frontiers | Frontiers in Oncology

TYPE Original Research
PUBLISHED 22 February 2024
DOI 10.3389/fonc.2024.1347856

OPEN ACCESS

EDITED BY
Jakub Nalepa,
Silesian University of Technology, Poland

REVIEWED BY
Inam Ullah,
Gachon University, Republic of Korea
Imad Rida,
University of Technology Compiegne, France
Yeliz Karaca,
University of Massachusetts Medical School, United States

*CORRESPONDENCE
Anum Masood
anum.masood@ntnu.no
Muhammad Attique Khan
attique.khan@ieee.org

RECEIVED 05 December 2023
ACCEPTED 05 February 2024
PUBLISHED 22 February 2024

CITATION
Jabeen K, Khan MA, Hameed MA, Alqahtani O, Alouane MT-H and Masood A (2024) A novel fusion framework of deep bottleneck residual convolutional neural network for breast cancer classification from mammogram images.
Front. Oncol. 14:1347856.
doi: 10.3389/fonc.2024.1347856

A novel fusion framework of deep bottleneck residual convolutional neural network for breast cancer classification from mammogram images

Kiran Jabeen[1], Muhammad Attique Khan[1,2]*, Mohamed Abdel Hameed[3], Omar Alqahtani[4], M. Turki-Hadj Alouane[4] and Anum Masood[5]*

[1]Department of Computer Science, HITEC University, Taxila, Pakistan, [2]Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon, [3]Department of Computer Science, Faculty of Computers and Information, Luxor University, Luxor, Egypt, [4]College of Computer Science, King Khalid University, Abha, Saudi Arabia, [5]Department of Physics, Norwegian University of Science and Technology, Trondheim, Norway

With over 2.1 million new cases of breast cancer diagnosed annually, the incidence and mortality rate of this disease pose severe global health issues for women. Identifying the disease's influence is the only practical way to lessen it immediately. Numerous research works have developed automated methods using different medical imaging to identify BC. Still, the precision of each strategy differs based on the available resources, the issue's nature, and the dataset being used. We proposed a novel deep bottleneck convolutional neural network with a quantum optimization algorithm for breast cancer classification and diagnosis from mammogram images. Two novel deep architectures named three-residual blocks bottleneck and four-residual blocks bottle have been proposed with parallel and single paths. Bayesian Optimization (BO) has been employed to initialize hyperparameter values and train the architectures on the selected dataset. Deep features are extracted from the global average pool layer of both models. After that, a kernel-based canonical correlation analysis and entropy technique is proposed for the extracted deep features fusion. The fused feature set is further refined using an optimization technique named quantum generalized normal distribution optimization. The selected features are finally classified using several neural network classifiers, such as bi-layered and wide-neural networks. The experimental process was conducted on a publicly available mammogram imaging dataset named INbreast, and a maximum accuracy of 96.5% was obtained. Moreover, for the proposed method, the sensitivity rate is 96.45, the precision rate is 96.5, the F1 score value is 96.64, the MCC value is 92.97%, and the Kappa value is 92.97%, respectively. The proposed architectures are further utilized for the diagnosis process of infected regions. In addition, a detailed comparison has been conducted with a few recent techniques showing the proposed framework's higher accuracy and precision rate.

KEYWORDS

breast cancer, healthcare, bottleneck residual blocks, deep learning, fusion, optimization

1 Introduction

The most frequent tumor in women worldwide is breast cancer (1). Breast cancer ranks as the second most prevalent ailment among women across the globe. In the year 2022, more than 2.5 million women experienced breast cancer screening, and tragically, approximately 6.6% of them yielded to the disease. Breast cancer originates from abnormal cell proliferation within the breast tissue, often leading to the formation of a breast tumor and the potential spread of cancer to other parts of the body (2). Cancerous tumors are referred to as malignant because they hinder normal body functions and push out healthy tissue (3).

In contrast, benign tumors are noncancerous, since they do not spread to other parts of the body or have the ability to develop more tumors (4). Numerous imaging modalities have been created to help lower the death rate associated with breast cancer and to assist in the early detection and treatment of the disease (5). Breast exams, mammograms, and biopsy are just a few of the numerous examinations used in the detection and diagnosis of breast cancer (6). The more popular method for detecting breast cancer is mammography (7). An effective diagnostic methodology is essential for the timely diagnosis of such malignancy to increase survival (8).

Breast cancer imaging is crucial in lowering this unacceptably high mortality rate. Early detection of breast cancer allows for quicker treatment and higher survival rates than late-stage detection, which is why screening programs have been established (9). One of the significant reasons of death for women globally is breast cancer. Early detection and treatment are the best strategies to stop this cancer from spreading (10). Breast cancer imaging methods are also crucial for assessing and monitoring cancer treatment (11). The best, most dependable, and most cost-effective way for finding early indications of breast cancer is still mammography screening. To see anomalies, radiologists must carefully review mammography images (12). Consequently, the Medical Committee advises women to undergo widespread early mammography screening (13). Women aged 40 and above should undergo an annual mammogram (14).

Recently, computer-aided diagnosis (CAD) systems have performed a vital role in medical imaging, especially for breast cancer diagnosis, to minimize the operator-dependent workload of radiologists when interpreting digital mammography images (15). The purpose of a CAD system is to correctly classify the malignant and benign images, as 65%–90% of images belong to benign cancer. The challenges that increase the false positive rate are masses, architectural distortions, microcalcifications, and asymmetry (16). The diagnosis of microcalcifications has also been clinically authorized for effective CAD systems. Therefore, CAD solutions for breast masses generate a lot of scientific interest (12). Radiologists can identify and distinguish between normal and diseased tissues with CAD systems (17).

Deep learning technology has recently gained widespread adoption in the medical sector. This adoption is driven by the significant patient load and the pressing requirement to enhance the accuracy of pathology diagnosis, particularly in the context of detecting and classifying breast lesions. This technology supports physicians' diagnostic efforts (18, 19). With convolutional neural networks (CNNs), the current deep learning algorithms have shown excellent performance in detecting and segmenting tumors in medical images (20). Deep-learning-based algorithms have demonstrated satisfactory performance in various computer vision applications, such as image classification, medical diagnosis, scene identification, disease prediction, and healthcare analysis (21). A few CAD systems collected straight features from images without doing segmentation or preprocessing work. This phase has the benefit of quick computing, but it has the drawback of extracting redundant and unnecessary information from the image's noisy areas (22). The most significant information is found in the deeper layer, which is immediately computed into features in deep learning (23). However, the training dataset size, the choice of hyperparameters, and the cross-validation value all affect how well deep learning models perform. A deep learning model may have convolutional, ReLu, max pooling, and fully connected hidden layers. In deep learning, the softmax layer functions as a classifier. Several methods for diagnosing and classifying breast cancer have been presented in the literature and have increased accuracy rates. However, they employed the transfer learning idea and concentrated on the pre-trained models (i.e., VGG16 (24), Alexnet (25), ResNet (24), MobileNet (26), and EfficienctNet (27)). For training purposes, those models need a large and well-balanced dataset of images; however, the publicly accessible breast cancer dataset is insufficient (28). In addition, the extraction of irrelevant feature extraction decreased the classification accuracy and increased the computation time (second).

As a result, a unique model that can train on a limited number of images and offer improved classification accuracy is frequently needed. This article proposes a novel deep bottleneck residual convolutional neural network fusion architecture for diagnosing and classifying breast cancer. The suggested technique additionally uses an optimization algorithm to increase accuracy and decrease computing time. The following are the principal contributions of this work:

- We proposed a novel bottleneck residual convolutional neural network (CNN) architecture with three parallel residual blocks and 76 hidden layers, including convolutional, average pooling, and fully connected.
- We proposed a novel single-path bottleneck residual CNN architecture with four residual blocks and 60 hidden layers.
- The hyperparameters of the proposed models are initialized using a Bayesian Optimization instead of manual alteration.
- A kernel-based canonical correlation analysis and entropy technique is proposed for the extracted deep features fusion. The fused feature set is further refined using an optimization approach named generalized normal distribution optimization.
- A detailed comparative analysis has been performed for the proposed method. In addition, a detailed ablation study has been performed.

2 Literature review

Despite researchers having created numerous feature extraction and disease classification strategies, there is still an opportunity for improvement (29). The researchers provided deep-learning methods for breast cancer diagnosis and classification (30). A novel approach for mass classification tasks that simultaneously trains on texture and deep Convolutional Neural Network (CNN) representations was offered by Zhang et al. (16). The CNN-based classification was merged with rotation-invariant features of the maximum response filter bank. The fusion procedure addressed CNN's shortcomings in capturing mass attributes after the reduction technique was implemented. The mini-MIAS and INbreast combined dataset and other publicly available datasets like CBIS-DDSM were used to train this model. The CBIS-DDSM dataset was used to test the reduction strategy and fusion, and the results demonstrated that this method exceeded expectations other models regarding accuracy (94.30%). You-Only-Look-Once (YOLO) was the foundation for the end-to-end system introduced by Baccouche et al. (18). This approach could classify and identify breast abnormalities in complete mammograms that could be of concern at the same time. The algorithm began by preprocessing the raw pictures, identifying the abnormal regions as breast lesions, and then classifying the lesions' pathology as either masses or calcifications. Two publicly available datasets were used to assess the model: one had 2,907 mammograms from the CBIS-DDSM's Curated Breast Imaging Subset, and the other contained 235 mammograms from the INbreast database. The assessment procedure also utilized a privately assembled dataset consisting of 487 mammograms. Furthermore, a fusion model method was suggested to improve detection and classification accuracy.

PatchSample decomposition is a revolutionary technique Harris et al. (31) devised for learning sparse approximations and making classification judgments. In contrast to BlockBoost, the prior method, PatchSample, builds larger dictionaries that encompass a wider variety of visual data from every point inside the region of interest (ROI) and spatially specific information. Notably, a combined dataset of mammograms from two separate providers was utilized to train and examine the approach. The experimental results described that applying PatchSample decomposition to a combined dataset consisting of the MLO view regions of interest from both the MIAS and CBIS-DDSM datasets could result in classification accuracy (ACC) of up to 67.80% and (AUC) of 73.21%. Deep transfer learning techniques were used by TİRYAKİ et al. (32) to classify calcification diseases and breast cancer masses. Convolutional neural networks were trained and tested on a dataset of 3,360 patches taken from the CBIS-DDSM and (DDSM) mammography databases. Resnet50, NASNet, Xception, and EfficientNet-B7 network backbones were used to apply transfer learning. The Xception network produced the best categorization results out of all of them. In particular, an area under the curve (AUC) of 0.9317 was obtained for the five-way classification problem using the original CBIS-DDSM test data. A fresh architecture for a capsule network was proposed by Soulami et al. (33), which significantly reduced the original capsule network's computing time by a factor of 6.5. This improvement made training breast mass areas of interest (ROIs) on less expensive GPUs possible. The proposed architecture was refined by adding data augmentation techniques and changing the number of kernels and capsules. The higher effectiveness of our capsule-based approach in the one-stage classification of suspicious breast masses was demonstrated by evaluation findings across four categories of breast density. The model's accuracy in binary classification—which discerns between normal and abnormal masses—was 96.03%. The model had a 77.78% accuracy rate in the multi-class classification of breast masses into benign, malignant, and normal categories.

A novel multi-stage transfer learning (TL) technique for differentiating between benign and malignant mammographic breast masses was presented by Ayana et al. (34). Images of cancer cell lines and pre-trained models from ImageNet were employed in this technique. The three publicly available datasets used to train the model were DDSM, MIAS, and INbreast. Furthermore, training was conducted using a composite dataset that included photos from all three sources. The average area under the curve for the DDSM, MIAS, INbreast, and mixed datasets was 1, 0.9993, 0.9994, and 0.9998, respectively, according to the fivefold cross-validation results. Using the MIAS and INbreast databases, Aslan et al. (35) sorted mammography images into normal, benign, and malignant categories. After the photos were preprocessed, the processed images were fed into two different end-to-end deep networks. While the second network was intended to have a hybrid structure that combined both the CNN and Bidirectional Long Short-Term Memories (BiLSTM), the first network was composed entirely of a CNN. For the MIAS dataset, the first and second hybrid architectures produced classification accuracy of 97.56% and 97.60%, respectively. A deep learning algorithm-based training technique that enhances edge detail and reduces false positives for automated early detection of breast cancer was presented by Devendhiran et al. (36). The recommended approach combines an optimization strategy with a CNN to produce a classification model for the identification of breast cancer. Utilizing a hybrid approach, the advantages of the Whale Optimization Algorithm (WOA) and the Marine Predators Algorithm (MPA) were merged to determine the optimal hyperparameter values for the CNN framework. The proposed technique leveraged a pre-trained convolutional neural network model called Inception v3 and DenseNet. By comparing the attainment of two hybrid models, the research showed that the MPA-WOA with DenseNet attained an exactness rate of 94% and 95% for the CBIS-DDSM and MIAS datasets, respectively.

This literature review discusses many deep-learning algorithms for breast cancer diagnosis and classification. Researchers have looked into techniques including using You-Only-Look-Once (YOLO) for end-to-end systems, combining texture and deep Convolutional Neural Network (CNN) representations, and introducing PatchSample decomposition for sparse approximations. Several network backbones, including Resnet50, NASNet, Xception, and EfficientNet-B7, have been used in transfer learning approaches to categorize masses of breast cancer and calcification illnesses. A brand-new capsule network design showed decreased processing times and increased efficiency. When distinguishing between benign and malignant mammographic breast masses, multi-stage transfer learning methods that used pre-trained models and composite datasets demonstrated

encouraging results. High classification accuracy was attained by hybrid architectures that combined Bidirectional Long Short-Term Memories (BiLSTM) with CNN. Furthermore, on datasets like CBIS-DDSM and MIAS, a deep learning algorithm that combined optimization algorithms and pre-trained models, including Inception v3 and DenseNet, revealed efficient automated early detection of breast cancer with high accuracy rates. Overall, these findings demonstrate how the field of breast cancer diagnosis is changing through creative methods and point to the possibility of further advancements in precision and efficiency.

3 Proposed methodology

In this section, a proposed methodology for breast cancer classification has been performed using mammogram images. Figure 1 presents the proposed method, which consists of several steps. The INbreast dataset was employed in the first step, and the data augmentation process was performed using the traditional technique to enhance the quantity of dataset and to improve the accuracy of the model. Two novel CNN architectures named the three-block bottleneck residual model and the four-block bottleneck residual model have been proposed. Bayesian Optimization has been employed to initialize hyperparameters throughout the training phase. Features are extracted from the global average pooling layer and fused using a novel technique canonical correlation analysis-based technique. Afterward, a quantum generalized normal distribution optimization algorithm was implemented, and the best features were selected. In the next step, neural network classifiers are utilized for the classification process. Finally, the tumor diagnosis has been performed for the malignant tumor using an explainable AI technique.

3.1 Dataset of the proposed work

INbreast data were gathered at Centro Hospitalar de S. Joao [CHSJ], Breast Centre, Porto, a university hospital in Portugal. The study (cases) covered a total of 115 people. A total of 410 mammograms with CC and MLO images were performed (37). The INbreast dataset includes two categories: benign and malignant. Figure 2 displays sample images from this dataset, containing 410 images belonging to 115 patients. These images come in two different sizes: $2,560 \times 3,328$ pixels and $3,328 \times 4,084$ pixels. In the experimental process, 108 mammogram images of masses were used (38).

3.2 Proposed bottleneck layered model

In a layered model, a "bottleneck" or "bottleneck layer" usually refers to a particular architectural element frequently employed in deep neural networks, especially in CNNs. A bottleneck layer's goal is to lower the feature map's dimensionality while retaining crucial data, which might result in more effective and computationally less expensive models. A particular kind of neural network building block is called a bottleneck block. There are three main parts to each bottleneck block.

1×1 Convolutional layer. This layer looks at a tiny portion of the input data, similar to a small filter. It employs small filters with a 1×1 pixel size because it is called "1 × 1." By reducing the amount of characteristics or channels in the data, this layer helps conserve computational resources.

3×3 Convolutional layer. The 3 × 3 convolutional layer employs larger 3 × 3 filters to detect intricate patterns and features within the data. It functions on the decreased number of channels generated by the preceding 1 × 1 convolutional layer.

1×1 Convolutional layer. The 1 × 1 convolutional layer comes after the 3 × 3 convolution, performing another round of 1×1 convolution. This additional step increases the number of features, thus revitalizing and enriching the data representation.

3.2.1 Three-block bottleneck layered model

In this work, a three-block bottleneck layered model has been proposed for the classification of breast cancer into malignant and benign. Each block follows the arrangement described above.

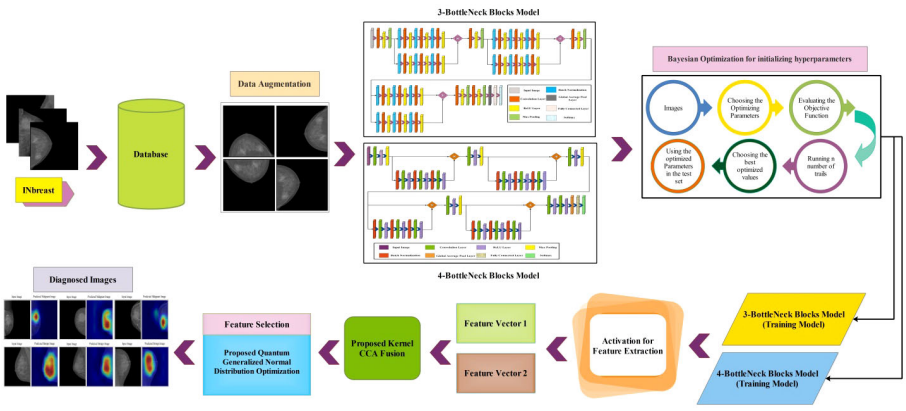

FIGURE 1
Proposed methodology for breast cancer classification and diagnosis.

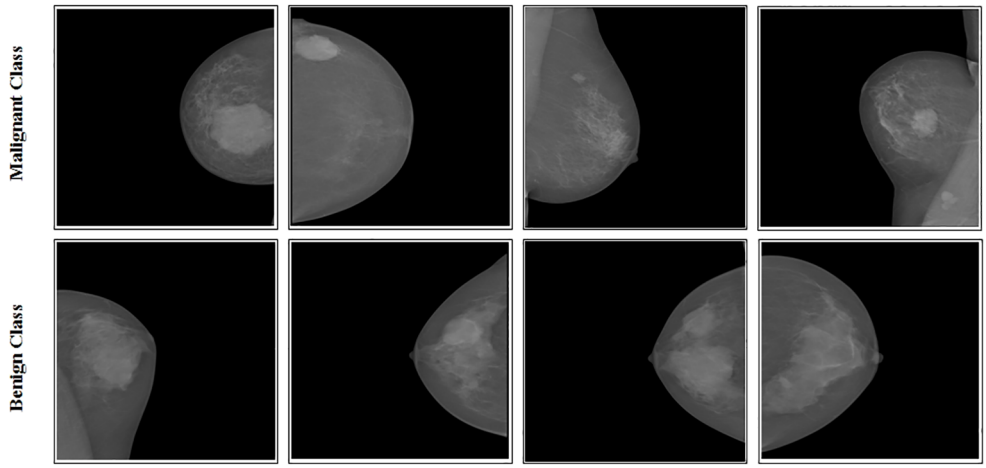

FIGURE 2
Samples images of INbreast dataset.

Figure 3 displays the architecture of the three-block bottleneck layered model. In this network architecture, the initial input dimensions are 227 × 227 × 3, subsequently processed by the first convolutional layer with a depth of 32, a filter size of 3 × 3, a stride of 2, and a ReLU activation layer. Following this, a maxpooling layer with a 3 × 3 filter and a stride of 1 is applied. After that, two bottleneck blocks are added in parallel, each including a batch normalization layer, convolution layer of depth 64, filter size of 1 × 1, stride of 1, and ReLU activation layer. Then, a second batch normalization layer is added to this block, followed by a convolution layer of depth of 64, filter size of 3 × 3, stride of 1, and a ReLU activation layer. After that, a convolution layer is added with depth of 64, filter size of 3 × 3, and stride two, followed by the ReLU activation layer and max pooling layer.

Next, two blocks are appended in parallel, each comprising a batch normalization layer, followed by a convolution layer and a ReLU activation layer. A second batch normalization layer is added, followed by a convolution layer and a ReLU activation layer. Similarly, a third batch normalization layer is introduced, succeeded by a convolution layer and another ReLU activation layer. This sequence is repeated once more with the addition of two more blocks.

Subsequently, a convolution layer with a depth of 1,024, a filter size of 3 × 3, and a stride of 2 is incorporated, followed by a ReLU activation layer. Following this, another convolution layer is introduced with a depth of 2,048, a filter size of 3 × 3, and a stride of 2, followed by a ReLU activation layer. Finally, the network concludes with a global average pool layer, a fully connected layer, and a softmax layer. Figure 4 shows the detailed architecture of the three-block bottleneck layered model. The number of trained parameters of three-block bottleneck residual model is 15.9M.

3.2.2 Four-block bottleneck layered model

The phrase “four bottleneck blocks” refers to how this neural network is built by stacking three of these bottleneck blocks. Each block follows the above-described structure. Figure 5 shows the proposed architecture of four-block bottleneck layered.

In this network architecture, the initial input dimensions are 227 × 227 × 3, subsequently processed by the first convolutional layer with a depth of 32, a filter size of 3 × 3, a stride of 2, and a ReLU activation layer. Following this, a maxpooling layer with a 3 × 3 filter and a stride of 1 is applied. After that, a bottleneck block is added, including a batch normalization layer, convolution layer of depth of 64, filter size of 1 × 1, stride of 1, and ReLU activation layer. Then, a second batch normalization layer was added to this block, followed by a convolution layer of depth of 64, filter size of 3 × 3, stride of 1, and a ReLU activation layer.

Following that, a block is added, which consists of a batch normalization layer, followed by a convolution layer and a ReLU activation layer. Subsequently, a second batch normalization layer is introduced, followed by a convolution layer and a ReLU activation layer. Similarly, a third batch normalization layer is integrated, with a convolution layer and another ReLU activation layer. This sequence is repeated once more with the addition of two more blocks.

Afterward, a convolution layer is introduced with a depth of 1,024, a filter size of 3 × 3, and a stride of 2, followed by a ReLU activation layer. Following this, another convolution layer includes a depth of 2,048, a filter size of 3 × 3, and a stride of 2, followed by a ReLU activation layer. To conclude, the network is finalized by adding a global average pool layer, a fully connected layer, and a softmax layer. Figure 6 shows the detailed architecture of the four-block bottleneck layered model. In this figure, the details of layers and weights are described. The number of trained parameters of the four-block bottleneck residual model is 25.1M.

3.2.3 Training models and features extraction

The proposed model training process has been described in this section. A 50:50 approach has been opted in the training process, meaning that 50% of the images have been employed, and the remaining images have been utilized for the testing method. Several hyperparameters have been selected in the training process by employing the Bayesian Optimization algorithm (39), such as a

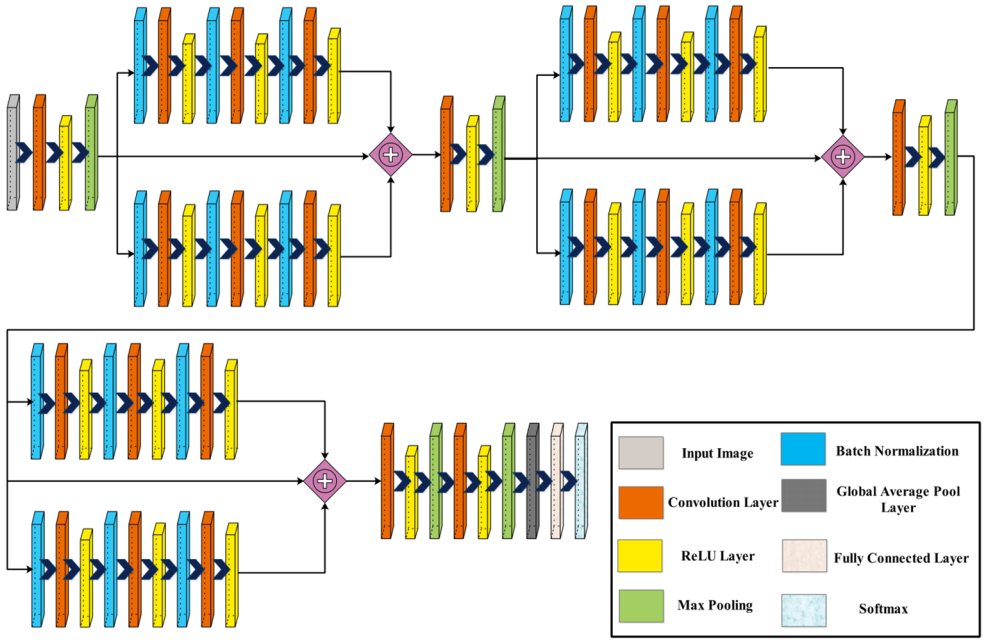

FIGURE 3
Proposed three-block bottleneck layered model architecture.

learning rate value of 0.000241, a momentum value of 0.776, epochs of 50, mini-batch size of 64, and stochastic gradient descent as an optimizer. After that, both models were trained and later utilized for feature extraction. The number of extracted features for both model is 2,048. The testing images have been implemented for the testing feature extraction. The global average pooling layer has been selected for the feature extraction. For both models, 2,048 features are extracted and mathematically presented as follows.

Consider two trained proposed deep learning architectures, such as $\Delta_1 \in$ proposed three-block bottleneck architecture and $\Delta_2 \in$ proposed four-block bottleneck architecture, respectively. The global average pool layer has been selected as a feature layer in both models and defined by L_1 for Δ_1 and L_2 for Δ_2, respectively. Hence, the activation has been performed as follows:

$$\tilde{F}_1(k) = A(\Delta_1, L_1), \qquad \tilde{F}_1(k) \in \mathbb{R}$$

$$\tilde{F}_2(k) = A(\Delta_1, L_2), \qquad \tilde{F}_2(k) \in \mathbb{R}$$

where $\tilde{F}_1(k)$ and $\tilde{F}_2(k)$ are feature matrix of dimensional $N \times 2,048$ and $N \times 2,048$, respectively. In the next stage, an enhanced fusion technique is employed to fuse the extracted features.

3.3 Proposed features fusion

A novel features fusion technique has been proposed in this work and presented under this section for the fusion of $\tilde{F}_1(k)$ and $\tilde{F}_2(k)$. A kernel-based canonical correlation analysis and entropy technique have been implemented for feature fusion. The original CCA technique (40) is a linear algorithm that can potentially reveal the linear correlation between two feature vectors. However, there is a non-linearity problem among them; therefore, we employed a kernel-based CCA technique for the fusion process.

Consider two feature matrix sets $\tilde{F}_1(k)$ and $\tilde{F}_2(k)$, defined as $\tilde{F}_1 = (u_1, u_2, \ldots, u_p)^T$ and $\tilde{F}_2 = (v_1, v_2, \ldots, v_q)^T$. If each variable has n sample points, then the matrix $U_{p\times n} = (u_1, u_2, \ldots, u_n)$ and $V_{q\times n} = (v_1, v_2, \ldots, v_n)$ are created. Suppose ψ maps the original extracted feature vector $U_{p\times n}$ into a high-dimensional feature space F_u that is $\psi : u_i \rightarrow \psi(u_i) \in F_u$; φ map $V_{q\times n}$ into feature space F_v that is $\varphi : v_i \rightarrow \psi(v_i) \in F_v$; therefore, two feature matrices $\psi(U) = [\psi(u_1), \psi(u_2), \ldots, \psi(u_n)]$ and $\psi(V) = [\psi(v_1), \psi(v_2), \ldots, \psi(v_n)]$ are designed.

The aim of kernel CCA is to find two basic vectors α_ψ and b_φ in feature space such that the correlation coefficient between $Q = \alpha_\psi^T \psi(U)$ and $R = b_\varphi^T \varphi(V)$ is maximized, can be formulated by Equations 1, 2.

$$\rho_{Q,R} = \frac{\alpha_\psi^T \psi(U)\varphi(V)^T b_\varphi}{\sqrt{\alpha_\psi^T \psi(U)\psi(U)^T \alpha_\psi}\sqrt{b_\varphi^T \varphi(V)\varphi(V)^T b_\varphi}} \tag{1}$$

$$\begin{cases} \alpha_\psi = \sum_{i=1}^n \xi_i \ \psi(u_i) = \psi(U)\xi \\ b_\varphi = \sum_{i=1}^n \eta_i \ \ \varphi(v_i) = \varphi(V)\eta \end{cases} \tag{2}$$

By putting the value of α_ψ and b_φ into $\rho_{Q,R}$, we computed the following equation (Equation 3):

$$\rho_{Q,R} = \frac{\xi^T \psi(U)^T \psi(U)\varphi(V)^T \varphi(V)\eta}{\sqrt{\xi^T \psi(U)^T \psi(U)\psi(U)^T \psi(U)\xi}\sqrt{\eta^T \varphi(V)^T \varphi(V)\varphi(V)^T \varphi(V)\eta}} \tag{3}$$

By employing the kernel trick, the inner product between $\psi(u_i)$ and $\varphi(u_j)$ can be replaced by a kernel function $K_u(u_i, u_j)$. In this work, we employed the Gaussian kernel function instead of the

Sr.	Name	Activations	Sr.	Name	Activations
1	imageinput 227x227x3 images	227x227x3	39	relu_14 ReLU	114x114x128
2	conv_1 32 3x3x3 convolutions with stride [1 1] and padding 'same'	227x227x32	40	conv_10 256 3x3x256 convolutions with stride [1 1] and padding 'same'	114x114x128
3	relu_1 ReLU	227x227x32	41	relu_10 ReLU	114x114x256
4	maxpool_1 3x3 max pooling with stride [1 1] and padding 'same'	227x227x32	42	batchnorm_9 Batch normalization with 256 channels	114x114x256
5	batchnorm_4 Batch normalization with 32 channels	227x227x32	43	conv_11 128 1x1x256 convolutions with stride [1 1] and padding 'same'	114x114x128
6	conv_6 64 1x1x32 convolutions with stride [1 1] and padding 'same'	227x227x64	44	relu_11 ReLU	114x114x128
7	relu_6 ReLU	227x227x64	45	addition_2 Element-wise addition of 3 inputs	114x114x128
8	batchnorm_5 Batch normalization with 64 channels	227x227x64	46	conv_15 512 3x3x128 convolutions with stride [2 2] and padding 'same'	57x57x512
9	conv_7 64 3x3x64 convolutions with stride [1 1] and padding 'same'	227x227x64	47	relu_15 ReLU	57x57x512
10	relu_7 ReLU	227x227x64	48	maxpool_3 3x3 max pooling with stride [1 1] and padding 'same'	57x57x512
11	batchnorm_6 Batch normalization with 64 channels	227x227x64	49	batchnorm_13 Batch normalization with 512 channels	57x57x512
12	batchnorm_1 Batch normalization with 32 channels	227x227x32	50	conv_16 1024 1x1x512 convolutions with stride [1 1] and padding 'same'	57x57x1024
13	conv_8 32 1x1x64 convolutions with stride [1 1] and padding 'same'	227x227x32	51	relu_16 ReLU	57x57x1024
14	conv_2 64 1x1x32 convolutions with stride [1 1] and padding ' same'	227x227x64	52	batchnorm_14 Batch normalization with 1024 channels	57x57x1024
15	relu_2 ReLU	227x227x64	53	conv_17 2048 3x3x1024 convolutions with stride [1 1] and padding 'same'	57x57x2048
16	batchnorm_2 Batch normalization with 64 channels	227x227x64	54	relu_17 ReLU	57x57x2048
17	conv_3 64 3x3x64 convolutions with stride [1 1] and padding ' same'	227x227x64	55	batchnorm_15 Batch normalization with 2048 channels	57x57x2048
18	relu_3 ReLU	227x227x64	56	conv_18 512 1x1x2048 convolutions with stride [1 1] and padding 'same'	57x57x512
19	batchnorm_3 Batch normalization with 64 channels	227x227x64	57	relu_18 ReLU	57x57x512
20	conv_4 32 1x1x64 convolutions with stride [1 1] and padding ' same'	227x227x32	58	batchnorm_16 Batch normalization with 512 channels	57x57x512
21	relu_4 ReLU	227x227x32	59	conv_19 1024 1x1x512 convolutions with stride [1 1] and padding 'same'	57x57x1024
22	relu_8 ReLU	227x227x32	60	relu_19 ReLU	57x57x1024
23	addition_1 Element-wise addition of 3 inputs	227x227x32	61	batchnorm_17 Batch normalization with 1024 channels	57x57x1024
24	conv_5 128 3x3x32 convolutions with stride [2 2] and padding ' same'	114x114x128	62	conv_20 2048 3x3x1024 convolutions with stride [1 1] and padding 'same'	57x57x2048
25	relu_5 ReLU	114x114x128	63	relu_20 ReLU	57x57x2048
26	maxpool_2 3x3 max pooling with stride [1 1] and padding 'same'	114x114x128	64	batchnorm_18 Batch normalization with 2048 channels	57x57x2048
27	batchnorm_7 Batch normalization with 128 channels	114x114x128	65	conv_21 512 1x1x2048 convolutions with stride [1 1] and padding 'same'	57x57x512
28	conv_9 256 1x1x128 convolutions with stride [1 1] and padding 'same'	114x114x256	66	relu_21 ReLU	57x57x512
29	relu_9 ReLU	114x114x256	67	addition_3 Element-wise addition of 3 inputs	57x57x512
30	batchnorm_8 Batch normalization with 256 channels	114x114x256	68	conv_22 1024 3x3x512 convolutions with stride [2 2] and padding 'same'	29x29x1024
31	batchnorm_10 Batch normalization with 128 channels	114x114x128	69	relu_22 ReLU	29x29x1024
32	conv_12 64 1x1x128 convolutions with stride [1 1] and padding 'same'	114x114x64	70	maxpool_4 3x3 max pooling with stride [1 1] and padding 'same'	29x29x1024
33	relu_12 ReLU	114x114x64	71	conv_23 2048 3x3x1024 convolutions with stride [2 2] and padding 'same'	15x15x2048
34	batchnorm_11 Batch normalization with 128 channels	114x114x64	72	relu_23 ReLU	15x15x2048
35	conv_13 64 3x3x64 convolutions with stride [1 1] and padding 'same'	114x114x64	73	maxpool_5 3x3 max pooling with stride [1 1] and padding 'same'	15x15x2048
36	relu_13 ReLU	114x114x64	74	gapool Global average pooling	1x1x2048
37	batchnorm_12 Batch normalization with 64 channels	114x114x64	75	fc 10 fully connected layer	10
38	conv_14 128 1x1x64 convolutions with stride [1 1] and padding 'same'	114x114x128	76	softmax softmax	10

FIGURE 4
Proposed three-block bottleneck layered detailed architecture.

sigmoid function. Mathematically, this function is defined as follows is defined by Equations 4, 5:

$$K_u(u_i,\ u_j) = \psi(u_i)^T \psi(u_j) \tag{4}$$

$$K_v(v_i,\ v_j) = \varphi(v_i)^T \varphi(v_j) \tag{5}$$

Hence, the two kernel matrices are obtained as follows by Equations 6, 7:

$$(K_u)_{ij} = K_u(u_i,\ u_j) \tag{6}$$

$$(K_v)_{ij} = K_v(v_i,\ v_j) \tag{7}$$

Hence, the correlation between Q, R is defined as follows defined by Equation 8:

$$\rho_{Q,R} = \frac{\xi^T K_u K_v \eta}{\sqrt{\xi^T K_u K_u \xi}\sqrt{\eta^T K_v K_v \eta}} \tag{8}$$

where $\xi^T K_u K_u \xi = 1$ and $\eta^T K_v K_v \eta = 1$. Hence, finally, maximizing the correlation coefficient among ξ and η is defined as follows defined by Equation 9:

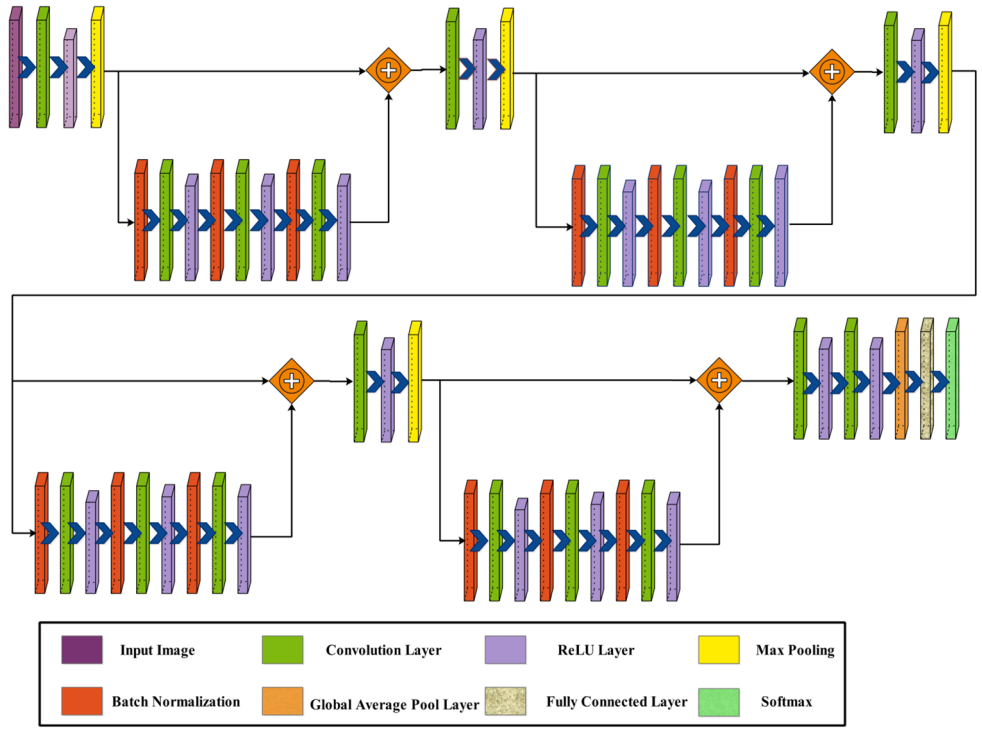

FIGURE 5
Proposed four-block bottleneck layered model architecture.

$$\max_{\xi,\eta} \xi K_u K_v \eta \quad \text{s.t.} \quad \xi^T K_u K_u \xi = \eta^T K_v K_v \eta = 1 \tag{9}$$

Based on the Lagrange multiplier method, the optimization problem is transformed as follows by Equation 10:

$$\begin{bmatrix} & K_u K_v \\ K_v K_v & \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \lambda \begin{bmatrix} K_u K_u & \\ & K_v K_v \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \tag{10}$$

The entropy is employed on the transformed matrix to handle the problem of uncertainty. The returned feature matrix $K(u, v)$ is optimized using the quantum generalized normal distribution algorithm. The vector size of the fused feature is $1,168 \times 4,096$.

3.4 Quantum generalized normal distribution optimization algorithm

A novel optimization algorithm called GNDO has recently been suggested by Zhang et al. (41). In this work, we improved the working of GNDO by employing the quantum mechanism. The GNDO is structured in a pretty simple way, and it intends to share information through both global and local exploration and exploitation. The present ideal location and mean position influence the generalized normal distribution model utilized for local exploitation. On the other hand, global exploration is associated with the selection of three individuals at random.

Local exploitation. Finding better answers inside a search space composed of everyone's current placements is known as local exploitation. An optimal generalized normal distribution model can be created based on the link between the normal distribution and the individual distribution of the population. Mathematically, the optimization method is formulated by Equations 11–27.

$$Z_j^\tau = \partial_j + \complement_j \times \ CT, \ j = 1, 2, 3, \ldots, n \tag{11}$$

where the trailing vector Z_j^τ represents the *jth* individual's trajectory at time τ, while ∂_j denotes their generalized mean position, $\complement_j$ signifies the generalized standard variance, and CT serves as the penalty factor. Furthermore, ∂_j, $\complement_j$, and $\in$ can be characterized as:

$$\partial_j = \frac{1}{3}(V_j^\tau + V_{best}^\tau + K) \tag{12}$$

$$\complement_j = \sqrt{\frac{1}{3}\left[(V_j^\tau - \ \partial)^2 + (V_{best}^\tau - \ \partial)^2 + (K - \partial)^2\right]} \tag{13}$$

$$CT = \begin{cases} \sqrt{-log(\gamma_1)} \ \times \cos(2\pi\gamma_2), & if \ \ x \le y \\ \sqrt{-log(\gamma_1)} \times cos(2\pi\gamma_2 + \pi), & Otherwise \end{cases} \tag{14}$$

where x, y, γ_1, and γ_2 are randomly generated numbers within the range of 0–1, and V_{best}^τ represents the present best position. Additionally, K denotes the current population's mean position, and it can be calculated using the following method:

$$K = \frac{\sum_{j=1}^{n} V_j^\tau}{n} \tag{15}$$

Next, the roles of the three parameters utilized, namely , ∂_j, $\complement_j$, and $\in$ within the designed local exploitation strategy are explained.

Sr.	Name	Activations	Sr.	Name	Activations
1	imageinput 227x227x3 images	227x227x3	31	conv_11 256 1x1x128 convolutions with stride [1 1] and padding 'same'	57x57x256
2	conv_1 32 3x3x3 convolutions with stride [1 1] and padding 'same'	227x227x32	32	relu_11 ReLU	57x57x256
3	relu_1 ReLU	227x227x32	33	batchnorm_7 Batch normalization with 256 channels	57x57x256
4	maxpool_1 3x3 max pooling with stride [1 1] and padding 'same'	227x227x32	34	conv_12 256 3x3x256 convolutions with stride [1 1] and padding 'same'	57x57x256
5	conv_2 64 3x3x64 convolutions with stride [1 1] and padding 'same'	227x227x64	35	relu_12 ReLU	57x57x256
6	relu_2 ReLU	227x227x64	36	batchnorm_8 Batch normalization with 256 channels	57x57x256
7	batchnorm_1 Batch normalization with 64 channels	227x227x64	37	conv_13 128 1x1x256 convolutions with stride [1 1] and padding 'same'	57x57x128
8	conv_3 32 3x3x64 convolutions with stride [1 1] and padding 'same'	227x227x64	38	relu_13 ReLU	57x57x128
9	relu_3 ReLU	227x227x64	39	batchnorm_9 Batch normalization with 128 channels	57x57x128
10	batchnorm_2 Batch normalization with 64 channels	227x227x64	40	addition_3 Element-wise addition of 2 inputs	57x57x128
11	conv_7 32 1x1x64 convolutions with stride [1 1] and padding 'same'	227x227x32	41	conv_6 256 3x3x128 convolutions with stride [2 2] and padding 'same'	29x29x256
12	relu_7 ReLU	227x227x32	42	relu_6 ReLU	29x29x256
13	batchnorm_3 Batch normalization with 32 channels	227x227x32	43	maxpool_4 3x3 max pooling with stride [1 1] and padding 'same'	29x29x256
14	addition_1 Element-wise addition of 2 inputs	227x227x32	44	conv_14 512 1x1x256 convolutions with stride [1 1] and padding 'same'	29x29x512
15	conv_4 64 3x3x32 convolutions with stride [2 2] and padding 'same'	114x114x64	45	relu_14 ReLU	29x29x512
16	relu_4 ReLU	114x114x64	46	batchnorm_10 Batch normalization with 512 channels	29x29x512
17	maxpool_2 3x3 max pooling with stride [1 1] and padding 'same'	114x114x64	47	conv_15 512 3x3x512 convolutions with stride [1 1] and padding 'same'	29x29x512
18	conv_8 128 1x1x64 convolutions with stride [1 1] and padding 'same'	114x114x128	48	relu_15 ReLU	29x29x512
19	relu_8 ReLU	114x114x128	49	batchnorm_11 Batch normalization with 512 channels	29x29x512
20	batchnorm_4 Batch normalization with 128 channels	114x114x128	50	conv_16 256 1x1x512 convolutions with stride [1 1] and padding 'same'	29x29x256
21	conv_9 128 3x3x128 convolutions with stride [1 1] and padding 'same'	114x114x128	51	relu_16 ReLU	29x29x256
22	relu_9 ReLU	114x114x128	52	batchnorm_12 Batch normalization with 256 channels	29x29x256
23	batchnorm_5 Batch normalization with 128 channels	114x114x128	53	addition_4 Element-wise addition of 2 inputs	29x29x256
24	conv_10 64 1x1x128 convolutions with stride [1 1] and padding 'same'	114x114x64	54	conv_17 1024 3x3x256 convolutions with stride [2 2] and padding 'same'	15x15x1024
25	relu_10 ReLU	114x114x64	55	relu_17 ReLU	15x15x1024
26	batchnorm_6 Batch normalization with 64 channels	114x114x64	56	conv_18 2048 3x3x1024 convolutions with stride [2 2] and padding 'same'	8x8x2048
27	addition_2 Element-wise addition of 2 inputs	114x114x64	57	relu_18 ReLU	8x8x2048
28	conv_5 128 3x3x64 convolutions with stride [2 2] and padding 'same'	57x57x128	58	gapool Global average pooling	1x1x2048
29	relu_5 ReLU	57x57x128	59	fc 10 fully connected layer	10
30	maxpool_3 3x3 max pooling with stride [1 1] and padding 'same'	57x57x128	60	softmax softmax	10

FIGURE 6
Proposed four-block bottleneck layered model detailed architecture.

- The generalized mean position, denoted as ∂_j, plays a crucial role. The current best individual is represented as V_{best}^{τ} carries valuable insights about the global optimal solution. Consequently, the *j*th individual, V_j^{τ}, is attracted within the direction of the current best individual, V_{best}^{τ}, increasing its likelihood of discovering an improved solution. It is important to note that if V_{best}^{τ} becomes trapped in a local optimum, all individuals will continue to gravitate toward the direction of V_{best}^{τ}. This behavior can lead to premature convergence of the entire population. To resolve this, the mean position of the current population, denoted as K, is initialized. Individuals can shift toward the direction lying between the best individual, V_{best}^{τ}, and the mean position, K. Furthermore, even if the position of the current best individual, V_{best}^{τ}, remains unchanged for several generations, the mean position K evolves over generations, offering potential benefits in discovering improved solutions. Therefore, including the mean position K in the designed local exploitation strategy can improve the probability of partially avoiding local optima.
- The generalized standard variance, denoted as $\complement_j$, serves the purpose of augmenting the local search capability of the

GNDO approach. Furthermore, based on Equations (12) and (13), generalized standard variances $\complement_j$ can be considered as a random sequence utilized for performing local searches around the generalized mean position ∂_j. Additionally, as per Equation (13), when the distance between the position of the jth individual, V_j^τ, and both the mean position, K, and the position of the best individual, V_{best}^τ, is greater, the variation in the generated random sequence becomes more pronounced. In other words, when the fitness value of an individual V_j^τ is very poor, there is a low likelihood of it discovering a better solution nearby. Consequently, a random sequence with pronounced fluctuation can assist such individuals in searching for improved solutions. Conversely, when an individual, V_j^τ possesses a good fitness value, there's a higher probability of finding a better solution nearby. Therefore, a random sequence with less fluctuation can aid these individuals in achieving better solutions.

- The penalty factor, denoted as $\in$, plays a role in the GNDO algorithm by increasing the level of randomness in the generated generalized standard variance. Most penalty factors typically fall within the range of −1 to 1. It's important to note that the generated generalized standard variances are always positive. Consequently, the penalty factor can expand the search directions of GNDO, thereby augmenting its search capability.

Global exploration. Global exploration involves systematically searching a solution space in order to identify promising regions that may contain valuable solutions. In GNDO, the global exploration is executed by utilizing a trio of randomly chosen individuals, a concept that can be formulated as follows:

$$Z_j^\tau = V_j^\tau + \underbrace{\rho \times(|\gamma_3| \times Z_1)}_{Local\ \ Information\ \ Sharing} + \underbrace{(1-\rho) \times (|\gamma_4| \times Z_2)}_{Global\ \ Information\ \ Sharing} \quad (16)$$

Here, γ_3 and γ_4 are two random numbers following a standard normal distribution, and ρ, referred to as the adjustment parameter, is a random number within the range of 0–1. Additionally, there are two trail vectors denoted as Z_1 and Z_2. Furthermore, the computation of Z_1 and Z_2 can be described as follows:

$$Z_1 = \begin{cases} V_j^\tau - V_{l1}^\tau, & if\ \ f(V_j^\tau) < f(V_{l1}^\tau) \\ V_{l1}^\tau - V_j^\tau, & Otherwise \end{cases} \quad (17)$$

$$Z_2 = \begin{cases} V_{l2}^\tau - V_{l3}^\tau, & if\ \ f(V_{l2}^\tau) < f(V_{l3}^\tau) \\ V_{l3}^\tau - V_{l2}^\tau, & Otherwise \end{cases} \quad (18)$$

where $l1$, $l2$, and $l3$ represent three randomly selected integers ranging from 1 to n, with the condition that $l1 \neq l2 \neq l3 \neq j$. Referring to Equations (7) and (8), the second term on the right-hand side of Equation (6) can be termed as the "local learning term," signifying that solution p1 shares information with solution j. Similarly, the third term on the right-hand side of Equation (16) can be expressed as "global information sharing," describing that individual j receives information from individuals $l2$ and $l3$. The adjustment parameter, ρ, serves the purpose of balancing the two information-sharing strategies. Additionally, γ_3 and γ_4 are random numbers following a standard normal distribution, which extends the search space of GNDO during the global search process. The use of the absolute symbol in Equation (16) is maintained for consistency with the screening mechanism described in Equations (17) and (18).

Novelty-proposed quantum mechanism. Various techniques have been employed in selecting features. The metaheuristics have various shortcomings, including the unbalanced exploration and exploitation that impacts the algorithms' ability to converge. Since optimization algorithms are typically used for feature selection, pattern recognition applications are expected to have excellent accuracy. Applying the quantum theory significantly enhances the typical GNDO optimization algorithm's performance and accuracy. The initial population is generated with size N as:

$$Z_j^k(k+1) = Z_{min} + r \times (Z_{max} - Z_{min}) \quad (19)$$

where Z_i denotes the value of ith solution, $r \in [0,1]$ denotes a random number, Z_{max} and Z_{min} represent maximum and minimum search space limits, and k is a current iteration of a feature space. The fitness is computed in the next step and determine the best solution based on the minimization function. To update the solution of the original GNDO algorithm, the following formulations have been employed.

$$\Phi_{pd} = Z_{i,j}(k+1) \quad (20)$$

$$= \begin{cases} \Omega_i - \beta \times \left(G_{best} - Z_{i,j}(k)\right) \times ln\left(\frac{1}{u}\right), & if\ \ t > 0.5 \\ \Omega_i + \beta \times \left(G_{best} - Z_{i,j}(k)\right) \times ln\left(\frac{1}{u}\right), & if\ \ t \leq 0.5 \end{cases} \quad (21)$$

$$\Omega_i = \theta + P_{best} + (1-\theta) \star g_{best(i)} \quad (22)$$

$$G_{best} = \frac{1}{N}\sum_{i=1}^{N} p\ \ Best_i \quad (23)$$

$$\mathrm{pd}(Z_{i,k+1}^j) = \frac{1}{L_{i,k}^j} \exp\left(-\frac{2\left|z_{i,k+1}^j - \aleph_{i,k}^j\right|}{L_{i,k}^j}\right) \quad (24)$$

The symbol Ω_i denotes the local attractor, P_{best} is best position of ith feature space, and $g_{best(i)}$ is the best feature among the entire feature space. Updating the population is continued until the stop condition is not met. In the end, neural network classifiers are utilized to classify the chosen characteristics such as narrow neural networks and tri-layered neural networks. A feature vector of $N \times 1,572$ has been attained for this work.

A fitness function specified in Equation (25a) processes this procedure.

$$F_{fc} = \left|\frac{f(\varnothing_{best})}{f(\varnothing_i^k)}\right| \quad (28)$$

Every iteration, we take into account the average value of the chosen features, and the cost function is described as:

$$\varnothing_{cost} = \gamma_{\alpha} \times \delta_{error} + \gamma_{\beta} \times \left(\frac{num\_feature}{\max\_feature}\right) \quad (26a)$$

$$\delta_{error} = 1 - \omega_{accuracy} \quad (27)$$

In the aforementioned equation, the coefficients are denoted by γ_{α} and γ_{β}, with values of 0.99 and 0.01 for ω_{α} and ω_{β}, respectively. The cost function is represented by $\varnothing_{cost}$, and the accuracy derived from the fitness function is represented by $\omega_{accuracy}$. Neural network classifiers are ultimately utilized to classify the final set of optimized features.

4 Experimental results and analysis

Experimental setup. Experiments were conducted using the INbreast dataset to classify breast cancer. The dataset was divided into a 50:50 split, with 50% of images from each class allocated for training and the remaining 50% for testing. A strong technique called cross-validation was employed to stop overfitting. A 10-fold cross-validation approach was employed for the testing results. Several hyperparameters have been selected in the training process, such as a momentum value of 0.776, epochs of 50, mini-batch size of 64, and stochastic gradient descent as an optimizer and learning rate value of 0.000241. All experiments were carried out on a computer with a Core i7 processor, 128GB of RAM, and a 12GB graphics card of RTX 3060 utilizing MATLAB2023a.

Classifiers and performance measures. For the classification task, multiple neural network classifiers were employed, including wide neural network (Wi-NN), medium neural network (Me-NN), narrow neural network (Na-NN), bi-layered neural network (Bi-NN), and tri-layered neural network (Ti-NN). The following metrics are used to calculate each classifier's performance: accuracy (Acc), F1 score, precision rate, sensitivity rate, FNR, G-Measure, Kappa, and AUC. Every classifier's time is likewise recorded in the test results.

The best performing classifier was identified by computing performance indicators across a variety of neural networks. There are 10 hidden layers and one fully connected layer in an Na-NN, 25 layers and one fully connected layer in an Me-NN, 100 hidden layers and one fully connected layer in a Wi-NN, 10 layers and two fully connected layers in a Tri-layered NN, and 10 layers and three fully connected layers in a Bi-NN.

Experiments of the proposed framework. The evaluation of the proposed framework has been performed based on the following experiments:

- Experiment 1—classification using a proposed four-block bottleneck network deep features.
- Experiment 2—classification using a proposed three-block bottleneck network deep features.
- Experiment 3—classification using a proposed features fusion approach.
- Experiment 4—classification utilizing the proposed quantum GNDO best feature selection technique.
- Comparison of the proposed framework accuracy and time with several neural nets such as Alexnet, VGG19, Resnet50, and a few more.
- Comparison among proposed quantum GNDO and few other optimization algorithms in terms of accuracy and time.

4.1 Proposed numerical results

Table 1 presents the classification results using a proposed four-block bottleneck network deep features on the INbreast dataset. The top-performing classifier is the Wi-NN, achieving an impressive accuracy of 95.3%. This classifier also exhibits a sensitivity rate of 95.25%, precision rate of 95.3%, MCC of 90.57%, Kappa value of 90.56%, and F1 score of 95.49%, with a corresponding false negative rate (FNR) of 4.75%. The Tri-NN classifier scored the second-highest accuracy, which is 94.9%. Each classifier's computational time is also recorded, and it is found that the medium NN classifier is computationally faster than the other classifiers on the list. The noted time of MNN classifier is 102.92 s, whereas the highest time of this experiment is 343.18 s. Figure 7A shows a confusion matrix of this experiment. In this figure, the correct prediction rate of a malignant class is 94.8%, whereas the FNR value was 5.2%.

Table 2 presents the classification results using a proposed three-block bottleneck network deep features on INbreast dataset. The medium neural network (MNN) classifier is a top-performing classifier with an impressive accuracy of 94.6%. This classifier also shows a sensitivity rate of 94.55%, precision rate of 94.65%, MCC of 89.19%, Kappa value of 89.19%, and F1 score of 94.84%, with a corresponding FNR value of 5.45%. The WNN classifier achieved the second-highest accuracy of 94.1%. Figure 7B shows a confusion matrix of the MNN classifier that can be utilized to confirm the performance of MNN classifier. This figure shows that the correct prediction rate of a malignant class has been 93.9%. For every classifier, the computing time is also mentioned, and the minimum reported time of a bi-layered neural network is 43.705 s. In contrast, the wide-NN classifier has the highest consumed time of 181.96 s. Compared to experiment 1, the computational time of this experiment is slighter; however, the precision rate of the first experiment is better.

Table 3 describes the classification results of the proposed feature fusion. In this experiment, the features of the proposed three- and four-block bottleneck have been fused and the numerical results computed. The top-performing classifier of this experiment is wide neural network, which achieved an impressive accuracy of 94.7%, which a confusion matrix can confirm, as shown in Figure 7C. This classifier also shows a sensitivity rate of 94.7%, precision rate of 94.65%, MCC of 89.37%, Kappa of 89.37%, and F1 score of 94.88%, with a corresponding FNR of 5.3%. Compared with Table 2, the performance of this experiment has been improved; however, the performance of Table 1 (experiment 1) has been better. Moreover, the computational time of the fusion process is better than that of experiments 1 and 2. The fusion process'

TABLE 1 Classification results using a proposed four-block bottleneck network deep features on INbreast dataset.

Classifiers	Acc (%)	Sensitivity Rate (%)	Precision Rate %	F1 Score (%)	FNR (%)	MCC (%)	Kappa (%)	AUC	Time (s)
Na-NN	94.3	94.25	94.25	94.48	5.75	88.51	88.51	0.96	122.47
Me-NN	94.3	94.3	94.35	94.58	5.7	88.68	88.68	0.96	**102.92**
Wi-NN	**95.3**	**95.25**	**95.3**	**95.49**	**4.75**	**90.57**	**90.56**	**0.97**	343.18
Bi-NN	94.6	94.6	94.6	94.83	5.4	89.19	89.19	0.97	123.39
Ti-NN	94.9	94.85	94.85	95.07	5.15	89.71	89.71	0.97	176.51

Bold values denotes the best results.

minimum and highest noted computational time is 31.923 (s) and 41.374 (s), respectively.

Finally, the proposed quantum GNDO-based feature selection technique was applied to the fused feature vector, and the best features were selected. Neural network classifiers are provided with best-selected information in order to improve classification accuracy. Table 4 shows the classification results of the proposed quantum GNDO feature selection technique, which obtained the best accuracy of 96.5% using a bi-layered neural network classifier. This classifier's other listed performance measures are sensitivity rate of 96.45%, precision rate of 96.5%, MCC of 92.97%, Kappa of 92.97%, and F1 score of 96.64%. The Me-NN classifier got the second best accuracy of 96.2%. The confusion matrix of the Wi-NN classifier is shown in Figure 7D, which can be utilized to confirm the overall computed performance measures. This figure shows that the malignant class correct prediction rate is 95.9%, better than the previous three experiments. Every classifier's computation time has been recorded, and the Bi-NN classifier demonstrates a relatively shorter execution time (5.9234 s) compared to the other classifiers. Compared to the performance of the recent three experiments, the feature selection technique improved the computational time. It increased the overall performance of the proposed framework (i.e., accuracy, precision, Kappa, G measure).

4.2 Discussion

2This section explains the suggested framework along with a thorough discussion of it. In the first stage, the accuracy and computation time of the suggested framework's performance were compared to those of many neural nets. Alexnet, VGG19, Resnet50, ResNet101, MobileNet-V2, and Densenet201 are the neural nets that were chosen. The accuracy and time-based comparison are displayed in Figure 8. The accuracy-based comparison of several deep neural networks, including pre-trained and proposed three- and four-block BN networks, is displayed on the left side of this

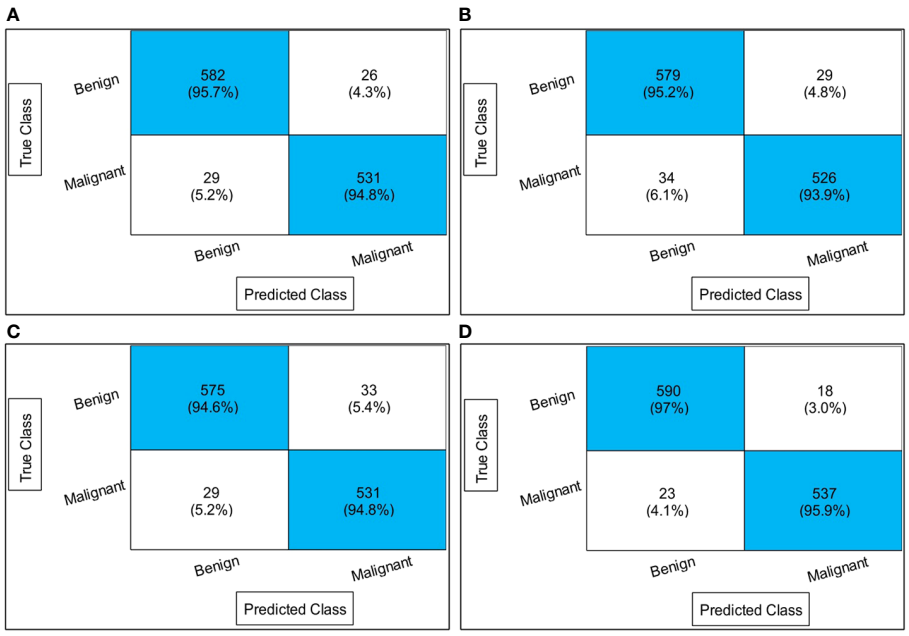

FIGURE 7
Confusion matrices of all experiments: **(A)** confusion matrix of experiment 1, **(B)** confusion matrix of experiment 2, **(C)** confusion matrix of experiment 3, and **(D)** confusion matrix of experiment 4.

TABLE 2 Classification results using a proposed three-block bottleneck network deep features on INbreast dataset.

Classifiers	Acc (%)	Sensitivity Rate (%)	Precision Rate %	F1 Score (%)	FNR (%)	MCC (%)	Kappa (%)	AUC	Time (s)
Na-NN	93.3	93.35	93.3	93.60	6.65	86.62	86.62	0.95	61.901
Me-NN	**94.6**	**94.55**	**94.65**	**94.84**	**5.45**	**89.19**	**89.19**	**0.98**	87.218
Wi-NN	94.1	94.15	94.05	94.29	5.85	88.18	88.17	0.98	181.96
Bi-NN	92.3	92.3	92.3	92.60	7.7	84.56	84.56	0.96	**43.705**
Ti-NN	93.4	93.4	93.45	93.68	6.6	86.79	86.79	0.96	59.943

Bold values denotes the best results.

image. The proposed four-block bottleneck reached the highest accuracy of 95.3% bottleneck CNN design (four-block BN), while 94.6% was obtained by the second proposed architecture, three-block bottleneck (three-block BN). The accuracy of the remaining pre-trained models was 90.2, 90.9, 91.3, 92, 92.6, and 93.1%, in that order. A time-based comparison is also made, and it is found that because of their bigger parameters, the AlexNet and VGG19 models require the longest runs, 344.56 (s) and 326.1 (s), respectively. The suggested three-block BN architecture completed its execution in 87.22 s at the very least, and the four-block BN architecture took 102.92 s.

Different optimization algorithms, including the original GNDO, the proposed quantum GNDO, PSO, Whale Optimization, Jaya Optimization, BCO, Ant Lion Optimization, and Tree Growth Optimization, are compared in Figure 9. Each technique is substituted for the suggested quantum GNDO algorithm in Figure 1, and the outcomes (accuracy and time) are computed. This graphic displays the accuracy plot on the left and the time plot on the right. By analyzing this left-side plot, the Q-GNDO algorithm's accuracy is higher than that of all the other specified methods. Furthermore, the graphic on the right side demonstrates that the computing time of the suggested GNDO algorithm is lower than that of the other feature selection methods.

Student's t-test analysis. A statistical test called the Student's t-test is carried out to contrast the means of two independent groups and analyze if there is a difference between them that is statistically significant. We initially selected two classifiers, Me-NN and Ti-NN, based on the highest and lowest accuracy. Then, we define a hypothesis that h_o = there is no significant difference in the accuracy of the selected classifiers. Table 5 describes the accuracies of selected classifiers. Initially, the difference is computed, as shown in this table, and the mean value is 0.775. After that, a standard deviation is computed, and a value of 0.531 is obtained by employing Equation 28.

TABLE 3 Classification results using the proposed fusion of three-block bottleneck network and four-block bottleneck deep features.

Classifiers	Acc (%)	Sensitivity Rate (%)	Precision Rate %	F1 Score (%)	FNR (%)	MCC (%)	Kappa (%)	AUC	Time (s)
Na-NN	94.3	94.25	94.3	94.52	5.75	88.51	88.50	0.96	34.417
Me-NN	94.3	94.25	94.25	94.49	5.75	88.51	88.51	0.98	**31.923**
Wi-NN	**94.7**	**94.7**	**94.65**	**94.88**	**5.3**	**89.37**	**89.37**	**0.98**	41.374
Bi-NN	93.7	93.65	93.65	93.88	6.35	87.32	87.31	0.95	33.409
Ti-NN	93.1	93.05	93.05	93.32	6.95	86.11	86.11	0.95	40.857

Bold values denotes the best results.

TABLE 4 The final classification results utilizing the proposed quantum GNDO based best feature selection technique.

Classifiers	Acc (%)	Sensitivity Rate (%)	Precision Rate %	F1 Score (%)	FNR (%)	MCC (%)	Kappa (%)	AUC	Time (s)
Na-NN	96.1	96.05	96.1	96.24	3.95	92.11	92.11	0.97	6.4629
Me-NN	96.2	96.2	96.25	96.39	3.8	92.45	92.45	0.97	6.4024
Wi-NN	95.8	95.75	95.85	96.01	4.25	91.61	91.59	0.98	10.089
Bi-NN	**96.5**	**96.45**	**96.5**	**96.64**	**3.55**	**92.97**	**92.97**	**0.98**	**5.9234**
Ti-NN	96.1	96.05	96.2	96.34	3.95	92.30	92.27	0.97	6.1177

Bold values denotes the best results.

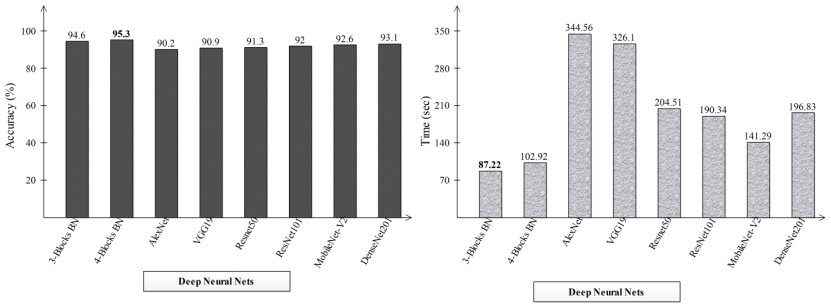

FIGURE 8
Comparison of proposed deep neural nets with pre-trained deep neural networks in terms of accuracy (%) and time (s).

$$Standard\ \ Deviation = \sigma = \sqrt{\frac{\sum_{i=1}^{N}(Diff_i - \mu)^2}{N-1}} \tag{28}$$

By employing the value of mean and standard deviation, the final t value is computed by the following equation by Equation 29:

$$t - selection = t = \frac{\sqrt{N} \times \mu}{\sigma} \tag{29}$$

By this equation, the obtained value of t is 2.919, which is finally employed for the confidence interval analysis. We checked the value of t at p=0.05, and the returned value from the t-table is (−3.182, +3.182), which shows that the value of t falls under this interval. Hence, our hypothesis has been accepted.

In the last step, an indirect comparison of the proposed framework's accuracy with the recently published methods has been conducted, as shown in Table 6. In this table, the authors in (30) obtained an accuracy of 83.19%, later improved by authors in (42) at 95.6%. The other listed methods in this table, such as (43–45), obtained accuracies of 95.1%, 93.0%, and 96.0%, respectively. The proposed method obtained an accuracy of 96.5%, which is better than the recently presented methods. In addition, for the proposed method, the sensitivity rate is 96.45, the precision rate is 96.5, the F1 score value is 96.64, the MCC value is 92.97%, and the Kappa value is 92.97%, respectively. Figure 10 illustrates the visual results of the proposed architecture using the GradCAM technique. The GradCAM is utilized for the localization of cancer regions (malignant) after employing the deeper information of the last convolutional layer. A red-to-blue scale is typically used in Grad-CAM heatmaps, with red denoting the most relevant and blue the least important. The color's intensity indicates the level of significance. Moreover, benign label images are also shown in this image for visualization.

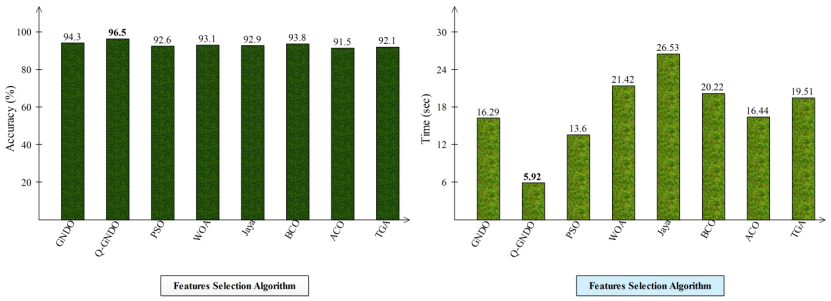

FIGURE 9
Comparison of proposed quantum GNDO features selection technique with several selection algorithms in terms of accuracy (%) and time (s).

TABLE 5 Selected classifiers for the evaluation of Student's t-test.

Classifiers	Four-Bottleneck Block Model	Three-Bottleneck Blocks Model	Fusion	Generalized Normal Distribution Optimization
Medium NN	94.3	94.6	94.3	96.2
Tri-layered NN	94.9	93.4	93.1	96.1
Difference	**0.6**	**1.2**	**1.2**	**0.1**

TABLE 6 Comparison of the proposed architecture accuracy with recent techniques.

References	Year	Dataset	Method	Accuracy (%)
(30)	2023	INbreast	CAD methodology	83.19
(42)	2023	INbreast	Feature Selection and Enhancement network (FSE-Net)	95.6
(43)	2023	INbreast	computer Aided Diagnosis	95.1
(44)	2023	INbreast	ResNet-50 convolutional neural network	93.0
(45)	2022	INbreast	TwoViewDensityNet, an end-to-end deep learning-based method	96.0
Proposed		INbreast		Accuracy=96.5%, Sensitivity Rate= 96.45, Precision rate=96.5, F1 Score=96.64

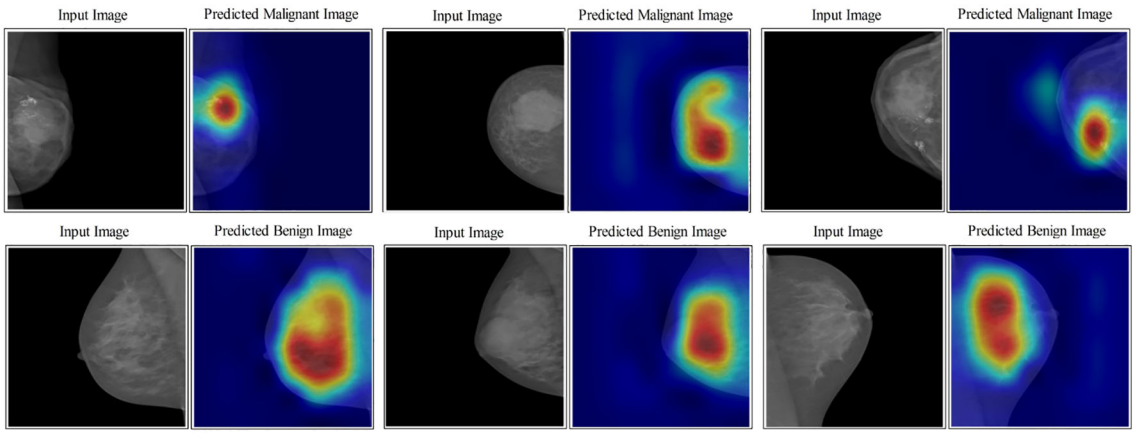

FIGURE 10
Lesion localization results based on proposed four- and three-block bottleneck architectures using GradCAM.

5 Conclusion

In this proposed work, two novel deep learning architectures (four- and three-block bottleneck) have been proposed along with a novel Kernel CCA fusion and quantum GNDO optimization for the improved classification accuracy of malignant and benign breast cancer. The proposed architectures consist of a few parallel blocks and a single bypass layer that improved the learning of a model on the selected mammogram images. Bayesian Optimization is employed to initialize hyperparameters of both architectures and train up to 100 epochs. Features are extracted from the deeper layers and fused using a novel Kernel CCA approach. Only important features that improved the accuracy and precision rate of the proposed framework compared to the individual proposed architectures are fused in this step. In addition, to make the proposed framework more efficient, we proposed a novel quantum GNDO optimization algorithm that selects the best features. The selection process improved the accuracy to 96.5% and considerably reduced the computational time. The proposed framework is compared with state-of-the-art (SOTA) techniques and achieves an enhanced accuracy.

Clinical challenge and future directions. In clinical practice, this proposed architecture can face the following challenges: i) a higher amount of training data is required; ii) high-computational computers are required; and iii) an AI expert is required in the clinic to evaluate the output. There are a few limitations of this work that can be considered as an improvement in the future. The one major limitation of this work is deeper layer feature extraction and fusion of these features instead of fusion within the network. The out-of-the-network fusion process consumed extra time. A self-attention and vision transformer network will be designed to fuse the information for improved accuracy and less computational time.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

KJ: Conceptualization, Data curation, Methodology, Software, Writing – original draft. MK: Conceptualization, Methodology, Software, Supervision, Writing – original draft. MH: Conceptualization, Investigation, Methodology, Software, Writing – original draft. OA: Conceptualization, Formal Analysis, Methodology, Project administration, Resources, Software, Writing – review & editing. MT-H: Formal Analysis, Funding acquisition, Investigation, Methodology, Resources, Validation, Visualization, Writing – review & editing. AM: Funding acquisition, Methodology, Project administration, Supervision, Visualization, Writing – original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number RGP.2/146/44.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Prinzi F, Insalaco M, Orlando A, Gaglio S, Vitabile S. A yolo-based model for breast cancer detection in mammograms. *Cogn Comput*. (2023), 1–14. doi: 10.1007/s12559-023-10189-6.

2. Masood A, Naseem U, Kim J. Multi-level swin transformer enabled automatic segmentation and classification of breast metastases.

3. El-Tanani M, Al Khatib AO, Al-Najjar BO, Shakya AK, El-Tanani Y, Lee Y-F, et al. Cellular and molecular basis of therapeutic approaches to breast cancer. *Cell signal*. (2023) 101:110492. doi: 10.1016/j.cellsig.2022.110492.

4. Chaurasiya S, Rajak R. Comparative analysis of machine learning algorithms in breast cancer classification. *Wireless Pers Commun*. (2023), 1–10. doi: 10.21203/rs.3.rs-1772158/v1.

5. Tasnim Z, Shamrat FJM, Islam MS, Rahman MT, Aronya BS, Muna JN, et al. Classification of breast cancer cell images using multiple convolution neural network architectures. *Int J Adv Comput Sci Appl*. (2021) 12:1–19. doi: 10.14569/IJACSA.2021.0120934.

6. Sherman ME, Vierkant RA, Winham SJ, Vachon CM, Carter JM, Pacheco-Spann L, et al. Benign breast disease and breast cancer risk in the percutaneous biopsy era. *JAMA Surg*. (2023). doi: 10.1001/jamasurg.2023.6382.

7. Hanis TM, Islam MA, Musa KI. Diagnostic accuracy of machine learning models on mammography in breast cancer classification: a meta-analysis. *Diagnostics*. (2022) 12:1643. doi: 10.3390/diagnostics12071643.

8. Chakravarthy SS, Rajaguru H. Automatic detection and classification of mammograms using improved extreme learning machine with deep learning. *Irbm*. (2022) 43:49–61. doi: 10.1016/j.irbm.2020.12.004.

9. W. H. Organization. *Global breast cancer initiative implementation framework: assessing, strengthening and scaling-up of services for the early detection and management of breast cancer*. New York: World Health Organization (2023).

10. Hossin MM, Shamrat FJM, Bhuiyan MR, Hira RA, Khan T, Molla S. Breast cancer detection: an effective comparison of different machine learning algorithms on the Wisconsin dataset. *Bull Electric Eng Inf*. (2023) 12:2446–56. doi: 10.11591/eei.v12i4.

11. Balkenende L, Teuwen J, Mann RM. Application of deep learning in breast cancer imaging. *Semin Nucl Med*. (2022), 584–96. doi: 10.1053/j.semnuclmed.2022.02.003.

12. Singh H, Sharma V, Singh D. Comparative analysis of proficiencies of various textures and geometric features in breast mass classification using k-nearest neighbor. *Visual Comput Industry Biomed Art*. (2022) 5:1–19. doi: 10.1186/s42492-021-00100-1.

13. Lou Q, Li Y, Qian Y, Lu F, Ma J. Mammogram classification based on a novel convolutional neural network with efficient channel attention. *Comput Biol Med*. (2022) 150:106082. doi: 10.1016/j.compbiomed.2022.106082.

14. Hosni Mahmoud HA, Alharbi AH, Alghamdi NS. Breast cancer detection through feature clustering and deep learning. *Intel Automation Soft Comput*. 31:2022.

15. Luo Y, Lu Z, Liu L, Huang Q. Deep fusion of human-machine knowledge with attention mechanism for breast cancer diagnosis. *Biomed Signal Process Control*. (2023) 84:104784. doi: 10.1016/j.bspc.2023.104784.

16. Zhang Q, Li Y, Zhao G, Man P, Lin Y, Wang M. A novel algorithm for breast mass classification in digital mammography based on feature fusion. *J Healthc Eng*. (2020), 1–11, 2020. doi: 10.1155/2020/8860011.

17. Zahoor S, Shoaib U, Lali IU. Breast cancer mammograms classification using deep neural network and entropy-controlled whale optimization algorithm,". *Diagnostics*. (2022) 12:557. doi: 10.3390/diagnostics12020557.

18. Baccouche A, Garcia-Zapirain B, Olea CC, Elmaghraby AS. Breast lesions detection and classification via YOLO-based fusion models. *Comput Mater Continua*. (2021) 69:31–42. doi: 10.32604/cmc.2021.018461.

19. Sharma S, Kumar S. The Xception model: A potential feature extractor in breast cancer histology images classification. *ICT Express*. (2022) 8:101–8. doi: 10.1016/j.icte.2021.11.010.

20. Zhao Z, Du S, Xu Z, Yin Z, Huang X, Huang X, et al. SwinHR: Hemodynamic-powered hierarchical vision transformer for breast tumor segmentation. *Comput Biol Med*. (2024), 107939. doi: 10.1016/j.compbiomed.2024.107939.

21. Shamrat FJM, Azam S, Karim A, Ahmed K, Bui FM, De Boer F. High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. *Comput Biol Med*. (2023) 155:106646. doi: 10.1016/j.compbiomed.2023.106646.

22. Muduli D, Kumar RR, Pradhan J, Kumar A. An empirical evaluation of extreme learning machine uncertainty quantification for automated breast cancer detection. *Neural Comput Appl*. (2023), 1–16. doi: 10.1007/s00521-023-08992-1.

23. Atrey K, Singh BK, Bodhey NK, Pachori RB. Mammography and ultrasound based dual modality classification of breast cancer using a hybrid deep learning approach. *Biomed Signal Process Control*. (2023) 86:104919. doi: 10.1016/j.bspc.2023.104919.

24. Mascarenhas S, Agarwal M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification, in: *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, . pp. 96–9.

25. Shanthi T, Sabeenian R. Modified Alexnet architecture for classification of diabetic retinopathy images. *Comput Electric Eng*. (2019) 76:56–64. doi: 10.1016/j.compeleceng.2019.03.004.

26. Sinha D, El-Sharkawy M. (2019). Thin mobilenet: An enhanced mobilenet architecture, in: *2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, . pp. 0280–5.

27. Marques G, Agarwal D, de la Torre Díez I. Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. *Appl soft comput*. (2020) 96:106691. doi: 10.1016/j.asoc.2020.106691.

28. Özcan H. BUS-CAD: A computer-aided diagnosis system for breast tumor classification in ultrasound images using grid-search-optimized machine learning algorithms with extended and Boruta-selected features. *Int J Imaging Syst Technol*. (2023).

29. Oyelade ON, Irunokhai EA, Wang H. A twin convolutional neural network with hybrid binary optimizer for multimodal breast cancer digital image classification. *Sci Rep*. (2024) 14:692. doi: 10.1038/s41598-024-51329-8.

30. Chakravarthy SS, Bharanidharan N, Rajaguru H. Deep learning-based metaheuristic weighted K-nearest neighbor algorithm for the severity classification of breast cancer. *IRBM*. (2023) 44:100749. doi: 10.1016/j.irbm.2022.100749.

31. Harris CE, Makrogiannis S. Breast mass characterization using sparse approximations of patch-sampled deep features. *Med Imaging 2023: Computer-Aided Diagn*. (2023), 563–9. doi: 10.1117/12.2653926.

32. TİRYAKİ VM. Deep transfer learning to classify mass and calcification pathologies from screen film mammograms. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*. (2023) 12:57–65. doi: 10.17798/bitlisfen.1190134.

33. Soulami KB, Kaabouch N, Saidi MN. Breast cancer: Classification of suspicious regions in digital mammograms based on capsule network. *Biomed Signal Process Control*. (2022) 76:103696. doi: 10.1016/j.bspc.2022.103696.

34. Ayana G, Park J, Choe S-w. Patchless multi-stage transfer learning for improved mammographic breast mass classification. *Cancers*. (2022) 14:1280. doi: 10.3390/cancers14051280.

35. Aslan MF. A hybrid end-to-end learning approach for breast cancer diagnosis: convolutional recurrent network. *Comput Electric Eng*. (2023) 105:108562. doi: 10.1016/j.compeleceng.2022.108562.

36. Devendhiran A, Saradha S, Gowrishankar Jayaraman PM, Vinodhini A. Breast cancer mammography classification using convolutional neural networks and WOA-MPA optimization. *Tuijin Jishu/Journal Propulsion Technol*. (2023) 44:2859–73. doi: 10.52783/tjjpt.v44.i4.1376.

37. Deb SD, Jha RK. Segmentation of pectoral muscle from mammograms using U-Net having densely connected convolutional layers. *Multimed Tools Appl*. (2023), 1–22. doi: 10.1007/s11042-023-16394-7.

38. Jabeen K, Khan MA, Balili J, Alhaisoni M, Almujally NA, Alrashidi H, et al. BC2NetRF: breast cancer classification from mammogram images using enhanced deep learning features and equilibrium-jaya controlled regula falsi-based features selection. *Diagnostics*. (2023) 13:1238. doi: 10.3390/diagnostics13071238.

39. Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdiscip Reviews: Data Min Knowl Discovery*. (2023) 13:e1484. doi: 10.1002/widm.1484.

40. Correa NM, Li Y-O, Adali T, Calhoun VD. Canonical correlation analysis for feature-based fusion of biomedical imaging modalities and its application to detection of associative networks in schizophrenia. *IEEE J select topics Signal Process*. (2008) 2:998–1007. doi: 10.1109/JSTSP.2008.2008265.

41. Zhang Y, Jin Z, Mirjalili S. Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models. *Energy Convers Manage*. (2020) 224:113301. doi: 10.1016/j.enconman.2020.113301.

42. Liao C, Wen X, Qi S, Liu Y, Cao R. FSE-Net: feature selection and enhancement network for mammogram classification. *Phys Med Biol*. (2023) 68:195001. doi: 10.1088/1361-6560/acf559.

43. Das HS, Das A, Neog A, Mallik S, Bora K, Zhao Z. Breast cancer detection: Shallow convolutional neural network against deep convolutional neural networks based approach. *Front Genet*. (2023) 13:1097207. doi: 10.3389/fgene.2022.1097207.

44. Rahman H, Naik Bukht TF, Ahmad R, Almadhor A, Javed AR. Efficient breast cancer diagnosis from complex mammographic images using deep convolutional neural network. *Comput Intell Neurosci*. 2023:2023.

45. Busaleh M, Hussain M, Aboalsamh HA, Al Sultan SA. TwoViewDensityNet: two-view mammographic breast density classification based on deep convolutional neural network. *Mathematics*. (2022) 10:4610. doi: 10.3390/math10234610.