Computer Methods and Programs in Biomedicine 157 (2018) 19–30

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Deep Convolutional Neural Networks for breast cancer screening

Hiba Chougrad[a,*], Hamid Zouaki[a], Omar Alheyane[b]

[a] Laboratory of Computer Science and Mathematics and their Applications (LIMA), Faculty of science, University Chouaib Doukkali, El Jadida 24000, Morocco
[b] Laboratory of Fundamental Mathematics (LMF), Faculty of science, University Chouaib Doukkali, El Jadida 24000, Morocco

ARTICLE INFO

Article history:
Received 10 February 2017
Revised 24 December 2017
Accepted 10 January 2018

Keywords:
Deep learning
Convolutional Neural Network
Transfer learning
Computer-aided Diagnosis
Breast cancer
Breast mass lesion classification

ABSTRACT

Background and objective: Radiologists often have a hard time classifying mammography mass lesions which leads to unnecessary breast biopsies to remove suspicions and this ends up adding exorbitant expenses to an already burdened patient and health care system.

Methods: In this paper we developed a Computer-aided Diagnosis (CAD) system based on deep Convolutional Neural Networks (CNN) that aims to help the radiologist classify mammography mass lesions. Deep learning usually requires large datasets to train networks of a certain depth from scratch. Transfer learning is an effective method to deal with relatively small datasets as in the case of medical images, although it can be tricky as we can easily start overfitting.

Results: In this work, we explore the importance of transfer learning and we experimentally determine the best fine-tuning strategy to adopt when training a CNN model. We were able to successfully fine-tune some of the recent, most powerful CNNs and achieved better results compared to other state-of-the-art methods which classified the same public datasets. For instance we achieved 97.35% accuracy and 0.98 AUC on the DDSM database, 95.50% accuracy and 0.97 AUC on the INbreast database and 96.67% accuracy and 0.96 AUC on the BCDR database. Furthermore, after pre-processing and normalizing all the extracted Regions of Interest (ROIs) from the full mammograms, we merged all the datasets to build one large set of images and used it to fine-tune our CNNs. The CNN model which achieved the best results, a 98.94% accuracy, was used as a baseline to build the Breast Cancer Screening Framework. To evaluate the proposed CAD system and its efficiency to classify new images, we tested it on an independent database (MIAS) and got 98.23% accuracy and 0.99 AUC.

Conclusion: The results obtained demonstrate that the proposed framework is performant and can indeed be used to predict if the mass lesions are benign or malignant.

1. Introduction

Breast cancer is one of the most common invasive diseases among women worldwide. In 2016, there were more than 2.8 million women with a history of breast cancer in the U.S and this includes women currently being treated and women who have finished treatment. In 2017, 1,688,780 new cases of breast cancer are expected to be diagnosed and 600,920 cancer deaths are projected to occur, though death rates have been decreasing since 1989. These decreases are thought to be the result of treatment advances, increased awareness and earlier detection through screening [1]. Mammography is the recommended imaging modality for breast cancer screening [2], it is more useful as an early detection tool before the appearance of the physical symptoms. Early diagnosis of the disease via mammography screening increases the chances of recovery dramatically [2]. However, the accuracy of the diagnosis can be affected by the image quality or the radiologist's expertise prone to errors. The average error rate among radiologists is around 30%, according to some studies [3,4]. In some recent surveys [5], error in diagnosis was the most common cause of litigation against radiologists. The majority of such cases arose from failure to diagnose breast cancer on mammography [5]. To reduce the rate of false-negative diagnoses, lesions with a 2% chance being malignant are recommended for a biopsy [6]. However, only 15–30% of the biopsies are found to be malignant [6]. As a result, the unnecessary biopsies end up costing so much in terms of time, money or even discomforts that can occur for some patients due to anxiety or panic attacks. It is therefore substantial to improve the accuracy of the radiologic diagnosis to increase the positive predictive value of mammography.

Computer-aided Diagnosis (CAD) systems aim at giving a second objective opinion to assist the radiologist medical image in-

[*] Corresponding author.
E-mail address: chougrad.h@ucd.ac.ma (H. Chougrad).

https://doi.org/10.1016/j.cmpb.2018.01.011
0169-2607/© 2018 Elsevier B.V. All rights reserved.

terpretation and diagnosis. CAD systems are especially used as applications that perform the labeling or differentiation between benign and malignant lesions. In the last few years, deep learning [7–10] especially through Convolutional Neural Networks (CNNs) [11] has been proved to work very well in vision tasks. Some of the recently proposed CAD systems adopted the renowned deep learning techniques and obtained promising results [12–14]. The deep learning CADs were introduced to different medical domains, for example we can mention pulmonary Peri-fissural nodule classification [12] or Interstitial lung disease and Thoraco-abdominal lymph node classification [14] and many others. We were particularly interested in the works on breast lesions [15–18]. Most of the proposed methods involved CNNs but in a traditional way, where they use only the extracted CNN features or combine them with some other hand-crafted descriptors to carry out the classification task [17,18]. However, the most interesting aspect of using CNNs is the end-to-end supervised learning process which does not rely on complex engineered descriptors and instead uses the whole raw image [11].

Convolutional Neural Networks learn discriminative features automatically, their architecture is particularly adapted to take advantage of the 2D structure of the input image, but more importantly one of their most impressive characteristic is that they generalize surprisingly well to other recognition tasks [14,16]. In order to train deep CNNs we need large annotated datasets which are lacking in the medical domain especially for breast cancer. Moreover, training a CNN from scratch requires high computational power, large memory resources and time, and with the little data provided we can easily start overfitting. One way to overcome this is to use transfer learning [19] from natural images (for example ImageNet which has more than 1.2 million images categorized under 1000 classes) and perform a fine-tuning as proposed in [14].

Transfer learning is commonly used in deep learning applications. It has been very effective in the medical domain [13,14] when the amount of data is normally limited. Using transfer learning from natural images to breast cancer mammography images, has not yet been fully explored in the literature. And, as far as we know the only work [16] which uses transfer learning to classify breast lesions, employs small sized datasets and the deep Convolutional Neural Network CNN-F [20] as a model. We propose to perform the learning on different other datasets using some of the recent, well-engineered and deepest CNN architectures.

In this paper we exploit three of the most impressive CNN models recently proposed VGG16, ResNet50 and Inception v3 [21–23] trained on ImageNet [24]. We investigate the importance of transfer learning instead of random initialization for each model, and explore the impact of the number of fine-tuned layers on the final results. Our primary aim is to make use of these state-of-the-art CNNs, and perform a transfer learning from natural images to mammography images, in order to build a powerful mass lesion classification tool which can assist the radiologist by giving him a “second-opinion” and help him make more accurate diagnoses.

The remainder of this paper is organized as follows: Section 2 describes the proposed approach to classify breast abnormalities into benign or malignant, in Section 3 we give details of the experimentations lead to evaluate the proposed approach and we show the results. Section 4 gives a brief discussion and finally Section 5 concludes the paper.

2. Materials and methods

2.1. Datasets

We used three public databases to perform the learning and to benchmark the deep learning architectures. We then merged all three datasets to boost the learning process, as it is known that deep learning and other modern nonlinear machine learning techniques get better with more data [25]. We will refer to this assembled database as the (Merged Dataset) MD dataset.

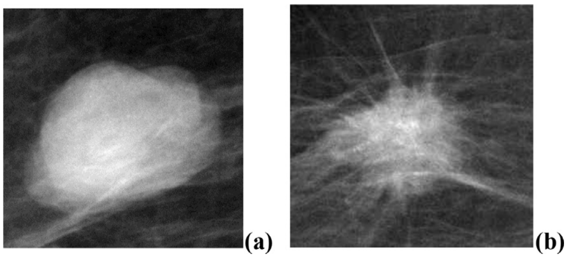


Fig. 1. Samples of mammography mass lesions; (a) benign; (b) malignant.

DDSM [26] the Digital Database of Screening Mammography (DDSM) is publicly available and contains more than 2600 cases, sorted according to the degree of severity of the findings. Every case in the DDSM contains four view mammograms from the same patient with associated ground truth and other information. For example the patient’s age, the screening exam’s date, the ACR (i.e. American College of Radiology reporting system) breast density that was specified by an expert radiologist. Cases containing suspicious regions are associated with a pixel level ground truth markings of the abnormalities. We were particularly interested in the benign versus the malignant cases, so we used a subset of 1329 cases containing a total of 5316 images, 641 cases of patients with benign mass lesions and 688 cases of patients with malignant mass lesions.

BCDR [27] the BCDR-F03 dataset from the Breast Cancer Digital Repository is a new dataset of film mammography composed by 736 biopsy-proven lesions of 344 patients. Each case includes clinical data for each patient, and both Cranio-caudal (CC) and Mediolateral oblique (MLO) view mammograms, which are available together with the coordinates of the lesion’s contours. BCDR-F03 is a binary class dataset composed of benign and malignant findings. We used a subset of 600 images from 300 patients, 300 images from 150 benign cases and 300 of images from 150 malignant cases.

INbreast [28] the INbreast database was built with full-field digital mammograms (in opposition to digitized mammograms); it is made publicly available together with precise annotations. It has a total of 115 cases including MLO and CC views from each breast yielding a total of 410 images. The database provides information regarding the patient’s age at the time of image acquisition, family history, ACR breast density annotation along with accurate contours of the findings that were made by specialists. The INbreast database presents a wide variability of cases which includes several types of lesions (masses, calcifications, asymmetries, and distortions). We were interested in the benign and malignant cases, so we assembled a subset of 50 cases including a total of 200 images, 100 images from 25 benign cases and 100 from malignant cases.

The Merged Dataset (MD) since deep CNNs perform better when used with large datasets, we combined all of the previous datasets (i.e. DDSM, BCDR, INbreast) to build a new big dataset to train our CNN models. To get a balanced dataset, we pre-processed and normalized all the previously mentioned datasets, as we will explain in the next section, so that we could obtain one big homogenous database of targeted regions of interest from both cases of patients with benign and malignant lesions (Fig. 1). We obtained a total of 6116 images from 1529 cases.

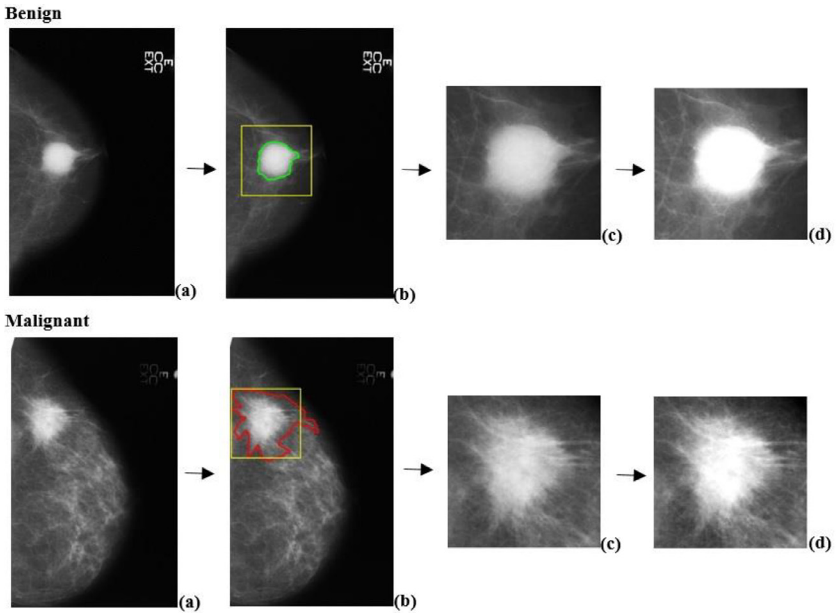


Fig. 2. Pre-processing of the mammograms; first row of the figure gives the example of a benign lesion and the second row a malignant lesion. (a) the original mammogram (b) illustration of the location and boundaries of the lesions annotated by imaging specialists (c) the cropped region of interest (d) the normalized ROI after applying Global Contrast Normalization. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Methodology

2.2.1. Pre-processing

In order to enhance the performance of a CAD system, pre-processing is a mandatory step when building the dataset. In our work, we extracted the lesions as fixed sized ROIs then we normalized them using global contrast normalization. Fig. 2 illustrates the pre-processing steps for extracting and normalizing the ROIs.

(1) **ROI extraction:** We used the ground-truth provided with each dataset to detect and crop the regions of interest (ROIs) from the images. The location and boundaries of the lesions were marked by imaging specialists. We used the provided coordinates to target and crop the bounding box of the lesions automatically. We fixed the input size ROIs to r x r pixels. We then rescaled the output image, for when the lesion is near the edges and the width or height of the cropped ROI is smaller than r.

(2) **Global contrast normalization (GCN):** Normalization, also called zero-centering is a standard step in medical image classification. It attempts to deal with external sources of data variation like illumination levels, the different scanners used in the digitalization process and how this can affect the pixel values. Global contrast normalization computes the mean of intensities for each image, and then subtracts it from each pixel of the image. Let $x_{i,j}$ be the tensor of an image ($x \in \mathbb{R}^{r\ x\ r}$) and $\bar{x}$ is the mean of the $x_{i,j}$ image intensities $\bar{x} = \frac{1}{r^2}\sum_{i,j} x_{i,j}$.

The tensor of the normalized image is $x'_{i,j} = x_{i,j} - \bar{x}$.

(3) **Data augmentation**: Deep learning models perform better when we have large datasets [25]. One very popular way to make our datasets bigger is data augmentation or jittering. Data augmentation can increase the size of the dataset to 10 times the original one or more, which helps prevent overfitting when training on very little data. The approach helps build simpler and robust models which can generalize better [29]. To perform data augmentation, the simplest way is to add noise or apply geometric transformation to existing data. Applying noise and transformations to images of lesions makes sense since this kind of data is very likely to be affected by all sort of noise and can be found in different sizes and orientations. Hence, all the transformations would boost the models to learn better.

The images were "augmented" using a series of random transformations so that the models would never see twice the exact same image. We used width and height shifts with a fraction of 0.25 from the total width or height of the image, a random rotation range of 0–40°, a shear range of 0.5 and a zoom range between [0.5–1.5]. We also flipped the images horizontally and applied the "fill mode" strategy for filling in newly created pixels, which can appear after a rotation or a width/height shift. To carry out the augmentation, we instantiated the *Keras* [30] *ImageDataGenerator* to generate batches of tensor image data with real-time data augmentation.

2.2.2. Deep Convolutional Neural Networks models

In the last few years, CNNs demonstrated impressive performance as they grew deeper and deeper; with the state-of-the-art networks going from 7 layers to 1000 layers. In this paper, we use some of these state-of-art architectures, pre-trained on ImageNet, for transfer learning from natural images to breast cancer images.

VGG16: There are several versions to the very deep convolutional network (VGG) [21] published by researchers from Oxford University, VGG16 is one of their best networks and is well known for its simplicity.

The architecture of this network is deep and simple, it mainly consists of an alternation between convolution layers and dropout layers. VGG was the first to use multiple small 3×3 filters in each convolutional layer and combine them in a sequence to emulate the effect of larger receptive fields. Although the network is simple in its architecture, it is very expensive in terms of memory and computational cost since the exponentially increasing kernels lead to higher computational time and a bigger size model.

The implemented VGG16 architecture is composed of 13 convolutional layers, 5 pooling layers and achieves 9.9% top-5 error on ImageNet.

Table 1
Total number of layers of each model including the fully-connected layers added (5 dense layers added to each model). As well as the number of convolutional layers fine-tuned for the different fine-tuning strategies adopted. Note that, a dense layer or convolutional layer followed by a non-linearity is counted as one layer.

Model	Total number of layers	Last 1 convolutional block	Last 2 convolutional blocks	Last 3 convolutional blocks
VGG16	23	4	8	12
ResNet50	179	12	22	34
Inceptionv3	221	25	44	58

ResNet50: The ResNet50 is one of the models proposed in the deep residual learning for image recognition [22] by the Microsoft research team .The authors came up with a simple and elegant idea. They take a standard deep CNN and add shortcut connections that bypass few convolutional layers at a time. The shortcut connections create residual blocks, where the output of the convolutional layers is added to the block's input tensor. For instance, the ResNet50 model is composed of 50 layers of similar blocks with shortcut connections. These connections keep the computation low and at the same time provide rich combination features.

The ResNet50 model used has one convolutional layer followed by a batch normalization layer, and has two pooling layers in between which there is a total of 16 residual modules. Two kinds of residual modules are alternated, one that has 4 convolutional layers and another with 3 convolutional layers and each convolutional layer is followed by batch normalization. The residual block with 4 convolutional layers is the first one used, followed by at least two or more residual blocks with 3 convolutional layers and so on. The implemented ResNet50 model achieves a 7.8% top-5 error on ImageNet.

Inception v3: The Google research team with Christian Szegedy were mainly focused on reducing the computational burden of CNNs while maintaining the same level of performance. They introduced a new module named "The inception module" which, for the most part, can be described as a 4 parallel pathways of 1×1, 3×3 and 5×5 convolution filters. And because of the parallel network implementation, in addition to the down sampling layers in each block, the model's execution time beats VGG or ResNet.

The research team proposed many models over the years which are more and more complex, the Inception v3 [23] model was introduced at about the same time as ResNet. The network was built with some new designing principles for example the use of 3×3 convolutions instead of 5×5 or 7×7 in the inception modules, also the expansion of width at each layer to increase the combination of features for the next layer, as well as the aim of constructing a network with a computational budget balanced between its depth and width.

The Inception v3 we implemented has 5 convolutional layers each one followed by a batch normalization layer, 2 pooling layers and 11 inception modules. The inception modules used contain different numbers of paths and convolution layers. Authors of the Inception v3 did not define an "Inception cell" and then repeatedly applied it to downscale the input. Therefore, the inception modules used, sometimes consist of 4, 6, 7, 9 or 10 convolutional layers followed by batch normalization and one pooling layer. The implemented model by Chollet [30] achieves 7.8% as a top-5 error on ImageNet, same as ResNet50.

2.2.3. Transfer learning and fine-tuning

As we begin to explore deep learning models from more specialized domains as the quantity of available data gets scarce. Even though we have impressive training methods nowadays, training deep learning models on small quantities of data is very difficult. The actual paradigm used to deal with this issue has come through the use of pre-trained neural networks [19].

Authors in [31] demonstrated that transfer of knowledge in networks could be achieved by first training a neural network on a domain for which there is a large amount of data, and then retraining that network on a related but different domain via fine-tuning its weights. [13,14] were able to show that transfer learning can be beneficial even between two unrelated domains (natural vs. medical). The advantage of using pre-trained models extends beyond the limited data issue, where it was proven to be an effective initialization technique for many complex models [32,33]

We propose to investigate the adequacy of this technique for our case of study, either to deal with our little data or as a way for initializing the models. Fig. 3 gives the schema for the models setup. The CNN models are built differently but the same procedure is applied to all of them:

- For starters we kept the original networks architectures up till the fully-connected layers.
- The original fully-connected layers were built for the ImageNet dataset with 1000 outputs for 1000 class categories. We removed these last fully-connected layers and built our own fully-connected model, on top of the convolutional part of the models, suited to our number of classes (i.e. 2 classes "Benign" and "Malignant").

The new customized models (VGG16, ResNet50, Inception v3) will be used to train on the different datasets while adopting different training strategies. We first conduct an ablation study where we initialize our models randomly. Then, we use the models as fixed feature extractors and use the Softmax layer on top as a classifier. Finally, we adopt a fine-tuning strategy and study the impact of the fraction fine-tuned on the final results (Fig. 3).

The first convolutional layers of a CNN learn generic features and can perform more like edge detectors, which should be useful to many tasks, but the following layers become progressively more specific to the details of the classes contained in the dataset [34]. In accordance with this statement and since mammographic mass lesion images are very different from ImageNet images, we propose to fine-tune our models to adjust the features of the last convolutional blocks and make them more data-specific; we fine-tune the weights of the pre-trained networks using the new set of images by resuming the backpropagation on the unfrozen layers.

We propose a detailed study on the impact of the chosen fraction of convolutional layers (unfrozen layers) to fine-tune, on the final results in the experimental section. Table 1 gives the number of layers of each model and the number of layers we choose to fine-tune, while the rest of the model is frozen for the different fine-tuning strategies adopted (1 block, 2 blocks, 3 blocks and all the blocks). Since the models are very different a convolutional block varies from one model to another as shown in Fig. 3. For VGG16 a convolutional block contains 3 convolutional operations followed by an activation and a pooling layer. In the case of ResNet50 the convolutional block is a residual block while for the inception v3 it is an inception module.

2.2.4. Regularization and the choice of hyper-parameters

Choosing the right parameters when fine-tuning is tricky. The optimization is done using Stochastic Gradient Descent (SGD)

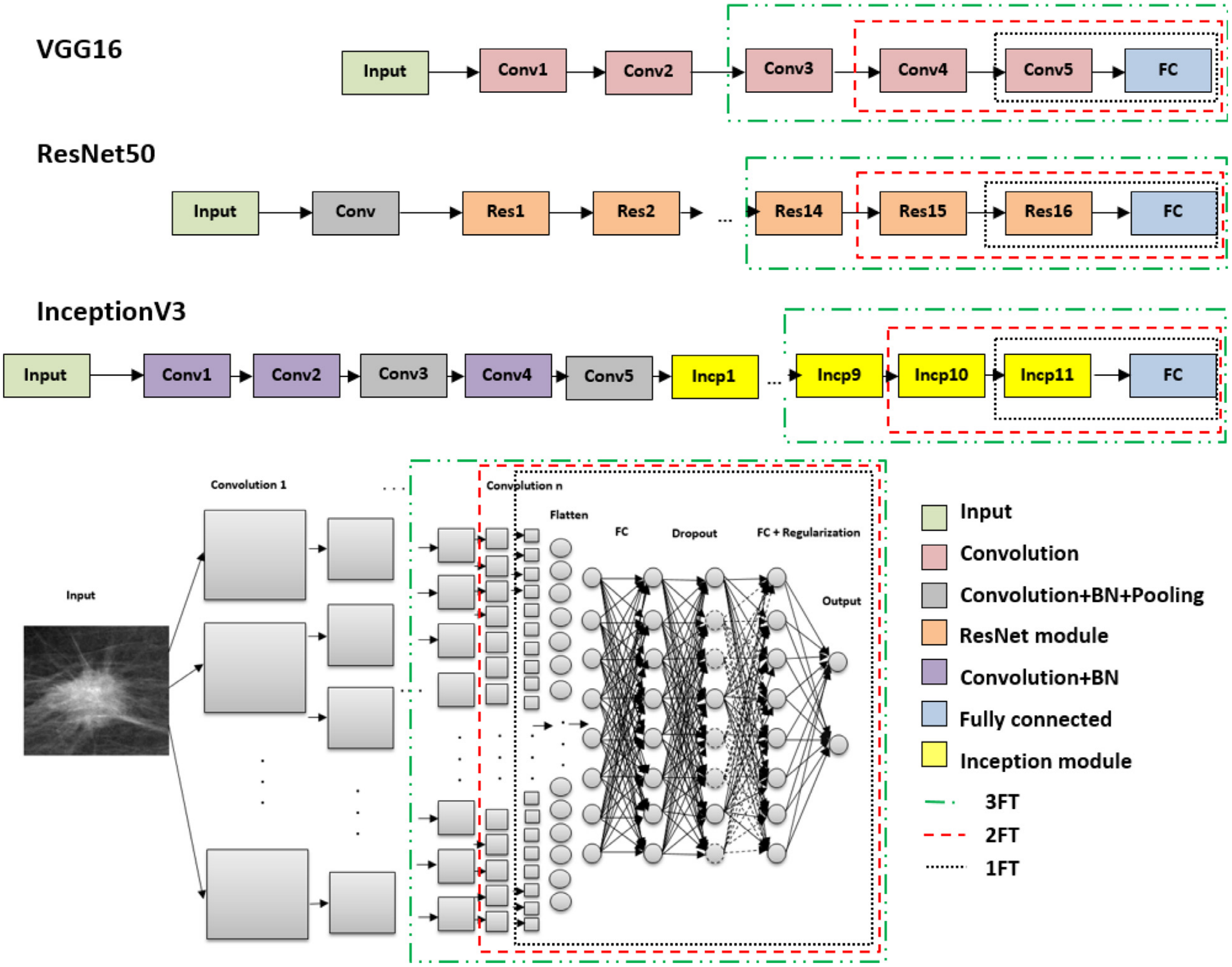


Fig. 3. Schema representing the different architectures and implementations of the models using transfer learning while adopting a fine-tuning strategy for some of the last convolutional blocks. The top part of the figure gives an overview of some of the layers composing each model; each layer is represented with a different color (keys in the bottom-right of the figure). The bottom-left part of the figure gives a detailed architecture of the customized fully-connected layers added to the bottom convolutional part of each model; note that the randomly turned-off activations in the dropout layer are represented with dotted circles. The three differently-colored dotted rectangular selections represent the different implementations of the models i.e. in each implementation the selected layers in the rectangle are fine-tuned while the rest of the model's layers are frozen.
{1FT = only one convolutional layer + the fully-connected part are fined-tuned while the rest of the layers are frozen;
2FT = two convolutional layers + the fully-connected part are fined-tuned while the rest of the layers are frozen;
3FT = three convolutional layer + the fully-connected part are fined-tuned while the rest of the layers are frozen;
BN = batch-normalization}.

rather than and adaptive learning rate optimizer, to make sure the magnitude of the learning rate stays small and not wreck the previously learned features [35].

When training the fully-connected model we used the adaptive ADAM optimizer [36] (Adaptive Moment Estimation). The method is designed to combine the advantages of two recently popular methods: AdaGrad and RMSProp, it computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients.

On the other hand, when fine-tuning we used the SGD optimizer with a small learning rate. We chose an initial learning rate of 1e-4 and it was divided by 10 each time the validation error stopped improving. We also adopted an early stopping strategy to monitor the validation loss with a patience set to 20 epochs i.e. the number of epochs to wait for the accuracy to get better, before stopping, if no progress is noted on the validation set.

Additionally, to improve the results and avoid overfitting, we used some tricks, for instance, we performed data augmentation, L2 regularization and dropout.

To ensure that our model generalizes well, we used L2 regularization (weight decay) to penalize large weights and prefer smaller ones. The L2 regularization penalty operates on the weight matrix W and is written as: $R(W) = \sum_i \sum_j W_{i,j}^2$.

The loss function then becomes: $L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda R(W)$ where λ is a hyper-parameter which controls the amount of the regularization we are applying. We used $\lambda = 1$ as it gave the best results.

Moreover, we added a dropout layer [37] so that it randomly turns off the activations at training time with a probability of .5. The randomly selected subset of activations are set to zero, which prevents some unit in one layer from relying too strongly on a single unit in the previous layer (Fig. 3).

2.2.5. The Breast Cancer Screening Framework

After fine-tuning the CNNs, we saved the weights of each model in HDF5 format and the structure in a JSON format. The model achieving the highest performance (see Table 2) i.e. the Inception v3 model trained on the merged database, which we will refer to

Table 2
Summary of the results obtained when fine-tuning the CNNs on the datasets.

Dataset	N of images	Model	Accuracy (%)	Std (%)	Time elapsed (Min)
DDSM	5316	VGG16	97.12%	±0.30	271.8
		ResNet50	97.27%	±0.34	122.1
		Inception v3	**97.35%**	±0.80	91.6
BCDR	600	VGG16	96.50%	±0.85	32.3
		ResNet50	96.50%	±2.30	17.6
		Inception v3	**96.67%**	±0.85	20.4
INbreast	200	VGG16	95.00%	±0.50	15.9
		ResNet50	92.50%	±2.36	10.1
		Inception v3	**95.50%**	±2.00	14.3
MD	6116	VGG16	98.64%	±0.22	326.3
		ResNet50	98.77%	±0.05	139.9
		Inception v3	**98.94%**	±0.22	64.7

as “Inceptionv 3-MD”, was used as a baseline to build the *Breast Cancer Screening Framework*.

To evaluate the performance of the Inceptionv 3-MD model, we tested it on an independent database. Initially, we one-hot encoded all of the database labels i.e. each label was represented by a vector p of size $K=2$ corresponding to the number of output classes. p is composed of the value 1 for the correct class and 0 for the other class.

Next, we fed the test images to the model and got the outputs probabilities from the Softmax classifier:

$$f_i(y) = \frac{exp(y_i)}{\sum_j exp(y_j)}$$

The Softmax function gives normalized class probabilities for the output being “Benign” or “Malignant”, the sum of these probabilities adds up to 1.

We could then measure the accuracy of the model by comparing the two vectors: first the Softmax vector that comes out of the classifier and contains the probabilities of the classes, and the other one is the one-hot encoded label vector.

To measure the distance between these two probability vectors, we use the cross-entropy loss:

$$L_i = \ -log\left(\frac{exp(f_{y_i})}{\sum_j exp(f_j)}\right)$$

where f_j is the jth element of the vector of class scores f. We denote the distance between the two vectors with D, the Softmax function with f and the label vector with p. The cross-entropy between a “true” distribution p and an estimated distribution f is defined as:

$$D(f, p) = -\sum_i p_i log(f_i)$$

When the ith entry corresponds to the correct class $p_i = 1$, the cost (i.e., distance) becomes $-log(f_i)$ and when the ith entry corresponds to the incorrect class, $p_i = 0$, the entry in f_i becomes irrelevant for the cost.

To further evaluate the general applicability of the model, we built a user-friendly interface based on the Inceptionv 3-MD to classify new images.

We used python and Tkinter to create the GUI (Graphic User Interface) and Keras [30] with Theano [38] to manage the model. The framework takes an image as an input, and gives as an output the predicted class label to be displayed as an output (Fig. 4).

3. Experimental results

The extracted ROIs were of size r × r ($r=300$), we rescaled them to be 224 × 224, so that they can be compatible with the original size of images from ImageNet which were used to train the original CNNs.

We train and evaluate the CNNs using a stratified 5-fold cross validation. The mean accuracy, standard deviation and time elapsed for training each model is reported and are used for comparing the different setups.

First, we investigate the extent that has transfer learning, through the use of pre-trained weights as an initialization for the CNNs, over training the models from scratch with a random initialization (Fig. 5). We use the pre-trained models as fixed features extractors and the Softmax layer on top as a classifier; we call this a 0 fine-tuning strategy (Fig. 6). Finally, we carry out an experimentation to find out the optimal number of layers we need to fine-tune for each model in order to get the best performance. We test with fine-tuning one, two, three and all of the convolutional blocks of our models and we examine their performance on DDSM, BCDR and INbreast databases. The blocks were fine-tuned for 90 epochs with a batch size of 128 images. We used the Keras library [30] with Theano [38] as a backend and the Cuda enabled GPU NVidia GTX 980 M.

3.1. Random initialization vs. transfer learning

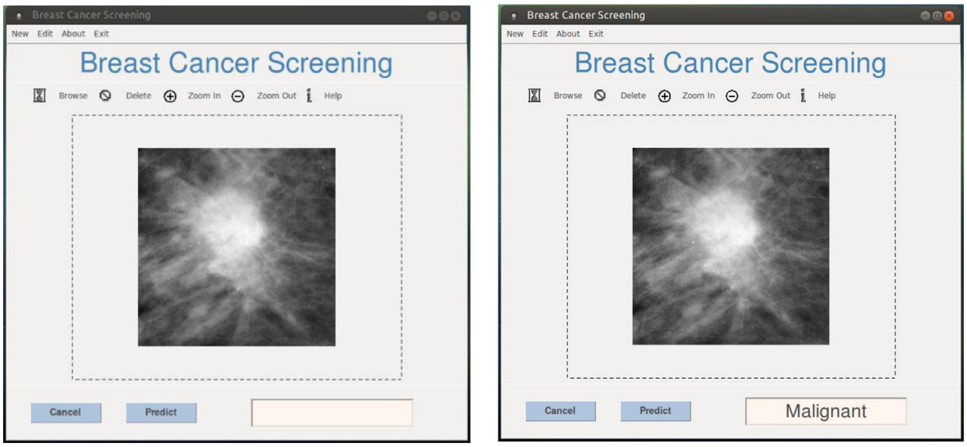
On the one hand, we randomly initialize all our models and train them on the datasets and on the other we use the pre-trained models as an initialization for our models. Fig. 5 gives the results from the comparison between the two different setups trained on DDSM, BCDR and INbreast.

Random initialization merely samples each weight from a standard distribution with a low deviation. The idea is to pick weight values at random following a distribution which would help the optimization process to converge to a meaningful solution.

The networks weights were initialized to a small random number generated from a Gaussian distribution, in our case the values were between 0 and 0.05. The low deviation allows to bias the network towards a simple 0 solution, without the bad repercussions of actually initializing the weights to 0 [35].

As an alternative, we perform transfer learning through the use of pre-trained weights, obtained from the CNNs training on ImageNet, as an initialization for our networks weights. In Fig. 5 we compare random initialization versus the transfer learning strategy (0 fine-tuning).

The 0 fine-tuning strategy which we can also refer to as the feature extraction mechanism is the basic way of doing transfer learning. We first remove the classification part of the networks (i.e. the fully-connected layers) which was responsible for giving the probabilities of an image as being from each of the 1000 classes in ImageNet. Then, we use the remaining part of the models as a fixed feature extractor that computes the CNN codes of each image from our datasets. We finally use a Softmax classifier to train on the obtained high-dimensional CNN codes (i.e. feature vectors).



Malignant

Fig. 4. Screenshots of the obtained result from our *Breast Cancer Screening Framework* based on the Inception v3-MD model; Case of an image containing a suspected malignant mass lesion.

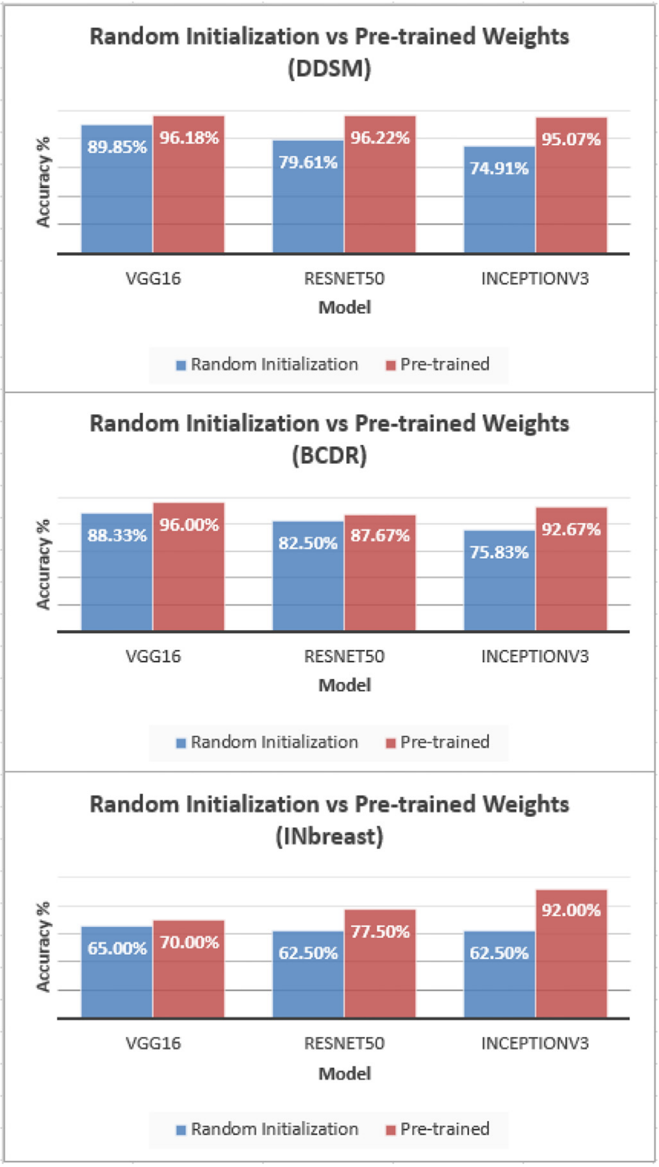


Fig. 5. Randomly initialized weights vs. pre-trained weights.

Fig. 5 gives the comparison summary of the performance of each network with the two initialization methods while trained on the same dataset. We can see that the pre-trained models outperformed randomly initialized models each time. The assumption is that it is beneficial to initialize the CNNs with weights from a pre-trained model since the pre-trained weights may already be good compared to randomly initialized weights. A pre-trained network on a large and diverse dataset like ImageNet learns to capture universal features like lines and edges in its early layers which can be relevant and useful to most classification problems.

3.2. Fine-tuning: How many layers are too many layers?

The nature of our data which is very different in content compared to the original dataset (ImageNet) allowed us to adopt a fine-tuning strategy.

We loaded the initialization weights of the models [29]; we froze the layers of the bottom architecture and trained the costumed fully-connected model on top with each dataset for 30 epochs.

Training the new fully-connected model before performing the fine-tuning is important, in order to start with properly trained weights and not wreck the learned convolutional base. Experimental results showed an improvement of 5% to 10% when training the newly added fully-connected layers before proceeding with the fine-tuning.

We investigated the optimal number of layers to fine-tune that would give us the best performance. Fig. 3 illustrates how we changed the number of frozen and unfrozen layers each time to measure the effect that has the number of layers fine-tuned on the results.

For instance, in the 1 fine-tuning (1FT), we froze all the layers until the last convolutional block. We unfroze the last convolutional block plus the fully-connected block by resuming back-propagation on them with a small learning rate (1e-4).

For the 2 fine-tuning (2FT) we unfroze two blocks instead of one and for 3 fine-tuning, three blocks (3FT). The All fine-tuning is when we unfroze all the layers and changed the values of all the weights of the models while back-propagating through the whole network.

Fig. 6 gives the accuracy results for fine-tuning the different fractions of the models while using the datasets. We can see that the accuracy increased when we went from 0 fine-tuning to 1 fine-tuning then from 1 fine-tuning to 2 fine-tuning. However, once the number of convolutional blocks exceeded two (i.e. 3 blocks and more) the accuracy started dropping.

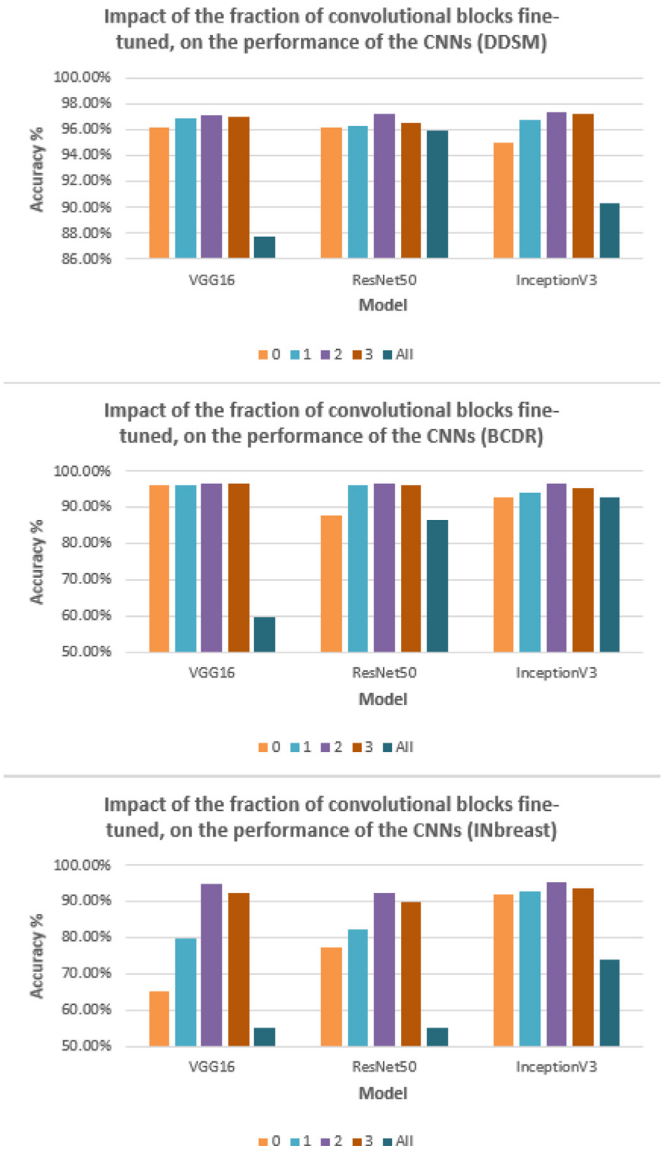


Fig. 6. Comparison of the results obtained using different fine-tuning strategies {0, 1, 2, 3, All} for transfer learning with our models while using our datasets (DDSM, BCDR, INbreast).

We can also see that 2 fine-tuning (2FT) is the clear choice when it comes to the optimal number of convolutional blocks we fine-tune. The assumption here is that while using the pre-trained weights is the better option, the last layers of the models learn more data specific features which in our case are very different from our type of data. Therefore, resuming back-propagation on the last convolutional layers may lead to a better performance as we'll start learning features that are more suited to our set of data.

Yet, fine-tuning too many layers leads to worse results. Perhaps, with the deep architectures and the small datasets used, the models started overfitting by learning irrelevant features due to their large entropic capacity. And it is showing in Fig. 3 for the INbreast database (which is the smallest one in our case of study) where we got an accuracy close to 50% for our test sets while fine-tuning all layers of VGG16 and ResNet50.

The idea here is that we need to make the last convolutional layers learn more data-specific features but there is no need for us to disturb the first convolutional layers as they're already well tuned to learn generic features especially if we don't have enough data to train on.

3.3. Classification results of the best CNN models implementations

Transfer learning using pre-trained weights from ImageNet while adopting a 2 fine-tuning strategy (2FT) is clearly the best setup for all our CNNs.

We use this implementation of our models to train on the datasets. Additionally, we train the CNNs also on the merged dataset (MD) and we compute the mean accuracy, standard deviation and time elapsed for each case.

Table 2 reports the performance and results of each model when fine-tuned on the different datasets.

We can see that all of the CNNs achieve good accuracy rates, but the Inception v3 model outperforms them all and over all the datasets. As it has already been shown in Table 1, we notice that the Inception v3 model is also the deepest network among the others. As a matter of fact, the network's depth affects its performance by making the learning easier. The advantage is that multiple layers makes it possible to learn features at various levels of abstraction, so that global features are progressively learned as combinations of local features along the depth of the network. When it comes to the computational time Inception v3 and ResNet50 are both much faster than VGG16. Although both networks are deeper than the latter, but still they have lower complexity, as they are not stacked up sequentially. On the other hand, we can clearly see that the size of data used to re-train the networks affects the results (Table 2), as it is well known that CNNs perform better when trained on larger sets of images. The results obtained on the merged dataset (MD) confirm that deep learning networks generalize better when provided with more data. The performance of the networks on the test set from INbreast compared to a larger dataset for example DDSM or MD shows that while the architecture and depth of the CNN model used is important, what is more important is the quality and quantity of the training data.

3.4. Monitoring the performance of our models

Due to the small number of training examples we have, compared to the thousands of images from ImageNet used to train the original model, we had to prevent the models from overfitting.

We performed data augmentation but it was not enough, because the augmented samples were still highly correlated. To remedy this, we forced the weights of the models to take smaller values by applying an L2 regularization and added a dropout layer.

Consequently, we needed to monitor the performance of each model as we do not want our models to start learning too-well, to the point that they cannot perform as well on never seen data.

Initially, we used a train/test random split with 80% of the data for training/validation and 20% for testing. We used the validation set to monitor and tune the hyper-parameters of each model as it trains. Fig. 7 illustrates the plots of accuracy and loss over the epochs for the Inception v3 model trained on the merged database ("Inceptionv 3-MD" model). As shown in Fig. 7, the validation set was checked during training to monitor progress, and to possibly reduce the learning rate when it reaches a plateau or to force an early stopping. The test set was then used as hold-out set to measure the model's performance on never-seen data.

After tuning the models and optimizing all of the hyper-parameters. We used the best setup for each CNN to train one final model on all the data using a stratified 5-fold cross validation. We saved the obtained finalized model for a later use or for making predictions on new data.

Stratified cross validation gives a less biased estimate of the model's skill on unseen data while it attempts to balance the number of instances of each class in each fold, to ensure that each fold is a good representative of the whole. For each dataset we create and evaluate multiple models on multiple subsets of the dataset.

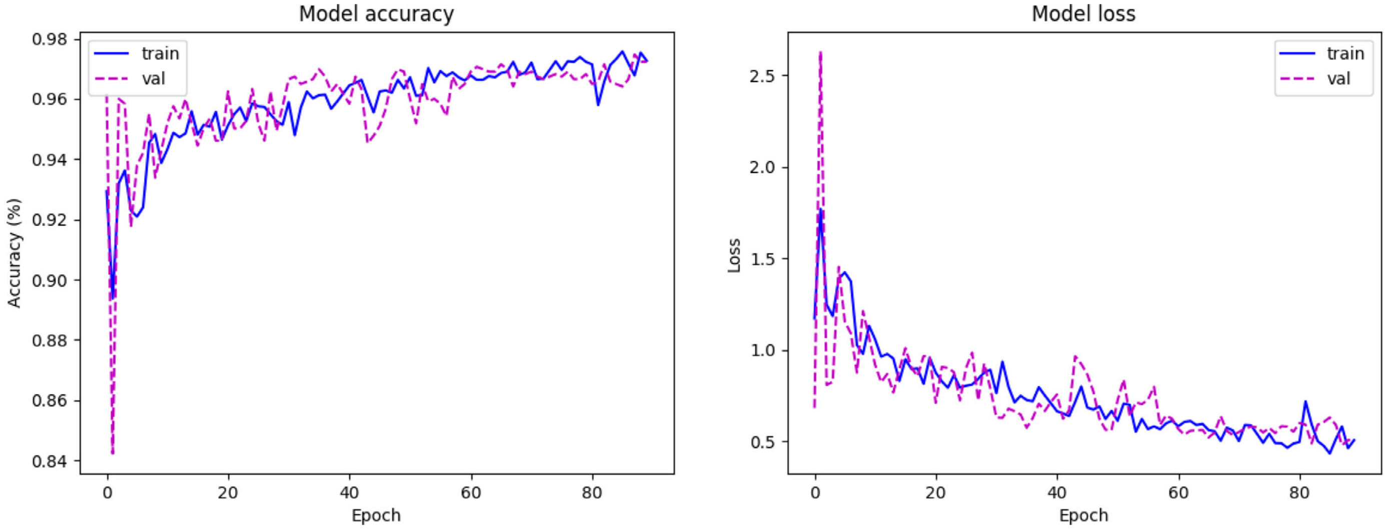


Fig. 7. The plot of accuracy and loss over the epochs for the Inception v3 model trained on the merged dataset (Inception v3-MD); the plots help in tuning the model and its hyper-parameters while monitoring its performance on the train and validation sets.

The reported mean and standard deviation at the end provide a robust estimate of the model's performance (see Table 2).

3.5. Testing the Inceptionv 3-MD on MIAS

The Mammographic Image Analysis Society (MIAS) [39] is a commonly used database. It contains 322 digitized film mammograms and includes radiologist's ground-truth markings on the locations of the suspected lesions. The MIAS data is classified using various criteria, we chose the images that were classified according the severity of the abnormality criteria so the mammograms not showing masses (Normal) were removed from the dataset and we kept only the mammograms with benign and malignant lesions which formed a subset of 113 images.

The subset contained images that were not previously seen by our CNN so it was used as a test set. We used the same preprocessing steps on the new images. The Inceptionv 3-MD was loaded and compiled to perform the classification. We measured the performance on this set using the overall accuracy, the Receiver Operating Characteristic curve (ROC) and the Area Under the Curve (AUC) metrics as they are the most adopted measures when it comes to evaluating classification systems.

Fig. 8 gives the ROC curve for the classification of the MIAS database using the pre-trained Inceptionv 3-MD model. The plot of the true positive rate against the false positive rate follows the left-hand border and then the top border of the ROC space, which indicates that the results are indeed accurate. Moreover, the model achieves an accuracy of 98.23% and AUC of 0.99 which indicates that the *Breast Cancer Screening Framework* is robust and can be trusted to classify new data.

3.6. Comparison summary of our work with others

Table 3 gives a comparison summary of some of the works that used CNNs for the classification of the datasets we used.

For instance, we have the work of Carneiro et al. [16] in which they first trained a separate CNN model for each view of the breast (MLO and CC). Then they used the features learned from each model to train a final CNN classifier which estimates if the case is benign or malignant. The authors tested their approach on the two publicly available datasets DDSM and INbreast and achieved an AUC of 0.97 and 0.91 respectively, while we achieved an AUC of 0.98 and 0.97. Jiao et al. [18] also classified DDSM database using a deep feature based framework which combined intensity information and deep features extracted from a trained CNN to predict the category of the test images. They used the accuracy measure to evaluate the performance of their approach and achieved 96.7% on DDSM while we achieved 97.35%.

Besides, Arevalo et al. [17] adopted a hybrid approach where they used a CNN to learn the representation of the mammog-

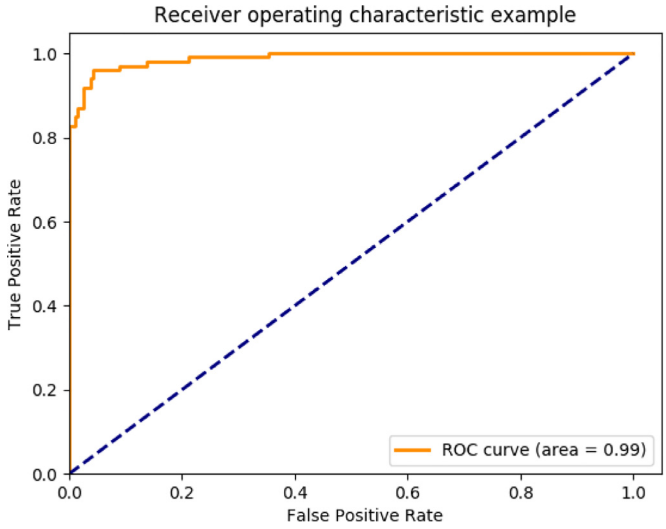


Fig. 8. Receiver Operating Characteristic (ROC) of the classification of MIAS using the Inception v3-MD model.

Table 3
Comparison summary of our approach with others in the literature.

Authors and date	Database	Accuracy	AUC
W. Peng et al. 2015	MIAS	96%	–
G. Carneiro et al. 2015	DDSM	–	0.97
	INbreast	–	0.91
J. Arevalo et al. 2015	BCDR	–	0.826
Z. Jiao et al. 2016	DDSM	96.7%	–
Ours	MIAS	**98.23%**	**0.99**
	DDSM	**97.35%**	**0.98**
	INbreast	95.50%	**0.97**
	BCDR	96.67%	**0.96**

raphy images in a supervised way, and then combined the obtained features with hand-crafted descriptors to classify mass lesions from the BCDR database. They achieved an AUC of 0.826 and we achieved 0.96 on the same dataset.

To evaluate the performance of our proposed *Breast Cancer Screening Framework* we used the MIAS dataset, we calculated both the AUC and the overall classification accuracy of the test images and got 0.99 and 98.23% respectively. In contrast, an evaluation of the same dataset by Peng et al. [15] while using ANN (Artificial Neural Networks) with texture features extracted from the images resulted in a 96% accuracy.

4. Discussion

To build an end-to-end powerful classification tool for breast cancer screening, we explored various setups and approaches.

First, we investigated the extent that has transfer learning over random initialization. We tested the performance of all our networks with the two approaches while training on three public datasets. The results demonstrated that initialization with pre-trained weights is advantageous, and that it may be due to the fact that the weights are already familiar with some universal features and patterns that were learned from ImageNet as opposed to random weights.

We then examined the possible ways of doing transfer learning. We adopted a “0 fine-tuning” strategy where we used the CNNs as feature extractors then classified the resulting CNN codes using a Softmax classifier. Afterwards, we started unfreezing the last convolutional blocks one by one until 3 blocks where the accuracy started to drop. For the purpose of the experiment, we further pushed the fine-tuning strategy to the extreme by fine-tuning all of the CNNs layers and we evaluated the performance in each case. The results indicated that while fine-tuning is beneficial and can lead to a better performance (i.e. if we compare “0 fine-tuning” to 1 fine-tuning and “1 fine-tuning” to “2 fine-tuning”), but too much fine-tuning, as for example the “All fine-tuning” strategy, leads to worse results. We found out that the optimal number of blocks to fine-tune was 2 convolutional blocks. This enabled us to keep the first layers which learn generic features and fine-tune only the last layers to make them learn more data-specific features.

Transfer learning and fine-tuning allowed us to use the learned ImageNet weights of different deep learning models as an initialization to our CNNs, and fine-tune them so as they can differentiate malignant breast mass lesions from benign ones.

The obtained results show a clear improvement over other proposed methods. Many of the works which classified mammography mass lesions employed simple neural networks, shallow Convolutional Neural Networks (i.e. not deep enough) or the combination of extracted CNN features with other hand-crafted descriptors. However, the most interesting aspect of CNNs is the end-to-end learning, leading to a better performance while using less complex algorithms. The better performance comes from the fact that the internal components self-optimize to maximize the overall system performance. And Compared to the traditional neural networks, CNNs reduce the computational cost as they have fewer parameters and are easier to train.

When comparing the obtained results of each CNN fine-tuned on the different datasets, we noticed that the depth of the model as well as its architecture affects its performance. The best results for each dataset were obtained using the Inception v3 model, which also happens to be the deepest network among the others. The Inception v3 seems to be more suited for fine-tuning, maybe it is because of its architecture which is deep but not stacked up, making it less sensitive to the vanishing gradient problem. As a result, fine-tuning the pre-trained Inception v3 model enabled us to achieve a better performance compared to the state-of-the-art methods which classified the same public datasets we used, and this in terms of both accuracy and AUC metrics.

Intensity normalization is an important preprocessing step in medical imaging. During image acquisition, different scanners and parameters are used for scanning the different patients or even the same patient sometimes, which may result in large intensity variations. Those variations can be more flagrant from one set of data to another (different illumination conditions, materials, expert in charge…etc.). This intensity variation can greatly undermine the performance of the proposed system for mammography analysis. Subsequently, before using the images and especially before merging all datasets, we used GCN normalization to reduce the intensity variation between images, which may have been taken under different conditions. The normalization helps phase out the intensity variations caused by the various lighting conditions. So, that we can effectively reduce intra-variations between images from the same dataset and inter-variations between images from different datasets.

To perform the transfer learning, we used datasets of different sizes and the results obtained indicated the existence of a correlation between the number of training data and the performance of the models. Thus, we combined all datasets to build one large set of images. The merged dataset was used to fine-tune the networks. Assuredly, the Inception v3 model outperformed the other networks using this set of data as it achieved 98.94% accuracy.

A deep CNN composed of many layers trained on a small dataset should have a large entropic capacity. The model is then able to store a lot of information, which gives it the potential to be highly accurate by exploiting more features. However, it can also make it more at risk of storing irrelevant features. To modulate the entropic capacity of our models, we had to first enlarge our datasets through data augmentation. Then, we only fine-tuned the last two convolutional layers of the models to get more dataset-specific features. In addition, we applied L2 regularization and dropout to disrupt complex co-adaptations on training data, and so we made the models focus on the more significant features from the images, for a better generalization. Furthermore, all the models were meticulously monitored to examine their performance on the training data and the validation data in order to optimize the hyper-parameters and select the best model. The latter was then used to assess both training and test sets simultaneously, to ensure that the model is not overfitting and that it performs equally well on never-seen data (the test data).

After tuning the models and choosing the best hyper-parameters, we trained one final model for each CNN using a stratified 5-fold cross-validation with all the data and we computed the mean accuracy, standard deviation and time elapsed for each experiment to evaluate the performance.

We used the Inception v3 model fine-tuned on the merged dataset to develop a powerful classification tool. The *Breast Cancer Screening Framework* can be used as a Computer-aided Diagnosis system that classifies mammography mass lesions. To evaluate the framework, we tested it using new images from the MIAS database, and we achieved an area under the curve (AUC) of 0.99. The results obtained outperform by a large margin human performance, with radiologists achieving a 0.82 AUC according to [40].

The developed framework could predict and provide the correct diagnosis for 98.23% of the images from MIAS, 97.35% from DDSM 95.50% for INbreast and 96.67% for BCDR. The results obtained from the receiver operating characteristic (ROC) curve analysis showed a high true-positive rate for all previous datasets, which means a high probability of correctly identifying malignant mass lesions as being cancerous.

Fig. 9 illustrates some examples of images that were misclassified (red frames) versus others that were correctly classified (green frames).

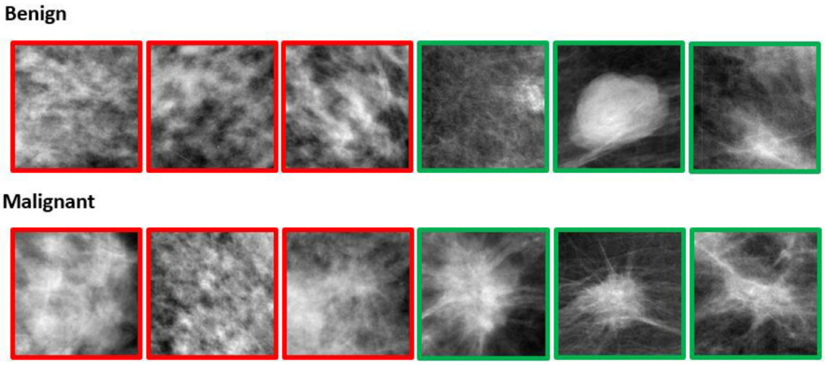


Fig. 9. Examples of regions of interest containing mass lesions; the first row contains benign lesions and the second row contains malignant lesions; the misclassified images are framed by a red bounding box (the 3 images on the left in both rows) and the correctly classified by a green one (the 3 images on the right in both rows). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Examining the misclassified images we can see that the texture of some of the benign and malignant images is similar. One possibility is that this is due to high breast density.

It is well known that cancer is more difficult to detect, in mammograms of women with radiographically dense breasts [41].

Breasts are made up of lobules, ducts, and fatty and fibrous connective tissue. The breasts are dense in the presence of a lot of glandular tissue and not much fat. On mammograms, dense breast tissue looks white. Breast masses or tumors also look white, hence, the dense tissue can hide tumors. On the other hand, fatty tissue looks almost black. On a black background it is easier to identify a tumor that looks white (Fig. 9). Therefore, mammograms can be less accurate in women with dense breasts.

This suggests that we can further improve our framework, if we were to carefully select the images to train on and give the model more challenging examples to learn from. We can also include additional imaging techniques in the learning process, such as Breast Ultrasound or breast MRI (Magnetic Resonance Imaging), to help get a clearer view of the breast, especially for cases with high breast density [42,43].

5. Conclusions

In summary, we can conclude that integrating the recent well-engineered deep learning CNNs through transfer learning into the screening mechanism brings an apparent improvement compared to other approaches. The fine-tuning strategy proposed improves the state-of-the-art accuracy classification on many public datasets. The Inception v3 model trained on the merged dataset, which achieved the best accuracy rate overall, was used to develop a mass lesion classification tool. The *Breast Cancer Screening Framework* devised, could successfully classify many "never-seen" images of mammography mass lesions. It provided highly accurate diagnoses when distinguishing benign from malignant lesions. Therefore, its output could be used as a "second opinion" to assist the radiologist in giving more accurate diagnoses.

Our future work includes using deeper architectures as well as more challenging images to deal with the hindrance caused by mammograms of highly dense breasts. Besides, we suppose that it can also be beneficial to incorporate other imaging techniques in combination with mammography, in the learning process, to help build a robust and powerful breast mass lesion classification tool.

Conflict of interest statement

No conflict of interest.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

The BCDR database used in this work was a courtesy of MA Guevara Lopez and coauthors, Breast Cancer Digital Repository Consortium.

The INBreast database used in this work was a courtesy of the Breast Research Group, INESC Porto, Portugal.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2018.01.011.

References

[1] R.L. Siegel, K.D. Miller, A. Jemal, Cancer Stat., CA. Cancer J. Clin. 67 (2017) (2017) 7–30, doi:10.3322/caac.21387.
[2] I. Schreer, Dense breast tissue as an important risk factor for breast cancer and implications for early detection, Breast Care 4 (2009) 89–92, doi:10.1159/000211954.
[3] K. Kerlikowske, P.A. Carney, B. Geller, M.T. Mandelson, S.H. Taplin, K. Malvin, V. Ernster, N. Urban, G. Cutter, R. Rosenberg, R. Ballard-Barbash, Performance of screening mammography among women with and without a first-degree relative with breast cancer, Ann. Intern. Med. 133 (2000) 855–863.
[4] L. Berlin, Radiologic errors, past, present and future, Diagnosis 1 (2014) 79–84, doi:10.1515/dx-2013-0012.
[5] J.S. Whang, S.R. Baker, R. Patel, L. Luk, A. Castro, The causes of medical malpractice suits against radiologists in the United States, Radiology 266 (2013) 548–554, doi:10.1148/radiol.12111119.
[6] E.A. Sickles, Periodic mammographic follow-up of probably benign lesions: results in 3,184 consecutive cases, Radiology 179 (1991) 463–468, doi:10.1148/radiology.179.2.2014293.
[7] Y. Bengio, Learning deep architectures for AI, found, Trends® Mach. Learn 2 (2009) 1–127, doi:10.1561/2200000006.
[8] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828, doi:10.1109/TPAMI.2013.50.
[9] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117, doi:10.1016/j.neunet.2014.09.003.
[10] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, doi:10.1038/nature14539.
[11] Y. LeCun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proc. 2010 IEEE Int. Symp. Circuits Syst., 2010, pp. 253–256, doi:10.1109/ISCAS.2010.5537907.
[12] F. Ciompi, B. de Hoop, S.J. van Riel, K. Chung, E.T. Scholten, M. Oudkerk, P.A. de Jong, M. Prokop, B. van Ginneken, Automatic classification of pulmonary perifissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box, Med. Image Anal. 26 (2015) 195–202, doi:10.1016/j.media.2015.08.001.
[13] H. Chen, D. Ni, J. Qin, S. Li, X. Yang, T. Wang, P.A. Heng, Standard plane localization in fetal ultrasound via domain transferred deep neural networks, IEEE J. Biomed. Health Inf. 19 (2015) 1627–1636, doi:10.1109/JBHI.2015.2425041.
[14] H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (2016) 1285–1298, doi:10.1109/TMI.2016.2528162.

[15] W. Peng, R.V. Mayorga, E.M.A. Hussein, An automated confirmatory system for analysis of mammograms, Comput. Methods Programs Biomed. 125 (2016) 134–144, doi:10.1016/j.cmpb.2015.09.019.

[16] G. Carneiro, J. Nascimento, A.P. Bradley, Unregistered multiview mammogram analysis with pre-trained deep learning models, in: Int. Conf. Med. Image Comput. Comput.-Assist. Interv., Springer, 2015, pp. 652–660. http://link.springer.com/chapter/10.1007/978-3-319-24574-4_78.

[17] J. Arevalo, F.A. González, R. Ramos-Pollán, J.L. Oliveira, M.A. Guevara Lopez, Representation learning for mammography mass lesion classification with convolutional neural networks, Comput. Methods Programs Biomed. 127 (2016) 248–257, doi:10.1016/j.cmpb.2015.12.014.

[18] Z. Jiao, X. Gao, Y. Wang, J. Li, A deep feature based framework for breast masses classification, Neurocomputing 197 (2016) 221–231, doi:10.1016/j.neucom.2016.02.060.

[19] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit, 2014, pp. 1717–1724. http://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Oquab_Learning_and_Transferring_2014_CVPR_paper.html. (accessed January 27, 2017).

[20] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, ArXiv14053531 Cs. (2014). http://arxiv.org/abs/1405.3531 (accessed June 2, 2017).

[21] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ArXiv14091556 Cs. (2014). http://arxiv.org/abs/1409.1556 (accessed February 10, 2017).

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. IEEE Conf. Comput. Vis.Pattern Recognit., 2016, pp. 2818–2826.

[24] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255, doi:10.1109/CVPR.2009.5206848.

[25] A. Halevy, P. Norvig, F. Pereira, The unreasonable effectiveness of data, IEEE Intell. Syst. 24 (2009) 8–12, doi:10.1109/MIS.2009.36.

[26] M. Heath, K. Bowyer, D. Kopans, R. Moore, W.P. Kegelmeyer, The digital database for screening mammography, in: Proc. 5th Int. Workshop Digit. Mammogr., Medical Physics Publishing, 2000, pp. 212–218. https://www3.nd.edu/~kwb/Heath_EtAl_IWDM_2000.pdf. (accessed January 27, 2017).

[27] M.G. Lopez, N. Posada, D.C. Moura, R.R. Pollán, J.M.F. Valiente, C.S. Ortega, M. Solar, G. Diaz-Herrero, I. Ramos, J. Loureiro, others, BCDR: a breast cancer digital repository, 15th Int. Conf. Exp. Mech., 2012 http://paginas.fe.up.pt/clme/icem15/ICEM15_CD/data/papers/3004.pdf accessed January 27, 2017.

[28] I.C. Moreira, I. Amaral, I. Domingues, A. Cardoso, M.J. Cardoso, J.S. Cardoso, INbreast: toward a full-field digital mammographic database, Acad. Radiol 19 (2012) 236–248, doi:10.1016/j.acra.2011.09.014.

[29] S.C. Wong, A. Gatt, V. Stamatescu, M.D. McDonnell, Understanding data augmentation for classification: when to warp?, ArXiv160908764 Cs. (2016). http://arxiv.org/abs/1609.08764.

[30] F. Chollet, Keras, GitHub, 2015 https://github.com/fchollet/keras.

[31] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, in: PMLR, 2012, pp. 17–36. http://proceedings.mlr.press/v27/bengio12a.html. (accessed October 19, 2017).

[32] H. Lakkaraju, R. Socher, C. Manning, Aspect specific sentiment analysis using hierarchical deep learning, NIPS Workshop Deep Learn. Represent. Learn., 2014.

[33] M. Jaderberg, K. Simonyan, A. Zisserman, Spatial transformer networks, in: Adv. Neural Inf. Process. Syst., 2015, pp. 2017–2025.

[34] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in: Adv. Neural Inf. Process. Syst., 2014, pp. 3320–3328.

[35] Y. Bengio, Practical Recommendations for Gradient-Based Training of Deep Architectures, in: Neural Netw. Tricks Trade, Springer, Berlin, Heidelberg, 2012, pp. 437–478, doi:10.1007/978-3-642-35289-8_26.

[36] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, ArXiv14126980 Cs. (2014). http://arxiv.org/abs/1412.6980 (accessed January 31, 2017).

[37] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.

[38] Theano Development Team, Theano: a python framework for fast computation of mathematical expressions, ArXiv E-Prints. abs/1605.02688 (2016). http://arxiv.org/abs/1605.02688.

[39] J Suckling, et al., The mammographic image analysis society digital mammogram database exerpta medica, Int. Congr. Ser. 1069 (1994) 375–378.

[40] J.G. Elmore, S.L. Jackson, L. Abraham, D.L. Miglioretti, P.A. Carney, B.M. Geller, B.C. Yankaskas, K. Kerlikowske, T. Onega, R.D. Rosenberg, E.A. Sickles, D.S.M. Buist, Variability in interpretive performance at screening mammography and radiologists' characteristics associated with accuracy1, Radiology 253 (2009) 641–651, doi:10.1148/radiol.2533082308.

[41] J.E. Joy, E.E. Penhoet, D.B. Petitti, I. of M. (US) and N.R.C. (US) C. on N.A. to E.D. and D. of B., Cancer, Benefits and Limitations of Mammography, National Academies Press, US, 2005 https://www.ncbi.nlm.nih.gov/books/NBK22311/ accessed June 8, 2017.

[42] W.A. Berg, Z. Zhang, D. Lehrer, R.A. Jong, E.D. Pisano, R.G. Barr, M. Böhm-Vélez, M.C. Mahoney, W.P. Evans, L.H. Larsen, M.J. Morton, E.B. Mendelson, D.M. Farria, J.B. Cormack, H.S. Marques, A. Adams, N.M. Yeh, G. Gabrielli, Detection of breast cancer with addition of annual screening ultrasound or a single screening MRI to mammography in women with elevated breast cancer risk, JAMA J. Am. Med. Assoc. 307 (2012) 1394–1404, doi:10.1001/jama.2012.388.

[43] D.S. Salem, R.M. Kamal, S.M. Mansour, L.A. Salah, R. Wessam, Breast imaging in the young: the role of magnetic resonance imaging in breast cancer screening, diagnosis and follow-up, J. Thorac. Dis. 5 (2013) S9–S18, doi:10.3978/j.issn.2072-1439.2013.05.02.