Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 6, 2026

AUTOMATTED CNN ARCHITECTURE BASED ON IMPROVED PSO EMPWERED WITH OBL FOR SKIN CANCER CLASSIFICATION

[1]Muhammad Asif Saleem, [2]Muhammad Umer Iqbal, [3]Muhammad Kashif Sidhu
[1]Faculty of Computer Science and Information Technology, Lahore Garrison University
[2]Faculty of Computer Science and Information Technology, Lahore Garrison University
[3]Department of Computer Sciences , Lahore Garrison University
[1]asif.saleem46@gmail.com, [2]umer.iqbal@lgu.edu.pk

DOI:https://doi.org/10.5281/zenodo.20680863

Keywords
CNN, Evolutionary Algorithms, Mutation Operator, Image classification

Article History
Received on 16 May 2026
Accepted on 12 June 2026
Published on 13 June 2026

Corresponding Author: *

Abstract
Early and accurate detection of skin cancer, particularly melanoma, is critical for effective treatment and improved patient outcomes. Convolutional Neural Networks (CNN) have shown promising performance in medical image analysis, however, designing an optimal CNN architecture is complex and time-consuming. This paper proposes an automated CNN architecture for skin cancer classification based on input data by improving Particle Swarm Optimization with mutation operator and Opposition-Based Learning (OBL). The HAM10000 dataset, which comprises 10,015 dermatoscopic images, was used to evaluate the proposed model. The hybrid MPSO-OBL optimized CNN was compared with several baseline models, including PSO-CNN, GA-CNN, LeNet, and Alex-Net. The results indicate that the proposed model outperforms the baseline models across all performance metrics, achieving an accuracy of 89.5%, The proposed technique also demonstrated efficient training time, and a relatively low number of parameters compared to Alex-Net, making it suitable for deployment in resource constrained environments. The study concludes that the integration of MPSO and OBL offers a robust and computationally efficient method for automating the CNN architecture design for skin cancer classification, paving the way for future advancements in medical image analysis.

Introduction

The rising incidence of skin cancer has necessitated the development of more efficient and accurate diagnostic tools. Among the various types of skin cancers, melanoma is particularly aggressive and accounts for a significant proportion of skin cancer-related deaths[1]. Early and accurate detection of skin lesions is crucial for effective treatment and improved survival[2]. Conventional diagnostic methods, primarily reliant on visual inspection by dermatologists, are often subjective and prone to errors[3]. As a result, there has been growing interest in automated diagnostic systems that leverage advancements in machine learning and image processing[4].

CNN have emerged as powerful tools for medical image analysis, demonstrating high efficacy in various classification tasks[5]. However, designing an optimal CNN architecture for specific applications such as skin cancer classification is a complex and time-consuming task that requires expert knowledge and extensive expertise in domain knowledge as well as desigining CNN architecture. Automated methods for CNN architecture design have thus gained atraction as a means of alleviating this challenge[6][7].

Random Search (RS) and Grid Search (GS) are widely recognized as non-adaptive methods for hyperparameter tuning in small Convolutional Neural Networks (CNNs) [8],[9],[10]. The term "non-adaptive" reflects the fact that these methods do not modify the search process based on intermediate results. Grid Search systematically trains the CNN using all possible combinations of hyperparameter values within a predefined range. However, as noted in[11], GS is susceptible to the curse of dimensionality, wherein the number of combinations increases exponentially with the number of hyperparameters. This often results in inefficient exploration, particularly when excessive trials are devoted to less critical dimensions, leading to inadequate coverage of essential hyperparameter spaces. In [12], GS was employed for optimizing neural network hyperparameters, after which Bergstra and Bengio [13] conducted comparative experiments using RS. Their results indicate that RS generally yields superior models with reduced computational costs compared to GS. The efficacy of both RS and GS can be enhanced through expert-driven manual adjustments, introducing a degree of adaptiveness to these methods.

Adaptive approaches, such as Bayesian Optimization [14][15][16] and metaheuristic-based methods, refine the search space iteratively using information from previous stages. Despite the straightforward implementation of Bayesian Optimization techniques, contemporary research predominantly favors metaheuristic approaches, particularly evolutionary algorithms, due to their robust exploration capabilities within the search space. The application of evolutionary computation for neural network optimization, termed neuroevolution, has been an active area of research since the late 1980s, with significant contributions including NEAT [17]. Although neuroevolution is an established domain, recent advancements in computational power, particularly in GPU availability, have facilitated the efficient optimization of more complex neural networks. For instance, Sun et al. introduced EvoCNN, an evolutionary approach designed to automatically generate CNN architectures and initialize weights. The resulting models demonstrated competitive performance relative to more complex architectures[18]. Similarly, Ma et al. [19] presented a study utilizing evolutionary algorithms, which was tested on benchmark datasets and produced results competitive with state-of-the-art architectures. Additionally, Baldominos et al. [20] employed genetic algorithms (GA) and grammatical evolution to automate CNN architecture generation, further illustrating the potential of evolutionary techniques in this domain.

Particle Swarm Optimization (PSO) has been widely used for various optimization problems, owing to its simplicity and effectiveness. In the context of deep learning, PSO has been applied to optimize neural network parameters and architectures. Eberhart and Shi (2001) introduced PSO as an efficient optimization algorithm, and subsequent research has explored its application in optimizing CNN hyperparameters[21][22][23]. Despite its success, traditional PSO often faces challenges, such as premature convergence and stagnation[24].

This paper proposes an automated technique for designing CNN architectures for skin cancer classification, utilizing an optimized Particle Swarm Optimization (PSO) algorithm combined with Opposition-Based Learning. The proposed method focuses on both the creation of CNN architectures and the optimization of their hyperparameters, with the objective of improving classification performance while minimizing computational costs. This study offers a comprehensive analysis of the efficacy of the proposed hybrid optimization technique and its influence on the accuracy of skin cancer diagnosis.

Convolution Neural Network

This section provides a concise overview of key concepts related to CNN to facilitate a better understanding of this study. CNNs are a class of deep neural networks characterized by the application of convolutional operations in at least one layer, with the objective of extracting meaningful features from the data[25]. A typical CNN architecture comprises several layers, including the Input Layer, Convolutional Layer, Pooling Layer, Fully Connected Layer, and Output Layer[26]. The input layer receives the images, which are then processed through the convolutional layers where feature maps are generated using filters. The architecture of a convolutional layer is defined by parameters such as the number of feature maps, filter sizes, and additional settings like stride and padding[27]. The number of convolutional layers can vary depending on the specific CNN model[27]. For example, the architecture depicted in Figure 1 (elements not to scale) consists of two convolutional layers, each employing a distinct number of filters of varying sizes. In this architecture, images with dimensions of 96 × 96 × 3 are fed into the CNN, where the first two dimensions represent the width and height of the image, and the third dimension corresponds to the number of channels (3 for a color image). The first convolutional layer includes 32 filters, each measuring 46 × 46. Following the convolutional operation, the linear structure of the data is transformed into a non-linear one through the application of the ReLU activation function. As illustrated, the filter sizes are reduced during the pooling operation. Subsequently, these features are flattened into a one-dimensional vector, which is then passed through the fully connected (FC) layer to yield the classification result. The model error is calculated based on the output, and the network weights are adjusted accordingly to minimize this error.

The pooling layer is responsible for reducing the dimensions of the feature maps based on selected hyperparameters such as filter size, stride, and pooling method. The stride parameter, common to both convolutional and pooling layers, determines the step size as the filter moves across the input. During pooling, the method produces either the maximum or average value (depending on the chosen pooling strategy) within the filter's receptive field. As an alternative to conventional pooling, Springenberg et al. [18] proposed the "Strive" method, which utilizes a convolutional layer with a filter size of either 3 × 3 or 2 × 2 and a stride of 2. Notably, the filters in this layer do not possess weights to learn but are employed solely for dimensionality reduction, with the summary information subsequently passed to other layers.

Proposed Technique

This section explains the methodology employed to develop the automated CNN architecture for skin cancer classification using a an optimized PSO wiith mutation operator and Opposition based Learning. The methodology comprised input section ,Training sample selection, OMUT-PSO and identified CNN architecture. Figure 1 systematically illustrates the proposed technique for the automated generation of CNN architectures, utilizing Particle Swarm Optimization (PSO) empowered with a uniform Mutation Operator and Opposition-Based Learning (OBL). Algorithm 1 briefly ecplains the complete process of creating CNN architecture. The process begins with the input dataset, which consists of various types of skin lesions, including Actinic keratoses, Basal cell carcinoma, Benign keratosis-like lesions, Dermatofibroma, Melanoma, Melanocytic nevi, and Vascular lesions. These images serve as the foundational data for training the CNN.

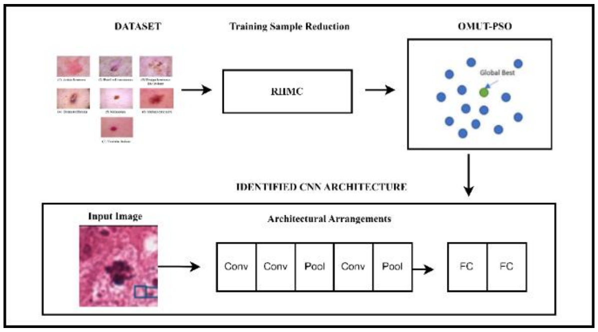

Figure 1:Framework proposed technique

Before proceeding to the architecture creation phase, the input dataset undergoes a training sample reduction process using a technique referred to as RHMC. It reduces the training sample size while retaining essential information, thereby alleviating the computational burden and enhancing the efficiency of the subsequent optimization process. Following this, the CNN architecture is automatically generated using an improved PSO algorithm, known as OMUT-PSO. This approach integrates a Mutation Operator with the standard PSO to refine the velocity computation of the particles. Figure 2 briefly elaborate anc compare the the improve velocity computation process.

The mutation step introduces variability in particle velocity, thereby improving the exploration of the search space. Algorithm 1 briefly explain the process of architecture creation process. Additionally, Opposition-Based Learning (OBL) is incorporated into this phase to expedite the selection of promising candidate architectures. OBL enhances the search process by considering both the current solutions and their opposites, facilitating faster convergence toward the optimal solution. The "Global Best" particle, highlighted in the diagram, represents the optimal CNN architecture identified during the optimization process.

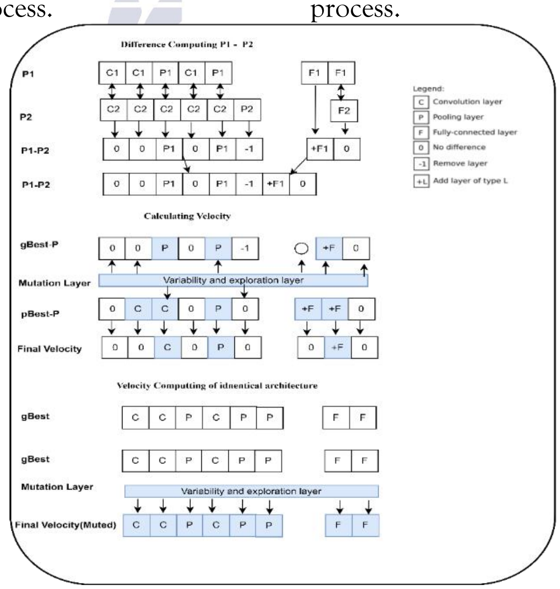

Figure 2:Improved Velocity Computation Process

Algorithm 1: Proposed OMUT-PSOCNN

1. **Mapping particles as CNN architecture**

Implementing a direct encoding strategy

2. **Initialization**

- Initialize swarm
- Initialize particles velocity

a. Velocity will determine how much architecture will change in each iteration.

3. **Fitness Evaluation**
- Calculate the fitness of the particle.

a. Training and validating a portion of training data.

4. **Update Velocity and Position**
- Calculate velocity by optimizing with the mutant operator.

a. Apply mutant operator with low probability for random changes in hyperparameter of architectural arrangements.
- Compute the fitness of muted architecture.
- Generate opposite architecture and evaluate its fitness.
- Compare both and update architecture based on fitness.

5. **Termination Criteria**
- If (No. of iteration == defined iterations || desired fitness)

Save architecture and start final training.

6. **Else**
- Repeat till a predefined number of iterations or fitness value.

Results

The proposed technique for automating the creation of CNN architectures was rigorously evaluated using publicly available datasets, specifically HAM-1000. The proposed technique is executed ten times. The experiments were conducted on a system equipped with an Intel Core i7 processor, an Nvidia GeForce GTX 1080 Ti GPU with 11 GB of memory, and 32 GB of RAM, running Windows 10. Python 3.7 was used for the development and testing of the algorithms, with TensorFlow employed for its robust numerical computation capabilities. TensorFlow's support for both CPU and GPU processing enabled efficient and rapid development.

Remarkably, the proposed technique demonstrated outstanding performance without the use of data augmentation methods. Figure 3 depicts the training curves for the proposed technique on selected dataset, showing a consistent increase in training accuracy throughout the training process. Figure 5 present a comprehensive comparison of accuracy, precision, and recall for architectures generated by OMUT-PSOCNN.

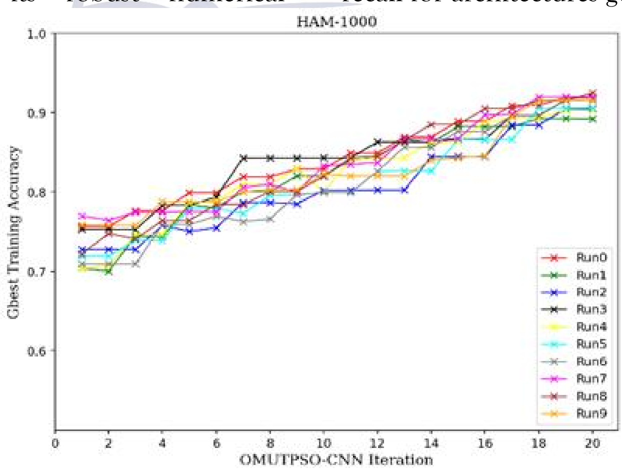

Figure 3:Training Accuracy of Proposed technique for ten runs

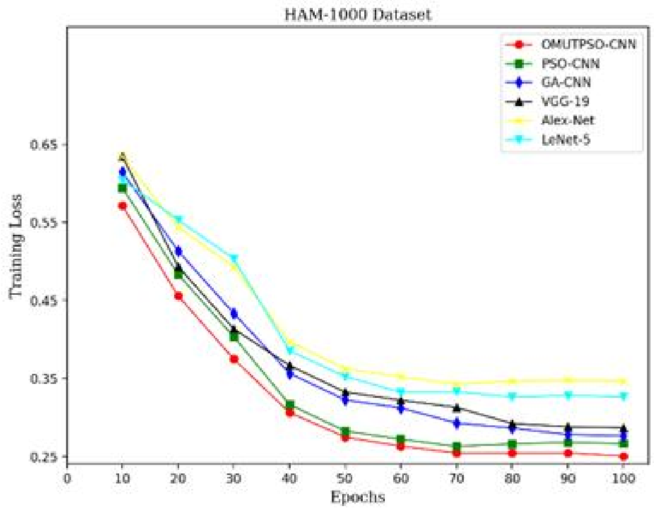

Figure 4:Convergence of Created CNN architecture on Full Training

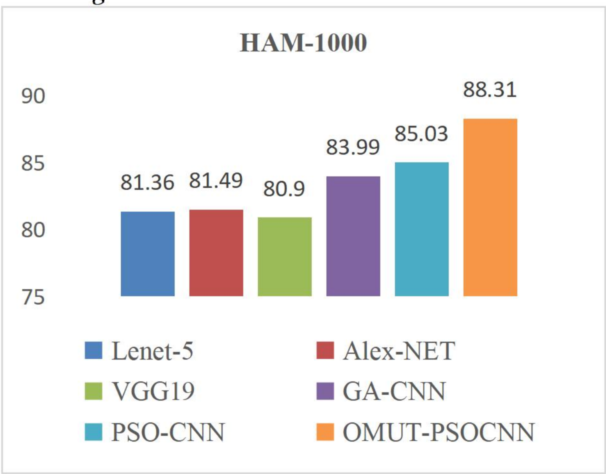

Figure 5:Comparative Analysis of Automatically Created CNN with Competitors

Further analysis involved training and evaluating the created architectures on the full datasets and comparing their performance against both handcrafted and existing automated architectures. Figure 4 illustrates the convergnec rate of architecture optimized by the proposed technique on selected the datasets, highlighting its superior performance relative to other approaches.

Conclusion

In conclusion, despite the significant advancements made by CNN in solving complex problems, the challenge of designing an optimal architecture for specific tasks persists. This paper addresses this challenge by introducing an automated approach for optimizing CNN architectural parameters using Mutant Particle Swarm Optimization (MPSO) and Opposition-Based Learning. The proposed OMUT-PSOCNN technique was evaluated on benchmark dataset, including HAM-1000, and compared with established architectures such as AlexNet, VGG-19, PSO-CNN, and GA-CNN. The experimental results demonstrate that OMUT-PSOCNN effectively identified optimal architectural parameter combinations with a computationally efficient approach, utilizing only 30 particles and 20 iterations. The architectures generated by OMUT-PSOCNN not only achieved superior accuracy but also exhibited reduced processing times during training and testing on the benchmark datasets. These findings underscore the efficiency and effectiveness of OMUT-PSOCNN in automating the creation of CNN architectures that are specifically tailored to the characteristics of the datasets in question.

References

[1] H. Younis, M. H. Bhatti, and M. Azeem, “Classification of skin cancer dermoscopy

images using transfer learning," *15th Int. Conf. Emerg. Technol. ICET 2019*, Dec. 2019, doi: 10.1109/ICET48972.2019.8994508.

[2] V. Anand, S. Gupta, D. Koundal, S. Mahajan, A. K. Pandit, and A. Zaguia, "Deep learning based automated diagnosis of skin diseases using dermoscopy," *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 3145–3160, 2022, doi: 10.32604/cmc.2022.022788.

[3] M. Fraiwan and E. Faouri, "On the Automatic Detection and Classification of Skin Cancer Using Deep Transfer Learning," *Sensors 2022, Vol. 22, Page 4963*, vol. 22, no. 13, p. 4963, Jun. 2022, doi: 10.3390/S22134963.

[4] M. Fraiwan and E. Faouri, "On the Automatic Detection and Classification of Skin Cancer Using Deep Transfer Learning," *Sensors*, vol. 22, no. 13, 2022, doi: 10.3390/s22134963.

[5] D. Bhatt *et al.*, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electron. 2021, Vol. 10, Page 2470*, vol. 10, no. 20, p. 2470, Oct. 2021, doi: 10.3390/ELECTRONICS10202470.

[6] A. Al Bataineh, D. Kaur, M. Al-khassaweneh, and E. Al-sharoa, "Automated CNN Architectural Design: A Simple and Efficient Methodology for Computer Vision Tasks," *Mathematics*, vol. 11, no. 5, pp. 1–17, 2023, doi: 10.3390/math11051141.

[7] B. Fielding, T. Lawrence, and L. Zhang, "Evolving and Ensembling Deep CNN Architectures for Image Classification," 2019, Accessed: Aug. 28, 2022. [Online]. Available: http://nrl.northumbria.ac.uk/id/eprint/40456/

[8] M. G. Ragab, S. J. Abdulkadir, and N. Aziz, "Random Search One Dimensional CNN for Human Activity Recognition," *2020 Int. Conf. Comput. Intell. ICCI 2020*, vol. 1, no. 1, pp. 86–91, Oct. 2020, doi: 10.1109/ICCI51257.2020.9247810.

[9] A. Nurhopipah and N. A. Larasati, "CNN hyperparameter optimization using random grid coarse-to-fine search for face classification," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 1, p. 1, 2021.

[10] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[11] F. Hutter, H. Hoos, and K. Leyton-Brown, "Sequential Model-based Optimization for General Algorithm Configuration Motivation," 2011.

[12] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation".

[13] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012, Accessed: Aug. 12, 2024. [Online]. Available: http://scikit-learn.sourceforge.net.

[14] M. I. Sameen, B. Pradhan, and S. Lee, "Application of convolutional neural networks featuring Bayesian optimization for landslide susceptibility assessment," *Catena*, vol. 186, no. August 2018, p. 104249, 2020, doi: 10.1016/j.catena.2019.104249.

[15] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.

[16] G. Atteia, A. A. Alhussan, and N. A. Samee, "BO-ALLCNN: Bayesian-Based Optimized CNN for Acute Lymphoblastic Leukemia Detection in Microscopic Blood Smear Images," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155520.

[17] E. Galván and P. Mooney, "Neuroevolution in Deep Neural Networks: Current Trends and Future Challenges," *IEEE Trans. Artif. Intell.*, vol. 2, no. 6, pp. 476–493, 2021, doi: 10.1109/TAI.2021.3067574.

[18] J. Dong, L. Zhang, B. Hou, and L. Feng, "A Memetic Algorithm for Evolving Deep Convolutional Neural Network in Image Classification," in *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 2020, pp. 1–19. doi: 10.1109/SSCI47803.2020.9308162.

[19] B. Ma, X. Li, Y. Xia, and Y. Zhang, "Autonomous Deep Learning: A Genetic DCNN Designer for Image Classification," *Neurocomputing*, vol. 379, pp. 152–161, Jul. 2018, doi: 10.1016/j.neucom.2019.10.007.

[20] A. Baldominos, Y. Saez, and P. Isasi, "Evolutionary convolutional neural networks: An application to handwriting recognition," *Neurocomputing*, vol. 283, pp. 38–52, Mar. 2018, doi: 10.1016/J.NEUCOM.2017.12.049.

[21] M. Carvalho and T. B. Ludermir, "Particle Swarm Optimization of Neural Network Architectures andWeights," pp. 336–339, Jul. 2008, doi: 10.1109/HIS.2007.45.

[22] B. Fielding and L. Zhang, "Evolving Image Classification Architectures with Enhanced Particle Swarm Optimisation," *IEEE Access*, vol. 6, pp. 68560–68575, 2018, doi: 10.1109/ACCESS.2018.2880416.

[23] B. Wang, B. Xue, and M. Zhang, "Particle Swarm Optimization for Evolving Deep Convolutional Neural Networks for Image Classification: Single- and Multi-Objective Approaches," *Nat. Comput. Ser.*, pp. 155–184, 2020, doi: 10.1007/978-981-15-3685-4_6/COVER.

[24] F. Marini, B. W.-C. and I. L. Systems, and undefined 2015, "Particle swarm optimization (PSO). A tutorial," *Elsevier*, no. 149, pp. 153–165, 2015, Accessed: Aug. 12, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743915002117

[25] A. Sharma and G. Phonsa, "Image Classification Using CNN," *SSRN Electron. J.*, vol. 2, no. 1, pp. 1–5, 2021, doi: 10.2139/ssrn.3833453.

[26] J. Brown, Z. Gharineiat, and N. Raj, "CNN Based Image Classification of Malicious UAVs," *Appl. Sci.*, vol. 13, no. 1, 2023, doi: 10.3390/app13010240.

[27] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.