ISSN 2007-9737

Designing Optimal CNNs Architectures Using Metaheuristic Algorithms Applied to the Classification of Alzheimer's Disease

Claudia I. Gonzalez*

TECNM/Tijuana Institute of Technology,
Division of Graduate Studies and Research,
Mexico

cgonzalez@tectijuana.mx

Abstract. Convolutional Neural Networks are extensively utilized across various industries, proving to be highly effective for tasks such as image or video processing, pattern recognition and classification. However, the design of CNN architectures presents significant challenges, particularly in determining the optimal CNN parameters. CNN architectures comprise numerous parameters, and their configurations can produce diverse classification results when applied to the same tasks. Typically, setting hyper-parameter values involves a complex search process, often relying on random search, extensive testing, or manual adjustment. To address this challenge, this study proposes the analysis and implementation of two meta-heuristic approaches: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithms. These approaches aim to automatically design optimal CNN architectures and enhance their performance. The optimized architectures are specifically employed in the classification of neurodegenerative diseases, with a focus on Alzheimer's image datasets.

Keywords. PSO, GA, Alzheimer classification, optimal convolutional neural networks, CNN optimization.

1 Introduction

Artificial Intelligence (AI) encompasses a wide range of technologies intended to enable machines to perform tasks that typically require human intelligence. Within AI, deep neural networks represent a subset of algorithms inspired by the structure and function of the human brain. Among these networks, convolutional neural networks (CNNs) stand out, which comprise multiple convolutions; these networks are adept at learning complex patterns and relationships in data through layers of interconnected nodes.

Their integration into AI systems has led to significant advancements in various domains, including robotics, computer vision, natural language processing and healthcare [1-2]. Although CNNs have wide application in industry, their architectural design presents challenges. These include managing high computational costs associated with information processing and determining optimal CNN parameters suitable for each individual problem [3]. Typically, setting hyper-parameter values involves a complex search process, often employing random searches, extensive testing, or manual adjustments.

To address this challenge, researchers have proposed implementing evolutionary computation methods to automatically design optimal CNN architectures and enhance their performance [4-6].

In the literature, there exists a plethora of meta-heuristic techniques utilized for optimizing CNN hyper-parameters, such as the ACO [7], FGSA [8], Whale optimization algorithm [9], Harris Hawks Optimization (HHO) [10], harmonic search (HS) [11], microcanonical optimization algorithm [12] and the differential evolution (DE) [13] to list a few.

The genetic algorithms (GA) algorithm also has been employed to automatically design CNN architectures, achieving favorable results. In [14] GA is used to optimize the CNN parameters and this is applied to Arabic Text Classification improving the accuracy of 4 to 5%.

In [15] the authors proposed a new GA for the CNN optimization architectures, the methodology is applied in three classification problems, including the Caltech256, MNIST and CIFAR10 datasets. In [16], an Optimizing CNN by Using GA is applied for COVID-19 Detection in Chest X - Ray Image. On the other hand, the PSO algorithms have been also applied to optimize CNN architectures and in a plethora of machine learning domains [17]. In [18] and [19], PSO is employed for the automatic design of CNN architectures; these strategies are evaluated on benchmark datasets, and the achieved results are comparable to state-of-the-art approaches.

In [20], a hybrid OLPSO (orthogonal learning PSO) algorithm is implemented to find the optimal number of CNN hyperparameters and it is applied to detect and diagnose plant diseases. In [21] and [22], the original PSO is used for optimizing CNNs architectures; in [21] is applied to the CIFAR100, ILSVRC-2012 and CIFAR-10 datasets, and in [22] CNNs are employed for identifying cancerous nodules in computed tomography scans of the lungs.

The mentioned studies highlight the advantages offered by GA and PSO in the optimization process, enhancing performance across a variety of tasks. This research work contributes by implementing a hybrid methodology that employs both PSO and GA algorithms to discover the optimal parameters for CNN architectures. The parameters under consideration include the number of convolution layers, the filter size utilized in each convolutional layer, the number of convolution filters, and the batch size. Both approaches aim to explore more diverse architectures generated by GA and PSO through random searches.

The optimized architectures are tested in the Alzheimer MRI Dataset [23]. Alzheimer's disease is a progressive neurodegenerative disease that causes between 60 and 80 percent of dementia cases in the world; it is suffered by elderly people and is an increasing problem with the aging of the population. MRI can help us find duplicates in the brain that are related to mild cognitive impairment and can provide information about patients with mild cognitive impairment could develop Alzheimer's disease. Despite being a disease that has no cure, early diagnosis and treatment can be of great help in stopping the progressive advance of the disease.

The importance of this work lies in developing a tool that can support specialists in the health area in making decisions for the identification of this disease and its classification or level of advancement according to the analysis of medical resonance images magnetic.

The structure of this paper is as follows: Section 2 provides an overview of convolutional neural networks and introduces the theories behind GA and PSO, covering the main definitions. Section 3 outlines the methodology for developing two optimization approaches, GA-CNN and PSO-CNN. Section 4 discusses the analysis of experimental results obtained from the optimized architectures. Lastly, Section 5 presents significant conclusions and outlines future works for research.

2 Background

2.1 Convolutional Neural Networks

These types of neural networks represent a specialized category of deep learning architectures specifically designed for computer vision applications, such as pattern recognition in images and videos, object detection, image classification, and among others. Inspired by the structure and function of the human visual system, CNNs prove highly effective for processing images by autonomously extracting hierarchical and pertinent features from images, making them exceptionally effective in this domain.

Several pre-designed CNN architectures have been proposed and achieved state-of-the-art performance in various computer vision tasks. In [24] the first successful CNN architecture was introduced (LeNet-5), which was primarily used for handwritten digit recognition tasks. LeNet-5 demonstrated the effectiveness of convolutional layers, pooling layers, and fully connected layers in learning hierarchical representations of visual data. AlexNet, introduced in [25], marked a breakthrough for CNNs when researchers won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a significant margin.

This success sparked a resurgence of interest in deep learning and CNNs, leading to rapid advances in the field. In subsequent years, numerous advancements and CNN architectures

ISSN 2007-9737

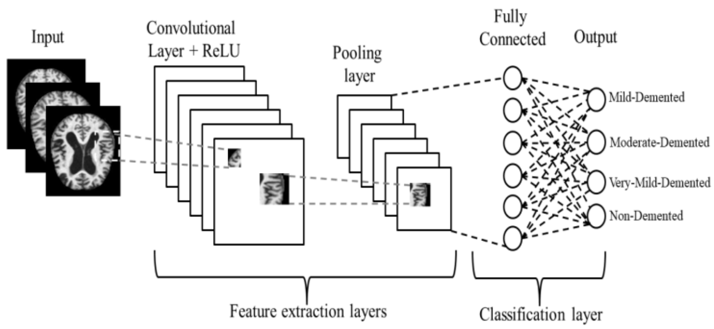

Fig. 1. Basic CNN architecture

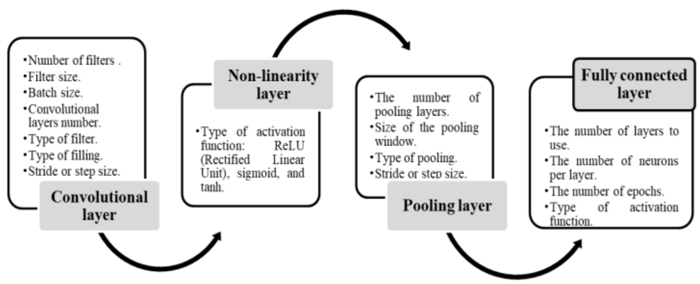

Fig. 2. Hyper-parameters of a CNN

were introduced, including VGGNet [26], GoogLeNet (Inception) [27], ResNet [28], and more. These architectures varied in terms of depth, parameter efficiency, computational complexity, and performance on various computer vision tasks.

The fundamental CNN architecture illustrated in Fig. 1 includes the following layers: input layer, convolutional layers, activation layer, pooling layer, and fully connected (dense) Layer. Additional layers or modifications can be introduced based on specific requirements or variations in network designs.

CNNs are considered powerful tools for visual data analysis, exhibiting remarkable performance in various real-world applications. Their capacity to autonomously learn hierarchical features from raw data has become indispensable in the field of computer vision, with ongoing research aimed at further enhancing their capabilities and efficiency.The hyperparameters of a CNN are of great relevance since the performance and effectiveness of the model depends on them. Experimenting with various combinations of hyperparameters can result in significant computational costs, requiring extensive computational resources and time for training and evaluation. CNNs often have high-dimensional parameter spaces, exploring this high-dimensional space to find the optimal architecture can be challenging and requires sophisticated optimization techniques.

In this sense an optimization approach using GA and PSO is presented to improve the classification accuracy and reduce the computational cost. The most relevant parameters that can be optimized in each CNN layer are presented in Fig. 2, including some others parameters such as the learning rate used to determine the step size of the optimization algorithm during training, the optimizer algorithm (Adam, stochastic gradient descent (SGD), and RMSprop) and the dropout which is a regularization technique implemented to prevent overfitting by randomly dropping neurons from the network during training [4].

2.2 Particle Swarm Optimization (PSO)

PSO, introduced by Eberhart and Kennedy in 1995 [29], is a robust optimization method known for its

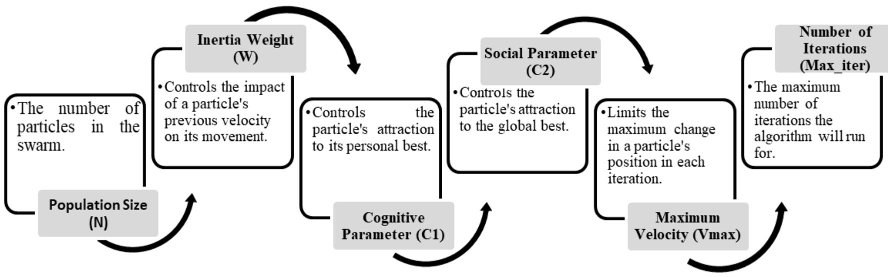

Fig. 3. PSO parameters

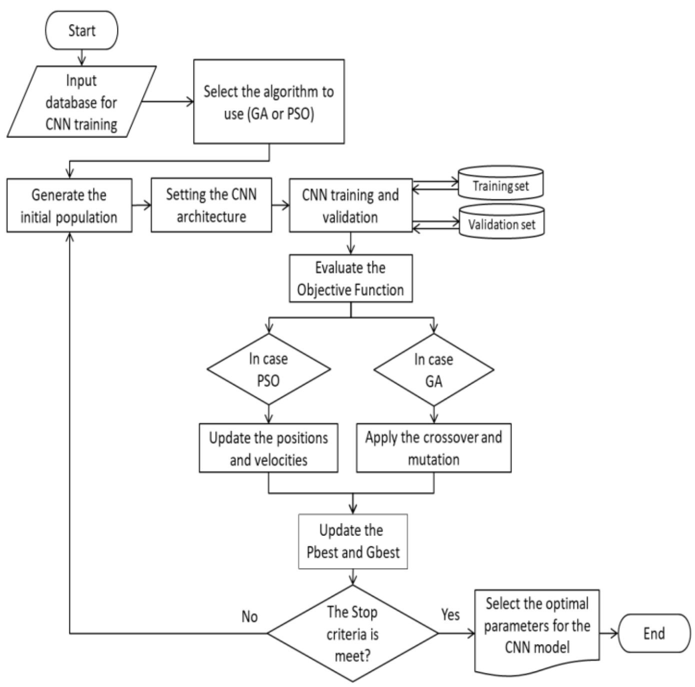

Fig. 3. Steps to optimize the CNN using GA and PSO algorithms

simplicity and efficiency in exploring solution spaces. This stochastic algorithm, inspired by swarm intelligence and the collective movements of birds and fish, involves individuals (particles) exploring a multi-dimensional search space to discover the best possible solutions.

In the algorithm, each particle denotes a potential solution and has a defined position and velocity within the search space. The main parameters of PSO are presented in Fig. 3. Additionally, other key concepts involved in the implementation include:

– Global best (gbest): it is the best solution found by any particle in the entire swarm. The swarm is a collection of particles, each of which explores the search space to find optimal solutions.
– Personal best (pbest): it is the best solution found by an individual particle in its own history.
– Fitness function: based on predefined criteria, this evaluates the quality of a solution.
– Inertia weight: it is a parameter to control the impact of a particle's velocity on its movement.

- Social and cognitive components: these parameters influence how the speed of a particle is updated based on its personal best and its global best.

2.3 Genetic Algorithms

In recent decades, Genetic Algorithms have gone from abstract theories to effective tools for problem solving. Their capacity to effectively navigate solution spaces and identify near-optimal solutions has made them indispensable across various domains. It is a search heuristic inspired by the process of natural evolution, which was proposed by the computer scientist John Holland in [30].

The more recent applications are focusing in 1) Parameter Tuning, where the research implements GA for tuning parameters to different problem domains and improve performance [14-16]. 2) Deep Learning Integration, to explore synergies between GAs and deep learning techniques for solving complex optimization and search problems. The principal concepts manage in the GA are the following:

- Chromosome: A potential solution represented in the form of a string of genes. Each gene typically represents a parameter of the problem being solved.
- Population: A collection of chromosomes representing potential solutions to the problem at hand.
- Fitness Function: A function that evaluates how good or bad a particular solution (chromosome) is. It quantifies the quality of the solution relative to other solutions.
- Selection: it is a process of selecting individuals from the population for reproduction based on their fitness. Those with higher fitness are prioritized for selection
- Crossover: it is a genetic operator utilized to combine the genetic information of two parent chromosomes, producing one or more offspring.
- Mutation: this operator is applied to introduce genetic diversity by altering one or more genes in a chromosome randomly.

3 CNN Architecture Optimized Using GA and PSO

This section introduces two optimization strategies, PSO-CNN and GA-CNN, which apply the PSO and GA algorithms to optimize the parameters of CNN architectures. As previously mentioned, varying CNN parameter values can lead to different results for the same task, underscoring the importance of finding optimal architectures.

The parameters selected for optimization in this work include the number of convolutional layers, the filter size for each convolutional layer, the filters number and the batch size. All of these parameters are important and impact the results, apart from the fact that it is a challenge for researchers or developers in these areas how to determine them, since they depend on the type of problem or application to be solved.

Figure 4 illustrates the general steps to optimize the CNN using GA and PSO algorithms; the procedure is explained below:

- Input the image dataset. This step involves choosing the dataset that will be processed and classified by CNN.
- Select the optimization algorithm to be used (GA or PSO).
  a) In the case of GA, generate the population for GA algorithm. The GA parameters are set to include the generations number, the number of populations, crossover and mutation values. The structure of the chromosome is presented in Fig. 5, and the parameters to execute the GA in Table 4.
  b) In the case of PSO, the parameters are setting with the values presented in Table 5, and the algorithm is ready to generate the particle population; each particle takes the structure presented in Fig. 5 and the properties of Tables 1 and 2.
- Setting the CNN architecture. The CNN is initialized with the parameters presented in Table 3 in conjunction with the parameter obtained by the GA or PSO in the optimization process. In this step, the CNN is ready for the training phase.
- CNN training and validation involves the CNN analyzing the Alzheimer's dataset using images for training, validation, and testing

$$Objective\ function = Classification\ accuracy \tag{1}$$

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8

Fig. 5. Structure of the particle (PSO-CNN) and the chromosome (GA-CNN)

Table 1. Search space compositions for the chromosome and particle structure

Coordinate	Layer	Hyper-Parameter	Search Space
p_1	--	Batch size in the training	[32, 256]
p_2	--	Convolutional layer number	[1, 3]
p_3	Layer 1	Number of filters	[32, 128]
p_4	Layer 1	Filter size	[1, 4]
p_5	Layer 2	Number of filters	[32, 128]
p_6	Layer 2	Filter size	[1, 4]
p_7	Layer 3	Number of filters	[32, 128]
p_8	Layer 3	Filter size	[1, 4]

Table 2. Filter size of the convolutional layer for the positions p_4, p_6, p_8

Value	Range
1	[3, 3]
2	[5, 5]
3	[7, 7]
4	[9, 9]

purposes. The CNN then returns the accuracy to PSO and GA as the objective function that both algorithms evaluate.

– Evaluate the objective function. The GA or PSO algorithms process the function expressed in (1) to obtain the best value.

– Update GA or PSO parameters.

  a) In the case of GA, in each generation, apply the crossover and mutation.

  b) In the case of PSO, this step consists of updating the velocity and position of each particle in each iteration.

– The process iterates, considering the evaluation of all solutions until the stopping criteria are met. For PSO, the stopping criterion is determined by the iterations number, and for GA, it is given by the number of generations.

– In the last step, the process obtains the optimal solution represented by Gbest which contains the optimal parameters to generate the CNN architecture.

3.1 GA-CNN and PSO-CNN Optimization Process

The Chromosome of real-numbers used in the GA-CNN approach and the particle structure of the PSO-CNN consists of eight positions; each one represents the parameter to be optimized, as shown in Fig. 5. Table 1 details the composition of the chromosomes or particles, describing the data that is controlled in each position and the corresponding search space.

As indicated to Table 1, positions p_4, p_6 and p_8 are indices that can take integer values between 1 to 4. According on the values generated by the GA or PSO algorithm, a mapping is performed using the values expressed in Table 2.

In this approach, based on the values to be optimized, the p_1 position is applied to control the batch size used in the training phase. The p_2 position determines the deep of the CNN in this case represented by the number of convolution layers; this is also used to control the activation of the positions p_3 to p_8.

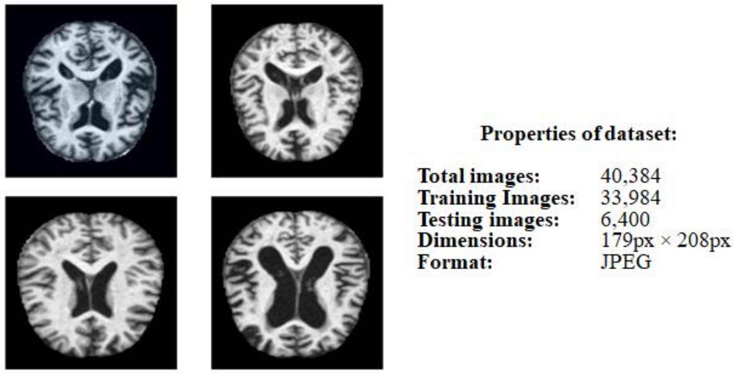

Fig. 6. Alzheimer MRI Dataset

Table 3. CNN training parameters

CNN parameters	
Epochs	20
Learning function	Adam
Non-linearity activation function	ReLU
Activation function (classifying layer)	Softmax

Table 4. Initial parameters used for the GA algorithm

GA parameters	
Population	15
Genera tions	30
Crossover	0.85
Mutation	0.02

Table 5. Initial parameters used for the PSO algorithm

PSO parameters	
Particles (N)	15
Inertial weight (W)	0.85
Iterations	30
Cognitive constant (C1)	2
Social constant (C2)	2

In case of the algorithm result in a value of one for p_2 position, only the p_3 and p_4 positions are activated; these positions determine the number of filters used in the first convolutional layer and the filter size. In case of the algorithm generates a value of three for the p_2, the positions from p_3 to p_8 will be activated. These positions will generate the number of filters for the layer 1, 2 and 3 (p_3, p_5 and p_7), the filter size of layer 1, 2 and 3 (p_4, p_6 and p_8).

Each of these values is distinct, promoting the creation of more heterogeneous CNN architectures. This representation can grow and will be applied to any number of convolutional layers, generating deeper CNN.

If position p_2 takes a different value, the rest of the particle or chromosome can expand accordingly. The objective function used in the two optimizations process is expressed in (1) and this represents the classification accuracy returned by

Table 6. Results of the GA-CNN and PSO-CNN for the Alzheimer MRI dataset

No.	No. Layers	Layer 1		Layer 2		Layer 3		Batch Size	(%) Recogn. Rate
		No. Filters	Filter Size	No. Filters	Filter Size	No. Filters	Filter Size		
GA	3	32	[3 × 3]	128	[9 × 9]	128	[5 × 5]	256	99.35
PSO	3	128	[3 × 3]	128	[3 × 3]	128	[5 × 5]	128	99.41

Table 7. Summary of 30 experiments for GA-CNN and PSO-CNN

Optimization approach	Best	Mean
GA-CNN	99.35%	98.70%
PSO-CNN	99.41%	99.22%
Non-Optimized	98.43%	95.70%

the CNN after this is trained using the dynamical parameters obtained by the GA or PSO algorithms.

4 Experiments

This section outlines the database utilized in the case studies (Augmented Alzheimer MRI Dataset V2), the fixed parameters for configuring of the CNN, the GA and PSO algorithm; also, the results achieved through the two optimization methods employed (GA-CNN and PSO-CNN), and the comparative analysis with alternative approaches.

4.1 Alzheimer MRI Dataset

In this study the experiments were implemented with the database "Augmented Alzheimer MRI Dataset V2" [26], which is an extension of synthetic augmentation of another open-source database. The dataset contains 4 different classes of Alzheimer disease: Non-Demented, Moderate-Demented, Mild-Demented, and Very-Mild-Demented; an example of this dataset and the properties are illustrated in Fig. 6.

4.2 Static Parameters for the Configuration of PSO, GA and the CNN

Table 3 presents the static parameters used to train the CNN. The dynamic parameters optimized by GA and PSO are the batch size, convolutional layers number, the filter size and the number of filters; these are represented in Tables 1 and 2.

The fixed parameters considered for the GA are the population, generations, crossover and mutation; these are described in Table 4. The parameters considered in the PSO are presented in Table 5, which are given by the number of particles, the iterations, the inertial weight, and the cognitive and social constants.

The parameters mentioned above were defined under experimentation and considering that these are limited because the time to evaluate each solution is too long considering the time it takes to train, test and calculate the accuracy of the network.

4.3 Results after Application of the GA-CNN and PSO-CNN Approaches

This Section presents the results of the GA-CNN and PSO-CNN approaches applied in the Alzheimer MRI Dataset; this last was distributed in 70% for training phase and 30% for testing. The objective function is given by the accuracy, expressed in (1). The experimental results consist of 30 executions for each optimization model. Table 6 presents the best accuracy and architecture obtained by the GA-CNN and PSO-NN.

The optimal architecture discovered through GA-CNN consists of three convolutional layers: the first layer employs 32 filters of size 3 × 3, the second layer employs 128 filters of size 9 × 9, and the third layer employs 128 filters of size 5 × 5. The batch size utilized is 256. This configuration achieved a classification accuracy of 99.35% and a mean accuracy of 98.70%.

PSO-CNN results presented a three-layer CNN configuration. The first layer utilized 128 filters with a filter size of 3 × 3, the second layer also employed 128 filters with a filter size of 3 × 3, and the third layer utilized 128 filters with a filter size of 5 × 5; the batch size employed was 128. PSO-CNN achieved a classification accuracy of 99.41% and a mean accuracy of 99.22%.

Table 7 provides a summary of the 30 executions obtained by applying the two approaches the non-optimized architecture is also considered. According to these results we can notice that the PSO-CNN optimization approach reaches the best accuracy with a value of 99.41% and a mean of 99.22%, this over the GA-CNN and the non- optimized architecture.

5 Conclusions

In general, this paper presents two approaches for optimizing CNN architectures using the GA and PSO algorithms, applied to the classification of Alzheimer's disease. The main contribution consists in providing an automatic and dynamic way to obtain some of the CNN hyperparameters, including the number of convolutional layers, the filter size for each convolutional layer, the number of convolutional filters, and the batch size. In the experiments were considered a maximum of three layers, but the approach can be extended to cover more complex structures and add other parameters.

Based on the results obtained from the two optimization methodologies, we can conclude that accuracy improved in all case studies, demonstrating robust performance with minimal parameters. The results highlight the significance of using optimization algorithms to identify the optimal parameters for CNN architectures.

For future work, it is important to implement these approaches on other datasets related with neurodegenerative disease to validate their robustness and advantages and to contribute with models in this important medical area. Additionally, the methodology can be extended or explored using different metaheuristic algorithms and consider other parameters of complex CNN architectures.

Furthermore, in the evaluation of the objective function another objective could be considered where the depth and number of parameters obtained by the optimization algorithms can be penalized to achieve simpler CNN architectures.

Acknowledgments

We thank TECNM/Tijuana Institute of Technology and CONAHCyT for support with the finances with the grant number CF-2023-I-555.

References

1. **Hemanth, D.J., Deperlioglu, O., Kose, U. (2020).** Retracted Article: An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. Neural Computing & Applications, Vol. 32, No. 3, pp. 707–721.
2. **Wang, W., Gang, J. (2018).** Application of convolutional neural network in natural language processing. International conference on information Systems and computer aided education, IEEE, pp. 64–70. DOI: 10.1109/ICISCAE.2018.8666928.
3. **Simonyan, K., Zisserman, A. (2014).** Very deep convolutional networks for large-scale image recognition. preprint arXiv: 1409.1556.
4. **Liang, S.D. (2018).** Optimization for deep convolutional neural networks: How slim can it go? IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 4, No. 2, pp. 171–179. DOI: 10.1109/TETCI.2018.2876573.
5. **Sun, Y., Xue, B., Zhang, M., Yen, G.G. (2019).** Evolving deep convolutional neural networks for image classification. IEEE Transactions on Evolutionary Computation, Vol. 24, No. 2, pp. 394–407. DOI: 10.1109/ TEVC.2019.2916183.
6. **Baldominos, A., Saez, Y., Isasi, P. (2018).** Evolutionary convolutional neural networks: An application to handwriting recognition.

Neurocomputing, Vol. 283, pp. 38–52. DOI: 10.1016/j.neucom.2017.12.049.

7. **Byla, E., Pang, W. (2020).** Deepswarm: Optimising convolutional neural networks using swarm intelligence. Advances in Computational Intelligence Systems: Contributions Presented at the 19th UK Workshop on Computational Intelligence, Springer International Publishing, pp. 119–130.
8. **Poma, Y., Melin, P., González, C.I., Martínez, G. E. (2020).** Optimization of convolutional neural networks using the fuzzy gravitational search algorithm. Journal of Automation, Mobile Robotics and Intelligent Systems, Vol. 14, No. 1, pp. 109–120.
9. **Zhang, N., Cai, Y.X., Wang, Y.Y., Tian, Y.T., Wang, X.L., Badami, B. (2020).** Skin cancer diagnosis based on optimized convolutional neural network. Artificial intelligence in medicine, Vol. 102, pp. 101756. DOI: 10.1016/j.artmed.2019.101756.
10. **Gadekallu, T.R., Srivastava, G., Liyanage, M., Iyapparaja, M., Chowdhary, C.L., Koppu, S., Maddikunta, P.K.R. (2022).** Hand gesture recognition based on a Harris hawks optimized convolution neural network. Computers and Electrical Engineering, Vol. 100, pp. 107836. DOI: 10.1016/j.compeleceng .2022.107836.
11. **Lee, W.Y., Park, S.M., Sim, K.B. (2018).** Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. Optik, Vol. 172, pp. 359–367. DOI: 10.1016/j.ijleo.2018.07.044.
12. **Gülcü, A., Kuş, Z. (2020).** Hyper-parameter selection in convolutional neural networks using microcanonical optimization algorithm. IEEE Access, Vol. 8, pp. 52528–52540. DOI: 10.1109/ACCESS.2020.2981141.
13. **Wang, B., Sun, Y., Xue, B., Zhang, M. (2018).** A hybrid differential evolution approach to designing deep convolutional neural networks for image classification. In: **Mitrovic, T., Xue, B., Li, X. (eds)** AI 2018: Advances in Artificial Intelligence. Lecture Notes in Computer Science, Springer, Cham. Vol 11320. DOI: 10.1007/978-3-030-03991-2_24.
14. **Alsaleh, D., Larabi-Marie-Sainte, S. (2021).** Arabic text classification using convolutional neural network and genetic algorithms. IEEE Access, Vol. 9, pp. 91670–91685. DOI: 10.11 09/ACCESS.2021.3091376.
15. **Johnson, F., Valderrama, A., Valle, C., Crawford, B., Soto, R., Ñanculef, R. (2020).** Automating configuration of convolutional neural network hyperparameters using genetic algorithm. IEEE Access, Vol. 8, pp. 156139–156152. DOI: 10.1109/ACCESS.2020. 3019245.
16. **Mahendra, M.I., Kurniawan, I. (2021).** Optimizing convolutional neural network by using genetic algorithm for COVID-19 detection in chest X-ray image. International Conference on Data Science and Its Applications, pp. 135–140. DOI: 10.1109/ ICoDSA53588.2021.9617546.
17. **Fernández-Junior, E.F., Yen, G.G. (2019).** Particle swarm optimization of deep neural networks architectures for image classification. Swarm and Evolutionary Computation, Vol. 49, pp. 62–74. DOI: 10.1016/j.swevo.2019.05.010
18. **Sun, Y., Xue, B., Zhang, M., Yen, G.G. (2018).** A particle swarm optimization-based flexible convolutional autoencoder for image classification. IEEE transactions on neural networks and learning systems, Vol. 30, No. 8, pp. 2295–2309. DOI: 10.1109/TNNLS.2018. 2881143.
19. **Singh, P., Chaudhury, S., Panigrahi, B. K. (2021).** Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network. Swarm and Evolutionary Computation, Vol. 63, pp. 100863. DOI: 10.1016/j.swevo.2021.100863.
20. **Darwish, A., Ezzat, D., Hassanien, A. E. (2020).** An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. Swarm and evolutionary computation, Vol. 52, pp. 100616. DOI: 10.1016/j.swevo.2019.100616.
21. **Chang, J., Lu, Y., Xue, P., Xu, Y., Wei, Z. (2022).** Automatic channel pruning via clustering and swarm intelligence optimization for CNN. Applied Intelligence, Vol. 52, No. 15,

pp. 17751–17771. DOI: 10.1007/s10489-022-03508-1.

22. **De-Pinho-Pinheiro, C.A., Nedjah, N., de-Macedo-Mourelle, L. (2020).** Detection and classification of pulmonary nodules using deep learning and swarm intelligence. Multimedia Tools and Applications, Vol. 79, No. 21, pp. 5437–15465. DOI: 10.1007/s11042-019-7473- z.
23. **LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998).** Gradient-based learning applied to document recognition. Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324. DOI: 10.1109/5.726791.
24. **Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017).** ImageNet classification with deep convolutional neural networks. Communications of the ACM, Vol. 60, No. 6, pp. 84–90. DOI: 10.1145/3065386.
25. **Simonyan, K., Zisserman, A. (2015).** Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations Computational and Biological Learning Society, pp. 1–14.
26. **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015).** Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.
27. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
28. **Kennedy, J., Eberhart, R. (1995).** Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.
29. **Holland, J.H. (1992).** Genetic algorithms. Scientific American, Vol. 267, pp. 66–72.

Article received on 18/05/2024; accepted on 25/08/2024.
**Corresponding author is Claudia I. Gonzalez.*