Hyper-parameter optimization of convolutional neural network based on particle swarm optimization algorithm

Zainab Fouad[1], Marco Alfonse[2], Mohamed Roushdy[3], Abdel-Badeeh M. Salem[4]
[1,2,4]Department of Computer Science, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt
[3]Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, Egypt

Article Info

Article history:

Received Aug 6, 2021
Revised Oct 16, 2021
Accepted Oct 31, 2021

Keywords:

Convolutional neural network
Hyper-parameter optimization
Image classification
Machine learning
MINST
Particle swarm optimization

ABSTRACT

Deep neural networks have accomplished enormous progress in tackling many problems. More specifically, convolutional neural network (CNN) is a category of deep networks that have been a dominant technique in computer vision tasks. Despite that these deep neural networks are highly effective; the ideal structure is still an issue that needs a lot of investigation. Deep Convolutional Neural Network model is usually designed manually by trials and repeated tests which enormously constrain its application. Many hyper-parameters of the CNN can affect the model performance. These parameters are depth of the network, numbers of convolutional layers, and numbers of kernels with their sizes. Therefore, it may be a huge challenge to design an appropriate CNN model that uses optimized hyper-parameters and reduces the reliance on manual involvement and domain expertise. In this paper, a design architecture method for CNNs is proposed by utilization of particle swarm optimization (PSO) algorithm to learn the optimal CNN hyper-parameters values. In the experiment, we used Modified National Institute of Standards and Technology (MNIST) database of handwritten digit recognition. The experiments showed that our proposed approach can find an architecture that is competitive to the state-of-the-art models with a testing error of 0.87%.

Corresponding Author:

Zainab Fouad
Department of Computer Science
Ain Shams University, Egypt
Email: zainab.mfouad@cis.asu.edu.eg

1. INTRODUCTION

One of the major tasks in computer vision is the image classification task, which includes a lot of applications [1]. Convolutional neural network (CNN) is considered the base for most image classification methods and had shown an amazing performance and results [2], [3]. CNN also has been widely used in speech recognition, image processing including handwritten classification, scene classification, pattern recognition, and gesture recognition [4], [5]. CNN architecture has two fundamental blocks [6]. The first block is the convolutional layers that provide the features extraction process. The second part is a fully connected network that is responsible for the classification.

CNN is a type of deep networks that solves a lot of computer vision tasks. It was first introduced in the 1980s by Yann LeCun [7]. The blocks of CNN are the convolutional layers, pooling layers, and fully connected layers [8]. These blocks are shown in Figure 1.

The core block of CNN is the convolutional layer. This layer consists of a set of filters (kernels) that detect the features with random sizes. Each filter is convolved across the image by sliding it with a given horizontal step, then moves vertically with another step, until the full image has been checked. If we add more convolution steps so, more deep features can be recognized. For example, in the task of classification, CNN detects edges from pixels in the 1st stage, then detects simple shapes in the 2nd stage, and then detects higher-level features. The outputs of the convolution process are passed through a nonlinear activation function. The most common nonlinear activation function is the rectified linear unit (ReLU). Decreasing the dimensionality of the feature map and keeping the critical data is the task of the pooling layer as shown in Figure 2. This layer has a filter component called the kernel, the yield of a kernel is the (mean or max) value of the region.

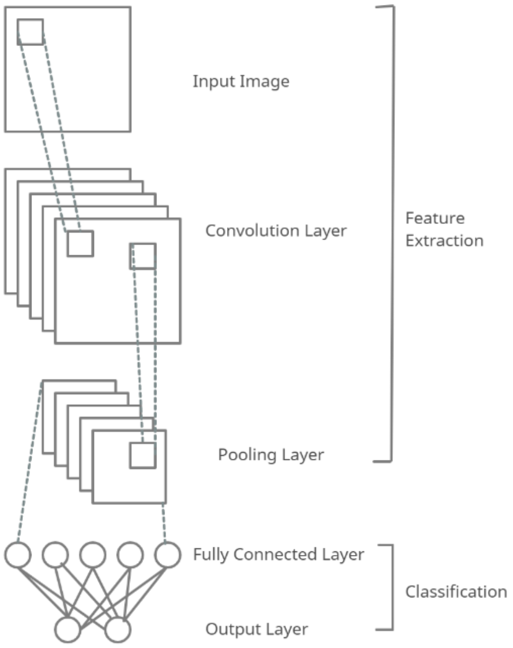

Figure 1. The CNN building blocks

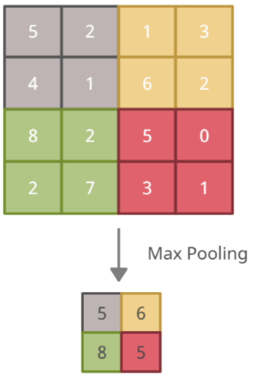

Figure 2. The pooling layer

After that, a fully connected layer is added to the network which can be any kind of classifier such as a multi-layer perceptron. When applying CNN-based methods, choosing the ideal hyper-parameters is a challenging task. From the machine learning perspective, there is no definite method to specify a certain structure of the network that will enhance the performance. Also, most architectures of the network are selected with prediction instead of a contemplate choice. We propose a method to learn the optimal CNN hyper-parameters values automatically that can lead to a competitive CNN structure and can deploy it to any application that uses CNN. This approach is based on PSO algorithm, only with 15 particles and 30 iterations, we can find architectures that achieve a testing error that is competitive to others designs and can be applied to most of the CNN architectures with no encoding strategy during computations or using parallel layers, it is just with simple CNN. Since the training of a CNN is a time-consuming task, we chose to use the PSO algorithm in the process of optimal CNN architectures searching; because its convergence is faster than genetic algorithms [9], [10]. Our contributions are:

- Introducing a CNN that uses optimized parameters in a simple way.
- Finding CNN architecture that achieves a testing error that is competitive to other designs.
- Using CNN architectures with no encoding strategy during computations or using parallel layers.
- First research that tests the CNN hyperparameters optimization problem with bee colony algorithm.

The organization of the paper is as follows; section 2 shows the related work. Section 3 shows the proposed method. The evaluation of the performance is performed using the MNIST dataset in section 4. The last section includes the conclusion.

2. RELATED WORK

Numerous researches had suggested many CNN architectures as AlexNet [11], also some of them increased the network depth to enhance the accuracy [12]. Others chose to add new inner configurations [13]. Although these models have demonstrated their productivity, a lot of them were designed manually. On the other hand, there are many attempts to design appropriate models such as Sehla and Afef [14] who attempted to optimize the CNN parameters based on genetic algorithm. More methods for optimization of the hyperparameters have been proposed by David *et al*. [15], Ilya and Frank [16], Francisco and Gary [17] who applied evolutionary algorithms (EAs). Also, several works have been done to optimize CNN hyper-parameters based on particle swarm optimization (PSO), Foysal et. al. applied a modified CNN form by optimizing only one CNN parameter which is convolution size [18]. The optimization was done by the model-based selection technique of particle swarm optimization. They used synthetic datasets and the classification accuracy was around 95. Sinha *et al.* [19] used PSO to optimize The CNN hyper-parameters of the first layer of a 13-layer CNN and got a classification error of 18.53% on CIFAR-10 dataset. Also, on the same dataset, they got an error of 22.5% of the 8-layer AlexNet. So, deeper CNN yields higher accuracy, but more computation cost is needed.

The parameters used in this paper are size of input image, filter size, and number of filters and they need an encoding strategy during the computations. Guo *et al.* [20] proposed that the configuration of CNN parameters is an optimization problem, and a distributed PSO (DPSO) was used to optimize it and get the best model of CNN globally and automatically. In DPSO, they designed the mix-variable encoding strategy and updated the operations to each particle so, each particle represents a CNN. Also, the distributed framework reduces the running time. Experimental results on MNIST dataset get an accuracy of 99.25% for PSO and 99.2% for DPSO. Yamasaki *et al.* [21] applied PSO to get the optimal parameters for CNN automatically. The best setting of the parameter was obtained for Alexnet with different five datasets. Accuracy was enhanced from 0.7% to 5.7% of the standard Alexnet-CNN. The global best parameters could not be guaranteed in this algorithm due to its randomness. Also, the best parameters change frequently and are not constant. Wang *et al.* [22] proposed three mechanisms, acceleration coefficients vectorizing for adaptation different ranges of CNN hyper-parameters, compound normal confidence distribution for exploration capability enhancing, and linear estimation scheme for fast fitness evaluation. PSO is used with these mechanisms to improve the quality of CNN hyper-parameters and less the computation cost. Experimental results on CIFAR-10 get a classification error of 8.67%.

3. RESEARCH METHOD

Swarm intelligence (SI) is based on the behavior of groups intelligence. The SI algorithms can powerfully find the optimal solution. The Bee colony algorithm is an optimization algorithm inspired by the behavior of honeybees to find the optimal solution and was proposed in 2005 [23], [24]. Kennedy and Eberhart developed the PSO algorithm in 2001 and it is considered one of the evolutionary algorithms [25]. PSO algorithm consists of several steps. First, initialization of particles (searching agents) positions (x) and the velocities (v). Second, insert the particles into a cost function to find local bests (pbest) and global best (gbest). The smallest cost for each particle is the pbest and the smallest cost among all the pbest is the gbest. The cost function is calculated based on the problem, such as an error that needs to be minimized or accuracy that needs to be maximized. Third, update the particles by (1) and (2) [25].

$$v_{n+1} = v_n + c_1 r_1 (p_{best} - x_n) + c_2 r_2 (g_{best} - x_n) \quad (1)$$

$$x_{n+1} = x_n + v_{n+1} \quad (2)$$

In (1) and (2), n is particle number, c1, c2 are learning factors to adjust each iteration step length, usually in practice equals 2 which gives best results in many problems, and it can be set with trails and errors, v is the particle velocity which based on pbest and gbest, x is the current particle (solution), r1() and r2() are random variables between (0,1).

To design a CNN model with an optimized parameter for our classification problem, we deployed the PSO algorithm to the CNN architecture. So, we have to solve a problem with numbers of factors that expresses the hyper-parameters of the CNN whereas guaranteeing a high classification accuracy. A CNN architecture is defined by various hyper-parameters. In this work, we will focus on the optimization of the convolution layer size and kernel size parameters that form the CNN structure. The fitness function is set to be the classification accuracy. So, we aim to find the ideal hyper-parameters values with higher accuracy and less error. Figure 3 shows a flowchart of the proposed method.

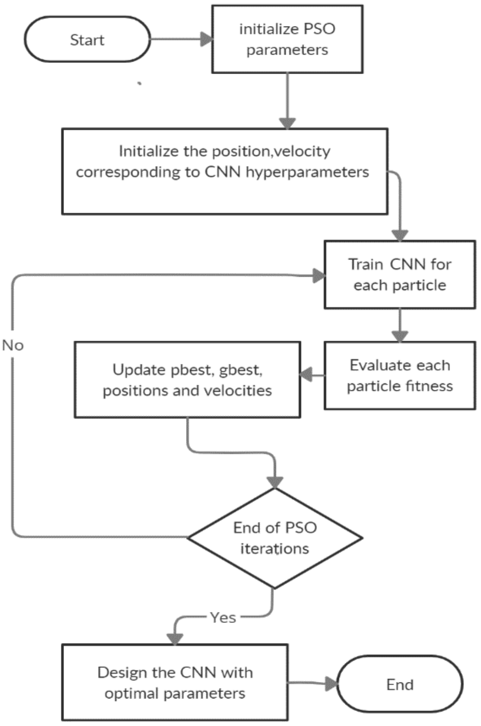
Start
initialize PSO parameters
Initialize the position,velocity corresponding to CNN hyperparameters
Train CNN for each particle
No
Update pbest, gbest, positions and velocities
Evaluate each particle fitness
End of PSO iterations
Yes
Design the CNN with optimal parameters
End

Figure 3. Proposed method flowchart

We define the hyper-parameters which are a set that contains first, the convolutional layer's parameters which contains the filters number and the filter size, second, fully connected layer's parameters which contains the size of the layer. So, the proposed method of hyper-parameters optimization aims to discover the hyper-parameters that maximize the classification accuracy of CNN. The first population is created randomly. The fitness function is set to be the classification accuracy of the CNN. If no change occurred in the individual hyper-parameters so, no need for CNN re-training. This process will be repeated until the iteration numbers are ended. The proposed method steps:

- Initialize parameters of the PSO (population size, iteration number, acceleration constants c1 and c2).
- Set the position and velocity of the particles corresponding to pre-defined CNN hyper- parameters.
- Train the CNN with the parameters' value of all particles.
- Measure the fitness value (classification accuracy) of all particles, after this select pbest and the swarm best particle position.
- Update the position X and velocity V of each particle.
- End this process and design the model of the CNN using the optimized parameters if the iteration numbers are finished. Else, repeat the process from step 3.

4. RESULTS AND DISCUSSION

We evaluated the proposed approach in this section using the MNIST dataset; sample image of the dataset is shown in Figure 4. The dataset has images of 28 × 28 grayscale pixel size of handwritten digits from 0 to 9. It has a 60,000 training and 10,000 testing samples. No further preprocessing is done on the data. The parameters that used in the experiments are shown in Table 1. These parameters are chosen by trials and errors. At the end of each epoch, a permutation is done to the training samples. To save the time of the training and avoid overfitting; if the loss of the training was less than the loss of validation, we terminate the training, else, continue till the end of the training iterations.

Figure 4. Sample of MINST dataset [26]

Table 1. Experiments parameters

Parameter	Value
PSO:	
Swarm size	15
Number of iterations	30
Acceleration factors	c1=2, c2 = 2
CNN initialization:	
Convolution layers number	2
Convolution kernel size	1-8
Number of kernels	1-128
Neurons numbers in the fully connected layer	1-300
CNN training:	
Number of epochs	100
Batch size	50
Learning rate	0.55

The global best parameters of PSO are convolutional layer's parameters={(6, 5), (36, 5)} and connected layer's parameters={192}. The proposed method is consistently enhancing the gbest particle training accuracy after each iteration. Figure 5 shows training accuracy for 10 iterations as a sample from 30 iterations.

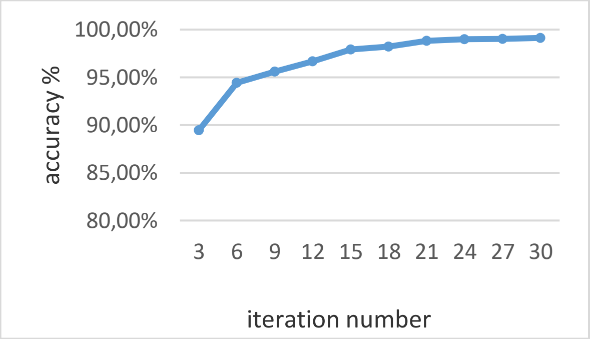

Figure 5. Training accuracy during the optimization process

The results of the experiments are represented in Table 2. We compared this result with a subset of representative methods which are psoCNN that automatically search for meaningful CNNs architectures with PSO algorithm [11], DPSO [14], LeNet-1 [27], LeNet- 4 [27], LeNet-5 [27], recurrent CNN that incorporate recurrent connections into each convolutional layer [28], PCANet-2, RANDNet-2 that employed the principal component, binary hashing and block-wise histograms analysis to the deep learning network [29], CAE-1 and CAE-2 that present an approach for training deterministic auto-encoders [30]. It showed that the optimized method is not just a competitive method but also it helps to improve other methods. We also tested our CNN hyperparameters optimization problem with another swarm intelligent algorithm which is the bee colony optimization algorithm.

Table 2. Comparison of MINST classification error of the state-of-the-art methods

Method	Test Error in %
psoCNN [17]	0.44
DPSO [20]	0.8
LeNet-1 [27]	1.7
LeNet-4 [27]	1.1
LeNet-5 [27]	0.95
Recurrent CNN [28]	0.31
PCANet-2 [29]	1.06
RANDNet-2 [29]	1.27
CAE-1 [30]	2.83
CAE-2 [30]	2.48
Proposed method	0.87
bee colony	1.02

Table 2 shows that our proposed method has a testing error of 0.87% using simple CNN with PSO and if we incorporate the skip connection or parallel layers it may yield less error. This error is competitive to the previously mentioned methods except for the psoCNN that has a testing error of 0.44%, considering that its architecture is more complex than our method, needs an encoding strategy during the computations, and the running time is longer, DPSO that has a testing error of 0.8, but its architecture was more complex as it designs a mix-variable encoding strategy and distributed framework to reduce the running time. Also, recurrent CNN has a testing error of 0.31% but it did not explore the best configuration and limited the search in constrained hyper-parameter space, meaning that no optimization is done in this method and it uses recurrent connections. LeNet-1, LeNet- 4, and LeNet-5 obtained a testing error of 1.7, 1.1, and 0.95 respectively, these networks have a small number of parameters with different types of last layer classifier. PCANet-2, RANDNet-2 that employed the principal component, binary hashing, and block-wise histograms analysis to the deep learning network obtained a testing error of 1.06 and 1.27 respectively. CAE-1 and CAE-2 that present an approach for training deterministic auto-encoders obtained a testing error of 2.83 and 2.48 respectively. Also, optimization with the bee colony algorithm gives an error of 1.02.

5. CONCLUSION AND FUTURE WORK

We proposed a method of CNN hyper-parameters optimization based on particle swarm optimization algorithm. The experiments on the MNIST dataset demonstrate an improvement to some of the state-of-the-art methods. The results represent that this method was able to find optimized parameters of the CNN model. With 15 particles and 30 iterations, we can find architectures that achieve a testing error of 0.87 which is competitive to other designs. We aim to extend the proposed method and add more hyper-parameters in the optimization process for our future work.

REFERENCES

[1] C. Dhaware and K. H. Wanjale, “Survey on image classification methods in image processing,” *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 4, no. 3, pp. 246–248, 2016.

[2] G. E. Dahl, D. Yu, L. Deng and A. Acero, “Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition,” *in IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30-42, Jan. 2012, doi: 10.1109/TASL.2011.2134090.

[3] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[4] A. G. Mahmoud, A. M. Hasan, and N. M. Hassan, “Convolutional neural networks framework for human hand gesture recognition,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no.4, pp. 2223-2230, August 202, doi: 10.11591/eei. v10i4.2926.

[5] M. A. Rasyidi and T. Bariyah, “Batik pattern recognition using convolutional neural network,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no.4, pp. 1430-1437, August 2020, doi: 10.11591/eei.v9i4.2385.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations*, Apr. 2015, pp. 1-14.

[7] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time-series," In Arbib MA, editor, The handbook of brain theory and neural networks, MIT Press, 1995.

[8] R. Yamashita, M. Nishio, R. K. G. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, pp. 611–629, Jun 2018, doi: 10.1007/s13244-018-0639-9.

[9] A. Sahu, S. K. Panigrahi and S. Pattnaik, "Fast convergence particle swarm optimization for functions optimization," *Procedia Technol.*, vol. 4, pp. 319–324, 2012, doi: 10.1016/j.protcy.2012.05.048.

[10] W. Hu and G. G. Yen, "Adaptive Multiobjective Particle Swarm Optimization Based on Parallel Cell Coordinate System," in *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 1-18, Feb. 2015, doi: 10.1109/TEVC.2013.2296151.

[11] M. Z. Alom *et al.*, "A Comprehensive Survey on Deep Learning Approaches," *arXiv preprint arXiv:1803.01164*, Mar. 2018.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, Jun. 2017, doi: 10.1145/3065386.

[13] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[14] Sehla Loussaief and Afef Abdelkrim, "Convolutional Neural Network Hyper-Parameters Optimization based on Genetic Algorithms," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 252-266, Jan. 2018, doi: 10.14569/IJACSA.2018.091031.

[15] D. O. Miguel, G. Sorrosal, C. Enrique, C. Martin and A. Alonso-Vicario, "Evolutionary algorithms for hyperparameter tuning on neural networks models," in *26th European Modeling and Simulation Symposium*, pp 402-409, Jan. 2014.

[16] I. Loshchilov and F. Hutter, "CMA-ES for hyperparameter optimization of deep neural networks," *International Conference on Learning Representations*, pp. 5549–5601, Apr. 2016.

[17] F. E. F. Junior and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Journal of Swarm and Evolutionary Computation*, vol. 49, pp 62-74, Sep. 2019, doi: 10.1016/j.swevo.2019.05.010.

[18] M. F. A. Foysal, N. Sultana, T. A. Rimi, and M. H. Rifat, "Convolutional Neural Network Hyper-Parameter Optimization Using Particle Swarm Optimization," *Emerging Technologies in Data Mining and Information Security*, pp. 363-373, 2021.

[19] T. Sinha, A. Haidar and B. Verma, "Particle Swarm Optimization Based Approach for Finding Optimal Values of Convolutional Neural Network Parameters," *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1-6, doi: 10.1109/CEC.2018.8477728.

[20] Yu Guo, Jian-Yu Li, and Zhi-Hui Zhan, "Efficient Hyperparameter Optimization for Convolution Neural Networks in Deep Learning: A Distributed Particle Swarm Optimization Approach," *Cybernetics and Systems*, vol. 52, no. 1, pp. 1-22, doi: 10.1080/01969722.2020.1827797.

[21] T. Yamasaki, T. Honma and K. Aizawa, "Efficient Optimization of Convolutional Neural Networks Using Particle Swarm Optimization," *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, 2017, pp. 70-73, doi: 10.1109/BigMM.2017.69.

[22] Y. Wang, H. Zhang, and G. Zhang Wang, "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks," *Swarm and Evolutionary Computation*, vol. 49, pp. 114-123, doi: 10.1016/j.swevo.2019.06.002.

[23] A. A. K. Taher and S. M. Kadhim, "Improvement of genetic algorithm using artificial bee colony," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 2125-2133, October 2020, doi: 10.11591/eei.v9i5.2233.

[24] Y. Bhor, N. S. Kakuste and S. Gaynar, "Optimization of Turning Parameters Using Taguchi's Method and Artificial Bee Colony Algorithm," *IOP Conference Series: Materials Science and Engineering, Volume 810, 2nd International Conference on Emerging trends in Manufacturing, Engines and Modelling (ICEMEM -2019)*, Mumbai, India, vol. 810, 23-24 December 2019, doi: 10.1088/1757-899X/810/1/012064.

[25] J. Kennedy and R. C. Eberhart, "Swarm Intelligence," *Academic Press*: San Diego, CA, 2001.

[26] Z. Shaukat, S. Ali, Q. A. Farooq, C. Xiao, S. Sahiba, and A. Ditta, "Cloud-based efficient scheme for handwritten digit recognition," *Multimedia Tools and Applications*, vol. 79, pp. 29537-29549, 2020, doi: 10.1007/s11042-020-09494-1.

[27] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[28] Ming Liang and Xiaolin Hu, "Recurrent convolutional neural network for object recognition," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3367-3375, doi: 10.1109/CVPR.2015.7298958.

[29] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng and Y. Ma, "PCANet: A Simple Deep Learning Baseline for Image Classification?," in *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017-5032, Dec. 2015, doi: 10.1109/TIP.2015.2475625.

[30] S. Rifai, P. Vincent, X. Muller, X. Glorot and Y. Bengio, "Contractive auto-encoders: explicit invariance during feature extraction," *ICML'11: Proceedings of the 28th International Conference on International Conference on Machine Learning*, Jun. 2011, pp. 833–840.

BIOGRAPHIES OF AUTHORS

Zainab Fouad received the bachelor’s degree in computer science from the Faculty of Computer and Information Science (FCIS), Ain Shams University (ASU), Egypt, in 2014, where she is currently pursuing the master’s degree. She is currently a Teaching Assistant with FCIS, ASU. Her research interests include machine learning areas.

Marco Alfonse is an Assistant Professor at the Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt. He got Ph.D. of Computer Science since August 2014, University of Ain Shams. His research interests: Semantic Web, Ontological Engineering, Medical Informatics, Smart Health, Opinion Mining, Internet of Things (IoT), Computer Vision, Machine Learning, Deep Learning, and Artificial Intelligence. He has published around 60 publications in refereed international journals and conferences.

Mohamed Roushdy Received his Ph.D. in 1993, M.Sc degree in 1984 and B.Sc. degree in 1979 from the Faculty of Science, Ain Shams University. His experimental doctoral research work was conducted at Bochum University in Germany during 1989-1991. He is a Professor of Computer Science since 2007 till now and Dean of the Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt since 2010 till 2018. He is a Dean of Faculty of Computers and Information Technology, Future University in Egypt from Sept. 2019 till now. His areas of research are Artificial Intelligence, Knowledge engineering, Machine Learning, Expert Systems, Intelligent Medical, Health Informatics, Image Processing, Pattern Recognition, Biometrics, and Medical Data Mining. He published more than 130 scientific papers in International Journals, International Conferences and Book Chapter. He got Ain Shams University Appreciation Award in Technological Sciences (Year 2018).

Abdel-Badeeh M. Salem was a Director of the Scientific Computing Center, Ain Shams University, Cairo, Egypt, from 1984 to 1990, where he was a professor of computer science with the Faculty of Science, from 1989 to 1996, a former Vice Dean of the Faculty of Computer and Information Sciences, from 1996 to 2007, and has been a Professor Emeritus of Computer Science, since 2007. His research interests include intelligent computing, expert systems, medical informatics, and intelligent e-learning technologies. He has published around 200 papers in refereed journals and conference proceedings in these areas. He has been involved in more than 200 conferences and workshops as an Int. Program Committee, Organizer, and Session Chair. He has authored and co-authored of 15 books in English and Arabic Languages. He also served as a member of many scientific societies. He is a member of the Editorial Board of many leading international conferences and journals. Frequency and Time Forum Award, the Carl Zeiss Research Award, the William F. Meggers Award and the Adolph Lomb Medal (OSA).