http://dx.doi.org/10.21622/ACE.2025.05.2.1799

SWARM INTELLIGENCE-DRIVEN MOBILENET OPTIMIZATION FOR BREAST CANCER CLASSIFICATION IN ULTRASOUND IMAGES

Marwa A. Elshenawy and Rania Kadry

Arab Academy for Science, Technology and Maritime Transport, College of Engineering and Technology, Computer Engineering Department, Egypt

Emails: {marwa_elshenawy@aast.edu, rania_kadry@aast.edu}

Received on, 15 November 2025 - Accepted on, 24 November 2025 - Published on, 22 December 2025

ABSTRACT

This study examines the effectiveness of swarm intelligence algorithms for optimizing MobileNet hyperparameters in breast cancer classification using ultrasound images (BCMID). Three optimization methods—Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), and the Whale Optimization Algorithm (WOA)—were applied to identify optimal learning rates, dropout rate, and the optimizer. The best hyperparameter sets discovered by each algorithm were used to retrain MobileNet to verify consistency and performance stability. The dataset consisted of clinically annotated breast ultrasound images representing benign, malignant, and normal cases. Model performance was assessed using accuracy, macro-precision, macro-recall, and macro-F1-score. The optimized models outperformed the baseline configuration, with ABC achieving 62%, PSO achieving 66%, and WOA achieving 62%. In terms of computational time, PSO required 7710 seconds, ABC 14,148 seconds, and WOA 7622 seconds, highlighting notable differences in optimization efficiency. These findings demonstrate that swarm-based optimization can enhance MobileNet's diagnostic performance while exhibiting varying computational costs, offering a reliable framework for computer-aided breast cancer detection in ultrasound imaging.

Keywords: Machine Learning, Breast Cancer Classification, Hyperparameter Optimization, Swarm Optimizers, BCMID ultrasound images

1. INTRODUCTION

The use of Artificial Intelligence (AI) in clinical practice is reshaping modern healthcare, particularly in oncology, where AI is becoming increasingly valuable for enhancing diagnostic precision, customizing treatment plans, and advancing research efforts. Cancer remains one of the primary global causes of illness and death, with roughly 18 million new cases and 10 million deaths recorded each year in recent reports, excluding non-melanoma skin cancers [1]. Among all cancer types, breast cancer is especially noteworthy. In 2022, it was the most frequently diagnosed cancer among women, accounting for over 2.3 million new cases and approximately 670,000 deaths worldwide [2]. Projections indicate that if current trends continue, the number of new breast cancer diagnoses may rise by 38%, while mortality could increase by 68% by the year 2050, reaching an estimated 3.2 million new cases and 1.1 million deaths annually [3]. Within this landscape, AI's capacity to process and interpret large, complex datasets—such as medical imaging, genomic profiles, and electronic health records—enables earlier detection, improved risk stratification, and personalized treatment strategies supported by precision oncology, ultimately contributing to reduced mortality and better patient outcomes [4–7].

Deep neural networks (DNNs) have emerged as the leading method for classifying breast lesions in ultrasound (US) imaging, showing notable gains in diagnostic accuracy and clinical interpretability. In the period following 2021, most studies relied on transfer learning using established convolutional neural network (CNN) models such as ResNet, DenseNet, and EfficientNet, which achieved strong accuracy and AUC values when distinguishing benign from malignant lesions on widely used datasets. More recent research, however, has shifted toward designing specialized DNN architectures that better capture the unique properties of ultrasound imaging [8,9]. These include attention-enhanced CNNs, hybrid CNN–transformer systems, and multi-view fusion networks that combine information from several US image planes to provide richer contextual understanding [10].

To further improve model generalization, researchers have incorporated strategies such as synthetic data generation with GANs, domain adaptation, and ensemble-based methods, all of which help minimize overfitting and bolster robustness across different ultrasound machines and scanning protocols [11,12]. Collectively, post-2021 DNN-based classifiers have progressed from straightforward binary models to more clinically focused and interpretable systems, reaching performance levels comparable to expert radiologists in controlled settings and advancing steadily toward deployment in real-world clinical environments.

Hyperparameter optimization plays a critical role in enhancing the accuracy and overall performance of convolutional neural network (CNN) classification models. Hyperparameters—such as learning rate, batch size, dropout rate, number of filters, and optimizer settings—directly influence how effectively a CNN learns patterns from data. Poorly chosen hyperparameters can lead to issues like slow convergence, underfitting, overfitting, or unstable training. Systematic optimization methods, including grid search, Bayesian optimization, evolutionary algorithms, and particle swarm optimization, help identify the optimal combination of hyperparameters that maximizes classification accuracy. By fine-tuning these values, models achieve better feature extraction, improved generalization to unseen data, and more robust decision-making. As a result, hyperparameter optimization is essential for unlocking the full potential of CNN architecture and ensuring high-quality, reliable classification performance.

Our work makes a significant contribution to the field of CNN-based medical image classification by systematically exploring multiple transfer learning models and leveraging various swarm optimization algorithms to optimize their hyperparameters. While transfer learning allows CNNs to benefit from pre-trained knowledge on large datasets, the choice of model and its hyperparameters greatly impacts performance. By integrating different swarm intelligence methods—such as Particle Swarm Optimization, Ant Colony Optimization, and others- our approach efficiently searches the hyperparameter space to identify optimal configurations, improving both convergence and classification accuracy. This comprehensive strategy not only enhances model performance but also provides a versatile framework for applying CNNs to diverse datasets, making it a robust and generalizable contribution to the development of high-accuracy, optimized deep learning models. The main contributions of our work are as follows:

1. We explore the optimization of hyperparameters across a transfer learning model on breast cancer ultrasound images.

2. Different swarm optimization algorithms, including Particle Swarm Optimization (PSO) and other swarm-based methods, are employed to efficiently identify the optimal MobileNet hyperparameter and the optimal swarm optimizer.

3. We systematically compare the performance of different optimizers in terms of both classification accuracy and computational time, providing insights into their effectiveness and efficiency.

The remainder of the article is structured as follows: Section 2 introduces the study and reviews related works, Section 3 details the materials and methods, Section 4 presents the experiments and results, and Section 5 provides conclusions and recommendations for future research.

http://dx.doi.org/10.21622/ACE.2025.05.2.1799

2. RELATED WORK

Deep learning, particularly Convolutional Neural Networks (CNNs), has become the foundation of modern breast cancer classification due to its ability to automatically extract high-level imaging features. Transfer learning has further accelerated progress by leveraging pretrained models such as VGG, ResNet, DenseNet, Inception, and NASNet for small medical datasets. For breast ultrasound images specifically, Kormpos et al. evaluated multiple pretrained architectures and proposed a hierarchical two-stage classifier that improved robustness over conventional flat multi-class models, demonstrating the effectiveness of CNNs for BUSI ultrasound classification [11]. Ahishakiye and Kanobe later introduced a hybrid DenseNet201 + Bayesian-Optimized FLN framework that achieved high accuracy by optimizing key learning parameters such as dropout, learning rate, and hidden-layer size [12]. These studies highlight both the potential of pretrained CNNs and the Sensitivity of their performance to hyperparameter selection.

Hyperparameter optimization has therefore emerged as a critical research direction. Among metaheuristic techniques, Particle Swarm Optimization (PSO) is one of the earliest and most widely adopted. Aguerchi et al. used PSO to optimize CNN hyperparameters—including filter size, stride, and network depth—for mammography, achieving accuracies above 98% on DDSM and MIAS datasets [13]. Their work remains a benchmark showing that swarm intelligence can efficiently explore complex CNN design spaces. Similarly, several PSO-CNN studies in other medical imaging domains report improvements in learning rate tuning, dropout selection, layer freezing, and classifier training when applied to pretrained models such as VGG-16, ResNet50, and DenseNet121.

Beyond PSO, other swarm-intelligence algorithms have been widely applied to CNN tuning. Ant Colony Optimization (ACO) has been used by Fauzi et al. to optimize ultrasound classification models based on texture features, outperforming non-optimized baselines [14]. Genetic Algorithms (GA) have long been adopted for evolving CNN architectures and optimizing filter parameters [15]. Grey Wolf Optimizer (GWO) [16] and Whale Optimization Algorithm (WOA) [17] have been used to tune deep architectures such as ResNet and Inception, often providing stronger convergence stability than PSO. Differential Evolution (DE) [18] has been employed to tune convolutional layers and learning hyperparameters, while Firefly Algorithm (FA) [19] and Cuckoo Search (CS) [20] have been reported to enhance CNN training performance across multiple medical image analysis tasks.

Overall, the literature demonstrates that swarm-intelligence optimizers play an increasingly important role in improving CNN classification performance, especially for medical images where fixed hyperparameters often fail to generalize. However, only limited studies systematically apply swarm-based optimization—particularly PSO and related algorithms—to multiple pretrained CNN models specifically for breast ultrasound images. This gap motivates the present study, which evaluates several CNN architectures and employs swarm-intelligence optimization to automatically identify hyperparameter configurations that maximize performance on breast ultrasound cancer classification.

3. MATERIALS AND METHODS

3.1. DATASET

In this section, we provide a detailed description of the breast ultrasound dataset used to evaluate our work. Moreover, Figure 1 illustrates samples of normal, benign, and malignant cases from the dataset. The Breast Cancer Multimodal Imaging Dataset (BCMID) [21] includes data from 323 adult female patients collected between 2019 and 2022, aged 26–82 years. Acquired at Ayadi Hospital in Alexandria, Egypt, the dataset contains ultrasound and mammography scans, along with validated reports summarizing the physician's BI-RADS-based diagnosis. For this study, only ultrasound images were used. The dataset comprises three classes—normal (152 images), benign (734 images), and malignant (259 images). Multiple ultrasound views or scans from different dates per patient result in a total number of images exceeding the number of patients.

http://dx.doi.org/10.21622/ACE.2025.05.2.1799

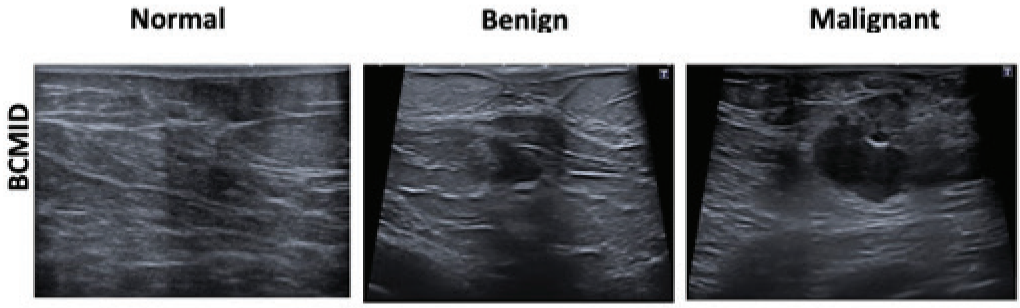

Figure 1: Ultrasound image samples from the BCMID dataset showing normal, benign, and malignant cases

3.2. DATA PREPROCESSING AND AUGMENTATION

Before starting to run the model, all available ultrasound images with validated diagnostic labels (normal (N), benign(B), or malignant(M))were included in the analysis, while corrupted, duplicate, or unlabeled images were excluded. Only ultrasound images were used. The dataset was divided using the ratio of 70% for training, 20% for testing, and 10% for validation. The data preprocessing pipeline also incorporated a dataset-specific procedure in which BCMID images were manually cropped to ensure that all clinically relevant regions were clearly included. This manual cropping step allowed the preprocessing stage to better focus on the essential areas of interest within each image. Subsequently, all images were resized to 224×224 pixels to comply with the input dimensions required by the pre-trained machine learning models. Pixel intensities were normalized to the [0,1] range to maintain consistency across the dataset. These preprocessing procedures were applied uniformly to the training, validation, and test sets in all experimental configurations.

To address the class imbalance depicted in Table 1, without altering the size or distribution of the dataset, we applied class weighting during model training. Instead of down-sampling or up-sampling the classes (which would either reduce the dataset excessively or introduce significant computational overhead), we assigned higher weights to underrepresented classes and lower weights to overrepresented ones as delivered in Figure 2. This approach allowed the model to pay proportionally more attention to minority classes while preserving the original dataset and ensuring efficient training.

To address the class imbalance present in the training data, class weights were computed and applied during model training. First, the class labels generated by the data loader were converted into their corresponding integer representations. These integer labels were then used with the balanced mode of "compute_class_weight", which calculates a weight for each class based on the inverse proportionality between class frequency and its representation in the dataset. In this approach, classes with fewer samples receive higher weights, while more frequent classes receive lower weights, ensuring that the model does not become biased toward the majority class. The resulting weight vector was subsequently transformed into a dictionary format compatible with Keras and passed to the model during training to promote balanced learning across classes.

Table 1: Detailed class distribution across the dataset

	Train			Validation			Test			Total
	N	B	M	N	B	M	N	B	M	
Dataset	96	499	184	18	75	22	38	160	53	1145
Total	779			115			251			1145

```
Class Labels: {'Benign': 0, 'Malignant': 1, 'Normal': 2}
Class Weights: {0: 0.5176314038589488, 1: 1.4094202898550725, 2: 2.7885304659498207}
```

Figure 2: Weight adjustments of classes

3.3. PRE-TRAINED MODEL

Researchers often utilize pre-trained models rather than training networks from scratch. Among the transfer learning architectures summarized in Table 2, the number of parameters was a key factor in our selection. MobileNetV2, with its lightweight design and exceptionally low parameter count, was chosen as the most suitable model for this study. Its compact architecture enables efficient training and reduced computational demands, which is particularly advantageous in the context of swarm-based hyperparameter optimization, where the model must be repeatedly trained and evaluated across multiple configurations. Furthermore, MobileNetV2 is well-suited for datasets of moderate size, such as BCMID, while maintaining competitive performance in medical image classification tasks.

Table 2: Transfer Model's parameters

Transfer Model Architecture	Approximate Number of Parameters
MobileNetV2	~3.5 million
MobileNetV1	~4.2 million
EfficientNetB0	~5.3 million
DenseNet121	~7 million
Xception	~22.9 million
ResNet50	~23.5 million
InceptionV3	~23.8 million
DenseNet201	~20 million
ResNet101	~44.5 million
ResNet152	~60 million
EfficientNetB7	~66 million
VGG16	~138 million
VGG19	~144 million

The MobileNetV2 [22] family of models was adopted in this study due to its emphasis on computational efficiency, making it particularly suitable for low-resource or edge computing environments. The original MobileNetV2 introduced depth-wise separable convolutions, which significantly reduce both the number of parameters and computational operations while maintaining high accuracy for visual recognition tasks. This design makes MobileNetV2 well-suited for image classification and object recognition on devices with limited computational capabilities, such as smartphones and IoT devices. Its efficiency allows for the deployment of complex neural network architectures in resource-constrained settings, enabling advanced machine learning applications in mobile and embedded contexts.

MobileNetV2 enhances the original MobileNet architecture using inverted residuals and linear bottlenecks, enabling efficient feature extraction with minimal computational cost. Its performance is controlled by several key hyperparameters:

- **Epochs:** Specifies the number of complete passes through the training dataset; optimizing epochs balances sufficient training for convergence with computational

http://dx.doi.org/10.21622/ACE.2025.05.2.1799

efficiency.

- **Learning rate:** Determines the step size at each iteration during training; a properly tuned learning rate ensures faster convergence and prevents divergence or overshooting.

- **Batch size:** Specifies the number of samples processed before updating model weights; larger batches provide more stable gradient estimates but require more memory.

- **Dropout rate:** Randomly deactivates a fraction of neurons during training to prevent overfitting, promoting better generalization.

- **Optimizer choice:** Influences how the network updates its weights; different optimizers (e.g., SGD, Adam, RMSprop) affect convergence speed, stability, and final model accuracy.

Careful tuning of these hyperparameters is crucial, particularly in medical image classification tasks, as it directly impacts model accuracy, training efficiency, and generalization while maintaining the computational efficiency that makes MobileNetV2 suitable for moderate-resource environments.

3.4. SWARM-BASED OPTIMIZERS

Swarm intelligence optimization refers to a family of population-based metaheuristic algorithms inspired by the collective behavior of decentralized biological systems such as bird flocks, fish schools, ant colonies, and bee swarms [23]. These algorithms operate through simple interactions among agents that cooperate to explore and exploit the search space efficiently, without requiring gradient information or problem-specific assumptions. Their robustness, parallelism, and ability to avoid local minima make swarm-based optimizers particularly effective for complex optimization problems in machine learning, engineering, pattern recognition, and deep learning hyperparameter tuning.

3.4.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization, introduced by Kennedy and Eberhart [24], models the social behavior of flocks of birds or schools of fish. Particles navigate the search space by adjusting their velocity based on their own best experience and the best-performing member of the swarm. PSO is widely used due to its simplicity, fast convergence, and efficiency in continuous optimization tasks, making it a strong candidate for neural network hyperparameter tuning and feature selection.

3.4.2. Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm, developed by Mirjalili and Lewis [25], is inspired by the bubble-net feeding strategy of humpback whales. WOA alternates between encircling prey, spiral bubble-net attacks, and random exploration, allowing it to escape local minima while converging toward optimal solutions. Its low complexity and strong exploitation ability make it effective for deep learning model optimization.

3.4.3. Artificial Bee Colony (ABC)

The Artificial Bee Colony algorithm, proposed by Karaboga [26], mimics the foraging behavior of honeybee swarms, dividing bees into employed, onlooker, and scout categories. ABC excels at balancing exploration and exploitation by dynamically allocating search effort based on nectar (fitness) quality. This makes it highly effective for feature selection, clustering, and neural network training.

http://dx.doi.org/10.21622/ACE.2025.05.2.1799

4. EXPERIMENTS AND RESULTS

This section covers experimental design, where MobileNetV2 is employed as the primary pre-trained convolutional neural network for breast ultrasound image classification. Due to its lightweight architecture and low computational cost, MobileNetV2 serves as an ideal baseline for evaluating hyperparameter optimization strategies. In this study, the model's key hyperparameters are tuned using three swarm-based optimization algorithms—Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), and Artificial Bee Colony (ABC). Each optimizer searches for an optimal configuration that enhances MobileNetV2's performance while maintaining computational efficiency.

The experimental workflow is divided into two main phases. In the first phase, all selected swarm-based hyperparameter optimization algorithms are applied to MobileNetV2 to identify their optimal configurations. During this step, both the computational time and the best hyperparameter solution produced by each optimizer are recorded. In the second phase, the best hyperparameters obtained from each algorithm are used to retrain MobileNetV2 on the same training set (with more epochs =100), after which the model is evaluated using the independent test dataset. This allows for a fair comparison of the algorithms based on their achieved evaluation metrics as well as their optimization efficiency, providing a comprehensive assessment of performance across both accuracy and runtime.

4.1. EXPERIMENTAL SETTINGS

The models utilized across all experiments were built using the Keras API with a TensorFlow backend. For multi-class classification, the output layer used a softmax activation function, and training was driven by the categorical cross-entropy loss. To ensure full reproducibility, a fixed random seed was applied, and TensorFlow was configured for deterministic behavior. Experiments were carried out on an Intel® Core™ i7-10510U CPU @ 1.80 GHz with 16 GB of RAM. The implementation of the swarm-based hyperparameter optimization was facilitated by the MEALPY Python library [27], which provides a comprehensive suite of state-of-the-art population-based meta-heuristic algorithms.

Table 3: Hyperparameter Settings

Model	Hyperparameter	Fixed value	Optimized Range
MobileNetV2	Epochs	3	-
	Learning Rate	-	0.1,0.01,0.001,0.0001
	Dropout	-	0.1,0.2,0.3,0.4,0.5,0.6,0.7
	Batch size	32	-
	Optimizer	-	SGD, Adam, RMSprop
PSO	Iterations=5, population size=5, w=0.4, c1= c2=2.05		
ABC	Iterations=5, population size=5, n_ limits=25		
WOA	Iterations=5, population size=5		

The hyperparameters for MobileNetV2 and the selected swarm optimization algorithms are summarized in Table 3. To ensure a fair and consistent comparison across all optimizers, both the number of iterations and the population size were fixed for every swarm-based algorithm evaluated in this study.

4.2. EVALUATION METRICS

To evaluate the effectiveness of the different methods in classifying breast ultrasound images into normal, benign, and malignant categories, several widely used performance metrics were employed: Accuracy, Macro Precision, Macro Recall, and Macro F1-Score.

http://dx.doi.org/10.21622/ACE.2025.05.2.1799

These metrics offer a comprehensive assessment of classification quality, with the macro-averaged measures being particularly valuable for datasets with class imbalance (such as those commonly encountered in medical imaging) since they weigh each class equally regardless of its frequency.

Accuracy is the total number of correctly predicted samples (regardless of class) divided by the total number of samples in the dataset. For multi-class problems (like normal, benign, malignant), accuracy is typically computed as depicted in equation (1):

$$\text{Accuracy}\,\frac{Number\ of\ Correct\ Predictions\ Across\ All\ Classes}{\boldsymbol{Total\ Number\ of\ Predictions}} \quad (1)$$

By first computing the Precision (equation 2) for each class individually, Macro Precision is calculated, then taking the average across all classes as given in equation (3).

$$\text{Precision (per class)}= \frac{Number\ of\ True\ Positives\ for\ that\ class}{/(Number\ of\ True\ Positives\ for\ that\ class+Number\ of\ False\ Positives\ for\ that\ class} \quad (2)$$

$$\text{Macro Precision} = \frac{Precision\ of\ Class\ 1 + Precision\ of\ Class\ 2 + Precision\ of\ Class\ 3}{\mathbf{3}} \quad (3)$$

Macro Recall is calculated by first computing Recall (also called Sensitivity) for each class individually, then taking the average across all classes (equations 4 and 5).

$$\text{Recall (per class)}= \frac{Number\ of\ True\ Positives\ for\ that\ class}{+\ class+Number\ of\ False\ Negatives\ for\ that\ class} \quad (4)$$

$$\text{Macro Recall} = \frac{Recall\ of\ Class\ 1 + Recall\ of\ Class\ 2 + Recall\ of\ Class\ 3}{\mathbf{3}} \quad (5)$$

The F1-Score for each class is the harmonic mean of that class's Precision and Recall (equations 6 and 7).

$$\text{F1-Score (per class)}= \frac{2 \times (\mathrm{Precision} \times \mathrm{Recall})}{\mathrm{Precision} + \mathrm{Recall}} \quad (6)$$

$$\text{Macro F1-Score} = \frac{\mathrm{F1{-}Score\ of\ Class\ 1 + F1{-}Score\ of\ Class\ 2 + F1{-}Score\ of\ Class\ 3}}{\mathbf{3}} \quad (7)$$

4.3. RESULTS AND DISCUSSION

The first phase focuses on evaluating the performance of the selected three swarm-based hyperparameter optimization algorithms when applied to MobileNetV2. Each optimizer is tasked with searching for the most effective hyperparameter configuration using the original training dataset. Throughout this optimization process, both the computational time required by each algorithm and the best hyperparameter set it identifies are documented in Table 4. This phase establishes a clear basis for comparing the optimizers in terms of their search efficiency and the quality of the solutions they produce.

Table 4: Hyperparameter values and inference time

Optimizer	Hyperparameter values			Time (sec)
	Learning Rate	Dropout	Optimizer	
PSO	0.001	0.1	Adam	7710
ABC	0.01	0.1	SGD	14148
WOA	0.01	0.1	SGD	7622

The second phase involves retraining MobileNetV2 using the optimal hyperparameters obtained from each swarm-based optimization algorithm depicted in Table 4. Using the same training dataset as in the first phase, the model is trained again, but an extended number of epochs (100) is used to ensure a more stable and representative evaluation of each hyperparameter configuration. After training, the resulting models are assessed on an independent test dataset. This phase enables a direct and fair comparison of the algorithms based on their final classification performance, allowing both their optimization effectiveness and practical impact on model accuracy to be fully evaluated. The performance measures of each model are delivered in Table 5.

Table 5: Model evaluations

Optimizer	Evaluation Measures			
	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
PSO	66%	58%	51%	54%
ABC	62%	53%	49%	50%
WOA	62%	53%	49%	50%

The combined analysis of computational time (Table 4) and evaluation metrics reveals clear differences in efficiency (Table 5) and effectiveness among the three swarm optimization algorithms. PSO delivered the best overall classification performance, achieving the highest accuracy (66%) and leading macro precision, recall, and F1-score. Notably, this superior performance was obtained with a moderate training time of 7710 seconds, making PSO both effective and computationally reasonable.

ABC, despite requiring the longest optimization time at 14,148 seconds—almost double that of PSO—did not translate its extended search into improved performance. Its accuracy (62%) and macro metrics remained lower, matching those of WOA. WOA, in contrast, achieved the same evaluation scores as ABC but required only 7622 seconds, demonstrating far greater computational efficiency.

Overall, PSO offers the best balance between accuracy and time cost, while WOA provides the most efficient runtime among the algorithms, yielding mid-level performance. ABC appears to be the least efficient, with high computation time but no corresponding gain in model quality.

5. CONCLUSION AND FUTURE WORK

The work investigated the use of swarm intelligence algorithms such as ABC, PSO, and WOA for hyperparameter optimization of MobileNet in breast cancer ultrasound image (BCMID) classification. The work was conducted in two phases. In the first phase, each algorithm was used to optimize critical hyperparameters, namely learning rate, dropout rate, and optimizer selection. In the second phase, MobileNet was retrained using optimized hyperparameters to ensure consistency and performance stability. The dataset was carefully managed with class weighting to address imbalance among benign, malignant, and normal cases. PSO achieved the highest classification performance, with 66% accuracy and superior macro-precision, macro-recall, and F1-score, requiring a moderate computational time of 7710 seconds. Both ABC and WOA achieved 62% accuracy, with WOA being more computationally efficient (7622 seconds) compared to ABC (14,148 seconds). These results demonstrate that swarm-based optimization can effectively enhance MobileNet's diagnostic capability, with PSO providing the best balance between accuracy and computational cost.

Future research could explore a broader range of swarm intelligence algorithms beyond ABC, PSO, and WOA. The hyperparameter search space could be expanded to include additional parameters, and the parameters of the swarm algorithms themselves could be optimized for better convergence and efficiency. Additionally, experimenting with different transfer learning models beyond MobileNet may improve performance. Combining these approaches with advanced data augmentation and multimodal imaging

http://dx.doi.org/10.21622/ACE.2025.05.2.1799

techniques could further enhance model generalization and diagnostic reliability. Finally, evaluating the optimized models in real-time clinical workflows would provide valuable insights into their practical applicability in breast cancer diagnosis.

The current study trains the model for only three epochs during hyperparameter optimization and relies on a single train/validation/test split, both of which may limit the reliability of the evaluation. Short training durations can prevent meaningful performance differences between hyperparameter configurations from emerging, while a single data split may introduce evaluation bias—especially with a moderately sized dataset. Future work should therefore increase the number of training epochs during hyperparameter tuning and employ more robust evaluation strategies, such as k-fold cross-validation or repeated hold-out splits, to ensure more reliable and statistically sound performance estimates.

REFERENCES

[1] G. Galvin, "Breast cancer deaths expected to climb by 68% worldwide by 2050, WHO warns ," Dec. 2025, euronews.com. [Online]. Available: https://www.euronews.com/health/2025/02/25/breast-cancer-deaths-expected-to-climb-by-68-worldwide-by-2050-who-warns

[2] "Latest global cancer data: Cancer burden rises to 19.3 million new cases and 10.0 million cancer deaths in 2020 – IARC," Dec. 2020. [Online]. Available: https://www.iarc.who.int/news-events/latest-global-cancer-data-cancer-burden-rises-to-19-3-million-new-cases-and-10-0-million-cancer-deaths-in-2020/

[3] U. Nations, "Breast cancer cases projected to rise by nearly 40 per cent by 2050, WHO warns," Dec. 2025. [Online]. Available: https://news.un.org/en/story/2025/02/1160391

[4] E. N. Huhulea et al., "Artificial Intelligence Advancements in Oncology: A Review of Current Trends and Future Directions," Biomedicines, vol. 13, no. 4, p. 951, Apr. 2025, doi: 10.3390/biomedicines13040951.

[5] J. Zhang, X. Guo, B. Wang, and W. Cui, "Automatic Detection of Invasive Ductal Carcinoma Based on the Fusion of Multi-Scale Residual Convolutional Neural Network and SVM," IEEE Access, vol. 9, pp. 40308–40317, 2021, doi: 10.1109/ACCESS.2021.3063803.

[6] A. Esteva et al., "A guide to deep learning in healthcare," Nat Med, vol. 25, no. 1, pp. 24–29, Jan. 2019, doi: 10.1038/s41591-018-0316-z.

[7] F. Jiang et al., "Artificial intelligence in healthcare: past, present and future," Stroke Vasc Neurol, vol. 2, no. 4, pp. 230–243, Dec. 2017, doi: 10.1136/svn-2017-000101.

[8] H. Afrin, N. B. Larson, M. Fatemi, and A. Alizad, "Deep Learning in Different Ultrasound Methods for Breast Cancer, from Diagnosis to Prognosis: Current Trends, Challenges, and an Analysis," Cancers (Basel), vol. 15, no. 12, p. 3139, Jun. 2023, doi: 10.3390/cancers15123139.

[9] M. Islam, M. Hossain, and S. Ahmed, "Enhancing breast cancer classification with an Efficient Deep Convolutional Neural Network (EDCNN)," Computers in Biology and Medicine , p. 174, 2024.

[10] Q. He, Q. Yang, and M. Xie, "HCTNet: A hybrid CNN-transformer network for breast ultrasound image segmentation," Comput Biol Med, vol. 155, p. 106629, Mar. 2023, doi: 10.1016/j.compbiomed.2023.106629.

[11] C. Kormpos, F. Zantalis, S. Katsoulis, and G. Koulouras, "Evaluating Deep Learning Architectures for Breast Tumor Classification and Ultrasound Image Detection Using Transfer Learning," Big Data and Cognitive Computing, vol. 9, no. 5, p. 111, Apr. 2025, doi: 10.3390/bdcc9050111.

[12] E. Ahishakiye and F. Kanobe, "Breast cancer classification using breast ultrasound images with a hybrid of transfer learning and Bayesian-optimized fast learning network," Discover Artificial Intelligence, vol. 5, no. 1, p. 81, May 2025, doi: 10.1007/s44163-025-00335-4.

[13] K. Aguerchi, Y. Jabrane, M. Habba, and A. H. El Hassani, "A CNN Hyperparameters Optimization Based on Particle Swarm Optimization for Mammography Breast Cancer Classification," J Imaging, vol. 10, no. 2, p. 30, Jan. 2024, doi: 10.3390/jimaging10020030.

[14] A. Fauzi, L. E. Lubis, R. Wandira, and S. Musthofa, "Hyperparameter optimization of breast ultrasound image classification models using ant colony optimization based on texture features," Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 4, pp. 673–686, Oct. 2025, doi: 10.35882/ijeeemi.v7i4.266.

[15] J. H. Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[17] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Advances in Engineering Software, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[18] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, Dec. 1997, doi: 10.1023/A:1008202821328.

[19] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," International Journal of Bio-Inspired Computation, vol. 2, no. 2, p. 78, 2010, doi: 10.1504/IJBIC.2010.032124.

[20] X.-S. Yang and Suash Deb, "Cuckoo Search via Lévy flights," in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009, pp. 210–214. doi: 10.1109/NABIC.2009.5393690.

[21] S. Tawfik, N. Ghatwary, A. Elgendy, O. Nasr, xujiong ye, and M. Elshenawy, "BCMID: Breast Cancer Multimodal Imaging Dataset (Updated)," Zenodo (CERN European Organization for Nuclear Research), Dec. 2025, doi: 10.5281/zenodo.14970847.

[22] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

[23] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999. doi: 10.1093/oso/9780195131581.001.0001.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[25] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Advances in Engineering Software, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[26] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization," Technical Report-Erciyes University, 2005.

[27] N. Van Thieu and S. Mirjalili, "MEALPY: An open-source library for latest meta-heuristic algorithms in Python," Journal of Systems Architecture, vol. 139, p. 102871, Jun. 2023, doi: 10.1016/j.sysarc.2023.102871.