Improving the performance of convolutional neural networks using a fuzzy logic based pooling method

Mai The Anh[1], Dang Thai Son[1], Le Van Chuong[1],
Le Thai An[1], Bui Quang Huy[1], Ngo Tri Nam Cuong[2*]

[1]Vinh University, 182 Le Duan, Truong Vinh, Nghe An, Vietnam;
[2]Institute of Electrical Engineering and Automation, LK43/4 Nguyen Van Loc, Ha Dong, Hanoi, Vietnam.
[*]Corresponding author: ncuong792000@gmail.com

Received 05 Feb. 2026; Revised 26 Mar. 2026; Accepted 15 Jun. 2026; Published 25 Jun. 2026.
DOI: https://doi.org/10.54939/1859-1043.j.mst.112.2026.29-37

ABSTRACT

In automatic image recognition and classification tasks, convolutional neural networks (CNN) are widely used, in which pooling operations play an important role in aggregating and highlighting image feature characteristics. This paper proposes a fuzzy logic based pooling method that integrates minimum, average, and maximum feature responses through a simple Takagi–Sugeno inference mechanism. Unlike conventional pooling methods, the proposed approach explicitly models uncertainty in feature aggregation, enabling a better balance between salient local features and contextual in order to preserve important image information. The effectiveness of the proposed method is evaluated on MNIST and CIFAR-10 datasets. Experimental results demonstrate that the proposed fuzzy pooling method achieves superior classification performance compared to conventional pooling methods, while maintaining comparable computational complexity. This enables the method to be effectively applied in CNN based image classification tasks.

Keywords: CNN; Deep learning; Fuzzy logic; Pooling layer; Classification.

1. INTRODUCTION

For decades, Artificial Neural Networks (ANNs) have been widely applied across numerous fields, including engineering, audio processing, image processing, medicine, disease diagnosis, and personal assistants. ANNs are computational systems that simulate biological neural networks, capable of seeking optimal solutions for a variety of problems, such as speech translation [1, 2], computer vision [3], and optimization [4]. Owing to their multi-layer architecture, ANNs can effectively remove noise, automatically learn intricate features, and efficiently handle highly nonlinear problems. The advancement of Graphics Processing Units (GPUs) has significantly accelerated the development of Deep Learning (DL), enabling the construction of networks with complex architectures through efficient parallel processing. DL has demonstrated its efficacy in numerous applications, particularly in classification tasks [5, 6]. Among these, the Convolutional Neural Network (CNN) stands out as one of the most prevalent architectures, employed in classification, object segmentation, image processing, natural language processing, and support systems. A CNN comprises several layers, such as convolutional, ReLU, pooling, and fully connected layers, allowing the network to automatically learn spatial features through the backpropagation process [7-10]. Several studies have proposed combining traditional pooling methods, such as max pooling and average pooling, to enhance CNN performance. A blending approach with a mixing ratio parameter was presented in [11], while soft pooling techniques based on the combination of these two methods were investigated in [12]. However, previous approaches have predominantly focused on blending pooling techniques without considering the uncertainty inherent in feature information, nor the minimum values of the features in the convolutional layer output. Some research has integrated fuzzy logic with neural networks, for instance, by employing fuzzy inference in the pooling layer or combining it with deep networks for image reconstruction

[13, 14]. This paper proposes a fuzzy pooling method that combines max pooling and average pooling based on fuzzy logic to improve CNN performance. The method utilizes the max, min, and average values of features to address uncertainty during the information extraction process and is evaluated on the MNIST and CIFAR-10 benchmark datasets. The main contributions include: proposing a novel pooling method for CNNs; applying fuzzy inference to enhance feature exploitation capabilities; evaluating its effectiveness on standard datasets; and maintaining the advantages of simplicity, speed, and low computational cost compared to traditional approaches. This study focuses on analyzing the impact of integrating fuzzy logic into pooling under controlled conditions, rather than aiming for the highest performance on complex models. The rest of this paper is constructed as follows. Introduction is presented in Section 1. Conventional pooling methods, such as average pooling, max pooling, and the proposed fuzzy combined pooling operation, are introduced in Section 2. In Section 3, the experiments, comparisons and discussions are implemented with classification problems using CNN on publicly available datasets i.e. MNIST and CIFAR-10 datasets. Finally, the conclusions are summarized in Section 4.

2. PROBLEM

2.1. Conventional pooling operations

Max pooling and average pooling are the two most popular conventional pooling forms in applications of CNNs. Max pooling method is implemented by taking the maximum value of a pooling spatial region and selecting the maximum value of a pooling spatial region for the received condensed feature maps. The max pooling function can be expressed as follows:

$$a_{kij} = \max_{(p,q)\in R_{ij}} (a_{kpq}) . \tag{1}$$

While average pooling method involves calculating the average value of a pooling region and selecting the average value of a pooling region for the condensed feature map. The average pooling function can be shown by the following expression:

$$a_{kij} = \frac{1}{|R_{ij}|} \sum_{(p,q)\in R_{ij}} a_{kpq} , \tag{2}$$

where a_{kij} is the output activation of the k^{th} feature map at (i, j), a_{kpq} is the input activation at (p, q) within a pooling region R_{ij}, and $|R_{ij}|$ is the size of the pooling region.

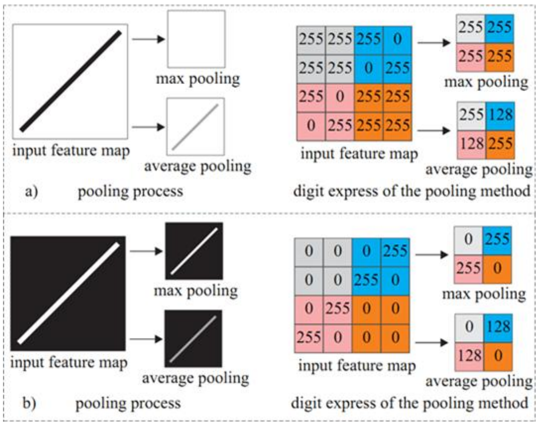

Figure 1. *Illustration of the drawbacks of max pooling (a) and average pooling (b).*

Both max and average pooling are efficient and simple methods widely used in many applications. However, as discussed in Section 1, they also have limitations: max pooling may overemphasize isolated high responses while discarding useful information, whereas average pooling may smooth out important local features. These shortcomings are illustrated in Figure 1.

Depending on the spatial distribution of activations within a pooling region, max pooling may discard relevant structural details when less informative responses exhibit higher intensity than the truly meaningful features. Conversely, average pooling can suppress important local patterns by uniformly averaging activations, particularly when low magnitude responses dominate the pooling region. These limitations motivate the need for an adaptive pooling mechanism that can balance extreme and contextual feature responses. In the following, we propose a fuzzy logic based pooling method built upon an inference mechanism that optimally combines the maximum, minimum, and average feature responses.

2.2. Fuzzy combined pooling operation

Max pooling effectively preserves strong local activations, such as edges and corner-like structures, which correspond to salient low level image features. However, it may overemphasize isolated responses while neglecting surrounding contextual information. In contrast, average pooling aggregates all activations within a local region, producing smoother feature representations but potentially suppressing discriminative details. Motivated by the complementary properties of these two operations, we propose a pooling method that integrates max and average pooling through fuzzy inference. Figure 2 illustrates the integration of the proposed fuzzy pooling into a CNN architecture. By adaptively combining sharp and smooth feature responses, the fuzzy pooling layer replaces the conventional pooling layer without modifying other components of the network, enabling balanced feature aggregation.

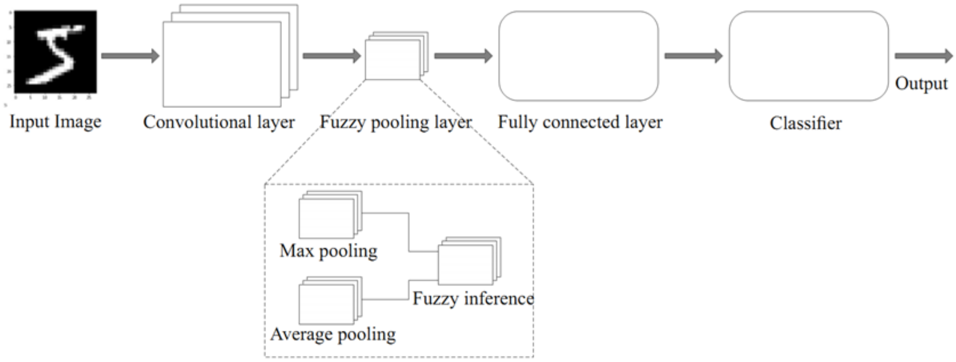

***Figure 2.** Architecture of a CNN with the fuzzy combined pooling operation.*

Consider a set of feature maps β^m with a dimension of $w \times h$, which stands for a input volume $w \times h \times d$ and $\beta = \{\beta^m \mid m = 1,2,...,d\}$. The proposed pooling method is sampled by a pooling window of size $f \times f$ with stride s. The number of spatial patches p^m can be extricated from feature maps β^m with $p = \{p^m \mid m = 1,2,...,d\}$. The number of patches ps which are received from an input volume β can be written as:

$$ps = \frac{(w - f + 2z_w)(h - f + 2z_h)}{2s + 2}, \tag{3}$$

where $z_w = (s-1)(w - s + f)/2$ and $z_h = (s-1)(h - s + f)/2$ present the zero-padding which is utilized in the patch extrication process on the width and height axis, respectively, of the input feature maps.

For an element p^m_{kij} ($i, j = \overline{1, f}$; $k = \overline{1, ps/d}$) of patches ps, values of min, max and average will be considered by pooling methods, respectively. Then two fuzzy sets $\tilde{x}_1$, $\tilde{x}_2$ with membership functions μ_1, μ_2 are utilized to combine the received min, max and average values. The membership functions of fuzzy sets, which are utilized for the fuzzification of the patches, can be chosen as follows:

$$\mu_1(x) = \begin{cases} 1 & \text{if } x < c^m_{\text{kij}} \\ \dfrac{1}{c^m_{\text{kij}} - a^m_{\text{kij}}}\left(x - a^m_{\text{kij}}\right) & \text{if } c^m_{\text{kij}} \le x \le a^m_{\text{kij}} \\ 0 & \text{if } x > a^m_{\text{kij}} \end{cases}; \tag{4}$$

$$\mu_2(x) = \begin{cases} 0 & \text{if } x < c^m_{\text{kij}} \\ \dfrac{1}{a^m_{\text{kij}} - c^m_{\text{kij}}}\left(x - c^m_{\text{kij}}\right) & \text{if } c^m_{\text{kij}} \le x \le a^m_{\text{kij}} \\ 1 & \text{if } x > a^m_{\text{kij}} \end{cases}. \tag{5}$$

Where c^m_{kij} and a^m_{kij} are min and max pooling function, respectively; $x = A^m_{kij}$ is the average pooling function, which presents the input of fuzzy inferences. The selection of linear membership functions in this paper is motivated by their simplicity, interpretability, and low computational cost. Linear functions allow efficient implementation within CNN architectures without introducing additional complex parameters. Although nonlinear membership functions such as Gaussian or sigmoid may provide higher flexibility in modeling uncertainty, they typically require additional parameter tuning and increase computational overhead.

Fuzzy inferences of the proposed fuzzy pooling can be considered by the rules of Takagi-Sugeno model:

R1: If the input x is $\tilde{x}_1$ then the output is $p^m_{kij} = A^m_{kij}$;

R2: If the input x is $\tilde{x}_2$ then the output is $p^m_{kij} = a^m_{kij}$.

The fuzzy inference system is deliberately designed with linear membership functions to ensure simplicity, interpretability, and computational efficiency. The fuzzy rules adaptively balance minimum, average, and maximum statistics based on local activation distributions, enabling smooth transitions between pooling behaviors.

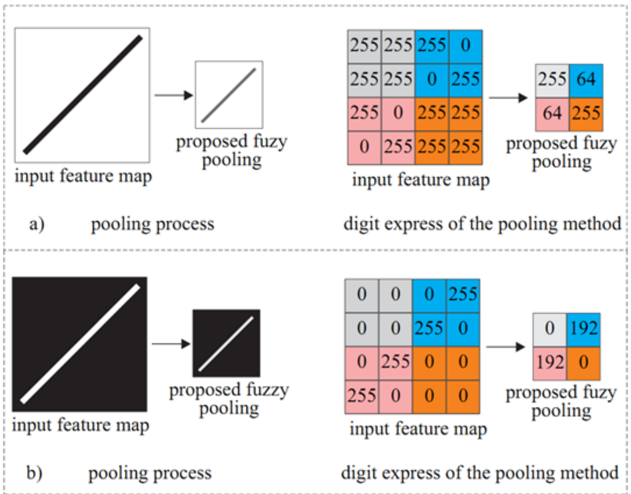

Figure 3. *Illustration of the proposed fuzzy pooling.*

Using the center of gravity for defuzzification the output values of the proposed fuzzy pooling can be expressed as follows:

$$F_{kij}^{m} = \frac{h_1 A_{kij}^{m} + h_2 a_{kij}^{m}}{h_1 + h_2}, \tag{6}$$

where $h_1 = \mu_1(A_{kij}^{m})$ and $h_2 = \mu_2(A_{kij}^{m})$ are the membership functions of the $k-th$ fuzzy set.

Figure 3 provides a visual comparison of the proposed fuzzy pooling method with max and average pooling. In this illustration, the average value of a feature patch is used as input to the fuzzy system, while the minimum and maximum values define the membership functions. Through fuzzy inference, the method adaptively combines the average and maximum pooling outputs, enabling smooth transitions between different pooling behaviors and effectively capturing both salient and contextual features.

The proposed fuzzy pooling method provides a balanced mechanism for feature aggregation by combining the strengths of conventional pooling operations. Specifically, max pooling is effective in preserving strong local activations but may ignore contextual information, whereas average pooling captures global context but may suppress discriminative features. By incorporating fuzzy inference, the proposed method adaptively adjusts the contribution of these pooling strategies based on the distribution of feature values. This enables a more flexible representation that preserves both salient and contextual information, which is beneficial for classification tasks.

3. RESULTS AND DISCUSSION

To evaluate the effectiveness of the proposed fuzzy combined pooling method, image classification experiments were conducted on standard datasets. The CNN models employed the same architecture, only varying the pooling methods, including max pooling, average pooling, and fuzzy pooling. The models were built using the Keras Sequential API, trained on an NVIDIA GTX1650 GPU with the Adam optimizer, categorical cross-entropy loss function, a batch size of 128, and a fixed number of epochs. No additional regularization techniques were applied in order to ensure fair comparison and reproducibility among the methods.

3.1. MNIST classification

The MNIST dataset is a typical dataset of handwritten images, which serves as the basis for standard machine leaning algorithms. The MNIST dataset is very appropriate for training and testing machine learning algorithms on real-world data. The MNIST dataset consists of a set of handwritten images of 60000 training data and 10000 testing data which have size of 28×28 in different dimension. Convolutional Neural Networks were utilized for recognition with MNIST dataset in some previously researches [10, 17]. There are different CNN models for classification on the MNIST dataset. In this paper, we construct a custom CNN for MNIST classification, as depicted in Figure 4, where the proposed fuzzy pooling method replaces the conventional pooling layers. The proposed CNN architecture for classification on MNIST dataset in this paper is constructed including a total of 8 layers. The first layer of the proposed model for MNIST classification is the convolution layer, which uses 64 kernels in 3×3 dimensions. Subsampling process in the second layer is implemented by the pooling layer with size of 2×2. The convolution layer in the third layer contains 128 kernels in 3x3 dimensions. In the fourth layer, the pooling layer is utilized with subsampling process in size of 2×2. The fifth and sixth layers continue to be constructed from layers of convolution with 256 kernels in 3×3 sizes and following a 2×2-dimension pooling layer. Two dense layers are built in final layers of the model where the "softmax" activation functions are utilized output classification distribution of probability for each class. Before feeding the data into the dense layers, the output of the pooling layer is flattened to combine all the found local characteristics in the previous convolutional layers and convert the received feature maps into a single 1D vector, which is needed to use fully connected layers after some convolutional and pooling layers. In convolution layers, "ReLU" activation functions are utilized to add nonlinearity to the network.

Different pooling methods, such as Max, Average and fuzzy, are arranged in the pooling layers of the model in Figure 4 to evaluate the performance of the proposed fuzzy pooling method.

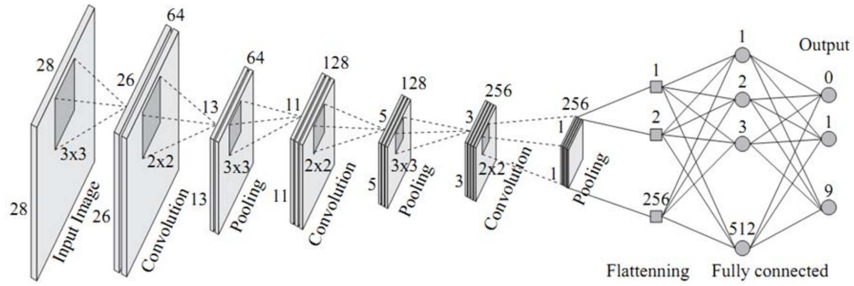

Figure 4. *Architecture of CNN model for the MNIST handwritten digit classification problem.*

Experimental results were obtained over 40 epochs. The training loss curves for different pooling methods on the MNIST dataset are presented in Figure 5. It can be observed that the proposed fuzzy pooling converges faster and achieves a lower final loss compared to max and average pooling, indicating more effective learning of discriminative features on this relatively simple dataset. Table 1 reports the classification performance of conventional and alternative pooling methods in terms of accuracy, precision, recall, and F1 score. The proposed fuzzy pooling method outperforms both max and average pooling across all metrics. This improvement can be attributed to the fuzzy mechanism's ability to preserve strong local features while maintaining contextual information. In particular, the higher F1 score demonstrates a better balance between precision and recall, indicating that the model produces more reliable and discriminative classification results.

Table 1. *Classification performance of the proposed CNN architecture with the MNIST dataset using different pooling methods.*

Method	**Classification performance (%) of the proposed CNN models**			
	Accuracy	**Precision**	**Recall**	**F1-Score**
Max pooling	98.09	97.84	97.19	98.13
Average pooling	97.4	97.41	96.16	96.27
Proposed pooling	99.42	98.89	97.65	99.43

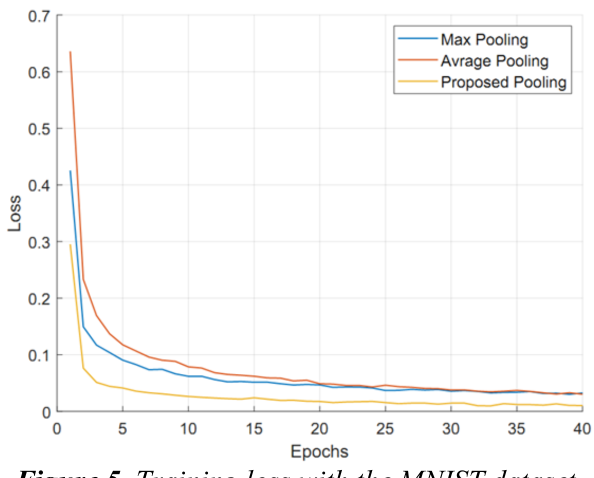

Figure 5. *Training loss with the MNIST dataset.*

3.2. CIFAR-10 classification

The CIFAR-10 dataset is one of the typical sample sets for machine learning algorithms in the Tiny Images dataset. This dataset consists of 60000 32×32 color images containing one of the 10 object classes, with 6000 RGB color images for each class, in which the ratio of 5:1 is chosen for training and testing purposes. The dataset is organized into five training batches and one testing batch with 10000 images in each batch. In each training batch, 1000 images are randomly selected for the testing batch. The training batches are also randomly selected from the remaining images.

For the CIFAR-10 classification, a CNN model with 12 layers is utilized with different pooling methods. The model contains four convolutional layers in 1, 2, 4, 5, 7, 8-th layers with 32, 32, 64, 64, 128, 128 convolution kernels, which have a kernel size of 3×3. Three pooling operations are implemented in 3, 6 and 9-th layers using different methods, such as max pooling, average pooling or proposed fuzzy pooling. The CNN model for CIFAR-10 classification is shown in Figure 6, demonstrating that the proposed pooling method can be effectively integrated into deeper architectures. Experimental results were obtained over 50 epochs, and the training loss curves for different pooling methods are presented in Figure 7. It can be observed that the proposed fuzzy pooling achieves a more stable convergence behavior and slightly lower final loss compared to max and average pooling. This indicates that the method enhances learning stability and improves feature representation in more complex classification tasks such as CIFAR-10. The classification results of the pooling methods are presented in Table 2. The proposed fuzzy combined pooling method demonstrates superior performance compared to traditional pooling methods in terms of accuracy, precision, recall, and F1-Score.

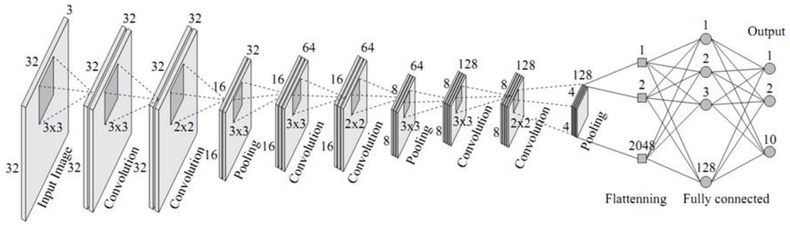

Figure 6. *Architecture of CNN model for image classification with CIFAR-10 dataset.*

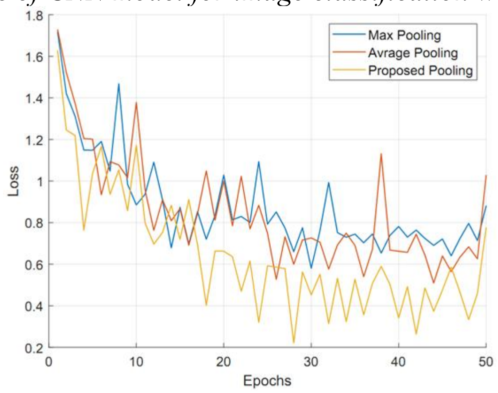

Figure 7. *Training loss with the CIFAR-10 dataset.*

Table 2. *Classification performance of the proposed CNN architecture with the CIFAR-10 dataset using different pooling methods.*

Method	**Classification performance (%) of the proposed CNN models**			
	Accuracy	**Precision**	**Recall**	**F1-Score**
Max pooling	75.23	83.13	69.61	75.95
Average pooling	78.16	85.81	72.36	77.08
Proposed pooling	79.71	87.02	75.39	78.52

In addition to classification performance, the computational cost of the proposed fuzzy pooling method was analyzed. Compared to conventional max and average pooling, the method introduces additional operations, including evaluation of linear membership functions and Sugeno fuzzy inference for weighted averaging. Although these operations slightly increase computational complexity and training time, the use of simple linear membership functions and straightforward Sugeno rules keeps the overhead minimal and acceptable. Experiments conducted on the same hardware configuration indicate that training time remains comparable to traditional pooling methods, demonstrating the practical efficiency of the proposed approach. For practical implementation, the proposed fuzzy pooling can be integrated as a custom layer in popular deep learning frameworks such as Keras or PyTorch. This integration does not require modifications to the overall CNN architecture, allowing direct application to existing models. The combination of low computational overhead and enhanced feature aggregation makes the method suitable for real-world applications, including image classification, and real-time processing tasks.

4. CONCLUSIONS

In this paper, an alternative fuzzy combined pooling method for CNN architectures is presented by combination of traditional pooling methods such as max and average using fuzzy logic inference. Experiments conducted on publicly available typical datasets have shown the advantages of the proposed alternative fuzzy combined pooling method, which significantly increases the classification performance of CNNs. Using the fuzzy pooling methods makes CNNs possible to overcome the disadvantages of conventional pooling methods, such as the removal of details from an image or the dilution of pertinent details from an image. Fuzzy inference allows combining low-level features from the data of images as edges or points when using the max pooling method, with the smooth features presented in the impact region covered by the feature maps in the average pooling method. However, analyzing the influence of parameters in using fuzzy inference has not been mentioned in the paper and it is also a future work to improve the performance of the fuzzy inference for pooling operation in CNNs.

Acknowledgement: *This research is supported by Ministry of Education and Training, Vietnam (Grant number: B2025-TDV-05).*

REFERENCES

[1]. R. Bhargava, N. Arivazhagan & K.S. Babu, *"Hybrid RMDL-CNN for speech recognition from unclear speech signal"*. Int J Speech Technol 28, 195–217, (2025).

[2]. S. Chinnaiyan, A.B. Haffishthullah, S. Naveen, et al. *"Optimized graph convolutional shunted self-attention neural network for multilingual speech-to-text training using cross-language voice conversion of speech representations"*. Int J Speech Technol 29, 22, (2026).

[3]. M. Bojarski et al., "End-to-end learning for self-driving cars," arXiv preprint, arXiv:1604.07316, (2016).

[4]. A. G. Gad, "*Particle swarm optimization algorithm and its applications: A systematic review,*" Archives of Computational Methods in Engineering, Vol. 29, No. 5, pp. 2531–2561, (2022).

[5]. Q. Li et al., "*EEG-based anxiety emotion classification using an optimized convolutional neural network and transformer,*" Signal, Image and Video Processing, Vol. 19, pp. 501–509, (2025).

[6]. J. Zhang et al., "*PCB defect recognition by image analysis using deep convolutional neural network,*" Journal of Electronic Testing, Vol. 40, No. 5, pp. 657–667, (2024).

[7]. P. S. Geidarov, "*Analytical calculation of weights of convolutional neural networks,*" Optical Memory and Neural Networks, Vol. 33, No. 2, pp. 157–177, (2024).

[8]. X. Lin et al., "*LIBS feature variable extraction method based on convolutional neural network,*" Journal of Applied Spectroscopy, Vol. 92, No. 1, pp. 218–224, (2025).

[9]. C. Tang et al., "*DeepSCNN: A simplicial convolutional neural network for deep learning,*" Applied Intelligence, Vol. 55, pp. 281–295, (2025).

[10]. M. Zhu et al., "*ShuffleNeXt: Modern lightweight convolutional neural network architecture,*" Pattern Analysis and Applications, Vol. 27, No. 4, pp. 123–135, (2024).

[11]. C. Y. Lee, P. Gallagher, and Z. Tu, "*Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree pooling,*" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 4, pp. 863–875, (2017).

[12]. R. Nirthika, S. Manivannan, and A. Ramanan, "*An experimental study on convolutional neural network-based pooling techniques for the classification of HEp-2 cell images,*" Proc. of 10th International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, (2021).

[13]. M. S. Greeshma and V. R. Bindu, "*Single image super-resolution using fuzzy deep convolutional networks,*" Proc. of International Conference on Technological Advancements in Power and Energy (TAP Energy), Kollam, India, (2017).

[14]. M. M. Hasan et al., "*FP-CNN: Fuzzy pooling-based convolutional neural network for lung ultrasound image classification with explainable AI,*" Computers in Biology and Medicine, Vol. 165, Article no. 107407, (2023).

[15]. S.B. Tharun, S. Jagatheswari, *"A U-shaped CNN with type-2 fuzzy pooling layer and dynamical feature extraction for colorectal polyp applications"*. Eur. Phys. J. Spec. Top. 234, 2627–2635 (2025).

[16]. M. Sarkar, A. Mandal, *"Atrous convolution and fuzzy pooling-based multi-channel CNN model for accurate follicle segmentation"*. Evolving Systems 17, 37 (2026).

[17]. C. Ozdemir, Y. Dogan, and Y. Kaya, "*A new local pooling approach for convolutional neural networks: Local binary pattern pooling,*" Multimedia Tools and Applications, Vol. 83, No. 12, pp. 34137–34151, (2023).

TÓM TẮT

Cải thiện hiệu năng mạng nơ ron tích chập bằng phương pháp gộp dựa trên logic mờ

Trong các bài toán nhận dạng và phân loại ảnh tự động, mạng nơ ron tích chập (CNN) được sử dụng rộng rãi, trong đó phép gộp (pooling) đóng vai trò quan trọng trong việc tổng hợp và làm nổi các đặc trưng dữ liệu ảnh. Bài báo này đề xuất một phương pháp gộp (pooling) dựa trên logic mờ, tích hợp các phản hồi đặc trưng tối thiểu, trung bình và tối đa thông qua một cơ chế suy luận Takagi Sugeno đơn giản. Khác với các phương pháp gộp truyền thống, cách tiếp cận được đề xuất mô hình hóa một cách tường minh tính bất định trong quá trình tổng hợp đặc trưng, từ đó cho phép đạt được sự cân bằng tốt hơn giữa các đặc trưng cục bộ nổi bật và ngữ cảnh, nhằm bảo toàn thông tin quan trọng của ảnh. Hiệu quả của phương pháp được đánh giá trên các bộ dữ liệu MNIST và CIFAR-10. Kết quả thực nghiệm cho thấy phương pháp gộp mờ được đề xuất đạt hiệu quả phân loại vượt trội so với các phương pháp gộp truyền thống, đồng thời vẫn duy trì độ phức tạp tính toán tương đương. Phương pháp của bài báo có thể áp dụng hiệu quả trong các bài toán phân loại ảnh dựa trên mạng nơ ron tích chập.

Từ khoá: CNN; Học sâu; Logic mờ; Lớp gộp; Phân lớp.