International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 1, January 2026)

Enhancing CNN Performance on MNIST through Metaheuristic Hyperparameter Optimization

Raniah Zaheer[1], Dr Humera Shaziya[2]

[1]*Department of Computer Science, St. Ann's College for Women, Mehdipatnam, Telangana, India*
[2]*Department of Informatics, Nizam College, Osmania University, Telangana, India*

***Abstract*— The performance of Deep Learning models is highly sensitive to their hyperparameters. Traditional methods like Grid Search and Random Search are often computationally expensive and inefficient. This paper presents an empirical comparison of four population-based and trajectory-based metaheuristic algorithms-Differential Evolution (DE), Particle Swarm Optimization (PSO), Simulated Annealing (SA), and a custom Archerfish Optimizer (AHO)-for the task of hyperparameter tuning a Convolutional Neural Network (CNN) on the MNIST dataset. The hyperparameter search space includes the learning rate, batch size, number of convolutional filters, and dropout rate. Our results, measured by final validation accuracy and computational time, indicate that while PSO and DE achieve the highest accuracy (~98.96%), SA offers a significant trade-off, converging to a good solution in approximately 60% of the time required by the population-based methods. The study demonstrates the efficacy of metaheuristics as efficient and effective tools for automated hyperparameter optimization in deep learning.**

***Keywords*— Metaheuristic Optimization, Hyperparameter Tuning, Convolutional Neural Networks, MNIST, Differential Evolution, Particle Swarm Optimization, Simulated Annealing.**

I. INTRODUCTION

The success of Convolutional Neural Networks (CNNs) in image recognition tasks is undeniable. However, achieving state-of-the-art performance requires careful selection of hyperparameters, which govern the training process and model architecture. Manual tuning is labor-intensive and relies heavily on expert intuition. Automated Hyperparameter Optimization (HPO) is thus a critical area of research in machine learning.

While exhaustive methods like Grid Search are guaranteed to find the optimal solution within a discrete search space, they are computationally prohibitive for high-dimensional problems. Random Search [1] offers a more efficient alternative but may still waste resources by evaluating poor configurations. In recent years, metaheuristic algorithms, inspired by natural phenomena, have emerged as powerful tools for global optimization in complex, non-convex search spaces.

This work investigates the application of four metaheuristic algorithms for tuning the hyperparameters of a CNN designed for the MNIST digit classification task:

i. Differential Evolution (DE)
ii. Particle Swarm Optimization (PSO)
iii. Simulated Annealing (SA)
iv. A custom Archerfish Optimizer (AHO)

We compare their performance in terms of final model accuracy and computational efficiency, providing insights into their suitability for deep learning HPO.

II. LITERATURE REVIEW

The performance of machine learning models, particularly deep neural networks, is critically dependent on their hyperparameters. Traditional manual tuning is time-consuming and requires expert knowledge, leading to the development of automated Hyperparameter Optimization (HPO) methods. Early approaches included Grid Search, which exhaustively searches a predefined hyperparameter space, but this method becomes computationally prohibitive as dimensionality increases. A significant advancement came from Bergstra & Bengio [1], who demonstrated that Random Search is often more efficient than Grid Search in high-dimensional spaces. Storn & Price [2] introduced Differential Evolution (DE), which creates new candidate solutions by combining existing ones using difference vectors. The application of these metaheuristics to deep learning has gained significant attention. Population-based algorithms offered more robust search capabilities. Kennedy & Eberhart [3] developed Particle Swarm Optimization (PSO), inspired by social behavior patterns like bird flocking.

Metaheuristic algorithms, inspired by natural phenomena, emerged as powerful alternatives for global optimization problems. These algorithms balance exploration (searching new areas) and exploitation (refining known good areas) more effectively than simple random sampling. Kirkpatrick, Gelatt, & Vecchi [4] introduced Simulated Annealing (SA), a trajectory-based algorithm inspired by the annealing process in metallurgy.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 1, January 2026)

Zaheer & Shaziya [5] provided a comprehensive study comparing various optimization algorithms in deep learning. Recent comprehensive reviews have solidified the position of metaheuristics in deep learning HPO. Ibrahim et al. [6] conducted an extensive survey specifically focusing on optimizing Convolutional Neural Networks (CNNs) through metaheuristic algorithms. Baskaran, Pratap, & Bansal [7] explored nature-inspired metaheuristic algorithms for CNN hyperparameter tuning in image classification tasks.

PSO has received particular attention due to its effectiveness and simplicity. Munsarif, Sam'an, & Fahrezi [8] proposed a modified PSO specifically designed for CNN hyperparameter optimization, incorporating adaptive parameters and specialized mutation operators to improve convergence speed and solution quality in image classification problems.Narayanan & Ganesh [9] provided a broader perspective on metaheuristics for HPO across machine learning, discussing the adaptation of these algorithms to handle the specific challenges of ML hyperparameter spaces, including mixed data types (continuous, discrete, categorical) and expensive function evaluations. The application of metaheuristics to specific domains has yielded impressive results.

In medical imaging, a domain where model accuracy is critical, Aguerchi et al. [10] demonstrated the successful application of PSO for optimizing CNN hyperparameters in mammography breast cancer classification. Their work showed that metaheuristic-optimized CNNs achieved superior performance compared to manually-tuned models, highlighting the practical significance of these methods in real-world applications where model performance directly impacts decision-making.

A. Research Gaps

The literature demonstrates a clear evolution from simple methods like Grid and Random Search to sophisticated metaheuristic approaches for HPO. Population-based algorithms like PSO and DE have proven particularly effective for tuning CNN architectures, consistently outperforming both traditional methods and single-solution metaheuristics like SA in terms of final solution quality. However, several research gaps remain. The computational cost of metaheuristics remains a concern, especially when combined with the already expensive training of deep neural networks. There is ongoing research into developing more efficient variants and hybrid approaches that combine the strengths of multiple algorithms.

Additionally, the adaptation of metaheuristics to handle increasingly complex neural architectures and the integration with other HPO methods like Bayesian optimization represent promising future directions. The collective evidence suggests that metaheuristic algorithms have established themselves as essential tools in the deep learning practitioner's toolkit, particularly for complex computer vision tasks where optimal hyperparameter configuration can significantly impact model performance.

III. METHODOLOGY

A. Problem Formulation

The HPO task is framed as a minimization problem. Let a candidate solution (an individual in the population or a state) be a vector x representing a set of hyperparameters: x = [learning_rate, batch_size, num_filters, dropout_rate]

The objective function f(x) is defined as 1 - validation_accuracy, where validation_accuracy is the performance of a CNN trained with hyperparameters x on the MNIST test set. The goal of the metaheuristics is to find x* that minimizes f(x).

The search space bounds for the hyperparameters are defined as follows:

- Learning Rate: [1e-4, 1e-2] (Log-scale)
- Batch Size: [16, 128] (Integer)
- Number of Filters: [8, 64] (Integer)
- Dropout Rate: [0.1, 0.6] (Continuous)

B. CNN Architecture and Training

A simple yet effective CNN architecture is employed:

i. *Convolutional Block 1:* A 2D convolutional layer with num_filters and a 3x3 kernel, followed by ReLU activation and a 2x2 max-pooling layer.

ii. *Convolutional Block 2:* A 2D convolutional layer with num_filters * 2 filters and a 3x3 kernel, followed by ReLU activation and a 2x2 max-pooling layer.

iii. *Classifier:* A fully connected layer mapping the flattened features to 128 units (with ReLU and Dropout), followed by a final output layer of 10 units.

The model is trained for a short cycle of 2 epochs using the Adam optimizer and Cross-Entropy loss. This limited training allows for a rapid evaluation of hyperparameter quality, which is essential for iterative metaheuristic search.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 1, January 2026)

C. Metaheuristic Algorithms

All algorithms were configured with a small population size (6 for DE, PSO, AHO) and a low number of iterations (4-6) to maintain a fixed, low computational budget, simulating a scenario where resources are limited.

i. *Differential Evolution (DE):* A population-based algorithm that creates new candidates by combining existing ones according to a difference vector strategy. Key parameters: F=0.8 (mutation scale) and CR=0.9 (crossover rate).

ii. *Particle Swarm Optimization (PSO):* Inspired by social behavior, where particles (candidate solutions) move through the search space influenced by their own best-known position and the swarm's best-known position. Key parameters: w=0.7 (inertia), c1=1.5, c2=1.5 (acceleration coefficients).

iii. *Simulated Annealing (SA):* A trajectory-based algorithm that probabilistically accepts worse solutions to escape local minima, with an "annealing schedule" that reduces this probability over time. Key parameters: T0=1.0 (initial temperature), alpha=0.8 (cooling rate).

iv. *Archerfish Optimizer (AHO):* A custom algorithm inspired by the hunting behavior of archerfish. Each individual "shoots" a water jet (a new candidate solution) towards a randomly selected target in the population. If the new solution is better, it replaces the current one.

D. Experimental Setup

- *Dataset:* MNIST (70,000 28x28 grayscale images of digits).
- *Environment:* Python with PyTorch, running on an NVIDIA GPU (CUDA) in a Kaggle environment.
- *Evaluation:* Each algorithm was run once with its predefined budget. The best-found hyperparameters were used to train a final model, and its accuracy on the separate test set was recorded. The wall-clock time for the entire optimization process was also measured.

IV. Results And Discussion

The following table summarizes the performance of the four metaheuristic algorithms:

Optimizer	Best Validation Accuracy	Best Hyperparameters [lr, batch, filters, drop]	Time (min)
PSO	**98.96%**	[4.64e-3, 122, 18, 0.6]	8.20
DE	98.92%	[1.64e-3, 104, 59, 0.35]	8.82
AHO	98.83%	[3.90e-3, 105, 38, 0.38]	8.53
SA	95.07%	[5.53e-3, 96, 42, 0.37]	**5.33**

Discussions

i. *Accuracy Performance:* PSO and DE, both sophisticated population-based algorithms, achieved the highest validation accuracy, closely followed by AHO. This suggests that their mechanisms for exploring the search space (social learning in PSO and differential mutation in DE) are highly effective for this HPO problem.

ii. *Computational Efficiency:* SA was the fastest algorithm, completing its search in just 5.33 minutes—approximately 65% of the time taken by PSO. This is expected as SA is a trajectory-based method that maintains only a single candidate solution, whereas population-based methods evaluate multiple candidates per iteration.

iii. *Performance-Speed Trade-off:* SA converged to a significantly lower accuracy (~95%) than the other methods. This indicates that with a very limited budget, it may converge prematurely to a local optimum. However, its speed makes it an attractive option for a very rough, initial hyperparameter sweep.

iv. *AHO Performance:* The custom AHO performed respectably, demonstrating that even simple bio-inspired mechanisms can be effective for HPO. Its performance was on par with DE and PSO, though slightly lower, suggesting its "shooting" mechanism provides a good balance between exploration and exploitation.

The bar chart below visualizes the accuracy comparison, clearly showing the performance gap between SA and the other three algorithms.

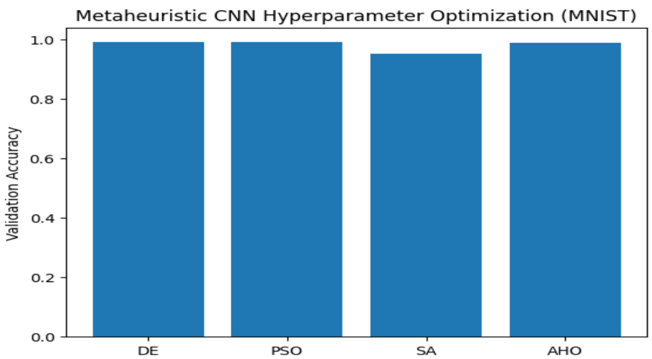

Figure 1: Comparison of final validation accuracy achieved by each metaheuristic optimizer.

V. Conclusion And Future Work

This study demonstrated the successful application of metaheuristic algorithms for hyperparameter tuning of a CNN on the MNIST dataset. Under a constrained computational budget, population-based algorithms like PSO and DE reliably found hyperparameter configurations yielding high accuracy (~98.9%), while Simulated Annealing provided a much faster, though less accurate, solution.

Future work will focus on:

i. *Scalability:* Testing these algorithms on more complex datasets (e.g., CIFAR-10, ImageNet) and larger CNN architectures.

ii. *Budget Analysis:* Conducting a more thorough analysis of performance versus computational budget (number of iterations and population size).

iii. *Advanced Metaheuristics:* Incorporating more recent and advanced metaheuristics like Gray Wolf Optimizer (GWO) or Harris Hawks Optimization (HHO).

iv. *Benchmarking:* Comparing these methods against Bayesian Optimization, a current state-of-the-art method for HPO.

In conclusion, metaheuristics present a powerful, flexible, and often underutilized approach to the hyperparameter optimization problem, capable of finding high-performing configurations efficiently.

REFERENCES

[1] Bergstra, James, and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization." Journal of Machine Learning Research, 2012.

[2] Storn, Rainer, and Kenneth Price. "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." Journal of Global Optimization, 1997.

[3] Kennedy, James, and Russell Eberhart. "Particle Swarm Optimization." Proceedings of ICNN'95 – International Conference on Neural Networks, 1995.

[4] Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. "Optimization by Simulated Annealing." Science, 1983.

[5] Zaheer, R., and H. Shaziya. "A Study of the Optimization Algorithms in Deep Learning." 2019 Third International Conference on Inventive Systems and Control (ICISC), IEEE, 2019, pp. 536–539.

[6] Ibrahim, Mohammad Q., Nermin K. Hussein, David Guinovart, and Mohammed Qaraad. "Optimizing Convolutional Neural Networks: A Comprehensive Review of Hyperparameter Tuning through Metaheuristic Algorithms." Archives of Computational Methods in Engineering, 2025, pp. 1–38.

[7] Baskaran, N. K., B. Pratap, and S. Bansal. "Hyperparameter Tuning of Convolutional Neural Networks Using Nature-Inspired Metaheuristic Algorithms for Image Classification." Nature-Inspired Metaheuristic Algorithms, CRC Press, 2025, pp. 119–155.

[8] Munsarif, M., M. Sam'an, and A. Fahrezi. "Convolution Neural Network Hyperparameter Optimization Using Modified Particle Swarm Optimization." Bulletin of Electrical Engineering and Informatics, vol. 13, no. 2, 2024, pp. 1268–1275.

[9] Narayanan, R., and N. Ganesh. "A Comprehensive Review of Metaheuristics for Hyperparameter Optimization in Machine Learning." Metaheuristics for Machine Learning: Algorithms and Applications, 2024, pp. 37–72.