POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione – MI
Dipartimento di Elettronica, Informazione e Bioingegneria
Master of Science in Biomedical Engineering – Ingegneria Biomedica

POLITECNICO
MILANO 1863

Breast arterial calcifications on mammograms: deep learning detection for women's cardiovascular risk stratification.

ADVISORS:

Prof. Giuseppe Baselli
Dr. Marina Codari, PhD
Prof. Francesco Sardanelli

AUTHOR:

Maria Giovanna Ienco 893794

Academic year 2018-2019

ABSTRACT

Cardiovascular disease is a major cause of death in women. Up to 20% of all cardiovascular events in women occur without the attendance of conventional risk factors, highlighting a lack in currently cardiovascular risk stratification methods. Breast Arterial Calcifications (BACs), detected on mammograms for breast cancer screening, though extraneous to this primary aim, have attracted the attention of researchers involved in cardiovascular disease prevention. BACs have been suggested as a "potential women-specific cardiovascular risk marker" providing the possibility of transforming the already widespread breast cancer screening program into a double test. The major obstacles to this goal, however, is the lack of a robust method to quantify BACs in mammograms for cardiovascular risk quantification and also adequate automatic support to the further workload asked to radiologists.

In this thesis work, we tackled the latter issue and implemented a deep learning model capable of classifying full breast images according to BACs presence ($BACs^+$) or absence ($BACs^-$). We developed a 16-layer convolutional neural network (CNN) using a transfer learning approach. We selected one of the most famous CNN classifiers trained on low resolution natural images, VGG16 net, and customized it in order to classify high-resolution mammograms. We maintained the structure and filters of the original convolutional base and replaced the fully connected part with three new fully connected layers. We selected the optimal number of hidden units of the fully connected layers and the number of convolutional layers to fine-tune. This structure and the relevant hyperparameters were optimized to learn the high-level task-related features while avoiding overfitting. Then, we trained from scratches the fully connected layers, composed by 256, 256 and 1 neurons each, and fine-tuned the last five convolutional layers. To account for class imbalance in the dataset ($BACs^+$prevalence of 10%), we randomly down sampled the majority $BACs^-$ class until reaching a prevalence of 30%. In addition, a weighted training approach was used. Data-augmentation was carried out avoid overfitting and also the training epochs were stopped as soon as the validation loss function reached its minimum.

We evaluated the resulting architecture and learning strategy performing a 7-fold cross validation using precision, recall, and F1 score as performance metrics. The models showed good performance in terms of precision (range = [0.842-0.950], mean = 0.864 and SD = 0.040) while showing lower recall values (range = [0.433 -0.772], mean = 0.667, SD = 0.132), resulting in a F1 score ranging from 0.653 to 0.840 with mean and standard deviation values equal to 0.744 ± 0.094.

The observation of saliency maps proved the reliability of BAC detection highlighting the ROI of the single BAC or of the most evident BAC of several ones. This allowed us to ascertain the feasibility of transforming global information, such as an image-level annotation, into a local one. Hence, we foresee that the CNN will support the radiologist both by sorting out the few $BACs^{+}$cases and indicating the ROI or ROIs to be closely examined for a future BACs ranking.

Further investigations are needed in order to reduce the number of false negatives before testing the BACs classifier performance on a new independent testing dataset. Despite the obvious need to further improve the model, the results are encouraging and legitimate future studies on the potential role of deep learning automatic BACs detection in the prevention of cardiovascular disease in women.

Sommario

List of figures

List of tables

Acknowledgments

I would like to thank those who made the success of this thesis possible and those who supported me in these months of hard work.

I thank the Politecnico di Milano for giving me the theoretical foundations and the tools necessary to face such a complex problem with cutting-edge methods.

I thank Prof. Baselli for having followed me with constancy and patience and for giving me the opportunity to exploit his precious experience.

I thank Marina, who throughout the thesis period has been my fixed point of reference, always ready to give me the right advice. I thank you for the passion for your work that I was allowed to share with you.

Finally, I want to thank Prof. Sardanelli and all the research team of the IRCCS Policlinico San Donato, without them, this work would not have existed. Thank you for your time and sharing your knowledge with me.

1. Introduction

1.1 Breast arterial calcifications and cardiovascular risk

Cardiovascular disease (CVD) is the most common cause of death overall, causing the loss of around 7.9 million men and women each year. According to the World Health Organization, the number of deaths is destined to rise to 22.2 million by 2030[1]. The 85% of CVD demises are the consequences of strokes and ischemic heart diseases, which occupy the first two places in the ranking of global causes of deaths, the top 10 shown in *Figure 1.1*.

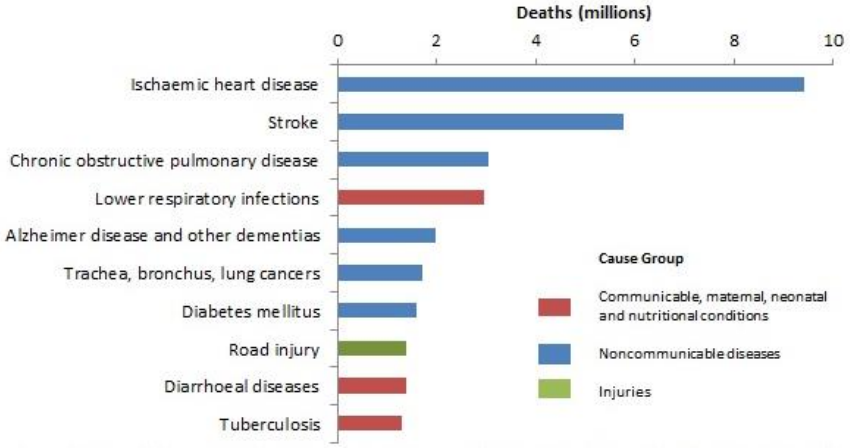


Figure 1.1 *Top 10 global causes of deaths, 2016. Ischemic heart disease and stroke are the two major cause of death worldwide [www.who.int]*

CVD constitutes a major healthcare problem that requires new prevention and risk stratification strategies, especially for women[2]. Many women with conventional risk factors have never experienced coronary heart diseases[3], while up to 20% of all coronary events in women occur without the attendance of major risk factors[4]. These facts suggest sex-specific risk factors or cofactors (like pregnancy complications, oral conception, menopausal therapies, hormonal fertility[5]) not included in the actual CVD risk assessment.

One of the most widely used CVD risk scores is the Framingham Risk Score (FRS), developed in 2008[6]. The FRS is a sex-specific algorithm that estimates the risk of manifesting clinical CVDs in the next 10 years. Age, sex, total cholesterol level, high-density lipoprotein, systolic blood pressure, smoke, diabetes, hypertension and other known vascular diseases are the FRS covariates. Current guidelines recommend the inclusion of also sex and ethnicity into the calculation[7], however the predictive value of risk scores based only on demographic and life-style factors is still poor. Actually, the lack of reliable, effective screening modalities remains and constitutes one of the major barriers to improve CVD outcomes in women[8]. Among noninvasive imaging biomarkers, coronary artery calcium (CAC) score is the most potent marker of subclinical cardiovascular disease and has been demonstrated to enhance risk prediction in women[9]. American and European guidelines recommend it to improve cardiovascular risk assessment in asymptomatic individuals with low-intermediate risk[10]. CACs are calcifications of the tunica intima (inner layer) (Figure 1.2) of the coronary arteries, vessels that wrap around the entire heart for suppling it. CACs are the results of the atherosclerotic process, an inflammatory process leading to lipid deposits and luminal narrowing[11]. CACs can be seen on a non-contrast chest computed tomogram (NCCT). However, a widespread CAC screening program, similar to mammography for breast cancer prevention, would expose women to excessive radiation, with a too unspecific indication. Furthermore, in some countries such as United States, assurance companies don't cover the costs of such screening.

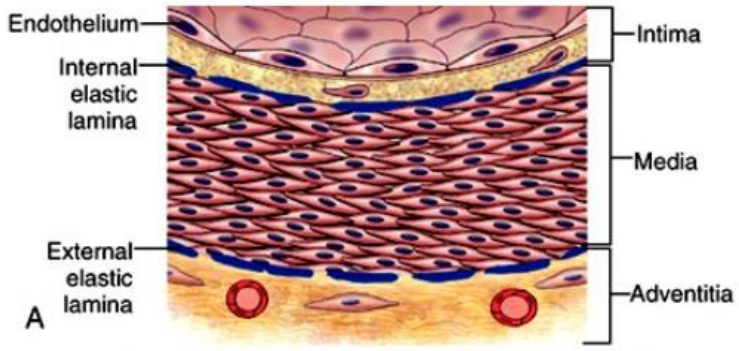


Figure 1.2 *Artery wall layers. Arteries are composed of three layers: tunica Intima, tunica media and tunica adventitia. Tunica intima is the layer in direct contact with the blood and is where the CACs occur. Tunica media is a muscular layer that lets arteries handle the high pressures from the heart. Tunica adventitia is the outermost layer that wraps the vessel.*
[Elsevier. Kumar et al: Robbins Basic Pathology 8e – www.strudentconsult.com]

Breast Arterial Calcifications (BACs) are localized calcific depositions in the tunica media of breast arteries and are a manifestation of the Mönckeberg's medial calcific sclerosis, notably different from atherosclerotic process involved in CACs formation[12]. Calcifications are diffuse within the tunica media of medium and small muscular arteries involving nonocclusive circumferential thickening (*Figure 1.3*) which results in stiffer, less compliant vessels.

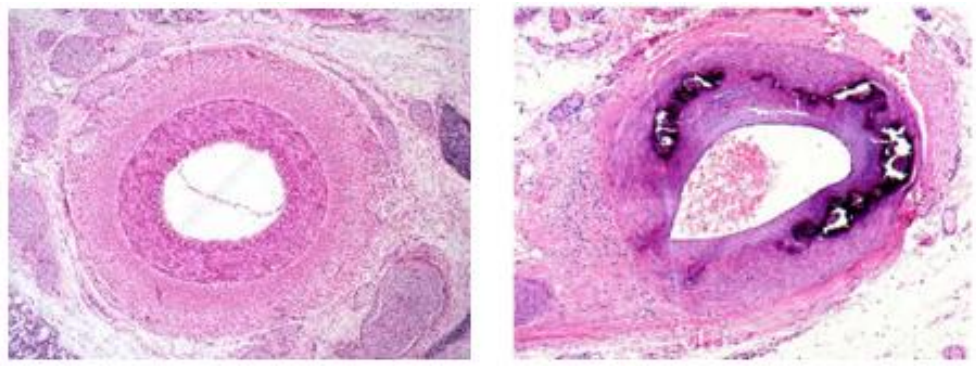


Figure 1.3 *Healthy and Mönckeberg's calcific arteries. Cross section of human artery in normal conditions [a] and with Mönckeberg calcifications in tunica media [b]. [www.sciencephoto.com, www.memorangapp.com]*

BACs prevalence varies depending on population age and comorbidities. In screening mammography population-based cohort studies it is reported to range from 10% to 12%. However, BACs prevalence can reach 70% in women aged 70 years or more in women with chronic kidney diseases[1314].

BACs are easily recognizable on breast cancer screening mammograms, where they appear as linear, parallel opacities on both sides of the vessel lumen (*Figure 1.4*). which justifies the term of "tram-track appearance", in particularly evident BACs[15]. However, BACs can assume several aspects: involving vessels, or only one side of them, or can also appears as small intense dots superimposed on the artery lumen (*Figure 1.5).*

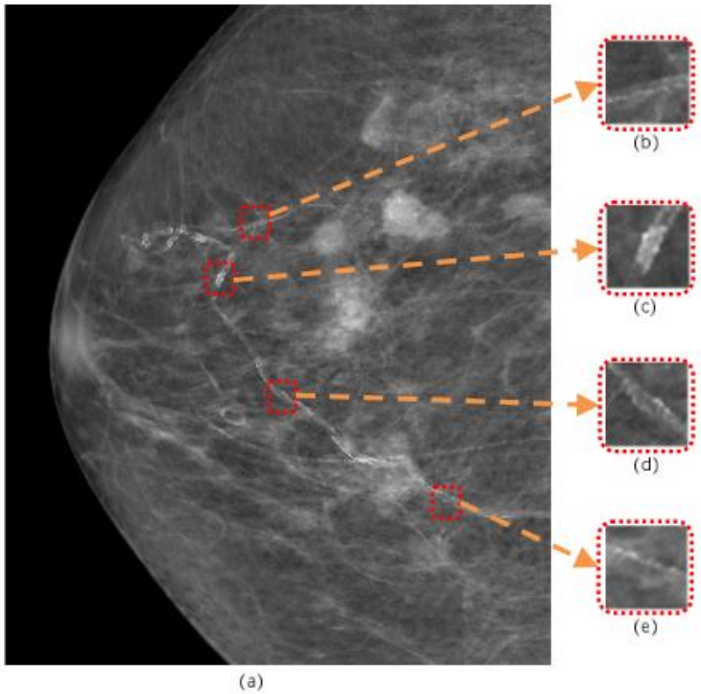


Figure 1.4 *BACs illustration from a clipped mammogram craniocaudal view.(a) A mammogram. (b) – (e) Examples of different appearance patterns of calcific arteries on the same mammogram. [from[16]]*

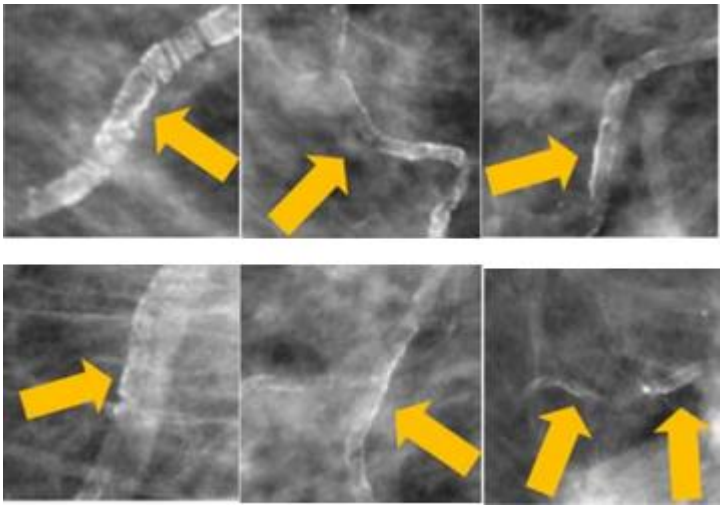


Figure 1.5 *Various appearance patterns of BACs. Breast arterial calcifications in zoomed mammograms are indicated with yellow arrows. Different appearance is due to different amount of calcium deposition and 2D projection effects. [from[16]]*

From an oncological perspective, BACs do not represent a sign of breast cancer and for this reason are ignored or barely reported as present/assent. Nevertheless, BAC become more interesting for the research community as a "potential women-specific CVD risk marker[17]". Several studies investigated the association among BAC seen on breast cancer screening mammograms, traditional cardiovascular risk factors and CVD events As reported on a recent meta-analysis published by Hendriks *et al.* [13], age and diabetes are directly associated with BACs prevalence while no associations were found with other CVD risk factors such as obesity, hypertension and dyslipidemia. BACs are instead associated with an increase of CVD events suggesting that "medial arterial calcifications might contribute to CVD through a pathway distinct from the intimal atherosclerotic process[13]".

In a study[18] performed on a retrospectively selected sample of 292 women who underwent both mammography and NCCT, Margolies *et al*. investigated the association between CACs and BACs assessed using quantitative scores (0 to 12). They compared them with FSR and the 2013 Cholesterol Guidelines Pooled Cohort Equations (PCE). Their results showed that BACs are associated with increasing age ($p<0.001$), hypertension ($p=0.007$) and chronic kidney disease ($p<0.0001$) and all BACs variables are predictive of the CACs score ($p<0.0001$). BACs>0 had area under the curve of 0.73 for identification of women with CACs>0, equivalent to both FSR (0.72) and PCE (0.71). For the identification of high-risk CACs (score from 4 to 12) BACs>0 increased the area under the curve curves for FRS (0.72 to 0.77; $p=0.15$) and PCE (0.71 to 0.76; $p=0.11$). BACs resulted to be superior to standard cardiovascular risk factors and to be strongly quantitatively associated with CACs.

In this light, we should exploit current breast cancer screening mammographic program to obtain a double test. Mammograms for breast cancer screening could be further exploited for CVD prevention without any additional radiation exposure or cost. Bui *et al.*[8] in their review wrote: " At the very last, we strongly believe that the presence of BAC should initiate a personalized patient-provider discussion surrounding lifestyle changes and targeted medical therapies for prevention of cardiovascular disease or consideration for referral for cardiovascular risk assessment by specialist". Their proposal, anticipated

and shared by many researchers actively involved in the cause, is reported schematically in *Figure 1.6*

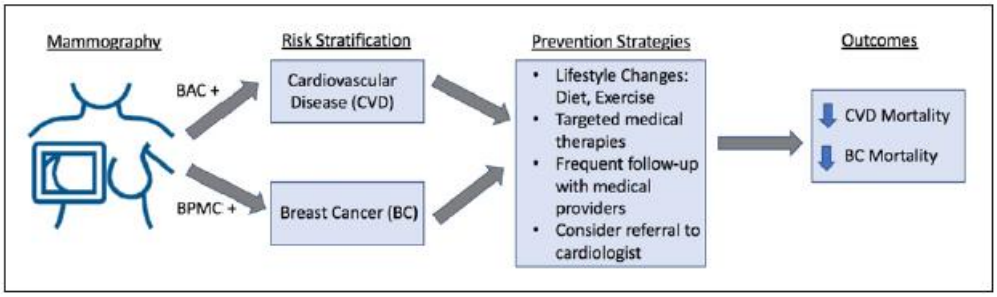


Figure 1.6 *Cardiovascular Disease + Breast Cancer screening program [from.[8]]*

Harnessing the full potential of digital screening mammograms can enhance prevention of the two leading causes of death in women, namely CVD and breast cancer. Still, this ambitious goal lacks two important points: a) so far, it doesn't exist a robust quantitative or semi-quantitative scale to quantify BAC load, to stratify women's CV risk [19]; b) an AI or deep learning (DL) tool to assist radiologist in this further diagnostic effort.

The latter issue is focused by this thesis and is motivated by the difficulty of BACs detection, also considering its difficulty even to expert radiologists, which could distract them from the primary cancer prevention purpose. Indeed, the above introduction, has clearly illustrated such problem, since BACs: a) can be significant even if a single lesion was radiologically detectable; b) have variable aspects and dimensions; c) their localization in the breast is unpredictable.

Therefore, it was decided to provide a DL CNN tool, trained, validate, and possibly tested on a sufficiently large database of mammograms annotated as BACs positive ($BACs^+$) or negative ($BACs^-$). Currently 293 $BACs^+$and 2575 $BACs^-$ images were available, to be increased in the near future. Accordingly, the CNN outcome is limited to the indication of a highly probable presence of one or BACs, avoiding the specific analysis and waste of radiologist's time on the prevalent $BACs^-$ cases, since the demographic prevalence in women undergoing mammographic screening ranges from 10% to 12% . Nonetheless, limited localization of at least the prominent BAC lesion is given by the CNN decision heat map.

1.2 Digital mammography and BACs

1.2.1 Image acquisition process and characteristics

Mammography is a specific type of two-dimensional (2D) breast imaging that uses low-energy X-rays, usually around 30 kVp[20]. It is mainly used for the detection of breast cancer at early stage ahead of palpable breast nodules. During a mammogram, a patient's breast is placed between two plastic plates and compressed to reduce projection thickness as much as possible. Then, an X-ray machine produces a burst of X-rays that passes through the breast to a detector located on the opposite side (*Figure 1.7*).

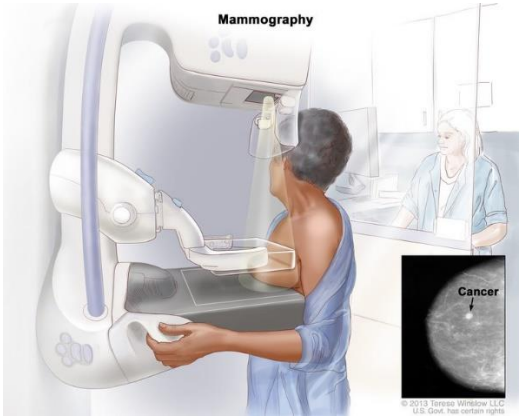


Figure 1.7 *Patient undergoing mammography. Patient's breast is compressed and crossed by X-rays that attenuated from breast tissues and collected from detector generate the image. [www.teresewinslow.com]*

In digital full field mammography, the processes of image acquisition, storage/retrieval, and display are separated. Acquisition is often performed using a Flat Panel Detector (FPD) made by a high-resolution matrix of light sensitive elements (charge coupled devices or thin film transistors), each of which captures and image pixel. Conversion from X-rays to visible light is performed by a thin scintillation layer (frequently, thallium activated cesium iodide, CsI: Tl)). Scatter suppression is normally enhanced by a collimation grid overlapped to the FPD. The natively digital image is output by the FPD, stored on disk, and displayed either on a high-resolution radiological screen or on a film by laser printing. Quantization of signal levels occurs in the analog-to-digital conversion process, during the read-out of the FPD, since the FPD elements are spatially discrete pixels but still storing analogic values in the form of electrical charge. The number

of bits digitization must be adequate to represent subtle difference in X-ray attenuation by tissue over a wide dynamic range of X-ray exposure[21].

However, the method of ex-post digitalization of analogic fluorescence memory plates is still in use, due to the higher flexibility in resolution and precision settings. In place of an RX film, a memory plate is used. This detector, in place of photosensitive AgCl has a thin layer of fluorophore that does not immediately generate fluorescence, but stores energy for a while. Thus, only the stimulation by a laser beam in the dark room of a laser scanner is able to cause the stimulated fluorescence. In this way, the image stored in the memory plate is read-out scanning it by the laser beam, sensing and digitalizing the emission at each spot. Differently from FPDs, resolution and precision are hence determined by the read-out phase settings.

Digital mammograms are high resolved images but there is not an absolute standard for spatial resolution. The minimum pixel size on the detector required for digital mammography has been subject for debate. The size of the pixel element on currently available detectors ranges between 50 μm and 100 μm. Digital mammograms are usually represented with 4096 gray levels using 12 bits per pixel[22], but gray level resolution can change between vendors too. The routine screening mammography includes the acquisition of two standard views for each breast, namely, craniocaudal (CC) and mediolateral oblique (MLO) views, thus providing four images per patient (as shown in *Figure 1.8*).

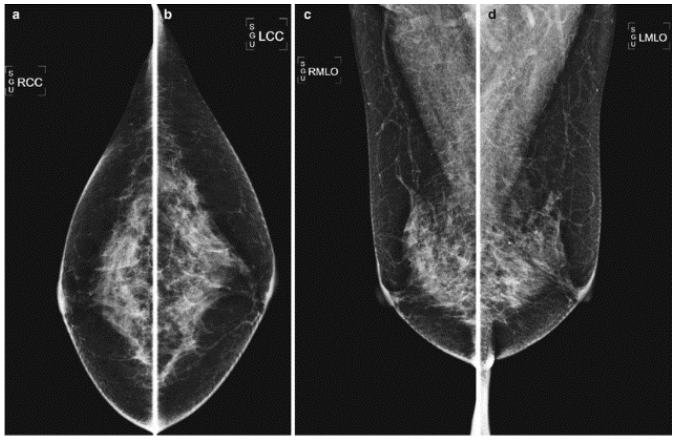
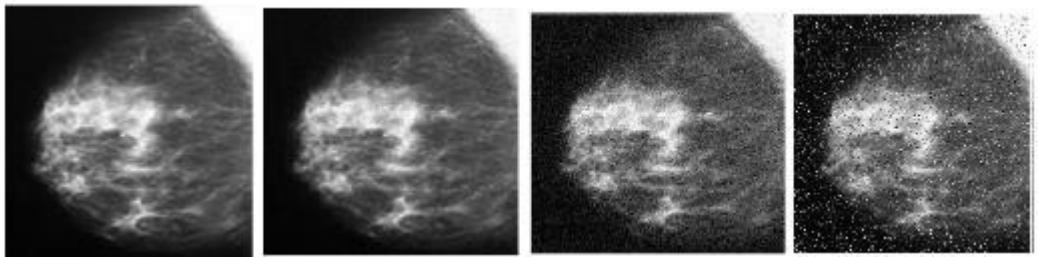


Figure 1.8 *Routine screening mammography standard views. In order: Right CC, Left CC, Right MLO, Left MLO*

Image quality is affected by the size, shape and X-ray absorption properties (i.e., density) of the imaged breast. In addition, X-ray beam quality, geometric un-sharpness and contrast, resolution, detector system noise, and scatter suppression are not standardized in the market and can add additional variability[23].

Salt and Pepper, Gaussian and Poisson Noise are the main types of noises that affect mammogram images. They are the result of different image acquisition processes but altogether can create problems to the fine analysis and interpretation of the breast image leading to wrong diagnosis. On mammographic image formation process the modulation transfer function (Amplitude of Fourier transform of the point spread function, PSF) adds a blurring which degrades the image spatial resolution. The core element is the focal spot size in the X-ray tube anode, the sized of which approximately gives the FWHM size of the PSF. Therefore, in mammographers the focal size is about a half of other X-ray imagers ranging from 0.3 to 0.6 mm. Quantum noise comes from acquisition system low-counts X-ray photons and can be described by a Poisson distribution, which causes random errors called Poisson Noise (*Figure 1.9 [b]*) and is the predominant noise in digital mammograms, given the very low exposure. The electronic noise from digital mammography systems can be modeled as an additive Gaussian noise (*Figure 1.9* [c]). Salt and pepper noise is the consequence of sudden changes of image signal and is frequent in digital imaging systems when the conversion of FPD data to image is quicker[24]. This phenomenon changes some pixels with minimum or maximum intensities randomly, appearing as black and white dots in the image (*Figure 1.9* [d]).



[a] [b] [c] [d]

Figure 1.9 *Mammogram image with noise. [a] Test image, Poisson Noise [b], Gaussian Nose [c], Salt and Pepper Noise [d]. [from[25]]*

1.2.2 Breast anatomy representation on digital mammograms

The breasts are complex structures located on the anterior thoracic wall, in the pectoral region, overlying the chest muscles. Each breast is mainly composed of glandular tissue specialized in milk production, supportive tissue (dense breast tissue), fatty tissue (non-dense breast tissue) but also contains lymph vessels, lymph nodes and blood vessels (*Figure 1.10*). The glandular part named parenchyma includes 15 to 20 sections called lobes arranged like the petals of a daisy. Each lobe has many smaller structures called lobules where milk is produced. Lobes and lobules are linked by tiny tubes called ducts conveying milk to the nipple. There are not muscles in the breast, but they lie between the breast and the chest.

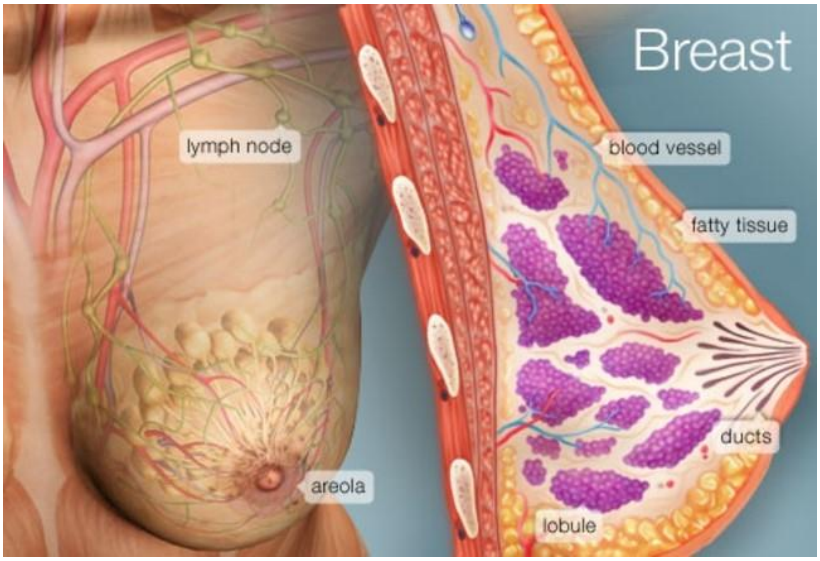


Figure 1.10 *Breast anatomy[www.webmd.com]*

The arterial supply to the breast (*Figure 1.11*) is primarily derived from branches of the internal thoracic artery (a branch of the subclavian artery), intercostal arteries, and the lateral thoracic artery. From the surface, arterial branches of the internal and thoracic arteries arborize across the breast and go deep into the breast parenchyma[26]. The venous anatomy of the breast parallels the arterial anatomy in the deep breast tissues but superficially the venous anatomy is variable and differs from arterial location[27].

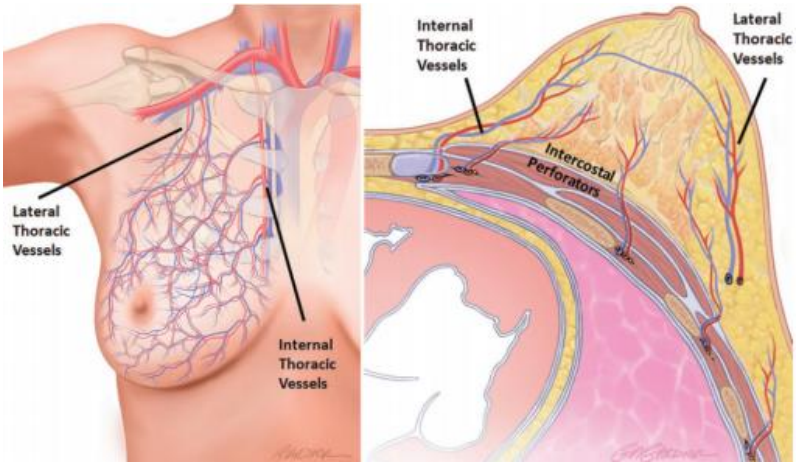
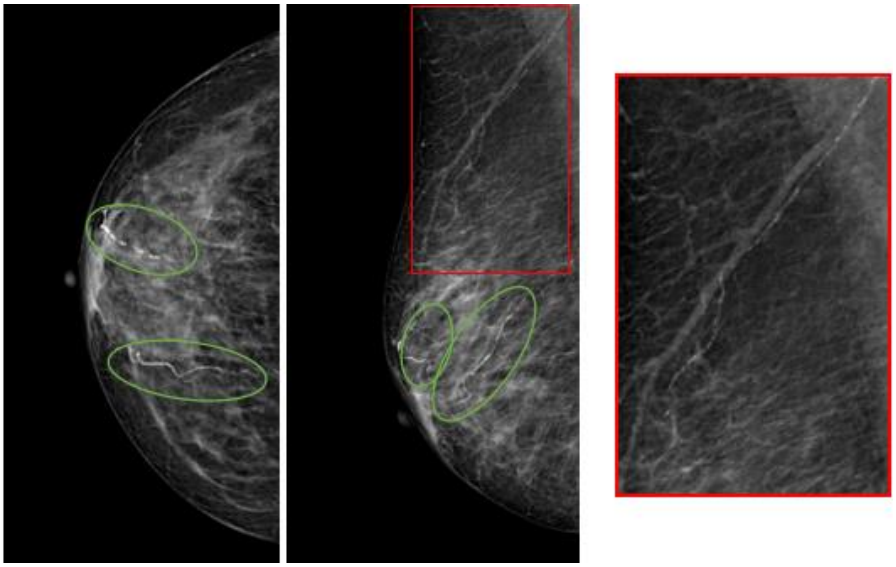


Figure 1.11 *Arterial and venous anatomy of the breast. Paired arteries and veins are found in the perforating branches of the internal thoracic and lateral thoracic vessels. Intercostal vessels perforate the chest wall musculature to supply deeper parenchymal tissues of the breast [from[28]]*

The variability of the breast vascular system visualization given by the variable number of branches of each principal vessel, variable position and the overlap in the 2D mammograms projection, complicates the BACs detection. Combining the two views per breast radiologists can inspect the complex vascular anatomy, which cannot be visible by a single projection thus enhancing BAC detection[29] (*Figure 1.12*).



[a] [b] [c]

Figure 1.12 *BACs on right CC and MLO views. Right CC [a], Right MLO [b] views of the same breast with severe very evident BACs. Not all BACs visible in MLO view (green) are visible in CC view. Signed in red and zoomed [c] BACs visible only in MLO view.*

The mammographic image brightness and contrast also depend on breast density, which decreases with aging. On a mammogram, non-dense breast tissue appears dark and transparent instead dense breast tissue appears as a solid white area, which makes it difficult to see through (*Figure 1.13*). Radiologists quantify breast density as the ratio between non-dense and dense tissues. Usually, there is an inverse relationship between patient age and mammographic breast density[30].

It has been found that women with dense breasts have a four to six higher risk of late breast cancer detection compared with women having no glandular tissue or with little glandular tissue, within the same age range. It is a common belief that one of the reasons could be the masking of lesions by the overlying breast dense tissue making difficult for radiologists detect cancer on early stage [31]. In the same way, BACs can be covered from dense tissue, thus being not visible in mammography. Fortunately, BACs hidden in one view are often shown up in the other one.

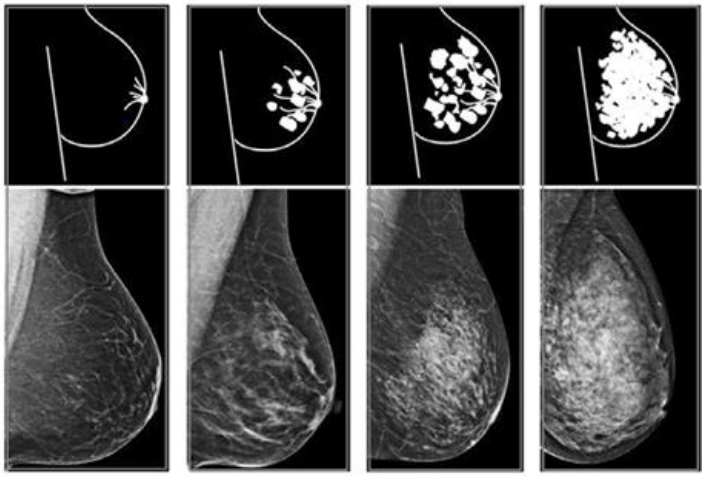


Figure 1.13 *Effect of breast density on digital mammography visualization. From less dense breast (first on the left) to the densest (last on the right) [www.mayo.edu]*

Moreover, BACs are not the only type of calcifications that can be seen on digital mammograms. Hernández *et al*. provide an exhaustive description of calcifications categories according to BI-RADS 5th edition in their article[32]. They report that many of these calcifications have a benign origin such in the case of response of inflammatory disease of ducts or coarse calcifications in benign nodules. Other calcifications can be the caused by malignant disease or high-risk lesion. Since our purpose in this section is to give

an idea of disturbing factors in BAC detection, we will just show some examples among the cases that they reported without specifying and going deep into benignant/malignant origin. They provide a first description based on microcalcification distribution showed in *Figure 1.14.* Every configuration could be a confounding source in BACs detection (for example in the case of a calcified dot that appears in mammography superimposed on a vase but in reality they are located in different panels), but particularly in linear distribution the calcific deposition suggests a deposits within a duct but in some cases is not too easy distinguish between vessels and milk's ducts (*Figure 1.16*). They talk about also dermal, milk of calcium (small particles of calcium oxalate settling within saccular dilatations of the terminal duct lobular units), nodular, intraductal calcifications and here we report some images as examples.

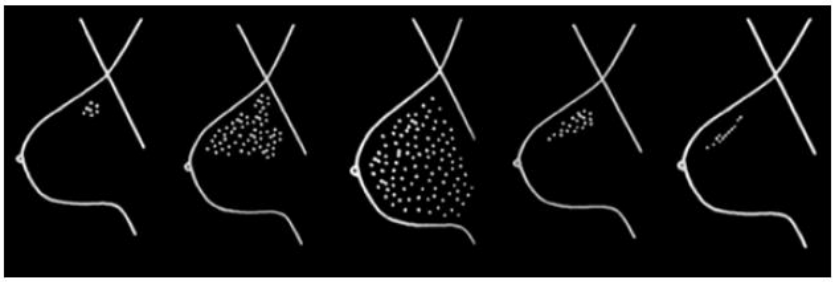
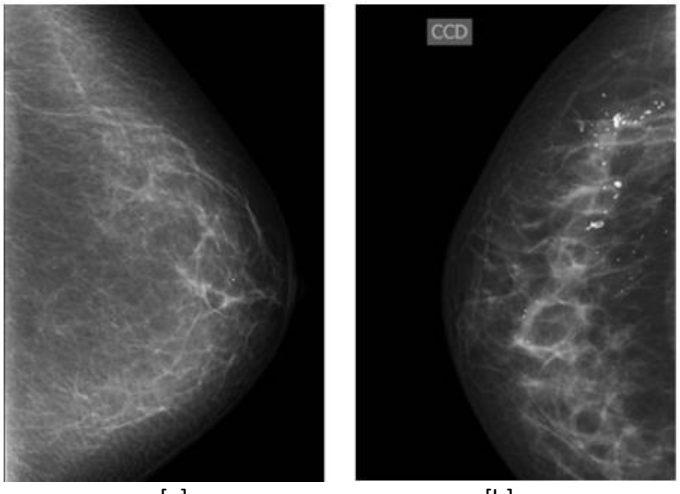


Figure 1.14 *Schematic of the BI-RADS microcalcification distribution descriptors. In order: Grouped, Regional, Diffuse, Segmental, Linear. [from32]*



[a] [b]

Figure 1.15 *Examples of microcalcifications on mammogram. Round microcalcifications diffusely distributed within the breast (little white spots) [a] Regional distribution [b]. [from[32]]*

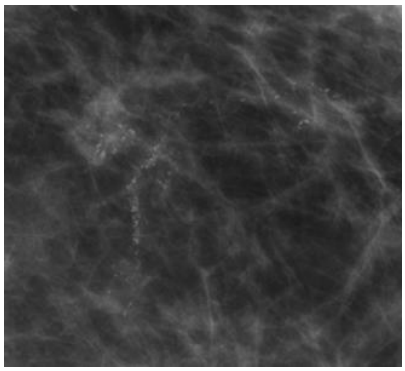
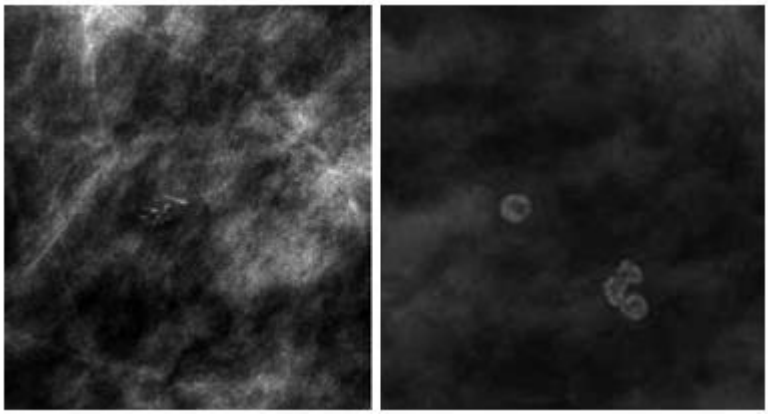


Figure 1.16 *Example of microcalcifications in linear distribution. [from[32]]*



[a] [b]

Figure 1.17 *Dermal [a] and milk of calcium [b] calcifications. [from[32]]*

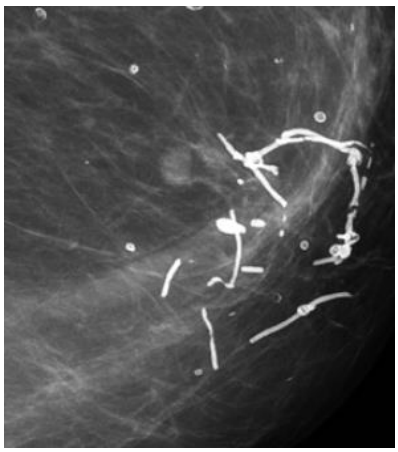
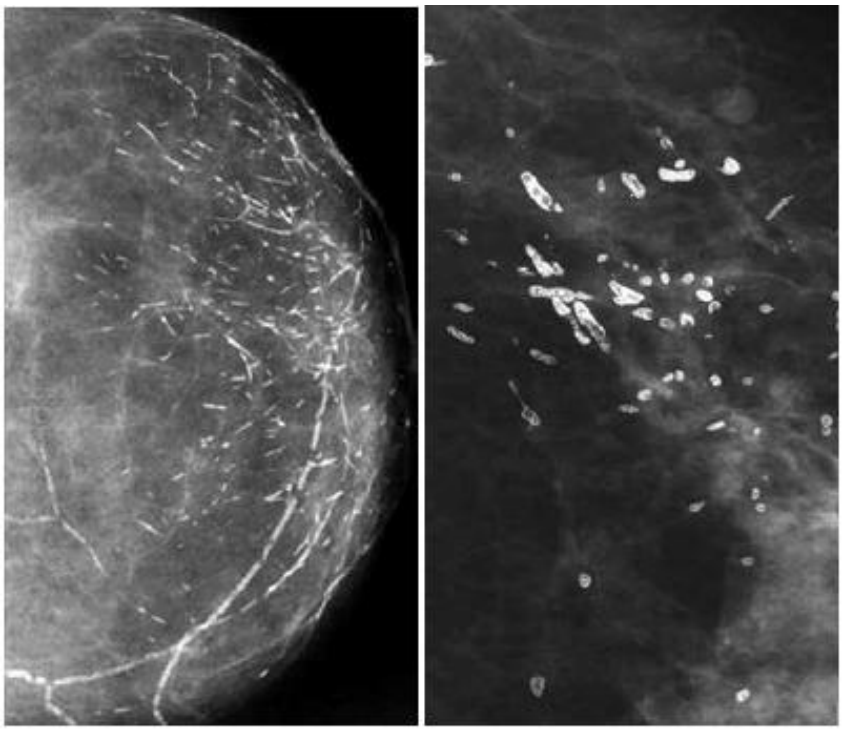


Figure 1.18 *Suture calcifications. Calcification forming knots. Linear or tubular calcifications that may present knots. Common in patient who have undergone radiotherapy.[from[32]]*



[a] [b]

Figure 1.19 *Examples of thick linear calcifications. [a] Originating within a duct. Vascular calcifications are present too but is difficult to distinguish between them. [b]Originating in the duct wall. [from[32]]*

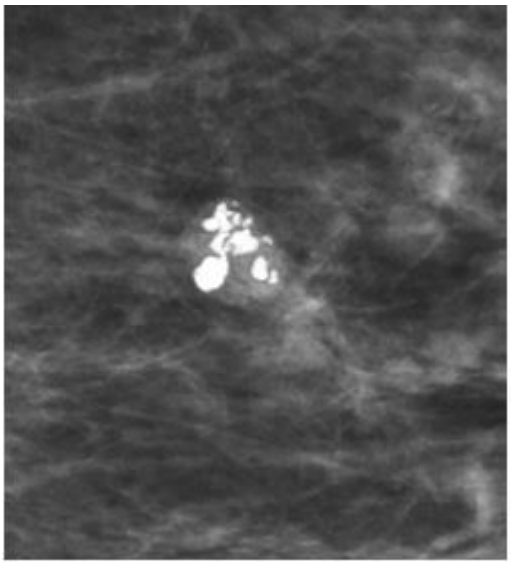


Figure 1.20 *Popcorn calcifications. A nodule with coarse calcifications. [from[32]]*

1.3 Detection and quantification of breast arterial calcifications on digital mammography

Achieving an effective method of BACs assessment is not an easy task. Many scientists in the literature have tried to make their own version, but none of them has proved to be eligible as standard method to be used by clinicians. One of the biggest problems is surely the heterogeneity in BACs appearance in mammographic images. They can have different appearance patterns (tubular, single or parallel structures, little bright spots) and added to the topological complexity and vessels overlapped on two-dimensional projections make both BACs identification and quantification real challenges[33].

Many studies described BACs on a dichotomous scale[34 35] , other in a semi-quantitative scale[18 36] but there are few cases of quantitative scale, too. Part of quantitative methods are based on manual segmentation measure[37 38] , which is time consuming and operator dependent. Aiming at including BAC assessment within a screening test, it is very important to minimize the reporting time and, even more important, operator-dependency of the manual segmentation process[2].

Recent studies have focused on the realization of automatic BAC segmentation methods, addressing operator-dependency by using multiple readers to establish the reference standards[33 39]. In some case, BACs are segmented, just to exclude them from the microcalcification detection for breast cancer diagnosis. In this light, several algorithms have been proposed for automated BAC detection. For example, Mordang *et al.*[40] used a GentleBoost classifier to remove BACs from mammograms in microcalcification detection by a set of manually designed features obtaining a reduction of the number of false positives per case by 29% on average. In another study, Cheng *et al.*[16 41] developed a twosteps fully automatic algorithm. In the first step with a random walk-based tracking algorithm they found BAC paths, then with the second step based on a linking algorithm they grouped BAC paths into BACs. Their proposed method was tested on 40 mammograms and achieved performance of 93.8±1.3% in sensitivity and 84.7±3.9% in specificity.

Despite these efforts, the automatic BAC detection is far from clinical deployment. There is the need to go deeper and invest in a method that should: i) account for the diversity in shape, location, appearance and prevalence of BACs; ii) be insensitive to the variability of different machines; iii) not suffer the influence of other complex or abnormal structures of the breast that appear intense in mammograms; and iv) eliminate operator dependence.

1.4 Deep learning and breast arterial calcification

From the first appearance in 2006[42] as a new field of research field within machine learning, Deep Learning (DL) algorithms have been extensively applied in image analysis [43] [44]. The versatility of this approach has allowed to tackle a wide range of task, ranging from pattern classification and detection in natural images[45], to a system to screen coronavirus (COVID-19) disease pneumonia in the currently worldwide spread of pandemia[46].

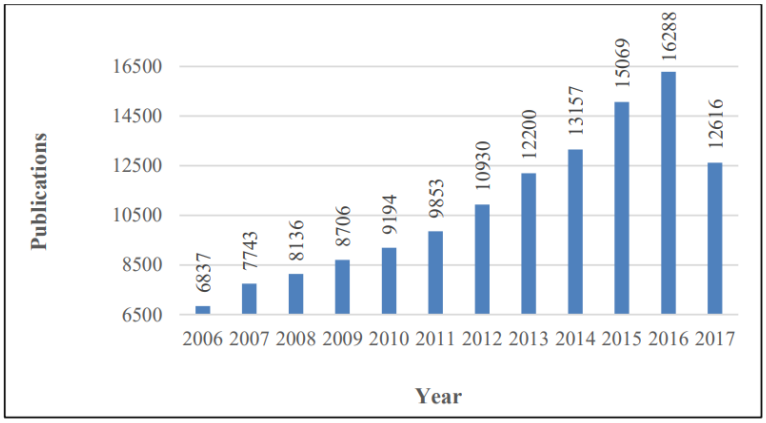


Figure 1.21 *Growth of the number of publications in Deep Learning, Sciencedirect database (Jan 2006-Jun 2017) [from [42]]*

Convolutional Neural Network (CNN) is one of the successful and popular deep learning techniques to perform image classification allowing to outperform traditional classifier in several cases[47].The main strength of CNN is that it can automatically perform both the feature extraction and classification tasks by a fully data-driven strategy, so that there is no need for manual feature extraction and selection[48].

Due to their peculiar characteristics CNNs have been applied also to BACs detection. In a recent study, Wang et al.[39] investigated the potential of deep learning for BACs detection on mammograms. To date, and to the best of our knowledge, this study is the only documented case of deep learning application for BAC detection. In their study, the authors developed a 12-layer convolutional neural network to discriminate BACs from non-BACs pixels using a pixelwise, patch-based procedure for BACs-detection and segmentation. They asked to expert radiologists to manually provide the boundaries of BACs in mammograms, then they extracted from these images a number P of batches of size 95×95 pixels from BAC-regions and a number P of batches of same size from non-BAC regions. The image patches were fed to the CNN trained to provide the probability of the central pixel to belong to the BACs class or not. They tried to address the operator-dependency asking multiple readers to establish the presence and location of BACs and using the boundaries provided by the most experienced reader. Known that the determination of BACs boundary is considerably more subjective than BACs location and they used segmentations provided from only one reader to train the net, as a consequence this approach is not fully operator independent.

THESIS AIM

The first aim of this thesis is to implement a binary classifier of mammographic images to discriminate the presence/absence of BACs using pretrained CNN; i.e. the manual annotation used for training, validation, and possibly testing and the trained CNN output is dichotomic: positive ($BACs^+$) to the presence of at least one BAC, negative ($BACs^-$), no BAC detected. The second purpose is to verify the support to manual segmentation of the main BAC (or BACs) offered to the radiologist in $BACs^+$ cases, thanks to the focus on a limited breast region provided by the CNN heat/saliency map.

Training a CNN on a database with whole image annotation, i.e. the single dichotomous annotation $BACs^+/BACs^-$ is the proposed strategy expressly chosen to reduce operator-dependency and to obtain a less biased result. Here we want to investigate the feasibility of the proposed method and find the network structure and transfer-learning approach that best suits the problem.

2. Methods

2.1 Deep learning and Convolutional Neural Networks for binary classification

Machine learning (ML) is a term introduced by Arthur Samuel in 1959 to describe an artificial intelligence (AI) subfield [49]. ML includes all those approaches that allow computers to learn from data without being explicitly programmed. Deep learning (DL) has emerged as one of the most promising machine learning techniques (*Figure 2.1*). DL methods belong to representation-learning methods with multiple levels of representation, which process raw data to perform classification or detection tasks[50] .

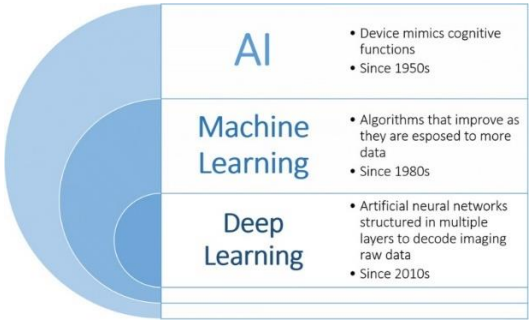


Figure 2.1 *Artificial intelligence, machine learning and deep learning relationship. [From[51]]*

Neural network architecture is structured in layers composed of interconnected nodes. This structure mimics the interconnection of natural neurons, which perform summation of inputs each weighted by the relevant synapsis strengths, next firing if and only if a threshold is reached. So, each node of the artificial neural network (ANN) executes a weighted sum of the input data that are subsequently passed to a highly nonlinear activation function. Weights are dynamically optimized during the training phase, similar to the long-term-potentiation/depotentiation of natural synapses. We can distinguish between three different kinds of layers: the input layer, which receives input data; the hidden layer(s), which are in charge to extract the patterns within the data; and the output layer, which provides the processing results. Deep neural networks were introduced to improve on the performance of conventional ANN adding many hidden layers, which characterize the depth of the network. In deep learning multiple linear and non-linear processing units are arranged in deep architectures to model high level information abstraction present in the

data[52]. There are several deep learning techniques including auto-encoders, restricted Boltzmann machines, deep belief networks[53] and deep convolutional neural networks (CNNs).

CNNs are feedforward networks in which the information flow takes place in one direction only. Their architecture is biologically inspired by the visual cortex of the human brain, which consists of alternating layers of simple and complex cells[54]. It is worth recalling that convolution is the main signal or image processing step, which simply consists of a scalar product (point-by-point multiply and overall summation) of data (in this case the CNN input or the output of the previous layer) with a set or fixed coefficients (in this case, the synaptic weights). Most feature extraction methods (Fourier, filtering, cross-correlation, matched filtering, texture analysis, etc.) are actually based on convolutions. CNNs architectures may strongly vary among different tasks but are generally composed by convolutional and pooling (or subsampling layers) steps grouped in blocks stacked on top of each other to form a deep model. Interestingly, pooling and next un-pooling implements the multiscale approach, which in the traditional signal/image processing is implemented by wavelet techniques, with self-similar convolutions done at different scales. Stacked modules are always followed by one or more fully connected layers, as in standard feedforward neural network. In *Figure 2.2* an illustration is shown of a most popular basic CNN architecture for a toy image classification task provided from Rawat *et al.* in their review[53]. An input image is passed to the network and is processed through different convolutional and pooling stages. Then, representations of image content (i.e. extracted image features) are processed by one or more fully connected layers. The last fully connected layer (output layer) provides an estimate of the input image class label.

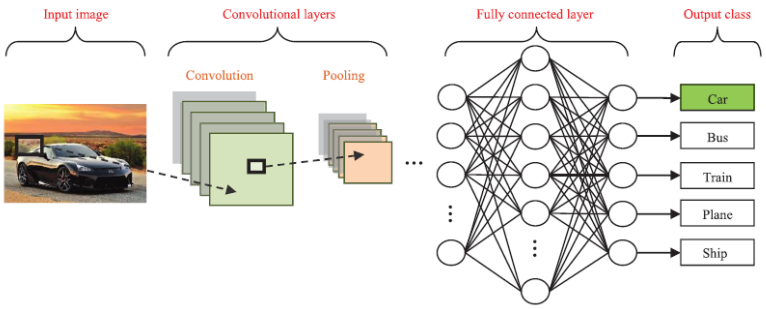


Figure 2.2 *Convolutional Neural Network classification pipeline. [From [55]]*

Convolutional layers

The Convolutional base of a CNN works as feature extractors. It aims at learning the feature representations of their input images. The neurons in the convolutional layers are organized in feature map and each neuron has a receptive field, which is connected to a neighborhood of neurons in the previous layers via a set of weights arranged in a matrix called kernel. Inputs are convolved with the kernel weights to compute new feature maps. Each convolved result is then summed to a bias value and sent through a non-linear activation function that typifies the neuron and allows the extraction of nonlinear features (*Figure 2.3*).

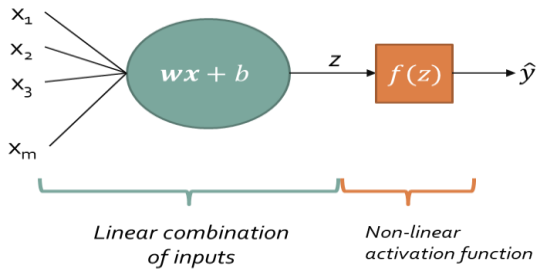


Figure 2.3 *How a neuron processes inputs to obtain its output.*

Each feature map is the results of the application of a specific kernel, nevertheless the same convolutional layer contains multiple filtering kernels thus representing a high-dimensional feature space. Each neuron is characterized by a specific activation function. Traditional activation functions are the sigmoid and hyperbolic tangent functions defined, respectively, as:

$$f(x) = \frac{1}{1+e-x} \qquad \textit{Equation 2.1} \qquad , f(x) = \tanh(x) \qquad \textit{Equation 2.2}$$

Where f is the neuron output as a function of its input of x (convolution result plus bias). Sigmoid activation function looks like a S-shape (*Figure 2.5*) and its output ranges between 0 and 1. We can distinguish three regions: zero-saturation region, linear region and one-saturation region.

Recently, rectified linear units (ReLU) and their variants became popular, due to their simplicity and efficiency. Introduced by Nair and Hilton in 2010 [56] the ReLU is a piecewise linear function defined as:

$$f(x) = \max(x, 0) \quad \textit{Equation 2.3}$$

i.e. it retails only the positive part of the activation by reducing the negative part to zero, promoting faster computations. It was demonstrated that ReLU leads to faster convergence [56] and not to suffer from the vanishing gradient problem, in which the lower layers have gradients near to zero because of the saturation of higher layers, in the back-propagation algorithm [57]. ReLU shows possible disadvantages during the optimization process since the gradient is zero when the unit is not active (the derivative of the ReLU is 1 in the positive part and 0 in the negative part) [58] .This may fall to sub-optimal solutions of training where not activated neurons will be never retrieved. Thus, ReLU can lead to slow convergence of the training process when gradients are constant near to zero. Maas *et al.* [58] tried to solve this problem by introducing Leaky Rectified Linear Units (Leaky ReLU), a variant of traditional ReLU that allows for small nonzero gradients when the unit is not active. Leaky ReLU is defined as follow:

$$f(x) = \max(x, 0) + \lambda \min(x, 0) \quad \textit{Equation 2.4}$$

Where λ is a predefined parameter within the range (0,1).

Pooling layers

Pooling layers have the purpose of reducing the feature maps spatial resolution and achieve spatial invariance to input scale changes and distortions. Pooling create partitions of the input image into a set of non-overlapping sub-regions and presents as output only one value per partition, calculated according to a specific rule. *Figure 2.4* represents two of the most used pooling types: Max pooling and Average pooling. Max pooling aggregation layers propagate the maximum value within a receptive field to the next layer, while Average pooling ones propagate the average value. Max pooling followed by the activation function is similar to a non-exclusive OR operations (alias, winner-takes-all), while Average pooling has a smoother, more linear behavior.

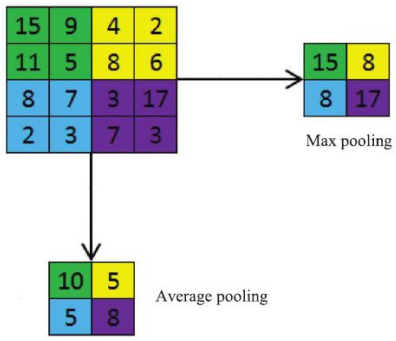


Figure 2.4 *Max pooling and Average pooling with a 2x2 filter. [From [59]]*

Fully connected layers

Fully connected layers follow the convolutional and pooling layers stacked on top of each other. Fully connected layers interpret these features and perform the function of high-level reasoning [60]. For classification problems, it is a standard solution to use the SoftMax operator [60] on top of a deep CNN. Nevertheless, there are other possible alternatives like radial basis functions or support vector machine. In binary classification task problems, a simple widely used solution is to stack on the top of the CNN a further one neuron layer with a sigmoid activation function. To perform a crispy binary classification a threshold between 0 and 1 is set. In *Figure 2.5* is shown an example of binary classification using Sigmoid activation function and threshold equal to 0.5. Images with CNN output under 0.5 are associated to class 0, otherwise to class 1.

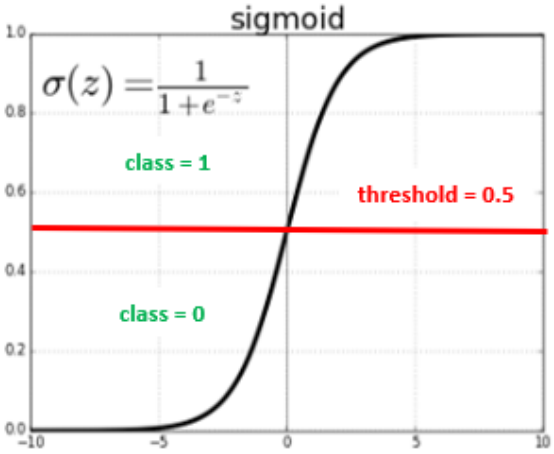


Figure 2.5 *Binary classification using Sigmoid activation function.*

Training

In order to obtain the desired network output, CNNs use learning algorithms to adjust their free parameters (i.e. weights and biases). Backpropagation is the most common algorithm used for this purpose [61] [62] [63]. Backpropagation computes the gradient of an objective function (loss function) to determine how to adjust network parameters to minimize errors that effect its performances. In binary classification problem the most widely used Loss function is Binary Cross Entropy.

Cross-Entropy Loss is defined as follows:

$$CE = -\sum_i^C y_i \log(\hat{y}_i) \qquad \textit{Equation 2.5}$$

Where y_i and $\hat{y}_i$ are the ground truth and the CNN estimate for each class i in C respectively.

Since in Binary Cross-Entropy we have two classes, *Equation 2.5* becomes:

$$BCE = -\sum_i^{C=2} y_i \log(\hat{y}) = -t_1 \log(\hat{y}_1) - (1 - y_1)\log(1 - \hat{y}_1) \qquad \textit{Equation 2.6}$$

In order to adjust the free parameters using the loss function gradient, an optimization algorithm needs to be chosen. The most basic but used one is Gradient descent optimizer [64]. It is a first-order optimization algorithm which is directly dependent on the loss function gradient and the learning rate.

Importantly, given the CNN structure and the optimization algorithm, still some parameters of the latter are free and must be optimized, such as: the learning rate, the number of iterations (epochs), batch size, hidden layers, hidden units, activations functions and related parameters, possible optimizer parameters, parameters related to regularization methods etc. In order to distinguish these few figures from the high number of free parameters to be optimized upon the training dataset, the former ones are called hyperparameters. Learning rate is a very important hyperparameter because it determines the rate and speed of the learning process and is present in all optimization algorithm. Various optimizers have been proposed such as RMSprop, Adam, AdaDelta, AdaMax,

Adagrad, Nadam. Each of them has advantages and disadvantages compared to the others. For a description, please refer to the review [64] .

A common problem with training CNNs is overfitting, which means that the model poorly performs on a data that do not belong to the training dataset. Overfitting affects the model ability to generalize on unseen data and is a major challenge for deep CNNs given the high number of free parameters, unless a huge training-set was available. Usually, the first step in developing a neural network model is to divide the dataset into three subsets, namely training, validation and test datasets [65] .The training dataset is used to train the network. To prevent overfitting, the performance is simultaneously monitored and validated for an independent dataset, namely the validation dataset. As the training goes on, the performance of the network is continuously improved for the training dataset, if the same trend does not occur in the validation loss function it means that the model overfits the training dataset. Monitoring the validation Loss value at each epoch allow to visualize when overfitting occurs. One epoch is when an entire dataset is passed both forward and backward through the neural network only once. An increase in validation loss value in several successive epochs identifies overfitting and is conventionally used as a condition to stop the training process. Finally, testing dataset is used to prove model generalizability on unseen data.

Evaluation metrics

When CNN are used to develop binary classifiers, validation and testing performances and their improvements are usually evaluated using common classification metrics derived from the confusion matrix.

		Estimate	
		Positive	Negative
Real	Positive	**TP**	**FN**
	Negative	**FP**	**TN**

Table 2.1 *Confusion matrix.*
TP: true positives; TN: true negatives; FP: false positives: FN: false negatives

Table 2.1 describes the confusion matrix, where each element is defined as:

- True positives (TP): number of $BACs^{+}$images correctly classified
- False negatives (FN) : number of $BACs^{+}$images incorrectly classified
- True positives (FP): number of $BACs^{-}$ images incorrectly classified
- False negatives (TN) : number of $BACs^{-}$ images correctly classified

Accuracy is the most common evaluation metric used for the evaluation of most traditional DL classifiers. It is defined as the number of correct predictions divided by the total number of predicitons:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \qquad \textit{Equation 2.7}$$

However, accuracy can be misleading while evaluating imbalanced data sets because it can be biased by the classification results perfomed on data belonging to the majority class, making it difficult for a classifier to performe well on the minority class [66]. So, other metrics were proposed for handling imbalanced data sets like precision, recall and F1 score.

Precision and recall are defined as follows:

$$Precision = \frac{TP}{TP+FP} \qquad \textit{Equation 2.8}$$

$$Recall = \frac{TP}{TP+FN} \qquad \textit{Equation 2.9}$$

High precision means that the algorithm returns more relevant results than irrelevant ones (so it is a measure of model reliability), while high recall means that it returns most of the relevant results (it's a measure of model ability in detecting a specific class). Precision and recall are often in conflict and improving recall reduces precision and vice versa. So, the trade off between performance metrics is choosen based on the application domain (e.g. cancer identification, predicting truck driver accidents etc.).

F1 score is the harmonic mean of the precision and recall (*Equation 2.4*). Therefore, FP and FN are equally costly. F1 score ranges between [0,1]. F1 score represents a better evaluation metric compared to accuracy when prior probabilities are very between classes [67].

$$F_1\ score = 2 * \frac{Precision*Recall}{Recall+Precision} \qquad \textit{Equation 2.10}$$

Another largerly used evaluation tool is the receiver operating characteristic (ROC) curve. ROC curve is a graphical plot that illustrates the classifier performaces as its discrimination threshold is varied. It plots the true positive rate (TP rate) vs the false positive rate (FP rate) at various threshold setting .TP rate is also known as classifier sensitivity and correspons to the already introduced recall. On the other hand, the FP rate is defined as folloows:

$$FP\ rate = \frac{FP}{FP+TN} \quad Equation\ 2.11$$

FP rate denotes the percentage of the misclassified negative examples, and TP rate is the percentage of the correctly classified positive examples. The point with coordinates (0,1) in the ROC space represents the ideal point (perfect performances) (*Figure 2.6*).

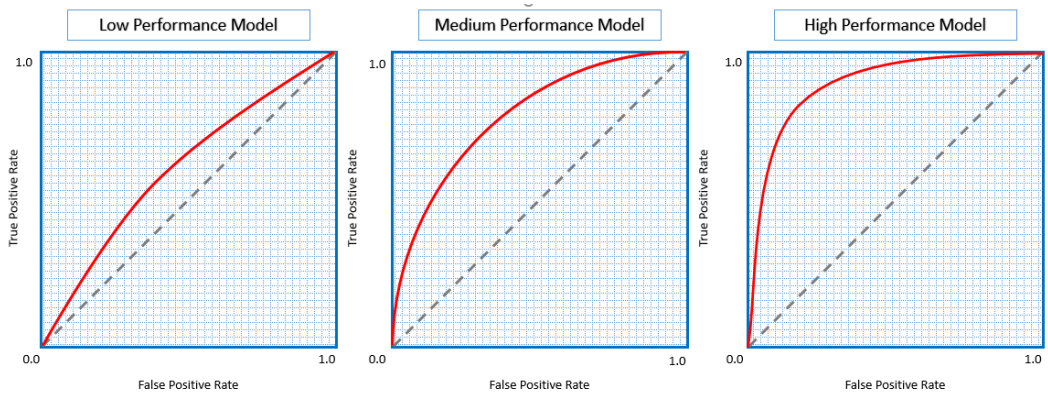


Figure 2.6 *Example of ROC curves.*

ROC area under the curve (AUC) is obtained calculating the area under the ROC curve. The AUC is one of the several different tools introduced to evaluate model performances but represents one of most used metrics. Since the ROC AUC is a portion of the area of the unit square, its value will always vary between 0 and 1, where AUC of one represents a model with perfect discrimination performances while a model with AUC of 0.5 has no discriminant ability [68]. However, ROC AUC, although widely used even in cases of unbalanced datasets, does not place more emphasis on one class over the other, so it does not well reflect model performance in the minority class.

2.2 Network-based transfer learning

Transfer learning helps in solving the problem of an insufficient training-set specific to the target problem (i.e., BACs in mammograms). A previous training is performed on a more general problem (e.g., general object recognition, for which wide annotate data-bases are already available, thus transferring the knowledge from the source domain to the target domain by relaxing the assumption that the training data and the test data must be independent and identically distributed (*Figure 2.7*). For this reason, transfer learning has recently successful been used in various deep learning applications and CNNs already trained in the source domain are available [69] [70]. The driving hypothesis is that the source domain recognition and the target one share a major set of common features, the extraction capability of which can be usefully transferred from the former to the latter.

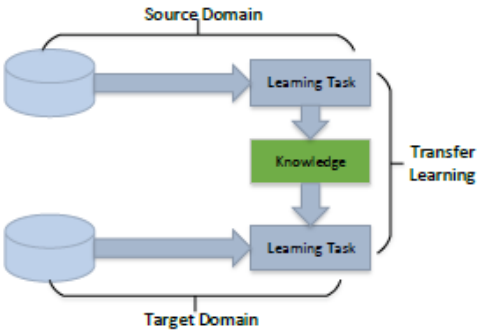


Figure 2.7 *Learning process of transfer learning. Knowledge transferred from source to the target domain in order to solve the target task. [from [70]]*

Tan *et al.*[70] in their survey summarize deep transfer learning into four categories:

- Instances-based deep transfer learning
- Mapping based deep transfer learning
- Network-based deep transfer learning
- Adversarial-based deep transfer learning

We will discuss deeply the "Network-based deep transfer learning" approach, since it is the one that we applied in this work.

Network-based deep transfer learning entails the reuse of part of a network pre-trained in the source domain, including its network structure and connection parameters, and transfer it into the deep neural network used in the target domain (*Figure 2.8*). It is based

on the assumption that neural network works like the human brain, that is an iterative and continuous abstraction process.

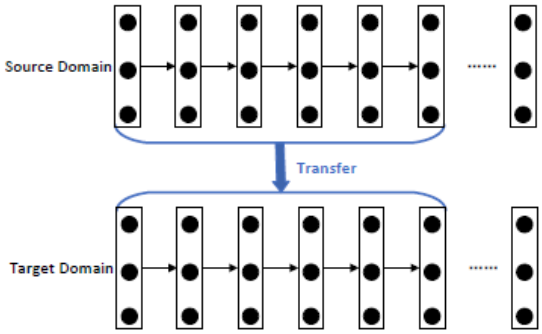


Figure 2.8 *Sketch map of the network-based deep transfer learning. Part of the network trained in source domain with large-scale training dataset is transferred to be part of a new network designed for target domain. [from [70]]*

ImageNet [71] [72] is a research project aim to develop a large database of annotated natural images to build "a useful resource for researchers, educators and students and all you who share our passion for pictures". It includes millions of 224x224 pixels RGB natural images labelled according to 1000 classes (Figure 2.9). ImageNet database allows to perform the ImageNet Large Scale Visual Recognition Challenge which invites researchers from all over the world to challenge each other to create object detection and image classification algorithms. Those challenges led, from 2010 onwards, to the development of well-known deep CNNs architectures like AlexNet [73], GoogLeNet [74], VGG16 [60] , ResNet [75], which were built from scratch and trained over the ImageNet database. As the models are trained on a large dataset, they learned a good representation of low level features like edges, spatial, rotation and shapes that can be shared to enable the knowledge transfer and act as low-level feature extractor for new images in different computer vision problems [76]. Abovementioned deep CNNs architectures and corresponding weights are open source and available as Python libraries.

Figure 2.9 *Example of images and labels of ImageNet dataset. [from[77]]*

Due to the limited amount of data available in the biomedical field, transfer learning from natural image datasets has become a *de-facto* method for deep learning applications to medical imaging [78]. However, there are fundamental differences in data-size, features and task specifications between natural image classification and the target medical tasks. Current standard practice involves using an existing architecture pre-trained on natural images and then to fine-tune it on medical imaging data. Fine-tuning process begins with the initialization of target network parameters using those learned by the pre-trained network except for the last fully connected layer, whose node number depends on the number of target classes [59]. Indeed, the last fully connected layer is strictly task-related and must be built up and trained from scratches. In literature there are examples of successful applications obtained by training only the fully connected part of the network maintaining the pre-existed part fixed [79] [80] ,in other cases it was also necessary to fine-tune all convolutional layers [81] [82] .Generally, the early layers of a CNN learn low level image features applicable to most vision task while in the late layers the network learns high-level features specific to the application at issue *(Figure 2.10).* So, the number of convolutional layers to fine-tune depends on the significance of the distance between the source and target applications. Shin *et al.* [59] suggest that an effective fine-tuning technique is to start from the last layer and then incrementally include more layers until the desired performance level is reached.

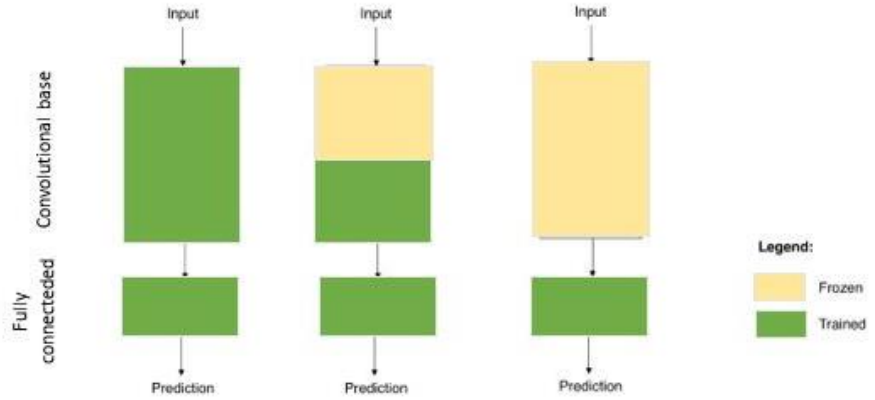


Figure 2.10 *Different fine-tuning strategies. [a] Train both convolutional and fully connected layers. [b] Train fully connected layers and high-level specific task convolutional layers. [c] Train fully connected layers only.*

3. Protocol

3.1 System model

We used Python 3.6 language and open-source software libraries including:

- TensorFlow-gpu 1.13.1: TensorFlow is a widely used for machine learning applications such as neural network. In its GPU version supports running computations with GPU (Graphic Processing Unit), electronic circuit designed to rapidly manipulate memory to accelerate the creation of images in a frame buffer intended for output to a display device.
- Keras 2.3.0: Keras is a library that works as wrapper and it is a high-level neural network library that's build on the top of TensorFlow
- Scikit-image 0.14.2
- Scikit-learn 0.22.1
- Numpy 1.16.5
- OpenCV

All trainings were performed using NVIDIA GeForce RTX 2080TI (11GB on-board memory)

3.2 Dataset

For this project, we used a retrospectively selected database of mammograms that belongs to women who underwent screening mammogram at the IRCCS Policlinico San Donato (PSD) from January 2nd, 2018 to February 8th, 2018. The database included 719 mammographic exams belonging to 719 women patients, accounting for 2876 DICOM (Digital Imaging and Communications in Medicine) images format (1438 CC and MLO projections for each breast). This retrospective, single center study focused on a subgroup of women included in a larger retrospective monocentric research project approved by the local Ethics Committee (Ethics Committee of IRCCS Ospedale San Raffaele; protocol code SenoRetro; approved on November 9th, 2017 and amended on July 18th, 2019.

For privacy protections, all images and the accompanying information was pseudo-anonymized by patient identity coding known only by one of the clinicians in charge at PSD. Hence, the unlikely event of back trancing a subject was kept in the clinician hands, exclusively for clinical purposes, under the responsibility of the curing MD. All non-useful demographic data were not transferred. Also, the date of examination was unknown to us. Given the fairly large group of subjects, the pseudo-anonymization can be considered close to full anonymization. Due to low image quality 2 subjects were excluded from the dataset. The final image dataset accounted for 2868 images (1434 MLO and CC views respectively) that belong to 717 women.

Data were acquired from three different devices. *Table 3.1* reports the main specifications and the number of mammograms acquired with each of them. The image matrix size varies even when the same machine is used. Number of columns varies in a range from 2368 to 2784, while number of rows is fixed for every machine and is reported in *Table 3.1*.

	Number of acquisitions	**Spatial resolution[µm]**	**Number of gray levels**	**Image rows**
Device 1	586	81.4	16384	3580
Device 2	129	81.4	16384	3584
Device 3	2	100.0	4096	2850

Table 3.1 *Acquisition systems and images properties.*

Three expert readers (R1, R2 and R3) labeled available images as $BACs^{+}$or $BACs^{-}$. At patient level, a single $BACs^{+}$ image of the four, implied the patient classification as $BACs^{+}$. R1 is a medical student adequately trained for BACs identification by a breast imager with a ten-year experience, R2 is a radiology resident with a three-year experience in reading mammography, R3 is a PhD student with a medical degree and one-year experience in BACs identification. R1 and R2 labelled the images at patient-level and provided information about BACs laterality. Disagreement were solved asking for the opinion of a fourth external reader (a breast radiologist). R3 performed a second level screening re-examining the single images of patients labelled as $BACs^{+}$ by R1 and R2, in order to detect BACs in further views. *Figure 3.1* shows one example of patient and image labeling. At the patient level, even if BACs are present in one breast, the patient is marked as $BACs_p^{+}$ (i.e. class = 1) if at least one projections present BACs, otherwise $BACs_p^{-}$ (i.e. class = 0), where p represents the code of a specific patient. At image-level each mammographic view was labeled as $BACs^{+}$ if showing at least a single BAC (see *Figure 3.1*).

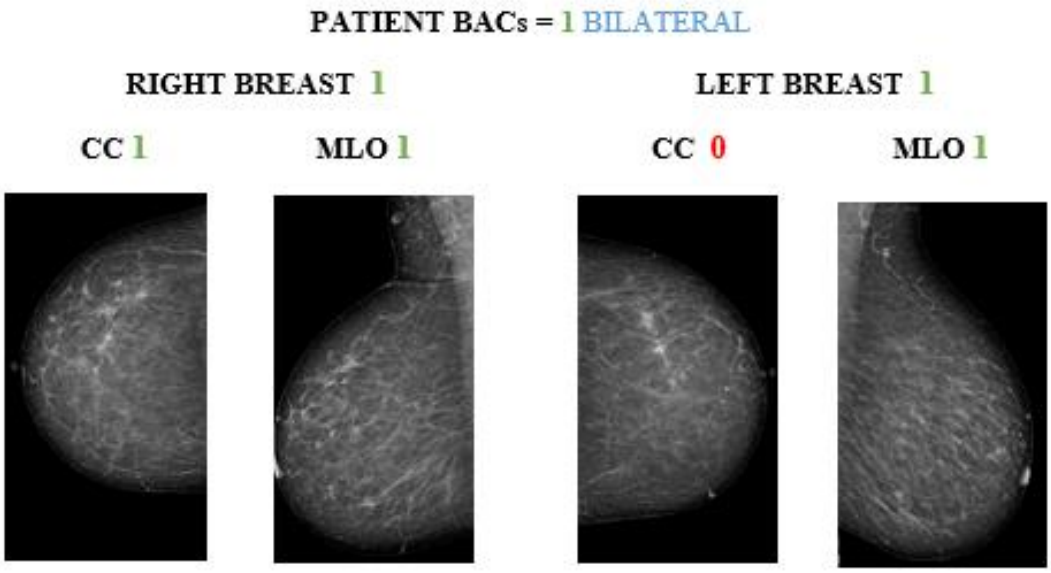


Figure 3.1 *Example of image and patient labels. CC and MLO views of right breast and MLO view of left breast are labelled as BACs = 1. Left MLO is labelled as BACs = 0 because is not visible in the image. BACs are present in both breasts, so, at the patient level we have a label BACs = 1, bilateral.*

The average age of the patients is 60.2 ± 9.0 years (average ± standard deviation) and women with BACs are on average older (66.7 ± 9.3 years) than woman without BACs (59.2 ± 8.5 years). Distribution of patients' age according to BACs class is showed in *Figure 3.2.*

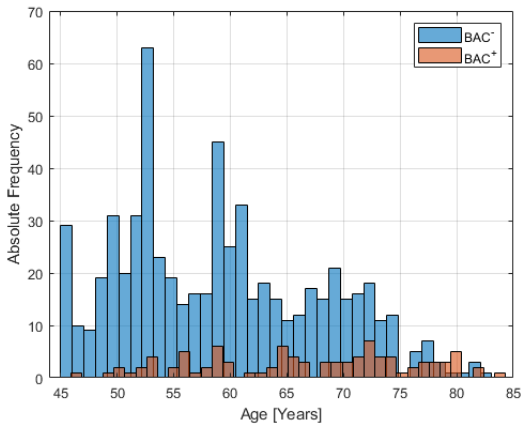


Figure 3.2 *Distribution of patients' age per BACs classes. Histograms of patient distribution according to age of women with and without breast arterial calcifications.*

Table 3.2 and *Table 3.3* report respectively a summary of labels per patient and image level. BACs prevalence per patient is 13.24% and falls within the standards found in literature[13] [14]. 7.53% are women with bilateral BACs and 5.72% with unilateral BACs. BACs prevalence per image is 10.21 %

	BAC+ total	**BAC+ unilateral**	**BAC+ bilateral**	**BAC-**	**Total patients**
Frequency	95	41	54	622	717
Prevalence [%]	13.24	5.72	7.53	86.76	100.00%

Table 3.2 *Labels per patient. Number of patients with or without breast arterial calcifications in our dataset.*

	BAC+	**BAC-**	**Total images**
Frequency	293	2575	2868
Prevalence [%]	10.21	89.79	100.00%

Table 3.3 *Labels per image. Number of images labelled as presenting BACs or not in our dataset.*

3.3 Preprocessing

Preprocessing was performed in two steps: data preparation on the whole dataset, ahead of any DL step (training, validation), and online data augmentation by resampling on images included in the training set.

Data preparation

During data preparation, images were cropped and saved into a hdf5 file in single-precision floating-point format. Cropping aimed at extracting the region of interest (ROI) from original mammograms, to rescale the extracted ROI to a fixed shape. Intensity histogram normalization was performed after the cropping, to avoid biases from empty regions.

In order to be processed by the CNN, all images must have the same number of pixel rows and columns that define the size of the input layer. The image size was set to 1536x768 pixels to preserve information content related to BACs (which may be very small) respect hardware capability. Furthermore, we set the matrix number rows and columns multiple of 32 dues to the used architecture, further details will follow. As a consequence, small breasts had a higher magnification than large ones. Nonetheless, this was hypothesized to not be a confounding factor, since the CNN training had in any-case to deal with different BAC dimensions (whether real or apparent). Also, sizing of BACs was not under the scope, in which case ex-post rescaling would be very simple.

Pixel intensity normalization allows to improve the convergence of learning process and represents a very common practice in deep learning image preprocessing. This process focuses relative contrasts, rather than absolute intensities, highly influenced by the image acquisition device and protocol, as well as by breast size and density.

Figure 3.3 shows processing steps applied on one mammogram and *Figure 3.5* the gray level distribution at the end. Steps can be summarized in:

1. **ROI selection and cropping**

   Mammographic images are characterized by a gray level bimodal distribution (see histogram in *Figure 3.4*) in which one peak refers to biological tissues and the

other one to the background. To segment breast tissues, we used the Otsu threshold [83]. Then, binarized images smallest rectangular area surrounding the biggest over-threshold connected points area, i.e. the breast.

2. ROI normalization and background isolation

Over-threshold pixels belonging to the breast region are normalized in order to obtain a zero-mean distribution with variance equal to 1 as follows:

$$p(x,y) = {(p(x,y) - mean(ROI))}\Big/{std(ROI)} \quad \textit{Equation 3.1}$$

When $p(x,y)$ is the pixel at x row and y column, ROI refers to the pixels belonging to the breast. Background pixels are all set equal to a fixed value empirically set to -20. No further histogram equalization was performed.

3. Rigid image resizing

A rigid rescaling is performed in order to match the longest side of the image with the longest side of the standard and to include all the breast area on it. Then if the resized image doesn't have the same size of the input layer it was padded with background pixels to fix size mismatch. During this process, breast resized image position was fixed to the top right or left corner of the new image depending on breast side, while background pixels were added according to initial image orientation. Centroid position was used to detect breast side. *Figure 3.5* shows an example of histogram after preprocessing.

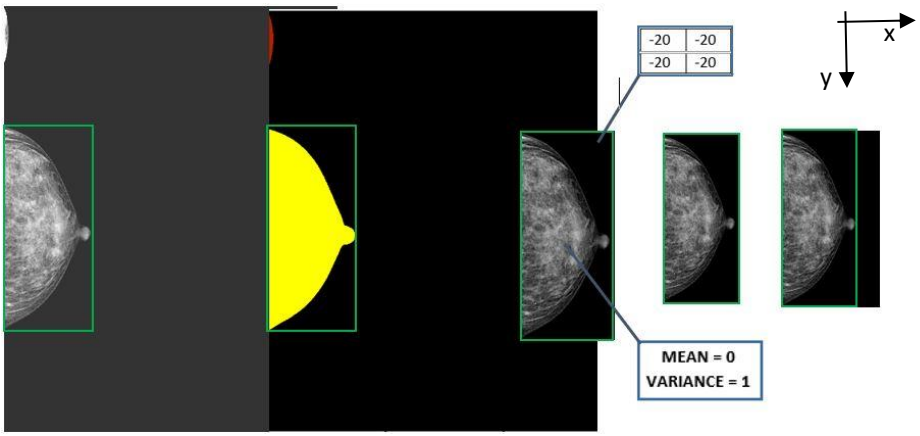


Figure 3.3 *Steps involved in data preparation applied to a CC view mammogram. Starting from a 3580x2784 pixels image (first one) in which the breast can be included in an area of 1732x753 pixels (green rectangle), we obtain a 1536x768 pixels image (last one) in which the gray-scale pixels values of the breast have zero mean and variance = 1, the background is isolated (putting all pixels values = -20) and occupy the smaller area possible. It should be noted that the initial image, in addition to the breast, contained a small portion of shoulder (second image, in red) which was removed.*

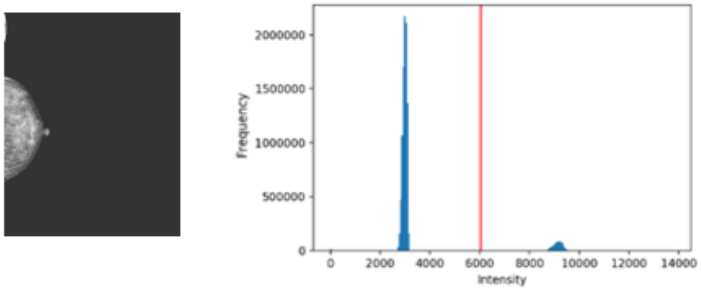


Figure 3.4 Gray-scale pixels intensities histogram before preprocessing. Image (left) and histogram (right). Red line corresponds to Otsu's threshold.

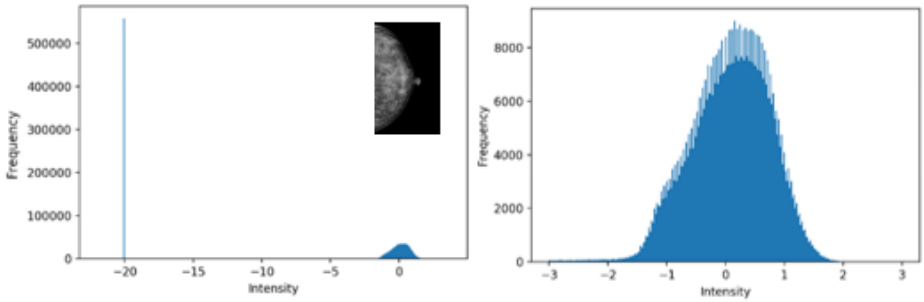


Figure 3.5 *Gray-scale pixels intensities histogram after preprocessing. Whole image histogram (left), breast pixels histogram (right).*

Data augmentation

Data augmentation is a dataspace solution to the problem of limited data which in turn causes the overfitting problem [84]. Several studies have shown that this technique can improve the performance of deep learning approaches by enlarging the number of training samples using different kinds of image transformations, ; see for instance [85]. In addition, it was proved that adding noise to images can help CNNs to learn more robust features [86]. Our BACs recognition model must overcome issues related to variability in scale, position, noises in the image, and more. The aim of data augmentation is to bake these translational invariances into dataset such that the resulting model will perform well despite these variability within data. We applied data augmentation online, i.e. during training. Each image, before being processed by the network, underwent several transformations. Included transformations were geometric transformations, noise addition and filtering.

Geometric transformations:

- **Vertical and horizontal flip**: both applied independently with a probability of 50%. *Figure 3.6* shows the effects of different flipping in BACs appearance.
- **Zoom**: randomly selected in a uniform distribution [-30 %, 5 %]
- **Width shift**: randomly selected in a uniform distribution $[-0.001n_c, 0.001n_c]$ pixels, where n_c represents the number of columns in the image
- **Height shift**: randomly selected in a uniform distribution $[-0.001n_r, 0.001n_r]$ pixels, where n_r represents the number of columns in the image
- **Rotation**: randomly selected in a uniform distribution [-3, 3] degrees with step equal to 10^{-16}.

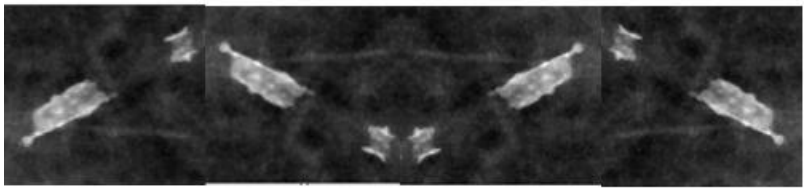


Figure 3.6 *Effect of vertical and horizontal flip in BACs appearance. [a] Original no flipped patch including BACs [b] Vertical flip [c] Horizontal flip [d] Vertical + Horizontal flip.*

Types of noise:

- **Gaussian noise** whose probability density function is defined by a mean value randomly selected in the range [0, 0.5] with step equal to 10^{-16} and standard deviation in range [0.01, 0.4].
- **Salt and pepper noise** covering from the 0.01 to 1 % of breast pixels, selected randomly. Salt pixels have intensity randomly selected in the range $[I_{max}, 1.2I_{max}]$ with I_{max} representing maximum image intensity while pepper pixels intensities belongs to the range $[1.2I_{min}, I_{min}]$ with I_{min} *($I_{min} < 0$* due to normalization*) representing the minimum image intensity value*. The ratio between bright and dark noisy pixels ranged from 0 to 100 %.

Type of filtering:

- **Gaussian filtering:** application of a gaussian filter with a randomly selected kernel size in range [3, 7] pixels
- **Average filtering** with a randomly selected kernel size in range [3, 7] pixels

Geometric transformations were always applied, while the addition of all type of noise and filtering take place with a probability of 12.5% each and the application of one exclude the others. In order keep the learning process simple, no noise and filtering are applied in 50% of cases. Noises and filtering were applied only to the breast pixels and not to the background. After noise and filter application breast pixel intensities were normalized in order to maintain a 0 mean and unitary variance distribution. Geometric transformations parameters have been chosen to preserve the labels associated with the images by not cutting out ROIs possibly containing BACs.

3.4 CNN architecture building up

We implemented our binary mammograms classifier for BACs detection using a network-based deep transfer learning strategy.

We selected one of the most famous deep pretrained neural networks, namely the VGG16 architecture which was trained, validated and tested on the very large natural images dataset ImageNet [60]. We used its convolutional base as features extractor, while modifying the dense layers part. We stacked VGG16 convolutional base with a new designed classifier composed by three fully connected layers. To allow the convolutional base able to learn high-level features related to the target specific task, while exploiting the ability of the pretrained VGG16 architecture to extract low-level features. To this aim, we froze low-level filters weights and fine-tuned the weights of last convolutional base layers.

A limit of the actual stage of our work, is the lack of the test process, surrogating the performance of the CNN by the validation figures of merit. This limit emerged ex-post due to the needed training-set size experimentally verified, which gave further space to validation-sets only. Proper testing is scheduled in the near future as soon as further data are collected and annotated.

The number of hidden units of fully connected layers and number of fine-tuned convolutional base layers are closely related network hyperparameters since they determine the model complexity and its ability to fits to the target domain. For this reason, as a first fine tuning step we fixed the number of hidden units to investigate how many convolutional layers must be trained to learn useful high-level features. Then, once we narrowed down the range of possible solutions, we empirically assessed the cross effect of those two hyperparameters on network performances and chose the best hyperparameter set.

3.4.1 Features extractor

VGG16 is a 16 layers CNN proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University. This network won the first and second place in ImageNet Large Scale Visual Recognition Competition 2014 in object detection and image classification categories [60]. Net architecture is shown in *Figure 3.7.*

VGG16 architecture was designed to process RGB images and is characterized by an input layer with shape (224,224,3) pixels. Its convolutional base is composed of 13 convolutional layers using filters with kernel size of 3 × 3 pixels, which is the smallest size to capture the notion of left/right, up/down, center, convolutional stride (i.e., shift of the 3x3 weight matrix) was fixed to 1 pixel, while padding was chosen equal to 1 so that spatial resolution is preserved after convolution. Convolutional layers are grouped into two convolutional blocks composed by two layers and three blocks composed by three layers. Each block is then followed by a max-pooling layer that perform spatial pooling over a 2 × 2 pixels window, thus converging to a single recognition choice, among the hundreds of objects represented in the dataset. The stack of convolutional layers is followed by a fully connected neural network composed by two hidden layers of 4096 units with ReLU activation function followed by a SoftMax output layer of 1000 units, which reflects number of classes present of ImageNet database.

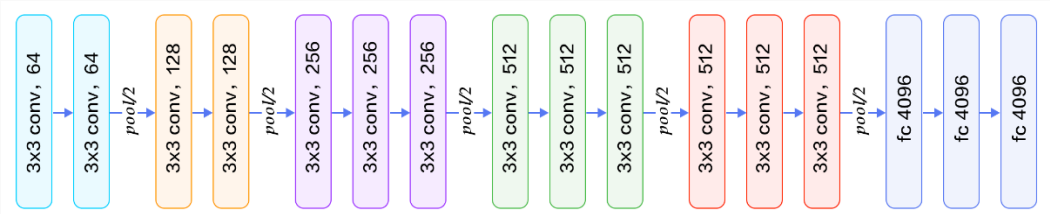


Figure 3.7 *VGG16 architecture. Originally designed for the ImageNet database, the image input size is 224 × 224 and after each Max Pooling layer feature maps dimension is halved. The feature extractor part of the net has as output a tensor of size 7 × 7 x 512, i.e. 512 feature maps of 7 × 7 pixels.*

There are several cases in the literature where VGG16 has been used for transfer learning in biomedical imaging applications. For example, Gardezi *et al.*[87] used this approach to build a classifier of normal and abnormal tissues on mammograms ROI obtaining 100% classification accuracy. Shallu *et al.* [88] demonstrated the ability of transfer learning in comparison with the fully trained network on the histopathological imaging

modality by considering three pre-trained networks and yielded the best performance with a fine-tuned VGG16 model, with 92.60% accuracy.

We chose the VGG16 because is a very deep network with a high number of convolutional filters that could potentially model the complexity of the task at issue, i.e. cofounding factors in BAC identification and image quality variability among mammograms. Furthermore, we found similarities with the CNN-net for BACs detection published by Wang *et al*. [39] and introduced in *Section 1.4*. Wang's convolutional base network is indeed composed of two blocks of two convolutional layers and two blocks of three convolutional layers, using filters with kernel size of 3 × 3 pixels. Each block, like in VGG16, is followed by a max-pooling layer (*Figure 3.8*).

Layer	# filters	Size	Output size
input	-	-	95×95
Conv	32	3×3	95×95
Conv	16	3×3	95×95
Pooling	-	$3 \times 3 s2$	47×47
Conv	64	3×3	47×47
Conv	32	3×3	47×47
Pooling	-	$3 \times 3 s2$	23×23
Conv	128	3×3	23×23
Conv	128	3×3	23×23
Conv	128	3×3	23×23
Pooling	-	$3 \times 3 s2$	11×11
Conv	256	3×3	11×11
Conv	128	3×3	11×11
Conv	128	3×3	11×11
Pooling	-	$3 \times 3 s2$	5×5
FC	128	-	128
FC	2	-	2

Figure 3.8 *Deep CNN architecture used in Wang et al. BACs detection strategy. [from [39]]*

In our application, we transferred the first part of the network, the convolutional base with its layers and 3 × 3 filter kernel size and stride 1. The advantage of transfer those convolutional layers is that they can be applied to images of any size regardless the size of images used during training.

Due to the small size of BACs compared to the entire area covered by the breast, strong image downgrade was not possible. Furthermore, the breasts often cover a rectangular area

characterized by a height to width ratio equal to 2. So, to reduce background pixels as much as possible we set input images size to 1536 × 768 pixels. Number rows and columns were forced to be multiple of 32 in order to match the network architecture, which required to halve image resolution 5 times (number of max pooling layers). In addition, we converted the mammograms to RGB images by presenting to the input layer a stack of three identical grayscale images along the channel dimension.

Image size does not impact the structure of the network, but only changes the size of the feature maps obtained after each convolutional step. As a result, the output of the last convolutional and max pooling layer we had a 48 × 24 × 512 pixels tensor, which translates into an increase in parameters to be estimated in the fully connected part *(Figure 3.9)*.

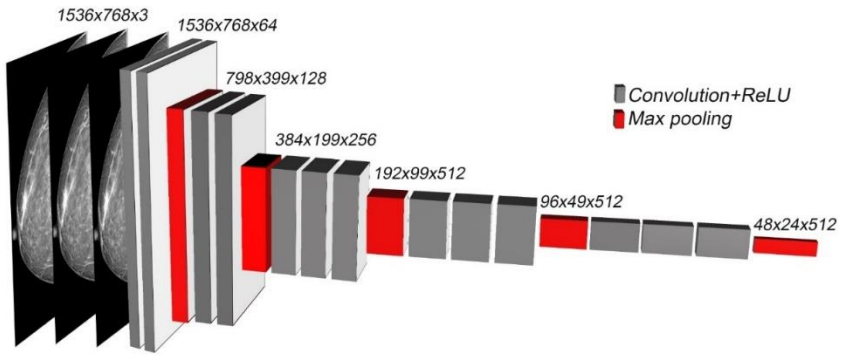


Figure 3.9 *Microarchitecture of our convolutional base transferred from VGG16.*

3.4.2 Fully connected layers

To design the fully connected part of the network, we mimicked the VGG16 original one, thus making only few changes necessary to fit our task and hardware capability. We designed the fully connected network as composed by two hidden layers with neurons characterized by leaky ReLU activation functions and one output neuron with a sigmoid activation function, according to our two-class classification task. Threshold for binary classify the output was fixed to 0.5. Each hidden layer was followed by a dropout layer to prevent overfitting [89]. Features can develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data. A dropout layer reduces this problem by randomly select a subset of hidden unit to

force neurons to learn independently from each other. A representation of the fully connected part is shown in *Figure 3.10.*

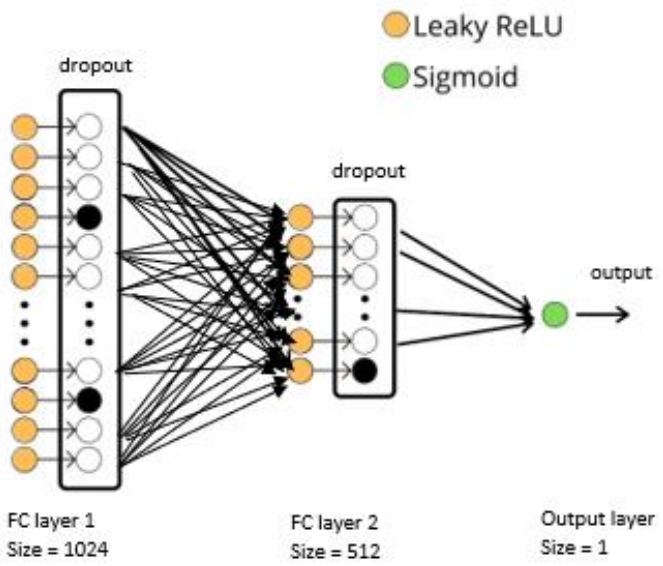


Figure 3.10 Schematic representation of our starting classifier fully connected (FC) part. There are two hidden layers having respectively 1024 and 512 units each, followed by a dropout layer. Black neurons are turned off because of dropout. Output layer consists on one neuron with sigmoid activation function.

As introduced before, we started our investigation with a fixed number of hidden units and after having fixed some parameters and narrowed the number of convolutional layers to train range, we tuned it. At first, we designed the dense network to reach the most complex network trainable without incurring in out of memory issues. The number of hidden units was set as power of two to exploit GPU efficiency and speed up the training phase. The final number of hidden units were respectively 1024 and 512 for the first and the second hidden layer, respectively. Both have a Leaky ReLU activation function with $\lambda = 0.3$ (*Section 2.1, Equation 2.4*). The dropout parameter was fixed to 0.3 (30% of neurons are turned off). Finally, we initialized weights using Glorot uniform distribution (also called Xavier uniform initializer) proposed by Glorot and Bengio [90]. Their method assumes that stabilizing weight gradients variances across layers can help the training process by avoiding the saturation and the excessive shrinkage of the gradient signal.

3.4.3 Training strategy

Ones we defined the convolutional base and fully connected architecture, we stacked them *(Figure 3.11)* and defined our training strategy.

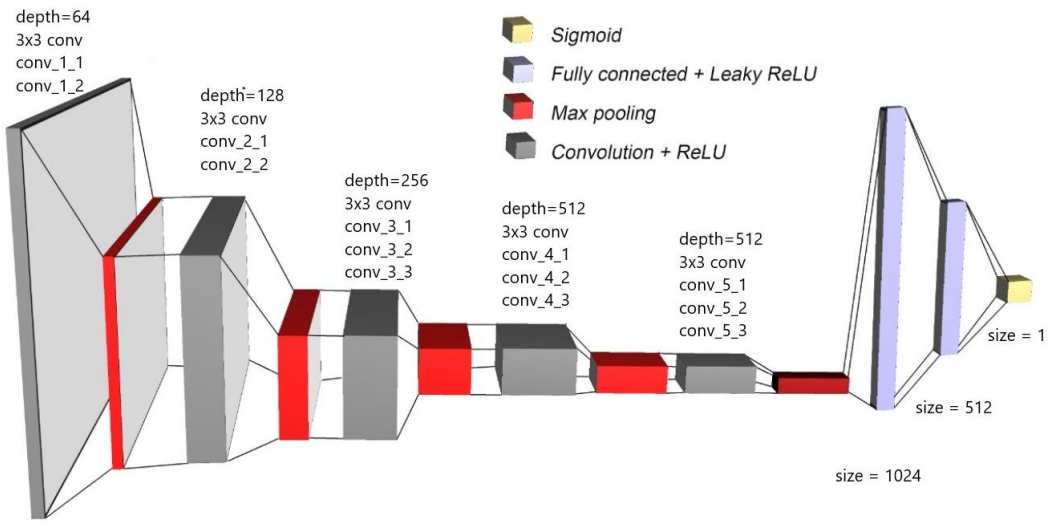


Figure 3.11 *CNN architecture obtained stacking VGG16 convolutional base and our designed fully connected part.*

Training strategy includes:

- Data split
- Class imbalance management
- Hyperparameter initialization
- Number of fine-tuned convolutional layer optimization
- Hyperparameter fine-tuning (learning rate, maximum number of epochs, number of fine-tuned convolutional layer, number of hidden units in dense layers)

Data split

Given that the database was enlarged in parallel with the development of the model, in this work we will refer to two different data splitting.

The first phase was performed on a subset of 168 patients (i.e., 684 images) part the whole dataset presented in section 3.2. $BACs^+$ prevalence per image was equal to 30%, with a significant overrepresentation of BACs compared to the epidemiological one, which is estimated to be 10.2%. This is justified for a better targeting of the training and the validation processes; conversely testing is foreseen on a dataset with the real prevalence. These data were splitted into training and validation sets (66:33 ratio) randomly sampling the entire database and checking for the resulting age matching. Hyperparameter initialization and number of fine-tuned convolutional layer optimization where performed on these data.

Next, passing to the whole dataset described in *Section 3.2* (717 patients, 2868 images) and a BACs prevalence per image of 10.2%, we performed the second splitting. After preliminary trials, we recognized that the predominant number of $BACs^-$ was severely biasing the training process, at least with our dataset, which has a significant, but not very large as needed in CNN training targeted to recognize fairly rare cases as $BACs^+$ ones. So, we decided to revert to the previously applied 30% prevalence by randomly under-sampling the $BACs^-$ class. To this aim, we randomly down sampled the majority class while keeping age distribution similar between the two BACs classes. For each patient in $BACs^+$ class, we randomly selected two patients among those 1 year younger or older than the selected positive subject. When no patients were present in this age range, the two closest patients $BACs^-$ were chosen. Finally, we randomly under-sampled negative subjects to reach the same $BACs^+$ prevalence of the dataset previously used. So, we obtained a database of 248 patients, 992 images. Then we performed three different subdivisions of the database (85:15 ratio), namely DB1, DB2, DB3. For each of them, the division between testing and validation set was made in the same way, thus approaching a Monte Carlo strategy, though limited to three extraction in place of many ones, for computational burden issues. We firstly shuffled the patients, then we randomly splitted the database preserving the same prevalence of bilateral and unilateral $BACs^+$ as well as $BACs^-$ patients in both training and validation sets.

Methodological note about class imbalance management

The note is presented here and not anticipated in Ch. "Methods", since the problem emerged ongoing.

The $BACs^{+}$ class is less prevalent than the $BACs^{-}$ one, as pointed out in the introduction section. Several solutions were previously proposed to reduce the class-imbalance problem. Those methods may alter both data distribution and algorithmic structure [91]. At the data level [92], solutions include many different forms of resampling such as random under-sampling the majority class, oversampling the minority class or combinations of them. At the algorithmic level [93], solutions include methods to adjust the loss function by assigning relatively higher costs to examples from minority classes. Since we hadn't enough data to under-sample the majority class without lose information and to avoid the possible overfitting caused by the oversampling performed adding repeated samples [94], when using the first data split, we opted for an algorithmic level approach. As anticipated, in the second data split, having added several samples, we combined an under-sampling of the majority class with the same algorithmic level approach.

We addressed class-imbalance using a cost-sensitive re-weighting method. The loss value that will be minimized by Adam optimizer at each batch will be the weighted sum of all individual losses, weighted by the inverse class frequency of the target class of the sample [95 96] .

Hyperparameter initialization

To start hyperparameter optimization, we *a-priori* initialized the number of epochs equal to 200. To speed up training time, we chose a batch size to be equal to 8 images, that correspond to the maximum batch size allowed by our hardware setup. Moreover, since we had to account a binary classification task, we used Binary Cross-Entropy loss-function. We used Adam optimizer with parameters setting equal to those reported by the authors of the paper [97]. We chose this optimization method because proved to be faster in convergence compared to other state of the art optimizers. Initially, the number of convolutional layers to train was put equal to three.

Learning rate is a very important hyperparameter because it determines the speed at which our model learns, but conversely it limits convergence stability. Unfortunately, it is not possible to estimate learning rate a priori [98] but there are evidences in literature that can be applied to the starting point choice, next proceeding by empirical adjustments. Bengio in his paper [99] reports that typical values of learning rate for a neural network with standardized inputs are less than 1 and greater than 10^{-6}.

Leslie Smith in 2017 [100] introduced a method to automatically find a range of feasible learning rates. Their method consists in running the model for several epochs starting from a very small learning rate and exponentially increasing it after each batch update. Next, plotting model accuracy versus learning rate, is possible to identify the minimum useful learning rate as the value where the accuracy starts to increase and the maximum useful learning rate as the value where the accuracy starts again to decrease. In our case, having an imbalanced dataset, accuracy as performance metric is not a good choice. Using the same reasoning, we defined the learning rate boundaries by plotting learning rate values versus class-weighted loss function values. An example is shown in *Figure 3.12.*

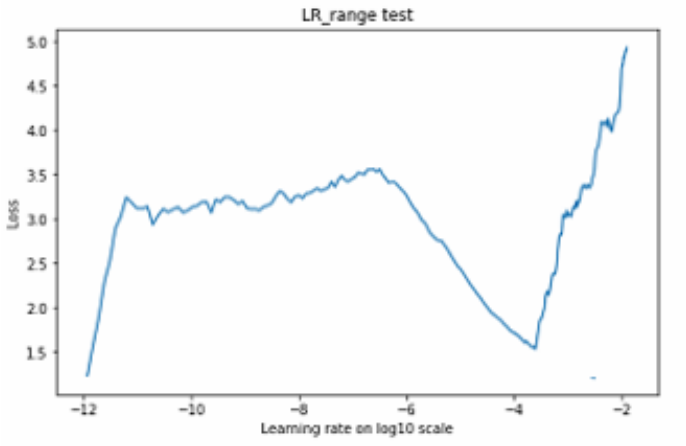


Figure 3.12 *Example of learning rate range estimation. Class-weighted loss function vs learning rate in Log10 scale is plotted. The minimum learning rate at which the networks already learns is identified by the point in which the loss starts to fast decrease(* 10^{-6} *). The maximum boundary is where loss starts to increase (* 10^{-4}*).*

Then we trained our model for a few epochs to empirically evaluate the best learning rate in the range identified. Finally, we chose to model learning rate decay over epochs as a half a cosine curve. It is called Cosine Annealing schedule and it is based on the idea to

start with relatively high learning rate for several iteration in the beginning to quickly approach a minimum, ending with several small learning rate iterations [101].

The learning rate lr_{ep} at each epoch ep, can be defined as following:

$$lr_{ep} = lr * \frac{\cos(\pi * {ep}/{EPH}) + 1}{2} \quad \textit{Equation 3.2}$$

lr is the starting learning rate and EPH is the maximum epoch (in which learning rate is zero).

Figure 3.13 shows an example of Cosine Annealing schedule with starting learning rate equal to $3 * 10^{-6}$ and maximum number of epochs EPH = 200.

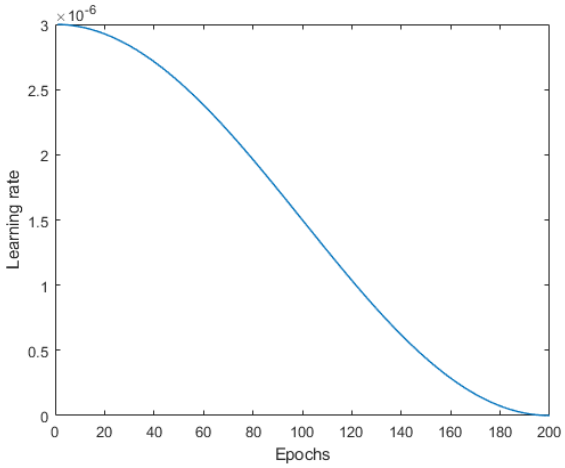


Figure 3.13 *Cosine Annealing schedule. Example of aggressive learning rate schedule where learning rate starts high and is dropped relatively rapidly to a minimum value near to zero*

Number of fine-tuned convolutional layer optimization

After having fixed other hyperparameters, we investigated the optimal number of convolutional layers to fine-tune. As proposed by Shin *et al.* [59] we ran a series of trainings starting to train the fully connected part only. Next, further convolutional layers were incrementally included until the desired performance level was reached. At this stage, we aimed at reducing overfitting by obtaining similar loss function decay in both the training and the validation sets. The addition of convolutional layers was stopped, passing to the fine-tuning of the final structure, as soon as the validation curve reached its minimum and started to grow, which is and established sign of overfitting.

Network hyperparameter fine-tuning

The monitoring of performance over the validation set aimed at the tuning of the following hyperparameters: i) the number of hidden units of the fully connected part; ii) the number of layers of the convolutional base; iii) the learning rate; and iv) the number of epochs (alias, iterations). Each parameter change made in this section was chosen by evaluating the effects on two different subdivisions of the database, namely DB1 and DB2. Lastly, when we found an eligible hyperparameter combination, we performed a training on a third dataset division, DB3.

Due to the increment in sample size, we needed to fine-tune both learning rate and number of epochs on the new dataset. For the learning rate, we used the same approach presented before. After having chosen the learning rate, we reduced the number of epochs by empirical observation of loss function evolution during trainings. Network structure, imbalanced-class compensation method, data augmentation, and other parameters were the same of the ones illustrated in the previous subsection.

Lastly, after having fixed all other hyperparameters, we investigated the best combination of number of hidden units of the fully connected layers and number of fine-tuned convolutional layers. We ran several training sections testing different combinations using the most widely used hyperparameter-tuning strategy based on a combination of a grid search and a manual search approach (e.g., [102] [103] [104]).

We decided to run each model two times, one per each database division DB1 and DB2. When a combination of parameters resulted to be not good enough with one division, in some cases we skipped the training with the other one. We ran each model for the same number of epochs, and we saved the model with its weights obtained at the epoch in which the validation loss function reached its minimum. This allowed us to save the best model before overfitting occurred.

3.5 Model evaluation

To evaluate performance and monitor the various tuning phases, we used some of the mostly used metrics in imbalanced dataset [66]. They are Precision, Recall, F1 score, ROC curve and ROC AUC (Area Under the ROC Curve). In order to give the same weight to FP and FN, in making decisions we mainly relied on the use of F1 score.

Since with small datasets the data splitting into training and validation dataset can affect the resulting training process and performances, we performed a k-fold cross validation [105] in order to evaluate the mean value of metrics. k-fold cross validation consists of randomly dividing all the observations into k groups and run k different training sessions. For each training session will be used all the observations, but each observation is used for validation in one training only. Finally, metrics are calculated for each K model and the mean (or median) of the results evaluated.

To asses if corrected predictions were done based on what we are looking for, i.e. BACs, we used a very easy but powerful and intuitive tool: saliency maps. Saliency maps are a reproduction of the input image in which at each pixel is assigned a level of intensity dependent from the importance that the related pixel had in computing prediction. To obtain them, we followed the Simonyan *et al.*[106] algorithm that computes the gradient of output category with respect to input image, thus providing a pixel-wise map of output sensitivity to change of local input values. If the network structure and hyperparameters are well designed and a $BACs^+$ image is correctly classified, the saliency map should highlight BACs pixels. To obtain a more intuitive output, we made our costume gradients color map in which the lower gradients of color map are transparent and then we overlapped it to the original input gray scale image.

4. Results

4.1 CNN architecture and final hyperparameters

4.1.1 Number of fine-tuned convolutional layer optimization

A summary of the initial database used to estimate the number of fine-tuned layers is showed on *Table 4.1* and *Table 4.2*. In this dataset, the $BACs^{+}$ prevalence per image was equal to 30%.

	BAC+ total	**BAC+ unilateral**	**BAC+ bilateral**	**BAC-**	**Total patients**
Frequency	51	20	31	117	168
Prevalence [%]	30.35	11.90	18.45	69.64	100.00%

Table 4.1 *Labels per patient. Number of patients with or without breast arterial calcifications in our dataset.*

	BAC+	**BAC-**	**Total images**
Frequency	108	576	684
Prevalence [%]	30	70	100.00%

Table 4.2 *Labels per image. Number of images labelled as presenting BACs or not in our dataset.*

Loss function vs learning rate plot resulting from learning rate test performed on these data is shown in *Figure 4.1*. The learning rate that best performed in this subset of images and that we used during training aimed to optimize number of convolutional layers to train resulted to be equal to $3 * 10^{-6}$.

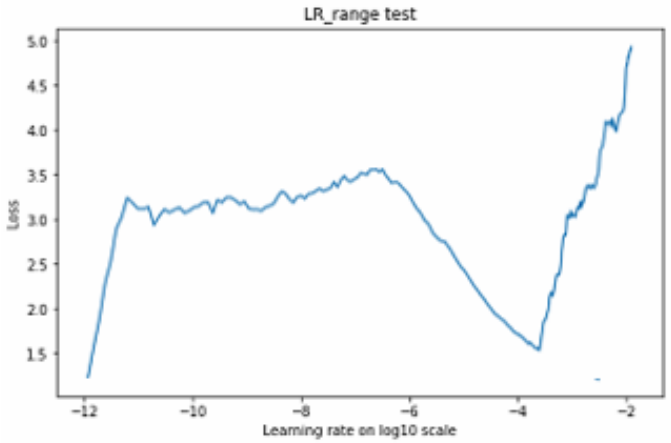


Figure 4.1 Learning rate test. Class-weighted loss function vs learning rate. At each batch update the learning rate is exponentially increased and corresponding loss function is reported. Good learning rates should be values from 10^{-6} *and* 10^{-4}*, where the cost function decreases.*

Results of our strategy to find the optimal number of convolutional layers to train are showed in *Figure 4.2.* We trained till the sixth convolutional layer before stopping, because overfitting occurred. It is visible a gradual improvement reaching the best performance training five convolutional layers because training and validation losses are very close to each other. After that, deterioration begins.

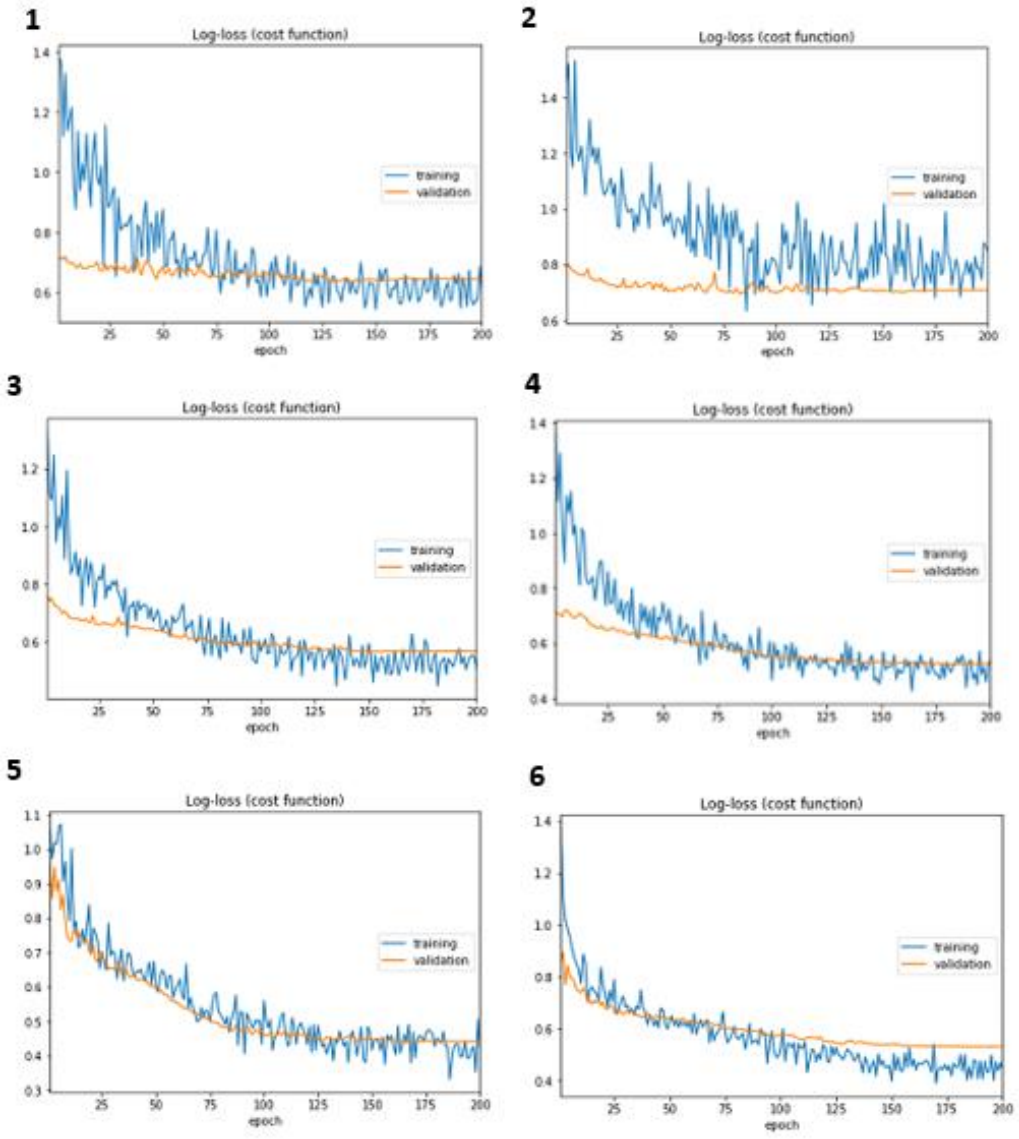


Figure 4.2 *Training and validation log-loss function (Binary Cross-Entropy) improving during training, while adding more convolutional layers to fine-tune. In order: 1, 2, 3, 4, 5, 6 convolutional layers tuned. We can see an initial small improvement from 3, then best performance is reached in model 5. Finally, a little overfitting starts to come from epochs 60 in model 6.*

The same evolution can be ascertained in the observation of the saliency maps, which allowed us to visualize if the CNN was really using BACs-related pixels to make prediction. Two examples of saliency maps overlapped on their mammograms are reporter respectively in *Figure 4.3* and *Figure 4.4*. In *Figure 4.3* we used model 1 (1 convolutional layer fine-tuned), in *Figure 4.4*, model 5, which according to the loss function plots are the worst and the best model. The saliency map is overlapped to its mammogram to highlight pixels that most influence the prediction. In *Figure 4.3* the worst performing 1-layer convolutional neural network shows the colored heat map pixels covering wide breast regions, missing the BACs pixels.

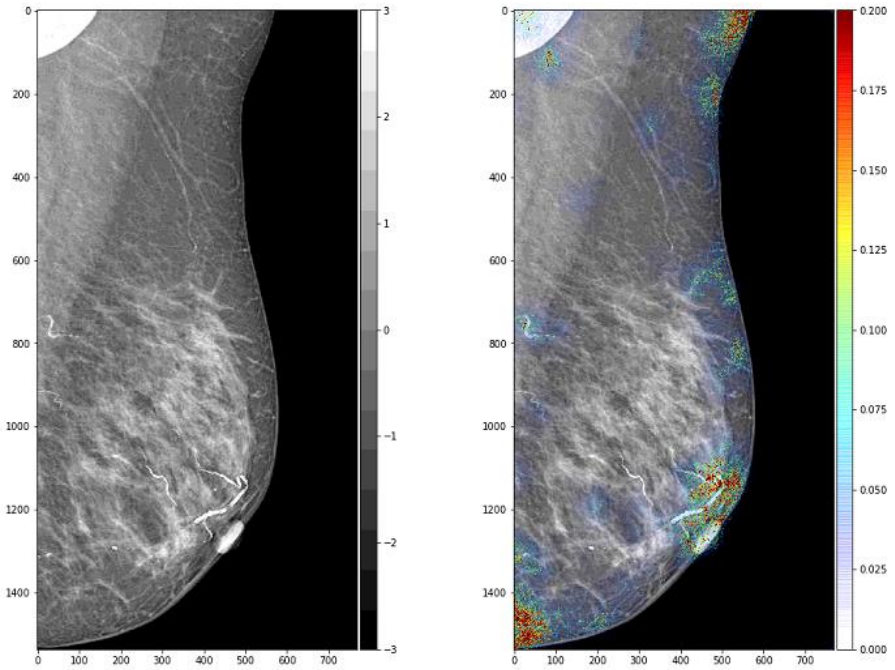


Figure 4.3 *Saliency map of the worst model. Worst model (1 convolutional layer fine-tuned) saliency map covers the entire breast.*

Passing to the best performing CNN, even if limited to only five convolutional layers, red pixels are shown in *Figure 4.4* well concentrated over the BACs ROI. It is worth noting, that both the hot ROI background and the BACs pixels are marked, showing that our dichotomic training did capture features relevant to object/surround contrast and differences (most probably, relevant to intensity and texture).

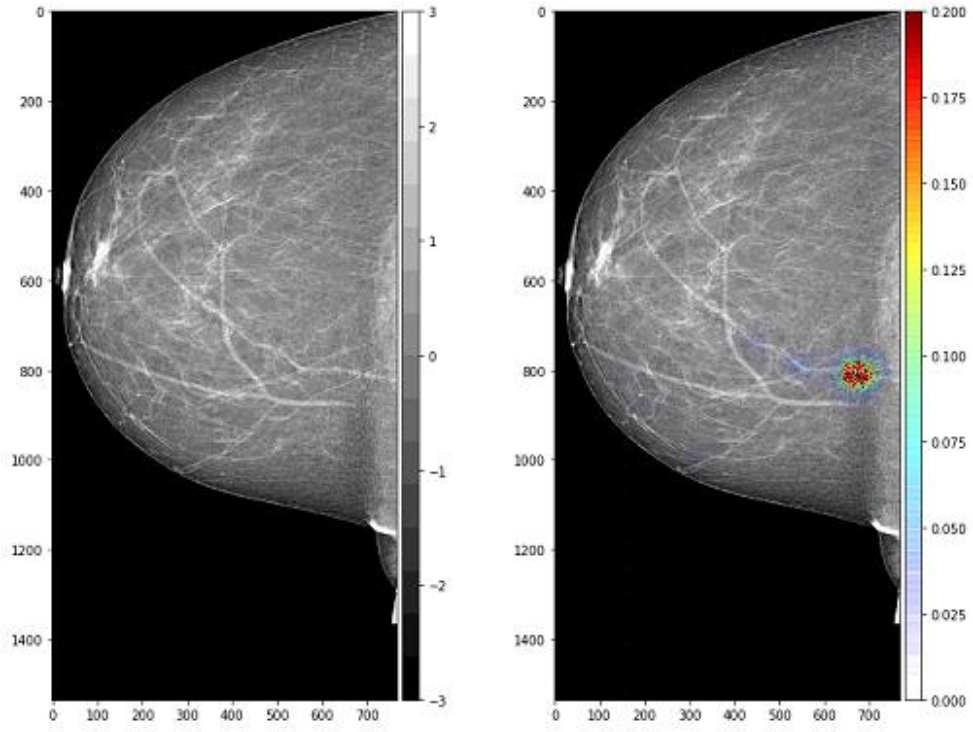


Figure 4.4 *Saliency map of the best model. Best model (5 convolutional layers fine-tuned) saliency map is overlapped to its mammogram and is concentrated over the BACs ROI.*

4.1.2 Network hyperparameters fine-tuning

A summary of the larger database, though under-sampling $BACs^-$ to elevate $BACs^+$ prevalence close to 30% (saving the age matching) is reported in *Table 4.3* and *Table 4.4*. $BACs^+$ prevalence per image is equal to 29.54%. The distribution of patient' age per BACs classes is reported in *Figure 4.5*. $BACs^+$ and $BACs^-$ age was 66.7 ± 9.3 and 66.5 ± 8.7, respectively.

	BACs+ total	**BACs+ unilateral**	**BACs+ bilateral**	**BACs-**	**Total patients**
Frequency	95	41	54	153	248
Prevalence [%]	38.31	16.53	21.77	61.69	100.00%

Table 4.3 *Labels per patient after resampling. Number of patients with or without breast arterial calcifications in our resampled dataset used to fine-tune hyperparameters.*

	BACs+	BACs-	Total images
Frequency	293	699	992
Prevalence [%]	29.54	70.46	100.00%

Table 4.4 *Labels per image after resampling. Number of images labelled as presenting BACs or not in our resampled dataset used to fine-tune hyperparameters.*

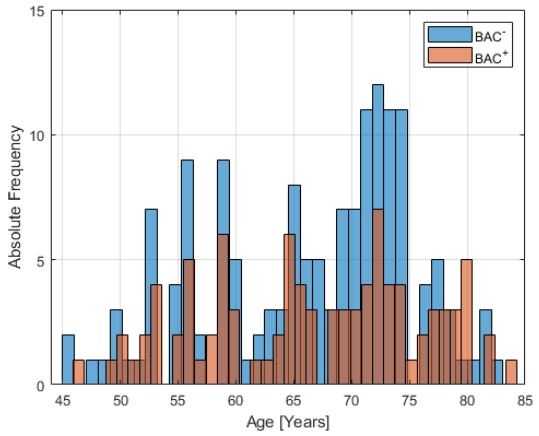


Figure 4.5 Distribution of patients' age per BACs classes in under-sampled database. Histograms of patient distribution according to age of women with and without breast arterial calcifications after under-sampling of the majority class.

Results relevant to the optimization of the learning rate for this database are shown in *Figure 4.6.* The best starting learning rate for our learning rate Cosine Annealing schedule was found to be equal to 10^{-5}.

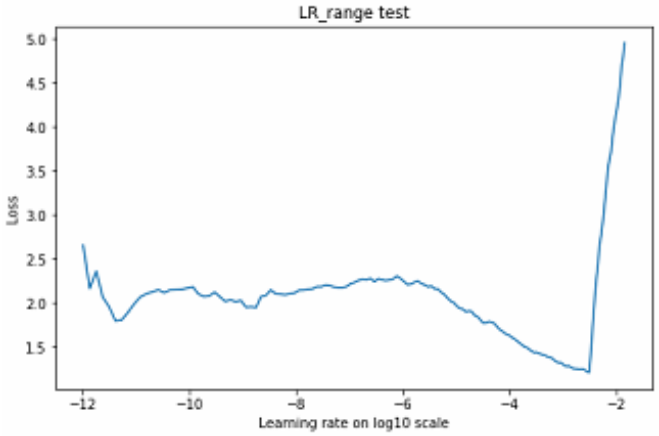


Figure 4.6 *Learning rate test result. Class-weighted loss function vs learning rate. At each batch update the learning rate is exponentially increased and corresponding loss function is reported. Good learning rates should be values around* 10^{-6} *and* 10^{-3}*, where the cost function decreases.*

Since *Figure 4.6* indicates that no learning can happen below a learning rate of 10^{-6}, we the lower boundary of Cosine Annealing to this level. After several trial and error tests, we came to the cosine arch lower value and epoch range reported in *Figure 4.7*, since this permitted the best compromise between learning speed and convergence stability. It corresponds to a Cosine Annealing schedule of 100 epochs, truncated at the 50th epoch (see *Equation 3.2*, *Section 3.4.3*).

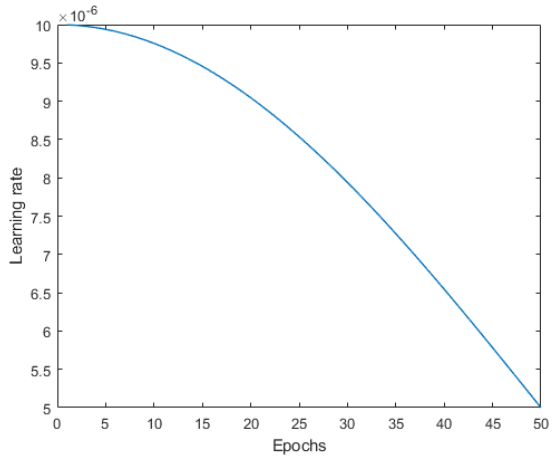


Figure 4.7 *Learning rate scheduling after fine-tuning. It corresponds to a truncated Cosine Annealing schedule, in order to avoid having learning rate too low at which our model would not learn.*

Specifications of each model tested are shown in *Table 4.5*. And results are reported in *Table 4.6*

Model	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Number of hidden units (1st layer)	1024	512	512	256	256	256	128	128
Number of hidden units (2nd layer)	512	512	512	256	256	256	128	128
Number of trained convolutional layers	5	4	5	4	5	6	5	6

Table 4.5 *Specifications of models trained in manual grid-search. Hidden units (1) and (2) refer respectively to the number of neurons of the first and second fully connected layers. Conv. Layers is the number. In green the best configuration.*

.

Model		Precision	Recall	F1 score	ROC AUC	Epoch
1	**DB1**	0.935	0.674	0.793	0.87	18
	DB2	/	/	/	/	/
2	**DB1**	0.833	0.814	0.823	0.93	27
	DB2	0.794	0.704	0.747	0.85	36
3	**DB1**	0.769	0.465	0.579	0.84	18
	DB2	/	/	/	/	/
4	**DB1**	/	/	/	/	/
	DB2	0.878	0.659	0.753	0.90	38
5	**DB1**	0.800	0.837	0.818	0.92	17
	DB2	0.837	0.818	0.827	0.94	45
6	**DB1**	0.837	0.720	0.774	0.87	24
	DB2	0.800	0.727	0.761	0.9	30
7	**DB1**	0.800	0.744	0.771	0.90	18
	DB2	0.871	0.772	0.819	0.94	30
8	**DB1**	0.833	0.68	0.749	0.87	11
	DB2	0.918	0.772	0.839	90.93	17

Table 4.6 *Metrics evaluated for each parameter combinations model. Precision, Recall, F1 score, Area Under the ROC Curve (ROC AUC) calculated on the validation set, for each model is shown. Epoch refers to the epoch in which the validation loss function value was minimum, at which point we saved the model weights as the optimal ones. Model 5 is highlighted in green, since it displayed the best performances in both DB1 and DB2 validation datasets.*

The best combination resulted to be the CNN with 2 fully connected layers having 256 neurons plus 5 further convolutional layers to be fine-tuned on the target classification (Model 5, highlighted in green in *Table 4.6*). We applied this configuration to a third subset partition DB3. In *Figure 4.8* we reported the evolution of the cost function, precision, recall, and F1 score for each epoch during training. We trained the model for 50 epochs, but it is visible that the minimum was reached around the 30th epoch, after which the validation loss function started to increase (sign of overfitting). Since we saved the model at the validation loss minimum, this is not a problem. A maximum of 50 computed epochs gave a wide margin in the unlikely case of a slower convergence. Confusion matrix and metrics computed on validation data relevant to the best model saved at epoch 29 are reported respectively in *Table 4.7* and *Table 4.8*. The ROC curve is shown in *Figure 4.9.*

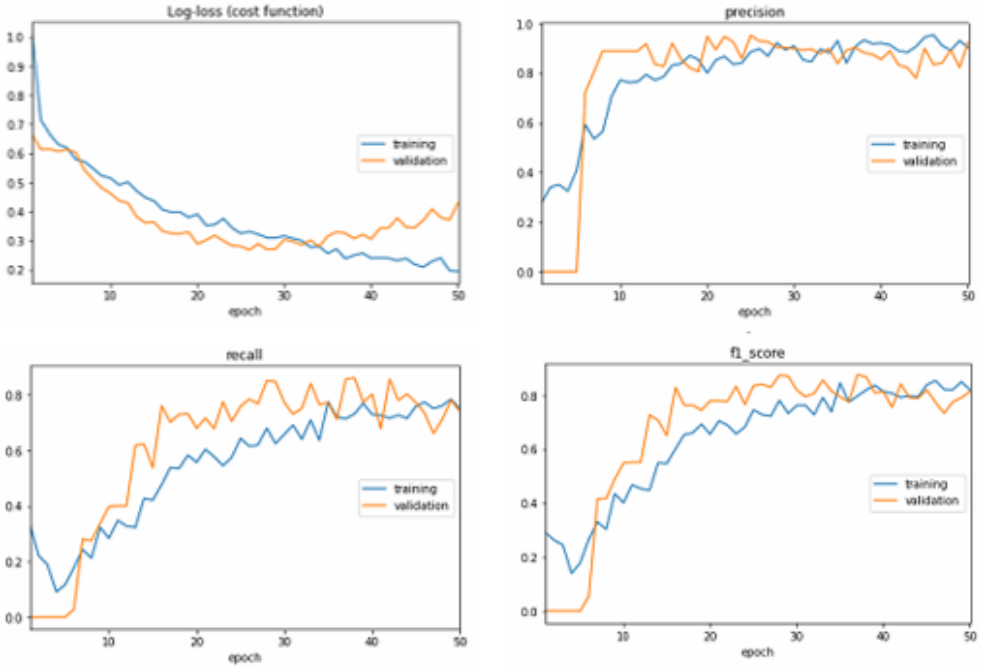


Figure 4.8 Evolution of loss function and metrics during training using a third division (DB3). Cost function, precision, recall and F1 score are reported for each epoch.

		Prediciton	
		Positive	Negative
Real	Positive	**36**	**8**
	Negative	**4**	**96**

Table 4.8 *Confusion matrix of validation data predictions.*

Precision	**0.900**
Recall	0.818
F1 score	0.857
ROC AUC	0.950

Table 4.7 *Metrics calculated on validation data.*

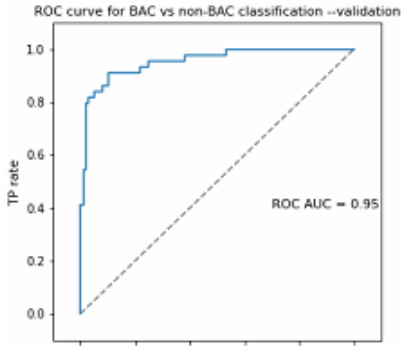


Figure 4.9 *ROC curve, best model, third database division (DB3), validation data.*

4.1.3 Final network architecture

Final network architecture is reported in *Table 4.9* We obtained a model having 14912065 parameters, of which 13.176.577 trainable and 1.735.488 non-trainable.

Layer	# filters	Size	Output shape	#parameters	Trainable
Input	-	-	1536 × 768 × 3	-	-
Conv_1_1	64	3 × 3	1536 × 768 × 64	1792	NO
Conv_1_2	64	3 × 3	1536 × 768 × 64	36928	NO
Max Pooling 1	-	2 × 2	768 × 399 × 64	0	-
Conv_2_1	128	3 × 3	768 × 399 × 128	73856	NO
Conv_2_2	128	3 × 3	768 × 399 × 128	147584	NO
Max Pooling 2	-	2 × 2	384 × 199 × 128	0	-
Conv_3_1	256	3 × 3	384 × 199 × 256	295168	NO
Conv_3_2	256	3 × 3	384 × 199 × 256	590080	NO
Conv_3_3	256	3 × 3	384 × 199 × 256	590080	NO
Max Pooling 3	-	2 × 2	192 × 99 × 256	0	-
Conv_4_1	512	3 × 3	192 × 99 × 512	1180160	NO
Conv_4_2	512	3 × 3	192 × 99 × 512	2359808	YES
Conv_4_3	512	3 × 3	192 × 99 × 512	2359808	YES
Max Pooling 4	-	2 × 2	96 × 49 × 512	0	-
Conv_5_1	512	3 × 3	96 × 49 × 512	2359808	YES
Conv_5_2	512	3 × 3	96 × 49 × 512	2359808	YES
Conv_5_3	512	3 × 3	96 × 49 × 512	2359808	YES
Max Pooling 5	-	2 × 2	48 × 24 × 512	0	-
FC1	256	-	256	131328	YES
FC2	256	-	256	65792	YES
Output	1	-	1	257	YES

Table 4.9 *Final CNN architecture.*

4.2 Model validation

A 7-fold cross-validation was performed. Resulting metrics for each model calculated in training data are fully reported in *Table 4.10*, in which Epoch refers to the epoch in which the loss function reached its minimum and the resulting model weights were saved. An average result is reported in *Table 4.11*. The same information obtained from validation data are shown in *Table 4.12* and *Table 4.13*. ROC curves for validation data of each model are reported in *Figure 4.10.*

K	Precision	Recall	F1 score	ROC AUC	Epoch
1	0.940	0.706	0.806	0.93	15
2	0.921	0.779	0.844	0.95	13
3	0.930	0.708	0.804	0.93	11
4	0.912	0.881	0.897	0.96	32
5	0.837	0.864	0.853	0.96	21
6	0.952	0.757	0.843	0.95	18
7	0.975	0.630	0.765	0.93	12

Table 4.10 *7-fold cross-validation results for training set.*

	Precision	Recall	F1 score	ROC AUC
Minimum	0.837	0.630	0.765	0.93
Mean ± SD	0.923 ± 0.043	0.760 ±0.089	0.830 ±0.042	0.94 ±0.01
Maximum	0.975	0.864	0.897	0.96

Table 4.11 *7-fold cross validation compressive result on training set.*

K	Precision	Recall	F1 score	ROC AUC	Epoch
1	0.868	0.673	0.758	0.86	15
2	0.854	0.759	0.803	0.9	13
3	0.866	0.433	0.577	0.72	11
4	0.950	0.760	0.840	0.94	32
5	0.829	0.772	0.799	0.91	21
6	0.842	0.533	0.653	0.86	18
7	0.843	0.729	0.782	0.89	12

Table 4.12 *7-fold cross-validation results for validation set.*

	Precision	Recall	F1 score	ROC AUC
Minimum	0.842	0.433	0.653	0.72
Mean ± SD	0.864 ± 0.040	0.667 ± 0.132	0.744 ± 0.094	0.86 ± 0.07
Maximum	0.950	0.772	0.840	0.94

Table 4.13 *7-fold cross-validation compressive results on validation set.*

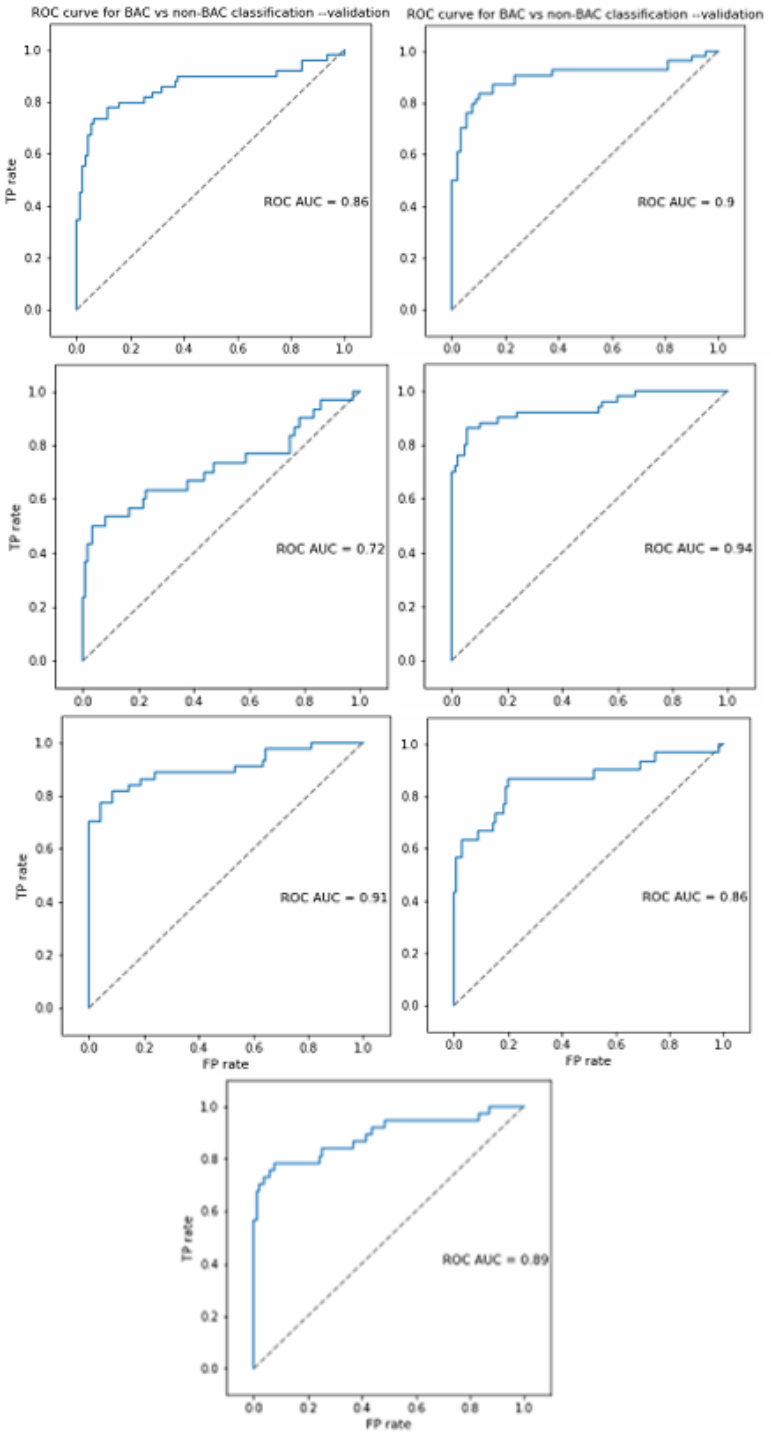


Figure 4.10 *ROC curves by validation data prediction for all 7-fold cross-validation models.*

4.3 Saliency maps

In the following, several representative examples of images and saliency maps are shown relevant to the validation dataset. *Figures 4.11–17* display true positive cases, where the presence of one or more BACs was correctly detected by the output score. Interestingly, scores were generally well above 0.5 threshold, close to 1. Moreover, the respective saliency map well localizes the single BACs ROI or, the case of multiple BACs, at least the ROI of the predominant one. Cases are shown of successful recognition with various combinations of BACs and breast features.

Several of the few cases of false negative and the relevant missed BACs are shown in *Figures 4.18-20*. In some cases, BACs were correctly detected by the saliency, but the output value of the network was slightly below the sigmoid classification threshold and the image was therefore classified as negative (*Figure 4.18*). In other false negative cases, the saliency highlighted different breast regions including BACs and not-BACs (*Figure 4.19*) or only non-BACs regions, ignoring BACs. In the first two cases scores were slightly below the 0.5 threshold, while in the last case, the output is close to 0.

Figure 4.21 shows an example of false positive in which saliency map focused on a tubular structure which is not a BACs and the score is just above the threshold.

Finally, in *Figure 4.22* two of many cases of true negative are shown. The scores are well below the classification threshold and the saliency maps don't highlight a particular region, but several regions scattered through the breast.

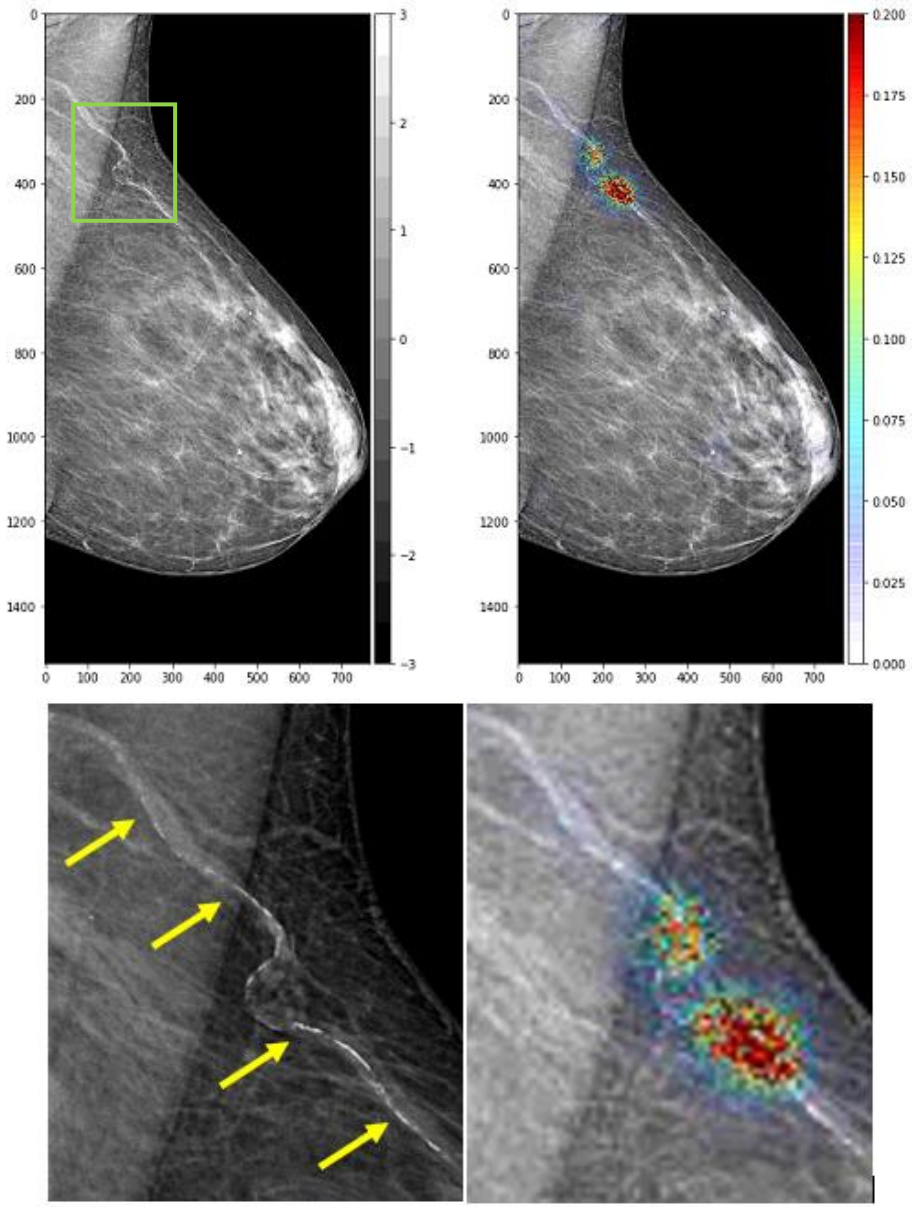


Figure 4.11 *Example of true positive image having severe BACs and its saliency map. (Up, left) Input $BACs^{+}$ mammogram belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red/green the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down, left) Zoomed BACs region. Calcifications are indicated by yellow arrows. (Down, right) Zoomed highlighted BACs regions. The output of the network for this image is equal to 0.9493 (classification threshold equal to 0.5).*

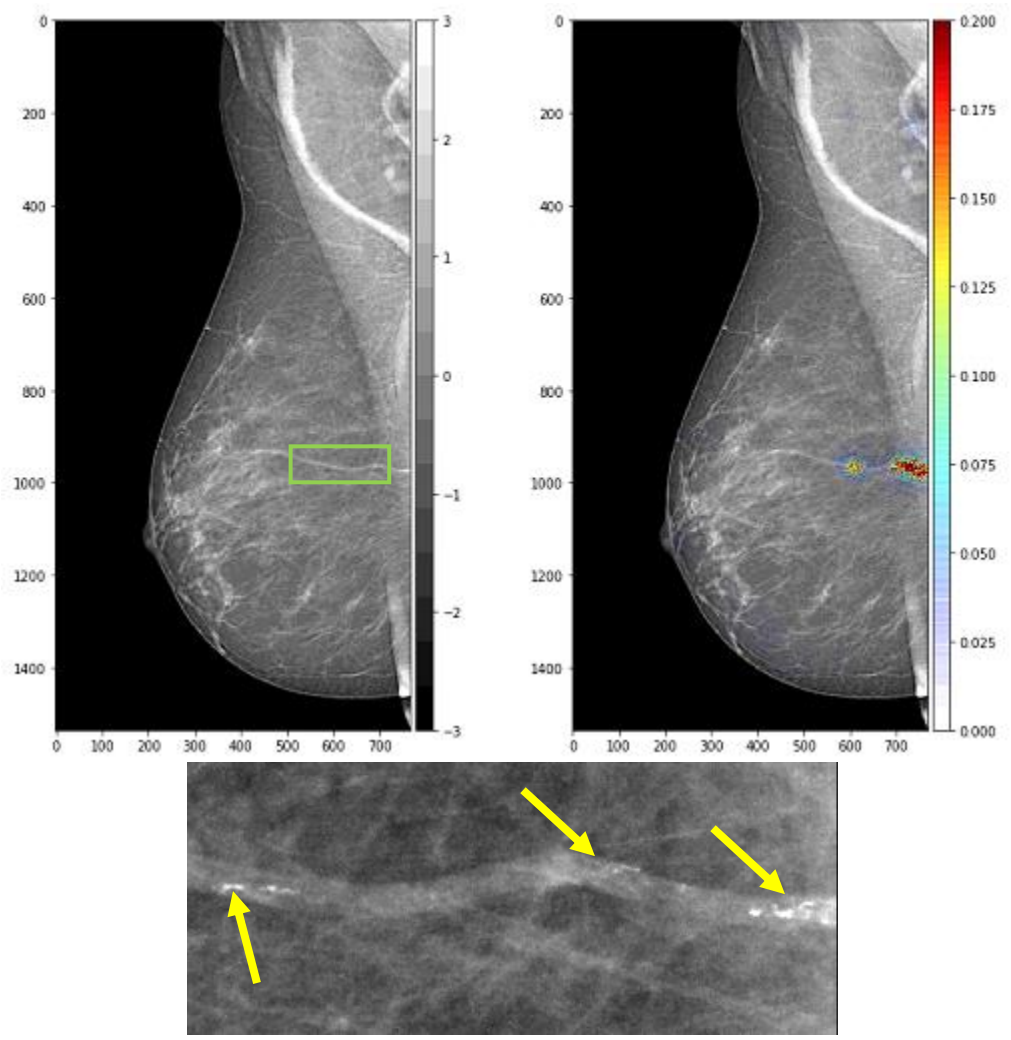


Figure 4.12 *Example of true positive image having sparse BACs and its saliency map. (Up, left) Input BACs⁺ mammogram belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down) Zoomed BACs region. BACs are the three white clusters indicated by yellow arrows. The output of the network for this image is equal to 0.9493 (classification threshold equal to 0.5).*

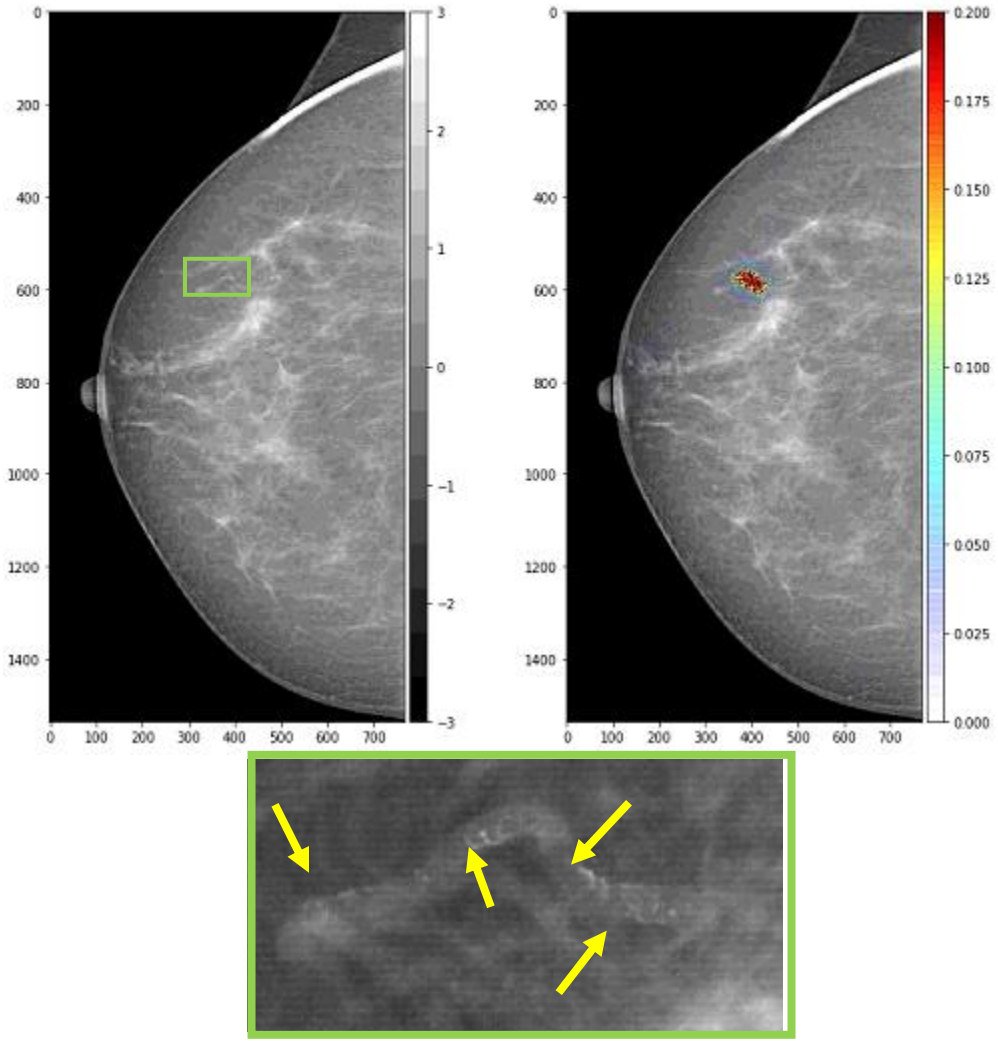


Figure 4.13 *Example of true positive having small BACs and its saliency map. (Up, left) Input $BACs^{+}$ mammogram belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down) Zoomed BACs region. BACs are white pixels indicated by yellow arrows. The output of the network for this image is equal to 0.9964 (classification threshold equal to 0.5).*

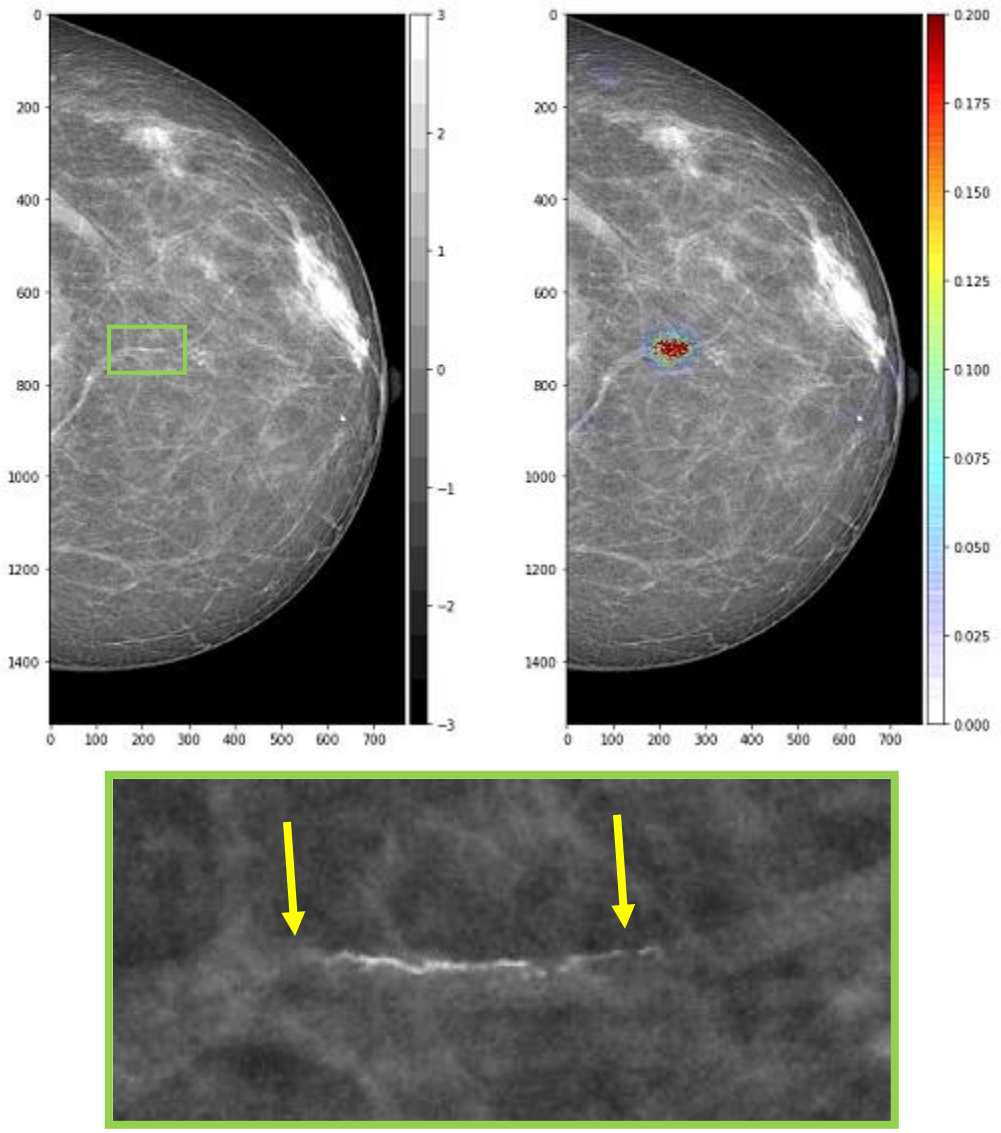


Figure 4.14 *Example of true positive image having linear one side BACs and its saliency map. (Up, left) Input BACs$^+$ image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down) Zoomed BACs region. BACs are white pixels between the two yellow arrows. The output of the network for this image is equal to 0.9987 (classification threshold equal to 0.5).*

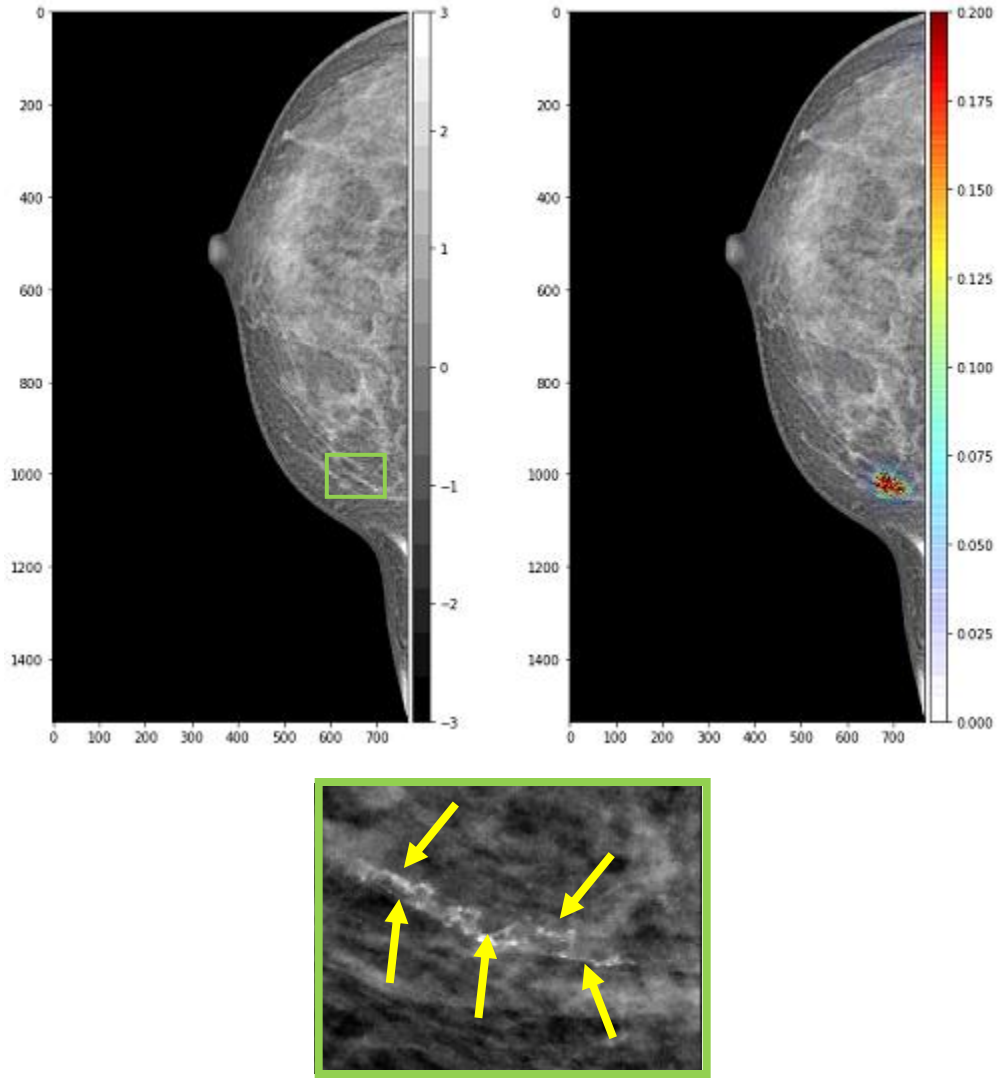


Figure 4.15 *Example of true positive image with dense breast and its saliency map. (Up, left) Input BACs⁺ mammogram belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down) Zoomed BACs region. BACs are white pixels indicated by yellow arrows. The output of the network for this image is equal to 0.9968 (classification threshold equal to 0.5).*

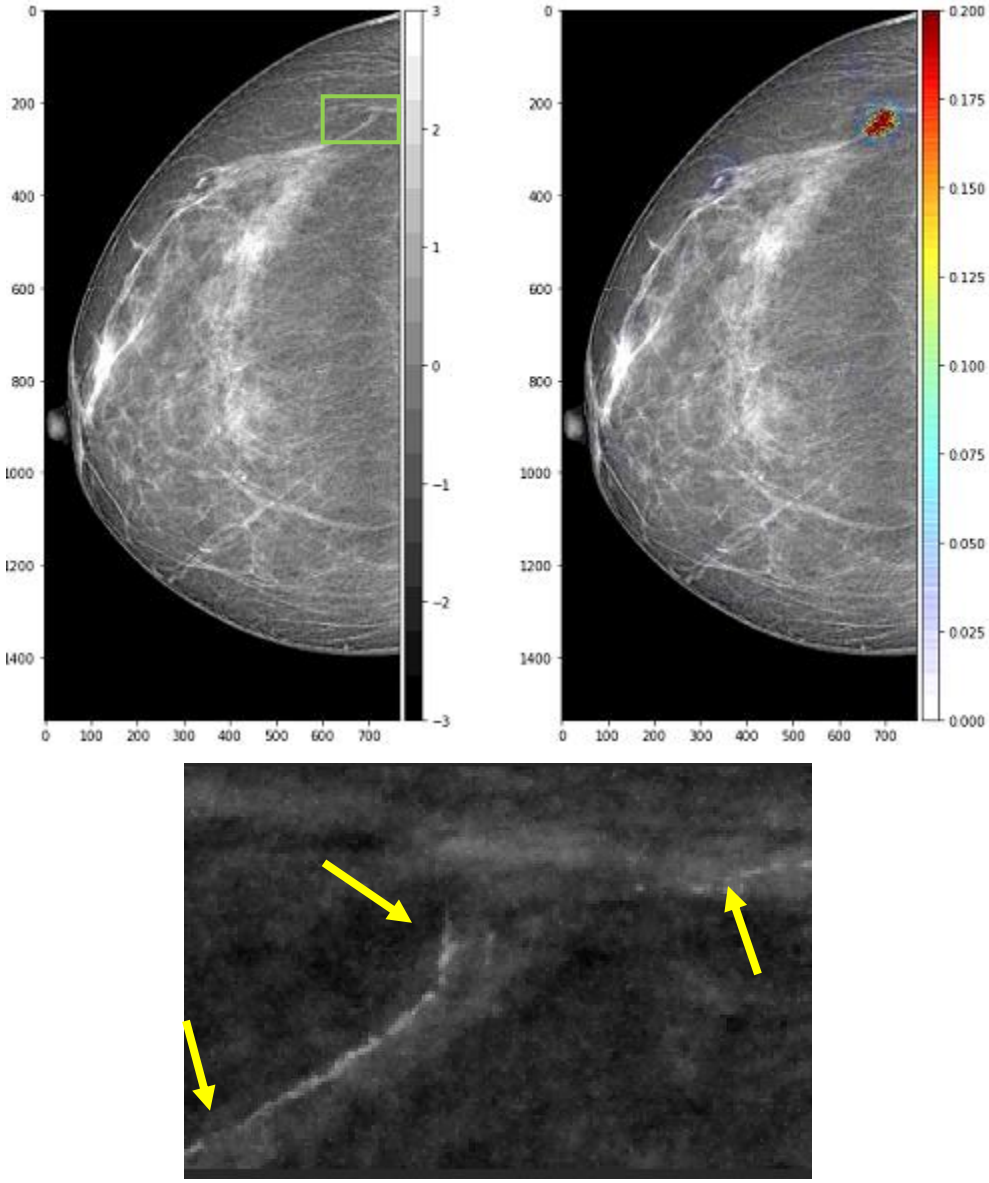


Figure 4.16 *Example of true positive image having and its saliency map. (Up, left) Input $BACs^{+}$ image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down) Zoomed BACs region. BACs are white pixels indicated by yellow arrows. The output of the network for this image is equal to 0.9705 (classification threshold equal to 0.5).*

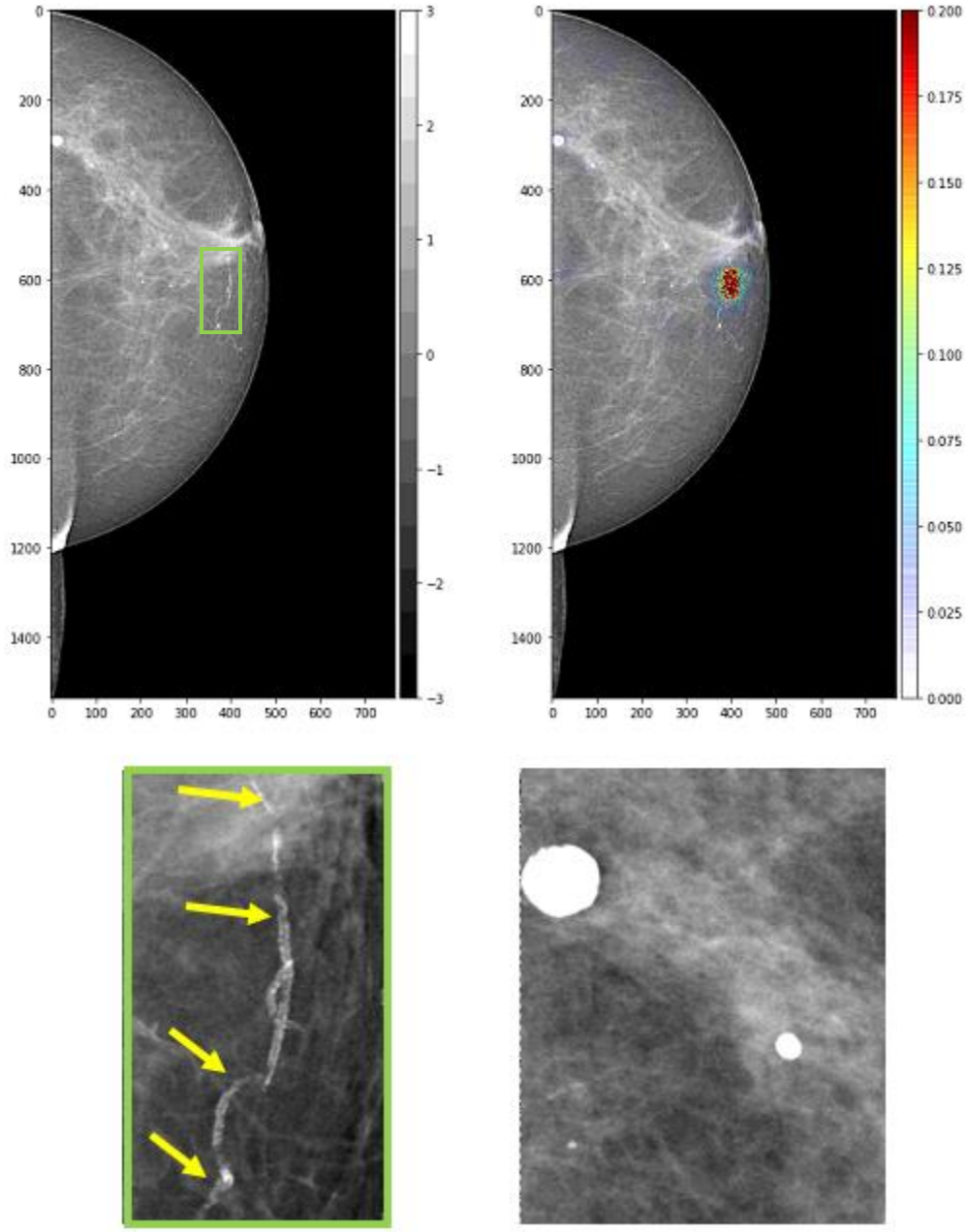


Figure 4.17 *Example of true positive having benign calcifications and its saliency map. (Up, left) Input $BACs^{+}$ image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They correspond to BACs area of the image. (Down, left) Zoomed BACs region. Very severe BACs indicated by yellow arrows. (Down, right) Zoomed region containing round benign calcifications. The prediction is not affected from the presence of other types of calcifications. The output of the network for this image is equal to 0.9997 (classification threshold equal to 0.5).*

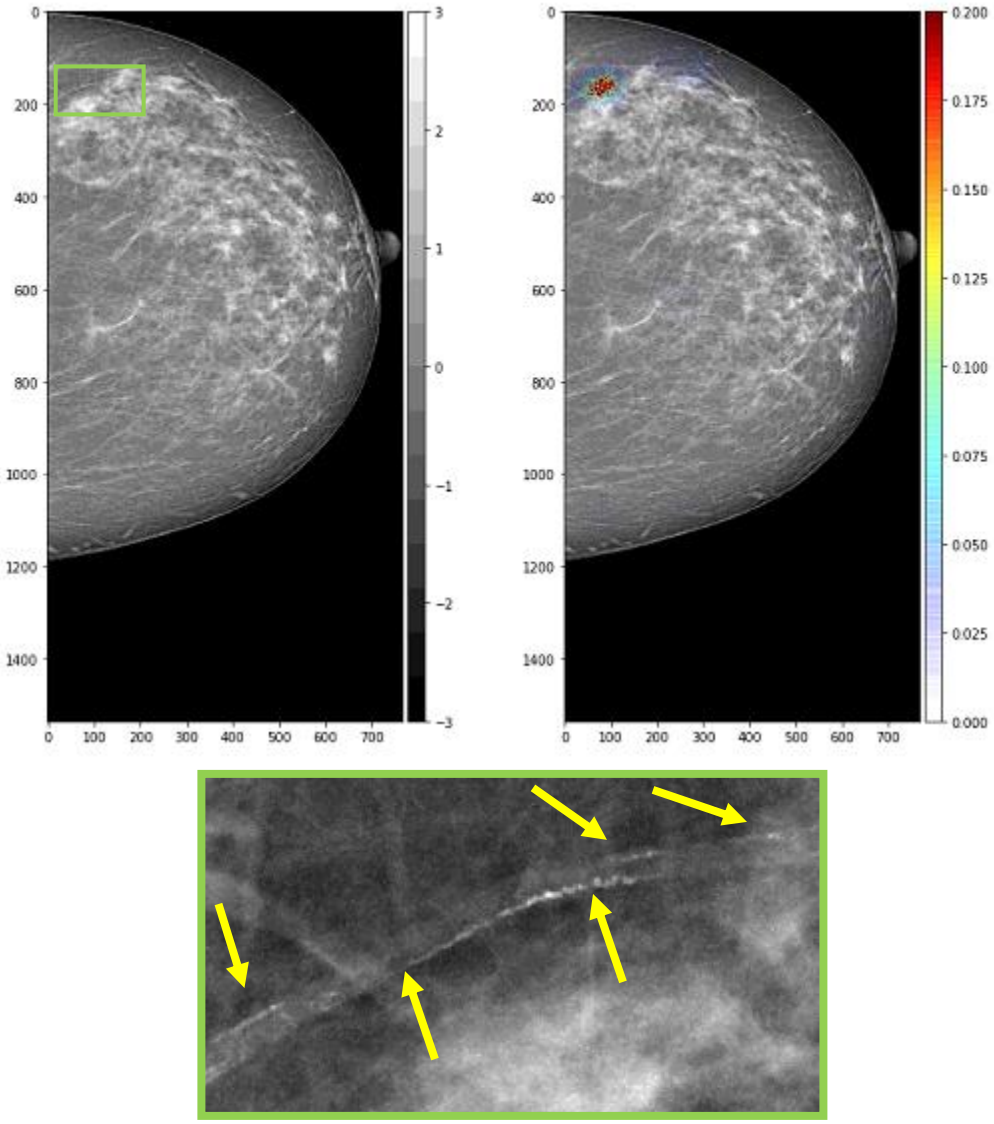


Figure 4.18 *Example of false negative image and its saliency map highlighting BACs region. (Up, left) Input BACs^{+} image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output highlighting the They are localized in an area without BACs. (Down) Zoomed highlighted region. The output of the network for this image is equal to 0.400 (classification threshold equal to 0.5).*

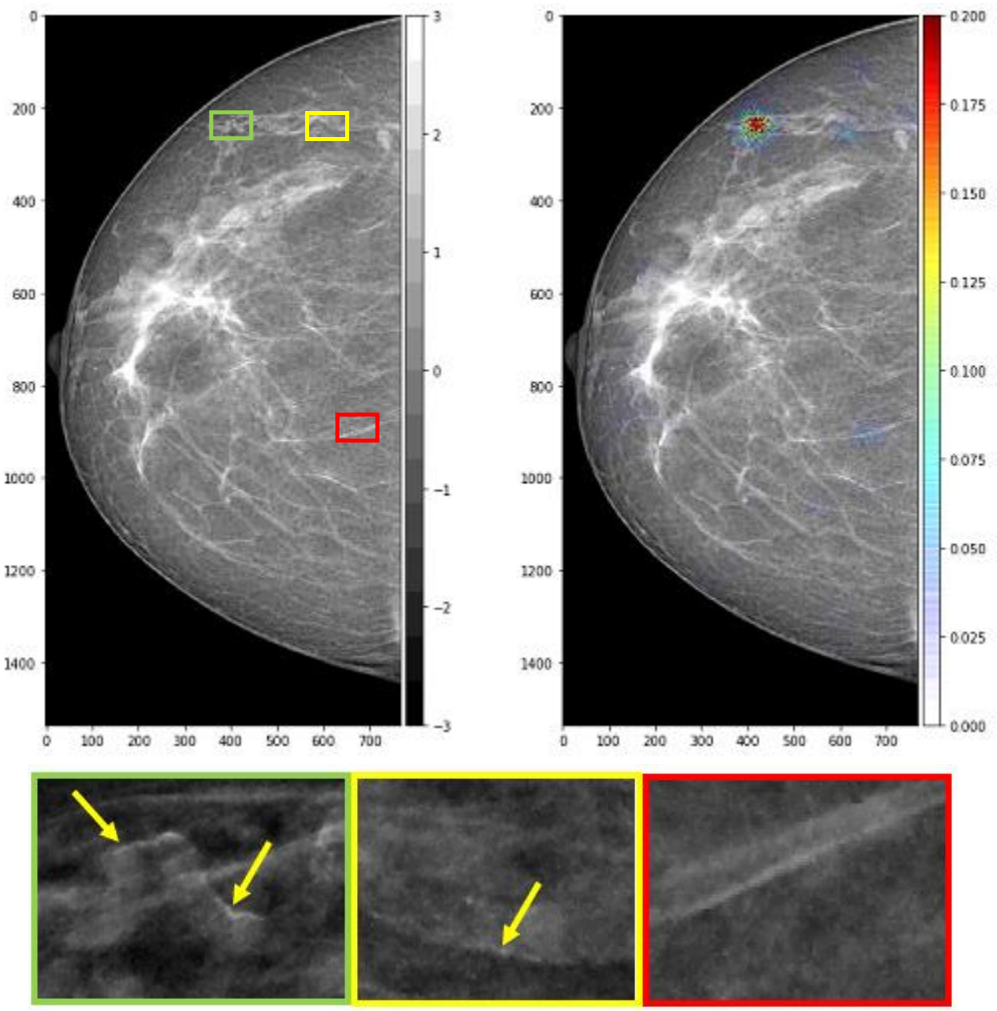


Figure 4.19 *Example of false negative image and its saliency map partially highlighting BACs region. (Up, left) Input $BACs^{+}$ image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red/blue the pixels that cause the most change in the output. They are localized in three different area of the breast. In the first, the most highlighted region in red, zoomed in the green rectangle (down, left) corresponds to a BACs region. The second one, in highlighted in blue e zoomed in the yellow rectangle (down, middle) was signed from our human reader as a dubious region. The third region contains absolutely no BACs. The output of the network for this image is equal to 0.4699 (classification threshold equal to 0.5).*

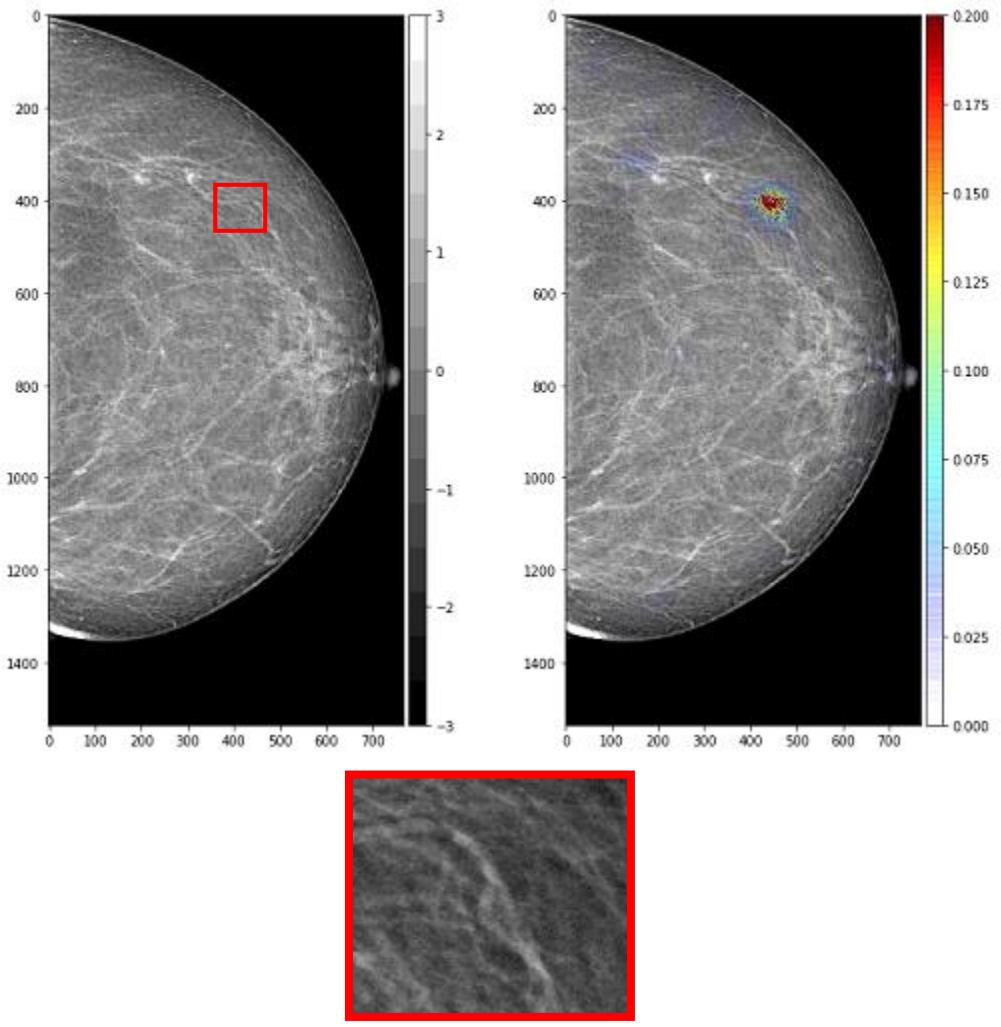


Figure 4.20 *Example of false negative image and its saliency map highlighting no BACs region. (Up, left) Input $BACs^+$ image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output highlighting a region not including BACs. (Down) Zoomed highlighted region. The output of the network for this image is equal to 0.1569 (classification threshold equal to 0.5).*

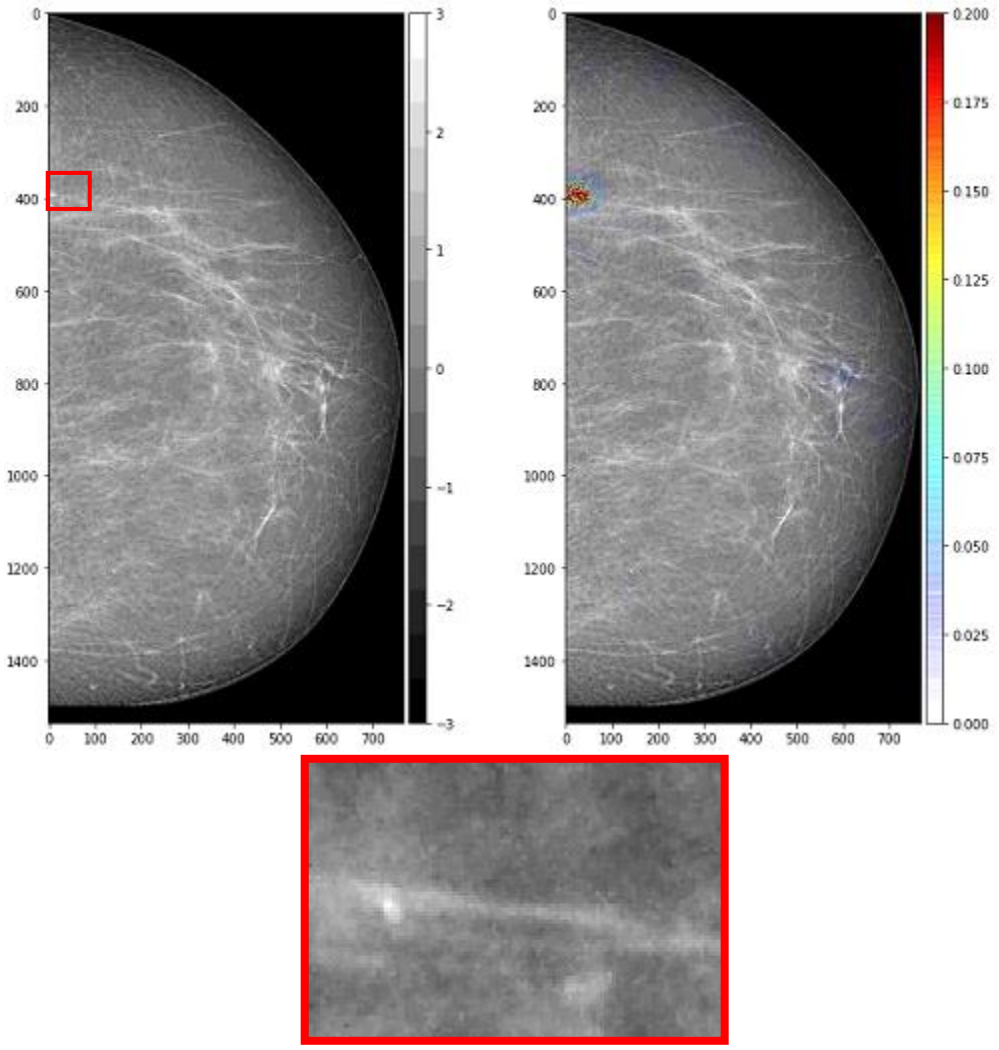


Figure 4.21 *Example of false positive image and its saliency map. (Up, left) Input $BACs^{-}$ image belonging to validation set, after preprocessing, (up, right) saliency map overlapped to the input image. In red the pixels that cause the most change in the output. They are localized in an area without BACs. (Down) Zoomed highlighted region. This image has an output of the output neuron equal to 0.6635 (classification threshold equal to 0.5).*

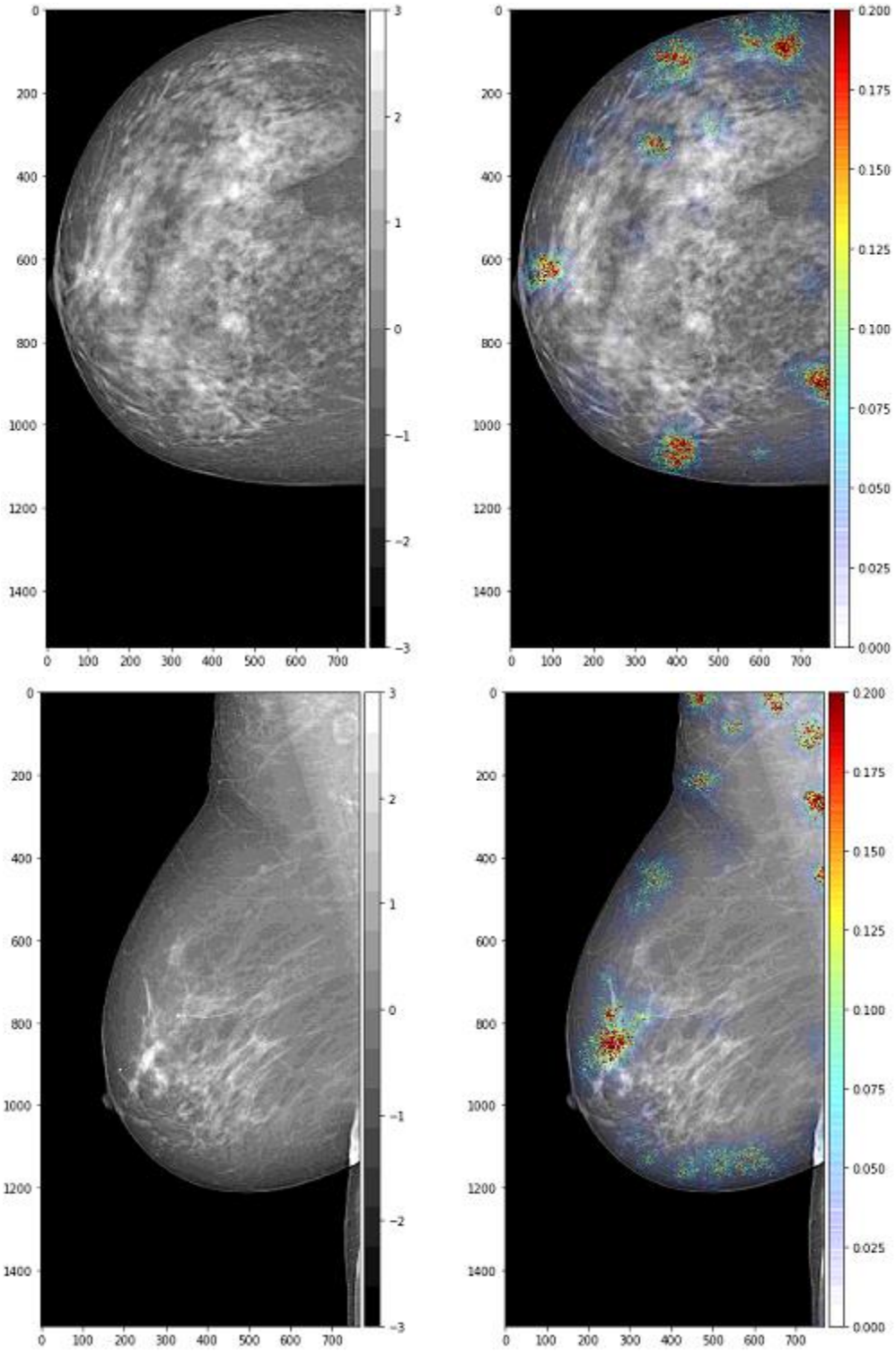


Figure 4.22 *Examples of true negative images and their saliency maps. The outputs of the net having these images as inputs are respectively 0.0284 and 0.04725*

5. DISCUSSION, CONCLUSION AND FUTURE AIMS

We investigated the feasibility of building an automated system able to add to the current cancer screening mammograms the further, cost-free information about the presence of breast arterial calcifications (BACs), recently indicated as a potential women-specific cardiovascular risk marker. Importantly, breast cancer and cardiovascular risk are major life-threatening causes specific to women population, starting from their middle age. Clearly, this work did not deal with breast cancer detection; nonetheless, it was assumed to exploit breast cancer screening mammograms without any modification, thus sharing the same confounding factors given by the wide variability of breast size and density, also aiming at avoid any confusion of BACs with benign or malignant oncological lesions of any nature.

Few experiments relevant to automatic BACs segmentation have been recently describe, though suffering of sever biases due to manual annotation gold standards. For this purpose, we formulated the problem as a two-class classification problem. A full mammogram, after preprocessing, is fed into a deep CNN that extracts and weights imaging features to provide information about BACs presence ($BACs^{+}$) or absence ($BACs^{-}$). To this automatic screening, the only information about the ROI or ROIs of highest influence for the classification is given to the radiologist in the form of heat-map. In the end, the duty of a quantitative or semi-quantitative reporting on BAC number, size, and severity is fully left to the radiologist. Nonetheless, a great relief of workload is given by pinpointing the $BACs^{+}$ cases, with a prevalence of 10%, which is hoped to result in about 10 times less manual preliminary screening. It is also unlikely that BACs will ever be a diagnostic tool per-se; still, they may drive selective decision about further costly and invasive investigations on cardiovascular risk, up to coronarography.

We overcame the problem of the huge datasets needed to train and validate a deep learning model using a transfer learning strategy. Thus, our deep CNN classifier was built starting from the VGG16 convolutional base, pretrained for general image recognition on thousands of non-medical images. This base was customized according to our input image size, also investigating the number of pretrained layers to be kept, frozen to the original

training (basic image feature extraction), how many further ones to be fine-tuned to the target problem (tuning of high level features), eventually stacking additional layers to lead to the final dichotomic classification. The result was a CNN-based classifier newly designed in three stages: i) frozen to general purpose feature extraction; ii) fine-tuned to the target medical images; iii) trained from scratch the fully connected layers to deliver the wanted output.

We set most of network hyperparameters using a-priori knowledge derived by the available literature. We preserved most of the hyperparameters that characterize the convolutional base of the VGG16 Network like neuron receptive field size, number of filters for each convolutional as well as number and properties of pooling layers [60]. BACs are expected to appear at different locations on different mammographic projections and across different subjects. This potential limitation turned out to be main reason for using convolutional neural networks for BACs detection. Indeed, the parameter sharing approach that characterize the VGG16 model allows to reduce the number of model parameters while learning a high number of imaging features that are applied at all image locations. Parameter sharing assumption may not be effective when the CNN is expected to learn specific features at different image locations (e.g. like facial characteristics in portrait pictures [107]). On the contrary, it is a powerful approach when specific features have to be found at the same time in multiple locations, like in our case. For this reason, we fixed the convolutional base of the VGG16 model pretrained natural images to exploit its robustness in detecting a large number of low-level features and train the remaining part of our classifiers using a limited amount of data compared to the large number of parameters that characterize the model.

We also selected the optimization algorithm among those available in the literature. Among them, we choose the Adam optimizer since it proved to be computationally efficient and suitable for machine learning projects that deal high-dimensional parameter spaces[97]. Indeed, one of the downsides of VGG16-derived models is the large number of parameters to be optimized during the training process. The number of parameters in VGG16-derived models are usually higher that 10^7 and are mainly located in the fully connected part of the network. In addition, Adam optimizer have been reported to be

suitable for machine learning projects that need to be developed using large database[97]. The detection of $BACs^+$ images will require, on a long-term perspective, to train our CNN on a massive number of images allowing us to have a large absolute number of positive cases even considering a prevalence near to 10%. This condition makes Adam a suitable solution for the task at issue also for future development steps.

Remaining hyperparameters were then optimized through a two-step optimization process that allowed us to determine: the optimal learning rate, the maximum number of epochs, the number of convolutional layers to fine-tune, and the number of hidden units of the fully connected classifier. Among those hyperparameters, the number of convolutional layers to fine tune and the number of hidden units in the fully connected layers largely impacted model performances. The systematic identification of the number convolutional layers to be fine-tuned allowed us to extract the high-level features related to BACs appearance useful to discriminate them from other type of calcifications or hyperintense tubular breast structures. As previously stated, BAC prevalence is relatively low in our database while different confounding factors were highly prevalent in both $BACs^+$ and $BACs^-$ images. In this light, we took advantage of the high prevalence of cofounding factor in both negative and positive cases to train the proposed network to distinguish between BACs and other confounding structures. As a result, the proposed network showed the ability to focus on arterial calcifications even in presence of several confounding hyperintense regions, as shown in Figure 5.1

The chosen CNN architecture allows to extract a very large number of imaging features, which represents a desirable characteristic when dealing with complex and highly variable morphologies. Nonetheless, the extraction of many imaging features may cause model overfitting. To reduce model complexity and prevent overfitting we downsized as much as possible the number hidden units in the fully connected layers. This process allowed us to select and nonlinearly combine most meaningful features improving at the same time model performances and training efficiency. In this project, a good tradeoff between the need of a high-dimensional feature space and the need to avoid overfitting

was reached using a rigorous parameter search process that allowed to reach good performances in the validation database.

The resulting deep CNN classifier was composed by 16 layers: 13 convolutional layers of which 8 were kept frozen and 5 were fine-tuned on our dataset images, followed by three fully connected layers with respectively 256 neurons, 256 neurons, and 1 neuron, respectively. The last one with a sigmoidal activation function in order to implement the dichotomic classification.

Importantly, it was possible to fix most of the structure by means of the training and validation over a preliminary and reduced database of 168 patients and 684 images with a 30% prevalence of $BACs^{+}$ resulting from the annotation by three expert viewers. Conversely, the subsequent increase of the annotated database up to 717 patients and 2684 images matching the 10% demographic prevalence did not achieve the expected results, due to still a too low size with such an imbalanced prevalence.

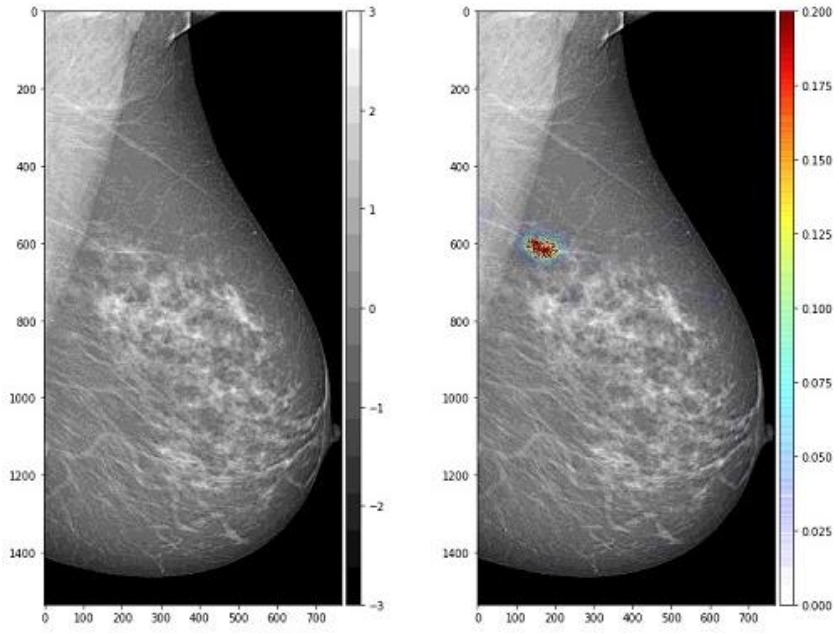


Figure 5.1 *Example of breast arterial calcification correctly detected in presence of several high intensity objects with tubular morphology.*

For this reason, it was decided: i) to randomly down sample the $BACs^{-}$subjects to go back to a 30% prevalence of $BACs^{+}$, optimal for training (248 patients, 992 images); ii) not to attempt at this stage the final testing phase, given the insufficient size of the annotated database reached so far. The significant size increase of the final dataset,

compared to the preliminary one, was conversely exploited for a better validation, considering a random partition in 3 subsets DB1, DB2, DB3. Each parameter change was chosen by evaluating the effects on both DB1 and DB2. Lastly, when we found an eligible hyperparameter combination, we performed a further training on DB3.

In place of testing (scheduled as soon as further data will be available) we further evaluated the resulting architecture and learning strategy performing a 7-fold cross validation using the F1 score as performance metric to account for the class imbalance. Model performances calculated on the training datasets ranged from 0.765 to 0.897 with mean and SD values equal to 0.830 ± 0.042. In the same way, F1 scores calculated on the validation datasets varied across models, ranging from 0.653 to 0.840 with mean and SD values equal to 0.744 ± 0.094. The training process was stopped once the validation dataset loss function reached its minimum.

Our results suggest that with the current strategy the model was not able to extract all the information contained within the training data. So, further investigations are needed to fully exploit a larger training dataset information content and test the BACs classifier performance on a new independent testing dataset. The models showed generally good performance in terms of precision (range = [0.842-0.950], mean = 0.864 and SD = 0.0.040) while showing lower recall values (range = [0.433 -0.772], mean = 0.667, SD = 0.132) in the validation set. Therefore, future studies will focus on the improvement of F1 values thus leading to false negatives reduction without increasing the number of false positives.

Saliency maps show that other types of breast calcifications do not impact positive predictions. However, sometimes other tubular breast structures are incorrectly classified as BACs. In addition, we found different appearance on false negatives. In some cases, the saliency focused on other structures of the breast, ignoring BACs. In other cases, BACs were correctly detected by the saliency, but the output value of the network was slightly below the sigmoid classification threshold and the image was therefore classified as negative. Moreover, the observation of the saliency maps has allowed us to ascertain the feasibility of transforming global information, such as an image-level label, into a local one, allowing the localization of BACs within the large breast area. Indeed, as shown in the several reported examples, the saliency maps of true BACs^+ images clearly highlight

ROIs around BACs, or at the least the main one, thus adding visual evidence to the reliability of estimated class and potentially driving the detailed radiologist's inspection.

It should be also noted that when multiple BAC segments are present in the image, not all of them are highlighted by the saliency map (see figure 5.2). This is due to the binary nature of the proposed classification problem, where the network is asked to detect at least a single BAC to classify the full image as positive. This issue may be overcome in the future by using the furtherly improved CNN binary classifier to build a new regression model able to estimate BACs burthen on a continuous scale. The enriched feature space selected to predict the BACs burden will allow to detect all BACs contained in the images towards a fully automatic method of BACs segmentation like the one proposed by Shimoda *et al.* [108], that combined CNN feature maps and saliency maps.

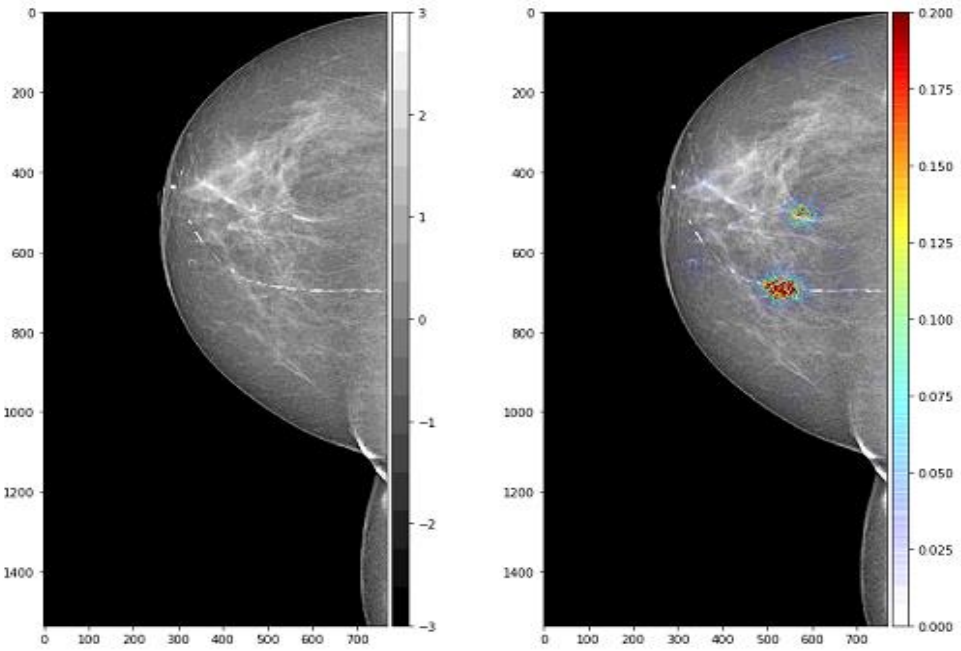


Figure 5.2 *Example of breast artery with multiple calcified segments. Despite the large extension of BAC segments along the vessel, the saliency map shows only one bright spot as a result of the binary classification task performed the developed convolutional neural network.*

What has been said previously reveals the need to further improve the model, with particular focus on the positive class prediction performance. The current model has limitations, mainly related to the small portion of the hyperparameter space investigated

so far. Both the model parameters and hyperparameters need to be further tuned on more data to improve classification performances. In fact, despite the use of transfer learning and data augmentation techniques the huge number of 13.176.577 parameters has to be trained. Lastly, the problem of class imbalance was not totally addressed by the implemented weighted training strategy, leading to a high number of false negatives compared to false positives.

In the future, further development will focus on model performance improvement by enlarging the database reaching a 50:50 prevalence. With this dataset, we will deeply investigate more the hyperparameter space until reaching very high performances in both negative and positive classes. At this point it will be necessary to test the predictive power of the CCN on a dataset that reflect the real $BACs^+$ prevalence in the cancer screening population. Lastly, it will be necessary to avoid possible data mismatch issues to make the model prediction effective in real world domain. Once having obtained a highly performing binary BACs classifier, the next step will be using transfer learning to build a CNN based regression model that automatically detects BACs and calculates a quantitative BAC score allowing the screened population to be segmented in more than two classes. BACs are expected to play an important role in the identification of moderate risk women, task that requires by definition the identification of more than two classes. Finally, as a very long-term goal, this quantitative score regression will be used to stratify women CV risk exploiting mammographic content to reduce the mortality of two of the main cause of death among women, namely CVDs and breast cancer.

The proposed model allows to detect mammographic images positive to BACs presence with promising performances. Moreover, it apparently allows to localize BACs position within the image avoiding confounding structures present in mammograms. These results pave the way towards a fully automatic detection and grading of BACs in routine mammograms acquired for breast cancer screening purposes to improve CVD risk stratification in asymptomatic women. However, further improvements on larger image dataset are needed to improve model performance and prove results generalizability on an independent testing dataset.

Bibliography

1. World Health Organization (WHO). Technical package for cardiovascular disease management in primary health care. *Report*. 2016:1-76. doi:10.1016/j.cortex.2008.06.011

2. Trimboli RM, Codari M, Guazzi M, Sardanelli F. Screening mammography beyond breast cancer: breast arterial calcifications as a sex-specific biomarker of cardiovascular risk. *Eur J Radiol*. 2019;119(August). doi:10.1016/j.ejrad.2019.08.005

3. Greenland P, Knoll MD, Stamler J, et al. Major Risk Factors as Antecedents of Fatal and Nonfatal Coronary Heart Disease Events. *J Am Med Assoc*. 2003;290(7):891-897. doi:10.1001/jama.290.7.891

4. Rinkuniene E, Petrulioniene Ž, Laucevičius A, Ringailaite E, Laučyte A. Prevalence of conventional risk factors in patients with coronary heart disease. *Medicina (B Aires)*. 2009;45(2):140-146. doi:10.3390/medicina45020018

5. Wenger NK. Transforming cardiovascular disease prevention in women: Time for the pygmalion construct to end. *Cardiol*. 2015;130(1):62-68. doi:10.1159/000370018

6. D'Agostino RB, Vasan RS, Pencina MJ, et al. General cardiovascular risk profile for use in primary care: The Framingham heart study. *Circulation*. 2008;117(6):743-753. doi:10.1161/CIRCULATIONAHA.107.699579

7. DeFilippis AP, Young R, Carrubba CJ, McEvoy JW, Budoff MJ B, RS, Kronmal RA, McClelland RL, Nasir K BM. An analysis of calibration and discrimination among multiple cardiovascular risk scores in a modern multiethnic cohort. *Ann Intern Med*. 2015;162(4):266–275. doi:0.7326/M14-1281

8. Bui QM, Daniels LB. A Review of the Role of Breast Arterial Calcification for Cardiovascular Risk Stratification in Women. *Circulation*. 2019;139(8):1094-1101. doi:10.1161/CIRCULATIONAHA.118.038092

9. Kavousi M, Desai CS, Ayers C, et al. Prevalence and prognostic implications of coronary artery calcification in low-risk women: A meta-analysis. *JAMA - J Am Med Assoc*. 2016;316(20):2126-2134. doi:10.1001/jama.2016.17020

10. Greenland P, Blaha MJ, Budoff MJ, Erbel R, Watson KE. Coronary Calcium Score and Cardiovascular Risk. *J Am Coll Cardiol*. 2018;72(4):434-447. doi:10.1016/j.jacc.2018.05.027

11. Iribarren C, Molloi S. Breast Arterial Calcification: A New Marker of Cardiovascular Risk? *Curr Cardiovasc Risk Rep*. 2013;7(2):126-135. doi:10.1007/s12170-013-0290-4

12. Zazzeroni L, Faggioli G, Pasquinelli G. Mechanisms of Arterial Calci fi cation : The Role of Matrix Vesicles. *Eur J Vasc Endovasc Surg*. 2018;55(3):425-432. doi:10.1016/j.ejvs.2017.12.009

13. Hendriks EJE, De Jong PA, van der Graaf Y, Mali WPTM, van der Schouw YT, Beulens JWJ. Breast arterial calcifications: A systematic review and meta-analysis of

their determinants and their association with cardiovascular events. *Atherosclerosis*. 2015;239(1):11-20. doi:10.1016/j.atherosclerosis.2014.12.035

14. Rennenberg RJMW, Schurgers LJ, Kroon AA, Stehouwer CDA. Arterial calcifications. *J Cell Mol Med*. 2010;14(9):2203-2210. doi:10.1111/j.1582-4934.2010.01139.x

15. Polonsky TS, Greenland P. Breast Arterial Calcification: Expanding the Reach of Cardiovascular Prevention. *Circulation*. 2017;135(6):499-501. doi:10.1161/CIRCULATIONAHA.116.025277

16. Cheng JZ, Cole EB, Pisano ED, Shen D. Detection of arterial calcification in mammograms by random walks. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*. 2009;5636 LNCS(CC):713-724. doi:10.1007/978-3-642-02498-6_59

17. Kemmeren JM, van Noord PAH, Beijerinck D, Fracheboud J, Banga J-D, van der Graaf Y. Arterial Calcification Found on Breast Cancer Screening Mammograms and Cardiovascular Mortality in Women: The DOM Project. *Am J Epidemiol*. 1998;147(4):333-341. doi:10.1093/oxfordjournals.aje.a009455

18. Margolies L, Salvatore M, Hecht HS, et al. Digital Mammography and Screening for Coronary Artery Disease. *JACC Cardiovasc Imaging*. 2016;9(4):350-360. doi:10.1016/j.jcmg.2015.10.022

19. Trimboli R.M., Codari M., Cozzi M., Monti C. B., Capra D., Nenna C., Spinelli D., Di Leo G., Baselli G SF. Semiquantitative score of breast arterial calcifications on mammography (BAC-SS): intra- and inter-reader reproducibility. Submitted. 2020.

20. Tzikopoulos SD, Mavroforakis ME, Georgiou H V., Dimitropoulos N, Theodoridis S. A fully automated scheme for mammographic segmentation and classification based on breast density and asymmetry. *Comput Methods Programs Biomed*. 2011;102(1):47-63. doi:10.1016/j.cmpb.2010.11.016

21. Pisano, Etta D., Yaffe Martin J. KCM. DIGITAL MAMMOGRAPHY. In: *DIGITAL MAMMOGRAPHY*. 1st ed. ; 2004:363-371.

22. Rangayyan RM, Nguyen TM, Nandi AK. Effect of Pixel Resolution on Texture Features of Breast Masses in Mammograms. 2010;23(5):547-553. doi:10.1007/s10278-009-9238-0

23. Baldelli P, Phelan N, Breast N, Programme S, Egan G. A novel method for contrast-to-noise ratio (CNR) evaluation of digital mammography detectors. 2009;(September 2016). doi:10.1007/s00330-009-1409-3

24. Jen CC, Yu SS. Automatic detection of abnormal mammograms in mammographic images. *Expert Syst Appl*. 2015;42(6):3048-3055. doi:10.1016/j.eswa.2014.11.061

25. Joseph AM, John MG, Dhas AS. Mammogram image denoising filters: A comparative study. *2017 Conf Emerg Devices Smart Syst ICEDSS 2017*. 2017;(November 2018):184-189. doi:10.1109/ICEDSS.2017.8073679

26. Cunningham L. The anatomy of the arteries and veins of the breast. *J Surg Oncol*. 1977;9(1):71-85. doi:10.1002/jso.2930090112

27. Taylor GI, Caddy CM, Watterson PA CJ. The venous territories (venosomes) of the human body: experimental study and clinical implications. *Plast Reconstr Surg*. 1990;86(2):185-213. doi:10.1097/00006534-199008000-00001

28. Jesinger RA, Lattin GE, Ballard EA, Zelasko SM, Glassman LM. Vascular abnormalities of the breast: Arterial and venous disorders, vascular masses, and mimic lesions with radiologic-pathologic correlation. *Radiographics*. 2011;31(7):117-137. doi:10.1148/rg.317115503

29. Sweeney RJI, Lewis SJ, Hogg P, McEntee MF. A review of mammographic positioning image quality criteria for the craniocaudal projection. *Br J Radiol*. 2018;91(1082):1-9. doi:10.1259/bjr.20170611

30. Checka, C. M., Chun, J. E., Schnabel, F. R., Lee, J., & Toth H. The Relationship of Mammographic Density and Age: Implications for Breast Cancer Screening. *Am J Roentgenol*. 2012;198(3):W292–W295. doi:10.2214/ajr.10.6049

31. Linver MN. 4-19 Mammographic Density and the Risk and Detection of Breast Cancer. *Breast Dis*. 2008;18(4):364-365. doi:10.1016/S1043-321X(07)80400-0

32. Arancibia Hernández PL, Taub Estrada T, López Pizarro A, Díaz Cisternas ML, Sáez Tapia C. Breast calcifications: Description and classification according to BI-RADS 5th edition. *Rev Chil Radiol*. 2016;22(2):80-91. doi:10.1016/j.rchira.2016.06.004

33. Cheng JZ, Chen CM, Cole EB, Pisano ED, Shen D. Automated delineation of calcified vessels in mammography by tracking with uncertainty and graphical linking techniques. *IEEE Trans Med Imaging*. 2012. doi:10.1109/TMI.2012.2215880

34. Van Noord PAH, Beijerinck D, Kemmeren JM, Van Der Graaf Y. Mammograms may convey more than breast cancer risk: Breast arterial calcification and arterio-sclerotic related diseases in women of the DOM cohort. In: *European Journal of Cancer Prevention*. ; 1996.

35. Moshyedi AC, Puthawala AH, Kurland RJ, O'Leary DH. Breast arterial calcification: Association with coronary artery disease: Work in progress. *Radiology*. 1995. doi:10.1148/radiology.194.1.7997548

36. Kelly BS, ScanlON E, Heneghan H, et al. Breast Arterial Calcification on screening mammography can predict significant Coronary Artery Disease in women. *Clin Imaging*. 2018. doi:10.1016/j.clinimag.2017.10.021

37. Molloi S, Xu T, Ducote J, Iribarren C. Quantification of breast arterial calcification using full field digital mammography. *Med Phys*. 2008. doi:10.1118/1.2868756

38. Molloi S, Mehraien T, Iribarren C, Smith C, Ducote JL, Feig SA. Reproducibility of Breast Arterial Calcium Mass Quantification Using Digital Mammography. *Acad Radiol*. 2009. doi:10.1016/j.acra.2008.08.011

39. Wang J, Ding H, Bidgoli FA, et al. Detecting Cardiovascular Disease from Mammograms with Deep Learning. *IEEE Trans Med Imaging*. 2017. doi:10.1109/TMI.2017.2655486

40. Mordang JJ, Gubern-Mérida A, Den Heeten G, Karssemeijer N. Reducing false positives of microcalcification detection systems by removal of breast arterial calcifications. *Med Phys*. 2016;43(4):1676-1687. doi:10.1118/1.4943376

41. Cheng JZ, Chen CM, Shen D. Identification of breast vascular calcium deposition in digital mammography by linear structure analysis. *Proc - Int Symp Biomed Imaging*. 2012:126-129. doi:10.1109/ISBI.2012.6235500

42. Vargas R, Mosavi A, Ruiz R. Deep Learning: A Review. *Adv Intell Syst Comput*. 2018;(July). doi:10.20944/preprints201810.0218.v1

43. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*. 1980;36(4):193-202. doi:10.1007/BF00344251

44. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436-444. doi:10.1038/nature14539

45. Garcia-Perez A, Gheriss F, Bedford D, Garcia-Perez A, Gheriss F, Bedford D. Measurement, Reliability, and Validity. *Des Track Knowl Manag Metrics*. 2019:163-182. doi:10.1108/978-1-78973-723-320191012

46. Xu X, Jiang X, Ma C, et al. Deep Learning System to Screen Coronavirus Disease 2019 Pneumonia. 2020:1-29. http://arxiv.org/abs/2002.09334.

47. Mane H, Gopala V, Matcha R. Image classification using Deep learning. 2018;(August). doi:10.14419/ijet.v7i2.7.10892

48. Shaheen F, Verma B, Asafuddoula M. Impact of Automatic Feature Extraction in Deep Learning Architecture. *2016 Int Conf Digit Image Comput Tech Appl DICTA 2016*. 2016. doi:10.1109/DICTA.2016.7797053

49. Samuel A.L. Some Studies in Machine Learning. *IBM J Res Dev*. 2000;44(1.2).

50. Lecun Y, Bengio Y, Hinton G. Deep learning. 2015;(May). doi:10.1038/nature14539

51. Pesapane F, Codari M, Sardanelli F. Artificial intelligence in medical imaging : threat or opportunity ? Radiologists again at the forefront of innovation in medicine. 2018.

52. Deng L, Way OM, Yu D, Way OM. Deep Learning : Methods and Applications.

53. Anwar SM, Majid M, Qayyum A, Awais M, Alnowami M, Khan MK. Medical Image Analysis using Convolutional Neural Networks: A Review. *J Med Syst*. 2018;42(11). doi:10.1007/s10916-018-1088-1

54. Hubel BYDH, Wiesel ADTN. RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT ' S VISUAL CORTEX From the Neurophysiolojy Laboratory , Department of Pharmacology central nervous

system is the great diversity of its cell types and inter- receptive fields o. 1962:106-154.

55. Rawat W. Deep Convolutional Neural Networks for Image Classification : A Comprehensive Review. 2017;2449:2352-2449. doi:10.1162/NECO

56. Nair Vinoid, Geoffrey E.Hilton (Department of Computer Science, University of Toronto, Toronto, ON M5S 2G4 C. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proc 27th Int Conf Mach Learn*. 2010:807-814.

57. Yoshua Bengio, Patrice Simard, Paolo Frasconi. Learning Long-term Dependencies with Gradient Descent is Difficult. *IEEE Trans Neural Netw*. 2014;5(2):157. http://www.dsi.unifi.it/~paolo/ps/tnn-94-gradient.pdf.

58. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. *ICML Work Deep Learn Audio, Speech Lang Process*. 2013;28.

59. Tajbakhsh N, Shin JY, Gurudu SR, et al. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Trans Med Imaging*. 2016;35(5):1299-1312. doi:10.1109/TMI.2016.2535302

60. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc*. 2015:1-14.

61. lecun-89e.pdf.

62. Andrew G, Bilmes J. Backpropagation in Sequential Deep Neural Networks. *Nips*. 2013:1-9. https://homes.cs.washington.edu/~galen/files/nips2013.pdf.

63. Srinivas S, Venkatesh Babu R. Learning neural network architectures using backpropagation. *Br Mach Vis Conf 2016, BMVC 2016*. 2016;2016-Septe:104.1-104.11. doi:10.5244/C.30.104

64. Ruder S. An overview of gradient descent optimization. 2016:1-14.

65. Alibakshi A. Strategies to develop robust neural network models: prediction of flash point as a case study. *Anal Chim Acta*. 2018;(May). doi:10.1016/j.aca.2018.05.015

66. Guo X, Yin Y, Dong C, Yang G, Zhou G. On the Class Imbalance Problem *. 2016;(October). doi:10.1109/ICNC.2008.871

67. Jeni LA, Cohn JF, De La Torre F. Facing Imbalanced Data--Recommendations for the Use of Performance Metrics. In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE; 2013:245-251. doi:10.1109/ACII.2013.47

68. Demler O V., Pencina MJ, D'Agostino RB. Misuse of DeLong test to compare AUCs for nested models. *Stat Med*. 2012;31(23):2577-2587. doi:10.1002/sim.5328

69. Weiss K, Khoshgoftaar TM, Wang D. *A Survey of Transfer Learning*. Springer International Publishing; 2016. doi:10.1186/s40537-016-0043-6

70. Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*. 2018;11141 LNCS:270-279. doi:10.1007/978-3-030-01424-7_27

71. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis*. 2015;115(3):211-252. doi:10.1007/s11263-015-0816-y

72. Deng J, Dong W, Socher R, Li L-J, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE; 2009:248-255. doi:10.1109/CVPR.2009.5206848

73. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM*. 2017;60(6):84-90. doi:10.1145/3065386

74. Szegedy C, Wei Liu, Yangqing Jia, et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2015:1-9. doi:10.1109/CVPR.2015.7298594

75. He K. Deep Residual Learning for Image Recognition.

76. Tammina S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *Int J Sci Res Publ*. 2019;9(10):p9420. doi:10.29322/ijsrp.9.10.2019.p9420

77. Wu Z, Nagarajan T, Kumar A, et al. BlockDrop: Dynamic Inference Paths in Residual Networks. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*. 2018;(June):8817-8826. doi:10.1109/CVPR.2018.00919

78. Raghu M. Transfusion : Understanding Transfer Learning for Medical Imaging. 2019;(NeurIPS).

79. Bar Y, Diamant I, Wolf L, Lieberman S, Konen E, Greenspan H. Chest pathology detection using deep learning with non-medical training. *Proc - Int Symp Biomed Imaging*. 2015;2015-July:294-297. doi:10.1109/ISBI.2015.7163871

80. Arevalo J, Gonzalez FA, Ramos-Pollan R, Oliveira JL, Guevara Lopez MA. Convolutional neural networks for mammography mass lesion classification. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE; 2015:797-800. doi:10.1109/EMBC.2015.7318482

81. Carneiro G, Nascimento J, Bradley AP. Unregistered Multiview Mammogram Analysis with Pre-trained Deep Learning Models. In: ; 2015:652-660. doi:10.1007/978-3-319-24574-4_78

82. Chen H, Ni D, Qin J, et al. Standard Plane Localization in Fetal Ultrasound via Domain Transferred Deep Neural Networks. *IEEE J Biomed Heal Informatics*. 2015;19(5):1627-1636. doi:10.1109/JBHI.2015.2425041

83. Otsu N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans Syst Man Cybern*. 1979;9(1):62-66. doi:10.1109/TSMC.1979.4310076

84. Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. *J Big Data*. 2019. doi:10.1186/s40537-019-0197-0

85. Chatfield K, Simonyan K, Vedaldi A, Zisserman A. Return of the Devil in the Details : Delving Deep into Convolutional Nets. :1-11.

86. Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. *J Big Data*. 2019;6(1):60. doi:10.1186/s40537-019-0197-0

87. Gardezi SJS, Awais M, Faye I, Meriaudeau F. Mammogram classification using deep learning features. In: *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE; 2017:485-488. doi:10.1109/ICSIPA.2017.8120660

88. Shallu, Mehra R. Breast cancer histology images classification: Training from scratch or transfer learning? *ICT Express*. 2018;4(4):247-254. doi:10.1016/j.icte.2018.10.007

89. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15:1929-1958.

90. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. 2010;9:249-256.

91. Kotsiantis S, Kanellopoulos D, Pintelas P. Handling imbalanced datasets : A review. *Science (80-)*. 2006;30(1):25-36. doi:10.1007/978-0-387-09823-4_45

92. Chawla N V, Bowyer KW, Hall LO. SMOTE : Synthetic Minority Over-sampling Technique. 2002;16:321-357.

93. Provost F, Fawcett T. Robust Classification for Imprecise Environments. September 2000. http://arxiv.org/abs/cs/0009007.

94. Cui Y, Jia M, Lin T, Tech C. Class-Balanced Loss Based on Effective Number of Samples.

95. Huang C, Loy CC, Tang X. Learning Deep Representation for Imbalanced Classification.

96. Wang Y. Learning to Model the Tail. 2017;(Nips).

97. Kingma DP, Ba JL. Adam: A method for stochastic optimization. *3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc*. 2015:1-15.

98. Russel Reed. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks (A Bradford Book) Paperback*.; 1999.

99. Bengio Y. Practical recommendations for gradient-based training of deep architectures. *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*. 2012;7700 LECTU:437-478. doi:10.1007/978-3-642-35289-8-26

100. Smith LN. Cyclical learning rates for training neural networks. *Proc - 2017 IEEE Winter Conf Appl Comput Vision, WACV 2017*. 2017;(April):464-472. doi:10.1109/WACV.2017.58

101. Loshchilov I, Hutter F. Sgdr: Stochastic Gradient Descent with warm Restarts. 2017:1-16.

102. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition. *proc IEEE*. 1998. http://ieeexplore.ieee.org/document/726791/#full-text-section.

103. Xu B, Wang N, Chen T, Li M. Empirical Evaluation of Rectified Activations in Convolutional Network. 2015. http://arxiv.org/abs/1505.00853.

104. Hinton G, Hinton G. A Practical Guide to Training Restricted Boltzmann Machines A Practical Guide to Training Restricted Boltzmann Machines. 2010.

105. Stone M. Cross-validatory choice and assessment of statistical predictions. *R Stat Soc*. 1974;36:111-147.

106. Simonyan K. Deep Inside Convolutional Networks : Visualising Image Classification Models and Saliency Maps arXiv : 1312 . 6034v2 [cs . CV] 19 Apr 2014. 2013:1-8.

107. Taigman Y, Yang M, Ranzato M, Wolf L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE; 2014:1701-1708. doi:10.1109/CVPR.2014.220

108. Shimoda W, Yanai K. Weakly-supervised segmentation by combining CNN feature maps and object saliency maps. *Proc - Int Conf Pattern Recognit*. 2016;0:1935-1940. doi:10.1109/ICPR.2016.7899919