www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(4), 2026, 152-170. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i4.1486

Type-1 and Type-2 Fuzzy Parameter Adaptation in PSO for CNN Architecture Optimization Applied to Facial Emotion Recognition

Alexis Campos [0000-0003-4373-4226], Patricia Melin [0000-0001-5798-1426]*, Daniela Sánchez [0000-0001-9802-1907]
Tijuana Institute of Technology/TecNM, Tijuana, Mexico
D24210008@tectijuana.edu.mx, pmelin@tectijuana.mx*, daniela.sanchez@tectijuana.edu.mx
✉Corresponding author: pmelin@tectijuana.mx

Abstract. It is known that through the use of Convolutional Neural Networks (CNNs) classification problems can be solved, such as facial emotion classification. Fuzzy logic has allowed to reduce the uncertainty that may exist when making a decision. In our work, a convolutional neural network model is proposed that allows, through a fuzzy Mamdani inference system, to choose the optimal hyperparameters within Particle Swarm Optimization (PSO), adjusting variables such as inertial weight and cognitive and social acceleration coefficients, in order to determine the number of convolutional layers and filters to be used within the CNN model. We evaluate Type-1 and interval Type-2 fuzzy adaptation within an identical search space and training setup, focusing on how the treatment of uncertainty shapes the quality of the resulting architectures. The experimental results allow an evaluation of the proposed hybrid system for facial emotion classification using the FER2013 dataset.
Keywords: Convolutional neural network, particle swarm optimization, facial emotion recognition, fuzzy logic, type-1, type-2.

Article information

Received: Jun 26, 2026
Accepted: Jul 11, 2026

1 Introduction

Emotional states are complex to determine because they involve multiple interacting factors, people often experience multiple emotions simultaneously, making it difficult to isolate individual affective states. Verbal communication includes several methods like conversation with a person, phone calls and letters. For each of these ways to communicate something has different characteristics, but all enable immediate or almost instantaneous feedback between speakers, the response is important for emotional expression and interpretation (Izard, 1991). Non-verbal communication could include body poses, vocal, facial expressions, and gesticulation.

Facial expressions are important for sharing emotion in interactions, these kinds of movements occur spontaneously without voluntary regulation, ensuring they are credible reflections of emotional state, therefore facial expression analysis is a topic in affective computing research (Sauter, 2017).

FER under unconstrained conditions remains challenging because facial appearance is often affected by occlusions illumination changes head pose variation and low image resolution. These factors increase variability in the visual data and make learned representations less reliable while dataset properties add further difficulty. In FER2013, for example, the distribution of emotion classes is uneven, which can bias training and lead to unstable validation performance when different network architectures are compared.

This work investigates a hybrid intelligent framework in which PSO is used to explore convolutional neural network architectural hyperparameters, including the number of convolutional layers and the number of filters per layer. At the same time, a Mamdani fuzzy inference system adaptively adjusts key PSO parameters, namely the inertia weight and acceleration coefficients. The motivation is practical: evaluating CNN architecture is noisy and computationally expensive, and fixed PSO parameters may not provide a good balance between exploration and exploitation throughout the search process. Adaptive control is therefore expected to stabilize the search behavior under uncertain fitness evaluations. We also examine the hypothesis that Type-2 fuzzy logic, by explicitly modeling uncertainty in the inference process, can manage this variability more effectively than a conventional Type-1 fuzzy system.

© Editorial Académica Dragón Azteca S. de R.L. de C.V. (EDITADA.ORG). All rights reserved.

Most related studies either fix PSO parameters throughout the search or introduce fuzzy adaptation without a direct comparison between Type-1 and Type-2 formulations within the same FER-driven architecture search setting. As a result, it remains unclear whether the added complexity of Type-2 fuzzy systems leads to measurable benefits under identical experimental conditions. To address this gap, this work evaluates both formulations within a single and consistent experimental pipeline.

The contribution is threefold. First, we propose a PSO-based CNN architecture search in which parameter adaptation is governed by a fuzzy inference system that uses indicators of search progress and fitness improvement. Second, we conduct a controlled comparison between Type-1 and Interval Type-2 fuzzy systems while keeping the search space, training protocol, and evaluation criteria fixed. Third, we report statistical and computational results from 30 independent runs, which makes it possible to examine how uncertainty modeling influences classification accuracy, loss behavior, and training efficiency when fitness evaluations are noisy.

The paper is organized along the same lines. Section 2 reviews the background on convolutional neural networks particle swarm optimization facial emotion recognition and fuzzy logic. Section 3 places the proposed approach within existing work and Section 4 describes the methodology with a focus on fuzzy based parameter adaptation. Section 5 presents the experimental results together with statistical validation. Section 6 concludes the paper and outlines directions for future research.

2 Background

This section outlines the main components that underpin the proposed approach. We first review the basic principles of convolutional neural networks, which serve as the core modeling framework for facial emotion recognition. We then introduce particle swarm optimization and the fuzzy logic concepts required to motivate the adaptive Mamdani fuzzy system used to adjust PSO parameters during the architecture search. Both Type-1 and Type-2 formulations are described at a level sufficient to support the design choices made in the following sections.

2.1 Convolutional Neural Network

Convolutional neural networks are deep learning models designed for grid-structured data such as images. They learn feature representations through sequences of convolutional layers that apply trainable filters to the input, producing feature maps that are typically followed by nonlinear activation functions and, in some cases, pooling or down-sampling operations. This layered structure supports hierarchical representation learning. Early layers usually capture simple local patterns such as edges or textures while deeper layers combine these signals into more abstract facial structures that are useful for recognition. Architectural choices play a central role in CNN performance. Network depth defined by the number of convolutional layers and network width defined by the number of filters per layer directly affect representational capacity generalization and computational cost. These considerations are especially important for FER2013 where images are low resolution at 48×48 and emotion classes are imbalanced. Models with too much capacity tend to overfit the limited visual detail while shallow or narrow architectures may fail to capture subtle facial cues. As a result, CNN effectiveness in facial emotion recognition, as in other computer vision tasks, depends strongly on architectural decisions related to layer configuration, filter allocation, and overall connectivity (N. Sharma et al., 2018). When architectures are designed using random parameters, they necessitate substantial expertise and experimentation with some metaheuristic optimization algorithms allied with fuzzy logic that can automate this process (Papic et al., 2025).

2.2 Particle Swarm Optimization

Inspired by the collective behavior seen in bird flock's particle swarm optimization offers clear advantages over genetic algorithms. It usually relies on fewer control parameters and tends to converge faster in continuous search spaces. Its population based structure also makes parallel evaluation of candidate solutions straightforward which is especially useful when fitness evaluation is computationally expensive (Marini & Walczak, 2015), but his performance mostly depends on three key control parameters, which are particularly the inertia weight (ω), cognitive coefficient (c_1), and social coefficient (c_2) (Bansal et al., 2011).

(Kennedy & Eberhart, 1995) introduced PSO in 1995, drawing inspiration from the social behavior of bird flocking and fish schooling. The original formulation relied on constant acceleration coefficients and did not include an inertia weight, which often resulted in unstable swarm dynamics. (Shi & Eberhart, 1999) introduced the inertia weight ω to balance exploration and exploitation, as we can see in Equation 1.

$$v_i(t+1) = \omega \cdot v_i(t) \; + \; c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) \; + \; c_2 \cdot r_2 \cdot (gbest - x_i(t)) \quad (1)$$

In this equation, the current velocity $v_i(t)$ is the particle's movement at time, which is capturing both the direction and the magnitude of its displacement. The inertia term $\omega \cdot v_i(t)$acts as a continuity mechanism, allowing the particle to keep its previous trajectory and encouraging exploration of the search space. The component $c_1 \cdot r_1 \cdot (p_{\text{best}_i} - x_i(t))$ this is effect of

individual memory, guiding the particle toward the best position it has previously determined. The term $c_2 \cdot r_2 \cdot (g_{\text{best}} - x_i(t))$ is the social aspect of the algorithm, attracting the particle toward the best global solution discovered by the swarm. The interaction of these three factors, individual experience and collective influence, produces the new velocity $v_i(t+1)$, that balances exploration of new regions with exploitation of promising solutions, thus enhancing the effectiveness of the optimization process (Veiga et al., 2022).

$$\omega(t) = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) \cdot \frac{t}{T_{\text{max}}} \tag{2}$$

In Equation 2, ω_{max} is representing the initial maximum value of the inertia weight, while ω_{min} denotes the minimum value reached at the end of the process. The variable t is the current iteration, and T_{max} correspond to the total number of iterations. This linear decline starts with high inertia weight to encourage exploration of the search space in early iterations, then gradually decrease it to favor exploitation and convergence to promising regions, this schedule balances exploration and exploitation, helping to avoid early convergence and facilitating the discovery of optimal solutions (Shi & Eberhart, 1998).

PSO behavior is specified by two tendencies, exploration this search for diverse regions of the solution space to identify the global optimum, the process is driven by a high inertia weight, which keeps velocity diversity, and requires a careful balance in the cognitive and social parameters. In contrast exploitation focuses on refining solutions in promising regions to speed up convergence toward optimal results. It is encouraged by a low inertia weight which reduces particle velocity and by a high social parameter that pulls the swarm toward shared best solutions.

2.3 Facial Emotion Recognition

The FER2013 dataset (Zahara et al., 2020) was introduced as part of a Kaggle challenge focused on facial expression recognition in unconstrained environments, which consists of 35,887 grayscale images with a resolution of 48 × 48 pixels and covers seven emotion categories which are angry, disgust, fear, happy, sad, surprise, and neutral. The dataset shows a clear difference in class sizes with the happy category having about sixteen times more samples than disgust. The images are low resolution which limits fine facial detail and data collected in real settings also includes occlusions lighting changes and pose variation, open contribution process for emotion labeling may include inconsistencies or labeling errors. All these factors make FER2013 a challenging benchmark for studying optimization methods in facial emotion recognition.

2.4 Facial Emotion Recognition

Fuzzy logic offers a practical way to deal with uncertainty and to support adaptive control in optimization problems (Berenji, 1992), In Type-1 fuzzy systems PSO parameters are adjusted using linguistic rules that reflect search progress and changes in fitness. These systems assume that membership functions are defined exactly. In practice this assumption can fail when the fitness signal is noisy when expert knowledge is inconsistent or when measurements are uncertain (Mendel, 2024).

Interval Type-2 fuzzy logic systems extend Type-1 systems by allowing uncertainty in the membership functions through a Footprint of Uncertainty. Instead of assigning a single membership value to an input, IT2FLS represent it as a range (Castillo & Melin, 2014). This makes it possible to reflect ambiguity in how inputs are interpreted and to reason under less precise conditions. As a result, IT2FLS can provide more stable behavior in uncertain environments and have been applied to control problems such as robotics and power systems (Karnik & Mendel, 1999), its application to metaheuristic optimization for deep neural network architecture search remains largely unexplored (Mittal et al., 2020).

IT2FLS extends Type-1 by adding a FOU in membership functions (Castillo et al., 2007), in these systems, each input is like a parameter to an interval $[\mu_L(x), \mu_U(x)]$ rather than a single membership value $\mu(x)$, providing a representation of uncertainty. This approach allows uncertainty to be represented more naturally within the system. Linguistic uncertainty appears because different experts may understand terms like high or low in slightly different ways. Input values can also be affected by measurement noise since observations often vary due to random effects. Uncertainty may further arise at the rule level because not all rules are supported with the same degree of confidence. By accounting for these sources of imprecision, IT2FLS provide a more flexible way to handle uncertainty in complex environments (Wu, 2013).

3 Related Works

This section reviews prior work in four areas that are central to this study including CNN hyperparameter optimization particle swarm optimization variants the use of fuzzy logic in swarm-based methods and facial expression recognition. Within this literature, we identify gaps that motivate a direct comparison between Type-1 and Type-2 fuzzy logic for adaptive control of PSO in an FER architecture search setting.

Early CNNs like AlexNet (Chen et al., 2022), VGGNet (He et al., 2018), and ResNet (Kalaivani et al., 2024), (Liao et al., 2025) were manually designed through iterative experimentation. This approach is prolonged, demands expertise, and often generates suboptimal designs constrained by human cognitive limitations.

Grid search evaluates every possible hyperparameter combination, while random search samples combinations at random. In the context of CNN architecture studies, where a single evaluation can require hours of training, this quickly becomes impractical. (Bergstra & Bengio, 2012) demonstrated that random search can outperform grid search when only a small number of hyperparameters truly matter, as it samples more unique values along critical dimensions. However, both approaches waste computational resources evaluating poor-performing regions of the search space (Schröder et al., 2025).

Genetic algorithms (GA) have been widely applied to CNN optimization. Naaman et al. (Waysi Naaman et al., 2025) for example, used a GA to evolve both network architectures and hyperparameters. At the same time, GAs introduces additional control parameters, such as crossover rate, mutation rate, and selection pressure, which also require tuning. The discrete nature of genetic operators can further cause abrupt changes in network structure, making the search process less stable.

PSO has emerged as a promising alternative because it requires fewer control parameters and naturally supports continuous search spaces (R. Sharma et al., 2026). (Almansour et al., 2025) applied PSO to optimize CNN hyperparameters, reporting faster convergence than GA. (Raza et al., 2024) used PSO to simultaneously optimize architecture and data augmentation strategies for medical image classification.

In (Elhawat & Altınkaya, 2025), a PSO-based fuzzy PID controller was proposed for frequency regulation in stand-alone synchronous generators. In real-time experiments, it outperformed conventional PLC PID and fuzzy PID controllers by reducing overshoot, undershoot, and settling time. (Shirani Shamsabadi et al., 2025), fuzzy logic was used to model mastitis in dairy cattle by comparing three ANFIS classifiers.

A fuzzy system was proposed in (Niknam et al., 2009) to adaptively adjust PSO parameters (ω, c_1, c_2) based on the progress of the optimization process and observed improvements in fitness fuzzy rules were used to adjust PSO parameters. This led to better clustering performance in power distribution networks and shorter convergence time compared with standard PSO.

(Alfi & Fateh, 2011) proposed a fuzzy PSO controller that used iteration progress success rate and swarm diversity as inputs. With Gaussian membership functions the controller adapted PSO behavior and improved tracking accuracy in robotic manipulators compared with fixed parameter PSO.

(Castillo et al., 2016) compared Type-1 Interval Type-2 and Generalized Type-2 fuzzy logic systems for complex nonlinear plants. Using alpha-plane theory and the Karnik–Mendel algorithm their simulations showed that Generalized Type-2 systems were more robust and efficient under different noise levels and membership function settings.

Several FER papers report performance improvements when models learn stronger representations. Similar trends have been reported in transformer-based and zero-shot methods (Pazandeh & Fatemizadeh, 2025), (Bhati et al., 2025), as well as in studies focused on multi-view data and micro-expression analysis (Behzad, 2025),(Alwan, 2025). A similar pattern appears in vision-language pretraining studies (Khan et al., 2025). Together, these results suggest that current FER research is moving toward more adaptive representations to better handle high intra-class variability.

From the fuzzy-logic perspective, the context is also relevant. The Python Interval Type-2 Fuzzy Logic Systems framework (PyIT2FLS) (Haghrah et al., 2025) has made Type-1 and Interval Type-2 experimentation more practical and reproducible, while recent studies continue advancing uncertainty-aware fuzzy modeling (Mohammadi Lanbaran et al., 2025).

4 Proposed Methodology

This section shows our fuzzy logic-based adaptive PSO framework for optimizing CNN architectures. It outlines the overall system design, the fuzzy adapter implementation for both Type-1 and Type-2 variants, the definition of the CNN model and key implementation points.

The framework comprises four components, as illustrated in Fig. 1. First, the Particle Population consists of N candidate solutions exploring the CNN architecture search space in parallel. Each particle is a six-dimensional vector defining the number of convolutional layers and filters per layer. After that the CNN Evaluation module constructs each architecture, trains it on FER2013, and determines validation accuracy. This module defines the fitness function for the search process. The PSO Update phase, updates particle velocities and positions using standard Particle Swarm Optimization equations based on their personal best solutions (p_{best}) and the global best solution (g_{best}). These components form a conventional PSO system; however, our main contribution lies in the Fuzzy System for parameter adaptation. This fuzzy system tracks the state of optimization and adjusts PSO parameters. (ω, c_1, c_2) to balance exploration and exploitation. In each iteration, adaptation is computed from the current search state before applying the velocity-position update with the adapted parameters.

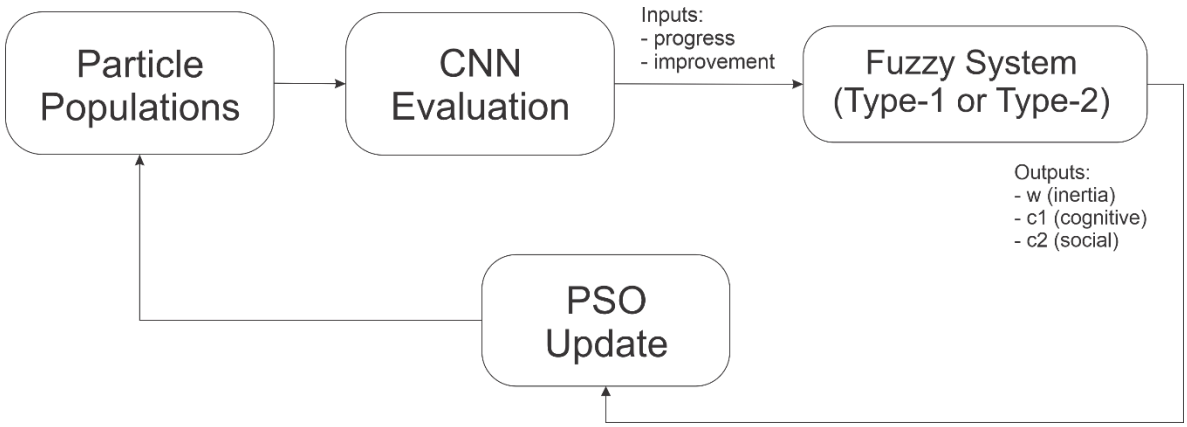

Fig. 1 PSO-CNN framework with fuzzy adaptation.

The system works through a recurring loop in which components shared specific information to coordinate the search. For each iteration, the particles, which represent encoded CNN architectures, move from the Population to the Evaluator through the vector that contains the architectural hyperparameters. The Evaluator returns a scalar value (negative fitness, because PSO minimizes) that reflects the architecture's effectiveness. This fitness information updates the swarm memory in the PSO Engine, and it is also provided to the fuzzy system together with additional contextual data. Specifically, the fuzzy system processes two input signals first one is progress, which indicates how far the search has advanced in terms of iterations, and the second one is improvement, which measures the relative change in the best fitness compared with the previous iteration, in these signals transmit temporal and qualitative data. The fuzzy system uses membership functions and linguistic rules on these inputs to produce three outputs these are ω (inertia weight), c_1 (cognitive coefficient), and c_2 (social coefficient). The inertia weight governs the degree to which particles keep their past motion, c_1 determines attraction to personal best, and c_2 governs attraction to global best. The adapted parameters return to the PSO Engine, modifying velocity updates to balance exploration in new regions with exploitation of promising solutions, as we can see in Fig. 2.

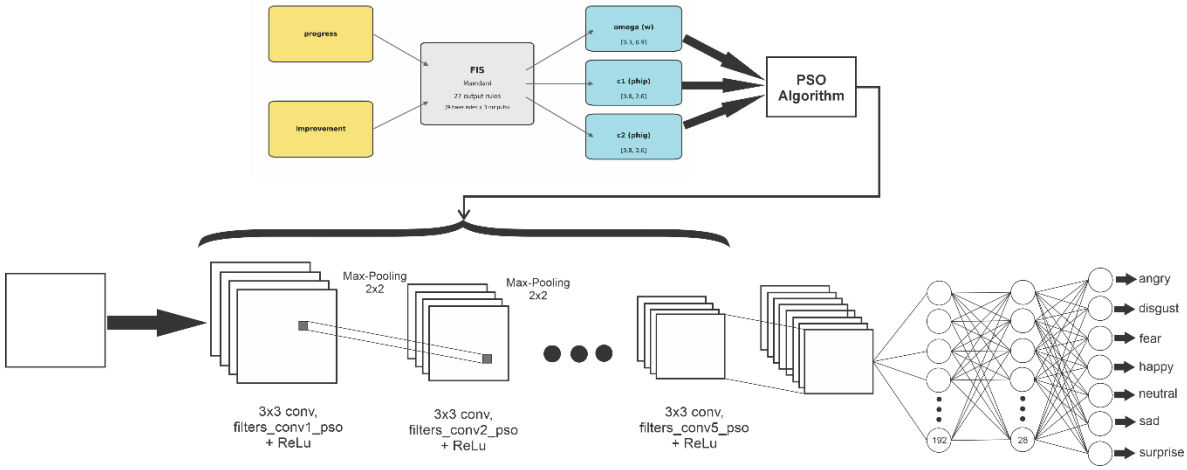

Fig. 2 Data flow for one PSO iteration with fuzzy adaptation.

4.1 Input Variables

The fuzzy adaptive system receives two input variables that summarize the state of the optimization process. The first one captures the temporal stage of the search (how far the algorithm has progressed), while the second one captures the quality of recent progress (how much the best fitness has improved). Together, these inputs allow the fuzzy system to decide whether to encourage exploration (diversification) or exploitation (refinement) by adjusting the PSO parameters.

4.1.1 Search Progress

The progress variable provides temporal context and is defined as progress(t) = t / $T_{\max}$ in the domain from 0.0 to 1.0 where t represent the current iteration and $T_{\max}$ is the total number of iterations allocated to the search. For Type-1, these regions are triangular sets with smooth overlap, as we can see in Fig. 3.

The progress variable provides temporal context and is defined as progress(t) = t / T_{max} in the domain from 0.0 to 1.0 where t represent the current iteration and T_{max} is the total number of iterations allocated to the search. For Type-1, these regions are triangular sets with smooth overlap, as we can see in Fig. 3.

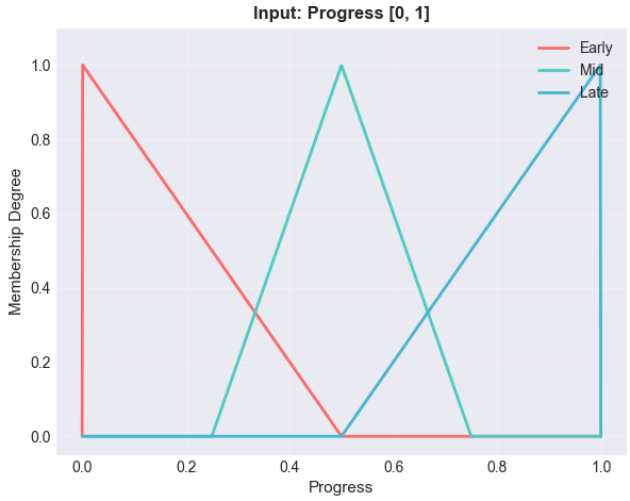

Fig. 3 Type-1 fuzzy membership functions for the variable "Progress"

We model three overlapping linguistic regions: Early, Mid, and Late, which correspond to exploration, transition, and refinement stages.

For the Type-2 case each set is expanded into an interval Type-2 form which creates a FOU of about ten percent of the support width. This is used to represent uncertainty in phase boundaries as shown in Fig. 4.

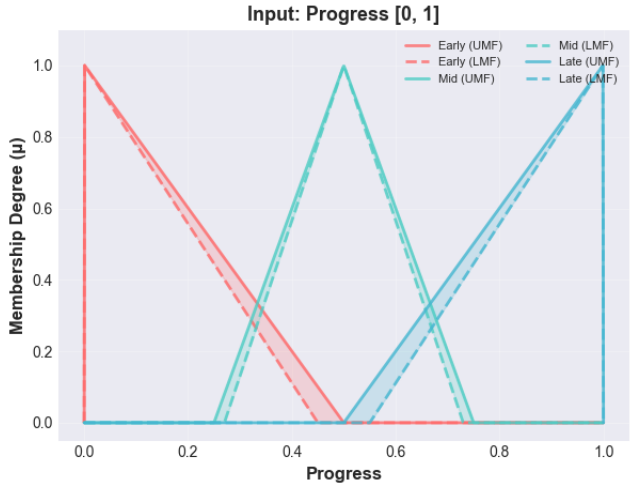

Fig. 4 Type-2 fuzzy membership functions for the variable "Progress"

4.1.2 Fitness Improvement

The second input is basically how the global best fitness changes and gives a sense of solution quality without depending on the number of iterations. We use three linguistic terms Bad, Small, and Good. In practice this input helps tell apart cases where the search is stuck from those showing limited progress or clear improvement. Type-1 uses triangular sets as shown in Fig. 5.

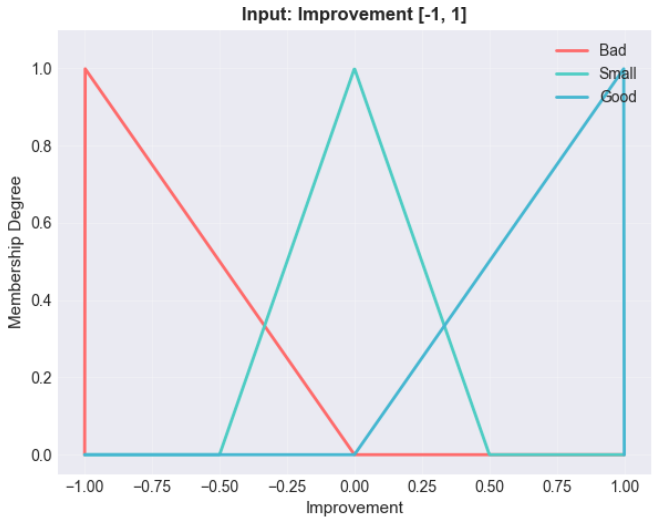

Fig. 5 Type-1 fuzzy membership functions for the variable "Improvement"

Type-2 follows the same structure but adds UMF and LMF pairs with a narrower lower support which creates an FOU of about ten percent to account for noisy improvements as illustrated in Fig. 6.

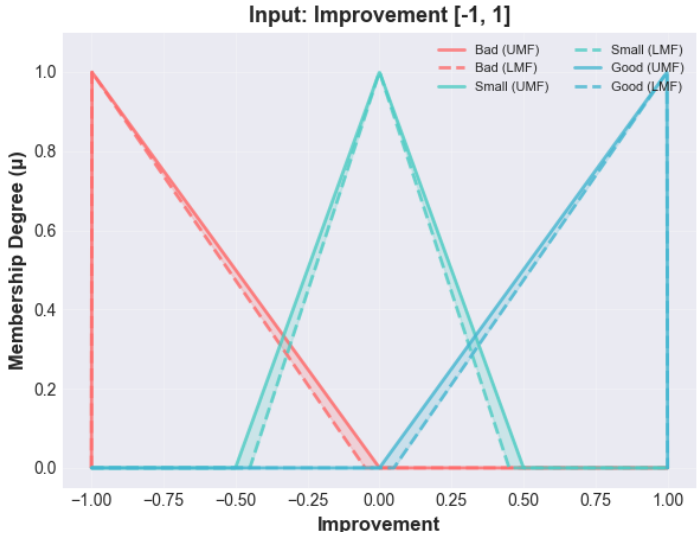

Fig. 6 Type-2 fuzzy membership functions for the variable "Improvement".

4.2 Output Variables

The fuzzy system produces three PSO parameters the inertia weight the cognitive coefficient and the social coefficient. Together they shape how particles keep momentum learn from their own experience and follow the swarm which allows the search to shift gradually from exploration toward exploitation.

4.2.1 Inertia Weight

Inertia weight controls keeps persistence, for high values favor broad search, and low values favor local refinement, is restrict ω from 0.3 to 0.9 and model it with three overlapping linguistic terms (Low, Med, High).

Type-1 uses triangular sets, and Type-2 introduces UMF/LMF pairs to represent uncertainty before type reduction, as we can see in Fig. 7 and Fig. 8.

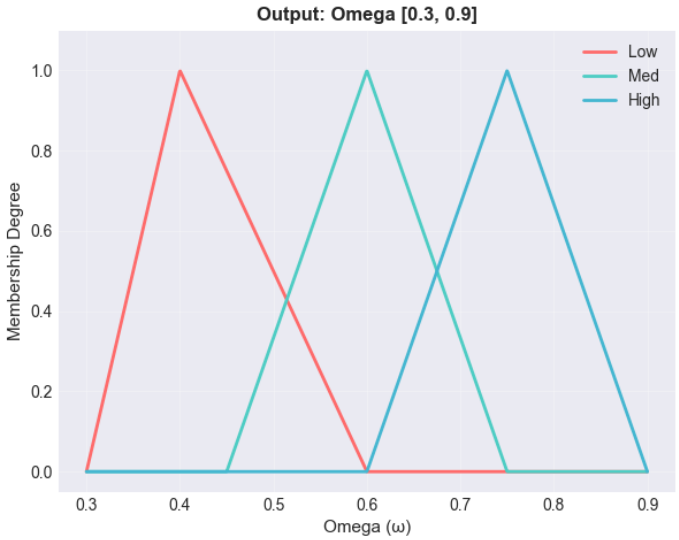

Fig. 7 Type-1 fuzzy membership functions for the output variable "Omega (ω)".

Type-2 uses the same supports with slightly contracted lower functions to define FOUs, yielding interval outputs before type reduction.

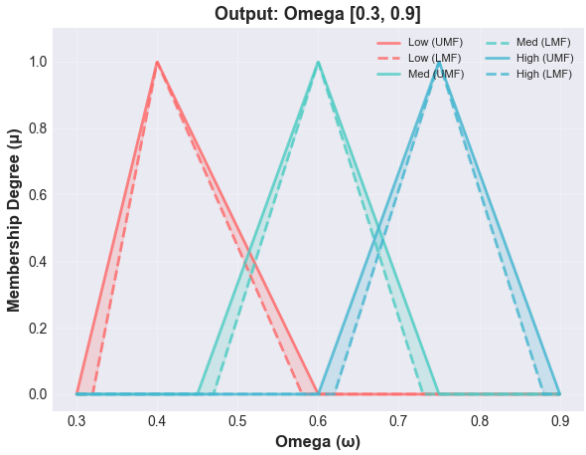

Fig. 8 Type-2 fuzzy membership functions for the output variable "Omega (ω)".

4.2.2 Cognitive and Social Coefficients (c_1, c_2)

The coefficients c_1 and c_2 are balancing individual versus collective attraction. Higher c_1 promotes independent search around personal bests, whereas higher c_2 strengthens convergence toward the global best, both coefficients are modeled from 0.8 to 2.6 with three overlapping terms (Low, Med, High), allowing a more balanced adjustment between individual autonomy and swarm cohesion.

The Type-1 fuzzy membership functions for the cognitive and social coefficients (c_1 and c_2) are illustrated in Figs. 9 and 10, respectively, using the same three overlapping linguistic terms (Low, Med, High).

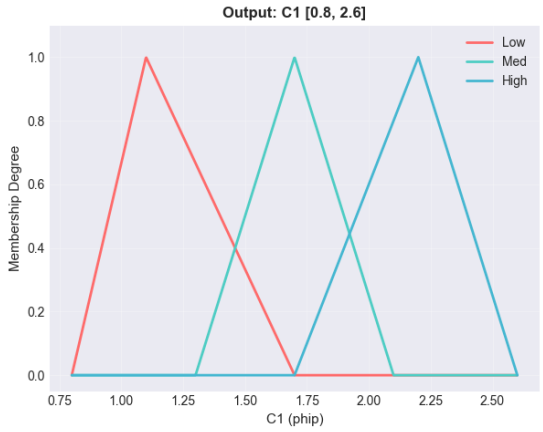

Fig. 9 Type-1 fuzzy membership functions for the output variable "c_1".

Fig. 9 shows the graphical definition of the Type-1 fuzzy membership functions used for the cognitive coefficient c_1.

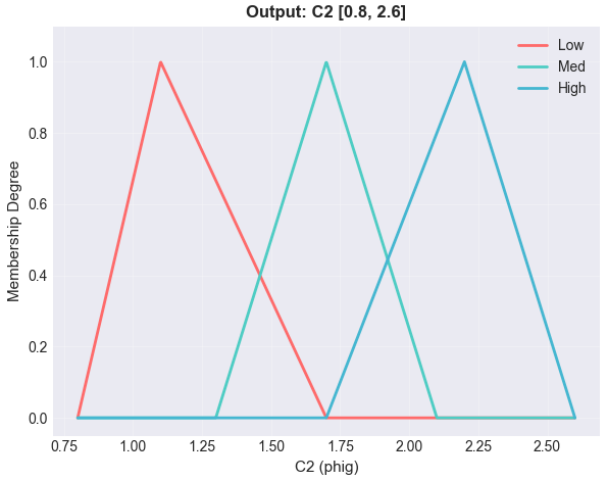

Fig. 10 Type-1 fuzzy membership functions for the output variable "c_2".

After defining the crisp Type-1 baseline we then introduce the Type-2 versions to represent uncertainty in how the coefficients are selected using FOU. This can be visualized in Fig. 11 for c_1, and Fig. 12 for c_2, the interval formulation keeps the same structure while adding uncertainty bounds.

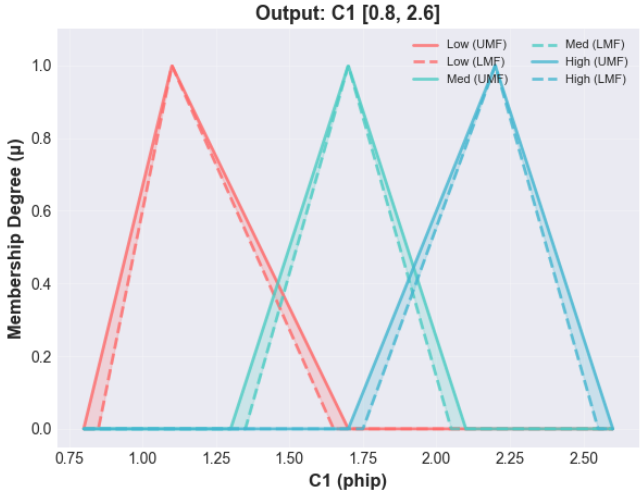

Fig. 11 Type-2 fuzzy membership functions for the output variable "c_1".

Type-1 uses triangular membership functions. Type-2 preserves the same support limits but narrows lower memberships, producing interval-valued outputs before type reduction and improving robustness under noisy evaluations.

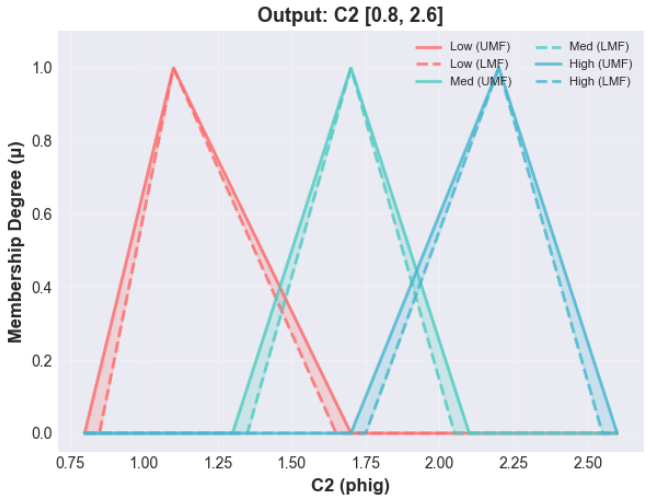

Fig. 12 Type-2 fuzzy membership functions for the output variable "c_2".

4.3 Fuzzy Rules

The rule base consists of 27 Mamdani rules, following a simple scheduling idea where early iterations emphasize exploration and later iterations emphasize exploitation with the improvement signal guiding how fast this shift happens. For the inertia weight the rules gradually reduce its value over time unless weak improvement suggests a need to explore again. For c_1 the rules move from higher to lower values as the search progresses which limits individual wandering. For c_2 the trend is reversed so that convergence is reinforced in later stages, this is summarized in Fig. 13.

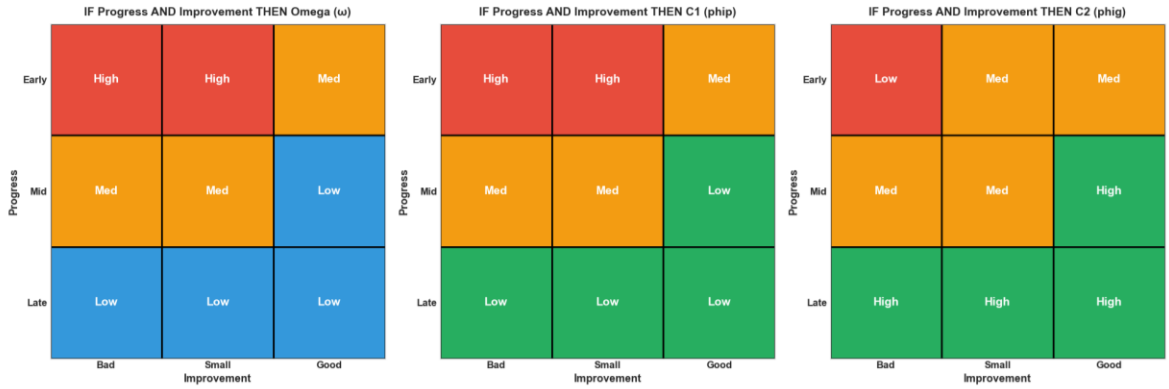

Fig. 13 Fuzzy rules matrix for Omega (ω), c_1 and c_2 based on Progress and Improvement.

4.4 Fuzzy Methodology

This subsection explains the inference process shared by both fuzzy systems using a common structure. We first describe the Type-1 Mamdani pipeline as the baseline case and then present the Type-2 extension which adds uncertainty through interval memberships and a type-reduction step. Laying out the two procedures side by side makes it easier to see which steps are the same and which differences account for the performance gap discussed in Section 5.

4.4.1 Type-1 Mamdani Inference

The Type-1 fuzzy system follows the standard Mamdani inference process and is implemented in four sequential stages. During fuzzification the crisp input values for progress and improvement are mapped to fuzzy membership degrees. Each input is evaluated against all linguistic terms defined for that variable and the result indicates how strongly the input belongs to each fuzzy set.

During rule evaluation the system computes firing strengths for all 27 rules in the rule base. Each firing strength is obtained using the product t-norm by multiplying the membership degrees of the antecedent terms which yields a single value that reflects how strongly a rule is activated by the current inputs. For example the rule IF progress IS Early AND improvement IS Good fires with a strength equal to the product of the membership of progress in Early and improvement in Good.

In the aggregation stage the outputs of all active rules are combined using the maximum s-norm. For each output variable, the system builds an aggregated fuzzy set by taking the maximum firing strength among all rules associated with each linguistic term. This results in three fuzzy sets that represent the recommended values for the PSO parameters.

Finally, the defuzzification stage converts these aggregated fuzzy sets into crisp numerical values using the center of gravity method. This computes the weighted average of the output domain, where the weight at each point is given by its membership degree in the aggregated fuzzy set. The result is three crisp parameter values that the PSO algorithm will use in the current iteration.

4.4.2 Type-2 Mamdani Inference

The Type-2 fuzzy system builds on the Type-1 procedure by adding steps that explicitly account for uncertainty. During fuzzification each crisp input is mapped to interval-valued memberships rather than single values. For every linguistic term both the upper and lower membership functions are evaluated which produces an interval that bounds the possible degree of membership. This interval reflects uncertainty in how the input should be interpreted.

Rule evaluation then operates on these intervals using interval arithmetic. Each rule is assigned a firing strength expressed as a range rather than a single number. Wider ranges indicate greater uncertainty about how strongly a rule applies under the current conditions.

During aggregation the interval firing strengths from all rules are combined to form Type-2 fuzzy sets for each output variable. These sets are described by their Footprints of Uncertainty which define the limits of the aggregated membership functions and carry the uncertainty forward to the final inference stage. The system then used type reduction with the Karnik–Mendel algorithm (Karnik & Mendel, 1999), this is an iterative procedure collapses each Type-2 fuzzy set into a crisp interval that represents a reasonable range of defuzzified values. The algorithm estimates the left and right centroid bounds by computing weighted averages of the rule consequents under different interpretations allowed by the FOU and updates the switching index until convergence is reached.

Finally defuzzification maps each output interval to a single scalar value by averaging its endpoints. This step produces three crisp PSO parameters that reflect uncertainty in both fitness evaluation and the assessment of the search state. By carrying this uncertainty through inference aggregation type reduction and defuzzification the Type-2 system supports more stable parameter adaptation than the Type-1 formulation.

4.5 Convolutional Neural Network Implementation

The architecture mapping translates each continuous particle position into a concrete CNN design in two steps. First the network depth is set by rounding the first coordinate and limiting it to a range between one and five. This keeps the model within the training budget and avoids unstable depths seen in early experiments. Next the remaining coordinates are rounded and constrained between 16 and 128 to define the number of filters in each convolutional layer. These bounds prevent degenerate designs and keep candidates trainable on our hardware while preserving a simple encoding for PSO that remains differentiable and easy to search.

The CNN model is constructed using a sequential architecture pattern. For each convolutional block, a convolutional layer with 3×3 kernels and ReLU activation is added, followed by a 2×2 max pooling layer for spatial dimension reduction. The first convolutional layer includes the input shape specification of (48, 48, 1) corresponding to grayscale facial expression images, while subsequent layers infer dimensions from their predecessors. All convolutional layers use same padding to maintain spatial dimensions before pooling.

Following the variable-depth convolutional feature extractor, the model appends a fixed classification head. This begins with a flattening operation to convert 2D feature maps to 1D feature vectors. Two fully connected layers with 192 and 28 neurons respectively, both using ReLU activation and L2 regularization with coefficient 0.0001, perform non-linear transformation of the learned features. A dropout layer with rate 0.5 is inserted after the first dense layer to mitigate overfitting. Finally, a 7-neuron output layer with softmax activation produces probability distributions over the seven emotion categories in FER2013.

4.6 PSO – Fuzzy Optimization

The complete optimization algorithm combines standard PSO dynamics with fuzzy parameter adaptation in a single procedure. The algorithm takes as input a fitness function for evaluating CNN architectures, a swarm size (set to 10 in our experiments), a maximum number of iterations (set to 20), a fuzzy system instance (either Type-1 or Type-2), and the lower and upper bounds of the search space.

Initialization begins by sampling particle positions uniformly within the bounded search space. Each position encodes a six-dimensional CNN architecture. Particle velocities are initialized with random values within predefined limits. For each particle, the initial position is stored as the personal best together with its fitness value. The global best is then identified as the particle with the lowest fitness in the initial swarm. We also keep track of the previous global best fitness to compute improvement during later iterations.

The main optimization loop runs for iterations, at each iteration, two signals are computed to describe the current state of the search. Progress is defined as the ratio between the current iteration and the maximum number of iterations. Improvement is measured as the relative change in the global best fitness compared to the previous iteration. These signals are then passed to the fuzzy system to adapt the PSO parameters before updating particle velocities and positions. The improvement signal is clipped to the range from -1 to 1 to ensure it remains within the fuzzy system's input domain. These signals are passed to the fuzzy system, which performs inference and returns adapted values for the three PSO parameters: inertia weight ω, cognitive coefficient c_1, and social coefficient c_2.

Using these adapted parameters, particle velocities are updated according to the standard PSO velocity equation, which combines inertial motion (scaled by ω), cognitive attraction toward personal best, and social attraction toward global best. Velocities are clamped to maximum magnitudes to prevent explosive growth. Particle positions are then updated by adding velocities, with positions subsequently clamped to search space bounds to enforce feasibility.

Each new particle position is evaluated using the fitness function which trains the corresponding CNN architecture and returns its validation performance. Personal best positions and fitness values are updated whenever a particle improves on its previous best result. The global best is updated if any personal best outperforms the current global best and the previous global best fitness is stored to compute improvement in the next iteration. Optional logging records the global best fitness and the adapted parameters at each step.

When the maximum number of iterations is reached or a convergence condition is met the algorithm returns the global best architecture encoding together with its fitness value.

5 Results and Analysis

We use the FER2013 dataset, which is commonly used in facial emotion recognition studies. The dataset is split into three parts. The training set contains 25121 images, which represents about 70 percent of the data. The validation set includes 7177 images, close to 20 percent, and the remaining 3589 images are used for testing. This split allows the models to be trained with enough data while keeping a meaningful validation set for PSO fitness evaluation and a separate test set for the final performance analysis.

Preprocessing begins by loading the FER2013 images, which are provided as space separated pixel values. These values are converted into 48 by 48 arrays with unsigned 8 bit integers. The pixel intensities are then normalized to a range between 0 and 1 by converting them to 32 bit floating point values and dividing by 255. The images are reshaped to N by 48 by 48 by 1 to match the single channel grayscale input expected by convolutional neural networks. Emotion labels are encoded as seven dimensional one hot vectors for multi class classification. Finally, the dataset is split using stratified sampling, with 70 percent used for training, 20 percent for validation during PSO optimization, and 10 percent reserved for final testing.

Data augmentation techniques such as rotation flipping and zooming are used during the PSO search phase. The goal is to expose each candidate architecture to a more varied set of training samples so that fitness evaluation reflects generalization rather than performance on a narrow data distribution. Using the same augmentation strategy for all particles keeps the comparison fair and avoids bias across architectures. Although augmentation increases training cost it helps reduce overfitting during search and makes the fitness signal more representative of real use. Once the search is complete the selected architecture can be retrained and the augmentation strategy adjusted if needed to further improve generalization.

We executed 30 independent experiments per optimization strategy to assess both the quality and consistency of each approach. Three configurations were evaluated under identical conditions, the standard PSO with fixed parameters (ω=0.5, c_1=0.5, c_2=0.5), PSO guided by a Type-1 Mamdani fuzzy system, and PSO guided by a Type-2 interval Mamdani fuzzy system. The static baseline yielded a mean training accuracy of 85.22%, while Type-1 Fuzzy PSO obtained 86.43% and Type-2 Fuzzy PSO reached 89.17%. When placed side by side, these numbers reveal a clear progression, adopting adaptive control adds 1.21 percentage points over fixed parameters, and incorporating uncertainty modeling through Type-2 membership functions contributes another 2.74 points on top of that.

Table 1 collects the full results across all 30 runs for the three optimization strategies. Training accuracy, loss, and macro F1-score are reported per experiment, allowing direct comparison of run-to-run consistency and peak attainment between the static baseline and the two fuzzy-adaptive variants.

Table 1. PSO Optimization runs

	PSO - Static			PSO - Type-1			PSO – Type-2		
Run	**Accuracy**	**Loss**	**F1 Score**	**Accuracy**	**Loss**	**F1 Score**	**Accuracy**	**Loss**	**F1 Score**
1	86.45%	40.08%	49.63%	93.21%	23.65%	51.93%	92.02%	26.53%	51.08%
2	89.00%	35.19%	49.75%	85.57%	51.07%	51.60%	94.36%	22.01%	51.69%
3	83.20%	52.00%	53.56%	84.12%	51.69%	50.37%	92.03%	28.01%	51.10%
4	83.37%	55.16%	51.32%	93.59%	23.14%	51.39%	84.80%	50.02%	52.58%
5	84.77%	49.31%	52.58%	82.67%	54.00%	52.09%	82.82%	56.60%	52.93%
6	89.93%	32.63%	50.03%	90.65%	28.88%	49.90%	87.06%	38.96%	50.89%
7	81.48%	58.73%	51.49%	86.66%	50.29%	53.63%	85.54%	49.12%	52.45%
8	83.27%	52.73%	51.48%	84.86%	50.48%	51.43%	91.17%	30.71%	51.02%
9	82.70%	53.76%	52.18%	89.68%	33.25%	51.29%	82.49%	55.57%	51.20%
10	93.14%	24.04%	51.13%	82.61%	56.01%	51.92%	84.20%	52.73%	53.31%
11	85.62%	48.46%	52.62%	87.97%	44.72%	51.56%	84.56%	51.07%	53.20%
12	85.71%	49.68%	52.86%	82.61%	56.01%	51.92%	92.91%	25.62%	51.03%
13	82.53%	58.44%	51.37%	84.66%	50.23%	53.68%	93.13%	25.14%	50.96%
14	84.44%	51.28%	51.43%	93.05%	24.50%	50.82%	95.27%	18.24%	52.67%
15	88.66%	35.19%	50.23%	83.39%	54.18%	52.04%	93.76%	22.62%	51.12%
16	78.55%	59.27%	49.84%	90.65%	28.88%	49.91%	85.29%	48.54%	52.25%
17	89.38%	33.18%	50.67%	80.55%	58.32%	51.85%	92.93%	24.69%	52.65%
18	83.61%	53.62%	52.54%	83.39%	54.18%	52.04%	85.87%	49.16%	53.76%
19	78.94%	62.45%	51.14%	94.02%	22.70%	51.97%	94.13%	21.64%	51.27%
20	82.94%	56.05%	53.19%	87.97%	44.71%	51.56%	85.42%	50.03%	53.30%
21	81.97%	57.99%	52.55%	80.55%	58.32%	51.85%	84.07%	52.58%	52.42%
22	83.89%	52.74%	53.08%	80.78%	59.44%	51.00%	93.25%	24.08%	51.92%
23	89.99%	32.06%	51.52%	93.58%	23.41%	49.72%	89.26%	33.68%	50.39%
24	86.31%	49.43%	53.02%	81.78%	55.02%	50.97%	89.78%	32.74%	49.85%
25	88.99%	34.46%	50.15%	89.68%	33.25%	51.29%	83.81%	50.98%	50.91%
26	83.30%	54.73%	52.30%	92.47%	25.62%	51.03%	88.99%	42.85%	54.30%
27	84.12%	54.15%	52.64%	84.98%	48.83%	52.50%	84.74%	54.10%	52.97%
28	85.06%	49.90%	51.58%	81.78%	55.02%	50.97%	94.20%	21.00%	51.19%
29	91.74%	28.63%	50.77%	88.60%	37.58%	49.73%	94.27%	21.67%	50.66%
30	83.59%	54.09%	53.02%	76.73%	68.09%	50.61%	93.02%	25.46%	51.61%

To make clear how each method treats the PSO parameters, Table 2 lists all relevant settings used in the experiments. This includes the general hyperparameters of the framework, the PSO parameters that control swarm behavior, and the fuzzy

system parameters that define how adaptation is carried out. Presenting them together helps clarify what is shared across methods and what differs between them.

Table 2. Hyperparameters Configuration

Hyperparameter	PSO-Static	PSO-Type-1 Fuzzy	PSO-Type-2 Fuzzy
PSO Hyperparameters			
Depth bounds	[1-5] layers	[1-5] layers	[1-5] layers
Filter bounds	[16-128] per layer	[16-128] per layer	[16-128] per layer
ω (inertia weight)	0.5 (fixed)	[0.3-0.9] fuzzy-adapted	[0.3-0.9] fuzzy-adapted
c_1 (cognitive coeff.)	0.5 (fixed)	[0.8-2.6] fuzzy-adapted	[0.8-2.6] fuzzy-adapted
c_2 (social coeff.)	0.5 (fixed)	[0.8-2.6] fuzzy-adapted	[0.8-2.6] fuzzy-adapted
Fuzzy system Hyperparameters			
Fuzzy inputs	—	progress, improvement	progress, improvement
Fuzzy outputs	—	ω, c_1, c_2	ω, c_1, c_2
	—	Center of Gravity	Karnik–Mendel + centroid

Table 3 highlights the main comparative patterns. Relative to the PSO-Static baseline, the Type-1 approach increases mean accuracy by 1.21 percentage points, rising from 85.22 percent to 86.43 percent. The Type-2 variant adds a further gain of 2.74 points, reaching 89.17 percent, and also lowers the loss from 44.18 percent under Type-1 to 36.87 percent. These changes point to a clear improvement in optimization behavior when uncertainty is modeled explicitly.

F1-macro scores remain very similar across methods at 51.66 percent for Static, 51.42 percent for Type-1, and 51.89 percent for Type-2. This suggests that the improvements are mainly reflected in overall accuracy and search quality rather than in strong shifts across individual emotion classes. Type-2 also produces the highest single-run accuracy at 95.27 percent and shows a higher lower bound at 82.49 percent than the other methods, which matters in scenarios where only a small number of optimization runs can be carried out.

The convergence curves show that all methods stabilize between iterations 15 and 20. The fixed-parameter baseline tends to level off earlier, while the fuzzy-guided approaches continue to improve during the middle and later stages of the search. This behavior is consistent with a better balance between exploration and exploitation.

Table 3. Classification Performance Metrics: PSO vs Type-1 vs Type-2 Fuzzy-PSO on FER2013 (30 runs)

Method	Accuracy	Loss	F1-Score	Std Dev	Best Accuracy
PSO-Static	85.22%	47.65%	51.66%	3.53%	93.14%
PSO – Type-1	86.43%	44.18%	51.42%	4.79%	94.02%
PSO - Type-2	89.17%	36.87%	51.89%	4.31%	95.27%

The results show a steady progression as the method moves from fixed PSO parameters to Type-1 and then to Type-2 fuzzy system. Compared with PSO-Static, the Type-1 approach increases mean accuracy by 1.21 percentage points and reduces loss

from 47.65 percent to 44.18 percent. This confirms that adapting PSO parameters during the search already provides a clear benefit over fixed settings. Moving from Type-1 to Type-2 adds a further 2.74 points in accuracy and lowers the loss to 36.87 percent. This indicates that explicitly accounting for uncertainty improves optimization beyond what a crisp fuzzy system can achieve. When these gains are considered alongside the observed reduction in training time for Type-2 architectures, from 3.04 minutes to 1.42 minutes, the Type-2 fuzzy PSO configuration appears better suited for practical use cases where both accuracy and computational cost are important.

5.1 Final test-set performance and additional analyses

The experiment 14 in PSO – Type-2 already gives us the test metrics we were looking for. After running the evaluation on the reserved 10% test split, the model came in at 53.09% test accuracy, with a test loss of 2.9867. Looking at the macro-averaged scores, F1 landed at 49.93%, precision at 51.98%, and recall at 49.10%, all calculated across 7,178 test images. These numbers suggest the architecture does manage to generalize beyond the validation set, though it's worth keeping in mind that FER2013's class imbalance still puts a ceiling on how high the macro-averaged scores can go.

Table 4 then breaks this down further, laying out precision, recall, and F1-score for each individual emotion class. This breakdown matters because it lets us see which emotions the model picks up on consistently and which ones still trip up the current architecture.

Table 4. Class-wise precision, recall, and F1-score on the test set

Class	Precision	Recall	F1 Score
Angry	48.03%	37.59%	42.17%
Disgust	50.77%	29.20%	37.08%
Fear	34.62%	40.91%	37.50%
Happy	76.88%	71.27%	73.96%
Neutral	47.41%	53.36%	50.21%
Sad	42.05%	41.28%	41.66%
Surprise	64.08%	70.09%	66.95%

As shown in Fig. 14, the confusion matrix follows the pattern we would expect from FER2013, happy and surprise come out recognized more reliably, while disgust and fear lag behind. A good chunk of the errors, meanwhile, comes from classes that tend to overlap visually, like angry, sad, and neutral. This tells us the model is picking up on real, meaningful structure in the data, but it also confirms that class imbalance is still holding back recall for the minority classes.

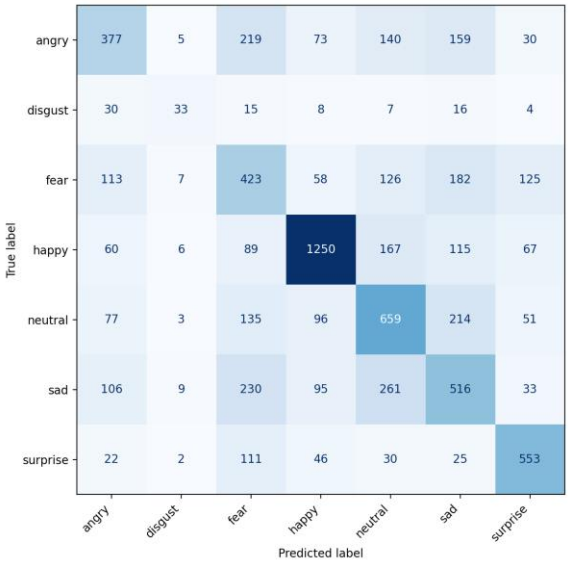

Fig.14 Confusion Matrix on the test set

Accuracy goes up mainly because the model is getting more predictions right overall, and that's largely driven by the frequent classes. Macro F1, on the other hand, stays lower since it weighs every class the same regardless of how common it is. In this particular run, happy and surprise show the biggest gains, while disgust and fear are still only getting modest recall. So yes, the accuracy improvement is real, but it's not spread evenly across classes. That's exactly why macro F1 hovers around 50% instead of climbing at the same pace as accuracy.

Interval Type-2 fuzzy control works by making the PSO updates less reactive to noisy validation feedback during CNN training. In other words, the swarm responds more smoothly to short-term fluctuations and doesn't overcommit to intermediate solutions that might not hold up. This is particularly relevant for FER2013, given that the training signal is noisy, the dataset is imbalanced, and even small architectural tweaks can cause validation behavior to swing.

5.2 Statistical Validation

Although the descriptive statistics in Table 3 point to clear differences in performance, these results still need to be tested to determine whether they reflect real advantages or could simply be due to random variation across runs. To this end, we carried out one-tailed Z-tests for independent samples at a significance level of $\alpha = 0.05$. Each method was executed 30 times ($n_1 = n_2 = 30$), and the rejection region corresponds to $z > 1.645$, a summary of this can be seen in Table 5.

The first comparison evaluates whether PSO - Type-2 systems outperforms PSO - Type-1, the resulting test statistic is $z = 2.33$, which falls inside the rejection region. This confirms, with 95% confidence, that the accuracy advantage of the Type-2 system over the Type-1 system is not a product of chance.

The second comparison tests whether PSO – Type-1 outperforms PSO-Static. Here, the test statistic is $z = 1.115$, which does not reach the critical threshold. Consequently, the 1.21 percentage point difference observed in Table 3 cannot be declared statistically significant at the 5% level. Although the Type-1 system does improve the mean, the margin is too narrow, relative to the run-to-run variance to rule out sampling noise as a possible explanation.

The third comparison contrasts PSO - Type-2 directly against PSO-Static. The test statistic rises to $z = 3.88$, well beyond the critical value, leaving no reasonable doubt that the Type-2 system produces genuinely higher accuracy than the fixed-parameter baseline.

Table 5. One-tailed Z-Test Results ($\alpha = 0.05$, $n_1 = n_2 = 30$)

Comparison	x1	x2	z statistic	Decision
Type-2 vs Type-1	89.17%	86.43%	2.33	Reject H_o
Type-1 vs PSO-Static	86.43%	85.22%	1.115	Fail to reject H_o
Type-2 vs PSO-Static	89.17%	85.22%	3.88	Reject H_o

Taken together, these tests draw a sharper picture than the raw averages alone. The Type-2 fuzzy system is the only variant whose improvement over both baselines reach statistical significance. Although the Type-1 system achieves a slightly higher mean accuracy than the PSO-Static baseline, the difference does not reach statistical significance. This result suggests that crisp membership functions, which do not model uncertainty explicitly, capture only part of the potential benefit of adaptive system. In practical terms, this means that choosing a Type-1 system over fixed parameters does not guarantee a reliable improvement. On the other hand, the Type-2 system shows a statistically supported gain, giving a clearer expectation of improved accuracy when uncertainty is taken into account.

6 Conclusions

This paper evaluates the automation of CNN architecture design for facial emotion recognition by comparing three optimization approaches, the first one includes a standard PSO with fixed parameters, the second one a PSO with a Type-1 Mamdani fuzzy system, and the third one PSO with an Interval Type-2 Mamdani fuzzy system. All three methods were tested on the FER2013 dataset using 30 independent runs. Across these experiments, the results followed a stable performance pattern, which helps separate the effects of adaptive parameter control from those of explicit uncertainty handling in neural architecture search.

The PSO with static parameter got a mean training accuracy of 85.22 percent and serves as a useful reference, with the introduction of Type-1 fuzzy system increased mean accuracy to 86.43 percent, corresponding to an improvement of 1.21 percentage points. This improvement is consistent with the system capacity to adapt PSO behavior over time, gradually moving the search from broader exploration toward more focused exploitation. On the other hand, the Type-2 fuzzy system produced a larger improvement, with a mean accuracy of 89.17 percent. A variability across runs remained in a similar range, with standard deviations of 3.53 percent for PSO-Static, 4.79 percent for Type-1, and 4.31 percent for Type-2. F1-macro

scores were comparable across all three methods at 51.66 percent, 51.42 percent, and 51.89 percent, suggesting that class balance did not change substantially, even though overall accuracy and loss did.

Both fuzzy methods used the same rule base, with the difference limited to how the membership functions were defined. Keeping this structure fixed allowed the optimization process to balance exploration and exploitation effectively across iterations. Early in the search, the rules favored diversity by assigning higher inertia weights and stronger cognitive influence. Later iterations shifted toward convergence by reducing inertia and increasing the social coefficient. In the Type-2 system, the use of FOU reduced the sensitivity to noisy fitness values, which commonly arise when CNNs are trained for only a few epochs. As a result, the parameter updates were generally more stable than those produced by the crisp membership functions in the Type-1 system.

The study has several limitations that should be acknowledged, experiments were conducted using only FER2013 dataset, this dataset is known to be challenging due to its low resolution and unconstrained acquisition conditions, it reflects only a single application domain. Evaluating the approach on additional datasets, such as CK+ or JAFFE, would provide a clearer picture of its ability to generalize.

Future work could extend the search space to include residual connections, attention mechanisms, and normalization layers, bringing the framework closer to current CNN design practice.

Overall, the results suggest that Type-2 fuzzy logic has clear practical advantages over Type-1 systems for adapting PSO parameters during CNN optimization. Representing uncertainty directly in the membership functions seems especially helpful when fitness evaluations are noisy and expensive to compute. For work in facial emotion recognition and related classification tasks, Type-2 fuzzy PSO offers a reasonable way to automate architecture design while keeping the process interpretable and reducing computational cost.

References

Alfi, A., & Fateh, M. M. (2011). Intelligent identification and control using improved fuzzy particle swarm optimization. *Expert Systems with Applications*, *38*(10), 12312–12317. https://doi.org/10.1016/J.ESWA.2011.04.009

Almansour, N., Sahran, S., Abdullah, A., Albashish, D., & Hamzah, J. C. (2025). Hyperparameter optimization of convolutional neural networks using particle swarm optimization for diabetic retinopathy detection. *PeerJ Computer Science*, *11*, e3273. https://doi.org/10.7717/PEERJ-CS.3273

Alwan, H. B. (2025). Develop, Design, and Evaluate a Real-Time Facial Micro-Expression Recognition Model for Mental Health Monitoring Using a Three-Dimensional Convolutional Neural Network. *IEEE Access*, *13*, 187602–187613. https://doi.org/10.1109/ACCESS.2025.3626566

Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S., & Abraham, A. (2011). Inertia weight strategies in particle swarm optimization. *Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011*, 633–640. https://doi.org/10.1109/NABIC.2011.6089659

Behzad, M. (2025). SMILE-VLM: Self-Supervised Multi-View Representation Learning Using Vision-Language Model for 3D/4D Facial Expression Recognition. *IEEE Access*, *13*, 143831–143842. https://doi.org/10.1109/ACCESS.2025.3598019

Berenji, H. R. (1992). Fuzzy Logic Controllers. In R. R. & Z. L. A. Yager (Ed.), *An Introduction to Fuzzy Logic Applications in Intelligent Systems* (pp. 69–96). Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-3640-6_4

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*, 281–315. https://doi.org/10.5555/2188385.2188395

Bhati, V. S., Tiwari, N., & Chawla, M. (2025). A Generalized Zero-Shot Deep Learning Classifier for Emotion Recognition Using Facial Expression Images. *IEEE Access*, *13*, 18687–18700. https://doi.org/10.1109/ACCESS.2025.3533580

Castillo, O., Amador-Angulo, L., Castro, J. R., & Garcia-Valdez, M. (2016). A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Information Sciences*, *354*, 257–274. https://doi.org/10.1016/J.INS.2016.03.026

Castillo, O., & Melin, P. (2014). A review on interval type-2 fuzzy logic applications in intelligent control. *Information Sciences*, *279*, 615–631. https://doi.org/10.1016/J.INS.2014.04.015

Castillo, O., Melin, P., Kacprzyk, J., & Pedrycz, W. (2007). Type-2 Fuzzy Logic: Theory and Applications. *2007 IEEE International Conference on Granular Computing*, 145–145. https://doi.org/10.1109/GRC.2007.118

Chen, H. C., Widodo, A. M., Wisnujati, A., Rahaman, M., Lin, J. C. W., Chen, L., & Weng, C. E. (2022). AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics*, *11*(6), 951. https://doi.org/10.3390/electronics11060951

Elhawat, M., & Altınkaya, H. (2025). Frequency Regulation of Stand-Alone Synchronous Generator via Induction Motor Speed Control Using a PSO-Fuzzy PID Controller. *Applied Sciences 2025, Vol. 15, Page 3634*, *15*(7), 3634. https://doi.org/10.3390/APP15073634

Haghrah, A. A., Ghaemi, S., & Badamchizadeh, M. A. (2025). PyIT2FLS: An open-source Python framework for flexible and scalable development of type 1 and interval type 2 fuzzy logic models. *SoftwareX*, *30*, 102146. https://doi.org/10.1016/J.SOFTX.2025.102146

He, J., Li, S., Shen, J., Liu, Y., Wang, J., & Jin, P. (2018). Facial expression recognition based on VGGNet convolutional neural network. *2018 Chinese Automation Congress (CAC)*, 4146–4151. https://doi.org/10.1109/CAC.2018.8623238

Izard, C. E. (1991). *The psychology of emotions*. Plenum Press. https://doi.org/10.1007/978-1-4899-0615-1

Kalaivani, S., Tharini, C., Viswa, T. M. S., Sara, K. Z. F., & Abinaya, S. T. (2024). ResNet-based classification for leaf disease detection. *Journal of the Institution of Engineers (India): Series B*, *106*(1), 1–14. https://doi.org/10.1007/s40031-024-01062-7

Karnik, N. N., & Mendel, J. M. (1999). Applications of type-2 fuzzy logic systems to forecasting of time-series. *Information Sciences*, *120*(1–4), 89–111. https://doi.org/10.1016/S0020-0255(99)00067-5

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95—International Conference on Neural Networks*, *4*, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Khan, A. S., Shafi, I., & Khan, A. H. (2025). Vision–language pretraining for image captioning using facial expression recognition. *IEEE Access*, *13*, 210435–210447. https://doi.org/10.1109/ACCESS.2025.3643197

Liao, J., Guo, L., Jiang, L., Yu, C., Liang, W., Li, K., & Pop, F. (2025). A machine learning-based feature extraction method for image classification using ResNet architecture. *Digital Signal Processing*, *160*, 105036. https://doi.org/10.1016/J.DSP.2025.105036

Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, *149*, 153–165. https://doi.org/10.1016/J.CHEMOLAB.2015.08.020

Mendel, J. M. (2024). Type-1 fuzzy sets and fuzzy logic. In *Explainable uncertain rule-based fuzzy systems* (pp. 17–73). Springer. https://doi.org/10.1007/978-3-031-35378-9_2

Mittal, K., Jain, A., Vaisla, K. S., Castillo, O., & Kacprzyk, J. (2020). A comprehensive review on type 2 fuzzy logic applications: Past, present and future. *Engineering Applications of Artificial Intelligence*, *95*, 103916. https://doi.org/10.1016/J.ENGAPPAI.2020.103916

Mohammadi Lanbaran, N., Naujokaitis, D., Kairaitis, G., & Radziukynas, V. (2025). Hybrid Hourly Solar Energy Forecasting Using BiLSTM Networks with Attention Mechanism, General Type-2 Fuzzy Logic Approach: A Comparative Study of Seasonal Variability in Lithuania. *Applied Sciences 2025, Vol. 15, Page 9672*, *15*(17), 9672. https://doi.org/10.3390/APP15179672

Niknam, T., Amiri, B., Olamaei, J., & Arefi, A. (2009). An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering. *Journal of Zhejiang University-SCIENCE A 2009 10:4*, *10*(4), 512–519. https://doi.org/10.1631/JZUS.A0820196

Papic, A., Herenda, F., Hubana, T., & Hodzic, M. (2025). Beyond Trial and Error: Comparative Analysis of Heuristic Strategies for Optimal Neural Network Architectures. *2025 24th International Symposium INFOTEH-JAHORINA, INFOTEH 2025 - Proceedings*. https://doi.org/10.1109/INFOTEH64129.2025.10959303

Pazandeh, A. M., & Fatemizadeh, E. (2025). Enhancing Vision Transformers for Facial Expression Recognition. *IEEE Access*, *13*, 144689–144698. https://doi.org/10.1109/ACCESS.2025.3598917

Raza, A., Musa, S., Shahrafidz Bin Khalid, A., Mansoor Alam, M., Mohd Su'Ud, M., & Noor, F. (2024). Enhancing Medical Image Classification Through PSO-Optimized Dual Deterministic Approach and Robust Transfer Learning. *IEEE Access*, *12*, 177144–177159. https://doi.org/10.1109/ACCESS.2024.3504266

Sauter, D. A. (2017). The nonverbal communication of positive emotions: An emotion family approach. *Emotion Review*, *9*(3), 222–234. https://doi.org/10.1177/1754073916667236

Schröder, S., Baratchi, M., & van Rijn, J. N. (2025). Overfitting in combined algorithm selection and hyperparameter optimization. In *Advances in Intelligent Data Analysis XXIII* (pp. 181–194). Springer. https://doi.org/10.1007/978-3-031-91398-3_14

Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Computer Science*, *132*, 377–384. https://doi.org/10.1016/J.PROCS.2018.05.198

Sharma, R., Matharu, J. S., & Parmar, K. S. (2026). A survey on particle swarm optimization: Evolution, adaptations and practical implementations. *Applied Soft Computing*, *186*, 114016. https://doi.org/10.1016/j.asoc.2025.114016

Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. *1998 IEEE International Conference on Evolutionary Computation Proceedings*, 69–73. https://doi.org/10.1109/ICEC.1998.699146

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, *3*, 1945–1950. https://doi.org/10.1109/CEC.1999.785511

Shirani Shamsabadi, J., Ansari Mahyari, S., & Ghaderi-Zefrehei, M. (2025). Adaptive neuro-fuzzy inference systems for improved mastitis classification and diagnosis. *Scientific Reports*, *15*, 20456. https://doi.org/10.1038/s41598-025-03008-5

Veiga, B., Faia, R., Pinto, T., & Vale, Z. (2022). Pyticle Swarm: A Particle Swarm Optimization Library for Python. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4169749

Waysi Naaman, D., Tahir Ahmed, B., & Ibrahim, M. I. (2025). Optimization by nature: A review of genetic algorithm techniques. *Indonesian Journal of Computer Science*, *14*(1), 268–284. https://doi.org/10.33022/ijcs.v14i1.4596

Wu, D. (2013). Approaches for reducing the computational cost of interval Type-2 fuzzy logic systems: Overview and comparisons. *IEEE Transactions on Fuzzy Systems*, *21*(1), 80–99. https://doi.org/10.1109/TFUZZ.2012.2201728

Zahara, L., Musa, P., Wibowo, E. P., Karim, I., & Musa, S. B. (2020). The Facial Emotion Recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1–9. https://doi.org/10.1109/ICIC50835.2020.9288560

Acknowledgments: The work was done with partial support from grants.

Data Availability Statement: The Dataset will be available on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.