ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov358

Efficient Training of ResNet-50 on Large Scale Datasets Using Enhanced Particle Swarm Optimization

A. W. C. K. Atugoda[1]; S. D. Fernando[2]

[1,2] Department of Computational Mathematics
University of Moratuwa
Colombo, SriLanka

Publication Date: 2025/11/17

Abstract: Deep Neural Networks (DNNs) such as ResNet-50 have achieved state-of-the-art results in large-scale image classification. However, their training performance depends strongly on optimization efficiency. Conventional methods like Stochastic Gradient Descent (SGD) often struggle with slow convergence, learning-rate sensitivity, and local minima entrapment. To overcome these challenges, this study introduces an Enhanced Particle Swarm Optimization (PSO) technique that adaptively adjusts particle dynamics to maintain a better exploration-exploitation balance during training. The method was evaluated on the CIFAR-10 dataset and compared against Standard PSO and SGD under identical experimental conditions. The results demonstrate that Enhanced PSO achieves superior validation accuracy (up to 95.7%), faster convergence, and a more stable weight distribution centered near zero. These characteristics reflect a well-regularized learning process with improved generalization. Overall, the Enhanced PSO framework provides a robust and scalable optimization approach for deep neural networks, offering a viable alternative to conventional gradient-based training algorithms.

Keywords: *Particle Swarm Optimization, Deep Neural Networks, ResNet-50, Convergence Stability, Weight Values Optimization.*

How to Cite: A. W. C. K. Atugoda; S. D. Fernando (2025) Efficient Training of ResNet-50 on Large Scale Datasets Using Enhanced Particle Swarm Optimization. *International Journal of Innovative Science and Research Technology*, 10(11), 555-559. https://doi.org/10.38124/ijisrt/25nov358

I. INTRODUCTION

Deep Neural Networks (DNNs) have been extensively applied across diverse fields such as engineering, science, biomedical research, and robotics, owing to their remarkable ability to address complex and nonlinear problem domains [1]. Among the available architectures, models such as ResNet-50 have become increasingly popular for large-scale data classification tasks, offering superior representational capacity compared to earlier models like LeNet-5. The performance of neural networks, however, critically depends on effective optimization, as achieving high accuracy in complex problem-solving scenarios requires efficient training of network parameters [2]. Traditionally, researchers have relied on conventional optimization techniques such as gradient descent and its variants for tuning the weight values of neural networks. While these approaches have achieved considerable success, they often encounter significant limitations, including susceptibility to local minima, slow convergence rates, and high computational overhead [3].

To address these challenges, recent research has increasingly focused on swarm intelligence-based optimization methods. Such approaches are recognized for their robustness and global search capabilities, making them particularly well-suited for overcoming the limitations of conventional methods and effectively handling large-scale, complex optimization problems [4].

Swarm Intelligence (SI) refers to the collective behavior that emerges from the interaction of homogeneous individuals with one another and with their environment in order to discover feasible solutions to complex problems [5]. Over the years, numerous SI-based algorithms have been developed, including Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), and Bacterial Foraging Optimization (BFO). Among these, PSO has emerged as one of the most widely adopted approaches, primarily due to its simplicity, the relatively small number of parameters that require tuning, and the fact that it does not rely on derivative information or complex mathematical simplifications [6].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov358

A. Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization technique inspired by the social behavior of bird flocking and fish schooling [7]. In PSO, a population of candidate solutions, called particles, explores the search space to optimize an objective function. Each particle has two main attributes: a position vector, representing a potential solution, and a velocity vector, which determines the direction and magnitude of its movement.

During the optimization process, particles update their velocities and positions iteratively based on two forms of knowledge: their own best found position (Pbest) and the best position identified by entire swarm (gbest). The velocity update is influenced by an inertia term, cognitive component and a social component. This mechanism allows to particle to balance exploration and exploitation of the search space. The algorithm proceeds until the stopping condition is satisfied, typically when the maximum number of iterations is reached or the desired solution accuracy is attained.

- *Velocity Update Equation*

$$V_i^{t+1} = \omega . V_i^{t} + c_1 . r_1 . (P_i^t - X_i^t) + c_2 . r_2 . (g^t - X_i^t) \quad (1)$$

Where,
V_i^{t}: Velocity of particle i at iteration t
ω : Inertia weight
c_1, c_2 : Acceleration coefficient
r_1, r_2 : Random numbers
P_i^t: Personal Best Position of the particle i
g^t: Global Best Position found by the swarm
X_i^t : Current position of particle i

- *Position Update Equation*

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

Where,
X_i^t : Position of particle i at iteration t
V_i^{t+1} : Updated velocity

II. RELATED WORKS

PSO has attracted significant attention as an alternative or complementary optimization technique in neural networks. Gad [8] provided a comprehensive review of PSO applications in deep learning, highlighting its use in weight optimization, hyper parameter tuning, and feature selection. The study emphasized PSO's derivative-free nature, which allows it to function effectively in highly nonlinear optimization landscapes, as well as its promise in identifying discriminative features. However, Gad noted that direct application of PSO to large-scale architectures such as ResNet is computationally expensive, limiting its scalability. The review therefore concludes that hybrid approaches, which integrate PSO with gradient-based methods, offer a more feasible direction for deep learning optimization.

Banks et al. [9] examined early applications of PSO in neural networks, focusing on shallow models such as feedforward neural networks and multilayer perceptrons. In these contexts, PSO was successfully applied to optimize network weights and biases as an alternative to traditional backpropagation. While this demonstrated the potential of PSO as a training method, the review was conducted before the widespread adoption of deep learning, and thus its scope was limited to relatively small architectures with modest parameter counts.

Yang [10] positioned PSO within the broader family of nature-inspired metaheuristic algorithms, noting its simplicity, minimal parameter requirements, derivative-free optimization, and strong global search ability. Nevertheless, he also identified several drawbacks that restrict its robustness in complex problem spaces, including premature convergence, sensitivity to parameter tuning, poor scalability in high-dimensional search spaces, and the lack of rigorous theoretical guarantees of convergence. These limitations suggest that, while PSO remains an attractive global optimizer, it requires modifications or hybridization to remain effective in large-scale, real-world applications.

To address some of these limitations, Serizawa and Fujita [11] proposed the Linearly Decreasing Weight Particle Swarm Optimization (LDW-PSO) for optimizing Convolutional Neural Networks (CNNs). By linearly decreasing the inertia weight, LDW-PSO improved the balance between exploration and exploitation, enabling more effective global search in the early stages and better convergence in later stages. This approach demonstrated advantages such as more stable convergence, reduced risk of premature stagnation, and higher classification accuracy compared to standard PSO and gradient-based methods. However, the study also revealed persistent drawbacks, including high computational cost for swarm-based optimization at each training step, scalability challenges when applied to deeper networks, and the need for careful parameter tuning.

The reviewed literature collectively establishes PSO as a promising optimization strategy for neural networks, with demonstrated success in shallow architectures, hyperparameter tuning, and CNN-level optimizations. However, significant limitations remain when extending PSO to very deep architectures such as ResNet-50. Challenges include the high computational overhead of swarm-based optimization in networks with millions of parameters, the tendency toward premature convergence in high-dimensional search spaces, and difficulties in integrating PSO efficiently with gradient-based training. Although variants such as LDW-PSO show potential improvements, their scalability to deeper models remains underexplored. Therefore, the research gap lies in developing enhanced PSO strategies that can be efficiently embedded into deep architectures like ResNet-50, balancing global search capability with computational feasibility to improve training performance without incurring excessive cost.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov358

III. MATERIALS AND METHODS

In this study, PSO is employed as an optimization strategy for training deep neural networks on large-scale datasets such as CIFAR 10. Unlike conventional applications of PSO that primarily optimized network weights, the proposed approach extends the search space to also include key training hyper parameters.

Each particle X_i in the swarm is represented as a joint vector of hyperparameters and network weights:

$$X_i = [\eta, m, \lambda, b, w] \quad (3)$$

Where,
η : Learning rate
m: Momentum coefficient
λ: Weight decay factor
b : Batch size
ω: Trainable weights

To ensure the efficient model training and fair comparison, a set of hyperparamters was defined within specific ranges. These parameters were tuned during the optimization process using SGD and PSO. Table 1 summarizes the range of values considered for each hyperparameter.

Table 1 Hyperparameter Search Space for Model Training

Parameter	**Range**
Learning Rate (η)	[0.0001,0.1]
Momentum Coefficient (m)	[0.5,0.99]
Weight Decay Factor (λ)	$[1 \times 10^{-6}, 1 \times 10^{-3}]$
Batch Size (b)	{64,128,256}
Trainable Weights (ω)	Learned during training

The overall workflow for the proposed PSO-enhanced ResNet-50 model applied to the CIFAR-10 dataset is illustrated in Figure 1.

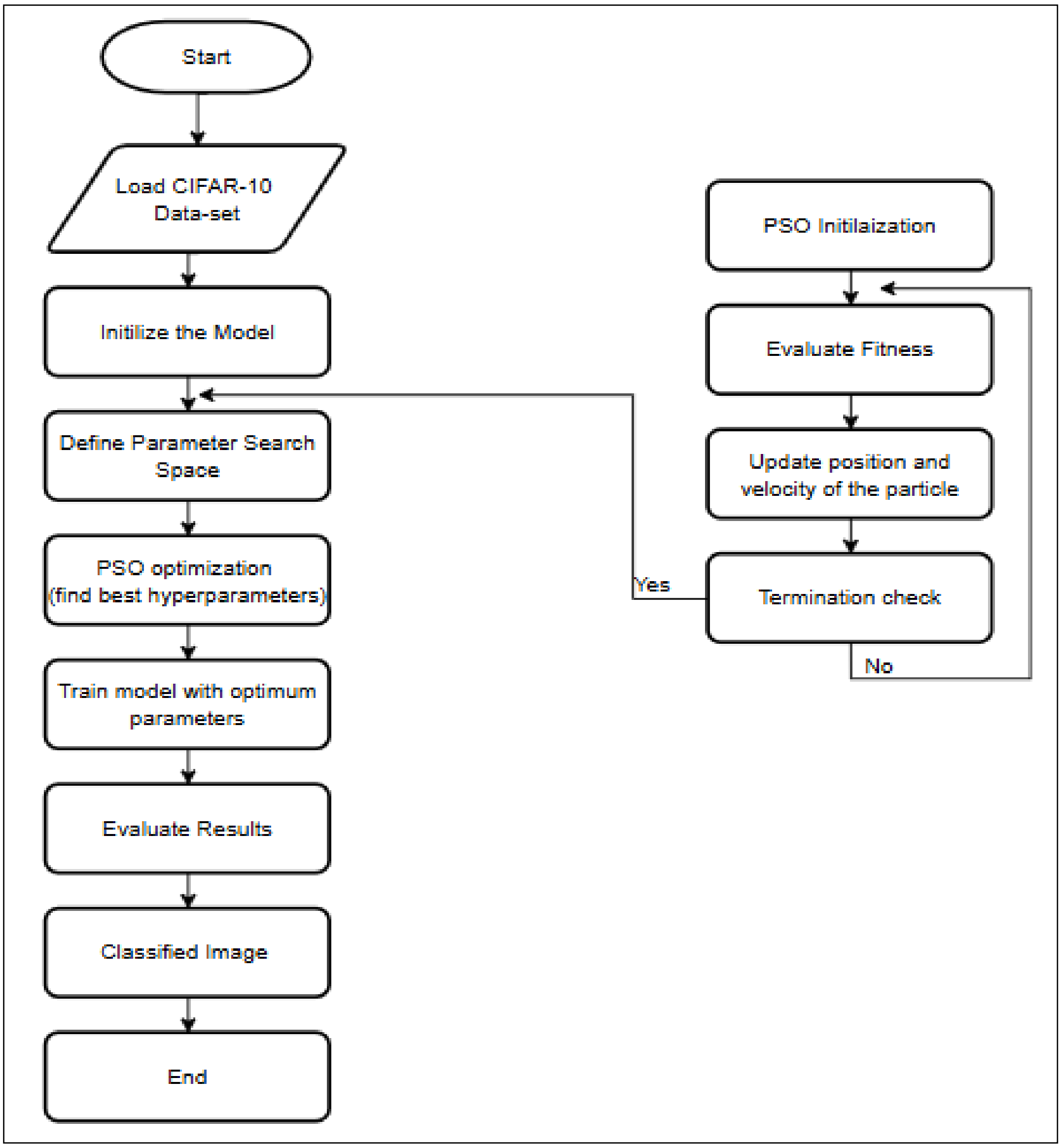

Fig 1 Flowchart of the Proposed Methodology

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov358

IV. RESULTS AND DISCUSSION

The experimental results obtained from training the ResNet-50 model using SGD, Standard PSO, and the Enhanced PSO approach are presented in this section. Each model was trained for ten epochs under identical experimental settings, and the validation accuracy was evaluated after each epoch. The corresponding performance outcomes are summarized in Table2.

Table 2 Validation Accuracy of Comparison of ResNet-50 Model Trained Using SGD, Standard PSO, and Enhanced PSO Approaches.

Epoch No.	**SGD (%)**	**Standard PSO (%)**	**Enhanced PSO (%)**
1	91.93	16.41	93.32
2	93.48	17.19	94.82
3	93.85	17.19	94.76
4	94.06	17.19	94.78
5	94.34	17.19	95.51
6	94.31	17.58	95.46
7	94.36	17.58	95.43
8	94.47	17.97	95.43
9	94.26	17.97	95.54
10	94.40	17.97	95.72

The ResNet-50 model trained using SGD exhibited rapid convergence during the initial epochs, achieving a stable validation accuracy after the fifth epoch. However, the improvement rate slowed down subsequently, indicating early saturation and limited exploration capability. In contrast the Standard PSO showed poor convergence behavior, resulting in considerably lower validation accuracy due to premature stagnation and inadequate parameter tuning. The Enhanced PSO approach, however, demonstrated a smoother and more consistent increase in validation accuracy across epochs, reflection its superior balance between exploration and exploitation during the optimization process.

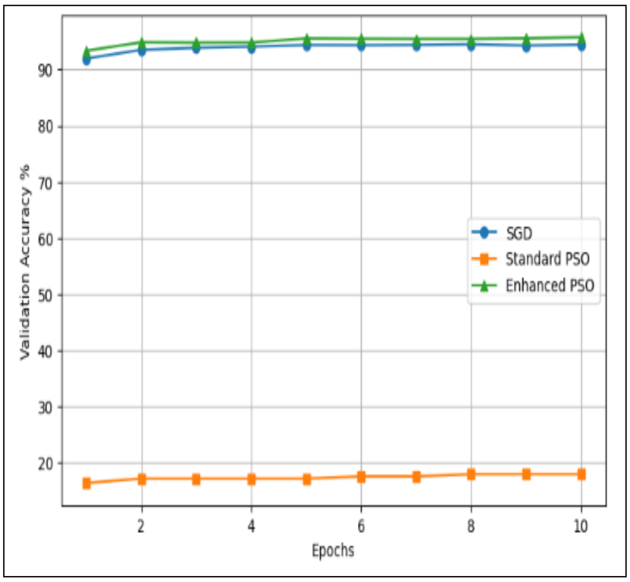

Fig 2 Validation Accuracy with Three Approaches On CIFAR 10 Dataaset.

According to the figure 2, the Enhanced PSO optimizer demonstrates superior and consistent validation accuracy compared to the Standard PSO and performs comparably or slightly better than SGD. The Enhanced PSO's stability across epochs indicates improved convergence behavior, effective exploration-exploitation balance, and strong generalization capability.

In contast, the standard PSO shows very poor accuracy, suggesting convergence stagnation and inefficient parameter updates. The SGD maintains a strong baseline, confirming its reliabilty for gradient-based optimization, but the Enhanced PSO slightly outperfoms it by achieving similar accuracy with greater stability and potentially faster convergence.

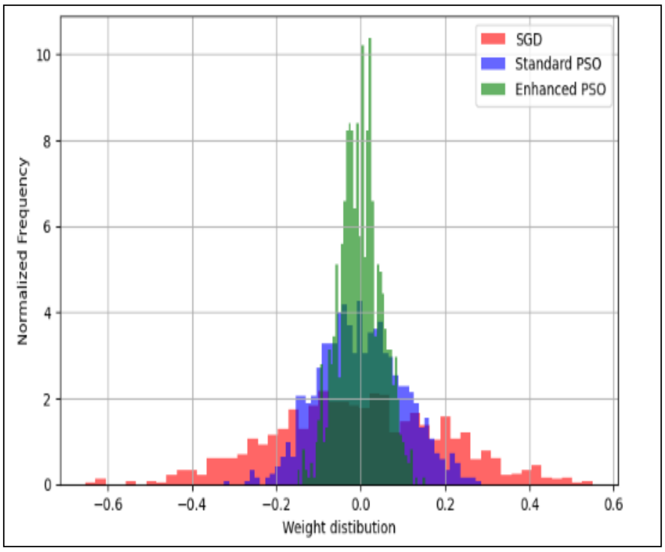

Fig 3 Weight Distribution After Training

The weight distribution analysis figure 3 reveals that the Enhanced PSO model achieves a significantly narrower and more centered distribution of weights compared to SGD and Standard PSO. The narrow peak around zero implies a well regularized learning process with reduced variance and improved gradient stability. Conversely, the broader distribution observed in SGD indicates unstable convergence and potential overfitting. Standard PSO performs moderately, offering better stability than SGD but less control than the Enhanced PSO confirms its superior ability to guide the optimization toward globally stable solutions, ensuring efficient learning and strong generalization performance.

The comparative analysis demonstrates that the Enhanced PSO optimizer significantly improves model performance over both the conventional SGD and the Standard PSO approaches. While SGD achieves high accuracy through local gradient updates, it remains sensitive to learning rates and initialization conditions. The Standard PSO, on the other hand, shows poor generalization due to uncontrolled particle convergence and lack of adaptive tuning. In contrast, the Enhanced PSO achieves superior stability and convergence by dynamically regulating particle movement and maintaining firm weight distributions. This hybridized optimization approach effectively bridges the exploration exploitation gap, resulting in consistently higher validation accuracy and improved learning robustness.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov358

V. CONCLUSION

This study presented and Enhanced PSO approach for the efficient training of deep neural networks, particularly the ResNet-50 architecture on large-scale datasets such as CIFAR-10. The experimental outcomes demonstrated that the proposed Enhanced PSO significantly outperformed the conventional Standard PSO and achieved comparable or slightly superior performance to SGD.

The Enhanced PSO maintained consistently high validation accuracy throughout the training process, indicating improved convergence stability and a balanced exploration-exploitation mechanism. The weight distribution analysis further confirmed that the Enhanced PSO produced a narrow and centered distribution of weights, reflecting a well, regularized and stable learning process. In contrast, the Standard PSO suffered from premature convergence and poor generalization, while SGD although reliable, was limited by sensitivity to hyperparameter selection and slower adaptability to dynamic search landscapes.

Overall, the proposed Enhanced PSO effectively bridges the gap between swarm-based global optimization and gradient-based local refinement. Its derivative free nature, combined with adaptive control of particle dynamics, enables robust training in high-dimensional spaces with reduced variance and faster convergence. Future research may extend this framework to hybrid architectures or transfer learning applications, optimizing both network parameter and hyperparameters simultaneously to further enhance scalability and computational efficiency across diverse deep learning domains. Hyperparameters simultaneously to further enhance scalability and computational efficiency across diverse deep learning domains.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Computational Mathematics, University of Moratuwa, for providing the computational resources and research support necessary for this study.

REFERENCES

[1]. Zhang H, et al. Developing a novel artificial intelligence model to estimate the capital cost of mining projects using deep neural network-based ant colony optimization algorithm. *Resour Policy.* 2020;66:101604. doi:10.1016/j.resourpol.2020.101604.
[2]. Aje OF, Josephat AA. Global Journal of Engineering and Technology Advances. *Glob J Eng Technol Adv.* 2020;3(3):1-6. doi:10.30574/gjeta.
[3]. Atugoda AWK, Fernando S. Improved Particle Swarm Optimization for Optimizing the Deep Convolutional Neural Network. In: *Proceedings of the International Conference on Information Technology Research (ICITR)*; 2023 Dec 8–9; Moratuwa, Sri Lanka. IEEE; 2023. p. 45–50.
[4]. Zhao X, Li J, Liu Y, Wang C. An efficient swarm intelligence approach to the optimization on high-dimensional problems. *Front Comput Neurosci.* 2024;18:1283974. doi:10.3389/fncom.2024.1283974.
[5]. Zhang Y, Wang S, Ji G. Swarm intelligence and its applications. *Sci World J.* 2013;2013:528069. doi:10.1155/2013/528069.
[6]. Nickabadi A, Ebadzadeh MM, Safabakhsh R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl Soft Comput J.* 2011;11(4):3658-70. doi:10.1016/j.asoc.2011.01.037.
[7]. Kennedy J, Eberhart R. Particle swarm optimization. *Proc IEEE Int Conf Neural Netw.* 1995;4:1942-8. doi:10.1109/ICNN.1995.488968.
[8]. Gad AG. Particle swarm optimization algorithm and its applications: A systematic review. *Arch Comput Methods Eng.* 2022;29(5):2531-61. doi:10.1007/s11831-021-09588-4.
[9]. Banks A, Vincent J, Anyakoha C. A review of particle swarm optimization. Part I: Background and development. *Nat Comput.* 2007;6(4):467-84.2. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–444.
[10]. Yang XS. *Nature-Inspired Optimization Algorithms.* Oxford: Elsevier; 2014.
[11]. Serizawa T, Fujita H**.** Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization. *arXiv preprint* arXiv:2001.05670. 2020.