Computers in Biology and Medicine 183 (2024) 109299

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

# Deep-AutoMO: Deep automated multiobjective neural network for trustworthy lesion malignancy diagnosis in the early stage via digital breast tomosynthesis

Xi Chen [a], Jiahuan Lv [a], Zeyu Wang [a], Genggeng Qin [b], Zhiguo Zhou [c,*]

[a] School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, China
[b] Department of Radiology, Nanfang Hospital, Southern Medical University, Guangzhou, China
[c] The Reliable Intelligence and Medical Innovation Laboratory (RIMI Lab), Department of Biostatistics & Data Science, University of Kansas Medical Center and University of Kansas Cancer Center, Kansas City, 66160, KS, USA

ARTICLE INFO

*Keywords:*
Digital breast tomosynthesis
Clinical diagnostic support
Multiobjective optimization
Evidential reasoning
Neural architecture search

ABSTRACT

Breast cancer is the most prevalent cancer in women, and early diagnosis of malignant lesions is crucial for developing treatment plans. Digital breast tomosynthesis (DBT) has emerged as a valuable tool for early breast cancer detection, as it can identify more lesions and improve the early detection rate. Deep learning has shown great potential in medical image-based cancer diagnosis, including DBT. However, deploying these models in clinical practice may be challenging due to concerns about reliability and robustness. In this study, we developed a novel deep automated multiobjective neural network (Deep-AutoMO) to build a trustworthy model and achieve balance, safety and robustness in a unified way. During the training stage, we introduced a multiobjective immune neural architecture search (MINAS) that simultaneously considers sensitivity and specificity as objective functions, aiming to strike a balance between the two. Each neural network in Deep-AutoMO comprises a combination of a ResNet block, a DenseNet block and a pooling layer. We employ Bayesian optimization to optimize the hyperparameters in the MINAS, enhancing the efficiency of the model training process. In the testing stage, evidential reasoning based on entropy (ERE) approach is proposed to build a safe and robust model. The experimental study on DBT images demonstrated that Deep-AutoMO achieves promising performance with a well-balanced trade-off between sensitivity and specificity, outperforming currently available methods. Moreover, the model's safety is ensured through uncertainty estimation, and its robustness is improved, making it a trustworthy tool for breast cancer diagnosis in clinical settings. We have shared the code on GitHub for other researchers to use. The code can be found at https://github.com/ChaoyangZhang-XJTU/Deep-AutoMO.

## 1. Introduction

Breast cancer is the most common malignancy in women compared to other types of cancer. A statistical study from 2017 to 2019 revealed that approximately 12.9 % of females were diagnosed with breast cancer at some point in their lives. The burden of breast cancer is significant, with an estimated 287,850 new cases of female breast cancer in U.S., accounting for approximately 15.0 % of all new cancer cases. Additionally, it is estimated that there will be 43,250 deaths caused by breast cancer in 2023 [1]. Early detection of breast cancer plays a crucial role in breast cancer treatment, which is one of the most promising ways to reduce the burden of cancer. Detecting malignancy at an early stage enables physicians to develop better treatment plans, leading to improved survival rates and longer survival times for patients [2].

Although digital mammography is the commonly used cost-effective diagnostic method in early-stage breast cancer detection, it has limitations. Mamography cannot detect all lesions because of the overlap between normal tissue and lesions. Moreover, the tissue superposition that leads to the masking effect may affect its sensitivity and specificity [3]. In contrast, the recently developed digital breast tomosynthesis (DBT), which produces pseudo 3D images by rotating an X-ray source in a partial arc around the breast while acquiring projection images, can visualize masses, architectural distortions and margins better and reduce the masking effect of superimposing glandular tissue [4]. Furthermore, DBT can reduce the number of false-positives and decrease the recall rate [5]. As such, it can improve the early-stage detection rate

* Corresponding author.
*E-mail address:* zzhou3@kumc.edu (Z. Zhou).

https://doi.org/10.1016/j.compbiomed.2024.109299
Received 28 March 2024; Received in revised form 28 July 2024; Accepted 16 October 2024
Available online 23 October 2024

because of its better ability to identify more lesions [6]. Due to these advantages, DBT is considered a promising tool for breast cancer screening and early detection. Its ability to improve lesion visualization and reduce false positives can lead to more accurate diagnoses and better treatment outcomes for patients with breast cancer.

Despite the promising detection performance of DBT, its large-scale clinical application raises concern about its potential impact on radiologists' workload. A typical DBT view always consists of an average of 60 sections, with 1-mm tomographic section spacing [7], and the interpretation time may be double that of traditional mammography, which will increase possible errors caused by reader fatigue and also increase costs. Many studies have shown that a clinical diagnostic support system (CDSS) can aid radiologists in detecting lesions more effectively and efficiently, including breast cancer [8].

Currently, artificial intelligence, especially deep learning (DL), has emerged as a powerful method to develop medical image-based CDSS in cancer screening and detection. We observe the high potential of DL-based CDSS in breast cancer data analysis, including DBT, and this technique enables the rapid development of a CDSS that is superior to conventional methods [9]. Although promising results have been obtained in deep learning studies, building a trustworthy model to translate DL-based models into clinical practice may face the following challenges:

Clinical datasets are always imbalanced between benign and malignant cases. Consequently, traditional learning algorithms are often designed to maximize the overall prediction accuracy so that models tend to have high accuracy for the majority class and may obtain poor results for the minority class, leading to an imbalance between sensitivity and specificity [10,11]. [12] artificially designed a dedicated deep learning classifier for DBT (DBT-DCNN) and conducted experiments on two DBT datasets. The DBT-DCNN architecture was developed from scratch and consists of five convolutional layers with 96, 128, 384, 192, and 128 filters, respectively. Dataset H1 from Hospital 1 contains 3166 positive and 1526 negative mass lesions, while dataset H2 from Hospital 2 contains 152 positive and 90 negative mass lesions. The model achieved sensitivity and specificity of 0.96 and 0.76 on dataset H1, and 0.81 and 0.94 on dataset H2, respectively. It can be seen that DBT-DCNN achieved imbalanced sensitivity and specificity when dealing with imbalanced datasets. The imbalance between sensitivity and specificity can lead to missed diagnoses or misdiagnoses in clinical practice, which is dangerous. Doctors may not accept such diagnostic clinical application models.

In addition, existing DL-based models usually only output the prediction results (category information, etc.) and do not consider the reliability of the prediction output. Recent research has shown that neural networks tend to make high confidence predictions even for completely unidentifiable inputs or unrelated inputs [13]. Therefore, it is necessary to know how reliable a prediction result is. If a model has a high uncertainty (low reliability) of the output result of a sample, then experts need to further inspect the sample. In contrast, if the uncertainty is extremely low (high reliability), we can be more confident in the model output. In the field of image processing, the research on the uncertainty of image segmentation is relatively rich, while the research on the uncertainty of image classification has not received much attention. Uncertainty estimation is very important to improve the reliability of classification results [14]. estimated uncertainties using MCDO with different types of networks including VGGNet [15], ResNet [16], and DenseNet [17] on dermoscopic images of 8 different skin lesion types. Authors used normalized predictive entropy as a measure for uncertainty and show that the accuracy can be increased when referring a fraction of uncertain samples to a medical expert [18]. performed uncertainty estimation on the publicly available Camelyon data sets for breast cancer detection on histopathological slides. The authors proposed a new method for uncertainty estimation called "M-heads" which added multiple output heads to the convolutional neural network (CNN) [19]. combined a number of convolutional neural networks with the same structure, but differing in the kernel size of their convolutional layers, each classifier computes the prediction uncertainty through Bayesian deep learning [20] and determined the weight based on the uncertainty in order to maximize performance while providing the uncertainty of each classification decision. However, these methods mainly focused on how to estimate the predictive uncertainty, and ignored the study of the balance how to improve the reliability and robustness of models together. It shows that there are still deficiencies in the study of classification uncertainty.

Third, when there is a distribution difference between test samples and training samples, the performance of the model will decline. These out-of-distribution (OOD) datasets typically include noisy samples with the same distribution as training samples, images from different datasets, and synthetic random noise [21–24]. OOD datasets poses a significant challenge in building a robust model that performs consistently across different datasets. Research indicated that noise in DBT can reduce the detectability of subtle lesions [25], thereby affecting the model performance. As such, building a robust model is another major challenge in this study. In summary, it is necessary to develop a unified model that can simultaneously overcome imbalanced data and unsafe and unrobust model challenges.

Motivated by the above challenges, a new deep automated multi-objective neural network (Deep-AutoMO) is developed. Deep-AutoMO takes into account both sensitivity and specificity as objective functions during the training stage to achieve a balanced model. This simultaneous consideration is achieved through the innovative multi-objective immune neural architecture search (MINAS) method, which maximizes the balance between sensitivity and specificity. Bayesian optimization is introduced to optimize the hyperparameters in MINAS, enhancing the efficiency and effectiveness of the model training process. Then, a Pareto-optimal model set that consists of multiple non-dominated Deep Neural Networks (DNNs) is produced. To build a safe and robust model, a new evidential reasoning based on entropy (ERE) approach is developed in the testing stage. The ERE approach allows for the estimation of uncertainty associated with the model's predictions, providing a measure of the reliability of the output. By analyzing the uncertainty, clinicians can make more informed decisions about further evaluation or diagnosis of the samples. Additionally, the ERE approach contributes to improving the model's robustness, enabling it to maintain consistent performance across different datasets and domain shifts. The experimental study on real patient DBT images demonstrated that Deep-AutoMO can not only obtain promising performance but can also obtain balanced results. Moreover, the estimated uncertainty through ERE can measure the safety of the output correctly, and the model robustness is also improved.

The main contributions of this paper are summarized as follows.

- Developing a new Deep-AutoMO model that achieves balance, safety and robustness together.
- Developing a new MINAS algorithm to achieve model balance.
- Developing an ERE that can estimate uncertainty and improve robustness together.
- Obtaining a promising detection performance, leading to potential clinical translation.

## 2. Related work

### 2.1. Evolutionary neural architecture search

Recently, the neural architecture search (NAS) approach [26] has garnered significant attention due to its ability to automatically design neural network architecture. Multiple methods have been developed for a NAS, among which the evolutionary computation-based neural architecture search (ENAS) strategy has shown excellent performance in solving practical optimization problems [27]. An ENAS needs a reasonable objective to find the best architecture. Single-objective ENAS

algorithms have only one objective, and most of them only pursue the maximum accuracy [28,29]. Almost all existing multiobjective ENAS algorithms use the maximum accuracy and minimum resource consumption as optimization objectives [30,31], while there are no studies on solving the imbalance between sensitivity and specificity. The weighted summation method is the simplest way to solve the multiobjective optimization problem and [32,33] weight multiple objective functions in a single objective function to perform a multiobjective ENAS [34,35]. used multiobjective optimization algorithms such as NSGA-II [36] and MOEA/D [37] for multiobjective ENAS to generate a Pareto-optimal model set.

Compared to the existing ENAS algorithms, our proposed MINAS not only differs in methodology from traditional single-objective ENAS, but also breaks through the limitations of current multiobjective ENAS algorithms that focus solely on accuracy and resource consumption. MINAS is designed to address the balance between sensitivity and specificity in target setting and optimization strategies. The MINAS algorithm treats sensitivity and specificity as multiple optimization objectives simultaneously. This approach allows MINAS to discover neural network architectures that perform well in balancing these two aspects, thereby avoiding biases that can arise from a single optimization objective. This capability makes MINAS particularly effective for tasks such as malignant lesion classification, providing a more efficient solution.

### 2.2. Uncertainty estimation

Uncertainty estimation is indeed crucial in establishing security and reliable models [38]. Currently, there are four types of uncertainty estimation strategies, including the test-time data augmentation (TTA), single deterministic method, Bayesian method and ensemble method. TTA tests enhanced samples of the original test sample and generates multiple probability outputs to quantify uncertainty [39]. [38] used TTA to obtain the entropy of the mean class probability to estimate uncertainty. In a single deterministic method, a DNN can introduce evidence theory to directly estimate uncertainty using evidential deep learning (EDL) [40] or introduce an additional neural network for uncertainty estimation [41]. Bayesian deep learning [20] is the mainstream method of uncertainty estimation; it updates the probability distribution of model parameters through training. However, it requires large computational complexity. Then, Monte Carlo dropout (MCDO) [42] was developed to approximate the posterior distribution for capturing the uncertainty without changing the architecture. Another method is the ensemble method, which assembles multiple deep neural networks, not only improving the accuracy but also obtaining the uncertainty of the prediction results [43].

Compared to the existing uncertainty estimation algorithms, our uncertainty estimation method not only introduces two approaches but also simultaneously considers the uncertainties of multiple models. We employ the MCDO and ERE methods. The MCDO method calculates uncertainties of multiple balanced models individually, while the ERE method fuses probabilities and uncertainties of these models to output the final output probability and uncertainty. The combination of MCDO and ERE enhances the safety and reliability of the model.

### 2.3. Robustness

Improving model robustness is crucial for deploying machine learning models in real-world scenarios with varying data distributions and sources. The current methods to achieve this can be broadly categorized into three categories. The first method is data processing, which includes standardizing the intensity distribution of a single image (such as Z-normalization [44]), matching the normalized intensities of all data in datasets from different sources (such as histogram matching [45]) and deep learning transferring methods [46]. Another method is developing domain-invariant architectures and using advanced training methods. In Ref. [47], a pseudovolumetric convolutional neural network with a deep preprocessor module and self-attention (PreSANet) was developed for the cross-institutional prediction of head and neck squamous cell carcinoma. The third method is domain adaptation, using adversarial learning to make models learn features independent of the source domain [48] or using autoencoders to extract domain-invariant features and use them to train a separate prediction model [49]. The experimental results demonstrated that these methods can alleviate the problem of domain shift. In addition [50], showed that the model ensemble method can also obtain good robustness.

Traditional robustness algorithms typically focus on data processing or model optimization, whereas we emphasize enhancing robustness through model fusion, which may be more suitable for applications requiring highly reliable predictions. We use the ERE method to enhance model robustness by integrating predictions from multiple balanced models along with their corresponding uncertainties. This ensemble method not only combines the strengths of multiple models but also provides more robust and reliable predictive capabilities when faced with diverse data distributions and sources.

## 3. Methods

### 3.1. Overview

The pipeline of Deep-AutoMO is shown in Fig. 1, which consists of training and testing stages. In the training stage, MINAS, which is a multiobjective evolutionary algorithm-based NAS method to generate a Pareto-optimal model set, is developed. We use $G$ to denote the maximal number of generations and $P_g = (A_1, ..., A_N)^T, g = 0, ..., G-1$ to denote a population, where $A_i, i = 1, ..., N$ is a particular DNN. First, $A_i$ in $P_0$ are randomly generated and then gradually become more balanced in terms of sensitivity and specificity in each subsequent generation (including clone, mutation, merge, deletion, and update). We formulate the problem as follows:

$$\text{maximize } F(A) = \left(f_{sen}, f_{spe}\right)^T, \tag{1}$$

where the objective vector $F(A)$ comprises the sensitivity $f_{sen}$ and specificity $f_{spe}$ for the validation data $D_{vld}$, and $A$ is a candidate DNN that is trained by minimizing loss of the training data $D_{trn}$. At the end of the evolution stage, a set of Pareto-optimal DNNs (referred to as the Pareto population) denoted as $P_{pareto} = (A_1, ..., A_N)^T$ is generated. Meanwhile, since there are several hyperparameters that may affect the performance of Deep-AutoMO, a Bayesian optimization algorithm is introduced to optimize the hyperparameters.

In testing stage, we developed an evidential reasoning based on entropy (ERE) method that comprehensively considers the performance of multiple DNNs in $P_{pareto}$. This is necessary because DNNs in $P_{pareto}$ may show different levels of sensitivity and specificity, and it is difficult to pick up a single DNN that performs optimally in all objectives from $P_{pareto}$. First, DNNs are selected from $P_{pareto}$ to form a new population $P_{fin} = (A_1, ..., A_{fin\_N})^T$ according to the Pareto grade. Second, the relative weights of DNNs in $P_{fin}$ are calculated, which are determined by $f^i_{sen}, f^i_{spe}$, $i = 1, ..., fin\_N$ on $D_{vld}$. Finally, these DNNs will be tested with testing data $D_{tst}$, and the output results will be fused by ERE to obtain the final performance of Deep-AutoMO.

### 3.2. Training stage

Due to the desirable properties of the multiobjective immune algorithm, including high distribution, self-adaptation, self-organization and good performance of the multiobjective immune algorithm [51], we propose a new multiobjective immune neural architecture search (MINAS) to generate the Pareto population in the training stage. The

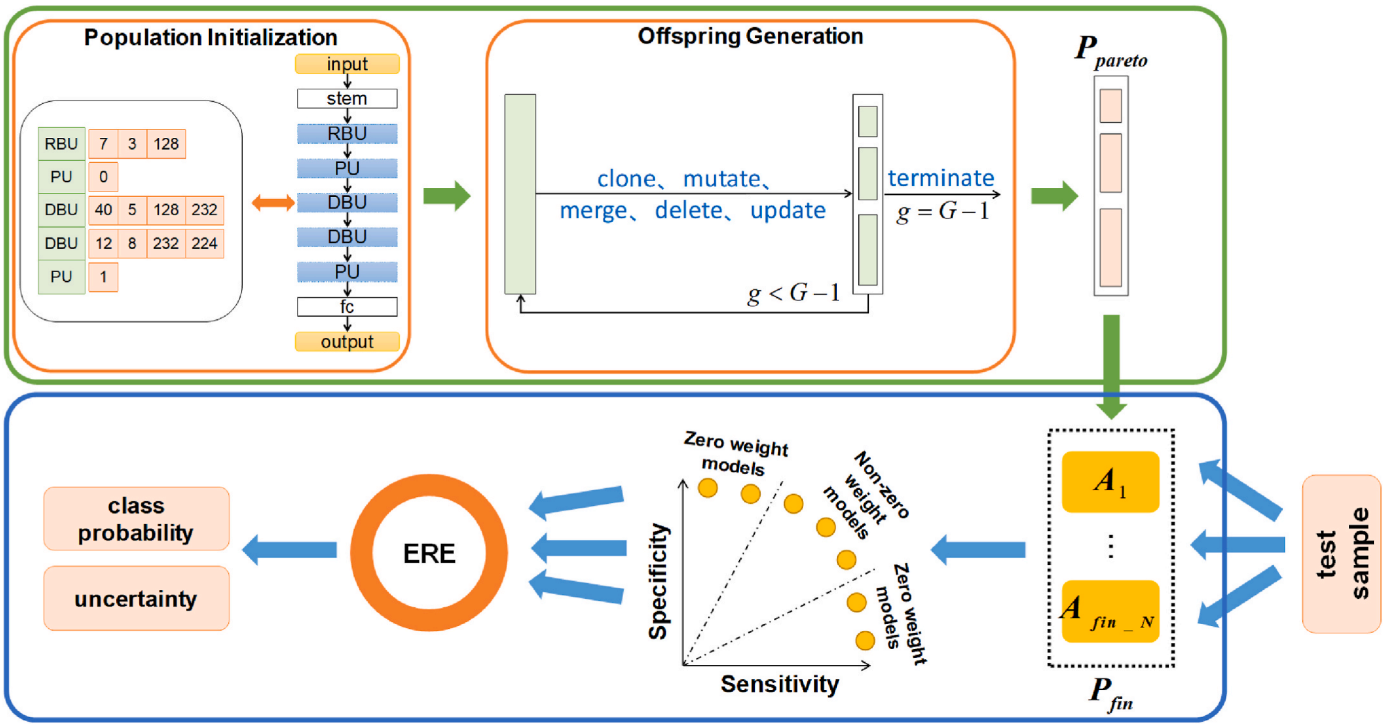



**Fig. 1.** Overview of Deep AutoMO stages. In training stage (green rectangle), each DNN is encoded as group of integer values, estimated through gradient descent, sorted and selected by multiple-objective optimization algorithm, and searched by Bayesian optimization algorithm for hyperparameters. In testing stage (blue rectangle), the optimized DNNs are integrated by ERE. Model performance includes sensitivity, specificity, AUC, and accuracy.

MINAS consists of population initialization, fitness estimation and offspring generation, which will be described as follows.

### 3.2.1. *Encoding and population initialization*

In this study, an efficient variable-length encoding strategy is introduced [27]. Every DNN is constructed using three types of units: ResNet block unit (RBU), DenseNet block unit (DBU) and pooling layer unit (PU), as shown in Fig. 2. We use the MINAS algorithm to determine the DNN unit types and the parameters for each unit. In particular, an RBU contains multiple RBs, whose encoded information is the number of RBs ('*amount*'), the input spatial size ('*in*'), and the output spatial size ('*out*'). In addition to the same encoded information as an RBU, the encoded information of a DBU has an extra parameter '$k$', making the '*out*' of each DB '$k$' larger than its '*in*'. For a DBU with '*amount*' DBs, the '*out*' of the DBU is $k \times amount$ greater than its '*in*'. A PU consists of only a single pooling layer, and only one parameter '*type*' is needed for encoding the pooling type. It should be noted that if the dimension of the input image is $d \times d$, then the maximum number of PUs in each DNN is $\lfloor log_2(d) \rfloor$, reducing the dimension of the input data to $1 \times 1$. A pictorial overview of this encoding strategy and the constraints for different parameters are shown in Fig. 3. Fig. 4 (a) shows an example DNN encoding with 5 units, including one RBU, two PUs and two DBUs. We save encoded information in a list with five one-dimensional arrays, and each array represents a unit. For example, in this DNN, the fourth unit is a DBU, so the fourth array of the list saves the values of four parameters in the DBU.

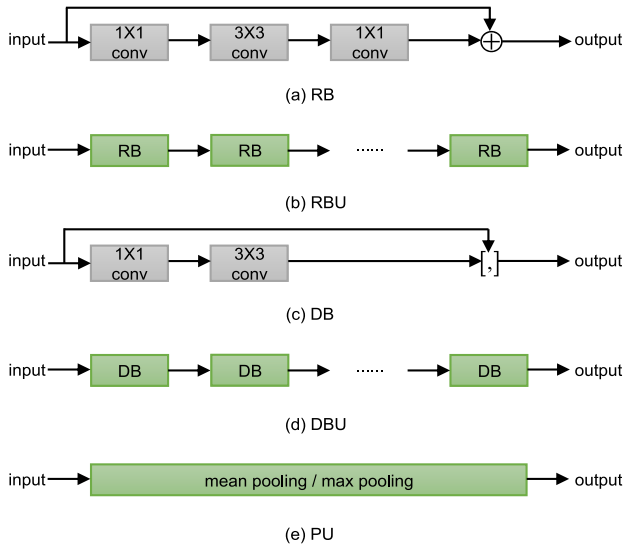



**Fig. 2.** (a) An example of a ResNet block (RB), which is composed of three convolutional layers and one shortcut connection. (b) ResNet block unit (RBU), which is composed of '*amount*' RBs. (c) An example of a DenseNet block (DB), which is composed of two convolutional layers and one concatenation of feature maps. (d) DenseNet block unit (DBU), which is composed of '*amount*' DBs. (e) An example of a pooling layer unit (PU), which consists of only a single pooling layer.

Regarding the initial population $P_0 = (A_1, ..., A_N)^T$ with $N$ DNNs, we encode it as $N$ lists. Each list represents a DNN and is randomly initialized. The list consists of multiple one-dimensional arrays, where each value represents a unit and its length depends on the type of unit. The random initialization of parameters in arrays is constrained as shown in Fig. 3 (b).

### 3.2.2. *Fitness estimation*

Fitness estimation is used to measure the performance of each DNN in the population and consists of two phases. In the first phase, each DNN is decoded based on encoded information. For example, after decoding according to the encoded information in Fig. 4 (a), a DNN with 5 units, a fixed stem and a fixed fc will be built. In the second phase, DNNs are trained with training dataset $D_{trn}$ through the stochastic gradient descent (SGD) algorithm [52]. At the end of the training, sensitivity and specificity can be obtained through the validation dataset $D_{vld}$, which will be considered the fitness value of each DNN. The

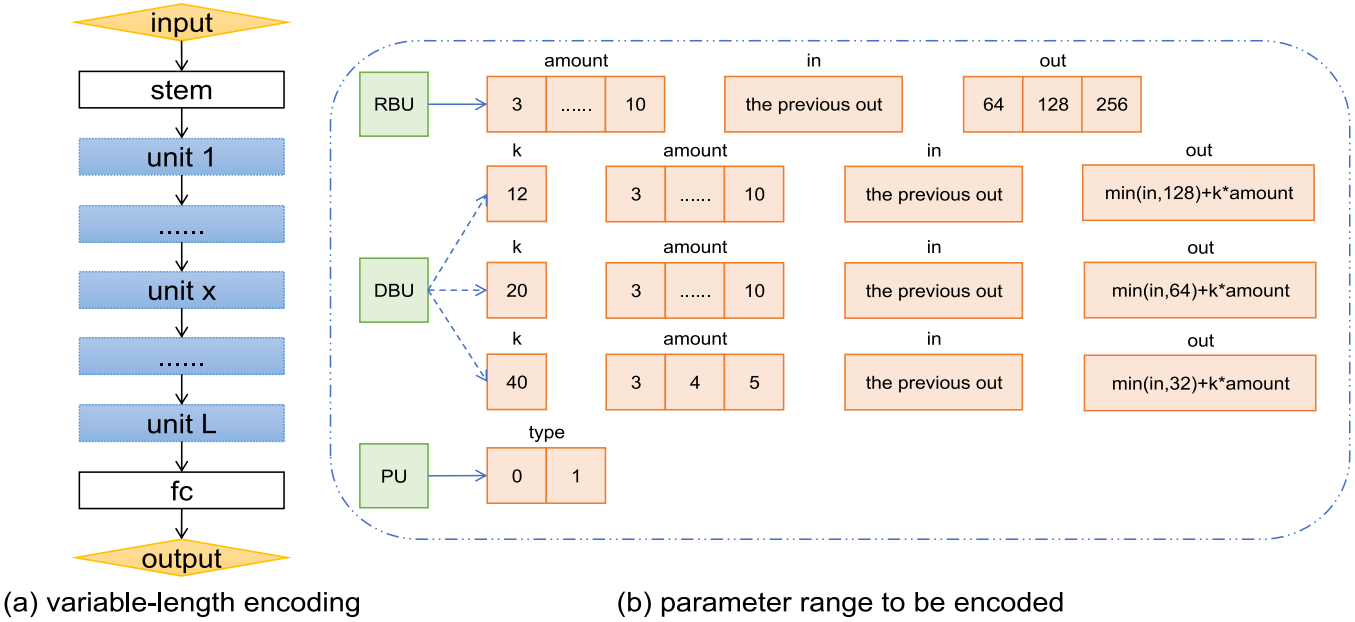



**Fig. 3.** A candidate DNN consists of stem, fc, $L$ searched units and is of variable length. Stem and fc are fixed. Stem is a convolution that transforms the input image into three channels, and fc is a classifier. Each unit can be an RBU, a DBU, or a PU, and a PU cannot be the first unit. Each unit has a specific range of parameters, and the parameter range of a DBU is closely related to '*k*'. The '*in*' of an RBU and a DBU is equal to the '*out*' of the previous RBU or DBU, especially the '*in*' of the first unit, which is fixed as 3. The '*out*' of a DBU is determined by '*k*', '*amount*', and '*in*'. For '*type*' of a PU, 0 indicates mean pooling, and 1 indicates max pooling. The input of each unit is equal to the output of its previous unit.

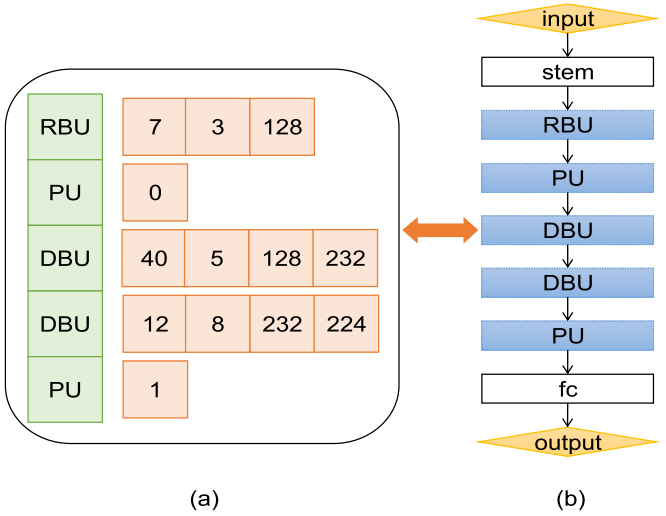



**Fig. 4.** Example of architectural encoding and decoding.

sensitivity and specificity of $A_i$ are calculated as follows:

$$f_{sen}^{i} = \frac{TP}{TP+FN}, f_{spe}^{i} = \frac{TN}{TN+FP}, \tag{2}$$

where $TP$ and $TN$ represent the number of true positives and true negatives, $FP$ and $FN$ are the number of false positives and false negatives, respectively.

### 3.2.3. Offspring generation

**Algorithm 1.** Framework of Training Stage

**Input:** The population size $N$, the maximal generation number G, and the mutation probability $MP$;
**Output:** Pareto population;

(*continued on next column*)

(*continued*)

**Step 1:** g ← 0, P0 ← Initialize a population with a size of N by using the proposed encoding strategy;
**Step 2:** Evaluate the fitness of DNNs in *P0*;
**Step 3:** while g < G do
  while |*Pg*, delete| < N do
    Pg, clone ← Cloned DNNs in Pg with proportional cloning strategy;
    Pg, mutate ← Perform the mutation operation at Pg, clone with *MP*;
    Pg, merge ← Pg ∪ Pg, *mutate*;
    Pg, delete ← Keep one DNN and delete the other duplicate DNNs;
    If |*Pg*, delete| < N then Pg ← Pg, delete;
  Pg, update ← Select N DNNs from Pg, *delete*;
  *Pg*+1 ← Pg, *update*;
  g ← *g*+1;
**Step 4:** PPareto ← Pg.

In our previous work, we introduced the iterative multiobjective immune algorithm (IMIA) for optimization in predicting distant failure in lung cancer treatment [53], which outperformed the traditional immune-inspired multiobjective algorithm. We follow the idea of offspring generation in IMIA. The central idea of offspring generation in IMIA is to create new candidate solutions (offspring) from the existing population iteratively. This process allows the algorithm to explore and refine the population over multiple iterations or generations, gradually improving the performance of the solutions. The offspring consist of clone, mutation, mergence, deletion, and update operators. When the offspring are finished, $P_{pareto}$ is generated. The workflow of offspring generation is shown in Fig. 5 and Algorithm 1.

**Clone.** To maintain the diversity of the population, we clone $A_i$ in $P_g$ $q_i$ times by proportional cloning according to its crowding distance [51] and obtain a clone population with a size of $N_C$ denoted by $P_{g,clone} = (A_1, ..., A_{N_C})^T$. $q_i$ is calculated as follows:

$$q_i = \left\lceil (N_C \times d(A_i)) \Big/ \sum_{j=1}^{N} d(A_j) \right\rceil, i = 1, ..., N, \tag{3}$$

where $\lceil\rceil$ is a ceiling operator and $d(A_i)$ is the crowding distance of $A_i$.

**Mutation.** To discover a better DNN architecture, the mutation

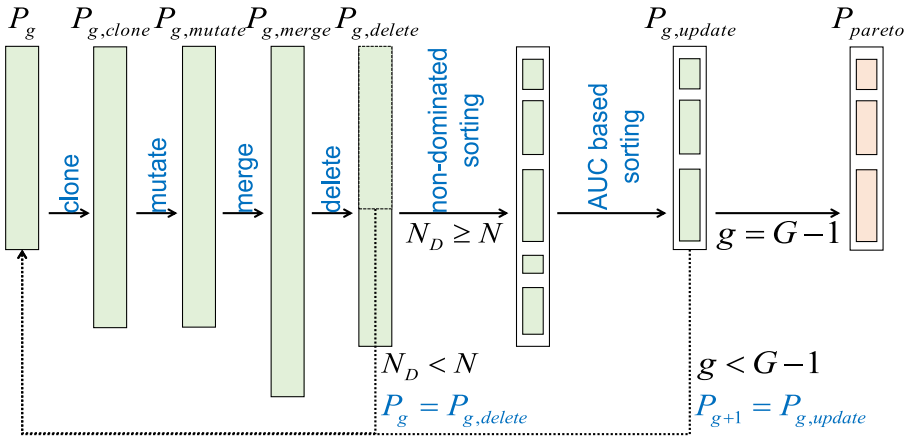



**Fig. 5.** Workflow of offspring generation. The green rectangles represent the population in optimization, and the orange rectangle represents the Pareto population. The blue text at the arrows indicates operations to be performed, and the black text indicates the conditions required to perform the operation.

operation is performed on $P_{g,clone} = (A_1, ..., A_{N_C})^T$ to generate diverse individuals to produce a new population $P_{g,mutate} = (A_1, ..., A_{N_C})^T$, and the predefined mutation probability is represented by *MP*. For each $A_i$, $i = 1, ..., N_C$ in population $P_{g,clone}$, a random mutation probability ($RMP_i$) is generated. If $RMP_i < MP$, a mutation point is randomly selected in $A_i$, and a unit can be added, deleted or modified. If the mutation is addition, an RBU, a DBU, or a PU can be added with equal probability. If the mutation is to modify an existing unit, the modification of the encoding information depends on the unit type. After performing the mutation operation, the encoded information needs to be adjusted as necessary if the constraint conditions are not met. Fig. 6 shows the mutation types and parameter ranges to be modified. Fig. 7 shows an example of "modifying a DBU", Fig. 7 (a) shows the selection of a unit to be modified, Fig. 7 (b) shows the selection of parameters with red circles for modification, and the red numbers in Fig. 7 (c) represent the parameters after mutation and the necessary adjustments to meet the constraints.

**Merge and Deletion.** To preserve elite DNNs of the previous generation, the populations $P_g$ and $P_{g,mutate}$ are merged to form $P_{g,merge}$ after the mutation is completed, leading to the same DNNs. As such, we only keep one and delete the other duplicate DNNs to produce the new population set $P_{g,delete} = (A_1, ..., A_{N_D})^T$. If $N_D < N$, set $P_g = P_{g,delete}$ and return to the cloning step. Otherwise, go to the next step.

**Update.** To keep the population size, *N* DNNs must be selected from the population $P_{g,delete}$. First, we estimate the fitness and the AUC of each individual model $A_i, i = 1, ..., N_D, N_D \geq N$. Then, the fast nondominated sorting method [36] is used to grade $A_i$ according to the fitness, and the $A_i$ in each level are ranked down according to AUC. *N* DNNs with high level and AUC are selected from $P_{g,delete}$ to form a new population $P_{g,update} = (A_1, ..., A_N)^T$.

**Terminate.** If the iteration achieves maximal generation *G*, $P_g$ is considered the Pareto population, that is, $P_{pareto} = P_g$, and the algorithm terminates. Otherwise, set $P_{g+1} = P_{g,update}$ and go to the cloning step.

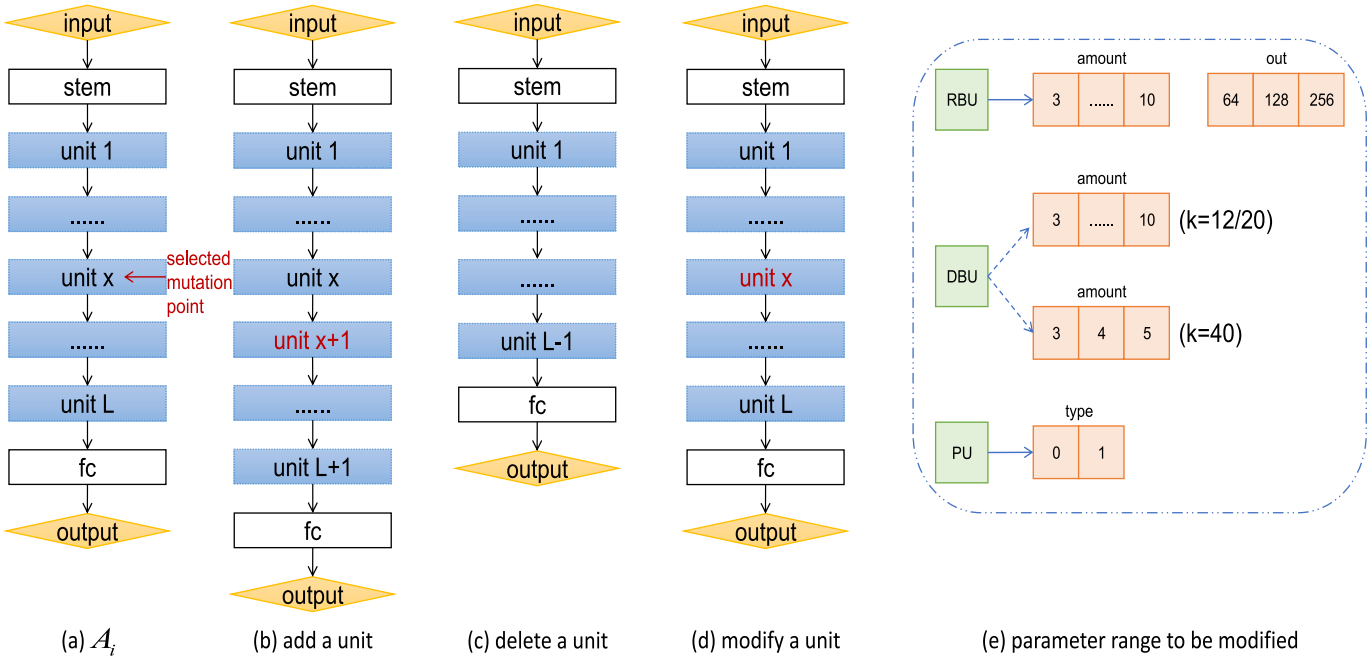
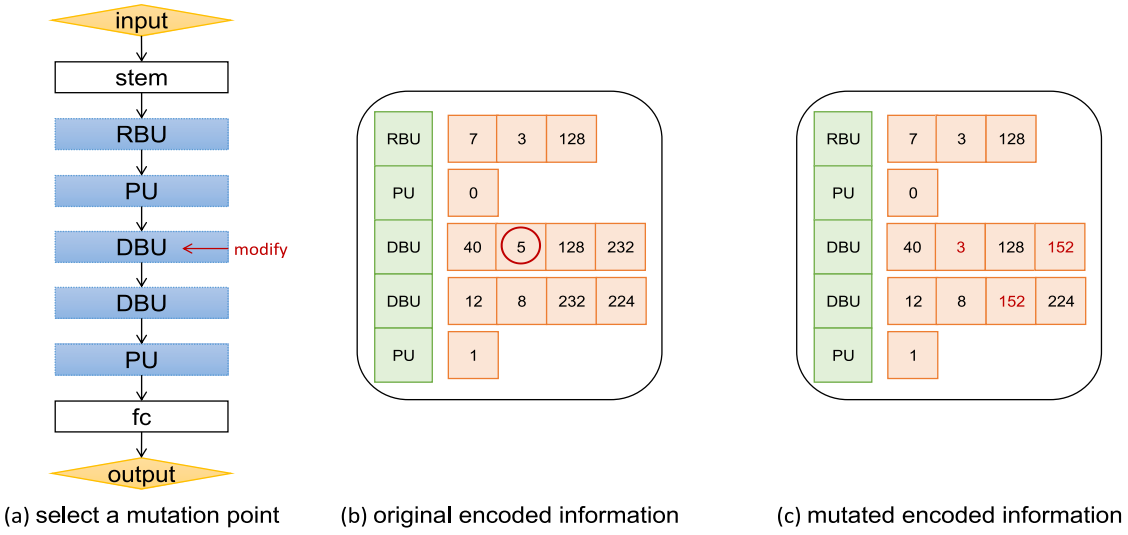



**Fig. 6.** The mutation types and parameter range to be modified (a) Select a unit (e.g., unit *x*). to mutate in a DNN of length *L*. (b) If the mutation type is addition, then add a DBU, an RBU, or a PU after unit *x*, and the length of the DNN becomes *L*+1. (c) If the mutation type is deletion, then remove unit *x*, and the length of the DNN becomes *L*-1. (d) If the mutation type is modification, then the parameters of unit *x* will be modified according to its type, where the modified parameters range in (e).




(a) select a mutation point (b) original encoded information (c) mutated encoded information

**Fig. 7.** An example of "modifying a DBU". The '*amount*' of the selected DBU is mutated from 5 to 3, and its '*out*' will become 152. To ensure that the input of each unit is equal to the output of its previous unit, the '*in*' of the next DBU needs to be adjusted to 152.

### 3.3. Testing stage

In order to address the difficulty of selecting a single Deep Neural Network (DNN) that is optimal in terms of both sensitivity and specificity, we select multiple DNNs from $P_{pareto}$ as an instead. Different DNNs may contribute differently to the predicted results, we then calculate their weights. After generating output probabilities and uncertainty for each DNN, we use the ERE method [50] to make our model safer and more robust. Finally, the prediction probabilities and uncertainty of the final model of the predicted sample are provided. The flow chart of ERE is shown in Fig. 1.

#### 3.3.1. Selecting DNNs

If the number of DNNs fused is too small, ERE may not achieve the optimal effect, and we need a variable *num* to control the lower limit of the number of DNNs. First, all the DNNs at the highest level of $P_{pareto}$ are fed into a new population $P_{fin}$. Suppose $P_{fin}$ contains *fin_N* DNNs. If $fin\_N < num$, continue to put all the DNNs in the next level of $P_{pareto}$ into $P_{fin}$ until $fin\_N \geq num$.

#### 3.3.2. Weight calculation

The weight of $A_i, i = 1, ..., fin\_N$ should be estimated, which is denoted by $w_i, i = 1, ..., fin\_N$. As a balanced DNN in terms of sensitivity and specificity is desired, the ratio between them is considered in the weight calculation. When the ratio is less than 0.5 or greater than 1, the DNN is considered extremely imbalanced, and $w_i$ is set to 0. Meanwhile, the AUC is a good indicator for model performance, and it is also considered. The $w_i$ is calculated as follows:

$$w_i = \begin{cases} \lambda \frac{f^i_{sen}}{f^i_{spe}} + (1-\lambda) AUC_i, when \ 0.5 \leq \frac{f^i_{sen}}{f^i_{spe}} \leq 1 \\ \lambda \frac{f^i_{spe}}{f^i_{sen}} + (1-\lambda) AUC_i, when \ 0.5 \leq \frac{f^i_{spe}}{f^i_{sen}} \leq 1 \\ 0, Other \ \ situations \end{cases}, \tag{4}$$

where $i = 1, 2, ..., fin\_N$, $\lambda$ is a predefined number that indicates the importance of balance, and $1 - \lambda$ indicates the importance of AUC. $f^i_{sen}$ and $f^i_{spe}$ represent the sensitivity and specificity of $A_i$ with validation dataset $D_{vld}$. Then, the weight is normalized, that is:

$$w_i = w_i / \sum_{i=1}^{fin\_N} w_i, i = 1, ..., fin\_N. \tag{5}$$

#### 3.3.3. Forward inference and uncertainty estimation

In this step, we aim to obtain the output probabilities and uncertainty of each DNN for input samples. Due to the extensive application of the dropout mechanism, MCDO is used during the inference phase to capture uncertainty without changing the DNN architecture. In the $K$ classification task, dropout is activation, and it is assumed that the Monte Carlo sampling number is $T$. For a specific $A_i$ of a test sample, the prediction probability of category $k$ of the $t$-th stochastic forward pass is $p^t_{k,i}$, where $k = 1, ..., K$, $t = 1, ..., T$, $i = 1, ..., fin\_N$. We use the mean class probabilities of $T$ predictions as the output probabilities of $A_i$, that is:

$$\bar{p}_{k,i} = \frac{1}{T} \sum_{t=1}^{T} p^t_{k,i}. \tag{6}$$

The uncertainty of $A_i$ is calculated by the prediction entropy, which is:

$$u_i = -\sum_{k=1}^{K} \bar{p}_{k,i} \cdot \log_K \bar{p}_{k,i}. \tag{7}$$

When the mean probabilities for each category are equal, $u_i$ reaches a maximum of 1.

#### 3.3.4. ERE

The input to ERE should contain probabilities $\bar{p}_{k,i}$ for each category and $u_i$, and their sum value should be 1. Whereas $\sum_k^K \bar{p}_{k,i}$ is already 1, and $u_i$ is at most 1, their sum is at most 2. To satisfy the constraints of the ERE strategy, $\bar{p}_{k,i}$ and $u_i$ are normalized:

$$p_{k,i} = \bar{p}_{k,i} / \left( \sum_{k}^{K} \bar{p}_{k,i} + u_i \right), \tag{8}$$

$$u_i = u_i / \left( \sum_{k}^{K} \bar{p}_{k,i} + u_i \right). \tag{9}$$

Then, the final output probability $p_k$ and uncertainty $u$ are obtained, that is:

$$p_k, u = ERE(p_{k,i}, u_i, w_i), \tag{10}$$

where ERE is:

$$p_k = \frac{\mu \times \left[\prod_{i=1}^{fin\_N}\left(w_i p_{k,i} + 1 - w_i \sum_{k=1}^{K} p_{k,i}\right) - \prod_{i=1}^{fin\_N}\left(1 - w_i \sum_{k=1}^{K} p_{k,i}\right)\right]}{1 - \mu \times \left[\prod_{i=1}^{fin\_N}(1 - w_i)\right]}, \quad (11)$$

$$u = \frac{\mu \times \left[\prod_{i=1}^{fin\_N}\left(1 - w_i \sum_{k=1}^{K} p_{k,i}\right) - \prod_{i=1}^{fin\_N}(1 - w_i)\right]}{1 - \mu \times \left[\prod_{i=1}^{fin\_N}(1 - w_i)\right]}. \quad (12)$$

Set $\eta = w_i \sum_{k=1}^{K} p_{k,i}$ and the normalized factor $\mu$ is:

$$\mu = \left[\sum_{k=1}^{K}\prod_{i=1}^{fin\_N}(w_i p_{k,i} + 1 - \eta) - \prod_{i=1}^{fin\_N}(1 - \eta)\right]^{-1}. \quad (13)$$

### 3.4. Bayesian optimization algorithm

Based on the above statements, three hyperparameters *MP*, $\lambda$, and *num* may affect the final output. The Bayesian optimization algorithm [54] can solve hyperparameter optimization problems and consists of two steps. First, the surrogate model is constructed according to evaluation points and their corresponding function values. Second, the next evaluation point is obtained by maximizing the acquisition function based on the surrogate model. After reaching the maximum number of iterations, the optimal evaluation point is obtained. In this paper, each evaluation point includes three hyperparameters, and the objective function is:

$$f_H = \min_{MP,\lambda,num}(1 - AUC), \quad (14)$$

where AUC is one of the performance metrics of Deep-AutoMO for the validation dataset.

## 4. Experiments and analysis

### 4.1. Materials

In this study, the main objective is to diagnose lesion malignancy in DBT. The DBT dataset in this study was collected from Nanfang Hospital (Guangzhou, China) and was approved by the ethics committee (NFEC-2021-191). Each sample has both craniocaudal (CC) and mediolateral oblique (MLO) views and was collected from a Hologic DBT system with an angular range of 15°. Each DBT series is composed of 20–86 slices, with 2457 × 1890 pixels per slice. Eight radiologists with more than three years of experience in breast cancer diagnosis selected one slice of the maximum lesion from each DBT to contour the lesion. We then drew a patch with the size 256x256 to cover the precise lesion. The purpose is to realize information integrity and not to introduce too much noise. Fig. 8 (a) and 8 (b) show a selected slice of the original DBT and the lesion contoured by radiologists, respectively. Fig. 8 (c) and 8 (d) show the corresponding patch we drew.

In this study, 960 patches were cropped from 960 DBT cases to produce a dataset, among which 277 are malignant patches and 683 are benign patches. The dataset was divided into 3:1:1 as training, validation and testing sets, respectively. Since malignant patches are fewer than benign patches, data augmentation including horizontal flip and 90-degree counterclockwise rotation was performed on malignant patches in the training set to obtain more balanced dataset. Finally, there are 495, 56, 56 malignant patches and 407, 138, 138 benign patches in the training set, validation set, and testing set, respectively.

### 4.2. Setup

In the training stage of Deep-AutoMO, the population size $N$ is 15, the clone population size $N_C$ is 30, the maximum number of generations $G$ is 10, the predefined mutation probability *MP* and balance indicator $\lambda$ are in the interval [0.5,1], and the lower limit *num* of the fusion number of DNNs is in the interval [1,4). For each DNN architecture in the population, PyTorch is used to train 100 epochs with training data $D_{trn}$, supervised by an early stopping strategy and with a patience of 5 epochs, to save computational cost. In addition, the batch size is 16, the SGD with an initial learning rate of 1e-3 and a momentum of 0.9 are used, the learning rate is decayed by a factor of 0.9 per 10 epochs, the weight decay is set to 5*1e-4, and the loss function is the cross-entropy.

In the testing stage, $P_{fin}$, which consists of five DNNs, is selected from $P_{pareto}$. To estimate the uncertainty, we conduct 10 stochastic forward passes with dropout activated for each sample to achieve Monte Carlo sampling, and the dropout rate is 0.5. After that, the outputs of these trained DNNs are fused through ERE to provide the performance of Deep-AutoMO. The above process was performed 10 times using the Bayesian optimization algorithm, and the hyperparameters used in Deep-AutoMO were selected by the validation AUC. To demonstrate the performance of the proposed model, we compared Deep-AutoMO with the general-purpose models such as ResNet-18 [16] and DenseNet-121 [17], as well as with other predictive models including DBT-DCNN [12], Shell & kernels [55] and 2D_3D_COR [56], which were specifically designed for DBT. Among them, Shell & kernels model is a joint deep learning model based on AlexNet, which uses a multi-objective optimization algorithm and a classifier fusion strategy to learn and fuse features from shell and kernel to accurately and reliably predict lesion malignancy in digital breast tomography. 2D_3D_COR model [56]

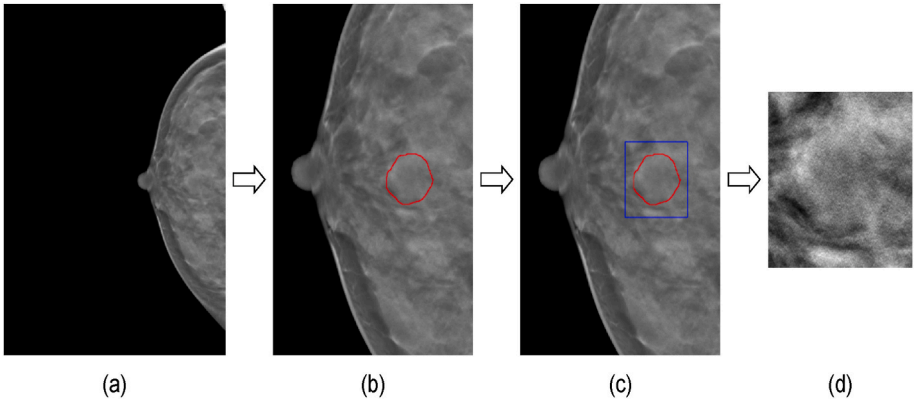



**Fig. 8.** The preprocessing of a DBT example. (a) Selected slice of the original DBT. (b) The lesion contoured by radiologists. (c) The patch of size 256x256 containing the lesion. (d) The cropped patch with size 256x256 as model input.

is a 2D/3D integrated network for the diagnosis of benign and malignant breast tumors, and a correlation strategy was introduced to describe feature correlations between slices in 3D volumes. Tumor classification was performed after fusion of 3D spatial correlation features and 2D image features.

### 4.3. Results and discussion

#### 4.3.1. Ablation study

In this section, we conducted an ablation experiment to investigate the performance of four different deep learning models on the lesion classification task. The four models include ResNet-18, ResNet-18 (multiobjective loss), OptMO-5 and Deep-AutoMO, each of which is trained and tested using the same dataset to ensure comparability of evaluation results. ResNet-18 is a model trained with accuracy as the optimization objective, which is usually used for a variety of image classification tasks through a traditional single objective loss function. ResNet-18 (multiobjective loss) is a model trained with sensitivity and specificity as the optimization objectives. It employs a multiobjective loss function designed to strike a balance between sensitivity and specificity, rather than just pursuing a single accuracy metric. OptMO-5 is one of the optimized models obtained through the MINAS algorithm. MINAS algorithm not only seeks to effectively balance sensitivity and specificity, but also explores the model architecture that is more suitable for the task. Deep-AutoMO is a model that uses ERE method to fuse multiple models optimized by MINAS algorithm. In order to explore the impact of the MINAS algorithm and ERE method on the lesion classification ability of the models, we evaluated the sensitivity, specificity, accuracy, AUC metrics of each model, and compared their balance metric. *Balance* is calculated as follows:

$$Balance = \frac{\min(SEN; SPE)}{\max(SEN; SPE)}. \tag{15}$$

Table 1 summarizes the performance and balance of four models. It can be observed that the balance of ResNet-18 (multiobjective loss) is greater than that of ResNet-18, and OptMO-5 is more balanced than ResNet-18 (multiobjective loss), indicating that multiobjective optimization and MINAS algorithm can effectively improve the balance between sensitivity and specificity. Furthermore, since the MINAS algorithm can automatically search neural network architectures that are more suitable for the task, OptMO-5's overall performance is superior to that of ResNet-18 (multiobjective loss), demonstrating the potential of the MINAS algorithm in enhancing overall model performance. Deep-AutoMO exhibits superior overall performance compared to OptMO-5, proving that the ERE algorithm also has significant advantages in enhancing model performance.

#### 4.3.2. Comparative study

In our study, we conducted an extensive evaluation of different models for breast lesion malignancy detection. We selected five individual Pareto-optimal DNNs labeled as OptMO-i (i = 1, 2, 3, 4, and 5) and fused them through ERE method. Table 2 summarizes the performance of ResNet-18, DenseNet-121, DBT-DCNN, Shell & kernel, 2D_3D_COR, OptMO-i (i = 1, 2, 3, 4, and 5) and Deep-AutoMO in breast lesion malignancy detection and lists the sensitivity (SEN), specificity (SPE), AUC and accuracy (ACC). Among these models, Deep-AutoMO achieved the highest specificity (0.8768), AUC (0.8925) and accuracy (0.8557), except that its sensitivity (0.8036) is slightly lower than that of 2D_3D_COR model as 0.0004. ResNet-18 and DenseNet-121 achieved almost the same performance, while the sensitivities of three DBT-specific models, DBT-DCNN, shell & kernel, and 2D_3D_COR, are better than ResNet-18 and DenseNet-121. However, their specificities are lower than those of ResNet-18 and DenseNet-121. In addition, sensitivity and accuracy of OptMO-i (i = 1, 2, 3, 4, and 5) are all better than those of ResNet18 and DenseNet-121, and only OptMO-4 has a lower AUC, and OptMO-4 and OptMO-5 have lower specificities. The results demonstrated the effectiveness of the MINAS and ERE algorithms, which are used for automated multiobjective neural structure search and model fusion. The combination of these techniques leads to the creation of Deep-AutoMO, a model with superior performance in breast lesion malignancy detection. These findings show the importance of leveraging advanced optimization methods and ensemble strategies to achieve excellent performance.

The number of parameters (Param), the number of floating-point operations (FLOP), and the time of inference latency (Latency) are commonly used to measure the computational complexity of a model. Param directly relates to the model's scale, while FLOP reflects the computational workload required for model inference, and Latency refers to the time needed for a model to complete inference tasks. These metrics are crucial in evaluating the computational complexity and practical application potential of a model. Table 3 provides a comparison of model size and inference speed for different methods. Since the proposed method is based on 2D images, methods based on 3D images are not considered when comparing computational complexity. From Tables 3 and it can be observed that the FLOP of the Deep-AutoMO model is relatively high, but Param and Latency remain within reasonable ranges. Although the model exhibits higher computational complexity, modern computing devices such as GPUs and TPUs enable it to complete inference tasks within reasonable time frames. The inference speed is 10.32 ms which is acceptable in clinical application. Consequently, the Deep-AutoMO model holds high potential for clinical application, particularly in the realm of high-precision medical image diagnosis and analysis.

**Table 1**
The ablation study about four models.

| Models | SEN | SPE | AUC | ACC | Balance |
|---|---|---|---|---|---|
| ResNet-18 [16] | 0.7321 | 0.8551 | 0.8595 | 0.8196 | 0.8562 |
| ResNet-18 (multiobjective loss) | 0.7321 | 0.8116 | 0.8759 | 0.7887 | 0.9020 |
| OptMO-5 | 0.8036 | 0.8333 | 0.8802 | 0.8247 | **0.9644** |
| Deep-AutoMO | **0.8036** | **0.8768** | **0.8925** | **0.8557** | 0.9165 |

**Table 2**
Performances of general-purpose models, other prediction models specifically designed for DBT, OptMO-i (i = 1, 2, 3, 4, and 5) and Deep-AutoMO for the malignant classification of breast lesions. The best results are shown in bold.

| Models | SEN | SPE | AUC | ACC |
|---|---|---|---|---|
| DBT-DCNN [12] | 0.7321 | 0.8043 | 0.8507 | 0.7835 |
| ResNet-18 [16] | 0.7321 | 0.8551 | 0.8595 | 0.8196 |
| DenseNet-121 [17] | 0.7143 | 0.8551 | 0.8636 | 0.8144 |
| Shell & kernel [55] | 0.7590 | 0.8340 | 0.8710 | 0.8120 |
| 2D 3D COR [56] | **0.8040** | 0.8260 | 0.8810 | 0.8200 |
| OptMO-1 | 0.7321 | 0.8623 | 0.8703 | 0.8247 |
| OptMO-2 | 0.7500 | 0.8623 | 0.8776 | 0.8299 |
| OptMO-3 | 0.7679 | 0.8768 | 0.8760 | 0.8454 |
| OptMO-4 | 0.7500 | 0.8478 | 0.8487 | 0.8196 |
| OptMO-5 | 0.8036 | 0.8333 | 0.8802 | 0.8247 |
| **Deep-AutoMO** | 0.8036 | **0.8768** | **0.8925** | **0.8557** |

**Table 3**
Comparison of the model size and inference speed of different methods.

| Models | Param (↓) | FLOP (↓) | Latency (↓) |
|---|---|---|---|
| DBT-DCNN [12] | 4.48 M | 23.84 G | **2.36 ms** |
| ResNet-18 [16] | 11.17 M | 2.27 G | 4.99 ms |
| DenseNet-121 [17] | 6.95 M | 3.64 G | 20.62 ms |
| OptMO-1 | 1.37 M | 2.29 G | 11.72 ms |
| OptMO-2 | **1.01 M** | **2.19 G** | 7.64 ms |
| OptMO-3 | 2.68 M | 4.41 G | 8.84 ms |
| OptMO-4 | 6.57 M | 12.52 G | 14.10 ms |
| OptMO-5 | 3.78 M | 14.78 G | 8.28 ms |
| Deep-AutoMO | 15.41 M | 36.19 G | 10.32 ms |

#### 4.3.3. Balance evaluation

In this study, *Balance* is used to measure the balance of sensitivity and specificity and is in the interval [0,1]. *Balance* is calculated according to Equation (15). If the sensitivity and specificity are equal, *Balance* is equal to 1, indicating that the model is the most balanced. If the difference in sensitivity and specificity is large, such as one approaching 0 and one approaching 1, *Balance* will be close to 0, indicating that the model is extremely imbalanced. The balance metric is used to quantify how well a model performs in terms of both sensitivity and specificity, indicating how balanced the model is in making accurate predictions for both positive and negative cases. Fig. 9 provides a summary of the balance metrics for these eight models. OptMO-5 is the most balanced among the eight models, reaching 0.9644. The average balance of OptMO-i (i = 1, 2, 3, 4, and 5) is 0.8887, which is 3.80 % and 6.39 % higher than that of ResNet-18 and DenseNet-121, respectively. Deep-AutoMO is the most balanced except for OptMO-5, reaching 0.9165, an improvement of 7.04 % and 9.72 % over ResNet-18 and DenseNet-121, respectively. DenseNet-12 is the most unbalanced. Balance is not the only indicator for evaluating a model in this study, and we always choose a best model based on a comprehensive consideration of several metrics. From a clinical application perspective, we are inclined to select the model with the highest AUC while the balance metric exceeds a certain threshold. Although OptMO-5 is the most balanced model, we choose Deep-AutoMO as the best model. These findings demonstrate the effectiveness of the MINAS algorithm in obtaining more balanced models in terms of sensitivity and specificity. Additionally, the ERE algorithm in Deep-AutoMO allows for further improvement in balance, making it a highly balanced model for breast lesion malignancy detection. The combination of these algorithms has proven beneficial in achieving models that provide accurate and well-balanced predictions for the task at hand.

#### 4.3.4. Safety evaluation

Fig. 10 illustrates the relationship between model performance and uncertainty. The test samples are sorted in increasing order of uncertainty, and different proportions of test samples with high uncertainty are removed to calculate the accuracy. When no samples are removed, the sensitivity, specificity, AUC and accuracy of Deep-AutoMO are 0.8036, 0.8768, 0.8925 and 0.8557, respectively. After removing 1/4, 2/4 and 3/4 samples with high uncertainty, the sensitivity becomes 0.8333, 0.8966 and 0.9286, the specificity becomes 0.9327, 0.9706 and 1.0000, the AUC becomes 0.9272, 0.9468 and 0.9571, and the accuracy becomes 0.9041, 0.9485, and 0.9796, respectively. By removing 1/4 of the test samples with high uncertainty at a time, the performance of Deep-AutoMO is gradually improved. When the removal ratio is 3/4, the sensitivity, specificity, AUC and accuracy are increased by 15.56 %, 14.05 %, 7.24 % and 14.48 %, respectively, compared with those at the beginning, indicating that the predicted results of samples with high uncertainty are more likely to be incorrect than those with low uncertainty. Deep-AutoMO can measure the degree of confidence through uncertainty estimation. By gradually removing samples with high uncertainty, the model's performance gradually increases, indicating that when the uncertainty of a test sample is high, the prediction result of this sample is more likely to be incorrect. Our proposed model not only can obtain the uncertainty values of each test sample but also can improve the reliability and accuracy of itself by removing the samples with high uncertainty, which can aid in physicians make more reliable and accurate decisions. Deep-AutoMO is not the first method that can reasonably measure the degree of confidence according to the uncertainty [14]. estimated uncertainties using MCDO with different types of networks on dermoscopic images of 8 different skin lesion types. Authors used normalized predictive entropy as a measure for uncertainty and showed that the accuracy can be increased when referring a fraction of uncertain samples to a medical expert. We used the MCDO method to compare with the proposed method, and its performance is shown in Fig. 10. It can be seen that although the performance of both models can be improved after removing a quarter of the high-uncertainty samples each time, the four metrics of Deep-AutoMO are all higher than those of MCDO, indicating that our method has more advantages in the classification of DBT lesions.

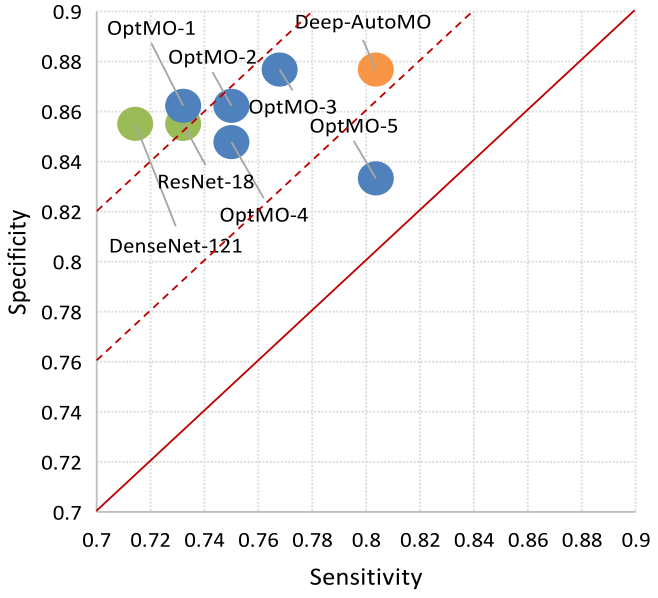



**Fig. 9.** Sensitivity and specificity of Deep-AutoMO (orange dot), OptMO-i (i = 1, 2, 3, 4, and 5) (blue dots), ResNet-18 and DenseNet-12 (green dots). The red solid and dotted lines are auxiliary lines. The model will be located on the real line if its sensitivity and specificity are equal. The dotted lines make it easy to compare the distance between different models and the real line.

#### 4.3.5. Robustness evaluation

In this study, Gaussian noise is added to the original test samples as OOD samples. In particular, the images are first represented as two-dimensional matrices of pixel values, and then random values with a normal distribution are generated and added to the pixel values of the images, generating a new image with Gaussian noise added. Table 4 shows the performance of ResNet-18, DenseNet-121 and Deep-AutoMO for the OOD samples. Compared with the performance for test samples without Gaussian noise, the sensitivity is increased by 27 %, 27 % and 9 %, the specificity is decreased by 47 %, 63 % and 30 %, the AUC is decreased by 9 %, 9 % and 5 %, and the accuracy is decreased by 28 %, 40 % and 19 %, respectively. Deep-AutoMO demonstrated the least decline in performance when exposed to OOD samples, indicating its robustness against noise and distribution differences. On the other hand, ResNet-18 was the most affected by the OOD samples, while DenseNet-121 also showed significant vulnerability to noise. The increased sensitivity of the three models is due to the great similarity between Gaussian noise and lesions, which makes models misclassify more negative samples as positive, also resulting in reduced specificity, AUC, and accuracy. In addition, except for sensitivity, the other three metrics on Deep-AutoMO are the highest, and ResNet-18 and DenseNet-121 both have specificities below 0.5 and accuracies below 0.6. This emphasizes that Deep-AutoMO is significantly more robust compared to the other models in handling OOD samples, despite robustness not being explicitly part of the architectural selection process. The study highlights the importance of model robustness and its potential implications in real-world scenarios where unseen data may vary in distribution and contain noisy or unexpected patterns. Deep-AutoMO's superior performance in the presence of OOD samples reinforces the effectiveness of the MINAS and ERE algorithms in generating a more robust and reliable model for breast lesion malignancy detection.

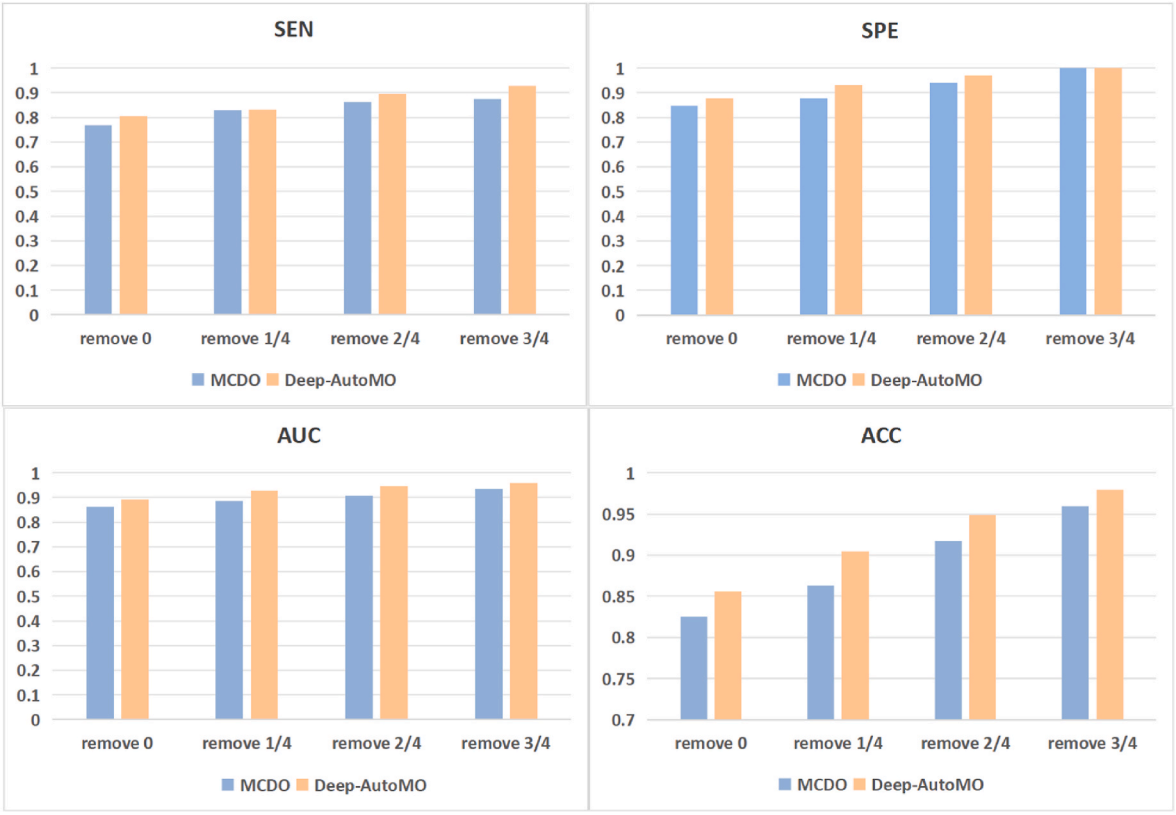



**Fig. 10.** The performance of MCDO and Deep-AutoMO models improves with the removal of highly uncertain samples.

**Table 4**
Performance of ResNet-18, DenseNet-121 and Deep-AutoMO with the OOD test samples. The change compared with that of the performance with the original dataset is in parentheses.

| Models | SEN | SPE | AUC | ACC |
|---|---|---|---|---|
| ResNet-18 [16] | **0.9286** (+27 %) | 0.4493 (−47 %) | 0.7820 (−9%) | 0.5876 (−28 %) |
| DenseNet-121 [17] | 0.9107 (+27 %) | 0.3188 (−63 %) | 0.7879 (−9%) | 0.4897 (−40 %) |
| **Deep-AutoMO** | 0.8750 **(+9 %)** | **0.6087 (-30 %)** | **0.8508 (-5%)** | **0.6856 (-19 %)** |

#### 4.3.6. *Designed architectures of Pareto-optimal DNNs*

Table 5 presents the architectures of Pareto-optimal DNNs found on DBT by the proposed algorithm. As seen from Table 5, the '*amount*' of DBUs is mostly 3, '*k*' is mostly 12, and the '*out*' of RBUs is mostly 64 or 128. It shows that the DNN can yield better performance when it contains units with these parameters, which provides prior knowledge for us to improve the MINAS algorithm in the future.

## 5. Conclusion

The proposed Deep-AutoMO model represents an advancement in breast lesion malignancy prediction using DBT data. In training stage, MINAS algorithm was introduced, sensitivity and specificity were taken as optimization objectives, and a set of balanced models are obtained. In testing stage, MCDO was used for uncertainty estimation and ERE was used to fuse the balanced models to obtain safer and more robust results. The experimental results demonstrate that Deep-AutoMO can obtain promising predictive performance. Moreover, compared with the general-purpose models and other predictive models which were specifically designed for DBT study, it is shown that Deep-AutoMO has achieved the best comprehensive performance. The effectiveness of MINAS algorithm and ERE algorithm in improving the performance and balance of the model was illustrated. In addition, after gradually removing the high-uncertainty test samples, the performance of Deep-AutoMO was gradually improved, which indicates that the prediction results of high-uncertainty samples are more prone to errors, and Deep-AutoMO improved the model safety through uncertainty estimation. When Gaussian noise was added to the original test samples, the results show that Deep-AutoMO is least affected by noise and has stronger robustness to out-of-distribution samples.

Although Deep-AutoMO achieved promising performance, safety, and robustness, demonstrating its potential to enhance breast cancer diagnosis and treatment planning in clinical practice, there are still some limitations in practice. In the field of deep learning, the design and optimization of neural networks has always been an important research area. With the development of differentiable architecture search [57] and dynamic multi-objective optimization [58] algorithms, we expect to find more flexible and powerful neural network architectures, which will further improve the performance and generalization ability of the model. More importantly, the proposed method should not be limited to improving the model performance, but can be applied to compressing existing deep learning models, which will be investigated in future. By combining architecture search and model compression techniques, we can significantly reduce model size and computational complexity while maintaining model performance, achieving the goal of efficiently deploying deep learning models in resource-constrained environments.

In addition, the performances of predictive models depend on extensive training data, but the high cost of annotating large-scale medical image datasets due to regulations such as medical privacy protection and medical research ethics limits the availability of abundant training data. Data distribution difference is also a major factor to affect the model performance. When a model trained on dataset from one domain (such as a source institution) encounters a sample from another domain, its performance always significantly declines. To solve

**Table 5**
Architecture information of OptMO-i (i = 1, 2, 3, 4, 5).

| Models | Id | Type | Configuration |
|---|---|---|---|
| OptMO-1 | 1 | RBU | amount = 3, in = 3, out = 128 |
| | 2 | PU | max pooling |
| | 3 | PU | max pooling |
| | 4 | RBU | amount = 8, in = 128, out = 64 |
| | 5 | PU | mean pooling |
| | 6 | DBU | k = 12, amount = 3, in = 64, out = 100 |
| | 7 | RBU | amount = 8, in = 100, out = 64 |
| | 8 | DBU | k = 20, amount = 3, in = 64, out = 124 |
| | 9 | PU | mean pooling |
| OptMO-2 | 1 | RBU | amount = 3, in = 3, out = 128 |
| | 2 | PU | max pooling |
| | 3 | PU | max pooling |
| | 4 | PU | max pooling |
| | 5 | DBU | k = 12, amount = 3, in = 128, out = 164 |
| | 6 | RBU | amount = 8, in = 164, out = 64 |
| | 7 | DBU | k = 20, amount = 3, in = 64, out = 124 |
| | 8 | PU | mean pooling |
| OptMO-3 | 1 | RBU | amount = 7, in = 3, out = 64 |
| | 2 | RBU | amount = 5, in = 64, out = 64 |
| | 3 | PU | mean pooling |
| | 4 | RBU | amount = 3, in = 64, out = 256 |
| | 5 | PU | max pooling |
| OptMO-4 | 1 | DBU | k = 20, amount = 6, in = 3, out = 123 |
| | 2 | RBU | amount = 10, in = 123, out = 128 |
| | 3 | PU | mean pooling |
| | 4 | DBU | k = 12, amount = 4, in = 128, out = 176 |
| | 5 | RBU | amount = 6, in = 176, out = 256 |
| | 6 | PU | mean pooling |
| | 7 | DBU | k = 40, amount = 3, in = 256, out = 152 |
| OptMO-5 | 1 | RBU | amount = 5, in = 3, out = 256 |
| | 2 | PU | max pooling |
| | 3 | DBU | k = 12, amount = 5, in = 256, out = 188 |
| | 4 | DBU | k = 12, amount = 4, in = 188, out = 176 |
| | 5 | DBU | k = 12, amount = 3, in = 176, out = 164 |

these two challenges, domain adaptation is a possible solution. By adjusting the model trained in source domain to adapt to the data distribution in target domain, the method can improve the model performance in target domain and enhance the model generalization ability. However, concerns about the privacy of source domain data have become a significant obstacle to the widespread application of domain adaptation in medical prediction. Considering the high privacy of medical information, the medical field typically does not provide complete source domain data. Therefore, building a reliable domain adaptation model that can preserve the patient privacy is becoming an urgent and complex challenge.

## CRediT authorship contribution statement

**Xi Chen:** Methodology. **Jiahuan Lv:** Methodology. **Zeyu Wang:** Methodology. **Genggeng Qin:** Data curation. **Zhiguo Zhou:** Methodology.

## Declaration of Competing Interest

None Declared.

## Acknowledgment

This work was supported in part by the National Key Research and Development Program of China (2022YFA1204203) and the Key Science and Technology Program of Shaanxi Province, China (2024GX-YBXM-036).

## References

[1] M. Bethesda, Cancer stat facts: female breast cancer. https://seer.cancer.gov/statfacts/html/breast.html, 2023.
[2] A.D. Singhi, L.D. Wood, Early detection of pancreatic cancer using dna-based molecular approaches, Nat. Rev. Gastroenterol. Hepatol. 18 (7) (2021) 457–468, https://doi.org/10.1038/s41575-021-00470-0.
[3] A. Sakai, Y. Onishi, M. Matsui, H. Adachi, A. Teramoto, K. Saito, H. Fujita, A method for the automated classification of benign and malignant masses on digital breast tomosynthesis images using machine learning and radiomic features, Radiol. Phys. Technol. 13 (2020) 27–36.
[4] K. Mendel, H. Li, D. Sheth, M. Giger, Transfer learning from convolutional neural networks for computer-aided diagnosis: a comparison of digital breast tomosynthesis and full-field digital mammography, Acad. Radiol. 26 (6) (2019) 735–743.
[5] E.L. Honig, L.A. Mullen, T. Amir, M.D. Alvin, M.K. Jones, E.B. Ambinder, E. T. Falomo, S.C. Harvey, Factors impacting false positive recall in screening mammography, Acad. Radiol. 26 (11) (2019) 1505–1512.
[6] D.U. Tari, D.R. De Lucia, M. Santarsiere, R. Santonastaso, F. Pinto, Practical challenges of dbt-guided vabb: harms and benefits, from literature to clinical experience, Cancers 15 (24) (DEC 2023), https://doi.org/10.3390/cancers15245720.
[7] V. Iotti, P. Giorgi Rossi, A. Nitrosi, S. Ravaioli, R. Vacondio, C. Campari, V. Marchesi, M. Ragazzi, M. Bertolini, G. Besutti, et al., Comparing two visualization protocols for tomosynthesis in screening: specificity and sensitivity of slabs versus planes plus slabs, Eur. Radiol. 29 (2019) 3802–3811.
[8] C. Mazo, C. Kearns, C. Mooney, W.M. Gallagher, Clinical decision support systems in breast cancer: a systematic review, Cancers 12 (2) (2020) 369.
[9] C. Zhang, J. Xu, R. Tang, J. Yang, W. Wang, X. Yu, S. Shi, Novel Research and Future Prospects of Artificial Intelligence in Cancer Diagnosis and Treatment, vol. 16, 2023, https://doi.org/10.1186/s13045-023-01514-5.
[10] P. Vuttipittayamongkol, E. Elyan, Overlap-based undersampling method for classification of imbalanced medical datasets, in: Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16, Springer, 2020, pp. 358–369.
[11] F. Thabtah, S. Hammoud, F. Kamalov, A. Gonsalves, Data imbalance in classification, Experimental evaluation 513 (2020), https://doi.org/10.1016/j.ins.2019.11.004.
[12] R. Ricciardi, G. Mettivier, M. Staffa, A. Sarno, G. Acampora, S. Minelli, A. Santoro, E. Antignani, A. Orientale, I. Pilotti, et al., A deep learning classifier for digital breast tomosynthesis, Phys. Med. 83 (2021) 184–193.
[13] J. Kim, J. Koo, S. Hwang, A unified benchmark for the unknown detection capability of deep neural networks, Expert Syst. Appl. 229 (A) (NOV 1 2023), https://doi.org/10.1016/j.eswa.2023.120461.
[14] A. Mobiny, A. Singh, H. Van Nguyen, Risk-aware machine learning classifier for skin lesion diagnosis, J. Clin. Med. 8 (8) (2019) 1241.
[15] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 arXiv preprint arXiv:1409.1556.
[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[17] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[18] J. Linmans, J. van der Laak, G. Litjens, Efficient out-of-distribution detection in digital pathology using multi-head convolutional neural networks, in: MIDL, 2020, pp. 465–478.
[19] J.E. Arco, A. Ortiz, J. Ramirez, F.J. Martinez-Murcia, Y.-D. Zhang, J.M. Gorriz, Uncertainty-driven ensembles of multi-scale deep architectures for image classification, Inf. Fusion 89 (2023) 53–65.
[20] V. Fortuin, Priors in bayesian deep learning: a review, Int. Stat. Rev. 90 (3) (2022) 563–591, https://doi.org/10.1111/insr.12502.
[21] B. Zhao, S. Yu, W. Ma, M. Yu, S. Mei, A. Wang, J. He, A. Yuille, A. Kortylewski, Ood-cv: a benchmark for robustness to out-of-distribution shifts of individual nuisances in natural images, in: European Conference on Computer Vision, Springer, 2022, pp. 163–180.
[22] D. Hendrycks, K. Gimpel, A Baseline for Detecting Misclassified and Out-Of-Distribution Examples in Neural Networks, 2016 arXiv preprint arXiv:1610.02136.
[23] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations. arXiv Preprint arXiv:1903, 2019 12261.
[24] X. Ran, M. Xu, L. Mei, Q. Xu, Q. Liu, Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation, Neural Network. 145 (2022) 199–208.
[25] H.-P. Chan, M.A. Helvie, M. Gao, L. Hadjiiski, C. Zhou, K. Garver, K.A. Klein, C. McLaughlin, R. Oudsema, W.T. Rahman, et al., Deep learning denoising of digital breast tomosynthesis: observer performance study of the effect on detection of microcalcifications in breast phantom images, Med. Phys. 50 (10) (2023) 6177–6189.
[26] Y. Sun, B. Xue, M. Zhang, G.G. Yen, J. Lv, Automatically designing cnn architectures using the genetic algorithm for image classification, IEEE Trans. Cybern. 50 (9) (2020) 3840–3854.
[27] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Completely automated cnn architecture design based on blocks, IEEE Transact. Neural Networks Learn. Syst. 31 (4) (2019) 1242–1254.

[28] Z. Chen, Y. Zhou, Z. Huang, Auto-creation of effective neural network architecture by evolutionary algorithm and resnet for image classification, in: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), IEEE, 2019, pp. 3895–3900.
[29] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Evolving deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 24 (2) (2019) 394–407.
[30] Z. Lu, I. Whalen, Y. Dhebar, K. Deb, E.D. Goodman, W. Banzhaf, V.N. Boddeti, Multiobjective evolutionary design of deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 25 (2) (2020) 277–291.
[31] T. Elsken, J.H. Metzen, F. Hutter, Efficient Multi-Objective Neural Architecture Search via Lamarckian Evolution, 2018 arXiv preprint arXiv:1804.09081.
[32] D. Laredo, Y. Qin, O. Schütze, J.-Q. Sun, Automatic Model Selection for Neural Networks, 2019 arXiv preprint arXiv:1905.06010.
[33] S. Gibb, H.M. La, S. Louis, A genetic algorithm for convolutional network structure optimization for concrete crack detection, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8.
[34] M. Loni, S. Sinaei, A. Zoljodi, M. Daneshtalab, M. Sjödin, Deepmaker: a multi-objective optimization framework for deep neural networks in embedded systems, Microprocess. Microsyst. 73 (2020) 102989.
[35] J. Huang, W. Sun, L. Huang, Deep neural networks compression learning based on multiobjective evolutionary algorithms, Neurocomputing 378 (2020) 260–269.
[36] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[37] W.-x. Wang, K.-s. Li, X.-z. Tao, F.-h. Gu, An improved moea/d algorithm with an adaptive evolutionary strategy, Inf. Sci. 539 (2020) 1–15.
[38] M. Dohopolski, L. Chen, D. Sher, J. Wang, Predicting lymph node metastasis in patients with oropharyngeal cancer by using a convolutional neural network with associated epistemic and aleatoric uncertainty, Phys. Med. Biol. 65 (22) (2020) 225002.
[39] M.S. Ayhan, P. Berens, Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks, in: Medical Imaging with Deep Learning, 2018.
[40] M. Sensoy, L. Kaplan, M. Kandemir, Evidential deep learning to quantify classification uncertainty, Adv. Neural Inf. Process. Syst. 31 (2018).
[41] T. Ramalho, M. Miranda, Density Estimation in Representation Space to Predict Model Uncertainty, 2019.
[42] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning, in: M. Balcan, K. Weinberger (Eds.), INTERNATIONAL CONFERENCE on MACHINE LEARNING, VOL 48, Vol. 48 of Proceedings of Machine Learning Research, 2016, 33rd International Conference on Machine Learning, JUN 20-22, 2016. New York, NY.
[43] J. Carrete, H. Montes-Campos, R. Wanzenboeck, E. Heid, G.K.H. Madsen, Deep ensembles vs committees for uncertainty estimation in neural-network force fields: comparison and application to active learning, JOURNAL OF CHEMICAL PHYSICS 158 (20) (MAY 28 2023), https://doi.org/10.1063/5.0146905.
[44] M. Bologna, V. Corino, L. Mainardi, Assessment of the effect of intensity standardization on the reliability of t1-weighted mri radiomic features: experiment on a virtual phantom, in: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference, vol. 2019, 2019, pp. 413–416.
[45] N. Bottenus, B.C. Byram, D. Hyun, Histogram matching for visual ultrasound image comparison, IEEE Trans. Ultrason. Ferroelectrics Freq. Control 68 (5) (2021) 1487–1495, https://doi.org/10.1109/TUFFC.2020.3035965.
[46] E. Kondrateva, M. Pominova, E. Popova, M. Sharaev, A. Bernstein, E. Burnaev, Domain shift in computer vision models for mri data analysis: an overview, Thirteenth International Conference on Machine Vision 11605 (2021) 126–133. SPIE.
[47] W.T. Le, E. Vorontsov, F.P. Romero, L. Seddik, M.M. Elsharief, P.F. Nguyen-Tan, D. Roberge, H. Bahig, S. Kadoury, Cross-institutional outcome prediction for head and neck cancer patients using self-attention neural networks, Sci. Rep. 12 (1) (FEB 24 2022), https://doi.org/10.1038/s41598-022-07034-5.
[48] A.-J. Gallego, J. Calvo-Zaragoza, R.B. Fisher, Incremental unsupervised domain-adversarial training of neural networks, IEEE Transact. Neural Networks Learn. Syst. 32 (11) (2021) 4864–4878, https://doi.org/10.1109/TNNLS.2020.3025954.
[49] M. Ilse, J. Tomczak, C. Louizos, M. Welling, Diva: Domain Invariant Variational Autoencoders [arxiv], 24 May 2019.
[50] X. Chen, J. Lv, D. Feng, X. Mou, L. Bai, S. Zhang, Z. Zhou, Automo-mixer: an automated multi-objective mixer model for balanced, safe and robust prediction in medicine 13583 (2022) 111–120. 13th International Workshop on Machine Learning in Medical Imaging (MLMI), SINGAPORE, Singapore, SEP 18, 2022.
[51] Z.G. Zhou, F. Liu, L.C. Jiao, Z.J. Zhou, J.B. Yang, M.G. Gong, X.P. Zhang, A bi-level belief rule based decision support system for diagnosis of lymph node metastasis in gastric cancer, Knowl. Base Syst. 54 (dec) (2013) 128–136.
[52] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers, Springer, 2010, pp. 177–186.
[53] Z. Zhou, M. Folkert, P. Iyengar, K. Westover, Y. Zhang, H. Choy, R. Timmerman, S. Jiang, J. Wang, Multi-objective radiomics model for predicting distant failure in lung sbrt, Phys. Med. Biol. 62 (11) (2017) 4460.
[54] X. Wang, Y. Jin, S. Schmitt, M. Olhofer, Recent Advances in Bayesian Optimization, vol. 55, 2023, https://doi.org/10.1145/3582078.
[55] Z. Zhou, G. Qin, P. Yan, H. Hao, S. Jiang, J. Wang, A shell and kernel descriptor based joint deep learning model for predicting breast lesion malignancy, Medical Imaging 2019: Computer-Aided Diagnosis 10950 (2019) 713–720. SPIE.
[56] X. Chen, X. Wang, J. Lv, G. Qin, Z. Zhou, An integrated network based on 2d/3d feature correlations for benign-malignant tumor classification and uncertainty estimation in digital breast tomosynthesis, Phys. Med. Biol. 68 (17) (2023) 175046.
[57] L. Hu, Z. Wang, H. Li, P. Wu, J. Mao, N. Zeng, -darts: light-weight differentiable architecture search with robustness enhancement strategy, Knowl. Base Syst. (2024) 111466.
[58] H. Li, Z. Wang, C. Lan, P. Wu, N. Zeng, A novel dynamic multiobjective optimization algorithm with non-inductive transfer learning based on multi-strategy adaptive selection, IEEE Transact. Neural Networks Learn. Syst. (2023).