



**First year master's project**

# Deep Learning based Breast Cancer Screening

Mohamed Yassine IHBACH
Wissale KHOUDRAJI

Supervised by

Pr. Ali IDRI
Phd. Hasnae ZEROUAOUI
Phd. Asmaa ZIZAAN

AL KHWARIZMI Department
University Mohamed VI Polytechnic
2020-2021

December 9, 2021

**Abstract**

Breast cancer (BC) became the most diagnosed cancer overtaking lung cancer, making it one of the deadliest diseases. Mammography is the gold standard for detecting early signs of breast cancer, which can help cure the disease during its early stages.

The objective of this project is to explore various deep learning techniques, that can be used to implement a system which learns how to detect instances of breast cancer in mammograms, and compare them. However, incorrect mammography interpretations are common and may harm patients through unnecessary treatments and operations (or a lack of treatments). Therefore, systems that can learn to detect breast cancer on their own could help reduce the number of incorrect interpretations and missed cases.

This research conducts an experimental evaluation of the newest deep Convolutional Neural Network (CNN) architectures for a binary classification of breast screening mammograms. Eight pre-trained architectures: VGG16, VGG19, DenseNet201, Inception_ResNet_V2, Inception_V3, ResNet_50, MobileNet_V2 and Xception were evaluated. The experimental evaluation were based on: four classification performance metrics (e.g. accuracy, precision, recall and f1-score), Scott Knott statistical test and Borda count voting system. The evalutation have been done over DDSM dataset with 4000 images with equal portion of benign and malignant. Moreover, DenseNet_V2 was the most efficient technique with an accuracy of 84.27% by transfer learning pre-trained ImagetNet weights.

**Keywords**

Computer-Aided Screening; Deep Convolution Neural Networks; Transfer Learning; Breast Cancer; Mammogram Classification; Image processing

# Contents

# List of Figures

# List of Tables

# List of Acronyms

**Adam** Adaptive moment estimation

**ANN** Artificial Neural Network

**BCD** Breast Cancer Detection

**BCS** Breast Cancer Screening

**BP** Backpropagation

**CAD** Computer-Aided Detection

**CBIS-DDSM** Curated Breast Imaging Subset of DDSM (dataset)

**CC** Bilateral craniocaudal (mammogram)

**CNN** Convolutional Neural Network

**CPU** Central Processing Unit

**DT** Decision Tree

**GPU** Graphical Processing Unit

**kNN** k-Nearest Neighbours

**mini-MIAS** mini Mammography Image Analysis Society (dataset)

**MLO** Mediolateral oblique (mammogram)

**MLP** Multi-Layer Perceptron

**MRI** Magnetic Resonance Imaging

**NB** Naive Bayes

**PNN** Probabilistic Neural Networks

**RAM** Random-Access Memory

**ReLU** Rectified Linear Unit

**RMSProp** Root Mean Square Prop

**ROI** Region of Interest

**SGD** Stochastic Gradient Descent

**SVM** Support Vector Machine

**TPU** Tensor Processing Unit

**WBCD** Wisconsin Breast Cancer Wisconsin (dataset)

# Acknowledgements

We have received a great deal of support and guidance throughout our first year Master at the University of Mohammed VI Polytechnic and would like to take the opportunity to thank those who helped us and motivated to always improve throughout this year.

First of all, We would like to thank our project supervisor, Pr Ali **IDRI**, whose expertise in the domain of machine learning application in the medical field, has been invaluable to this project. We would like to extend our thanks to our project's co-supervisors, Phd Hasnae **ZEROUAOUI** and Phd Asmaa **ZIZAAN**, for their practical knowledge of implementing deep learning systems and for their intuitive insights and help. We also wish to thank the professors of AL KHWARIZMI department for the knowledge they offered and offering us.

Finally, We would like to thank all our families and friends, especially our parents and siblings, for their love, encouragement and constant support to pursue our dreams.

# Chapter 1

# Introduction

## 1.1 Motivation

Breast cancer is the most common types of cancer amongst women, with statistics showing that 1 out of 8 females will be diagnosed with breast cancer in their lifetime. According to statistics released by IARC (International Agency for Research on Cancer) in December 2020, breast cancer has overtaken lung cancer as the most commonly-diagnosed cancer in the world. In the last two decades, the total number of people diagnosed with cancer almost doubled, from approximately 10 million in 2000 to 19.3 million in 2020 (who BC, 2021).

An efficient way to maximise the survival rate is by treating the disease prematurely using screening tests such as mammograms. However, no matter the expertise of radiologists scrutinizing mammograms, uncontrolled factors such as human error need to be minimised (Polat and Güneş, 2007), since the rate of missed breast cancers in the initial mammogram screenings are as high as 30% (Elter and Horsch, 2009). To show the complexity of mammogram interpretation, Figure 1.1 contains three different mammograms, either normal or abnormal (benign and malignant) cases, and how similar they all look to an untrained eye.

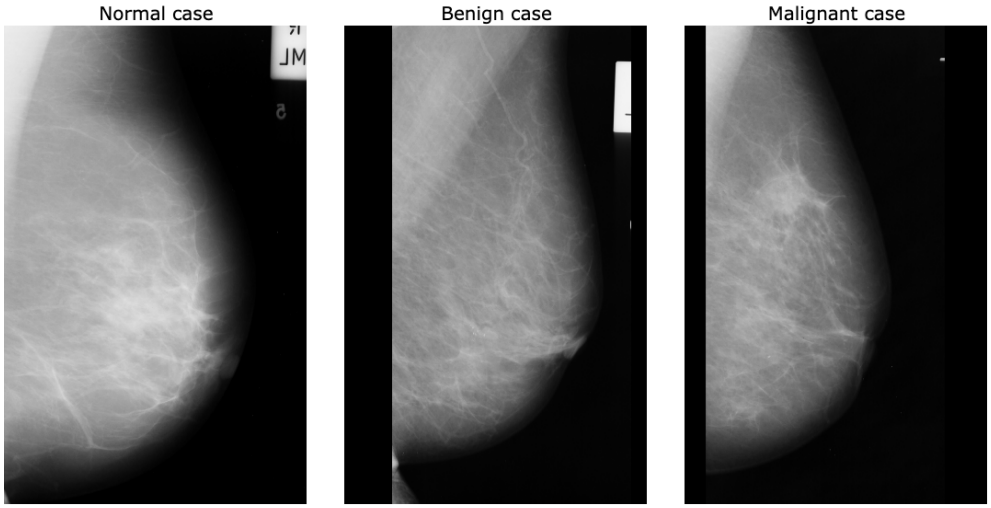



Figure 1.1: Example of three types of mammograms, including normal, benign and malignant cases. Figures extracted from the mini-MIAS dataset (Suckling, 1994).

The human error that can be done by the radiologists may lead to decisions that can eventually harm the patients. If the mammogram is classified as malignant, a breast biopsy (sample of breast tissue is removed) is generally recommended. Nevertheless, 40-60 % of biopsies are diagnosed as benign, obviously betraying the need for a correct mammography diagnosis to avoid unnecessary operations, anxiety and pain for patients (Hepsağ et al., 2017). On the one hand, breast cancer can be completely missed, leading to an absence of treatment for sick patients, while on the other hand, a case of breast cancer can be reported when in reality there is no cancerous tumor, which leads to the performance of unnecessary treatments (Elter and Horsch, 2009).

Thus, the use of computer-Assisted detection (CAD) software can help minimise the number of misinterpretations and increase the accuracy of mammography screening (Shen et al., 2017).

The motivation behind this project is to explore techniques to implement a deep learning system able to accurately detect breast cancer in order to prevent late treatments due to false negatives as well as avoid unnecessary treatments in case of false positives. Ultimately, the long-term goal of this project is to use the ensemble learning of several models based on pre-trained architectures to get higher accuracies with less computational cost. This will allow a general artificial intelligence system capable of detecting multiple forms of cancer with higher accuracies and complement radiologist expertise.

## 1.2 Problem Description

In theory, CAD systems based on deep learning techniques can highly increase the accuracy of mammogram screenings to catch early signs of breast cancer. However, such techniques require a huge amounts of data to catch up the cancer's underlying patterns, and require powerful computing resources.

## 1.3 Objectives

The main purpose of this project is the implementation of a deep learning model that can be able to learn how to detect cases of breast cancer in screening mammograms (mammograms taken before the appearing of any symptoms).

A large contextual survey should first be conducted to cover the context of the deep learning techniques applied to the field of breast cancer and to examine the existing results. This includes identifying the results obtained using different methods (e.g. traditional machine learning techniques). This step is essential since it will pilot the research towards the most promising fields, as well as the choice of techniques to be implemented and explored in the following chapters.

Finally, the final results achieved will be compared with the baseline which is a simple CNN neural network fully trained on the dataset used in this research, as well as the results published that used the same dataset.

## 1.4 Report Structure

**Introduction** Presents an overview of the subject's background through the problem description and the motivation behind this project, followed by the objectives that the project aims to achieve.

**Background & Literature review** Examines the literature and background around breast cancer detection techniques, from primitive cancer detection systems, passing by traditional machine learning methods, and finally the recent used deep learning techniques as well as the related works.

**Datasets & and Pre-processing** Describes the datasets used in this research and the pre-processing tasks applied.

**Experimental design** Explores the experimental design considerations regarding the deep learning methods to implement and the empirical method-

ology followed.

**Results & Discussion** Analyse the different results to assess the performance of the different techniques used and compare them to other results, including the baseline and relevant results appeared in the context survey.

**Conclusions** Summarises the project's accomplished objectives, its limitations, plans for future work, and a final reflection on the project as a whole.

# Chapter 2

# Background & Literature review

## 2.1 Introduction

This section is consecrated to explore the evolution of techniques applied in breast cancer detection, the background necessary for this project as well as the literature review showing the related works.

## 2.2 Breast Cancer Detection

### 2.2.1 Medical imagery screening tests & biopsies

Test screening are used to catch early signs of breast cancer before the appearance of any symptoms (e.g. lumps that can be felt by hand touching). Generally the methods used for breast cancer screening are mammograms, which are low-dosage X-rays around the breast, commonly used as initial screening tests. These mammograms have a black backgrounds and are in white for dense areas, which may conform to calcifications or masses (e.g. lumps or cysts). If a suspicious areas is detected, the mammograms are followed by ultra sounds for analysing these masses, and MRIs (MagneticResonance Imaging) for detailed imagery, usually used when a malignant tumour has been detected to get more information about it, such as the size and location, or finding additional ones (American Cancer Society, 2019). A biopsies can be conducted to confirm the screening tests' results if the screening raised a suspicious or potential presence of breast cancer. Biopsies consist of extracting cells from the breast's tissue for a lab analysis by pathologists to get more accurate results (Martin, Laura J., 2019).

As the biopsies are invasive, it is better for patients to use medical imagery tools to detect early signs of breast cancer that can be treated efficiently in-

stead of an immediate biopsy. Mammograms are the main imagery method used for early breast cancer screening (BCS) (Ramos-Pollán et al., 2012). Nevertheless, BCS using mammograms, and any form of cancer detection using medical imagery, relies on the classical diagnoses of expert radiologists (Osareh and Shadgar, 2010). The difficulty of correctly interpreting these mammograms may be subject to errors (Elter and Horsch, 2009). In fact, mammograms are 2D images of 3D breasts that correspond to the superposition of breast tissue, which increases the difficulty for a radiologist to correctly analyse patterns as masses often naturally form due to this superposition (Elter and Horsch, 2009).

### 2.2.2 Rule-based Breast Cancer Screening Systems

Since the 1970s, several CAD systems has been developed to assist and enhance radiologists interpretations of mammograms. However, first CAD systems were very primitive and did not offer much more knowledge than the experts radiologists. such undeveloped "expert" systems which consist of processing pixels manually to construct a rule-based systems that generally used *if-else-then* statements (Litjens et al., 2017).

### 2.2.3 Towards Supervised Machine Learning-based Systems

In the 1990s, machine learning techniques started to flourish and replacing rule-based expert systems, allowing the hidden patterns in the mammograms' images that was not recognised by the radiologists to be now caught by these new techniques (Litjens et al., 2017). Machine learning-based approaches, which are more general and adaptable to complex problems, were selected over statistical approaches, which are more restricted by hypothesis and less flexible, to replace expert systems as they proved its high efficiency on classification tasks and overcome statistical-based approaches (Paliwal and Kumar, 2009), particularly when dealing with large, complex and high-dimensional datasets like mammograms (Yue et al., 2018). This announce the transition from CAD systems that were completely designed by humans to systems that are trained on medical imagery datasets (Litjens et al., 2017).

yet, these machine learning algorithms could not work well on purely unprocessed data such as full-sized mammograms images. Actually, most of the machine learning algorithms tested over the task of BCS required pertinent chunck of information to be first extracted from the original images to figure out the given tasks, using algorithms such as k-Nearest Neighbour [kNN], Decision Trees [DT], Naive Bayes [NB] (Asri et al., 2016), Support Vector Machines [SVM] (Ramos-Pollán et al., 2012) and Artificial Neural Networks [ANN] (Yue et al., 2018). These relevant bits of information picked from the mammograms images correspond to features, and required to be extracted

by humans before being fed to the above-mentioned algorithms for training. These features range from visual information, such as colours, edges, corners, shapes and textures (Li and Allinson, 2008), to extracted information, such as the cell size, clump thickness, etc. (Yue et al., 2018).

The rightful following step in the evolution of BCD systems is making the model learn these features on its own from the data, instead of being fed a manually extracted features (Yala et al., 2019). Deep learning models, which are a neural networks with many hidden layers, are based in this approach. However, such models have not been successfully implemented until recent years since they require a very high computational power, mainly equipped with Graphical Processing Units (GPU) and lastely Tensor Processing Units (TPU) to be efficiently trained. That explains why until recent years, machine learning models have led the field of BCD, with some manual mammogram interpretations still being carried out by radiologists (Litjens et al., 2017), as shown in Figure 2.1.

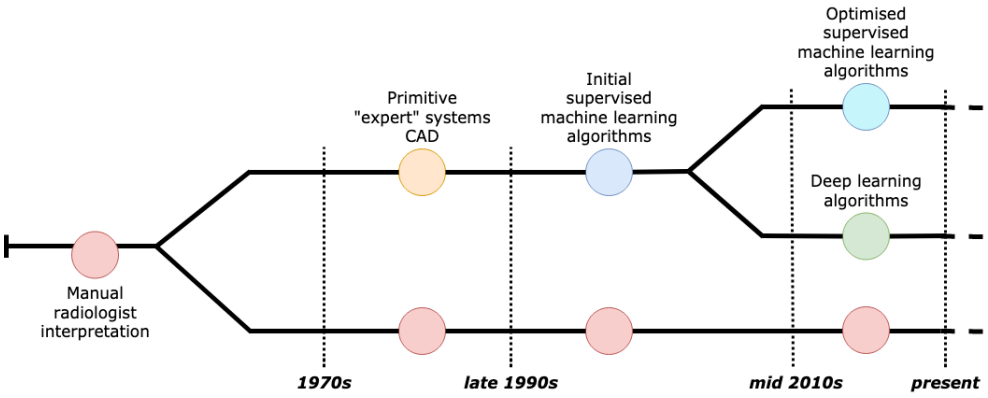



Figure 2.1: Timeline of the evolution of breast cancer detection (BCD) systems synthesising the information described in Sections 2.2.1 and 2.2.2.

## 2.3 Machine Learning meets Breast Cancer

### 2.3.1 Machine Learning Applications & Breast Cancer

#### Types of machine learning algorithms

Machine learning algorithms fall in different categories based on whether human supervision is required or not. The two main types of machine learning algorithms correspond to supervised and unsupervised learning. On the one hand, in supervised learning, the dataset is labelled, meaning every sample in the dataset includes a solution (Caruana and Niculescu-Mizil, 2006). This label $y$ is used to make a prediction $\hat{y}$ by fitting the input features $\mathbf{x}$ from a

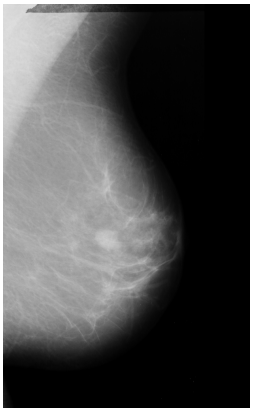
training dataset. The goal of a supervised learning algorithm is to determine the optimal parameters $\theta$ for the selected algorithm in order to minimise a loss function defined as $L(y, \hat{y})$, which corresponds to the error between $\hat{y}$ and $y$ (Litjens et al., 2017). A large variety of loss functions can be used such as the general Mean Squared Error (MSE) and Mean Absolute Error (MAE) loss functions, or more specific loss functions such as the Hinge Loss for SVMs (Géron, 2019). The main applications of supervised learning are classification and regression, with the former being the most relevant to BCD.

On the other hand, in unsupervised learning, the data is unlabelled, meaning only the input features $\mathbf{x}$ are available while the labels $y$ are not (Litjens et al., 2017). This means the algorithm cannot optimise its hyperparameters, which correspond its configurations used to define it before training (Bergstra et al., 2013), by minimising a loss function. Instead, the algorithm needs to automatically create clusters in the dataset in order to separate them into different groups. The main applications of unsupervised learning are clustering, anomaly detection, data visualisation and dimensionality reduction (Géron, 2019), rendering them irrelevant to breast cancer detection. Two other categories of machine learning algorithms exist, corresponding to semi-supervised learning and reinforcement learning, but are also irrelevant to the task of detecting breast cancer.

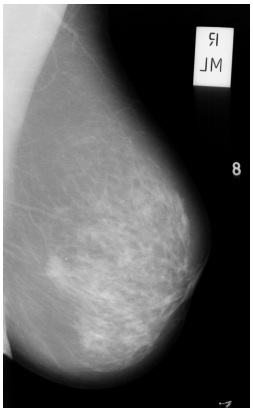
Among the two types of machine learning algorithms, the most pertinent one for the task of BCD is supervised learning as datasets of mammograms need to contain properly labelled data for each sample, indicating the status of the mammogram (Shen et al., 2017).

#### Detection of Breast Cancer

Detection corresponds to the classification of a medical image or exam, which is an interpretation that used to be entirely carried out by a radiologist before the appearance of CAD systems. The classification can be either binary or multi-class, depending on the data used (see Section 4.2). With datasets like the "Curated Breast Imaging Subset of DDSM" (Lee et al., 2017), the classification is binary as mammograms can only be classified as either "benign" or "malignant". However, when using datasets like the "Digital Database for Screening Mammography" (Heath et al., 2001), the classification becomes more interesting from a clinical point of view as mammograms can either be classified as "normal", "benign" or "malignant" (Litjens et al., 2017). An example of classification between benign and malignant mammograms can be found in Figure 2.2, reinforcing the complexity that comes with interpreting mammograms for radiologists and the need for accurate and reliable CAD systems to improve the prediction accuracies.



(a) A benign mammogram.



(b) A malignant mammogram.

Figure 2.2: Example of a breast mammogram classification, showing benign (left) and malignant (right) mammograms. Images retrieved from the mini-MIAS dataset (Suckling, 1994).

### 2.3.2 Comparison of BCD Supervised Learning Algorithms

Since the late 1990s, a rich array of supervised machine learning algorithms have been applied and tested against the task of BCD, contributing to improving accuracies for detecting breast cancer (Yue et al., 2018). The main types of algorithms used in BCD, which consist of k-Nearest Neighbour (kNN), Naive Bayes (NB), Support Vector Machines (SVM), Decision Trees (DT) and Artificial Neural Networks (ANN), are briefly explained in the ensuing sections from most simple to most complex. Their performances in BCD are then compared to draw a picture of the advantages and disadvantages that each method brings.

#### Supervised machine learning algorithms comparison

The five previously explored supervised machine learning algorithms all have one commonality: they heavily rely on the quality of the features extracted from the mammogram images as input to gain performance, and they cannot make use of the raw mammogram image in 2D space as input. This is confirmed by the datasets used for the task of BCD, as only datasets containing extracted features such as WBCD were used.

On their own, each algorithm showed limitations that prevent it from performing well. However, combined to form hybrid systems (e.g. DT + SVM + NB or ANN + AR), their performance increased in the form of higher accuracies or shorter training times, but so did their complexity to tune. It is worth noting that the slight differences when using identical algorithms on the same datasets can be due to the diverse training strategies involving unique

training/testing/validation splits, image pre-processing steps and number of folds in cross-validation (Yue et al., 2018).

The next step for these supervised learning algorithms is to move away from feature extraction and redirect the effort towards new models that can automatically extract features from images rather than optimising and fine-tuning the hyperparameters and training strategies of existing algorithms. The most efficient way nowadays to achieve this is through Convolutional Neural Networks (CNN), which ingest the raw mammogram images in 2D space rather than extracted features or flattened images (transformed from 2D to 1D) where all spatial information is lost.

## 2.4 CNNs & Deep Learning techniques

### 2.4.1 Convolution Neural Networks

#### Motivation for CNNs over traditional neural networks

CNNs are a type of neural network inspired by the human visual cortex, where neurons have local receptive fields that only react to visual stimuli originating from a region of the visual field. The combination of all receptive fields covered by overlapping neurons forms the whole visual field (Li and Allinson, 2008). This architecture makes them very efficient at performing complex visual tasks, marked by the first milestone for CNNs with the LeNet-5 architecture trained to recognise handwritten bank cheque digits (LeCun et al., 1998).

CNNs differ from traditional "shallow" ANNs as they are not fully connected. Indeed, CNNs are partially connected, with neurons in one layer only connected to a few neurons from the previous layer, meaning that they can work with large images (Géron, 2019). CNNs nowadays can work with images thousands of pixel large, including the ones found in mammogram datasets described in Chapter 3, processing them much faster than traditional machine learning methods. In a fully-connected neural network with only 100 neurons in the first layer, a 1,000 x 1,000-pixel image would already have 100,000,000 connections[1] in that first layer alone.

#### CNN structure

The structure of CNNs builds on top of the concepts of traditional ANNs by piling stacks of convolutional layers and pooling layers that are followed by a shallow ANN for classification (see Figure 2.3). The goal of the convolutional and pooling layers is to reduce the input images into a form that is simple

[1] $1000 \cdot 1,000 = 1,000,000$ px; $1,000,000$px $\cdot 100$ neurons $= 100,000,000$ connections.

enough to be processed by the fully connected layers, retaining only useful information from the original image (Shen et al., 2017).

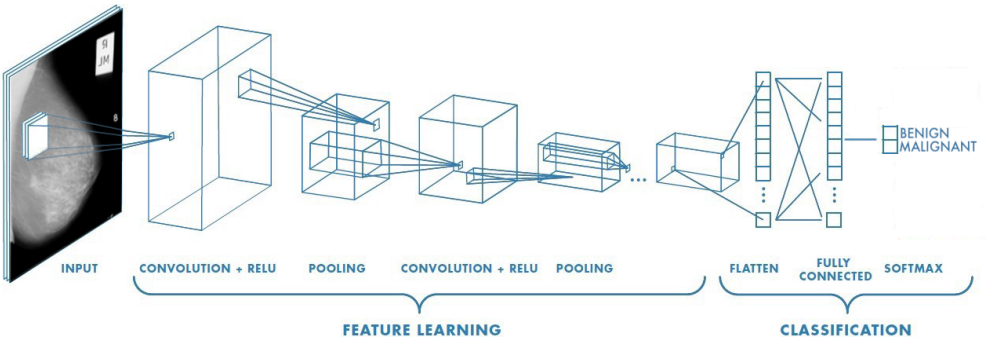



Figure 2.3: Example of a typical CNN adapted for multi-class breast cancer detection. Figure adapted from S. Saha (`https://tinyurl.com/y9mmosuq`).

**Convolution Layers** Neurons in the first convolutional layers are only connected to pixels in their receptive fields and not connected to every pixel in the image. Similarly, neurons in deeper convolutional layers are only connected to neurons in a small zone from the previous layer. This allows CNNs to first focus on low-level features, which are progressively assembled into high-level features as they get deeper. The more the receptive fields are spaced out (referred to as the stride), the smaller the next layer will be, thus considerably reducing the complexity of the CNN (Shen et al., 2017). Convolution is the mathematical operation that slides a moving a filter $f$ over an image $I$ to calculate a weighted sum (see Equation 2.1, Szeliski (2010)). This 2D operation is possible in CNNs as layers are represented in 2D space and do not need to be flattened into a 1D array like with traditional neural networks, thus preserving the spatial information of images (Szeliski, 2010).

$$\hat{I}(x,y) = (I * f)(x,y) = \sum_k \sum_l I(k,l) \cdot f(x-k, y-l) \tag{2.1}$$

The weights of the neurons in convolutional layers correspond to the filters, which are learned during the training phase by using optimisation techniques such as gradient descent. These filters allow a layer of neurons to highlight the areas of the image that activate the filter the most. As each layer has multiple filters, different features can be simultaneously detected in the layer's input. The result of each filter on a convolutional layer's input (the output of the previous layer) corresponds to a feature map. These multiple feature maps are all stacked together to form the convolutional layer's output.

**Pooling Layers** Pooling layers are similar to convolutional layers as neurons are only connected to neurons from a small region in the previous layer.

The difference is that this layer is not trainable as its neurons have no weights. Indeed, it is only used to downsample the image as it traverses the network to diminish the load on the GPU. It does so by calculating an aggregate of its inputs based on a function, which can either be a maximum or an average function (see Figure 2.4). *Maximum pooling* returns the maximum value of the covered portion of the image, whereas *average pooling* returns the average of all values. Maximum pooling is the preferred option in CNNs as it retains the most robust features only and acts as a noise suppressant by discarding noisy activations (Krizhevsky et al., 2012).

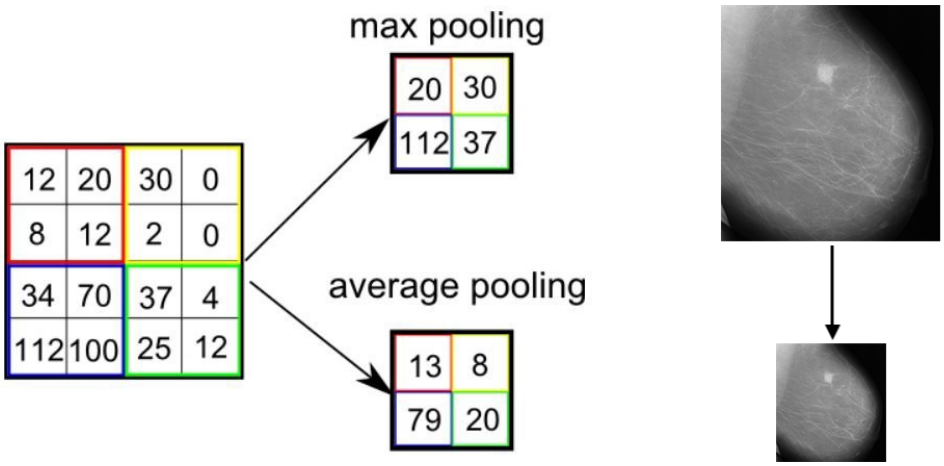



Figure 2.4: Difference between max pooling and average pooling using a 2x2 window and stride 2 (left) to downsample an image (right). Figure adapted from W. Ong (`https://tinyurl.com/y25cke6l`).

Other benefits of pooling layers are the lower memory usage and the number of trainable parameters linked to smaller images, and especially the invariance it provides to the CNN as it will not break down when fed images that contain features of different sizes than the ones seen in the training dataset (Shen et al., 2017).

**Fully connected layers & Activation functions** Similarly to ANNs, activation functions are used to connect the convolutional and pooling layers. The most common activation function used in CNNs is the Rectified Linear Unit (ReLU).

At the end of the stack of convolutional and pooling layers, a fully connected MLP is placed. This dense neural network takes the flattened output of the stacked convolutional and pooling layers (which are transformed from 2D to 1D) and performs the classification tasks by using the features learned by

the convolutional layers in a condensed format. Depending on the number of classes to predict, a softmax activation can be used for multi-class classification or a sigmoid for binary classification.

### 2.4.2 Regularisation techniques

The drawback of the power showcased by deep neural networks such as CNNs is their tendency to overfit the data. To this end, new regularisation techniques were introduced to prevent overfitting.

**Data augmentation** Another technique to counter small datasets is data augmentation, often used when attempting to learn a small dataset using complex deep learning models that have a large number of parameters, which may naturally lead to overfitting (Jadoon et al., 2017). The data is "augmented" by artificially creating similar realistic variants of the images found in the training set, considerably increasing the training set size. A varying amount of transformations can be applied to each image such as translation, rotation, scaling, shear, horizontal/vertical flips, brightness and contrast increases. Chen et al. show how applying data augmentation using affine transformations increases the accuracy from 83.6% to 88.14% using a ResNet-50 model (Chen et al., 2019).

#### Dropout

Simply implementing dropout, the most popular regularisation technique for deep neural networks, in any CNN has been proven to boost the accuracy by 1 to 2% at the cost of training time (Géron, 2019), even for state-of-the-art neural networks tested on large datasets like ImageNet (Srivastava et al., 2014). Despite training times being increased by 2 to 3 times, the gains in accuracy compensate for the extra time required to train the models implementing dropout.

Dropout works by randomly ignoring neurons in any layer (except the output layer) during the forward and backward passes of training, including its input and output connection weights, which helps prevents neurons from co-adapting too much with their neighbours and overfitting the data as they now have to be useful on their own. Essentially, new thinner networks are created at each training step (see Figure 2.5), and identical networks will never be sampled in the same training phases as there are $2^N$ possible networks, where $N$ is the number of dropable neurons. The number of neurons dropped in a layer is controlled by the dropout rate hyperparameter $p$, dictating the probability of a neuron being dropped. During testing, the neurons are no longer dropped, and an averaged ensemble of all the thinner trained networks

is used. This leads to models that generalise better thanks to neurons that are less sensitive to noise and small changes in the input (Srivastava et al., 2014).

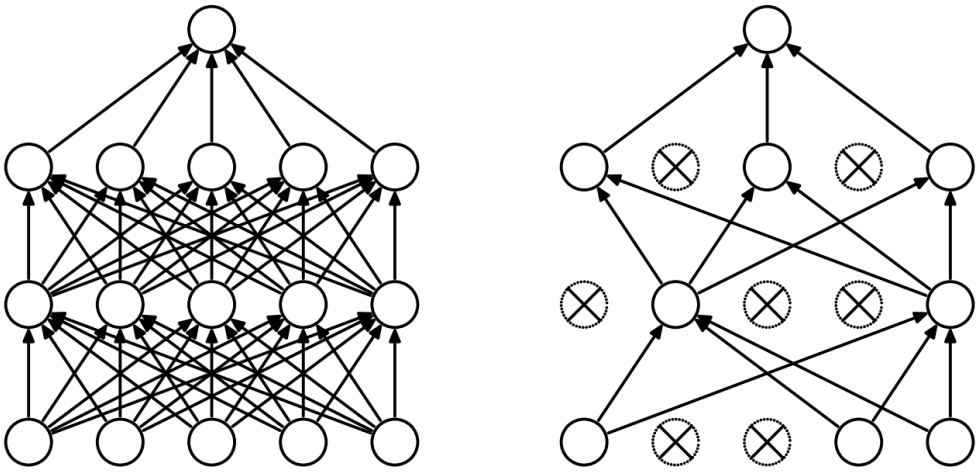


Figure 2.5: Example of standard neural network (left) and a neural network with dropout applied (right). Figure retrieved from Srivastava et al. (2014).

## 2.5 Transfer Learning in Breast Cancer Detection

### 2.5.1 Main challenges

Implementing deep learning models requires lots of data to achieve acceptable performance levels. However, labelled datasets are not always abundantly available as they require large amounts of time and computing resources to engineer large datasets with millions of images (Krizhevsky et al., 2012) like ImageNet, which contains over 15 million high definition images from 15,000 different classes (Deng et al., 2010). With databases of mammograms barely exceeding 10,000 images, one of the challenges of implementing a deep learning CAD system is to gain access to the computing resources needed to process the data and implement the model, as well as overcoming the problem of small amounts of data while avoiding overfitting it. Overfitting occurs when the model learns the data too well (e.g. detail and noise) and does not generalise well to new cases as it only recognises cases it has seen (Dietterich, 1995).

### 2.5.2 Transfer learning

A commonly used deep learning technique when only a little amount of data is available is transfer learning, which makes use of CNN models pre-trained on large general datasets. The knowledge gathered by high-performing CNNs in other general domains that have larger datasets can be transferred to a related

domain such as medical imagery (Falconi et al., 2019).

These models are designed to classify millions of images across thousands of different classes and can easily be adapted to any classification task by replacing the dense output layer that makes the actual prediction with a layer containing one neuron per class to predict. Falconi et al. demonstrated how transfer learning from general domain datasets such as ImageNet could be transferred to the domain of mammograms using datasets such as CBIS-DDSM, reaching accuracies of 78.4% with the ResNet-50 model (Falconi et al., 2019).

Shen et al. showed how using common CNN architectures such as VGG or ResNet pre-trained on ImageNet resulted in accuracy increases. For instance, the accuracy improves by 2-27% based on the number of patches used (Shen et al., 2017), while Diaz et al. demonstrated how two ResNet-50 models were tested with different weight initialisation: one on ImageNet weights and the other with random weights using the CBIS-DDSM dataset. The model using transfer learning achieved 84% accuracy compared to 75% accuracy (Diaz et al., 2018), clearly depicting the advantage of using CNNs with pre-trained weights.

### 2.5.3 CNN Architectures

Deep CNN models such as AlexNet, VGG and ResNet, which have won the *ImageNet Large Scale Visual Recognition Challenge*[2] in 2012, 2014 and 2015, remain very popular nowadays, especially in domains like medical imagery analysis where accuracies of 97.9% have been reached in BCD using VGG16 (Wang et al., 2016).

AlexNet is a CNN made up of five convolution layers that differs from traditional CNNs as not all convolution layers are separated by pooling layers (Krizhevsky et al., 2012). VGG architectures followed this concept by stacking multiple convolution layers with smaller filter sizes to pick up more complex features. Multiple variants of VGG exist based on the number of layers (VGG16 has sixteen layers and VGG19 has nineteen layers), but due to its depth, these takes a long time to train and run the risk of vanishing gradients (Simonyan and Zisserman, 2014). These are caused by the backpropagration algorithm coupled with gradient descent that lead to an exponentially smaller gradient while going back up the initial layers, which eventually prevents the network from learning as the weights and biases are no longer updated (Russell and Norvig, 2002). This problem does not occur in shallow ANNs.

[2]ImageNet challenge is a competition used to measure the performance of CNNs: `http://image-net.org/`.

More complex architectures were created subsequently to avoid creating deeper networks, such as ResNet, which uses residual modules, GoogLeNet, which uses inception modules, and MobileNet, which aims at maximising accuracy with little computing resources available.

### VGG16 & VGG19

VGG( Visual Geometry Group) is one of the excellent model, proposed by K. Simonyan and A. Zisserman of Oxford University which was used to win ILSVR(Imagenet Large Scale Visual Recognition Challenge) competition in 2014. It is best known for its architecture which has enabled it to achieve 92.7% accuracy in Imagenet (dataset of over 14 million images belonging to 1000 classes). VGG16, set as input to an RGB image of (224X224) size. As its name showed, It contains 16 layers including 13 convolution layers and 3 fully-connected. Each convolutional layer uses the filters with a very small receptive field (3x3), and after each convolutional layer, there will always be a ReLU correction layer, which can be regarded as a filter, allowing only positive values to be passed into the subsequent layers. The max-pooling operation follows some of the convolution, is performed over a (2x2) pixel window, with 2 stride. On the other hand, we have three fully connected layers: the first two layers each have 4096 nodes, and the third performs 1000-way ILSVRC classification, so it contains 1000 channels. The last layer is the Softmax layer. The main difference between VGG16 and VGG19 is the number of layers, the VGG19 has 19 layers but they have exactly the same in the last three fully connected layers.

### DenseNet201

DenseNet201 is a densely connected neural network. the objective of this type of architecture is to improve performance by concatenating all the entries from the previous layers because as the network performance improves, updating the weights during the training phase becomes tedious. The main components of DenseNet201 are: Density blocks, which are used to define the linking method of input and output. Transition layers, control the number of channels to reduce dimensionality. The advantage of this type of neural network is computational efficiency, by managing the number of parameters, this architecture guarantees the diversity of functions compared to traditional convolutive networks, and its low complexity, which allows the best performance.

### MobileNet_V2

MobileNet is part of the family of neural networks that specialize in computer vision for embedded systems, the convolution of which is replaced by a Depthwise Separable convolution with lesser number pf parameters. A depthwise separable convolution consists of 2 steps. At first, depthwise convolution applies on each filter of every input channels.Secondly, in the pointwise convolution uses $1 \times 1$ Kernel to increase the number of channels of each image to produce a linear combination of the product of the depthwise convolutional layer.

### Inception_V3

Inception_V3 is a deep convolutional network architecture proposed by Google. It consists of the inception modules which apply 1*1 convolution to truncate the input into a smaller intermediate block, then the 3*3 convolution to significantly reduce the computation cost and It uses RELU operation in each module.

### Xception

The Xception architecture, like Inception-V3, uses a Depthwise Separable convolution, is also known as "extreme" version of inception module. The two main differences between Inception-V3 and Xception are The difference in the order of convolution operations and existence of RELU. Indeed, Xception is the opposite, It uses 3*3 convolution of space, then 1*1 convolution of channel, and for the RELU function, Xception does not have it in each module.

### ResNet50

ResNet50 (residual network) is a typical neural network model, this ResNet50 model won the ImageNet challenge in 2015. It requires an input image size of 224 * 224 pixels and 50 layers. The architecture of ResNet50 is based on VGG architecture, it has many convolution layers mostly have 3*3 filters attached one by one and follows two rules: first, the layers for the same output feature map have the same number of filters. Secondly, if the size of the features map is halved, the number of filters is doubled to preserve the time complexity of each layer.

### Inception_ResNet_V2

Inception_ResNet_V2 is a CNN model that contains 164 layers. The input to the network has to be an impage of size 299x299. It is combines the two architectures (Inception and ResNet50) to further increase performance.

### 2.5.4 End-to-End model (E2E)

End-to-end (E2E) learning refers to a complex learning system represented by a single model (precisely a Deep Neural Network) that serves as the complete target system, taking advantage of Deep Neural Network's (DNNs) structure, formed from several layers, to solve complex problems. Similar to the human brain, each DNN layer (or bunch of layers) can specialize to perform intermediate tasks necessary for such problems. E2E can be done by applying gradient-based learning to the system as a whole.

The efficient of end-to-end learning has been demonstrated on many problems, such as speech recogniton (**?**) or Self-Driving Cars (Bojarski et al., 2016). However, Glasmachers pointed out potential inefficiencies of E2E learning, and how does not make optimal use of the modular design of present neural networks (Glasmachers, 2017).

## 2.6 Summary

CAD systems have been developed since the 1970s to assist radiologists in their interpretations of mammograms, starting with primitive expert systems. These systems were replaced by supervised machine learning algorithms but required hand-crafted features to be extracted from the images. The performance of the algorithms heavily relied on the quality of the features, which could not operate on raw mammograms. Therefore, deep learning algorithms have gained traction recently, notably CNNs, to automatically learn which features to extract from the images, thus preserving their 2D spatial property. However, CNNs require large amounts of data to avoid overfitting, which is not always available, especially in medical imagery.

# Chapter 3

# Datasets & Pre-processing

## 3.1 Introduction

In this section we will discuss the dataset used in this project, the pre-processing tasks that have been done over the original dataset as well as the additional pre-processing applied.

## 3.2 Datasets Description

This dataset is made of images from the DDSM and CBIS-DDSM datasets. Some Pre-Processing tasks have been done such exctraction the region of interest (ROIs) and resized to 299x299. This dataset contains 55890 training mammograms, with 7825 (14%) positive and the rest 48064 (86%) are negative (Eric A. Scuccimarra, Kaggle, 2018).

### 3.2.1 DDSM

The "Digital Database for Screening Mammography" (DDSM) dataset is a dataset initially released in 2007 and available online from the University of Florida. It holds 2,620 scanned film mammographies of normal, benign and malignant cases, all stored in Lossless JPEG format (LJPEG), which is obsolete nowadays (Heath et al., 2001).

### 3.2.2 CBIS-DDSM

The "Curated Breast Imaging Subset of DDSM" (CBIS-DDSM) dataset (Lee et al., 2017) is available online from The Cancer Imaging Archive (Clark et al., 2013). The dataset contains a total of 10,239 images in Digital Imaging and Communications in Medicine format (DICOM) gathered from 1,566 patients across 6,775 studies (Lee et al., 2017). This dataset is an updated and standardised subset of the older DDSM dataset (Heath et al., 2001), containing

only abnormal cases with benign and malignant tumours (no normal cases).

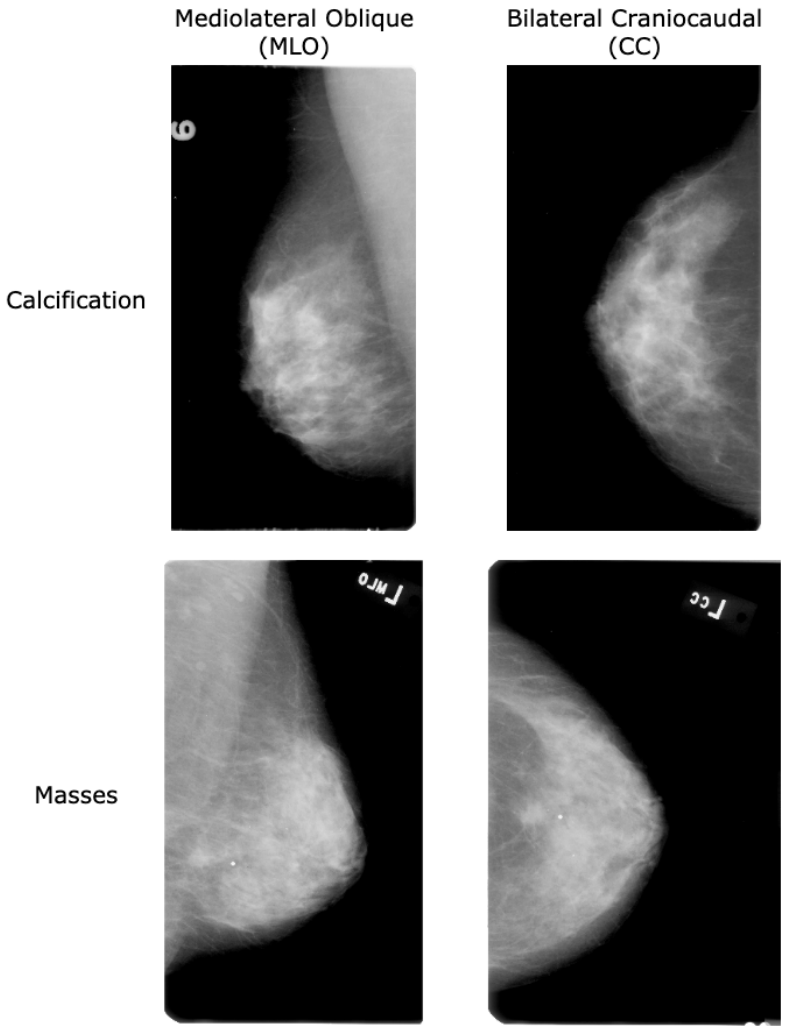



Figure 3.1: Types of structures and views captured by the CBIS-DDSM dataset (CC and MLO mammogram views are from the same patient).

The scans are a mix of the two most commonly used projections in routine mammogram X-ray scans: bilateral craniocaudal (CC) and mediolateral oblique (MLO) (Elter and Horsch, 2009). The dataset can be further be separated into two different types of structures that radiologists usually look for to detect early signs of breast cancer: calcifications (small flecks of calcium usually clustered together) and masses (e.g. cysts or lumps) ( *What Mammograms Show: Calcifications, Cysts, Fibroadenomas*, 2018). Figure 3.4 illustrates the different type of mammograms found in the dataset.

## 3.3 Pre-Processing tasks

The images composing the used dataset have been already pre-processed to remove the artifacts and the noise (Ahmed et al., 2020).

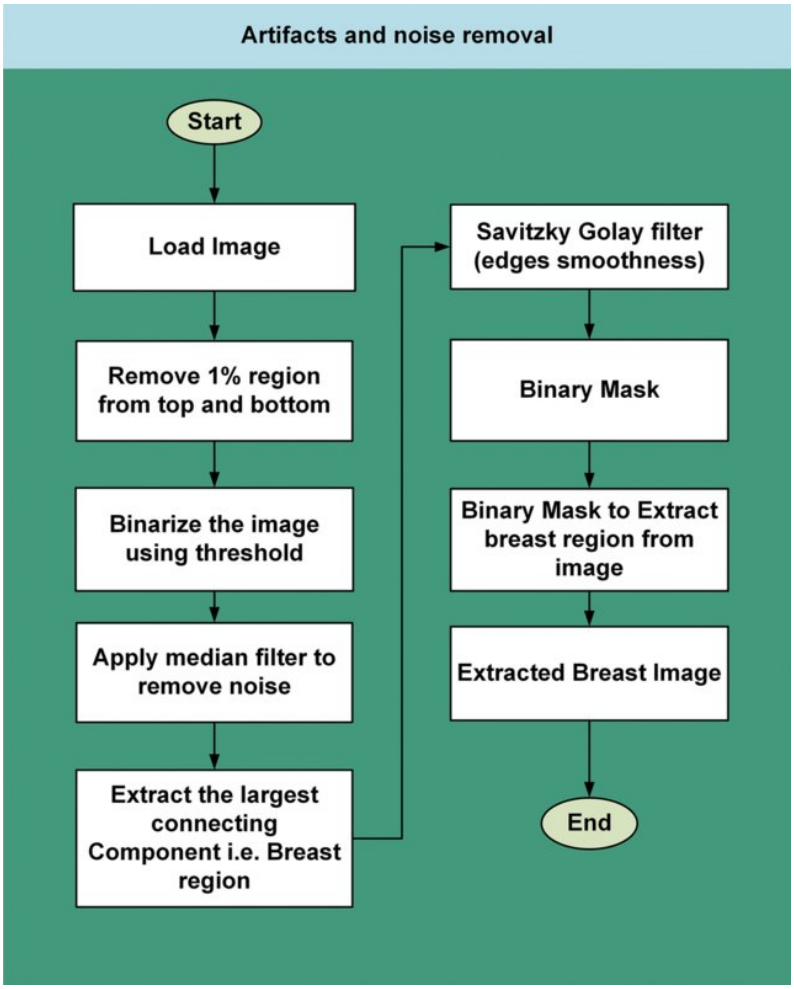



Figure 3.2: The different Pre-processing tasks applied over the DDSM and CBIS-DDSM.

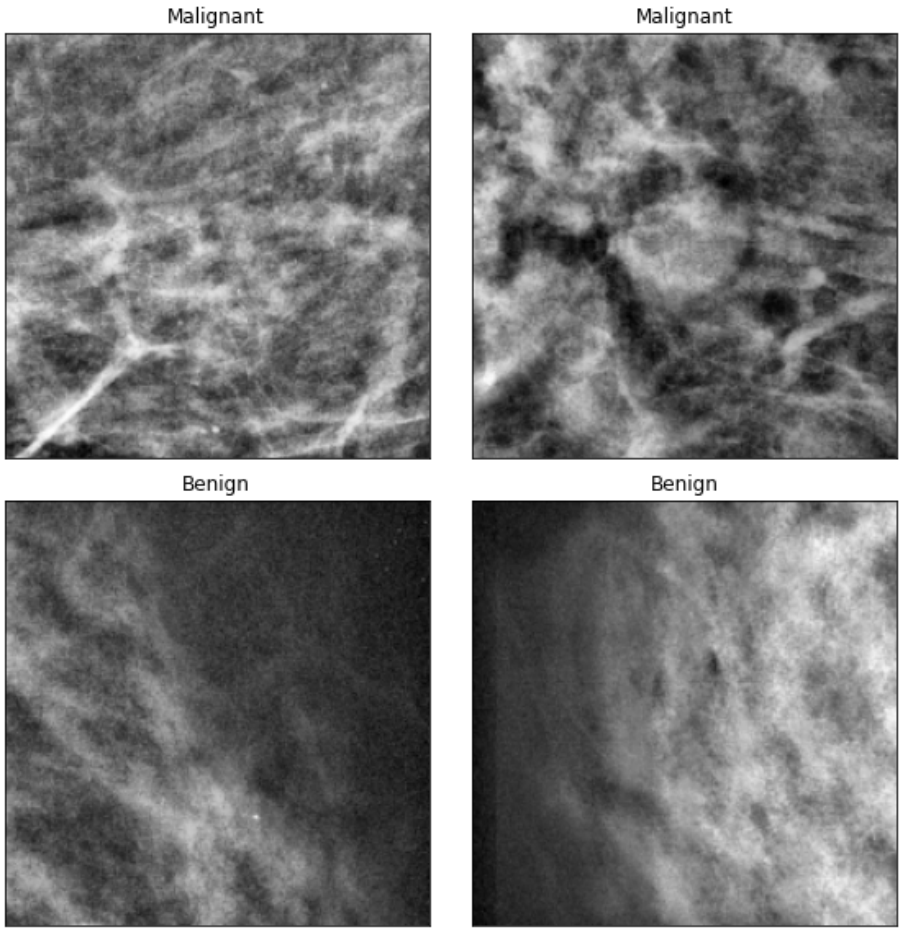



Figure 3.3: The extracted images at the end of the pre-processing tasks over DDSM and CBIS-DDSM.

## 3.4 Additional Pre-Processing tasks

Despite the multiple techniques applied over the original datasets to remove the noise to increase the performance of patterns recognition, other methods, such as contrast enhancing and intensity normalisation, are implemented to improve the images for higher efficiency.

### 3.4.1 Contrast Limited Adaptive Histogram Equalization

Histogram Equalization (HE) is a technique of image processing used to enhance images' contrast. It consists of stretching the range of intensity of the images, i.e. spreading out the frequent intensity values. This method generally increases the global contrast of images. This allows low local contrast areas to get a higher contrast.

Adaptive Histogram Equalization (AHE) alters from normal histogram equalization in the fact that the adaptive method computes many histograms, each corresponding to a different area of the image, and redistributes the lightness values of the image using them. Thus, it is useful for enhancing the local contrast and improving the appearance of edges in each section of an image (Pizer et al., 1987).

Contrast Limited Adaptive Histogram Equalization alters from AHE in limiting the contrast. In this method, the approach of limiting the contrast is applied to every neighborhood from where the transformation function is derived. CLAHE came to avoid the over amplification of noise that can be rise from AHE. (ZUIDERVELD, 1994)

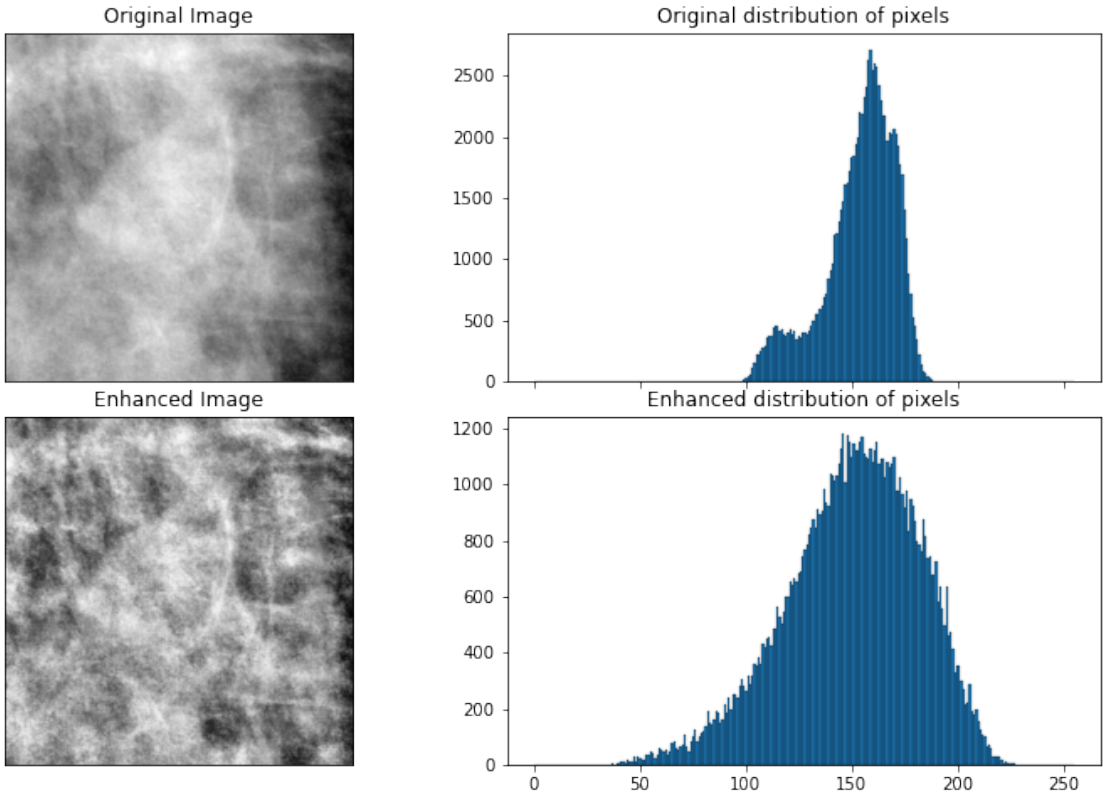



Figure 3.4: Enhancing the contrast using CLAHE technique.

### 3.4.2 Intensity Normalization

Intensity normalization, also known as image normalization, is a process in image processing that consist of bringing the range of pixel values to a fixed range, usually $[0, 1]$, without losing information or distorting the differences between values. It aims to convert the image in input into a range of pixel values that are more normal to the senses in general, and more adaptable to some algorithms such as KNN and Multi-layer Perceptron (MLP). The linear normalization applied to the pixel followed the formula :

$$I_N = (I - \text{Min})\frac{\text{newMax} - \text{newMin}}{\text{Max} - \text{Min}} + \text{newMin} \tag{3.1}$$

### 3.4.3 Images resizing

This task is not necessary. However, the main techniques used in this project rely on Pre-trained architectures, discussed in section 2.5.3, which are trained over enormous datasets such as ImageNet for image classification. Thus, it is highly recommended using the same shape of training images. Mainly, the architectures used in this project have either an RGB scale shape of (299,299,3) or (224,224,3). Therefore, our dataset with Gray-scale shape of (299,299) will be eihter or not resized then duplicated to get the recommended shape.

# Chapter 4

# Experimental Design

## 4.1 Introduction

Based on the machine learning and deep learning applications for the task of breast cancer detection established in Chapter 2, and the datasets available for this project, design decisions specific to the deep learning techniques implementation will be covered during this chapter, along with the reasoning behind the choices and general considerations.

## 4.2 Datasets

### 4.2.1 Dataset balance

As this is a classification task, the distribution of classes can widely affect the performance of the models as most algorithms are constructed to maximize the accuracy and minimize the errors. That leads to a high accuracy predicting the majority class and an inability to capture the minority class, which is mostly the purpose of building the model.

However, both the dataset and the approach adopted in this project prevent the side effect of the imbalanced data. As the dataset is fairly massive, under sample the majority class preserves the ability of the dataset to perform well on the classification task. In addition, the pre-trained architectures used in this task do not require massive dataset to perform properly (Marmanis et al., 2016).

### 4.2.2 Data split & KFold

The followed approach in this section is KFold split instead of the classical 80%/20% splitting as often used in machine learning and in breast cancer detection papers (Yue et al., 2018). Cross validation or KFold is favored over

the classical split because it allows the model to train on multiple train-test splits as well as a better indication on the performance of the algorithm over the unseen data.

Another reason for choosing this approach is the method that will be used to single out the best technique to classify breast cancer images. This method will be discussed afterward in chapter 5.

## 4.3 Models

At this stage, we are going to construct the end-to-end CNN models along with the specification of their parameters. End-to-End models refer to the model than can do the whole task itself, from extraction all levels of features as well as the task of classification for our case (see 2.5.4).

Due to the small nature of our dataset used, state-of-the-art CNN models pre-trained on large general datasets like ImageNet are used rather than creating a CNN from scratch, using what is called transfer learning (see Section 2.5.2) that is proven to work on breast cancer classification tasks (Shen et al., 2017; Falconi et al., 2019).

We are using eight popular CNN architectures available on Keras such as VGG19, ResNet50, InceptionV3, DenseNet201 and MobileNetV2 and others (Keras, 2020), beside a baseline model which will be constructed from scratch and fully trained over DDSM dataset. The baseline will be simple as possible and will be considered as reference to gauge the performance of the architectures.

### 4.3.1 Baseline model

The baseline is a simple model that provides a reasonable results on the classification task.

Our baseline model is composed of 3 convolution layers, each one is followed by a max pooling layer to reduce the dimentionality of the network. The fully connected layer is placed before the classification to flatten the results before the last layer where the decision is made.

### 4.3.2 CNN Architectures

The fully connected layers of the used architectures, originally designed for general classification of 1,000 different categories (Krizhevsky et al., 2012), are dropped from the model and replaced with a classification layer adapted

| Feature Extraction | | | Fully connected | Classification |
|---|---|---|---|---|
| **Convolution 1** | **Convolution 2** | **Convolution 3** | | |
| **# of filters: 32** | **# of filters: 64** | **# of filters: 128** | **Flattening** | |
| **Size of filters: 3x3** | **Size of filters: 3x3** | **Size of filters: 3x3** | **Dense layer: 512** | **Size: 2** |
| **Pooling: Max** | **Pooling: Max** | **Pooling: Max** | **Regularization: L2** | **Regularization: L2** |
| **Pooling size: 2x2** | **Pooling size: 2x2** | **Pooling size: 2x2** | **Dropout: 0.5** | |
| **Activation: ReLu** | **Activation: ReLu** | **Activation: ReLu** | **Activation: ReLu** | **SoftMax** |

Table 4.1: Baseline configurations

to our problem (see figure 4.1).

The configuration of the fully connected and classification layers is the same as the baseline (see table 4.1).

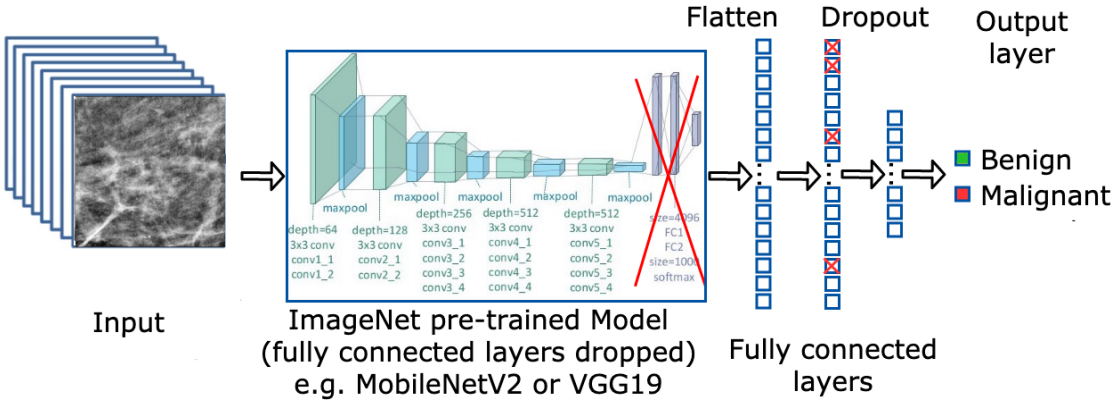



Figure 4.1: The implementation of the CNN architectures

## 4.4 Data fitting

### 4.4.1 Activation functions

Various activation functions can be used in the output layer of CNN. Commonly, a single neuron with the sigmoid function is adopted for binary classification as it gives an independent value in $[0, 1]$ that can be interpreted as probability of the positive class. Alternately, the softmax function transforms the output of the classification layer into probabilities for each class that sum

up to 1 (Litjens et al., 2017). These probabilities are dependant to each other, since each example have to belong to one and only one class, making it perfect for multi-class classification (each sample can only be either benign or malignant, there for to increase the probability of one class, it should decrease it for another).

However, the softmax function can be applied as well to binary classification with two neurons. Considering the future extension of this project, where the ensemble learning will be practiced with the weighted voting, the probabilities are essential. Thus, the softmax fits perfectly to our problem.

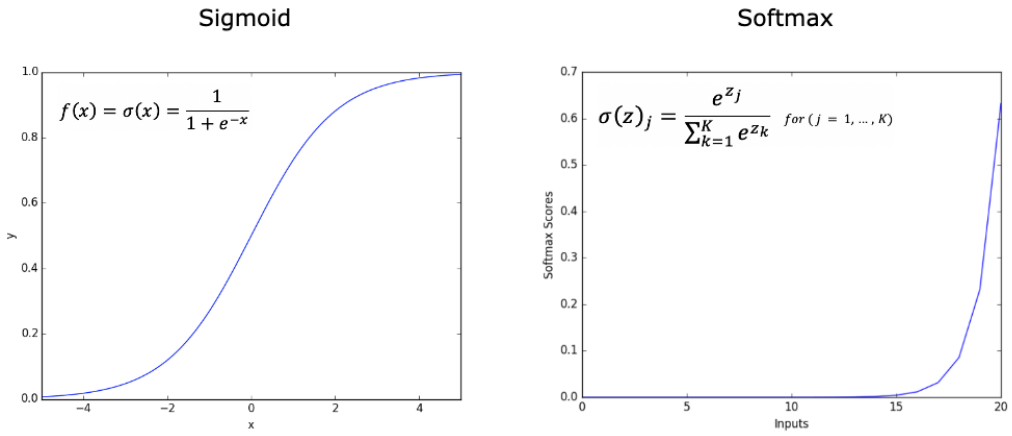



Figure 4.2: Visualisation of the sigmoid and softmax activation functions.

### 4.4.2 Loss function

Cross entropy is one of the most commonly used loss functions as it can be used for both binary and multi-class tasks (Litjens et al., 2017). Because probabilities are being estimated through the sigmoid and softmax activation functions, cross entropy is the ideal loss function as it heavily penalises the model when a low probability is predicted for the target class (Géron, 2019).

### 4.4.3 Optimiser

Due to the deep nature of the model, it is important to minimise the number of hyperparameters to control. Adaptive learning rate algorithms usually generalise better than traditional optimisers like SGD or momentum, which are slow to converge and require more fine-tuning.

The most general adaptive optimiser is *adaptive moment estimation* (Adam), which combines both momentum for more significant steps in the direction of

the steepest gradient and Root Mean Square Prop (RMSProp) for more accelerating on steep slopes than small slopes, making it the best choice for this model.

### 4.4.4 Varying Amounts of Transfer Learning

The architectures we are using in this project have different number of layers. Going backward unfreezing some layers may bias the comparison of the architectures as unfreezing two or three layers from an architecture of 200 layers is not the same as unfreezing two or three layers from a ten layers model. Thus, to better compare the performance of the architectures it should either re-train the whole layers (which is computationally expensive and does not take advantage of transfer learning) or freezing them all to take advantage of the knowledge gathered on ImageNet.

All the layers from the base architectures are frozen, enabling only the custom MLP with fully connected layers to fit the mammogram images. The initial training phase ends once the maximum number of epochs is reached, or the early stopping condition is met.

However, unfreezing more layers gives a better performance but it takes very long time to update the weights. The ImageNet weights transfer confirmed the performance that can be gained, as well the adaptive nature of CNNs when using knowledge learned from large general datasets for a more specific task like breast cancer detection.

## 4.5 Evaluation criteria

The following terminology is used to define the metrics used below:

- TP: True Positives (positive case correctly predicted as positive);
- TN: True Negatives (negative case correctly predicted as negative);
- FP: False Positives (negative case incorrectly predicted as positive);
- FN: False Negatives (positive case incorrectly predicted as negative).

### 4.5.1 Accuracy

Accuracy is what we often mean, when we use the term accuracy. It is the ratio of the samples well classified from the total number of samples from all classes. It is considered as the most famous metric to gauge the performance

of classifiers. However, it can misestimate the performance of the classifier if the dataset is not balanced.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

### 4.5.2 Precision & recall

Precision corresponds to the number of correct positive predictions (see Equation 4.2, Liu et al. (2009)), showing the model's ability to avoid labelling negative instances as positive.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Recall is the number of positive instances that are correctly predicted (see Equation 4.3, Liu et al. (2009)), showing how well the model can find all positive instances.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

### 4.5.3 F1 score

Together, precision and recall can be combined into a more concise metric, the *F1 score*, which corresponds to the harmonic mean of precision and recall (see Equation 4.4, Géron (2019)). To achieve a high F1 score, both precision and recall must be high (unlike a regular mean) because as the precision goes down, the recall goes up, and vice versa, making the F1 score a reliable metric for evaluating a classifier since a balance between precision and recall must be found (Géron, 2019).

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{TP}{TP + \frac{FN+FP}{2}} \tag{4.4}$$

### 4.5.4 Confusion matrix

This visual metric plots the number of predictions made for each class for each possible class in a table, with each row corresponding to the actual labels and each column corresponding to a prediction. It is beneficial for detecting which actual classes are being detected the most, and what predicted classes are being misclassified as (Bhardwaj and Tiwari, 2015; Liu et al., 2009). To further highlight the misclassifications and compare predictions with other classifiers, the confusion matrices are normalised to show a percentage rather than a count.

### 4.5.5 Statistical tests & voting systems

To aggregate the most efficient architecture we followed the process discussed thereafter in section 4.6 using a statistical test known as Scott Knott test as well as the single winner voting system called Borda count.

#### Scott Knott (SK)

The SK test is an analysis method used for group classifiers into statistically distinct rank. It was developed by Scott, A.J. and Knott, M. (Scott and Knott, 1974), used to remedy the overlapping problem posed by old methods, for example: the t-test. The SK is an hierarchical clustering algorithm used in the application of ANOVA, when the researcher is comparing treatment means, the Scott-Knott algorithm is based on the assumption that the distribution of errors is approximately normal, the absolute errors for all methods were transformed, for being used as input to the Scott-Knott test.

#### Borda count

The Borda count is a single-winner election method in which voters rank candidates in order pf preference. It elects the winner of an election by giving each candidate a certain number of points based on the position in which he is ranked by each voter. The winner is the candidate with the most sum of points. In the present study, we used Borda count technique to find the best performing DL technique.

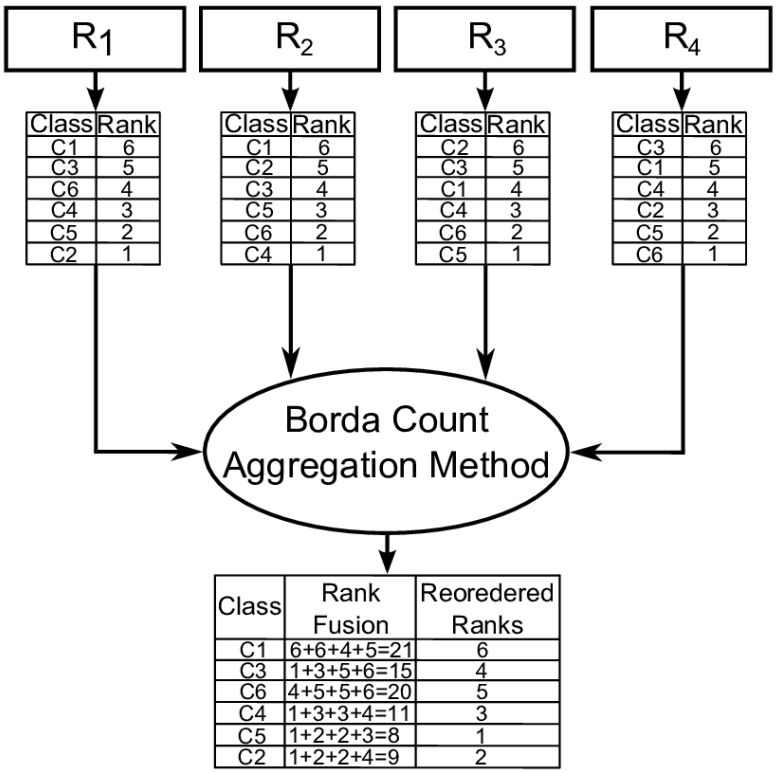



Figure 4.3: Borda count voting system. (Figure retrieved from (Sharif et al., 2015))

## 4.6 Experiment process

The experiment methodology followed in this project, to conduct the empirical results, made up of four steps as shown in

1. Consider the accuracy as the main performance metric and assess its value for all the CNN architectures (VGG16, VGG19, DenseNet201, MobileNet_V2, Xception, Inception_V2, ResNet50, Inception_ResNet_V2) as well as the baseline CNN model.

2. Select the deep learning architectures that outperforms the baseline model in term of accuracy.

3. Make clusters of Deep learning models using the Scott Knott test and select the models that belongs to the best cluster ( Best cluster means best accuracy. Two models of the same cluster are statistically indifferent).

4. Use the Borda count voting system based on the four performance metrics ( accuracy, precision , recall , f1-score) to rank the deep learning

models of the best Scott Knott cluster and select the winner deep learning architectures (i.e. top ranked architecture)

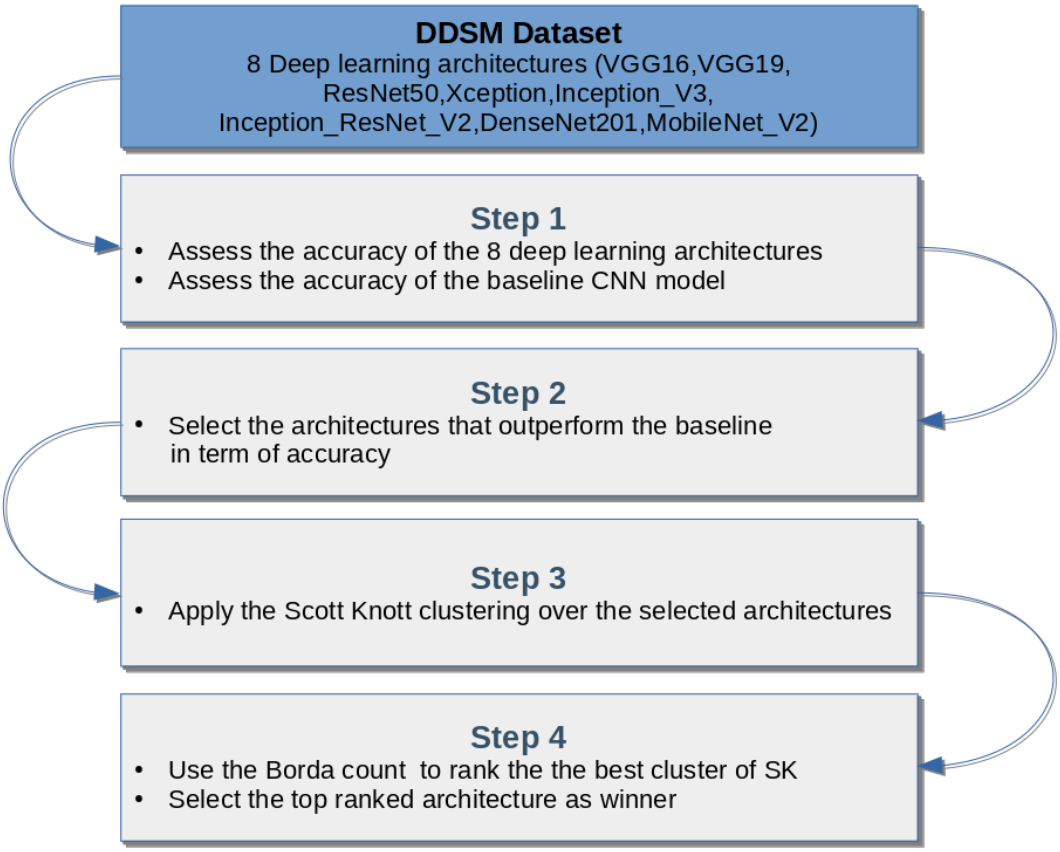



Figure 4.4: Experimental process.

## 4.7 General Design Decisions

### 4.7.1 Programming Language

Multiple languages, including Python, Java, R and Javascript, are considered for this project. Ultimately, Python is chosen for this project due to the familiarity and experience with the language; and the availability of open-source libraries for implementing common machine learning functionalities, as well as data pre-processing, manipulation and visualisation techniques found in deep learning systems to avoid manually implementing them (Raschka and Mirjalili, 2017), such as Tensorflow (Abadi et al., 2015), Keras (Chollet et al., 2015), SciKit-Learn (Pedregosa et al., 2011), Pandas (pandas development team, 2020), Matplotlib (Hunter, 2007), NumPy (Oliphant and developers, 2020) and Seaborn (Michael, 2020).

### 4.7.2 Tensor Processing Unit (TPU)

TPU is a Tensor Processing Unit, an Application Specific Integrated Circuit (ASIC) developed by Google to speed up operations on a Tensorflow graph, specifically for machine learning in neural networks for TensorFlow-based modeling to accelerate the training of models. TPU is designed to accelerate deep learning tasks develop using only TensorFlow (deep learning framework); but have not been developed for general purpose programming.

In term of performance, TPU speedup the training to 100 times much faster than CPU and 4 times the performance of GPU.

## 4.8 Summary

1. Model training:
   (a) models : Baseline CNN, VGG19, VGG16, ResNet50, Inception_V3, DenseNet201, MobileNet_V2, Inception_ResNet_V2 and Xception
   (b) Output layer activation function: softmax
   (c) Optimiser: Adam
   (d) Loss function: Cross entropy
   (e) Batch size : 32
   (f) Number of epochs : 200
   (g) initial Learning rate : 0.00001
   (h) Transfer learning : All layers are frozen
   (i) Generalisation to unseen data: validation loss and accuracy, early stopping and dropout
2. Evaluation Metrics:
   (a) Numerical metrics:
      - Accuracy
      - Precision & recall
      - F1 score
   (b) Visual metrics:
      - Confusion matrices
3. Experimental process:
   (a) Assess the accuracies of all the models
   (b) Choosing the models that outperform the Baseline

(c) Clustering the selected models using SK test

(d) Aggregate the most efficient CNN Architecture using Borda count based on the four metrics.

4. Programming language:

   (a) Python 3.9

   (b) Open-source frameworks:

      - Tensorflow & Keras
      - Scikit-Learn
      - NumPy, Pandas, MatplotLib & Seaborn

5. Processing unit : Cloud TPU service of the Google Cloud Platform

# Chapter 5

# Results & Discussion

## 5.1 Introduction

This section covers the results obtained through the experimentation of all the models mentioned in Section 4.3 to determine which model perform better over the mammograms dataset. Across each experiment, identical configurations are used to ensure that accurate comparisons can be made.

## 5.2 Training & Validation dataset

The final dataset contains the following number of samples:

- Total: 4000
  - Benign: 2000
  - Malignant: 2000

With the use of the cross-validation with a $K$=5, the training data will be 3200 samples with equal portion of each class. Thus, the test data (validation) will contain 800 samples also with equal portion of each class.

## 5.3 Model Used

The model described in Section 4.3 is used across all the experiments in this chapter. The following remain constant across the experiments:

- fully connected MLP with 512 hidden neurons and 2 output neurons;
- dropout layer using $p = 0.5$;
- Adam optimiser with an initial learning rate of 0.00001 for all the models that varies during the learning phase if the metric stopped improving for a 'waiting' number of epochs.

## 5.4 Baseline Results

As mentioned before, the baseline is as simple as possible and at same time gives acceptable results benchmark to compare the results obtained from the CNN pre-trained architectures. Contrarily to the other models, based on pre-trained architectures, the baseline is trained completely over the dataset, where the other models train only the classification layers keeping the weights of the other layers unchanged. The results achieved are shown is the table below.

| **Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|
| 82.07% | 80.98% | 84.25% | 82.5% |

Table 5.1: Results of the baseline CNN.

With only three convolutional layers and a small dataset, the baseline CNN gave satisfied results.

## 5.5 CNN Architectures results

Eight different CNN model architectures pre-trained on ImageNet (VGG19,VGG16, ResNet50, Inception_V3, DenseNet201, MobileNet_V2, Xception and Inception_ResNet_V2) are tested out over DDSM dataset.

The results found in Table 5.2 clearly reveal that DenseNet201 unlocks more performance than the other CNN architectures with a higher accuracy and F1 score. These results contradict Falconi's results on the CBIS-DDSM dataset, who finds that ResNet50 outperforms MobileNet (Falconi et al., 2019) while our ResNet50 was the weakest model showing some instabilities. However, MobileNet_V2 still outperforms Inception_V3. These results may differ due to the different pre-processing techniques being used, the size of dataset and the amount of variation of transfer learning.

In numbers, from 800 sample of test with equal portion of the two classes we can better recognize how many misclassification have been done by the most efficient model DenseNet201 in the confusion matrix below.

| CNN Architecture | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| ***VGG19*** | 81.00% | 80.76% | 81.85% | 81.14% |
| ***VGG16*** | 82.82% | 81.48% | 85.19% | 83.26% |
| ***InceptionV3*** | 80.75% | 80.77% | 81.10% | 80.87% |
| ***DenseNet201*** | **84.27%** | **84.55%** | **84.00%** | **84.27%** |
| ***MobileNetV2*** | 83.42% | 83.47% | 83.55% | 83.48% |
| ***Xception*** | 82.57% | 82.64% | 82.60% | 82.61% |
| ***InceptionResNetV2*** | 83.12% | 83.34% | 82.95% | 83.14% |
| ***ResNet50*** | 76.35% | 76.27% | 76.92% | 76.46% |

Table 5.2: Results achieved on the pre-trained architectures.

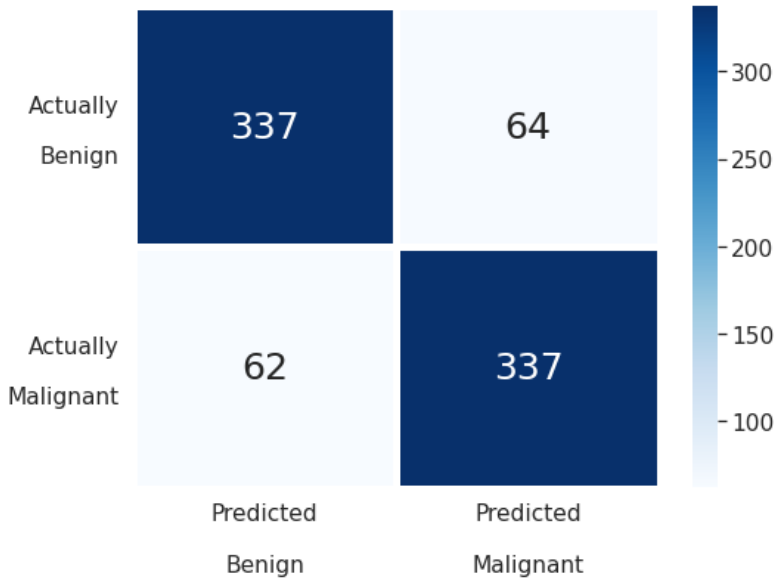



Figure 5.1: Average confusion matrix of the DenseNet201

## 5.6 Graphical Comparison

To better compare graphically the behavior of the models in one shot, the following figures show the evolution of loss and accuracy functions for both training and validation.

### 5.6.1 Loss function

The plots (5.2 & 5.3) of the history of the cross-entropy loss function over the epochs show that the models have converged and has reasonable loss. However, some models converged much faster than others (e.g. Baseline and resNet50).

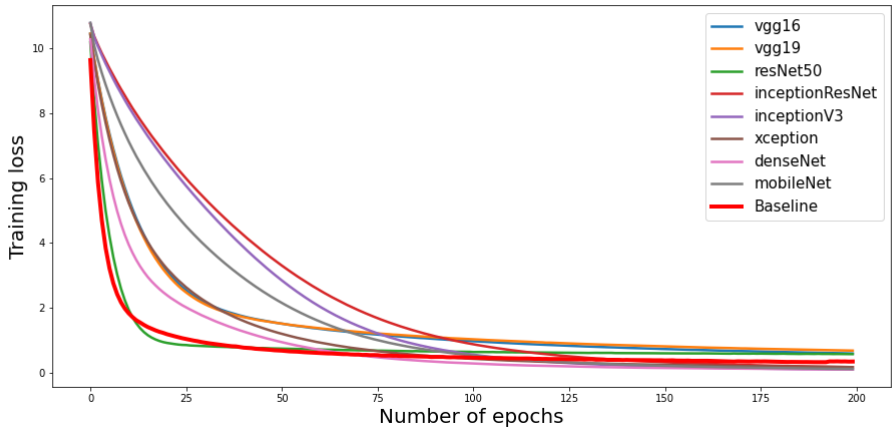



Figure 5.2: Training loss.

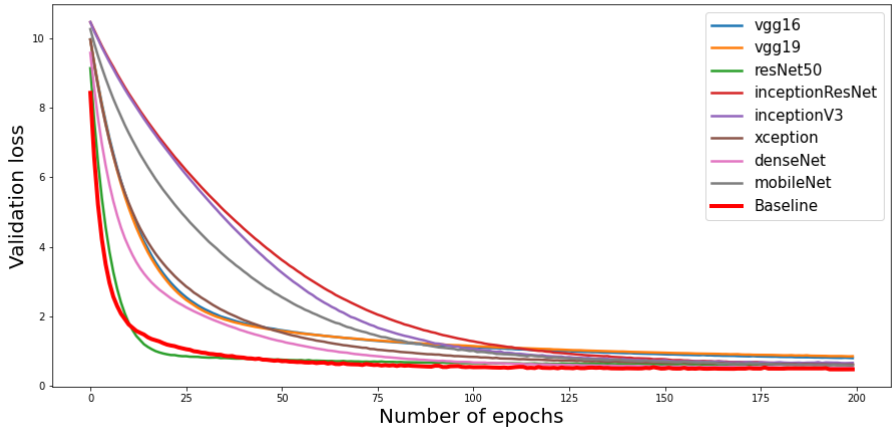



Figure 5.3: Validation loss.

The overall convergence number of epochs is 125, which can reduce the computational cost if the number of epochs have been reduced, Depending to the sensitivity of the model to small changes in weights.

### 5.6.2 Accuracy

The plots (5.4 & 5.5) show that the training process converged well. Some architectures (e.g. DenseNet, MobileNet, Xception and Inception) converged much quicker, thing than can be explained by the great generalization of these models and how the accuracy is less sensitive to the small changes of weights. Thus the number of epochs can be reduced without affecting the performance of the model. However, the curve of the other model keep increasing as they require more training to get better performance.

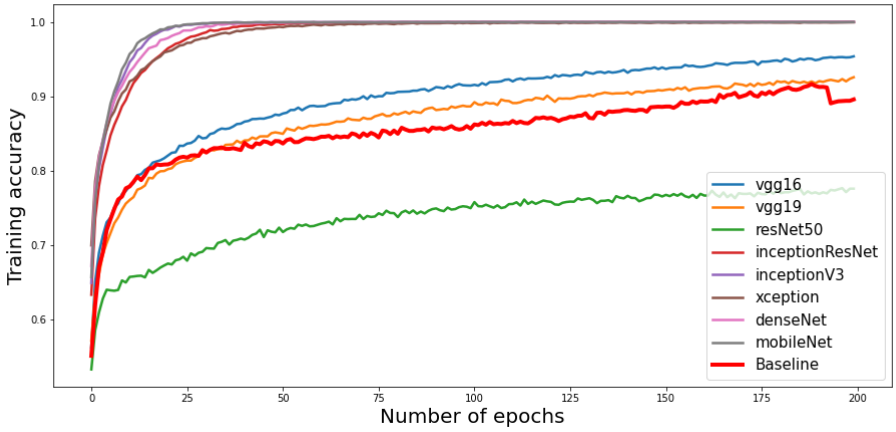



Figure 5.4: Training accuracy.

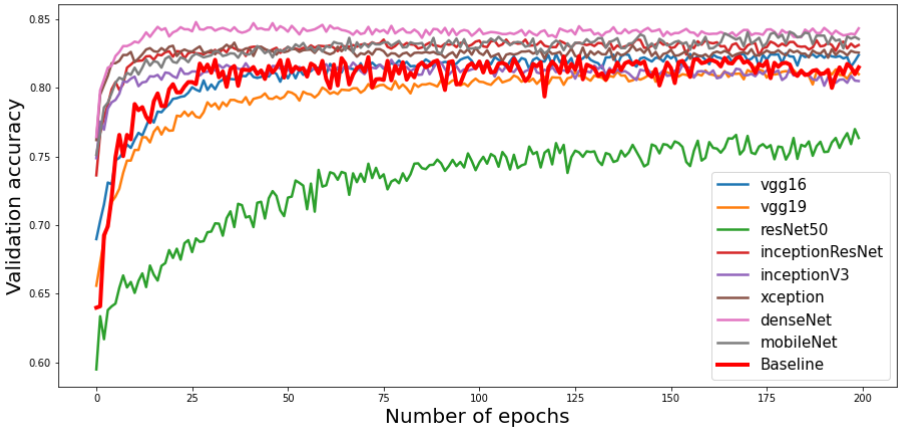



Figure 5.5: Validation accuracy.

## 5.7 Skott-Knott test

Based on the results obtained in section 5.5 to better select the best group of architectures, we applied the hierarchical clustering Scott Knott. Yet, the clustering gave only one cluster showing that all the architectures that outperform the baseline are statistically indifferent based on the accuracy.
For visual reasons, we applied the the clustering over the whole architectures as well as the baseline CNN.

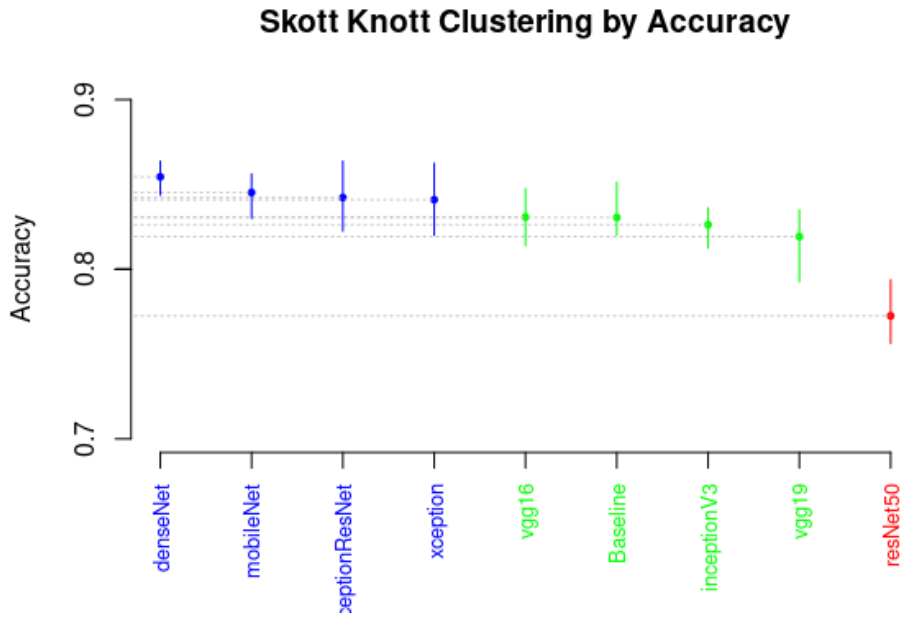



Figure 5.6: Scott-Knott Test.

The outcome is three clusters as is shown in the figure 5.6. The best cluster (in blue) is made of the architectures than outperform the baseline CNN, the second cluster (in green)ù composed of the models that showed similar performance to the baseline and finally ResNet50 as one-element cluster with the weakest efficiency.

## 5.8 Borda count & best model

Selecting the best model from the best cluster have been done by the borda count voting system taking in consideration the four metrics (accuracy , recall , precision and f1-score ).

| CNN Architecture | Score |
|---|---|
| ***DenseNet201*** | **16** |
| ***MobileNet_V2*** | 12 |
| ***Inception_ResNet_V2*** | 8 |
| ***Xception*** | 4 |

Table 5.3: Borda count scores and ranking.

The voting reveal the DenseNet201 as the winner and thus the most efficient architecture.

## 5.9 Results Summary

DenseNet201 technique gave the best results for DDSM classification since it belonged to the best clusters and was elected by the other metrics through the borda count.

# Chapter 6

# Conclusions

## 6.1 Achievements

The main goal of this project was to design, implement and compare various deep learning techniques capable of detecting cases of breast cancer in mammograms. An experimental design had been conducted to carry out the best CNN model.

The most positive result (84.27%) on DDSM came from transfer learning techniques, using ImageNet weights with DensNet201. However, other techniques did behave as expected and resulted in pleasant accuracies.

## 6.2 Limitations

Several limitations concerning this project lie mainly on the computational resources, since the dataset used were quite larger than the sample used to train the CNN models due to limitation imposed by google cloud TPU serive (The session cracks for unknown reason if the size of the dataset passed the 4000 images).

powerful resources may allow us to use more images as well as the data augmentation technique that allows the models to generalize much better.

Another known limitation concerning all breast cancer detection systems lies with the data. Indeed, the most widely used datasets of mammograms (e.g. DDSM) contain mammography data that mainly originates from white females located in North America (see Table 6.1), which naturally introduces bias to the model learning this data (Yala et al., 2019).

Different body types linked to the geographic location of the patients used to create these databases can have a direct impact on the mammograms themselves and not generalise to females from other cultures. For example, a recent study with 53,000 North American females showed how diets that include dairy

| | Data source | |
|---|---|---|
| **Race** | **MGH** | **WFUSM** |
| *Asian* | 2.06% | 0.20% |
| *Black* | 4.12% | 20.40% |
| *Spanish Surname* | 6.55% | 1.80% |
| *American Indian* | 0.00% | 0.10% |
| *Other* | 0.75% | 0.10% |
| *Unknown* | 30.34% | 0.30% |
| *White* | **56.18%** | **77.00%** |

Table 6.1: DDSM dataset patient population statistics (female). Data collected by Massachusetts General Hospital (MGH) and Wake Forest University School of Medicine (WFUSM) (Heath et al., 2001).

milk consumption might increase the risk of breast cancer by a maximum of 80% based on the consumption (Fraser et al., 2020). This means that if these deep learning algorithms were implemented in clinics outside western countries, they might not generalise well to other body morphologies (e.g. due to different diets based on the geolocation's culture). This limitation could be resolved by collecting more varied data from multiple locations around the world, not just a single region, which would also help deep learning algorithms as more data is always welcomed.

Another limitation in terms of the detection system's usability is the confidence of the predictions. Indeed, when given new test samples, the model predicts a class label, e.g. benign or malignant. However, these do not indicate the prediction's confidence, as it can be anywhere between the decision boundary's limit (not confident) and far from the decision boundary (confident). Therefore, from a clinical point of view, it is hard to make a decision based on the predictions made by a system similar to this one. Ideally, a probability-based confidence metric would be coupled with the predictions to motivate the next step after the screeting. For example, if the confidence of a malignant tumour is high (e.g. 99%), then breast-conserving surgery or chemotherapy can be recommended, whereas if the confidence is low (e.g. 54%), then further screening tests can be recommended instead.

Finally, the time frame of this project was a limiting factor in the final performance achieved as a fine-tuning method like grid search can try different

combinations of configurations and may increase the efficiency of the results obtained.

## 6.3 Future Work

This research will not stop in this stage, on going works have been done by us over a classical dataset (tabular) including many factors such as age, density of breast, bmi, cancer type, radiologist assessement and other factors, to build a CAD system more efficient that gather severel systems.

An extention of this research is conducted by Phd. Asmaa ZIZAAN, that investigates homogenous and heterogeneous ensemble learning using deep learning techniques as weak learner to get a much powerful learner, with different meta-learning techniques such as bagging, boosting and stacking for breast cancer imaging classification.

Another area of work that requires improvements is the mammogram preprocessing as it is often an area where significant performance gains can be found (Litjens et al., 2017). Artefacts such as tags on the x-rays and black backgrounds should all be removed using computer vision techniques to avoid having the CNN learn irrelevant features.

Also improvements can be made is the fine-tuning to extract better performance on the datasets and avoid overfitting. With the data-preprocessing mentioned above, images would be smaller (e.g. no redundant dark background), which would allow for quicker runtimes, which would allow fine-tuning algorithms like grid search to explore more combinations of configurations in order to unlock better solutions.

## 6.4 Reflections

This project was an exciting challenge from our point of view as it encompassed all the classical challenges that need to be faced when building deep learning algorithms, clearly showing that creating a solution with high performance is not as easy as it sounds.

# Bibliography


Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015), 'TensorFlow: Large-scale machine learning on heterogeneous systems'. Software available from tensorflow.org.
**URL:** *https://www.tensorflow.org/*

Ahmed, L., Iqbal, M., Aldabbas, H. and Saeed, S. (2020), 'Images data practices for semantic segmentation of breast cancer using deep neural network', *Journal of Ambient Intelligence and Humanized Computing* .

American Cancer Society (2019), 'American Cancer Society screening recommendations for women at average breast cancer risk', `https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/american-cancer-society-recommendations-for-the-early-detection-of-breast-cancer.html`. [Online] Accessed: 2021-07-22.

Asri, H., Mousannif, H., Al Moatassime, H. and Noel, T. (2016), Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis, *in* 'Procedia Computer Science', Vol. 83, Elsevier, pp. 1064–1069.

Bergstra, J., Yamins, D. and Cox, D. (2013), 'Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms', *Proceedings of the 12th Python in Science Conference* (Scipy), 13–19.

Bhardwaj, A. and Tiwari, A. (2015), 'Breast cancer diagnosis using Genetically Optimized Neural Network model', *Expert Systems with Applications* **42**(10), 4611–4620.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P.,

Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J. and Zieba, K. (2016), 'End to end learning for self-driving cars'.

Caruana, R. and Niculescu-Mizil, A. (2006), An empirical comparison of supervised learning algorithms, *in* 'ACM International Conference Proceeding Series', Vol. 148, ACM Press, New York, New York, USA, pp. 161–168.
**URL:** *http://portal.acm.org/citation.cfm?doid=1143844.1143865*

Chen, Y., Zhang, Q., Wu, Y., Liu, B., Wang, M. and Lin, Y. (2019), Fine-tuning ResNet for breast cancer classification from mammography, *in* 'Lecture Notes in Electrical Engineering', Vol. 536, Springer Verlag, pp. 83–96.
**URL:** *https://doi.org/10.1007/978-981-13-6837-0{_}7*

Chollet, F. et al. (2015), 'Keras', `https://github.com/fchollet/keras`.

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., Tarbox, L. and Prior, F. (2013), 'The cancer imaging archive (TCIA): Maintaining and operating a public information repository', *Journal of Digital Imaging* **26**(6), 1045–1057.

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li and Li Fei-Fei (2010), ImageNet: A large-scale hierarchical image database, *in* 'Institute of Electrical and Electronics Engineers (IEEE)', pp. 248–255.

Diaz, O., Marti, R., Llado, X. and Agarwal, R. (2018), Mass detection in mammograms using pre-trained deep learning models, *in* E. A. Krupinski, ed., '14th International Workshop on Breast Imaging (IWBI 2018)', Vol. 10718, SPIE, p. 12.
**URL:** *https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10718/2317681/Mass-detection-in-mammograms-using-pre-trained-deep-learning-models/10.1117/12.2317681.full*

Dietterich, T. (1995), 'Overfitting and undercomputing in machine learning', *ACM computing surveys (CSUR)* **27**(3), 326–327.

Elter, M. and Horsch, A. (2009), 'CADx of mammographic masses and clustered microcalcifications: A review', *Medical Physics* **36**(6), 2052–2068.

Eric A. Scuccimarra, Kaggle (2018), 'DDSM Mammography', `https://www.kaggle.com/skooch/ddsm-mammography`. [Online] Accessed: 2021-06-09.

Falconi, L. G., Perez, M. and Aguilar, W. G. (2019), 'Transfer Learning in Breast Mammogram Abnormalities Classification with Mobilenet and Nasnet', *International Conference on Systems, Signals, and Image Processing* **2019-June**, 109–114.

Fraser, G. E., Jaceldo-Siegl, K., Orlich, M., Mashchak, A., Sirirat, R. and Knutsen, S. (2020), 'Dairy, soy, and risk of breast cancer: those confounded milks', *International Journal of Epidemiology* .
**URL:** *https://academic.oup.com/ije/advance-article/doi/10.1093/ije/dyaa007/5743492*

Géron, A. (2019), *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, 2nd edn, O'Reilly Media.

Glasmachers, T. (2017), 'Limits of end-to-end learning'.

Heath, M., Bowyer, K., Kopans, D., Moore, R. and Kegelmeyer, W. P. (2001), The Digital Database for Screening Mammography, *in* 'Fifth International Workshop on Digital Mammography', Medical Physics Publishing, pp. 212–218.

Hepsağ, P. U., Özel, S. A. and Yazici, A. (2017), Using deep learning for mammography classification, *in* '2nd International Conference on Computer Science and Engineering, UBMK 2017', Institute of Electrical and Electronics Engineers Inc., pp. 418–423.

Hunter, J. D. (2007), 'Matplotlib: A 2d graphics environment', *Computing in Science & Engineering* **9**(3), 90–95.

Jadoon, M. M., Zhang, Q., Haq, I. U., Butt, S. and Jadoon, A. (2017), 'Three-Class Mammogram Classification Based on Descriptive CNN Features', *BioMed Research International* **2017**.

Keras (2020), 'Keras Applications', `https://keras.io/api/applications/`. [Online] Accessed: 2021-08-09.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), ImageNet Classification with Deep Convolutional Neural Networks, Technical report, Google.
**URL:** *http://code.google.com/p/cuda-convnet/*

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998), 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE* **86**(11), 2278–2323.

Lee, R. S., Gimenez, F., Hoogi, A., Miyake, K. K., Gorovoy, M. and Rubin, D. L. (2017), 'A curated mammography data set for use in computer-aided detection and diagnosis research', *Scientific Data* **4**.

Li, J. and Allinson, N. M. (2008), 'A comprehensive review of current local features for computer vision', *Neurocomputing* **71**(10-12), 1771–1787.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B. and Sánchez, C. I. (2017), 'A survey on deep learning in medical image analysis'.

Liu, X. Y., Wu, J. and Zhou, Z. H. (2009), 'Exploratory undersampling for class-imbalance learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **39**(2), 539–550.

Marmanis, D., Datcu, M., Esch, T. and Stilla, U. (2016), 'Deep learning earth observation classification using ImageNet pretrained networks', *IEEE Geoscience and Remote Sensing Letters* **13**(1), 105–109.
**URL:** *https://doi.org/10.1109/lgrs.2015.2499239*

Martin, Laura J. (2019), 'WebMD: Breast Biopsy for Breast Cancer Diagnosis', `https://www.webmd.com/breast-cancer/breast-biopsy`. [Online] Accessed: 2020-06-22.

Michael, W. (2020), 'seaborn: statistical data visualization', `https://seaborn.pydata.org/`. [Online] Accessed: 2021-08-09.

Oliphant, T. and developers, N. (2020), 'NumPy', `https://numpy.org/`. [Online] Accessed: 2021-08-09.

Osareh, A. and Shadgar, B. (2010), Machine learning techniques to diagnose breast cancer, *in* '2010 5th International Symposium on Health Informatics and Bioinformatics, HIBIT 2010', pp. 114–120.

Paliwal, M. and Kumar, U. A. (2009), 'Neural networks and statistical techniques: A review of applications'.

pandas development team, T. (2020), 'pandas-dev/pandas: Pandas'.
**URL:** *https://doi.org/10.5281/zenodo.3509134*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J. B. and Zuiderveld, K. (1987), 'Adaptive histogram equalization and its variations', *Computer Vision, Graphics, and Image Processing* **39**(3), 355–368.
**URL:** *https://doi.org/10.1016/s0734-189x(87)80186-x*

Polat, K. and Güneş, S. (2007), 'Breast cancer diagnosis using least square support vector machine', *Digital Signal Processing* **17**(4), 694–701.
**URL:** *https://linkinghub.elsevier.com/retrieve/pii/S1051200406001461*

Ramos-Pollán, R., Guevara-López, M. A., Suárez-Ortega, C., Díaz-Herrero, G., Franco-Valiente, J. M., Rubio-Del-Solar, M., González-De-Posada, N., Vaz, M. A. P., Loureiro, J. and Ramos, I. (2012), 'Discovering mammography-based machine learning classifiers for breast cancer diagnosis', *Journal of Medical Systems* **36**(4), 2259–2269.

Raschka, S. and Mirjalili, V. (2017), *Python machine learning*, Packt Publishing Ltd.

Russell, S. and Norvig, P. (2002), *Artificial intelligence: a modern approach*, Pearson.

Scott, A. J. and Knott, M. (1974), 'A cluster analysis method for grouping means in the analysis of variance', *Biometrics* **30**(3), 507.
**URL:** *https://doi.org/10.2307/2529204*

Sharif, M. M., Thrwat, A., Amin, I. I., Ella, A. and Hefeny, H. A. (2015), Enzyme function classification based on sequence alignment, *in* 'Advances in Intelligent Systems and Computing', Springer India, pp. 409–418.
**URL:** *https://doi.org/10.1007/978-81-322-2247-7*$_4$2

Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R. B. and Sieh, W. (2017), 'Deep Learning to Improve Breast Cancer Early Detection on Screening Mammography', *Scientific Reports* **9**(1).
**URL:** *http://arxiv.org/abs/1708.09427http://dx.doi.org/10.1038/s41598-019-48995-4*

Simonyan, K. and Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

Srivastava, N., Hinton, G., Krizhevsky, A. and Salakhutdinov, R. (2014), Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Technical report.

Szeliski, R. (2010), *Computer vision: algorithms and applications*, Springer Science & Business Media.

Wang, D., Khosla, A., Gargeya, R., Irshad, H. and Beck, A. H. (2016), 'Deep Learning for Identifying Metastatic Breast Cancer'.
**URL:** *http://arxiv.org/abs/1606.05718*

*What Mammograms Show: Calcifications, Cysts, Fibroadenomas* (2018), `https://www.breastcancer.org/symptoms/testing/types/mammograms/mamm_show`. [Online] Accessed: 2021-08-21.

who BC (2021), 'Breast cancer now most common form of cancer: Who taking action', `https://www.who.int/news/item/03-02-2021-breast-cancer-`

now-most-common-form-of-cancer-who-taking-action. [Online] Accessed: 2021-08-12.

Yala, A., Lehman, C., Schuster, T., Portnoi, T. and Barzilay, R. (2019), 'A Deep Learning Mammography-based Model for Improved Breast Cancer Risk Prediction', *Radiology* **292**(1), 60–66.
**URL:** *http://pubs.rsna.org/doi/10.1148/radiol.2019182716*

Yue, W., Wang, Z., Chen, H., Payne, A. and Liu, X. (2018), 'Machine learning with applications in breast cancer diagnosis and prognosis', *Designs* **2**(2), 13.
**URL:** *https://doi.org/10.3390/designs2020013*

ZUIDERVELD, K. (1994), 'Contrast limited adaptive histogram equalization', *Graphics Gems* **IV**, 474–485.
**URL:** *https://ci.nii.ac.jp/naid/10031105927/en/*