



Research article

BreaCNet: A high-accuracy breast thermogram classifier based on mobile convolutional neural network

Roslidar Roslidar^{1,2,3}, Mohd Syaryadhi², Khairun Saddami², Biswajeet Pradhan^{4,5,6}, Fitri Arnia^{2,3}, Maimun Syukri⁷ and Khairul Munadi^{2,3,8,*}

¹ Doctoral Program, School of Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia

² Department of Electrical and Computer Engineering, Universitas Syiah Kuala, Indonesia

³ Telematics Research Center, Universitas Syiah Kuala, Banda Aceh, Indonesia

⁴ Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and IT, University of Technology Sydney, Ultimo, Australia

⁵ Center of Excellence for Climate Change Research, King Abdulaziz University, Jeddah, Saudi Arabia

⁶ Earth Observation Center, Institute of Climate Change, Universiti Kebangsaan Malaysia, Bangi, Malaysia

⁷ Medical Faculty, Universitas Syiah Kuala, Banda Aceh, Indonesia

⁸ Tsunami and Disaster Mitigation Research Center, Universitas Syiah Kuala, Banda Aceh, Indonesia

* **Correspondence:** Email: khairul.munadi@unsyiah.ac.id.

Abstract: The presence of a well-trained, mobile CNN model with a high accuracy rate is imperative to build a mobile-based early breast cancer detector. In this study, we propose a mobile neural network model breast cancer mobile network (BreaCNet) and its implementation framework. BreaCNet consists of an effective segmentation algorithm for breast thermograms and a classifier based on the mobile CNN model. The segmentation algorithm employing edge detection and second-order polynomial curve fitting techniques can effectively capture the thermograms' region of interest (ROI), thereby facilitating efficient feature extraction. The classifier was developed based on ShuffleNet by adding one block consisting of a convolutional layer with 1028 filters. The modified Shufflenet demonstrated a good fit learning with 6.1 million parameters and 22 MB size. Simulation results showed that modified ShuffleNet alone resulted in a 72% accuracy rate, but the performance excelled to a 100% accuracy rate when integrated with the proposed segmentation algorithm. In terms of diagnostic accuracy of the normal and abnormal test, BreaCNet significantly improves the sensitivity rate from 43% to 100% and specificity of 100%. We confirmed that feeding only the ROI of the input dataset to the network can improve the classifier's performance. On the implementation aspect of BreaCNet, the on-device inference is recommended to ensure users' data privacy and handle an unreliable network connection.

Keywords: breast cancer; mobile cnn; self-screening; thermogram; segmentation

1. Introduction

Computer vision and deep learning (DL) have achieved the utmost progress in viewing images at the same level as that of humans [1] through the process of learning such as in medical image classification [2–8]. Supported by publicly accessible datasets, computer-aided works based on image processing and DL for medical interpretation have been increasingly improved. In breast cancer detection, DL has been employed to classify medical images of mammography [9, 10], ultrasound [11], histopathological image [12–16], and thermography [17–22]. Despite the high accuracy rate of the deep neural networks (NNs) applied to these modality images, the procedure for obtaining the images requires an individual to visit a specific hospital to perform the screening. It is a constraint for many people with limited mobility, such as those living far from the hospital or having other restrictions.

Moreover, thermography is a noninvasive early detector that can be promoted as a handy pre-cancer screening tool [23]. Early detection means identifying breast masses when they are still in the treatable stage with the least psychological and physical harm [24]. Therefore, developing and promoting of an early detector and self-screening tool for precancer are needed to prevent breast cancer and minimize the mortality rate.

Additionally, WHO has recommended that women should take responsibility for their health by performing a breast self-examination. Preliminary research [25] also confirmed that screening, which is a systematic procedure to identify an individual with an abnormality suggestive of cancer [26], can reduce the incidence rate. Hence, a handy screening tool is highly required to allow women to perform breast self-screening regularly.

A handy precancer screening tool based on thermography and DL can be an effective tool for breast self-examination. Supported by the availability of publicly accessible datasets and the projection of 13.1 billion global mobile devices in 2023 [27], we believe a handy self-screening device can be achieved at a low cost. In addition, smartphones integrated with a thermal camera [28–30] have also been introduced into the market.

Further, the performance of DL has inspired attempts to provide high-quality intelligent services on mobile devices. Nevertheless, our study indicated that the integration of DL and mobile devices is still at the preliminary stage. Thus, further work should be conducted by considering the fundamental requirements of a mobile application.

Requirements for a mobile application:

In deploying a DL model into a mobile application, we have to first decide the model inference location: on the cloud server or local mobile device [31]. Inference on the cloud server deploys a complex NN model and maintains the simplicity of the mobile application. However, some issues may arise as a result of this method, such as the lack of users' data privacy and the inability of some patients to use the application in areas with poor internet connection [32]. By contrast, the inference of a NN model on the local mobile device requires a less complex model that will allow the integration with a mobile application. For practical examples, Apple places a limit of 200 MB on the App Store [33], whereas Play Store requires that the compressed Android Package Kit be no more than 100 MB [34].

Since the intended mobile application is for breast cancer screening, a user's image has to be confidential. In addition, regular screening should not depend on the internet connection. Thus, we recom-

mend that the inference (i.e., classification or prediction task) be localized within the mobile device.

To enable on-device inference, the following requirements have to be met:

- The input image should contain rich features. To obtain rich features from an image captured using a cell phone, the image should be preprocessed with a simple and efficient algorithm.
- The mobile NN classifier should be deployable in the local mobile devices.
- As the application is for medical purposes, it should have the highest accuracy rate.

Considering the above requirements, we developed an efficient algorithm based on a convolutional neural network (CNN) model that can classify breast thermograms at a high accuracy rate. The classifier model breast cancer mobile network (BreaCNet), consists of a new segmentation algorithm and a mobile NN.

The contributions of this study are as follows:

- It highlights the mobile application requirements for breast thermogram classification.
- It proposes a simple segmentation algorithm that suits the characteristics of breast thermograms to provide rich features.
- It provides a good fit mobile CNN model based on ShuffleNet.
- It introduces a high accuracy classifier model called BreaCNet consisting of the proposed segmentation algorithm and the mobile CNN model.
- It proposes an implementation framework of the classifier model in a mobile application.

The rest of this paper is organized as follows. Section 2 presents the related works, and Section 3 describes the materials and methods used in this work. BreaCNet's development and its implementation framework for a mobile application are clearly explained in Section 4, followed by the model performance discussion in Section 5. Finally, Section 6 concludes this study.

2. Related work

Numerous studies have been devoted to breast cancer detection based on thermography and DL since 2018 [23]. The works mostly used the image datasets from the database for mastology research (DMR) [35]. The examples of breast thermograms downloaded from DMR are shown in Figure 1(a),(b) which presents the normal and abnormal thermograms in RGB and grayscale, respectively. The abnormal breast thermogram was obtained from a patient with a medical history of mammography and a sign of cancer on the right breast. The normal and abnormal thermograms were nearly indistinguishable by the naked eye. However, when the statistical feature analysis was employed, there was a significant difference found between the temperature distribution in the normal breast thermogram and that of the abnormal one.

As illustrated in Figure 1(c), the histogram of the normal breast showed that both sides of the breast have similar temperature distributions and a lower mean temperature compared with that of the abnormal one Figure 1(d). Thus, the symmetrical characteristics of a breast thermogram can indicate the signs of normality and abnormality in breast tissues [36] and can be an alternative medical imaging modality to detect breast cancer symptoms at an early stage.

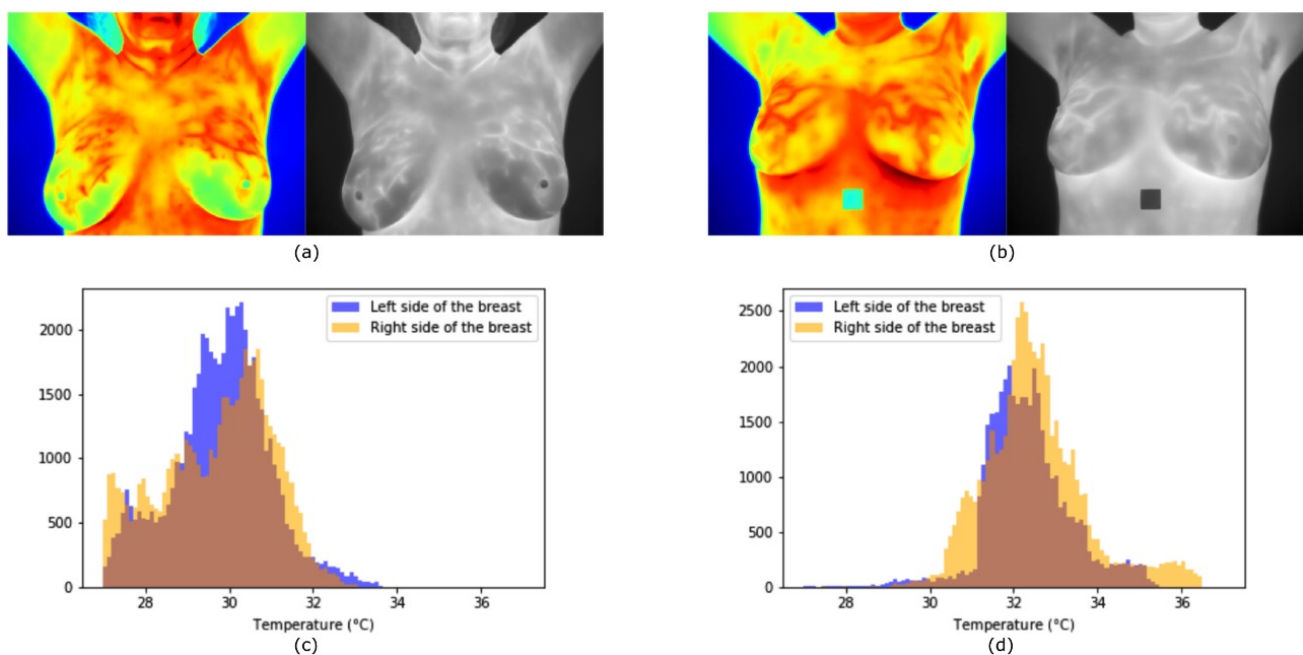


Figure 1. Breast thermograms from DMR. (a) normal thermogram, (b) abnormal thermogram, (c) the histogram distribution of (a) and (d) the histogram distribution of (b).

Lightweight CNN model:

A lightweight CNN is a compressed model in the perspective of weight and architecture [37]. The lightweight model contains few network parameters to minimize the memory usage and increase the computational speed. Many works have been conducted to develop a lightweight model, such as work by Winoto et al. [38] that built a CNN model with only 0.88 million parameter trained on SVHN dataset and Shuvo et al. [39] that developed a CNN model with 3.76 million parameters trained on lung sound dataset. Meanwhile, some lightweight pretrained models that were trained on DMR dataset as presented in Table 1 are MobileNetV2 [40], Xception [41], ResNet18 [42] and ShuffleNet [43].

Among those lightweight models, MobileNetV2 and ShuffleNet shows a minimal parameter learning. MobileNetV2 was developed based on MobileNet [44] that applied depthwise separable convolutions to reduce the computation in the first few layers. The computation was less because a parameter called width multiplier with a range value of $(0, 1]$ was introduced to lender the network uniformly at each layer. Another hyperparameter used to reduce the computational cost is a resolution multiplier to the input image with a value in the range of $(0, 1]$. MobileNetV2 maintains the simplicity of MobileNet and introduces linear bottlenecks and inverted residuals. Linear layers prevent nonlinearities from destroying information, whereas the inverted residual allows the shortcut connections between thin bottleneck layers. With its light architecture, MobileNetV2 has a computational cost of 300 million multiply-adds, 3.5 million parameters, and a 13 MB size. Meanwhile, ShuffleNet introduced a channel shuffle to help the information flow across feature channels to overcome information loss due to the use of the rectified linear unit (ReLU). Pointwise group convolutions were also employed to reduce the computational complexity of 1×1 convolution. Using this technique, ShuffleNet performed with only 1.4 million learning parameters and a 5.4 MB size.

Table 1. Properties of the most commonly used pre-trained CNNs on DMR [45].

Network	Depth	Size (MB)	Parameter (million)	Image input size
GoogleNet	22	5.2	7.0	224 × 224
Inceptionv3	48	89	23.9	299 × 299
DenseNet	201	77	20.0	224 × 224
MobileNetV2	53	13	3.5	224 × 224
ResNet18	18	44	11.7	224 × 224
ResNet50	50	96	25.6	224 × 224
ResNet101	101	167	44.6	224 × 224
Xception	71	85	22.9	299 × 299
ShuffleNet	50	5.4	1.4	224 × 224
AlexNet	8	227	61.0	227 × 227
VGG16	16	515	138	224 × 224
VGG19	19	535	144	224 × 224

We observed some studies that conducted on CNN model trained on DMR as shown in Table 2. MobileNetV2 and ShuffleNet had been fine-tuned, trained, and tested by Roslidar et al. [17] and were confirmed to outperform deep CNN in the binary classification of breast thermograms with optimum accuracy and low training loss rate. Fernandes et al. [18] stated that RestNet18, which uses the lowest number of parameters among other CNN models (ResNet34, ResNet50, ResNet152, VGG16 and VGG19), has excellent stability performance. Zuluaga-Gomez et al. [19] proposed a handcrafted CNN structure. However, its accuracy rate was only 92%. Meanwhile, Tello-Mijares et al. [21] trained AlexNet on DMR with segmentation preprocessing and achieved a 100% accuracy rate. Nevertheless, the segmentation in their study required a complex algorithm. Recently, Sánchez-Cauce et al. [22] proposed multiple inputs in the forms of breast thermograms and clinical data fed into CNN to improve the performance. Their system achieved a 97% accuracy rate.

The aforementioned studies indicated that light networks are more stable than deep networks in performing a classification task. However, simple CNN models built from scratch were found to have a lower accuracy rate than those of the pre-trained ones. Research by Tajbakhsh et al. [46] compared the performance of a pre-trained CNN with thoses of the handcrafted ones. They revealed that the implementation of a pre-trained CNN with fine-tuning and training on medical images can outperform or, in the worst case, as good as a CNN built from scratch. Moreover, fine-tuned CNNs are more robust to the size of the training dataset. Thus, fine-tuning pre-trained CNN models is a good strategy for building a CNN model for analyzing medical images that are usually exposed to a very limited dataset.

Based on the previous research findings, here, we implemented transfer learning on MobileNetV2 and ShuffleNet. These models cost few learning parameter with minimal memory usage. Moreover, these models have been confirmed to excellently perform when trained on DMR dataset and a binary group of classification [17].

Table 2. Works that implemented CNN on the DMR dataset.

Work	Model	Contribution	Limitation
Roslidar et al. [17]	Pre-trained Network	Confirmed that the lightweight outperform dense CNN	The learning curve was not provided
Fernandez et al. [18]	Pre-trained Network	Provided stability analysis on trained CNN	Did not propose a method to enrich the image feature
Zuluaga-Gomez et al. [19]	Proposed CNN	Proposed data augmentation	Low accuracy
Tello et al. [21]	Pre-trained Network	Worked on preprocessing and classification	Complex segmentation algorithm; the CNN model is not light
Sánchez-Cauce et al. [22]	Proposed CNN	Multi-input of breast thermograms and clinical data	Information about CNN development is unknown

3. Material and methods

Figure 2(a) shows the workflow of model development followed by the deployment. The input images were priorly segmented to allow rich features fed into a CNN model. The CNN model was built by applying transfer learning. Then, the model was deployed as a mobile or web-based application. Figure 2(b) demonstrates the model (BreacNet) development and implementation framework. More detailed descriptions of each working process of BreacNet development and implementation are described in Section 4.

3.1. Dataset

The breast thermogram dataset used in this study was obtained from DMR [35], which has been publicly used in related research. The thermograms were acquired using static and dynamic protocols [47]. The static protocol is a single captured image after 10–15 minutes of thermal stabilization during patients' resting period, whereas the dynamic protocol is a thermogram series captured every 15 seconds in five minutes. The images were captured from the front, left, and right sides of the patients' positions. We used the front images of 33 sick and 121 healthy patients. There were 121 frontal static images and 2581 frontal dynamic ones labeled as normal breast thermograms, and 33 frontal static images and 676 frontal dynamics ones labeled as abnormal breast thermograms. Thus, in total, we had 2702 normal breast thermograms and 709 abnormal ones.

The thermograms of normal and abnormal classes are imbalanced in number, in which the number of abnormal thermograms is far lower than that of normal ones. We started the training and testing dataset setups by grouping the thermograms of each patient, one group for the training and the other for the testing. Accordingly, we had an equal number of 586 for both the normal and abnormal thermograms for the training dataset. Then, we took 65 thermograms of each class for the testing dataset. Thus, in total, we used 1172 (90%) and 130 (10%) breast thermograms for the training and testing, respectively. For the validation dataset, we assigned 10% of the training data.

We used more thermograms for the model training than for the testing to enable more learning. This approach is supported by Cho et al. [48] which confirmed that accuracy is proportional to the number of training dataset.

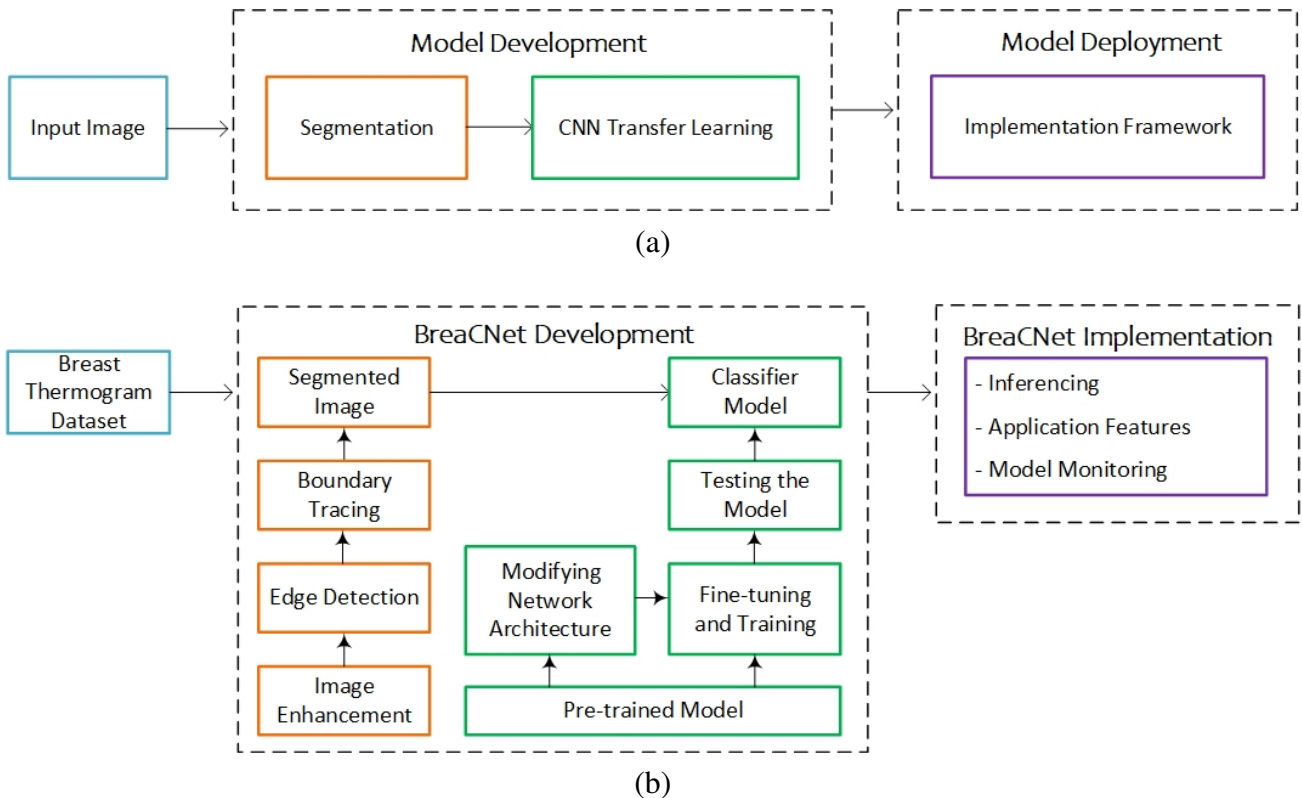


Figure 2. The workflow of (a) general CNN model development and deployment and (b) BreaCNet development and implementation.

3.2. Model development

In this study, the model is intended to classify breast thermograms and will be implemented as a mobile application to allow regular breast self-screening. As previously mentioned in Section 2, for a limited medical dataset, it is better to apply transfer learning because it will allow the model trained on a large dataset to transfer its knowledge to a smaller dataset. Since the breast thermogram dataset in this work was limited, we employed transfer learning to build the model.

Pre-trained models were fine-tuned and trained on the breast thermogram dataset. For each training validation with a 100% accuracy rate, the model was then tested with the testing dataset. To achieve the highest performance, we modified the architecture by adding more layers and filters. Thus, the NN will learn the input feature better.

The model performance was observed from the training and validation learning curves during the training process. The training learning curve shows how well the model learns, whereas the validation learning curve shows how well the model generalizes. We also measured the performance using the evaluation metrics. The evaluation metrics used here are the ones commonly considered in diagnostic medicine-accuracy, sensitivity, and specificity-which are calculated using Eqs (3.1)–(3.3) [49].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.1)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (3.2)$$

$$Specificity = \frac{TN}{TN + FP} \quad (3.3)$$

where TP , TN , FP and FN indicate true positive, true negative, false positive and false negative images. A true positive is an outcome where the model correctly classifies the abnormal category; a true negative is an outcome where the model correctly classifies the normal category; a false positive is an outcome where the model incorrectly predicts the abnormal category; a false negative is an outcome where the model incorrectly classifies the normal category. Meanwhile, sensitivity indicates the proportion of positive results correctly identified by the testing and specificity as the proportion of negative results correctly identified by the testing.

3.3. Model deployment

After obtaining a high-performance model, we designed the implementation framework for model deployment. The model deployment can be an application for mobile or web-based one. This study enclosed the part of the model deployment's implementation framework, which includes the inferencing preference, application overview, and model monitoring strategy. In determining the inference (classification task) location, we considered the primary usage, and tradeoffs that might arise. Inferencing on the cloud will allow complex model algorithm implementation. It is suitable for commercial or public service usage. However, for individual usage or self-screening, inferencing on the local mobile device is better because it will ensure data privacy and independence on the internet connection.

4. Proposed breacNet and its implementation framework

As shown in Figure 2(b), BreaCNet covers the image segmentation and classification processes. We developed an effective segmentation algorithm that compiles the image enhancement, edge detection, and boundary tracing to obtain the ROI of breast thermal images. BreaCNet's classifier model was built by employing transfer learning of the lightweight pre-trained CNN model. The classification process consisted of modifying the architecture of the pre-trained CNN model, fine-tuning, training, and testing the model repetitively until it achieved good fit performance with a high accuracy rate. Meanwhile, the BreaCNet deployment discussion covers the implementation framework of inferencing, application features, and model monitoring. Each step of the BreaCNet development and its implementation framework is explained below.

4.1. BreaCNet development

BreaCNet consists of segmentation and mobile CNN algorithms. The segmentation algorithm was built by considering the breast thermogram characteristics with an efficient algorithm. The objective was to obtain the region of interest (ROI) of each breast thermogram. Meanwhile, the CNN models were based on the pre-trained MobilenetV2 and Shufflenet that had been fine-tuned, trained, tested and

modified to achieve high accuracy. The model will be trained and tested using the segmented dataset and the raw dataset to assess the effects of the segmentation process on the model performance.

4.1.1. Segmentation

The quality of input influences the performance of CNN. Feeding only the ROI of breast thermograms to a CNN model may accelerate the feature learning because it will only learn the important parts of the input. Thus, we proposed a segmentation algorithm for breast thermograms to provide rich features of the input. The segmentation algorithm will define the ROI of the breast thermograms, which includes half of the armpit, collarbone, and chest, in which all breast tissues and nearby ganglion groups were analyzed [36].

The ROI extraction of breast thermogram images is challenging due to the amorphous nature and the lack of clear boundary in these images [50]. The ROI's unclear edges of breasts makes it difficult to accurately perform segmentation at the border of the inframammary fold-the anatomical boundary formed at the breast's inferior border-where it joins the chest. Moreover, each breast thermogram exposes various intensity distributions at the boundary of the ROI area. Thus, a specific and automatic segmentation algorithm that is applicable to all breast thermograms is required.

Here, we propose an automatic breast's ROI boundary tracer based on Sobel edge detection. The inevitable low contrast and noise around the inframammary folds [51] were addressed using second-order polynomial curve fitting. A similar method was proposed by Sathish et al. [52]; however, the number of breast thermograms that could be segmented using their proposed algorithm was minimal. Provided that the CNN model training requires much training data, we improved the segmentation algorithm to overcome this issue.

Unlike the work of Sathish that applied Canny filtering for edge detection, we employed Sobel filtering to sharply take the outer boundary of the breasts' ROI edges. The segmentation algorithm is presented in Algorithm 1. The algorithm consists of image smoothing, image edge detection, breasts' ROI boundary tracing, and image masking.

First, we converted the RGB image into a grayscale image. Then, we applied Gaussian filtering to the grayscale image for smoothing. We used a variance value of 3 since it was the best value in our trials. Using this variance value and the Sobel kernel, a sharp edge boundary could be generated. The edge boundary was further used to trace the outer boundary of the breast's ROI.

Before tracing the boundary, the image was divided at the image's central point (C_i) into the right and left side. Then, the outer boundaries of the right and left side were traced using the edge value of 1 from the Sobel edge detector. Meanwhile, the top boundary was obtained by scanning the image from the bottom to the top. The first nonzero pixel in the column was the initial point of the top border.

Afterwards, we approximated the bottom boundary using the second-order polynomial curve fitting, $p(x)$, for each side of the breast using Eq (4.1) [53].

$$p(x) = p_1x^2 + p_2x + p_3 \quad (4.1)$$

To minimize the computation, only four points were assigned for the polynomial curve fitting for the right and left sides of the breasts. The first point was determined by calculating the histogram of a horizontal projection profile (H_{pp}) from the bottom using Eq (4.2) [54].

Algorithm 1 Proposed breast thermogram segmentation.**Input:** Breast thermogram images in RGB, I_0 **Output:** ROI of the breast thermogram images, I_S

- 1: **for** each image in dataset **do**
- 2: $I_0 \leftarrow$ read the image
- 3: $I_g \leftarrow$ convert RGB image of I_0 to grayscale
- 4: $I_G \leftarrow$ apply Gauss filtering to I_g
- 5: $I_E \leftarrow$ apply Sobel filtering to I_G to find the edges
- 6: $C_t \leftarrow$ find the center of the image I_E
- 7: left and right boundary $\leftarrow I_E$ value is 1 from the left and right
- 8: top boundary \leftarrow scan I_E from the bottom, find the first pixel with value of 1
- 9: $H_{pp} \leftarrow$ calculate the sum of value 1 in each row from the left and right
- 10: the first point of the polynomial \leftarrow the first highest H_{pp} from the lowest position of I_E
- 11: the second, third, and fourth point of the polynomial \leftarrow the indices along the bottom boundary of edges of I_E
- 12: **end for**
- 13: **return** I_S

$$H_{pp} = \sum_x f(x, y) \quad (4.2)$$

The first pixel with the highest H_{pp} was the curve's first point. Since the edges of the bottom inframammary fold demonstrated discontinuity in some images (Figure 3 (c)), we applied some constraints to keep the indices inline. If the first point of (x, y) were L_{x1} and L_{y1} , the next points were:

$$Lx_m = Lx_{m-1} + C \quad (4.3)$$

and

$$Ly_m = Ly_{m-1} - m^2 \quad (4.4)$$

where m and C denote the following points and the increment in the distance between indices, respectively.

Then, indices of boundary tracing were applied to the original image to obtain the segmented breast thermogram. The segmentation processes, along with the results of each process, are shown in Figure 3. The original images Figure 3(a) as the inputs were first converted to Figure 3(b) grayscale images, which were then smoothed using the Gaussian filtering. Next, the edges were extracted using the Sobel edge detector resulting in Figure 3(c) images with edges. The information on edges was then used to obtain Figure 3(d) the outer boundary of the breast ROI. Finally, we obtained Figure 3(e) the segmented images after masking Figure 3(d) the indices of the outer boundary to the Figure 3(a) original images.

Our segmentation algorithm can segment all breast thermograms in the dataset, enabling sufficient training data for the NN. In addition, the algorithm requires a simple computation, allowing it to be integrated into the CNN model to support automatic segmentation in the mobile application.

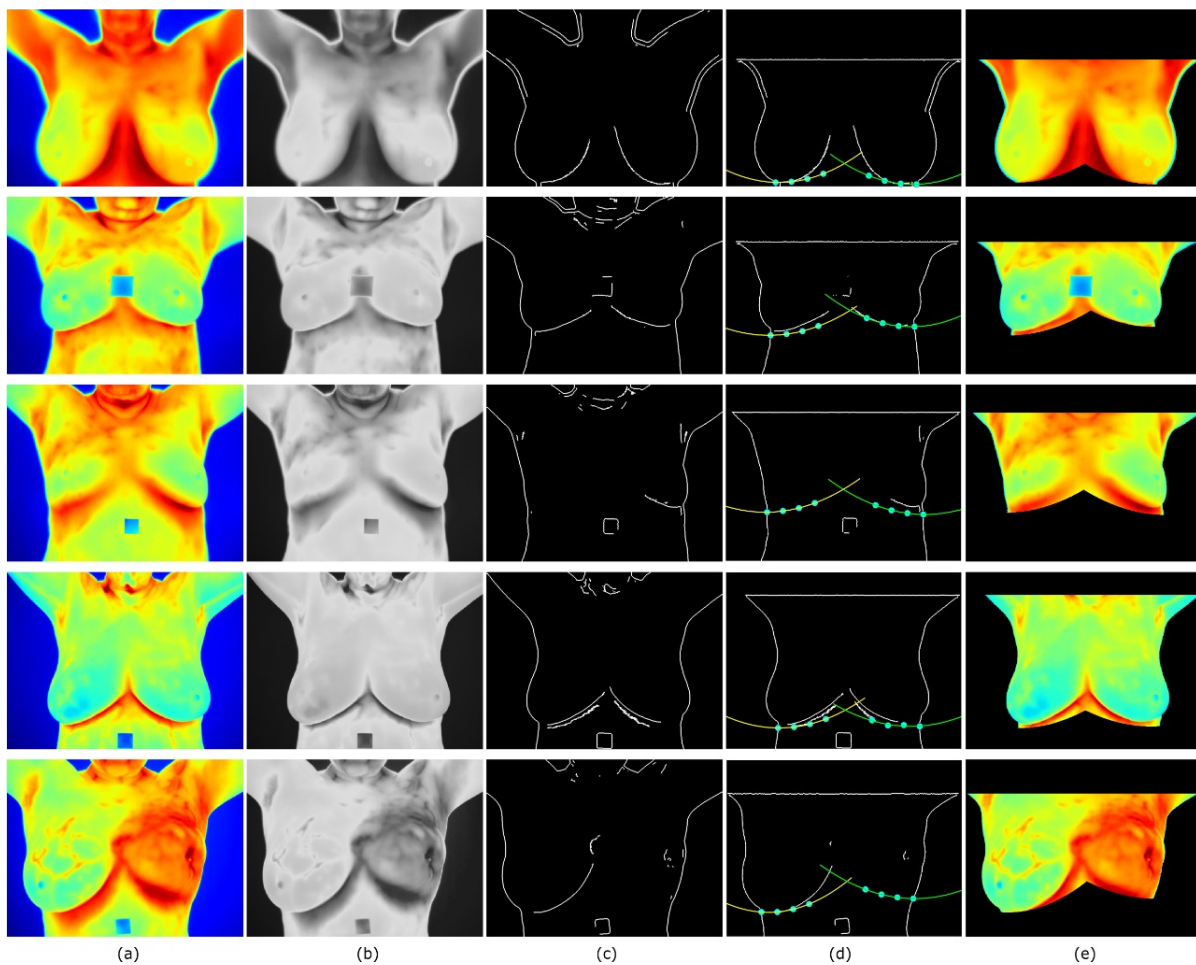


Figure 3. Breast thermogram segmentation process: (a) RGB images; (b) grayscale images; (c) images with edges; (d) boundaries of the images; (e) segmented images.

4.1.2. CNN model

A CNN is a DL network which takes an input, assigns learnable weight/biases to various aspects of the input, and classifies it into a specific group [55]. Generally, the input is an image. Image preprocessing is usually not required in CNNs as they can learn the features/characteristics, unlike conventional methods where a filter has to be hand-engineered. Generally, CNNs work similarly to a common NN that performs computations through a process of learning [56]. Two main functions that differentiate the CNNs from other NNs are the convolution and pooling functions (Figure 4).

The convolution function extracts the features from an image using a filter/kernel which consists of weight matrixes, resulting in feature maps. The weights of the kernels are randomly generated in the size of 1×1 , 3×3 , 5×5 or 7×7 . If the input is in RGB, which has three channels, then the kernel size will be $1 \times 1 \times 3$, $3 \times 3 \times 3$, $5 \times 5 \times 3$ or $7 \times 7 \times 3$. The number of filters is usually in the multiples of 2, such as 32, 64, 128 and so forth [56].

The feature maps become the input of the pooling, specifically after the application of nonlinearity [57]. The nonlinear activation function takes a real-valued input and squashes it into a small range, such as $[0, 1]$, for the ReLU activation function [58].

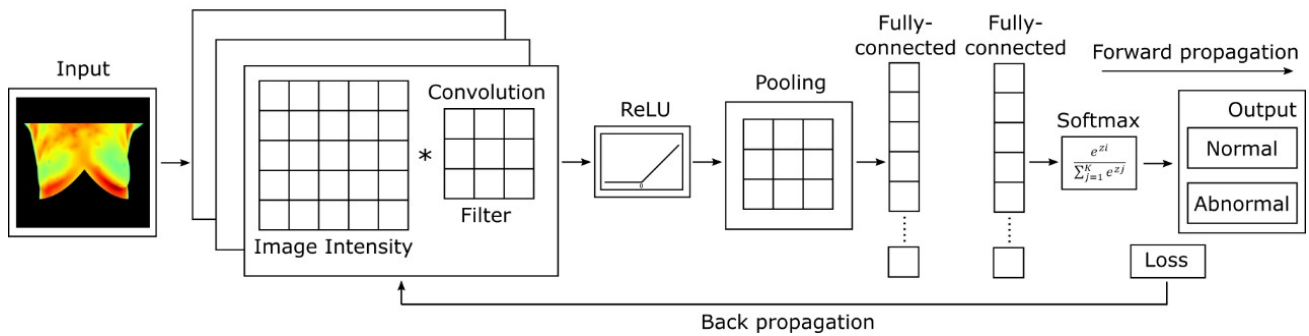


Figure 4. CNN diagram for breast thermogram classification in binary classes.

The pooling function progressively reduces the spatial size of the feature maps and keeps only the relevant features. Here, the maximum or average value of the feature matrix is determined by the function used (maximum, minimum, or average pooling) [55,59]. Thus, the number of parameters and computations in the network can be reduced.

Convolution and pooling are usually conducted in many layers to enable optimum feature learning. The output of the last pooling layer is flattened to justify the fully-connected layer that accepts an array input. A fully-connected layer is usually placed at the end of the output classification. The last fully-connected layer has a similar size to the number of classification class.

In this study, the classification task was performed using a lightweight CNN model to provide model inferencing on user-end devices for breast self-screening. The pre-trained MobileNetV2 [40] and ShuffleNet [43] were trained and fine-tuned to achieve optimal performance. Each network was trained and tested two times with the raw dataset (without preprocessing) and the segmented dataset. To optimize the accuracy rate, we modified the architecture of the networks. Then, for every pre-trained network with a training validation of 100%, we conducted the testing simulations.

Training and fine-tuning of CNN models:

Training and fine-tuning processes of the pre-trained model are presented in Figure 5. The initial step was loading and reading the breast thermogram dataset. Then, the dataset was divided into the training and testing dataset in the proportion of 90% and 10%, respectively.

The next step was training a network using the given dataset. The pre-trained network was loaded and fine-tuned with the learning parameters of optimization, initial learning rate (ILR), maximum epoch, mini-batch size (MBs), and momentum. For optimization, we employed the stochastic gradient descent optimizer with momentum (SGDM) [60]. Gradient descent enabled us to update each parameter in a network by iteratively selecting a direction that would reduce the error rate until the objective functions converged to the minimum value. The stochastic gradient descent is a variant of gradient descent computing only on a small subset random selection of data but can yield the same performance as the gradient descent with a low learning rate. The ILR was manually set up on a log scale from 10^{-3} to 10^{-4} . This method, called the learning rate grid search, boosts the order of magnitude where a good learning rate may reside and describes the relationship between the learning rate and performance [61].

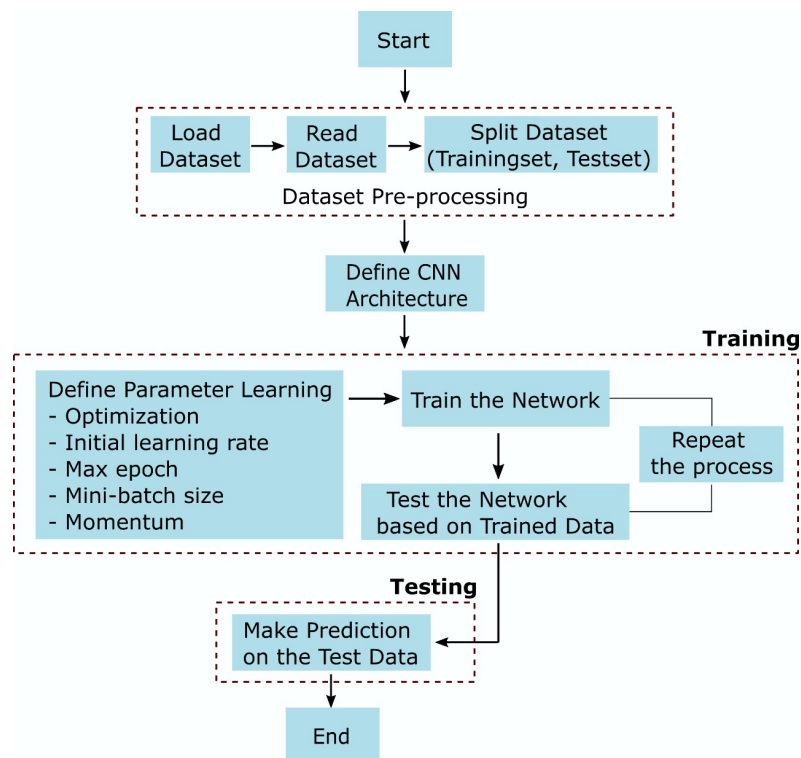


Figure 5. The training and fine-tuning process of the pre-trained network.

Further, we assigned MB sizes of 10 and 12, considering the small number of the training dataset and the computation resource. Meanwhile, the momentum, a moving average of the gradients to update the weight of the network, was set to 0.9 to avoid fluctuation (with smaller momentum) and shifting value (with higher momentum) [62].

The number of epochs, a hyperparameter that determines how many times the learning algorithm will work through the entire training dataset, was set from 50 and forth with a step size of 25. One epoch means that each sample in the training dataset has an opportunity to update the internal model parameters. An epoch comprises one or more batches.

Besides tuning the parameters, the raw and segmented datasets were fed alternately into the networks. Thus, we were able to assess the segmentation effects on training accuracy improvement. The final step was testing the trained network to predict the class of the testing dataset (raw or segmented). The prediction results were used to calculate the evaluation metrics and project the confusion matrix.

Proposed mobile CNN model:

As fine-tuning the pre-trained networks had not yet achieved optimum accuracy, the architecture of the base models was then modified. The last block was removed and replaced with a new activation function of convolution, ReLU and pooling. The number of filters was increased to generate more kernels for better learning. This procedure was performed for both pre-trained models. After the network modification, we repeated the training procedure until optimum accuracy was achieved. The modified MobileNetV2 was found to achieve a maximum accuracy rate of 98% using the segmented dataset, whereas the modified ShuffleNet could achieve a maximum accuracy rate of 100%.

The structure of the modified ShuffleNet is shown in Figure 6. We removed the last block of the

ShuffleNet, then added a new block of activation function as follows: one convolutional layer with 1028 filters, followed by the average pooling, ReLU, global average pooling, and fully connected layer of 256. Then, the dropout was applied with the probability of 50%. The last fully connected layer was connected to the output consisting of two classes with a softmax activation function.

The parameters of the modified ShuffleNet are summarized in Table 3. As more filters were employed, the learning parameter increased. The modified ShuffleNet was performed with 6.1 million learning parameters and 22 MB in size.

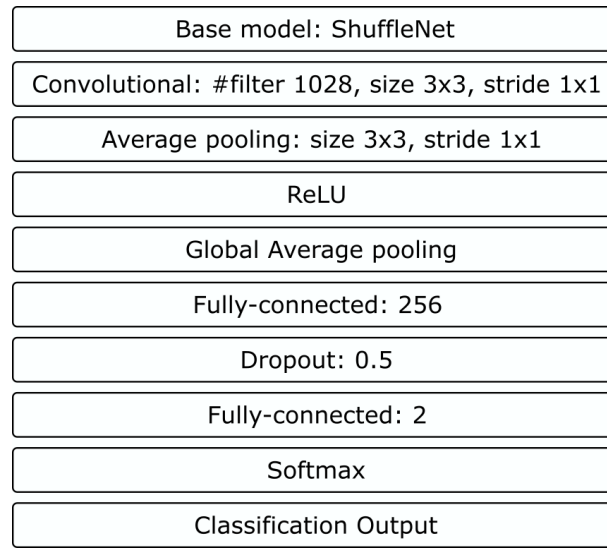


Figure 6. The proposed CNN architecture based on ShuffleNet.

Table 3. Parameter of the proposed CNN.

Layer	Input	Output	Filter size	Filter number	Stride	Padding	Probability
ShuffleNet-baseline	$224 \times 224 \times 3$	$7 \times 7 \times 544$	–	–	–	–	–
Convolution	$7 \times 7 \times 544$	$7 \times 7 \times 1028$	3×3	1028	1	'same'	–
Average pooling	$7 \times 7 \times 1028$	$7 \times 7 \times 1028$	3×3	–	1	'same'	–
ReLU	$7 \times 7 \times 1028$	$7 \times 7 \times 1028$	–	–	–	–	–
Global average pooling	$7 \times 7 \times 1028$	1028	–	–	–	–	–
Fully-connected1	1028	256	–	–	–	–	–
Dropout	–	–	–	–	–	–	50%
Fully-connected2	256	2	–	–	–	–	–

4.1.3. Testing result

The testing results were recorded and summarized in Table 4. Notably, the recorded testing results were those with 100% training accuracy after each fine-tuning. The testing results showed that training the ShuffleNet using an ILR of 10^{-3} and MBs of 10 can achieve the highest accuracy rate when the model was trained at 75 epochs. However, when we applied a lower ILR of 10^{-4} with 100 and 150 epochs, the accuracy rates decreased. MobileNetV2, on the other hand, did not show any trend when the learning parameters were tuned. Increasing the number of epochs also did not improve the learning.

Table 4. Classification results.

Model	Learning Parameter	Testing Dataset	Accuracy	Sensitivity	Specificity
ShuffleNet [43]	50 Epoch, ILR 0.001,	raw	0.71	0.48	0.94
	MBs 10, raw dataset	segmented	0.82	0.78	0.86
	50 Epoch, ILR 0.001,	raw	0.87	0.88	0.86
	MBs 10, segmented dataset	segmented	0.98	1.00	0.95
	75 Epoch, ILR 0.001,	raw	0.63	1.00	0.26
	MBs 10, segmented dataset	segmented	0.98	1.00	0.97
	50 Epoch, ILR 0.0001,	raw	0.71	0.48	0.94
	MBs 10, segmented dataset	segmented	0.87	0.88	0.86
	100 Epoch, ILR 0.0001,	segmented	0.82	0.78	0.86
	MBs 10, segmented dataset	segmented	0.59	0.60	0.58
MobileNetV2 [40]	50 Epoch, ILR 0.001,	raw	0.68	1.00	0.37
	MBs 10, raw dataset	segmented	0.96	1.00	0.92
	50 Epoch, ILR 0.001,	segmented	0.92	1.00	0.85
	MBs 10, segmented dataset	segmented	0.92	1.00	0.83
	75 Epoch, ILR 0.001,	segmented	0.92	1.00	0.83
	MBs 10, segmented dataset	raw	0.88	0.83	0.92
	100 Epoch, ILR 0.001,	segmented	0.87	0.82	0.92
Modified-MobileNetV2	100 Epoch, ILR 0.0005,	raw	0.75	0.51	1.00
	MBs 10, raw dataset	segmented	0.98	0.98	0.97
	100 Epoch, ILR 0.0005,	segmented	0.98	0.98	0.97
MB 10, segmented dataset					
Proposed Modified-ShuffleNet	100 Epoch, ILR 0.0005,	raw	0.72	0.43	1.00
	MBs 12, raw dataset	segmented	0.85	0.728	0.97
	100 Epoch, ILR 0.0005,	raw	0.98	0.98	0.98
	MBs 12, segmented dataset	segmented	1.00	1.00	1.00

The modified MobileNetV2 obtained the maximum accuracy rate of 98% when trained and tested with the segmented dataset, and impressively, the modified ShuffleNet excelled the learning with a 100% accuracy rate when trained using segmented dataset. On average, the accuracy rate improved by more than 9% when the segmented dataset trained the model. The classification results of the proposed model are also presented in image data as shown in Figure 7. The proposed model can correctly classify all breast thermal images of raw and the segmented dataset. While without segmentation algorithm, which is shown by the raw dataset, some False Positive numbers occurred.

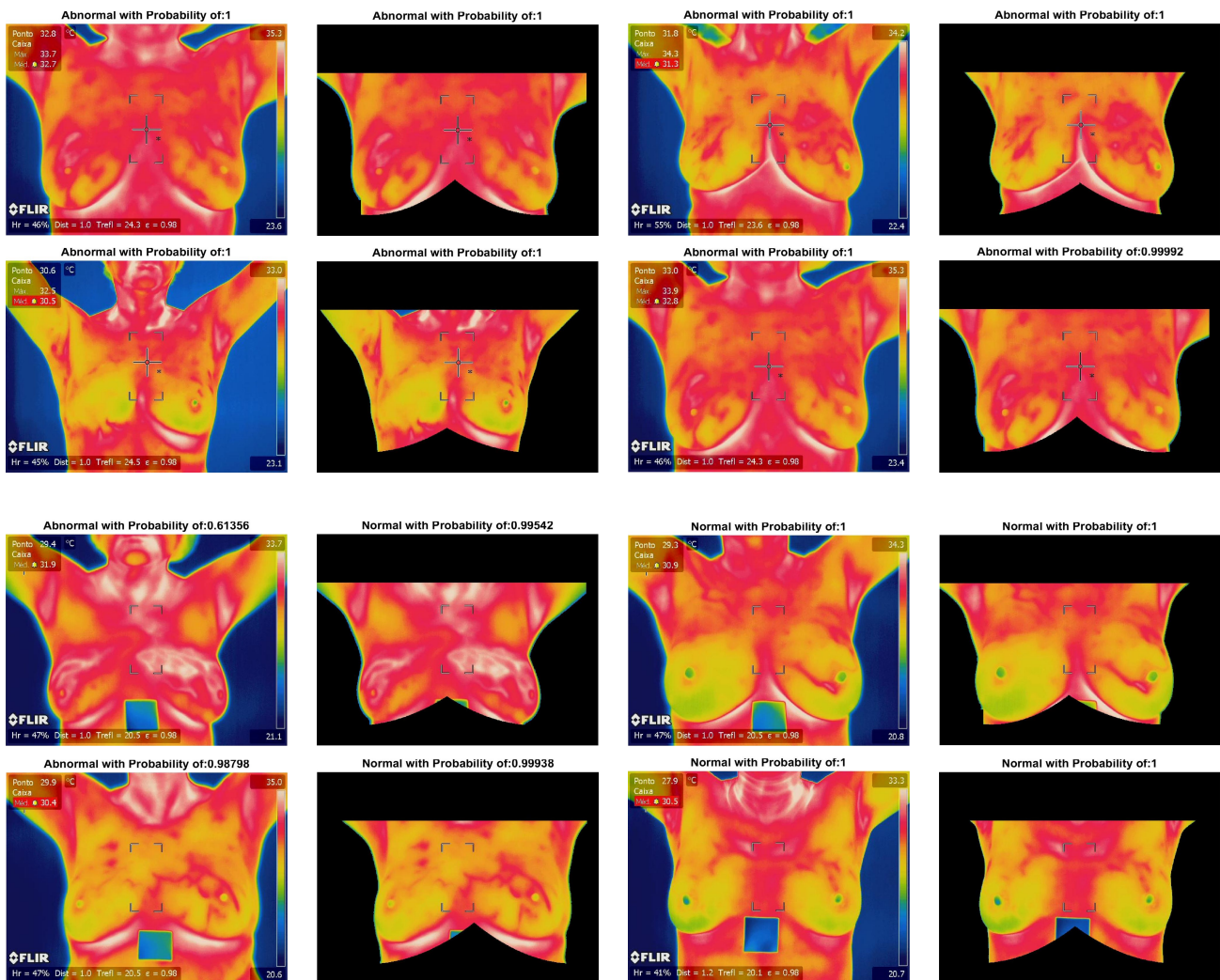


Figure 7. The classification result of (a) abnormal breast thermogram using raw and segmented dataset; (b) normal breast thermogram using raw and segmented dataset.

4.2. BreaCNet implementation framework

BreaCNet can be implemented as a mobile breast self-screening application as it costs only 6.1 million parameters and is 22 MB in size. The application will allow women to screen their breast condition independently. In this section, we propose a framework for the BreaCNet implementation for

a mobile application. As mentioned in Section 1, a regular breast self-screening tool should not depend on the internet connection. Moreover, it should keep the users' data private. Thus, it is necessary to locate the prediction task or inference on the local mobile device.

Inferencing on the local device will allow the prediction task to be executed using the mobile CPU. Users can capture their breasts using a thermal camera embedded in their smartphone and feed the image to the prediction model. The prediction result will appear in real time. However, the prediction result's accuracy may decrease as a result of feeding the indefinite images to the model prediction. BreaCNet was trained on a homogenous dataset produced by a specific thermal camera and a particular thermography protocol. Nevertheless, the App's users may use different thermal cameras to capture their breasts in various ways. Thus, continuous model monitoring is needed to maintain the model performance. Figure 8 shows the BreaCNet implementation framework. There are two parts of the framework: one part is for the application provider, whereas the other is for the application users. The description of the process involved is as follows.

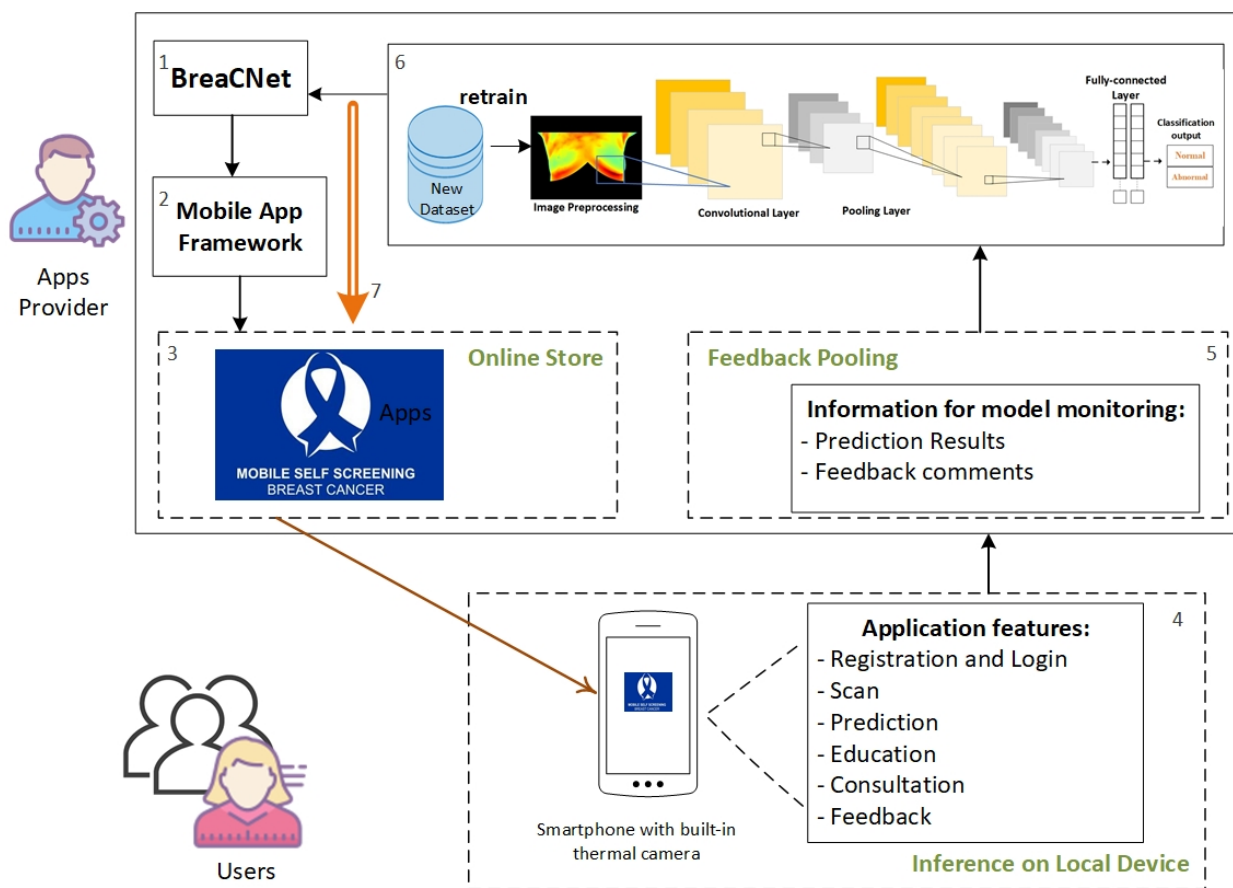


Figure 8. BreaCNet implementation framework: the illustration of steps involved in performing BreaCNet on the local device while monitoring the model.

Inferencing:

The prediction model of BreaCNet (1) is first optimized. Then, it is converted into a mobile framework. Among the platforms that can be used for the mobile application are CoreML, TensorFlow, Lite, and C#. Here, the converter takes the model and invokes the mobile formats to enable the on-device DL inference with low latency and small binary size.

The model is then deployed into an application (2). The challenge here is integrating the application programming interface (API) with the model. API enables interaction between data, applications, and devices. The integration and interaction method must be consistent across platforms. The model has to be bundled with the application code to allow smooth transfer to users. When deploying the model using cross-platforms, special attention is required to determine the target platforms and the possible devices.

Next, the application is provided for the users via online stores (3), such as the App Store and Play Store. As the inference is localized on the local device, the users do not need an internet connection to perform the prediction tasks. They can directly use the scan feature and obtain the prediction responses in real-time. Besides, using this app needs less attention span; thus, breast self-screening can be done regularly.

To protect the model, the encryption technique can be applied. By encrypting the weights and architecture or scrambling the model format and piecing it together at runtime, the predictive model can be kept black-boxed to end-users.

Application features:

On the users' side (4), there are various application features can be provided, such as "Registration and Login", "Scan", "Prediction", "Education", "Consultation" and "Feedback". The "Registration and Login" feature allows the users to get an independent identity to record their history and establish a connection with the "Consultation" feature function.

The "Scan" feature allows the users to capture their breasts using the built-in thermal camera in their smartphone. Then, they can load the thermograms into the system for the prediction tasks of breast screening. Next, the system will automatically execute the prediction task and generate real-time prediction results. The results will also be automatically sent to the server as a reference for model monitoring.

The "Education" feature provides educational information regarding breast cancer, thermography protocol, and recommendation based on the prediction results, whereas the "Consultation" feature is a service that connects users to social software products to enable communication with a medical expert. The "Feedback" feature permits users to send their comments regarding the application to the server. The application features can be extended for further needs.

Model monitoring:

The information on the prediction results and users' comments regarding the application will be pooled at the application provider's server (5). This information will be useful for the application provider to maintain the prediction accuracy rate and users' satisfaction. Maintenance is highly necessary for several reasons. First, the prediction accuracy rate may decrease due to various images being input into the model. Second, the mobile feature may need improvement to meet the users' needs. Third, other potential problems related to the application may exist, affecting the application's performance.

Occasionally, the model needs to be retrained (6). The dataset to retrain the model can be collected

from several sources, such as users who voluntarily share their breast thermograms to the server, related studies, and hospitals conducting thermography for breast cancer screening. As the training shows improved performance, the application has to be updated with a smooth transfer to the users (7).

5. Performance evaluation and discussion

We evaluated BreaCNet performance by observing the learning curve and testing result. Due to the use of DL for the classification task, there are two learning curves for each training and validation. One is the accuracy learning curve, calculated using the metric evaluation of accuracy. The second is the loss learning curve by which the parameters of the models are being optimized.

Figure 9 shows the training and validation accuracy learning curve. It demonstrates that in every validation, the accuracy rate is mostly higher than or similar to that of the training. This means that the learning is accurate. Meanwhile, the loss learning curve (Figure 10) demonstrates the learning loss decreasing to the point of stability and a minimal gap between the two final training and validation losses. The gap between the learning curves is referred to as the generalization curve, which is the model's ability to correctly adapt to new previously unseen data. Specifically, both validation and learning losses were low; thus, we confirmed BreaCNet has a good fit [62].

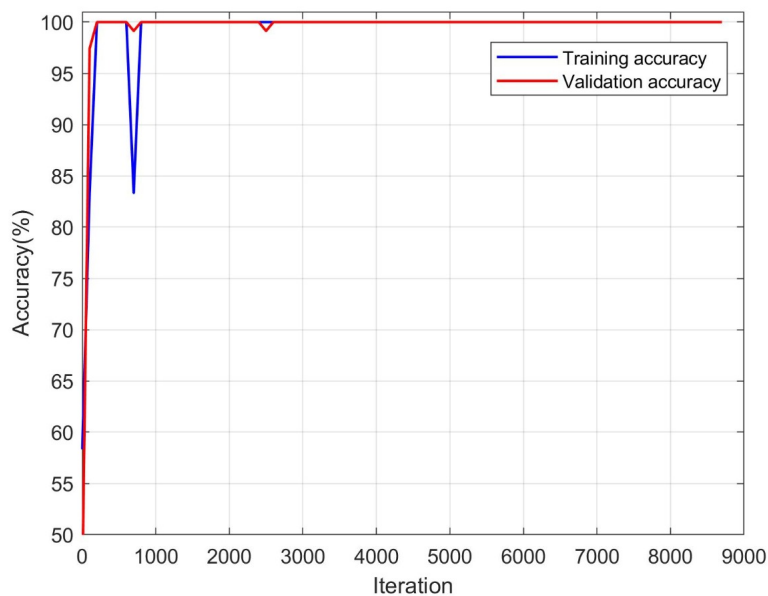


Figure 9. The accuracy learning curve of the proposed CNN model.

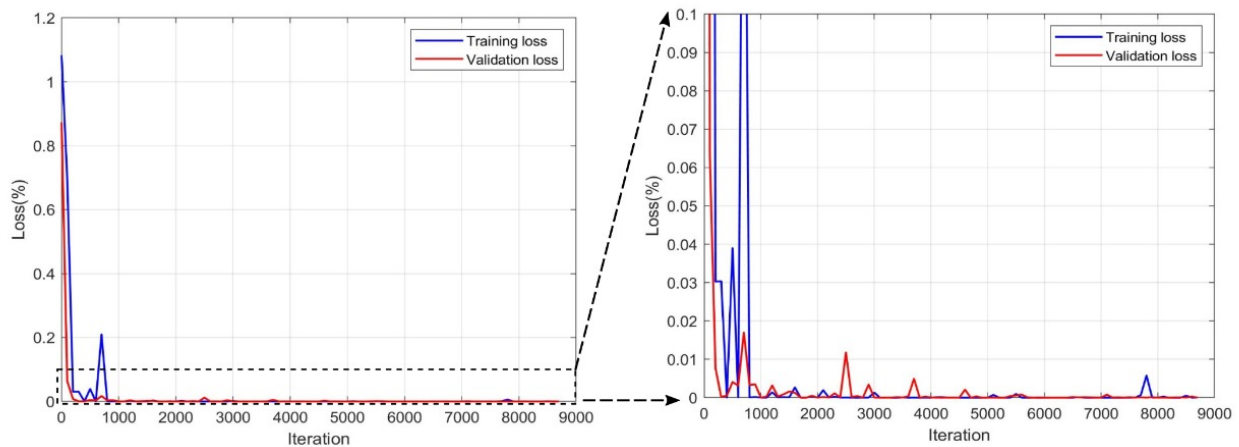


Figure 10. The loss learning curve of the proposed CNN model with its scale magnification.

We also observed feature learning using the raw and segmented datasets in the convolutional layer (Figure 11). The light and dark areas indicate positive and negative activations, respectively. Since the ReLU followed the convolutional layer, only the positive activations were used. Figure 11(a),(b) depict the feature mapping of the raw and segmented datasets, respectively, which reveals that the raw dataset causes more learning, whereas the segmented one activates only the important parts of the breast thermograms. Accordingly, feature learning becomes more effective.

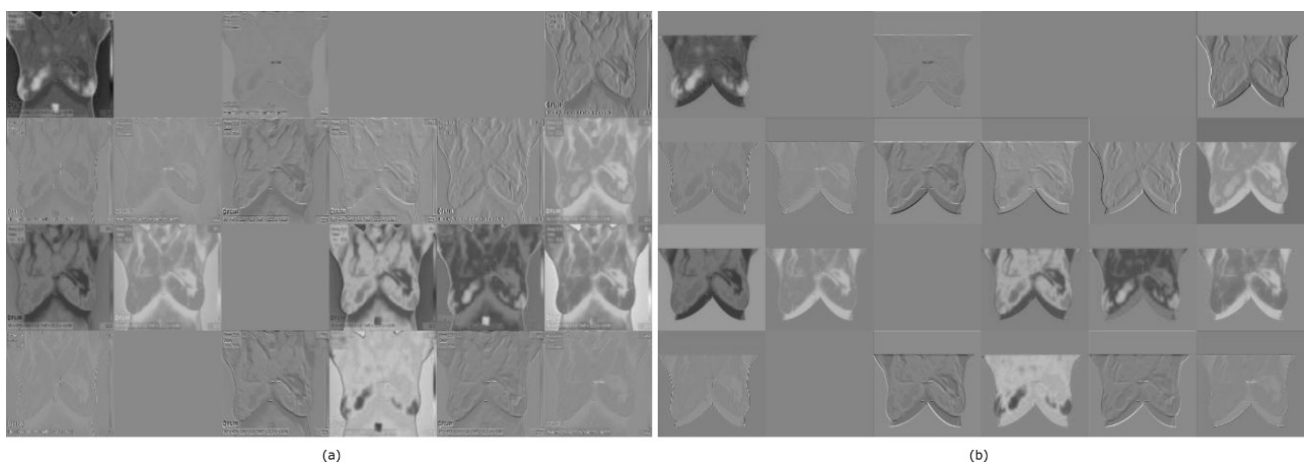


Figure 11. Comparison of feature mapping with (a) raw dataset and (b) segmented dataset.

BreaCNet, which consists of the proposed segmentation algorithm and modified ShuffleNet, has demonstrated the best performance, as presented with the confusion matrix in Figure 12. When the model was trained on the raw dataset, the accuracy was only 72% and 85% for raw and segmented

testing datasets, respectively. Similarly, when the model was trained on the segmented dataset, the accuracy and sensitivity significantly increased to 98% and 100% for raw and segmented testing datasets, respectively. It showed that the classification task supported by the enriched features of the input image performed better.

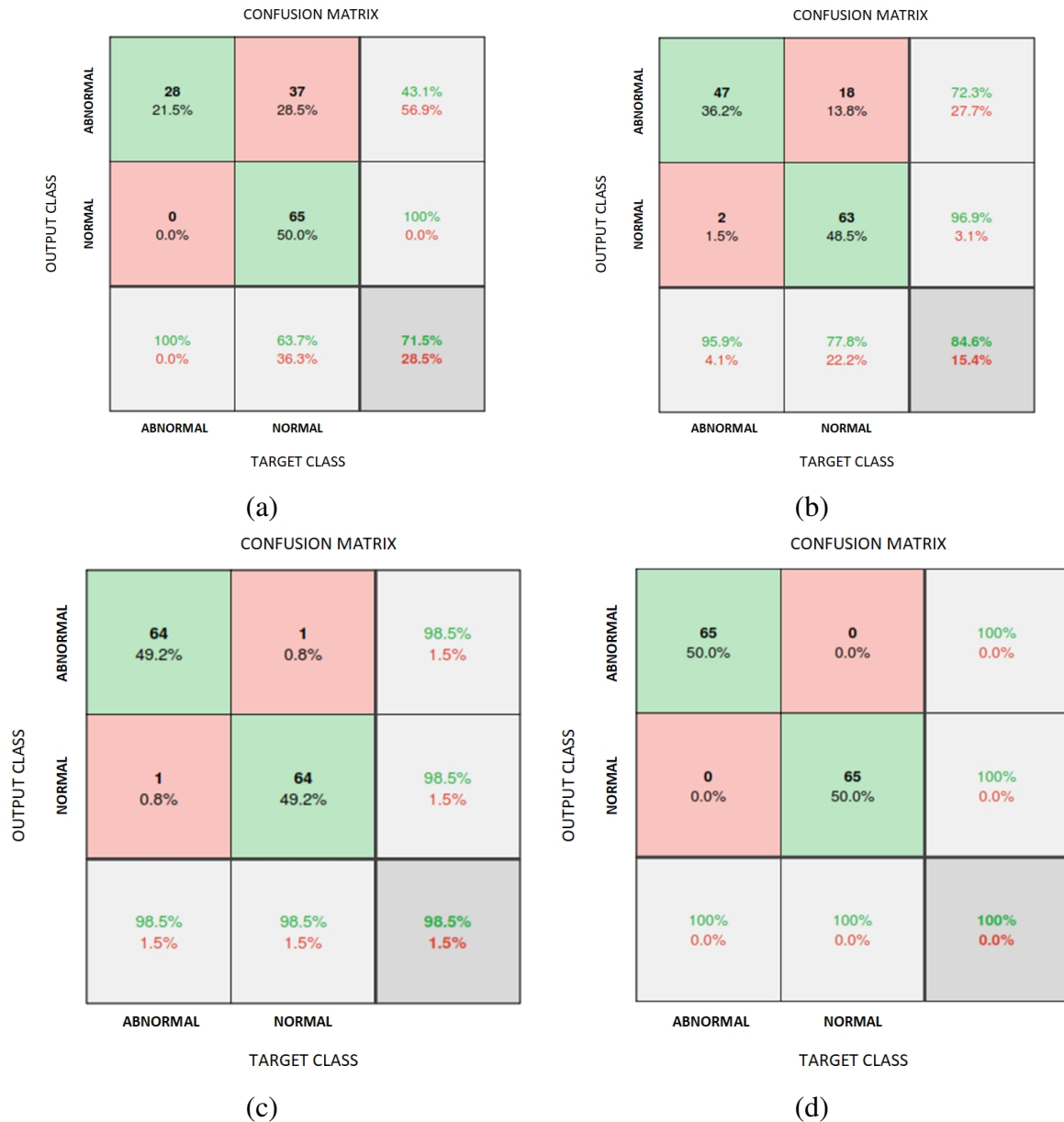


Figure 12. Confusion matrix of the proposed CNN model. (a) raw training and raw testing dataset; (b) raw training and segmented dataset; (c) segmented training and raw testing dataset; (d) segmented training and segmented testing dataset.

Comparison with similar works:

Our work is developing a breast thermal image classification model that begins with image process-

ing to facilitate the learning and improve the classification accuracy. As we refer to the related works in Table 1, Zuluaga-Gomez et al. [19] and Tello et al. [21] were also performed image preprocessing before training the CNN models. As we took the same approach, we compared various aspects of both works with BreaCNet in Table 5.

Table 5. Comparison of the proposed method with the similar works on breast thermogram segmentation and classification using CNN.

Works	Dataset	Segmentation algorithm	CNN model	Learning curve	Learning parameter	Model size	Accuracy
Zuluaga-Gomez [19]	57 patients; 19 healthy, 38 malignant; 50% training, 20% validation, 30% testing	not clearly described of grayscale mask and cropping method	proposed reduced convolutional layer (hand-crafted)	Unknown	Unknown	Unknown	92 %
Tello-Mijares et al. [21]	63 thermograms; 35 normal, 28 abnormal; the portion for training and testing was not mentioned	Gaussian filtering Curvature function k (cvt k) Gradient vector flow snake (GVF)	AlexNet	Unknown	Unknown	AlexNet 227 MB	100%
Proposed BreaCNet	154 patients; 121 healthy, 33 sick; 90% training, 10% testing	Gaussian filtering Sobel edge detector, Polynomial curve fitting	Modified ShuffleNet	Validation loss is lower than training loss	6.1 million	22 MB	100%

Zuluaga-Gomez et al. [19] demonstrated that data augmentation can increase the accuracy rate. Their proposed data augmentation generated horizontal and vertical flip, 0° – 45° rotation, 20% zoom, and noise normalization. The hyperparameter was defined using Bayesian optimization with a simple CNN structure. Their model achieved only an accuracy rate of 92%. Besides, the segmentation algorithm was not clearly described. The information about the learning curves and model size was also unknown.

Tello et al. [21] developed a segmentation technique and trained the CNN model of AlexNet to classify the breast thermograms into a binary class. Although they achieved 100% accuracy, the segmentation procedure was complex as it demanded numerous calculations to find the elliptic curvature

of the breast's ROI. Unfortunately, the learning curve of pre-trained model training was not presented and described; thus, the information about the model generalization was unknown. Moreover, the pre-trained model size of AlexNet was 227 MB in size, which was greater than the recent practical sizes of mobile applications that can be used by the industry [33, 34].

Meanwhile, BreaCNet demonstrated the best performance with an accuracy rate of 100%. The classifier performed less computation because the segmentation procedure was simple. Furthermore, the classification model was lightweight with 6.1 million parameters and 22 MB in size. It is worth noting that the segmentation algorithm has to be validated when applied to other breast thermogram datasets.

We also confirmed that using a segmented dataset as an input for training or testing can improve the performance of the classification task. Feeding only the informative features to the network model will enhance the feature learning performance and increase the accuracy rate.

Further, the increased accuracy rate as a result of filter addition clarifies that more filters enable more learning. As filters in CNNs function as feature detectors, more filters will trigger more detectors to learn the breast thermogram's complex feature better.

Finally, the model can be beneficial if it is integrated into a mobile application that is accessible at a low cost. The success of mobile self-screening also depends on the smartphone specification. Thus, we encourage the smartphone industry to produce mobile devices with adequate thermal cameras and computational ability. Hopefully, the availability of mobile self-screening for breast cancer will encourage all women to be aware of their breast's condition at the initial stage.

6. Conclusions

We built a classifier model, namely Breast Cancer mobile Network (BreaCNet), by integrating a proposed segmentation algorithm and a well-trained modified ShuffleNet model to classify breast thermograms into normal and abnormal binary classes. The segmentation algorithm was constructed using Sobel edge detection and the second-order polynomial curve fitting. The modified architecture of ShuffleNet was obtained by adding one convolutional layer with more filters and a dropout of 50% to reduce the parameter cost. We confirmed that feeding the segmented breast thermogram can improve the feature learning performance by more than 9%, and more filters enable more learning. The BreaCNet significantly increased the accuracy rate from 72% (using raw datasets) to 100% (using segmented dataset). Moreover, the BreaCNet learning curve showed a good fit with 6.1 million parameters and 22 MB in size. Thus, it has fulfilled the requirements of the on-device inference of a mobile application. For future work, the segmentation algorithm will be validated using other breast thermogram datasets to enable the application used for various breast thermal images specification. In addition, the model will be implemented as a mobile breast self-screening tool to support women's awareness of regular breast self-examination.

Acknowledgments

This work was supported in part by the Institute for Research and Community Service (LPPM) of Universitas Syiah Kuala, Indonesia, under the scheme of Penelitian Lektor Kepala 2020 funding, and by the Indonesian Ministry of Education, Culture, Research, and Technology (Kemdikbudristek) under

Penelitian Disertasi Doktor (PDD) grant 2021 and Universitas Syiah Kuala.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444. doi: 10.1038/nature14539.
2. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, et al., Dermatologist-level classification of skin cancer with deep neural networks, *Nature*, **542** (2017), 115–118. doi: 10.1038/nature21056.
3. P. Wang, X. Xiao, J. R. G. Brown, T. M. Berzin, M. Tu, F. Xiong, et al., Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy, *Nat. Biomed. Eng.*, **2** (2018), 741–748. doi: 10.1038/s41551-018-0301-3.
4. M. Hammad, A. M. Iliyasu, A. Subasi, E. S. L. Ho, A. A. A. El-Latif, A multitier deep learning model for arrhythmia detection, *IEEE Trans. Instrum. Meas.*, **1** (2021), 1–9. doi: 10.1109/TIM.2020.3033072.
5. J. G. Nam, S. Park, E. J. Hwang, J. H. Lee, K. N. Jin, K. Y. Lim, et al., Development and validation of deep learning-based automatic detection algorithm for malignant pulmonary nodules on chest radiographs, *Radiology*, **290** (2019), 218–228. doi: 10.1148/radiol.2018180237.
6. A. Sedik, A. M. Iliyasu, A. El-Rahiem, M. E. Abdel Samea, A. Abdel-Raheem, M. Hammad, et al., Deploying machine and deep learning models for efficient data-augmented detection of covid-19 infections, *Viruses*, **12** (2020), 769. doi: 10.3390/v12070769.
7. S. Xu, H. Wu, R. Bie, Cxnet-m1: Anomaly detection on chest x-rays with image-based deep learning, *IEEE Access*, **7** (2019), 4466–4477. doi: 10.1109/ACCESS.2018.2885997.
8. K. Munadi, K. Muchtar, N. Maulina, B. Pradhan, Image enhancement for tuberculosis detection using deep learning, *IEEE Access*, **8** (2020), 897–217. doi: 10.1109/ACCESS.2020.3041867.
9. H. Chougrad, H. Zouaki, O. Alheyane, Deep convolutional neural networks for breast cancer screening, *Comput. Methods Programs Biomed.*, **157** (2018), 19–30. doi: 10.1016/j.cmpb.2018.01.011.
10. M. A. Al-Masni, M. A. Al-Antari, J. M. Park, G. Gi, T. Y. Kim, P. Rivera, et al., Simultaneous detection and classification of breast masses in digital mammograms via a deep learning yolo-based cad system, *Comput. Methods Programs Biomed.*, **157** (2018), 85–94. doi: 10.1016/j.cmpb.2018.01.017.
11. H. Li, J. Weng, Y. Shi, W. Gu, Y. Mao, Y. Wang, et al., An improved deep learning approach for detection of thyroid papillary cancer in ultrasound images, *Sci. Rep.*, **8** (2018), 1–12. doi: 10.1038/s41598-018-25005-7.

12. H. K. Mewada, A. V. Patel, M. Hassaballah, M. H. Alkinani, K. Mahant, Spectral-spatial features integrated convolution neural network for breast cancer classification, *Sensors*, **20** (2020), 4747. doi: 10.3390/s20174747.
13. R. Yan, F. Ren, Z. Wang, L. Wang, T. Zhang, Y. Liu, et al., Breast cancer histopathological image classification using a hybrid deep neural network, *Methods*, **173** (2020), 52–60. doi: 10.1016/j.ymeth.2019.06.014.
14. A. Rakhlin, A. Shvets, V. Iglovikov, A. A. Kalinin, Deep convolutional neural networks for breast cancer histology image analysis, in *International Conference Image Analysis and Recognition*, Springer, (2018), 737–744.
15. D. Bardou, K. Zhang, S. M. Ahmad, Classification of breast cancer based on histology images using convolutional neural networks, *IEEE Access*, **6** (2018), 24680–24693. doi: 10.1109/ACCESS.2018.2831280.
16. D. M. Vo, N. Q. Nguyen, S. W. Lee, Classification of breast cancer histology images using incremental boosting convolution networks, *Inf. Sci.*, **482** (2019), 123–138. doi: 10.1016/j.ins.2018.12.089.
17. R. Roslidar, K. Saddami, F. Arnia, M. Syukri, K. Munadi, A study of fine-tuning CNN models based on thermal imaging for breast cancer classification, in *2019 IEEE International Conference on Cybernetics and Computational Intelligence*, (2019), 77–81.
18. F. J. Fernández-Ovies, E. S. Alférez-Baquero, E. J. de Andrés-Galiana, A. Cernea, Z. Fernández-Muñiz, J. L. Fernández-Martínez, Detection of breast cancer using infrared thermography and deep neural networks, in *International Work-Conference on Bioinformatics and Biomedical Engineering*, Springer, (2019), 514–523.
19. J. Zuluaga-Gomez, Z. Al Masry, K. Benaggoune, S. Meraghni, N. Zerhouni, A cnn-based methodology for breast cancer diagnosis using thermal images, *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, **9** (2021), 1–15. doi: 10.1080/21681163.2020.1824685.
20. J. C. Torres-Galván, E. Guevara, F. J. González, Comparison of deep learning architectures for pre-screening of breast cancer thermograms, in *2019 Photonics North*. IEEE, (2019), 1–2.
21. S. Tello-Mijares, F. Woo, F. Flores, Breast cancer identification via thermography image segmentation with a gradient vector flow and a convolutional neural network, *J. Healthc. Eng.*, **2019** (2019), 1–13. doi: 10.1155/2019/9807619.
22. R. Sánchez-Cauce, J. Pérez-Martín, M. Luque, Multi-input convolutional neural network for breast cancer detection using thermal images and clinical data, *Comput. Methods Programs Biomed.*, **204** (2021), 106045. doi: 10.1016/j.cmpb.2021.106045.
23. R. Roslidar, A. Rahman, R. Muharar, M. R. Syahputra, F. Arnia, M. Syukri, et al., A review on recent progress in thermal imaging and deep learning approaches for breast cancer detection, *IEEE Access*, **8** (2020), 116176–116194. doi: 10.1109/ACCESS.2020.3004056.
24. B. O. Anderson, S. Braun, S. Lim, R. A. Smith, S. Taplin, D. B. Thomas, et al., Early detection of breast cancer in countries with limited resources, *Breast J.*, **9** (2003), S51–S59. doi: 10.1046/j.1524-4741.9.s2.4.x.

25. R. Sankaranarayanan, K. Ramadas, S. Thara, R. Muwonge, J. Prabhakar, P. Augustine, et al., Clinical breast examination: preliminary results from a cluster randomized controlled trial in india, *J. Natl. Cancer Inst.*, **103** (2011), 1476–1480. doi: 10.1093/jnci/djr304.
26. World Health Organization, *Breast cancer: prevention and control*, (2019). Available from: <https://www.who.int/cancer/detection/breastcancer/en>.
27. Cisco, *Cisco annual internet report (2018–2023)*, Report, (2020). Available from: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
28. Caterpillar, *Integrated thermal imaging*, (2020). Available from: <https://www.catphones.com/en-dk/features/integrated-thermal-imaging>.
29. FLIR, *Blackview bv9800 pro featuring flir lepton thermal camera available now*, (2020). Available from: <https://www.flir.com/news-center/press-releases/blackview-bv9800-pro-featuring-flir-lepton-thermal-camera-available-now>.
30. Teledyne Fire, *Flir one gen 3*, 2020. Available from: <https://www.flir.com/products/flir-one-gen-3>.
31. J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, X. Zhu, Deep learning towards mobile applications, in *Proceeding of 2018 IEEE 38th International Conference on Distributed Computing Systems*, (2018), 1385–1393.
32. Roslidar, M. K. Muchamad, F. Arnia, M. Syukri, K. Munadi, A conceptual framework of deploying a trained cnn model for mobile breast self-screening, in *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, (2021), 533–537.
33. A. Koul, S. Ganju, M. Kasam, *Practical deep learning for cloud, mobile, and ddge: Real-world AI & computer-vision projects using python, keras & tensorflow*, O'Reilly Media, 2019.
34. *Apk expansion files*, 2020. Available from: <https://developer.android.com/google/play/expansion-files>.
35. L. Silva, D. Saade, G. Sequeiros, A. Silva, A. Paiva, R. Bravo, et al., A new database for breast research with infrared image, *J. Med. Imaging Health Inform.*, **4** (2014), 92–100. doi: 10.1166/jmihi.2014.1226.
36. T. B. Borchardt, A. Conci, R. C. Lima, R. Resmini, A. Sanchez, Breast thermography from an image processing viewpoint: A survey, *Signal Process.*, **93** (2013), 2785–2803. doi: 10.1016/j.sigpro.2012.08.012.
37. Y. Zhou, S. Chen, Y. Wang, W. Huan, Review of research on lightweight convolutional neural networks, in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference*, (2020), 1713–1720.
38. A. S. Winoto, M. Kristianus, C. Premachandra, Small and slim deep convolutional neural network for mobile device, *IEEE Access*, **8** (2020), 125210–125222. doi: 10.1109/ACCESS.2020.3005161.
39. S. B. Shuvo, S. N. Ali, S. I. Swapnil, T. Hasan, M. I. H. Bhuiyan, A lightweight cnn model for detecting respiratory diseases from lung auscultation sounds using emd-cwt-based hybrid scalogram, *IEEE J. Biomed. Health Inform.*, **2020** (2020).

40. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2018), 4510–4520.
41. F. Chollet, Xception: Deep learning with depthwise separable convolutions, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2017), 1251–1258.
42. 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), 770–778.
43. X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2018), 6848–6856.
44. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, preprint, arXiv:1704.04861.
45. MathWorks, *Pretrained deep neural networks*, 2020. Available from: <https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html>.
46. N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, et al., Convolutional neural networks for medical image analysis: Full training or fine tuning?, *IEEE Trans. Med. Imaging*, **35** (2016), 1299–1312. doi: 10.1109/TMI.2016.2535302.
47. M. A. Garduño-Ramón, S. G. Vega-Mancilla, L. A. Morales-Henández, R. A. Osornio-Rios, Supportive noninvasive tool for the diagnosis of breast cancer using a thermographic camera as sensor, *Sensors*, **17** (2017), 497. doi: 10.3390/s17030497.
48. J. Cho, K. Lee, E. Shin, G. Choy, S. Do, How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?, preprint, arXiv preprint arXiv:1511.06348.
49. X. H. Zhou, D. K. McClish, N. A. Obuchowski, *Statistical methods in diagnostic medicine*, John Wiley & Sons, (2009).
50. Q. Zhou, Z. Li, J. K. Aggarwal, Boundary extraction in thermal images by edge map, in *Proceedings of the 2004 ACM Symposium on Applied Computing*, (2004), 254–258.
51. G. Gui, K. Behranwala, N. Abdullah, J. Seet, P. Osin, A. Nerurkar, et al., The inframammary fold: contents, clinical significance and implications for immediate breast reconstruction, *Br. J. Plast. Surg.*, **57** (2004), 146–149. doi: 10.1016/j.bjps.2003.11.030.
52. D. Sathish, S. Kamath, K. Prasad, R. Kadavigere, R. J. Martis, Asymmetry analysis of breast thermograms using automated segmentation and texture features, *Signal Image and Video Process.*, **11** (2016), 745–752. doi: 10.1007/s11760-016-1018-y.
53. R. P. Canale, S. C. Chapra, *Numerical Methods for Engineers with Personal Computer Applications*, McGraw-Hill, 2000.
54. R. P. dos Santos, G. S. Clemente, T. I. Ren, G. D. Cavalcanti, Text line segmentation based on morphology and histogram projection, in *2009 10th International Conference on Document Analysis and Recognition*, IEEE, **11** (2009), 651–655. doi: 10.1109/ICDAR.2009.183.
55. A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. doi: 10.1145/3065386.

56. S. Khan, H. Rahmani, S. A. A. Shah, M. Bennamoun, A guide to convolutional neural networks for computer vision, *Synth. Lect. Comput. Vision*, **8** (2018), 1–207. doi: 10.2200/S00822ED1V01Y201712COV015.
57. A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, O’Reilly Media, (2019).
58. V. Nair, G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, ICML, (2010).
59. D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, in *International conference on artificial neural networks*, Springer, (2010), 92–101.
60. N. Qian, On the momentum term in gradient descent learning algorithms, *Neural networks*, **12** (1999), 145–151. doi: 10.1016/S0893-6080(98)00116-6.
61. S. C. Kothari, H. Oh, *Neural networks for pattern recognition*, Elsevier, (1993), 119–166.
62. I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, Massachusetts: The MIT Press, 2016.



AIMS Press

© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)