma, Y., Melin, P. (2022).** Adaptation of number of filters in the convolution layer of a convolutional neural network using the fuzzy gravitational search algorithm method and type-1 fuzzy logic. Computación y Sistemas, Vol. 26, No. 2, pp. 511–535.

35. **Pulido, M., Melin, P. (2024).** Bird swarm algorithm and particle swarm optimization in ensemble recurrent neural networks optimization for time series prediction. Computación y Sistemas, Vol. 28, No. 2, pp. 847–859.

36. **Ramírez, M., Melin, P., Castillo, O. (2024).** A new hybrid approach for clustering, classification and prediction of world development indicators combining general type-2 fuzzy systems and neural networks. Axioms, Vol. 13, No. 6, pp. 368.

37. **Roy, S., Jana, D. K., Mishra, A. (2024).** Linguistic interval type 2 fuzzy logic-based exigency vehicle routing: Iot system development for smart city applications with soft computing-based optimization. Franklin Open, Vol. 6, pp. 100057.

38. **Saatchi, R. (2024).** Fuzzy logic concepts, developments and implementation. Information, Vol. 15, No. 10, pp. 656.

39. **Sadek, N. A., Al-Dahan, Z. T., Rattan, S. A. (2024).** Comprehensive survey of the state-of-the-art deep learning models for diabetic retinopathy detection and grading using retinal fundus photography. Al-Nahrain Journal for Engineering Sciences, Vol. 27, No. 2, pp. 155–163.

40. **Sakka, E., M., M., Ivanovici, Chaari, L., Mothe, J. (2025).** A review of cnn applications in smart agriculture using multimodal data. Sensors, Vol. 25, No. 2, pp. 472.

41. **Sarwo (2024).** Improving offline handwritten text recognition accuracy with adam and sgd optimizers and convolutional neural networks models. 2024 International Conference on Intelligent Cybernetics Technology and Applications (ICICyTA), pp. 1135–1139.

42. **Tian, C., Zheng, M., Jiao, T., Zuo, W., Zhang, Y., Lin, C. W. (2024).** A self-supervised cnn for image watermark removal. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 34, No. 8, pp. 7566–7576.

43. **Trach, Y., Trach, R., Kuznietsov, P., Pryshchepa, A., Biedunkova, O., Kiersnowska, A., Statnyk, I. (2024).** Predicting the influence of ammonium toxicity levels in water using fuzzy logic and ann models. Sustainability, Vol. 16, No. 14, pp. 5835.

44. **Ullah, K., Ahsan, M., Hasanat, S. M., Haris, M., Yousaf, H., Raza, S. F., Tandon, R., Abid, S., Ullah, Z. (2024).** Short-term load forecasting: A comprehensive review and simulation study with cnn-lstm hybrids approach. IEEE Access, Vol. 12, pp. 111858–111881.

45. **Xie, X., Cheng, G., Wang, J., Li, K., Yao, X., Han, J. (2024).** Oriented r-cnn and beyond. International Journal of Computer Vision, Vol. 132, No. 7, pp. 2420–2442.

46. **Zainab, H., Raza, A., Khan, A., Khan, M. I., Arif, A. (2025).** Innovative ai solutions for mental health: Bridging detection and therapy. Global Journal of Emerging AI and Computing, Vol. 1, No. 1, pp. 51–58.

Article received on 11/08/2025; accepted on 27/09/2025.
**Corresponding author is Patricia Melin.*