FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Matching of Mammographic Lesions in Different Breast Projections

Simão Pedro Ribeiro Quintans

FINAL VERSION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: José Costa Pereira

Second Supervisor: Eduardo Meca Castro

June 28, 2021

Resumo

De todos os cancros, cancro da mama é o que causa mais mortes entre mulheres. Programas de rastreio do cancro da mama podem ajudar a decrescer esta mortalidade, visto que deteção e tratamento do tumor em fases iniciais aumentam a taxa de sobrevivência. Normalmente, um par de radiologistas fazem a interpretação das mamografias, no entanto o processo é longo e cansativo. Isto incentivou o desenvolvimento de sistemas de diagnósitco auxiliado por computador (CADx), para substituir o segundo radiologista, fazendo melhor uso do tempo de especialistas. No entanto, sistemas CADx são associados a taxas elevadas de falsos positivos, dado que a maior parte detes apenas usam uma vista (*craniocaudal* ou *mediolateral oblique*) da mamografia. O radiologista, por sua vez, usa ambas as projeções, baseando o seu diagnóstico em diferenças visíveis entre as duas vistas.

Quando se consideram as duas projeções da mamografia, a correspondência de lesões é um passo necessário para se fazer o diagnóstico. No entanto, isto é uma tarefa complexa, dado que podem existir vários candidatos a lesão, em cada uma das vistas, para se fazer correspondência.

Neste trabalho, um sistema que faz correspondências entre lesões é proposto. Este é composto por três blocos: detetor de candidatos, extração de caraterísticas e correspondência de lesões. O primeiro é uma replicação do trabalho de Ribli et al., e o seu propósito é detetar possíveis candidatos a lesão. O segundo é a extração de vetores de caraterísticas de cada candidato, quer usando a *backbone* do detetor de candidatos, quer extraindo caraterísticas mais tradicionais, ou usando uma rede neuronal treinada com a *triplet loss* para distinguir lesões. O terceiro é o cálculo da distância entre os vetores de caraterísticas, usando também heurísticas para restringir possíveis pares de candidatos incorretos, e a ordenação de distâncias para atribuir a correspondência de cada lesão.

Este trabalho oferece várias opções de possíveis extractores de caraterísticas e heurísticas a serem incroporados num sistema CADx que seja baseado em detetores de objetos. O facto do modelo treinado com a triplet loss ser competitivo com os restantos modelos, torna o sistema bastante mais viável, sendo que este oferece a possibilidade de a correspondência ser independente da deteção de candidatos. Heurísticas *"hard"* e *"soft"* são introduzidas como métodos para limitar correspondências.

O sistema é capaz de fazer correspondências de forma satisfatória, dado que a sua exatidão ($\sim 70\% - 85\%$) é significativamente maior que a probabilidade aleatória ($30\% - 40\%$) dos dados usados. Heurísticas *"hard"* têm resultados encorajantes na *precision@k*, dado que estas rejeitam um número significativo de falsos positivos gerados pelo detetor de lesões.

Abstract

Of all cancer diseases, breast cancer is the most lethal among women. It has been shown that breast cancer screening programs can decrease mortality, since early detection increases the chances of survival. Usually, a pair of radiologists interpret the screening mammograms, however the process is long and exhausting. This has encouraged the development of computer aided diagnosis (CADx) systems to replace the second radiologist, making a better use of human-experts' time. But CADx systems are associated with high false positive rates, since most of them only use one view (craniocaudal or mediolateral oblique) of the screening mammogram. Radiologist, on the other hand, use both views; frequently reasoning about the diagnosis by noticeable differences between the two views.

When considering both projections of a mammogram, lesion matching is a necessary step to perform diagnosis. However this is a complex task, since there might be various lesion candidates on both projections to match.

In this work, a matching system is proposed. The system is a cascade of three blocks: candidates detector, feature extraction and lesion matching. The first is a replication of Ribli et al.'s Faster R-CNN and its purpose is to find possible lesion candidates. The second is the feature vector extraction of each candidate, either by using the candidates detector's backbone, handcrafted features or a siamese network model trained for distinguish lesions. The third is the calculus of the distance between feature vector, also using some heuristics to restrain possible non-lesion pairs, and the ranking of the distances to match the lesions.

This work provides several options of possible feature extractors and heuristics to be incorporated into a CADx system based on object detectors. The fact that the triplet loss trained models obtained competitive results with the other features extractors is valuable, since it offers some independence between the detection and matching tasks. "Hard" heuristics and "soft" heurisitcs are introduced as methods to restrain matching.

The system is able to detect matches satisfactorily, since its accuracy ($\sim 70\% - 85\%$) is significantly higher than chance level ($30\% - 40\%$). "Hard" heuristics proposals achieved encouraging results on precision@k, due to its match and candidates exclusion methods, which rejects a significant number of false positives generated by the object detector.

Agradecimentos

Aos meus pais e à minha irmã, por estarem sempre presentes - nos bons e maus momentos.

À Faculdade de Engenharia, por me ter dado as bases necessárias para que este trabalho fosse possível.

Ao INESC-TEC e ao VCMI group, por me terem acolhido e acompanhado durante estes meses.

Aos meus orientadores, José e Eduardo, por todo o conhecimento partilhado, pelo apoio e dedicação.

Simão Quintans

“(...) ficam, vão connosco, Quando partirmos saberemos, é sempre assim (...)”

José Saramago - *A Jangada de Pedra*

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
CADx	Computer-aided Diagnosis
CC	Craniocaudal
CNN	Convolutional Neural Network
Fast R-CNN	Fast Region Based Convolutional Neural
Faster R-CNN	Faster Region Based Convolutional Neural
MLO	Mediolateral Oblique
R-CNN	Region Based Convolutional Neural Network
ROI	Region of Interest
RPN	Region Proposal Network

Chapter 1

Introduction

1.1 Context

Of all types of cancer, breast cancer is, among women, the most diagnosed and the leading cause of death , resulting in over half a million causalities every year [14]. To decrease mortality, breast cancer screening programs have been widely adopted, since early detection increases the chances of survival [15]. The mammogram is an x-ray based exam used for screening; In the generated image bright areas are more radio-opaque, while darker areas are more radio-transparent. This exam can be used to detect abnormal areas in the breast. Even if it's not possible to diagnose cancer based on these areas alone, they can indicate that further testing is needed. These breast changes usually are either calcifications or masses, although there are other less common variations [16].

Many countries in Europe already have national or regional programs in which women of a certain age range (around 50-70) periodically receive an invitation to get a screening mammography exam [17]. Each screening provides two views of each breast: mediolateral oblique (MLO), taken under 45 degrees, and craniocaudal (CC), taken top-down.

Screening mammograms are evaluated by radiologists. This is a long, monotonous process, which is therefore exhausting and prone to errors [3]. The radiologist reads both projections to diagnose each case. They locate tumors, by detecting gray-scale and morphology anomalies in breast tissue, in a single projection. Then, the radiologist compares relevant features in both views, to verify his assessment of the diagnosis [18, 19]. In many breast cancer screening programs, there is a second radiologist that reads and interprets the same exam, reassuring the assessment made by the first radiologist. Both interpret the images to decide if the woman needs to be recalled for further evaluation. This dual-assessment is more expensive than a single one, but it has been shown to increase the detection rate of screening [17].

Computer-aided diagnosis (CADx) systems have been used by radiologists as a substitute to the second reader, to lower the false negative rate. These reduce, to some extent, the current dependence on the radiologist's experience and workload [4] and the cost of the second radiologist [15, 18]. Even with the advancement in lesion detection systems, these are still not widely used, since they generate a high rate of false positives [2]. This can be related to the fact that, usually, these systems detect and classify lesions on a single view [18]. Having access to complementary information in both views may increase accuracy. Fused-feature models can be used to merge the information between different images, therefore yielding CADx systems that find lesions more robustly. This information can be fused either using projections from different breasts (left and right), or projections from different views (CC and MLO) [19].

1.2 Motivation

Recently, deep learning based CADx systems have been proposed. While in traditional methods the focus is on handcrafted features learning techniques [15], deep learning models can learn which features are relevant for classification [17] – even if some of these are invisible to the untrained eye –, and are robust to image transformations [5]. Moreover, currently these systems perform as well as a radiologist and improve results when they are used as support decision-maker [4]. Even though research in deep learning based systems is advancing, there are not many of these works that consider the complementary information between projections and there is not a clear method to fuse them.

When considering both projections of a mammogram, lesion matching is a necessary step to perform diagnosis. However this is a complex task; there might be various lesion candidates on both projections to match. Moreover, breast appearance is different in the CC and MLO views, which implies different lesion shapes and positions in both projections.

A lesion matching algorithm is a valuable addition to CADx systems. It has the potential to lead to better results than single-view lesion detection algorithms. Potentially reducing false positive and false negative rates and ultimately improving robustness on screening mammography CADx systems.

1.3 Contributions

The main goal of this work is to develop an algorithm that can find correct matches between lesion candidates on different projections of the same breast. The proposed system will detect lesions on different projections of the mammogram, extract features and compare them across views to find matches.

The proposed system is evaluated to understand how well it performs on lesion matching, what features are relevant to that purpose and to what extent lesion matching can improve the overall robustness of CADx systems.

1.4 Dissertation Structure

There are five chapters in this document, being the first the present one. The second chapter is the Literature Review. It describes the state of the art on deep learning techniques for breast cancer detection/diagnosis. Additionally, object detectors are also reviewed. Chapter 3 introduces the reader to convolutional neural networks, automatic object detection and handcrafted features used in the present work. Implementation, results of the technical work and their discussion are described in Chapter 4. At last, the conclusions of the proposed system and future work possibilities are present in Chapter 5.

Chapter 2

Literature review

This chapter describes the state of the art on breast cancer detection, object detection methods and multiview methods for breast cancer detection and classification.

2.1 Breast Cancer Screening

Breast cancer screening methods, traditionally, consist of two stages: detection and classification. The first stage detects lesion candidates, while the second, given the region of interest (ROI), classifies the type of lesion.

2.1.1 Detection

Zheng & Chan [1] proposed an algorithm that marks out suspicious tumor regions, based on various AI methods. Firstly, they subdivided the mammograms into blocks of 16x16 and used a fractal technique [20] to detect regions that have higher intensities than their surroundings, which are associated with the presence of masses. Then a discrete wavelet transform (DWT)-based multiresolution segmentation algorithm (MMRF) is proposed. It consists in: 1) the image is decomposed into subimages with different resolutions, and a Markov Random Filter (MRF) [21] is applied to remove possible noise; 2) at each resolution, a dogs and rabbit clustering algorithm [22] simplifies the identification of tumors, dividing data into different sets; 3) the image is decomposed using DWT [1] and the image is segmented, obtaining different regions according to their gray-levels and texture (Figure 2.1).

Pereira et al. [23] proposed a lesion detection system based on segmentation. First they remove the non-breast elements and then enhance the image using wavelet multiresolution processing. Finally a segmentation algorithm, proposed by Hammouche [24], is used. This consists in combining wavelet theory with a genetic algorithm to determine the appropriate thresholds levels needed for Otsu segmentation [25]. To reduce false positives, a post-processing algorithm compares the detected region's shape with previously defined ones, excluding the ones that are not considered similar.

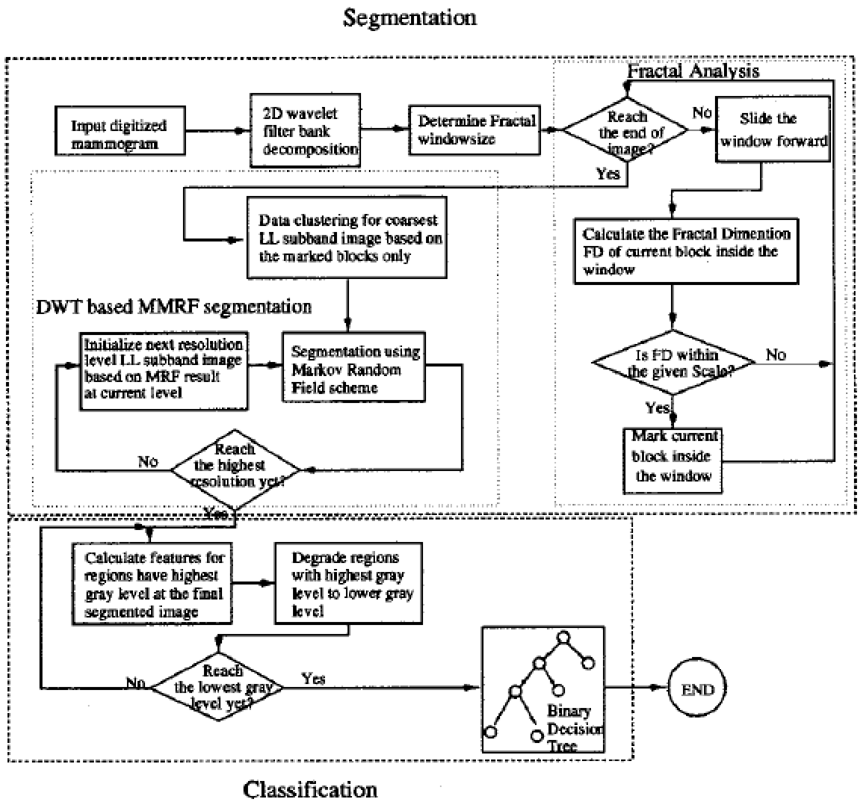

Segmentation
Fractal Analysis
Input digitized mammogram
2D wavelet filter bank decomposition
Determine Fractal windowsize
Reach the end of image?
No
Slide the window forward
Yes
Data clustering for coarsest LL subband image based on the marked blocks only
Calculate the Fractal Dimention FD of current block inside the window
DWT based MMRF segmentation
Initialize next resolution level LL subband image based on MRF result at current level
Segmentation using Markov Random Field scheme
Is FD within the given Scale?
No
Yes
Mark current block inside the window
No
Reach the highest resolution yet?
Yes
Calculate features for regions have highest gray level at the final segmented image
Degrade regions with highest gray level to lower gray level
No
Reach the lowest gray level yet?
Yes
Binary Decision Tree
END
Classification

Figure 2.1: Zhen & Chan [1] proposed segmentation algorithm.

With the development of deep learning over the last decades, various techniques have been proposed for breast cancer detection, surpassing traditional AI methods' results.

Dhungel et al. [2] proposed a combination of a multiresolution deep belief network classifier (m-DBN) [26] with a gaussian mixture model (GMM) [27] to the extraction of candidates (Figure 2.2). Then a cascade of two region based convolutional neural networks (R-CNNs) [28] process those regions and classify them as object or non-object. Some handcrafted features are extracted

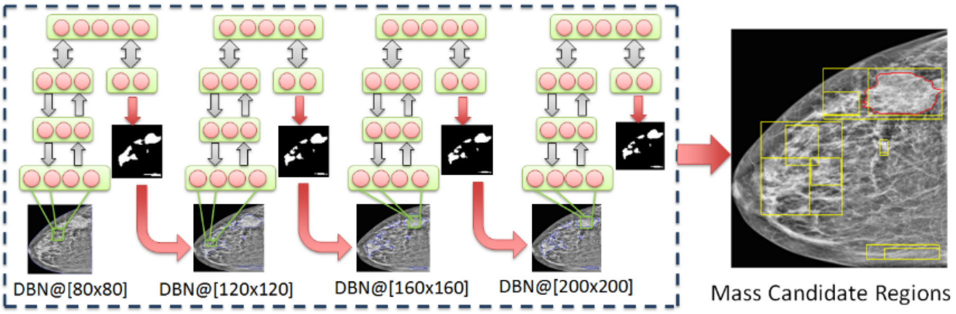


Figure 2.2: Candidate generation using m-DBN. At each iteration segmentation is done at a finer resolution improving the previous one.

from each of the R-CNN's resulting regions and are used to the inference of a cascade of two random forest classifiers [29] to reduce false positives. At last, the regions detected are clustered, merging regions that with a high overlap ratio using connected component analysis (Figure 2.3).

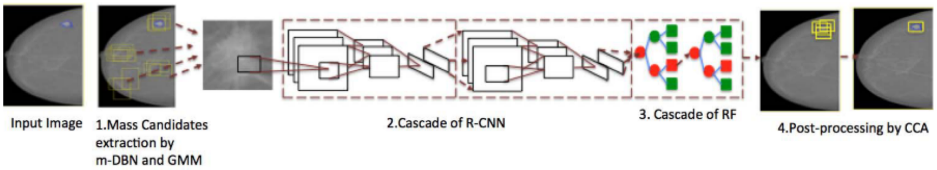


Figure 2.3: Dhungel et al. [2] proposed architecture.

Ribli et al. [3] proposed a method fully based on R-CNNs, using a faster region based convolutional neural network (Faster R-CNN) (Figure 2.4) [12] to detect mass candidates and classify them. This type of network, like R-CNN, is oriented to detect objects. It is composed of a convolutional neural network (CNN), a region proposal network (RPN), a ROI pooling layer and a final classifier. The output of this model is both the bounding box of the detected lesions and their classification. However, Faster R-CNN needs the bounding boxes of the dataset's objects, which is very limited in this clinical environment [3].

2.1.2 Classification

The introduction of CNNs has brought a revolution on computer interpretation of digital mammograms [17]. This type of network obtains high level features by the top layers of the model that are robust to image transformations and improve classification results [5]. Due to these advantages, CNNs have been used on various methods to either detect or classify breast cancer [2, 3, 4, 5, 15, 30, 31]. For time efficiency, as well to avoid situations where training data is very limited [3, 4, 5, 15, 31], pre-trained networks have also been used by some authors [31].

Kooi et al. in his work [32], made a classification comparison between handcrafted and CNN features. Their system took as input the segmented image; features where extracted and used for classification using various methods (support vector machine, multi-layer perceptron and gradient boosted trees). The results showed that CNNs achieve better results than handcrafted features, however, when features are combined, results are even better. Moreover, the model was compared to the radiologists' assessment, and performance on single view classification was similar.

Wang et al. [33] compared the accuracy on micro-calcifications segmentation between their stacked auto-encoders and machine learning algorithms (support vector machine, k-nearest neighbors and linear discriminant analysis). The results demonstrate that their deep learning model achieved better results than other machine learning methods because of the superior capacity of extracting features from the segmented image. That also led to a significant improvement in the discriminating accuracy between malign or benign micro-calcification compared to other learning methods.

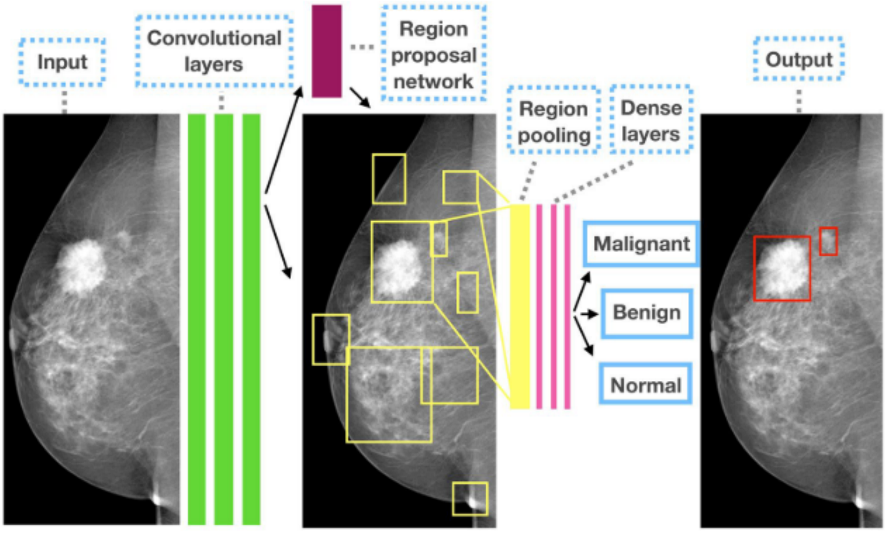


Figure 2.4: Ribli et al. [3] proposed architecture.

Levy et al. [31] proposed an end-to-end fully convolutional model. They used three architectures: an AlexNet [34], a GoogLeNet [35] and a baseline, which is inspired on the early layers of AlexNet. These models take as input ROIs of possible lesion locations in the mammogram. This proposal achieves state of the art results on classifying benign and malignant masses. However, these results might be biased since the training and testing sets are limited only to images that contain lesions.

Shen et al. [4] proposed a method that uses a pre-trained patch classifier to analyse the whole image. They use the shared weights property of CNNs to process various patches among the mammogram, in a sliding window mode, which results in a classification map of the various patches. Lastly, a cascade of two convolutional layers obtains the final classification (Figure 2.5). The model is trained in two phases: first on ROI annotated areas of the mammograms, second on the complete images. The proposed architecture requires ROI annotated images on the first phase, even if this model is fine-tuned to a dataset that lacks those annotations. This final phase can be valuable, since ROI annotated datasets are either small to train effectively or expensive to obtain.

2.2 Object Detection

In recent years, there has been significant progress in object detection algorithms. Those are of great significance to lesion matching, since it can be a straightforward method to spatially detect lesions in mammography screening. This kind of system can also extract lesion features that are used to compare projections.

In 2014 Girshick et al. [28] proposed one the first object detection systems based on deep learning: the region based convolutional neural network. They divided their system into three

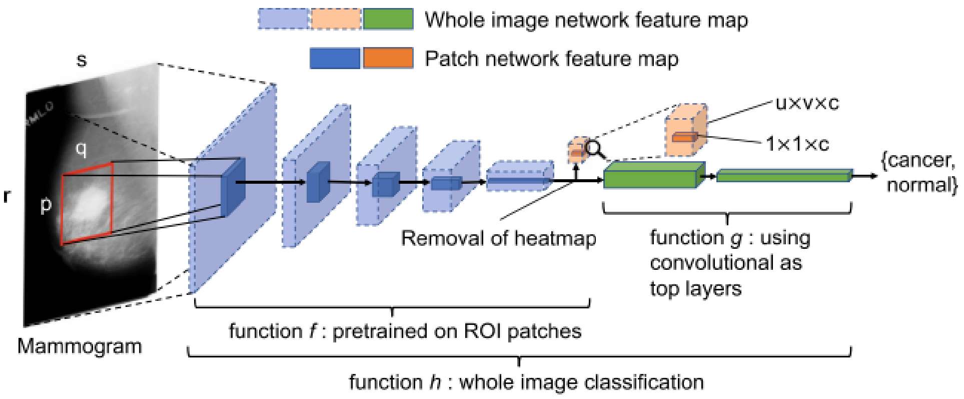

Whole image network feature map
Patch network feature map
s
r
q
p
MLO
Mammogram
u×v×c
1×1×c
{cancer, normal}
Removal of heatmap
function *g* : using convolutional as top layers
function *f* : pretrained on ROI patches
function *h* : whole image classification

Figure 2.5: Shen et al. [4] proposed architecture. f represents the patch classifier, g represents the map classifier and h represents the whole image classifier, which can be viewed as $h = g(f(x))$.

modules: the first generates region proposals, despite their category, using selective search [36]; the second is a CNN, that extracts a feature vector from each region proposal; the third is a set of class-specific support vector machines (SVM), that classify the detected objects. Lastly, a bounding box regressor – inspired on the deformable parts model employed in [36] –, is applied to reduce localization errors.

This method has three major disadvantages [28]:

1. It is hard to train, since it's a multistage pipeline: R-CNN first fine-tunes a CNN on object proposals, then fits an SVM to the CNN features, which acts as an object detector. Lastly, the bounding box regressor is learned;

2. Training needs a lot of time and space: for the SVMs and bounding box regressor, training features are extracted from each region and written in the disk, which means a CNN has to run and store a feature vector for all proposed regions in all images of the dataset;

3. Object detection is slow: at each object proposal, CNN extracts its features separately, before classification.

In 2015, Girshick [37] made a second proposal for an object detector: Fast R-CNN. This system would overcome the disadvantages of R-CNN, previously mentioned, through a single-stage training, using a multi-task loss, updating all network layers during training and not storing features in cache, saving hundreds of gigabytes of disk space. This model would train 9x faster than R-CNN and run 213x faster at test-time.

The system runs a series of convolutional and max pooling layers to create a convolutional feature map. For each region proposal, an ROI pooling layer extracts a feature vector from the feature map. Each of these is fed into a sequence of fully connected layers that branches into two output layers. Those are trained through a multi-task loss, i.e., both layers are trained in one training stage, avoiding pipelines and saving training time.

Faster R-CNN [12] differs from Fast R-CNN [37] on the method of detecting object proposals. Faster-RCNN dropped the selective search [36] used on both R-CNN [28] and Fast R-CNN, using a region proposal network (RPN), which shares computation with the Fast R-CNN detection network.

This system runs a series of convolutions in a sliding window mode, forming a feature map. At each window location, multiple regions are proposed, using multiple references boxes with different dimensions called anchor boxes. Proposals and their features (extracted from the feature map) are then used classify them –object/non-object. Afterwards, the regions classified as objects by the RPN, have their bounding box regression computed and classified according to the final labels, resulting the objects' locations and classifications.

Following Faster R-CNN [12], He et al. proposed Mask R-CNN [38]. This method's goal was to develop a comparably enabling framework for instance segmentation. Mask R-CNN extended Faster R-CNN by adding a new branch that predicts segmentation masks of each ROI, in parallel with the classification and bounding box regression.

Since Faster R-CNN was not designed to have pixel-to-pixel alignment between network inputs and outputs, which is crucial to construct the mask branch, a ROIAlign was developed to preserve exact spatial locations of the image, avoiding those misalignments. The introduction of ROIAlign layer proved to have a large impact, increasing mask accuracy by 10% to 50% on COCO dataset [39].

The previously mentioned object detectors are all two staged. The first generates a set of candidate object locations, while the second classifies each of the object locations. The one staged detectors' poorer results could be due to class imbalance: detectors would evaluate 10^4 to 10^5 candidate locations per image, but only a small fraction of these actually contained objects. This imbalance can make training inefficient, since most proposals would be negative, and those negatives could overwhelm training and lead to degenerate models [40].

Lin et al. [40] identified this issue and proposed RetinaNet, introducing the Focal Loss. This function applies a modulating term to the cross entropy loss to focus learning on hard negative examples (training gives more importance to misclassified objects than to misclassified background). Their proposal achieved state of the art results on object detection, offering a simple and highly effective solution.

2.3 Multiview Information Fusion

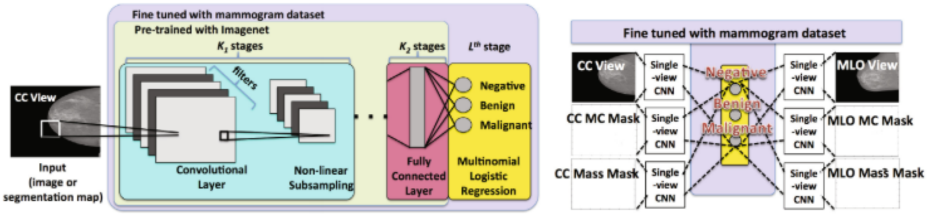
After the detection of lesion candidates, information fusion is an important step to correctly classify lesions, due to the clinical relevance in observing the same region in different views of the mammogram screening. This fusion can be either ipsilateral – when applied to the same breast on different projections (CC and MLO) –, or bilateral – when referring to the same projection on different breasts (left and right). Information of the images can be fused either by merging the feature vectors into one (e.g. concatenation, sum the vectors) and then classifying the lesion -

early fusion - or both views can be treated separately and the results of each image merged - late fusion [18].

2.3.1 Early Fusion

Wang et al. [19] proposed an ipsilateral method using extreme learning machine [41] and hand-crafted extracted features to detect lesions. Geometric and texture features are extracted from the ROI of each projection and some similarity features between CC and MLO views are also extracted. These three (similarity, CC and MLO features) are fused into a single vector. A selection of features is made by an heuristic. Finally these are submitted to the extreme learning machine, which predicts if the candidate is a lesion.

Carneiro et al. [5] proposed an end-to-end ipsilateral approach to classify lesions as malign, benign or negative, using multiple views of the breast. This system has as input the resized image of each view of the mammogram. The image is submited to a CNN, pre-trained on ImageNet [34], to extract features, then submitted to a fully connected layer, to obtain a single-view classification. Those extracted features are then used to classify the combination of both views, using a multinomial logistic regression layer (Figure 2.6). This system proposal's shows a clear improvement of multiview, over the single view methods, demonstrating that the CNN's higher level features contain a robust representation of the input image.



a) Single view CNN model b) Multiview CNN model

Figure 2.6: Carneiro et al. proposed architecture [5].

Geras et al. [30], like Carneiro et al. [5], proposed a multiview system based on CNNs, but instead of using two views, used all four (i.e. two of each breast). They also used the original dimension of the images, to preserve all the information. To reduce the computational requirements of handling full resolution, the CNN agressively downsamples the image, using larger strides on the first two convolutional and on the first pooling layers. This reduces greatly the size of the feature maps on the first layers. After the extraction of the features on each view, those are concatenated and submitted to a fully connected layer to be classified. Later, they resized the data to study the influence of resolution on the classification. It was verified that the best results were the ones referring to the original size of the data, showing the importance of preserving high resolution.

Khan et al. [15] also used the four views of the breast and a CNN to extract features. However, they used as input resized images of the ROIs and divided the classification into three stages: classification of the mammogram into normal or abnormal; classification of the abnormality into mass or calcification; classification of the lesion into malignant of benign.

2.3.2 Late Fusion

Shen et al. [4], in their end-to-end classification system, proposed an ipsilateral fusion, calculating the average classification of each pair of projections. This method significantly increased the results, compared to the single view approach. However it is questionable if this approach is the most efficient, since fusion of information is done only at the output.

Dhahbi et al. [42] proposed an ipsilateral fusion system. This receives both the CC and the MLO ROI of the breast, outputting the most similar pairs of mammograms from a reference database and their malignancy likelihood (Figure 2.7). Their proposal extracts multiresolution texture features from the input ROIs, using curvelet moments [6], from each view. A similarity score between the extracted features and other ROIs from the database is computed, using the inverse of the Euclidean distance. Both scores are fused using a weighted average, in which each weight is based on the reliability of the view (i.e. if in that projection the proposed lesion belongs to the same class as its neighbors, the more neighbors from the same class it has, the larger the weight is). At last a malignancy likelihood estimation is calculated.

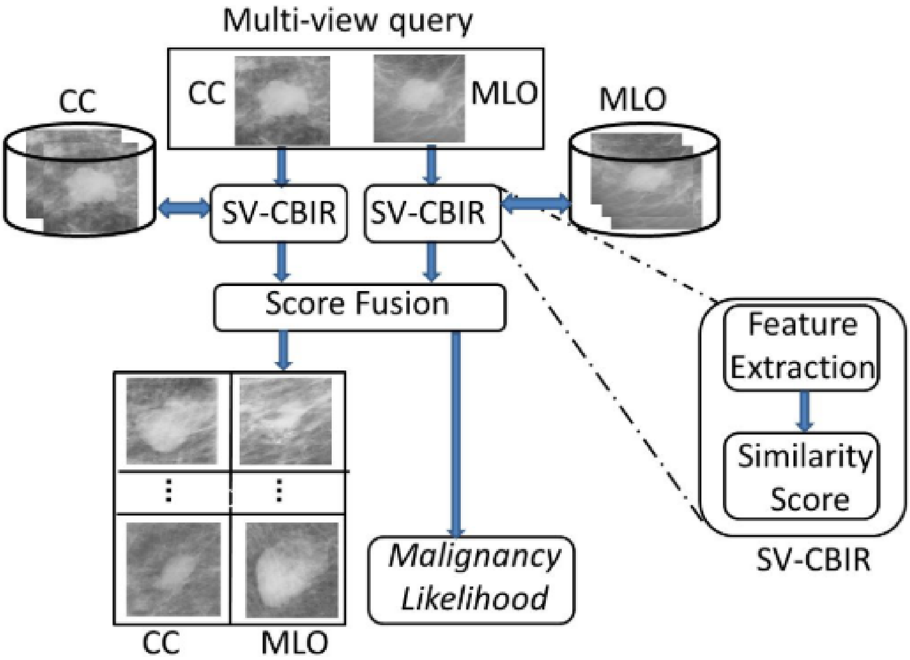


Figure 2.7: Dhahbi et al. proposed architecture [6].

Baâzaoui et al. [43] proposed and end-to-end model, which fuses information on four views (two of each breast), using similarity methods. The features are textural and are extracted using various 2D mathematical models. Then, a random forest classifier infers which distance metric is suitable to calculate similarity, based on the image's characteristics. The inferred distance metric is then used to compute similarity of each lesion's features in all views: the closer these metrics are to zero, the more similar images are. This system's procedure allows to classify lesions, ensuring semantic and visual similarities.

Chapter 3

Methodology

3.1 Convolutional Neural Network

3.1.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are computational systems, partially inspired on biological neural networks. These systems are known by their ability to learn from data to perform classification or regression tasks. An ANN is composed by multiple connected neurons.

Artificial Neural Networks are frequently just simple feed-forward networks. In this type of network, connections between nodes do not form cycles, i.e., the information flows in only one direction. One of the simplest networks of this type is the Multiple Layer Perceptron (MLP).

MLP networks consists of at least three layers of nodes: input layer, hidden layer and output layer (figure 3.1). The input layers' nodes receive the information directly from the the data:

$$o^{(0)} = X, \tag{3.1}$$

where $o^{(0)}$ is the input layer, containing N input nodes, and $X \in \mathbb{R}^N$ is the inputed data.

The subsequent node receives information from all the nodes from the previous layer and outputs a non-linear function (activation function) of the weighted sum of its inputs to all the nodes of the next layer:

$$o^{(l)} = f(W^{(l)}o^{(l-1)} + b^{(l)}), \tag{3.2}$$

where $o^{(l)}$ is the output vector of the lth layer and $f(.)$ is the activation function. $W^{(l)}$ and $b^{(l)}$ are the weights and *bias* vectors, which are the parameters that the algorithm learns during train.

The output of the MLP is given by equation :

$$\hat{y} = o^{(n)}, \tag{3.3}$$

where n is the last layer of the network.

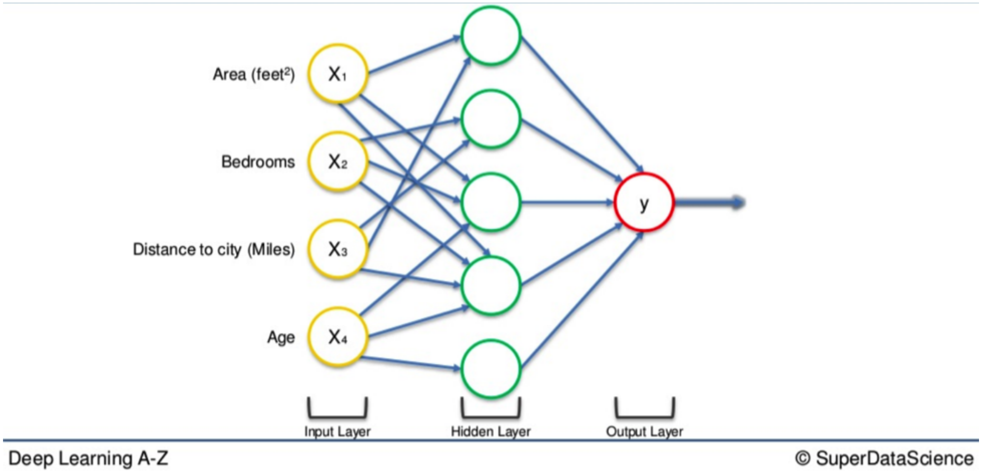


Figure 3.1: Multi Layer Perceptron
Source: SuperDataScience [7].

Activation Functions

Activation functions are the functions that introduce non-linearity in the network, to increase its discriminative power. Without those, the output of each node would be simply a linear combination of its inputs. The sigmoid (eq. 3.4) and the rectifier linear unit or ReLU (eq. 3.5) functions are widely used for this purpose. The first outputs a value between 0 and 1, which can be used to represent probabilities on classification problems. The ReLU zeroes out negative inputs and does not saturate at high values. (figure 3.2).

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{3.4}$$

$$R(z) = max(0,z) \tag{3.5}$$

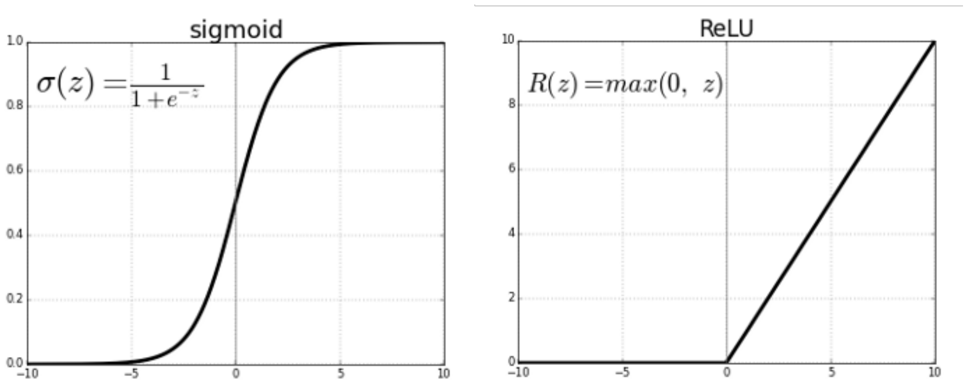


Figure 3.2: Sigmoid and ReLU activation functions
Source: TowardsDataScience [8].

Backpropagation

Backpropagation is an algorithm that is widely used for training the weight and bias parameters of a neural network. During training a loss function $L(y, \hat{y})$ is calculated using the predicted output $\hat{y}$ and the label (ground truth value) y. There are several loss functions relevant for different machine learning tasks. Cross Entropy (equation 3.6) and Mean Squared Error (equation 3.7) are some of the most popular loss functions used for classification and regression, respectively.

$$L(y,\hat{y}) = -\sum_{c=1}^{M} y_c \cdot log(\hat{y}_c), \tag{3.6}$$

where M is the number of classes and $y_c, \hat{y}_c$ the label and outputed prediction for class c.

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.7}$$

where n is the number of observations, y_i and $\hat{y}_i$ are the target and predicted i-th values of the observation set.

Then loss function's gradient is computed, from the last layer to the first, and aggregated to the existing gradient from the subsequent layer (the gradient propagates backwards). These are used to update the weights ($\frac{\partial L}{\partial W^{(l)}}$) and biases ($\frac{\partial L}{\partial b^{(l)}}$), starting from the last and finishing in the first layer.

Gradient Descent

The gradient descent is an optimization algorithm that is used to iteratively update weights and biases of a neural network. The main goal of this computation is to find a local minimum of the loss functions, i.e., the point where the network's predictions are the most accurate, according to the training data. To obtain that, parameters are updated in the opposite direction of the gradient of the loss function:

$$W_{t+1}^{(l)} = W_t^{(l)} - \eta \frac{\partial L}{\partial W^{(l)}} \tag{3.8}$$

$$b_{t+1}^{(l)} = b_t^{(l)} - \eta \frac{\partial L}{\partial b^{(l)}}, \tag{3.9}$$

where $W_t^{(l)}$ is the weight matrix of the l-th layer at the t-th training iteration, and η is the learning rate. The choice of the learning rate must be adequate - if it is too small, training will be very slow; if it is too large, the model may simply not converge.

Adam

There are some variations of the gradient descent algorithm, designed to improve convergence speed and avoid local minima. One of the most popular variations is Adam, which stands for

ADAptive Moment estimation [44].

Adam calculates adaptive learning rates for different parameters, using the first and second moment of the gradient. The parameters' update equations are hereby displayed:

$$\begin{aligned}
g_{(t)} &= \frac{\partial L}{\partial W^{(l)}_{(t-1)}} \\
m_{(t)} &= \beta_1 \cdot m_{(t-1)} + (1-\beta_1) \cdot g_{(t)} \\
v_{(t)} &= \beta_2 \cdot v_{(t-1)} + (1-\beta_2) \cdot g^2_{(t)} \\
\hat{m}_{(t)} &= m_{(t)}/(1-\beta_1^t) \\
\hat{v}_{(t)} &= v_{(t)}/(1-\beta_2^t) \\
W^{(l)}_{(t)} &= W^{(l)}_{(t-1)} - \eta \cdot \hat{m}_{(t)}/(\sqrt{\hat{v}_{(t)}} + \varepsilon) \qquad (3.10)
\end{aligned}$$

Where $m_{(t)}$ and $v_{(t)}$ are the first and second moment of the gradient, $\beta_1, \beta_2 \in [0,1[$ control the decay of the moving averages, and ε is a small constant to add numerical stability.

3.1.2 Convolutional Neural Networks

MLPs are not able to preserve spatial information (the input is the flattened image) and are very expensive, since each node is connected with all the nodes from the previous layer. The number of parameters (weights and biases) for an MLP with n hidden layers, i inputs and o outputs is given by:

$$\#parameters = i \cdot h_1 + \sum_{k=1}^{n-1} (h_k \cdot h_{k+1}) + h_n \cdot o + \sum_{k=1}^{n} h_k + o, \qquad (3.11)$$

where h_k is the number of parameters of the k-th hidden layer. While the number of the network's connections is equal to the number of weight parameters, which is the number of biases subtracted from the number of parameters:

$$\#connections = \#parameters - \#bias\_parameters \qquad (3.12)$$

This encourages the use of Convolutional Neural Networks (CNN), due to the parameter sharing and sparse connections among layers, - which make the network easier to train and more efficient - resulting in better generalization. The sparse connections also encourage the use of deeper networks, which can make the model learn higher level features.

CNNs can be divided into two main stages: feature extraction and classification (figure 3.3). The first is composed by convolutional, activation and pooling layers, which extract and preserve the information in a three dimensional fashion ($height \times width \times depth$); the second classifies the image, based on the flattened extracted information, using fully connected layers, similar to an MLP network.

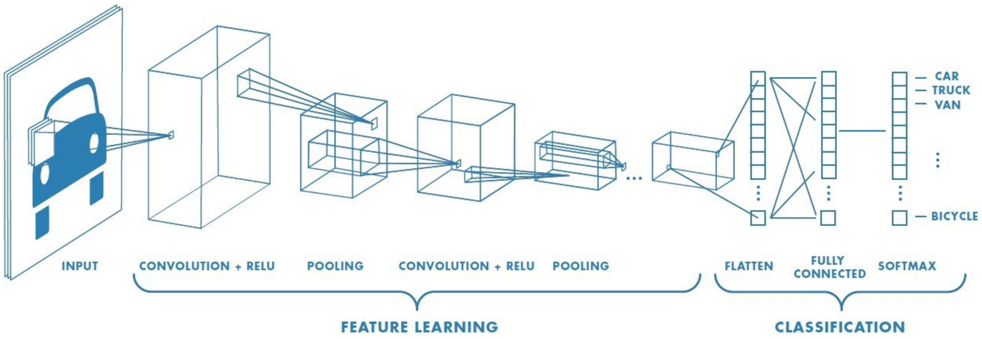


Figure 3.3: CNN structure
Source: TowardsDataScience [9].

Convolutional Layers

In convolutional layers, a set of kernels (typically a 3x3 matrix) run though the image, in a sliding window fashion, performing convolutions to create feature maps (figure 3.4a), using the following expression:

$$[I * k](u,v) = \sum_{y\in\mathbb{Z}} \sum_{x\in\mathbb{Z}} \sum_{c=1}^{C} I_c(x,y)k_c(x-u,y-v), \tag{3.13}$$

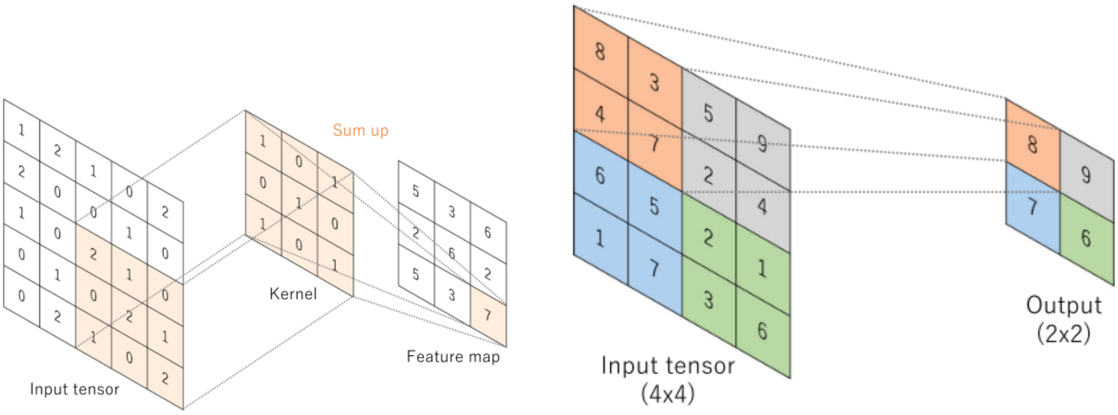
where matrix I is the image, k is the kernel and I_c, k_c is their representation in the channel c, of a total of C channels. The presented formula is just for one kernel. If a layer has 32 filters, it has 32 different ks and the result is their concatenation. The convolution operation allows to preserve spatial information and uses sparse connections. All convolutional layers use different kernels, obtaining different feature maps that represent different characteristics. As pointed by Yamashita [10], each kernel can be considered a feature extractor.

The kernel "movement" after each multiplication convolution is defined by a *stride* parameter, which is typically 1 pixel, but can take higher values to get faster convolutions and smaller outputs. Padding of the image is also often used to avoid loss of the borders' information at each convolutional layer - without padding, the kernels' convolutions are never centered in bordered pixels. Padding is also used to preserve the spatial dimensions of the input data along multiple layers.

Each element of the feature map is then passed through an activation function (typically a ReLU), obtaining the input to next layer of the network.

Pooling Layers

Pooling layers provide downsampling of the feature maps. They reduce the size of subsequent layers. The pooling layer divides the input tensor into patches and reduces each patch to one value by either selecting the maximum value (max pooling) or the average (average pooling). Like convolutional layers, pooling layers also have a stride parameter. This is usually the same size of the pooling kernel itself. Max pooling is one of the most popular pooling operations. Depending on the pooling-stride, it usually implements downsampling on the input tensor of the following layer. An example of a max pooling operation is displayed in figure 3.4b.



(a) Convolution example

(b) Max pooling example

Figure 3.4: Convolution and max-pooling examples [10].

Fully Connected Layer

As mentioned before, fully connected layers are similar to hidden layers in an MLP. These do not preserve spatial information, unlike their convolutional or pooling counterparts and can be used to funnel all neurons into a single unit to make a final prediction. The output feature maps from the feature extraction part are flattened into a 1D vector, which is fed to the first fully connected layer.

Regularization

If a model trains for too long or on limited data, it can become too adjusted to the training examples (overfit), which usually leads to lower accuracy at inference time on unseen data. Some regularization techniques can increase the model's ability to generalize, by "hampering" the "memorization" of the training data. Some of these techniques are:

- **Dropout:** one of the most common regularization techniques. During training, at each iteration random neurons of the network are "removed" from forward- and back-propagation. This is the equivalent of training a slightly different network. It causes the network to not focus the inference on few neurons, while limiting the ability of the neurons themselves to fit too much to individual training examples.

- **Data Augmentation:** adding new data is an effective way to avoid overfitting. However, it may be difficult to find relevant data to augment training. Data augmentation tackles this issue, by making random changes to the already existing training data - e.g., if an image contains a malignant cancer lesion, the same image rotated should also contain the malignant cancer lesion. Even if they are different images, from the diagnostic standpoint they contain the same (important) information: a malignant lesion. Other common data augmentation transformations are random horizontal or vertical flips, random rotations in general, (careful) crops and scaling (zoom in and zoom out).

VGG-16

The VGG-16 is a 16-layers deep CNN architecture proposed by Simonyan and Zisserman [45]. This model was developed for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The model receives an input of 224x224 pixels with 3-channels (i.e. 224x224x3) and outputs a vector of 1000 dimensions; this is the number of classes for that dataset. VGG-16 has also became a reference architecture widely used for different problems, due to the model's simplicity and to its satisfying results in various tasks.

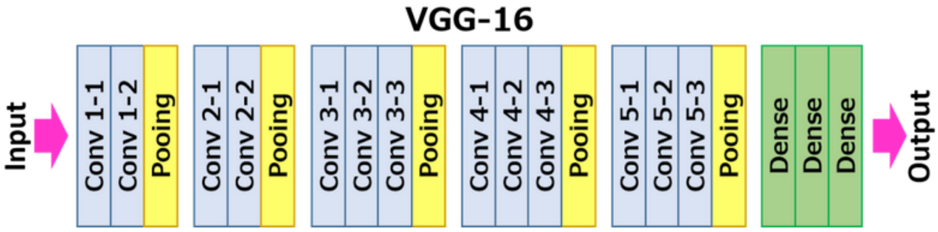


Figure 3.5: VGG-16 layers
Source: Neurohive [11].

The network is composed of a stack of convolutional layers, each followed by a relu, with 3x3 filters with stride and padding of 1 pixel. Five max-pooling layers are applied - note that only some convolutional layers are followed by pooling layers (figure 3.5) - over a 2x2 kernel with stride of 2 pixels.

The convolutional layers stack transforms the 3-channel deep 224x224 input patch, into a 7x7 patch with a depth of 512-channels. This is flattened and fed to two fully connected layers with 4096 nodes, each followed by a ReLU, and finally fed into another fully connected layer with 1000 output nodes (figure 3.6). However, it is often that the final layer of the network is substituted by another output layer, according to the desired number of outputs, either for classification or regression tasks.

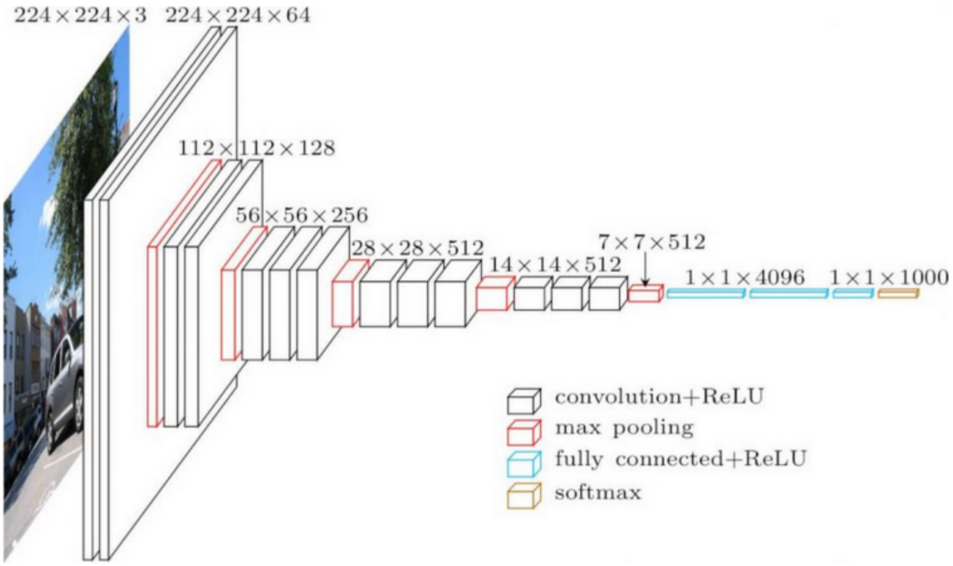
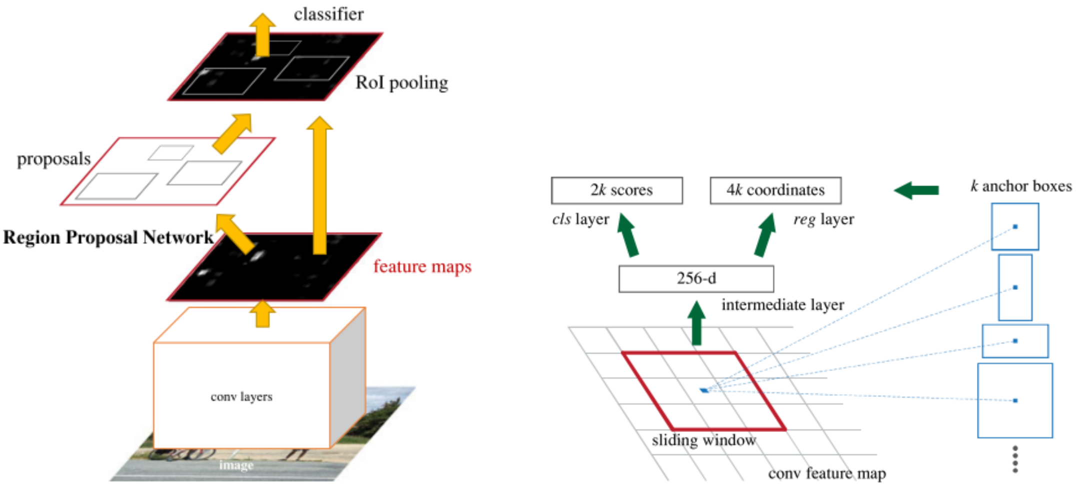


Figure 3.6: VGG-16 architecture
Source: Neurohive [11].

3.2 Faster R-CNN

The Faster R-CNN is a deep learning based object detector. This type of network is composed by a backbone (CNN model), a region proposal network (RPN), a ROIAlign layer and a classifier (fig. 3.7a).



(a) Faster R-CNN architecture

(b) Region proposals generation, using RPN

Figure 3.7: Faster R-CNN architecture and RPN.

Firstly, the Faster R-CNN's backbone produces several feature maps of the image (feature extraction). Then, the RPN runs a small network, taking as input a small portions of the feature maps, in a sliding window fashion. At each location of the feature maps, proposals are generated using k possible anchor boxes (k is the product between the number of possible sizes and scales of the box - figure 3.7b).

After the RPN generates proposals, each region proposal is classified into object or background and a bounding box is assigned to it by a regression model, in parallel, using the region's feature map section.

If a region is classified as an object by the RPN, the ROIAlign layer takes its feature map representation and converts into another fixed size feature map. However, different regions can have different sizes. To have a fixed dimension ($H \times W$) for every region, the ROIAlign layer divides each image with $h \times w$ size into a $H \times W$ grid of approximately $h/H \times w/W$ sized sub-windows. Each sub-window might contain multiple pixels, some of them only partially. To not lose information of those partial pixels, the values within the sub-window are sampled into four points, using bilinear interpolation. Finally, an average pooling operation is applied at each sub-window into the corresponding grid cell.

The resulting region's feature map is fed into a couple of fully connected layers and the resulting feature vector is fed into a classifier model, which classifies the region into its class and assigns a bounding box regression, both performed in parallel (figure 3.8).

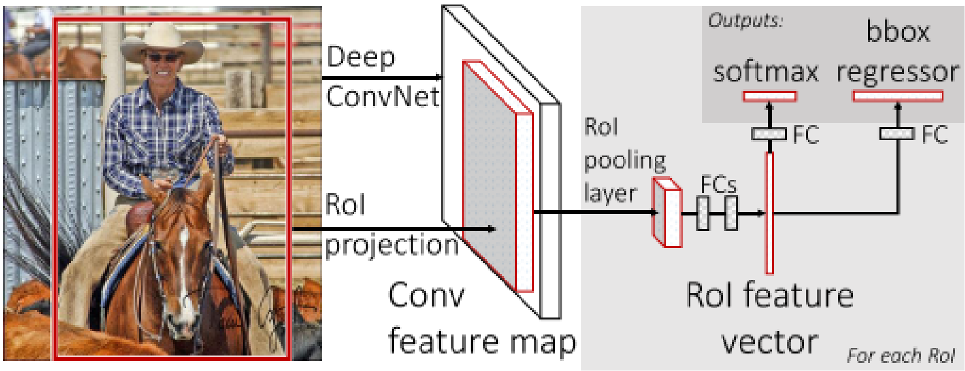


Figure 3.8: Faster R-CNN's classification and regression [12].

This process results in the classification and bounding box information of each region proposed and accepted by the RPN.

3.2.1 Faster R-CNN for Breast Cancer Detection

For the system that is to be proposed later in this chapter, a replication of Ribli et al.'s work [3] was made to act as a lesion detector. This Faster R-CNN uses a VGG-16 as backbone. This model was trained in CBIS-DDSM [46] and was tested in INbreast [47]. Both datasets are also used for this work, as described later in this chapter.

The weights of Ribli's model are available at a github repository [48] in the Caffe framework [49] format. They were converted to Pytorch (the selected deep learning framework for the system, as described later in chapter 4), and their dictionary keys were adjusted to fit a torchvision's Faster R-CNN, using a VGG-16 as backbone.

3.3 Handcrafted Features

An alternative way of representing images as vectors is to use handcrafted features - information obtained from the image that is not based in deep learning techniques.

3.3.1 Gray Level Co-occurrence Matrix

The gray level co-occurrence matrix (GLCM) is a matrix that is defined over an image, as the distribution of gray-scale values of all pixel pairs with a defined (location/position) offset.

Given an image I, its GLCM is defined by:

$$P_{\Delta x,\Delta y}(i,j)=\sum_{x=1}^{m}\sum_{y=1}^{n}\left\{\begin{matrix}1, & if \quad I(x,y)=i \quad and \quad I(x+\Delta x,y+\Delta y)=j\\ 0, & otherwise\end{matrix}\right\}, \tag{3.14}$$

where Δx and Δy are the offset parameters and the point $P_{\Delta x,\Delta y}(i,j)$ represents the number of times that the i-th and j-th pixel values appear in the image, given the offset parameters.

3.3.2 Wang et al. Feature-Vectors

In the present work, the feature-vectors from Wang et al. [19] are used to extract handcrafted features. These are extracted using the elements of the GLCM and some characteristics from the segmented lesions. The features are presented in table 3.1 and their variables are described in table 3.2.

3.4 Lesion Matching: Similarity

The similarity between candidates is evaluated by computing the Euclidean distance between the vectors – the lower it is, the most similar the lesions are:

$$d(f(CC),f(MLO))=||f(CC)-f(MLO)||_2^2, \tag{3.15}$$

where $f(.)$ is the feature vector of in a certain view, and $d(.,.)$ is the Euclidean distance between two feature vectors.

3.4.1 Triplet Loss

The triplet loss is a loss function that is used for metric learning. Models that use this loss function are trained using triplets, each containing a query image plus a positive and negative example (fig

Table 3.1: Handcrafted Features.

Feature Type	Name	Feature Expression
Morphology Features	Roundness	$g_1 = \frac{P^2}{A}$
	Entropy of standardized radius	$g_2 = -\sum_{k=1}^{100} p_k(\log(p_k))$
	Variance of standardized radius	$g_3 = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(d(i)-d_{avg})^2}$
	Ratio of area	$g_4 = \frac{1}{d_{avg}N}\sum_{i=1}^{N}(d(i)-d_{avg})$
	Roughness	$g_5 = \frac{1}{N}\sum_{i=1}^{N}\lvert d(i)-d(i+1)\rvert$
Texture features	Inverse difference moment	$t_1 = \sum\frac{P(i,j)}{1+(i-j)^2}$
	Entropy	$t_2 = \sum P(i,j)\times[-\ln(P(i,j))]$
	Energy	$t_3 = \sum P^2(i,j)$
	Correlated coefficient	$t_4 = \sum\frac{P(i,j)\times(i-\mu_x)\times(j-\mu_y)}{\delta_x\delta_y}$
	Contrast	$t_5 = (i-j)^2\times P(i,j)$

Table 3.2: Description of Variables.

Variable	Definition
P	Girth of the edge
A	Area
p_k	Probability of the standardized histogram
N	Number of edge points
$d(i)$	ith standardized radius of edge points
d_{avg}	Average standardized radius of edge points
$P(i,j)$	Element of row i and column j of the GLCM
μ_x	Mean value of P_x
μ_y	Mean value of P_y
δ_x	Standard deviation of P_x
δ_y	Standard deviation of P_y
P_x	$P_x = \sum_{j=1}^{N_j} P(i,j)$
P_y	$P_y = \sum_{i=1}^{N_i} P(i,j)$

3.9). The positive image is "*more similar*" to the query image than the negative [13]. In this work, a positive image would be a lesion that "*best matches*" the lesion presented as query, while the negative could be either a non-matching lesion or mammography's background tissue.

The goal of the triplet loss is to make the model learn an embedding function $f(.)$ that, for each image triplet $(q,\, p,\, n)$:

$$d(f(q),f(p)) < d(f(q),f(n)) + m, \tag{3.16}$$

where $d(.,.)$ is the Euclidean distance between two elements, and q,p,n are the query, positive

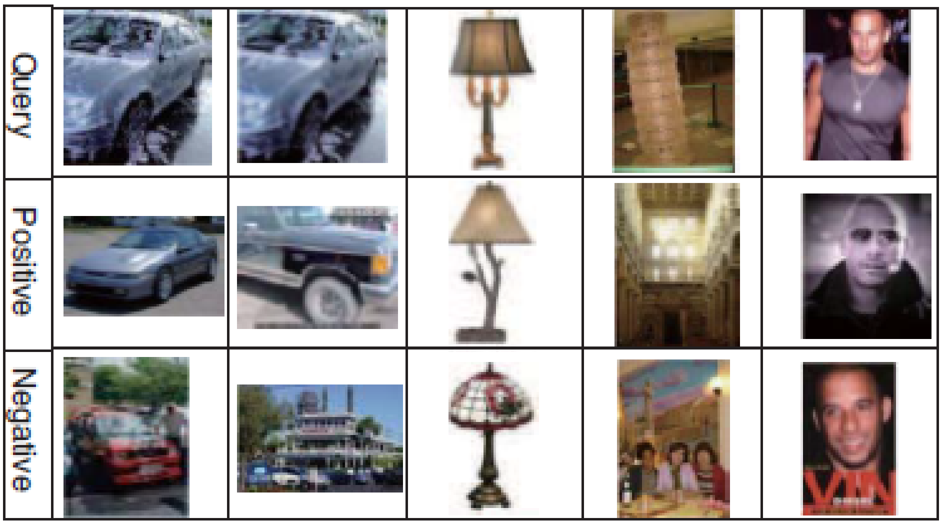


Figure 3.9: Sample images from triplets. Each column is a triplet. According to human raters, the positive images are more similar to their query than the negatives [13].

and negative images, respectively. This results in the triplet-loss function:

$$L(q,p,n) = max\{0, m + d(f(q), f(p)) - d(f(q), f(n))\}, \tag{3.17}$$

where m is an enforced margin parameter that defines the gap between the distance to the positive and to the negative images [50].

3.5 System Proposal

The proposed system is an early fusion method that uses both views from the same breast (ipsilateral). Our algorithm receives the images and outputs matched lesion pairs. It is done through a cascade of three stages: candidates detection, feature vector extraction and lesion matching (Figure 3.10).

Our system's goal is to detect matched lesions across different views, which can be valuable in the clinical environment to decrease false negative and false positive rates in breast cancer detection.

The "Faster R-CNN" is an object detector and is responsible for the generation of a set of candidates from each view (MLO or CC) of the same breast. In this work, no object detector is developed to detect lesions. Instead, Ribli et al.'s [3] Faster R-CNN is replicated. This model detects lesions and is trained in CBIS-DDSM [46] and tested in Inbreast [47] dataset.

The "feature vector extraction" block extracts characteristics from each candidate generated by the Faster R-CNN, which is used to compare with other candidates. Features can be either deep learning based - using the Faster R-CNN's backbone, or using models trained with the triplet loss

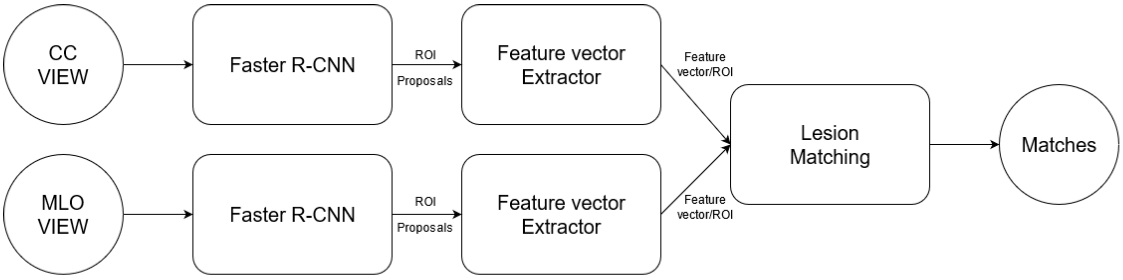


Figure 3.10: System overview.

-, handcrafted - morphological or textural [19] - or a fusion of both. Each feature vector can be viewed as a mathematical representation of the characteristics of its candidate.

"Lesion matching" is the final stage that matches candidates based on the similarity between their feature vectors, using the euclidean distance - the closer it is to 0, the more similar the candidates are. Matching can be enhanced by heuristics. Those can be hard (constrains matches) or soft (uses a multiplicative factor to encourage some matches), using some domain knowledge.

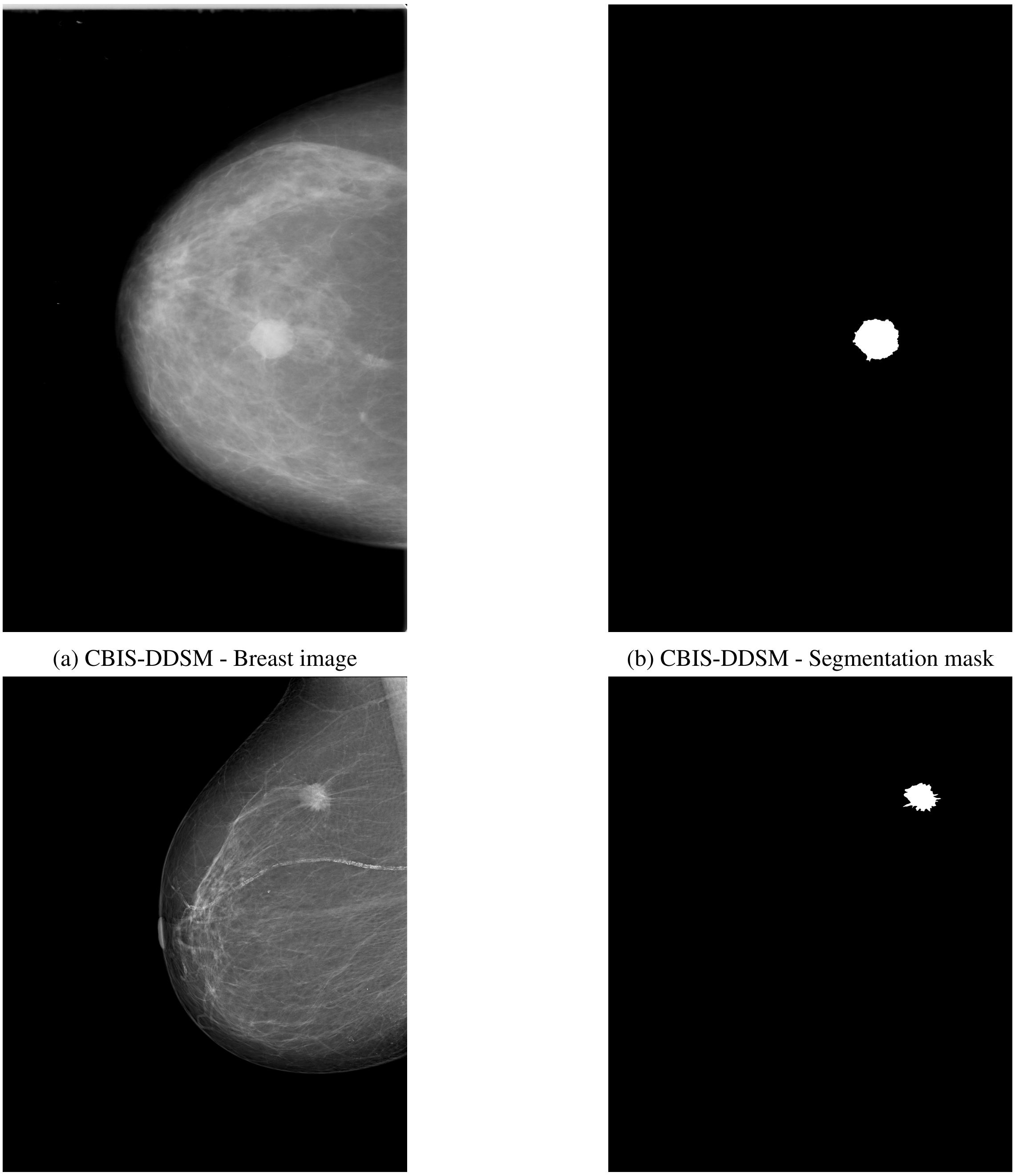
3.6 Datasets

Working with deep learning and computer vision is only possible if data exists to train the networks and validate results. Although most medical imaging datasets are proprietary, two widely used public datasets are used in this work: CBIS-DDSM and INbreast.

The Digital Database for screening mammography (DDSM) [51] is the most used dataset in literature. It contains $2,620$ mammography screening exams, in all four views (two images for each breast), adding up to a total of $10,480$ images. Each case has metadata containing the patient's age, date of the study, dense tissue category, resolution of the image, among others. Abnormal cases have additional information on the type of lesion (mass or calcification) and the breast imaging-reporting and data system (BI-RADS) [52] description. The dataset contains ROI annotations on identified lesions. The Curated Breast Imaging Subset of DDSM (CBIS-DDSM) [46] is a subset of DDSM, containing updated ROI annotations, metadata converted to CSV files, and images converted to DICOM files.

The INbreast dataset [47] contains 115 cases, totaling 410 full-field digital mammography (FFDM) high-resolution images. Both American College of Radiology (ACR) [53] breast density annotation and BI-RADS classification is provided. In abnormal cases, there is information about the type of the lesion and ROI annotated data. Additionally, on the MLO view, the pectoral muscle location is provided.

Breast samples and annotated data from both datasets are displayed in figure 3.11.



(a) CBIS-DDSM - Breast image

(b) CBIS-DDSM - Segmentation mask

(c) INbreast - Breast image

(d) INbreast - Segmentation mask

Figure 3.11: Breast image and annotated data from the different datasets used in this work.

Chapter 4

Implementation and Results

4.1 Frameworks and Dataset

All deep learning models described in this chapter were implemented in *Pytorch*. Image-level operations were mainly implemented using the *Pillow* packages, although *scikit-image* and *Open CV* were also used. For numeric operations, the *Numpy* package was used.

All the implemented code was written in Google Colaboratory [54]. The available GPU in this platform makes it well suited for machine learning problems.

4.1.1 Dataset

Annotated Data

All masses from the INbreast dataset [47] are used to create a custom dataset. ROIs are cropped from the breast images using the segmentation masks, maintaining the original size. The breast images are preprocessed using Ribli et al.'s method [3] - pixel values lower than 500 less than the image mode or 800 higher (the value range of the unprocessed images is 0-65535) were clipped, and rescaled to the range 0-255 (dividing the clipped values by 255).

In our experiments, only the ROIs with a known match are used for comparison; these are the ones suitable for algorithms that try to establish pairwise matches.

Faster R-CNN's candidates

For each breast image, objects detected by the Faster R-CNN's final classifier are also used in the final system. This dataset comprises 48 pairs (48 CC candidates are true positive lesions and have a true positive match in the MLO view). From the $3,624$ candidates generated by the Faster R-CNN, many are overlapping with other candidates and referring to the same object. Only non-overlapping candidates with highest scores are selected to be part of the custom dataset, in a total of 863 candidates (418 in CC view, 445 from MLO). Only candidates known to be lesions and to have a match are used.

4.2 Experimental Setups

The results presented are obtained using three experimental setups, which will be explained in detail later in this section. In both setups, matching is made through the assignment of the smallest Euclidean distance between candidates. Results are evaluated on Top1 - fraction of positive matches with minimum distance - and Top5 - fraction of positive matches contained in the top5 minimum distances.

ROIvCandidate Setup

In ROIvCandidate, each mass from a view (e.g. CC) is compared to all masses from the complimentary view (MLO) of that patient's breast. Additionally, regions are randomly sampled from the complimentary view of the patient's breast to have ten possible candidates.

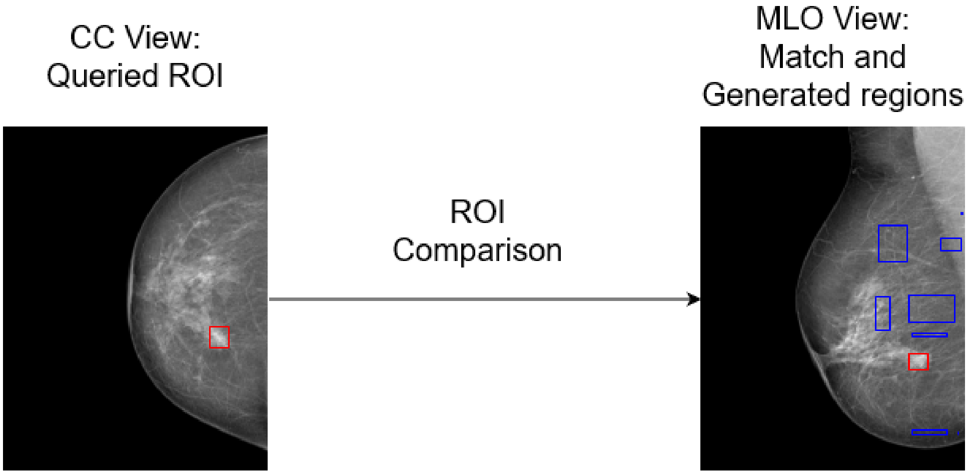


Figure 4.1: ROIvCandidate setup. On the left, the queried lesion (red box) and on the right the matching lesion (red box) and randomly generated regions (blue boxes) from the other view of the same breast.

ROIvROI Setup

In ROIvROI, each mass from a view (e.g. CC) is compared to all masses in the dataset that are present in the complimentary view (MLO).

In ROIvROI we aim to evaluate how effective the combined extracted features are at distinguishing lesions (i.e. lesions are compared at the dataset level). It is also a plausible way to evaluate matching with a perfect lesion detector (i.e., evaluate matching as if the object detector had 100% accuracy) (figure 4.2).

ROIvCandidate is perhaps the more realistic scenario, because it addresses the lesion-matching problem at the breast level. It is an easier task in the sense that its chance-level performance is significantly higher - 10% of correctly matching the lesion, *vs.* roughly 1.89% of ROIvROI setup.

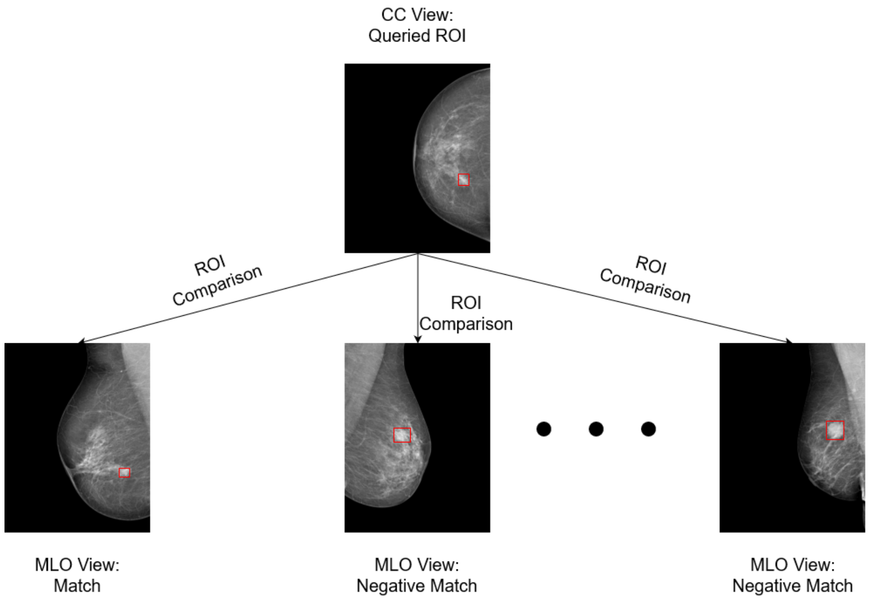


Figure 4.2: ROIvROI setup. On the top, the queried lesion (red box) and on the bottom multiple lesions from the other view of various breasts (red boxes), whereas only one is a match.

The false candidates in this setup are also, presumably, less similar than in the other setup since they are randomly generated areas.

Faster R-CNN ROIvCandidate Setup

This setup is similar to ROIvCandidate. Instead of comparing masses to randomly generated candidate regions from the other view of the patient's breast, the comparison is made between a known lesion from a view (e.g. CC) and the set of Faster R-CNN's proposed candidates from the other view (MLO) of this breast.

4.3 Feature Extraction

This block is responsible for extracting features from the proposed Faster R-CNN's candidates. For lesions that are a match, it is expected that the distance between their feature vectors is small. In this work, different feature extraction methods were used: Faster R-CNN's backbone features, handcrafted features and a fusion of both.

4.3.1 Faster R-CNN's backbone

As mentioned in the previous chapter, the Faster R-CNN has a CNN backbone that generates feature maps, which are used to make the object predictions. Due to this, the CNN backbone was used as a method to extract features. Additionally, the fully connected layers subsequent to the ROIAlign layer from the Faster R-CNN were used to extract some valuable higher-level features.

An L2 normalization was also applied to normalize the features vectors among different images (table 4.1).

Table 4.1: Different settings for feature extraction based on the Faster R-CNN model.

Name	CNN	Fully connected layers	L2 normalization
b1	Faster R-CNN's backbone	0	
b2	Faster R-CNN's backbone	1	
b3	Faster R-CNN's backbone	2	
b1_norm	Faster R-CNN's backbone	0	x
b2_norm	Faster R-CNN's backbone	1	x
b3_norm	Faster R-CNN's backbone	2	x

To evaluate the different combinations of feature extraction, ROIvROI and ROIvCandidate setups are used, and the results are displayed in table 4.2 (the best results for each metric are displayed in bold).

Table 4.2: Faster R-CNN's backbone's results for matching.

	ROIvROI		ROIvCandidate	
Model	Top1	Top5	Top1	Top5
b1	18 of 106 (16.98%)	46 of 106 (43.40%)	54 of 106 (50.94%)	95 of 106 (89.62%)
b2	18 of 106 (16.98%)	47 of 106 (44.34%)	**77 of 106 (72.64%)**	**102 of 106 (96.23%)**
b3	20 of 106 (18.87%)	44 of 106 (41.51%)	73 of 106 (68.88%)	**102 of 106 (96.23%)**
b1_norm	18 of 106 (16.98%)	**50 of 106 (47.17%)**	55 of 106 (51.89%)	93 of 106 (87.36%)
b2_norm	17 of 106 (16.04%)	46 of 106 (43.40%)	72 of 106 (67.92%)	**102 of 106 (96.23%)**
b3_norm	**21 of 106 (19.81%)**	45 of 106 (42.45%)	73 of 106 (68.88%)	101 of 106 (95.28%)

Discussion

- In ROIvCandidate, annotated ROIs are compared with randomly generated regions in the breast. While comparing the ROIs (objects) and the generated areas (background), it is expected that the accuracy is high, since the model is trained for detecting lesions. This, surprisingly, was not verified on the *b1* and *b1_norm* extractors(table 4.2), since they present poor results.

- L2 normalization does not clearly increases, neither decreases, the performance in the presented models, excluding the possibility of the features' scales interfering in matches.

4.3.2 Handcrafted Features

The handcrafted features are obtained through Wang et al.'s work [19]. Those are displayed in table 3.1 and their variables are defined in table 3.2. They are named accordingly in table 4.3. The feature vectors are separated by class - textural and morphological - and studied separately.

Table 4.3: Handcrafted Features' models.

Feature Type	**Name**	**Model Name**
Morphology Features	Roundness	$g1$
	Entropy of standardized radius	$g2$
	Variance of standardized radius	$g3$
	Ratio of area	$g4$
	Roughness	$g5$
Texture Features	Inverse difference moment	$t1$
	Entropy	$t2$
	Energy	$t3$
	Correlated coefficient	$t4$
	Contrast	$t5$

Morphological Features

Morphological features are obtained by direct application of the function to the ROI, the segmentation mask is used to isolate the region. An L2 normalization is applied to normalize feature vectors among different images.

Matching, while using these feature vectors, is evaluated using the ROIvROI setup, and its results are displayed in table 4.4 (the best results for each metric are displayed in bold).

Table 4.4: Morphological features' results for matching.

	ROIvROI	
Features	Top1	Top5
All	2 of 106 (1.89%)	15 of 106 (14.15%)
$g1$	**5 of 106 (4.72%)**	19 of 106 (17.92%)
$g2$	4 of 106 (3.77%)	9 of 106 (8.49%)
$g3$	4 of 106 (3.77%)	14 of 106 (13.21%)
$g4$	3 of 106 (2.83%)	16 of 106 (15.09%)
$g5$	2 of 106 (1.89%)	15 of 106 (14.15%)
All + L2	2 of 106 (1.89%)	**28 of 106 (26.42%)**

In the ROIvROI setup the results are slightly above chance level (1.89%). This fact and due to the non-existence of segmentation masks for the generated candidates, ROIvCandidates results for morphological features are omitted.

Textural Features

The textural features are obtained using the GLCM. This matrix is obtained using the built-in functions of the *scikit-image* package, followed by an auxiliary function, applied to extract the texture features. An L2 normalization is used as an attempt to normalize features vectors among different images. The results for this type of features are displayed in table 4.5 (the best results for each metric are displayed in bold).

Table 4.5: Texture features' results for matching.

	ROIvROI		ROIvCandidate	
Features	Top1	Top5	Top1	Top5
All	20 of 106 (18.87%)	**43 of 106 (40.57%)**	51 of 106 (48.11%)	92 of 106 (86.79%)
$t1$	7 of 106 (6.60%)	34 of 106 (32.08%)	41 of 106 (38.67%)	**94 of 106 (88.67%)**
$t2$	**21 of 106 (19.81%)**	42 of 106 (39.62%)	45 of 106 (42.45%)	93 of 106 (87.74%)
$t3$	7 of 106 (6.60%)	24 of 106 (22.64%)	**53 of 106 (50.00%)**	85 of 106 (80.19%)
$t4$	4 of 106 (3.77%)	29 of 106 (27.36%)	34 of 106 (32.08%)	84 of 106 (79.25%)
$t5$	1 of 106 (0.94%)	30 of 106 (28.30%)	47 of 106 (44.34%)	72 of 106 (67.92%)
All + L2	9 of 106 (8.49%)	30 of 106 (28.30%)	32 of 106 (30.19%)	90 of 106 (84.91%)

Discussion

- The morphological features do not achieve very good results, because of the similar appearance between masses in their segmentation masks, which led to similar feature vectors, even among non-matching lesions.
- The entropy (feature *t2* in table 4.5) achieved best results. In this feature, the $ln(.)$ factor generates high absolute values if the GLCM cointains zeros. Due to that, the "All" hypothesis had the second highest accuracy in ROIvROI, since the entropy value is large enough to have a much higher influence in the feature vector than the remaining features.
- L2 normalization, in texture features, lead to worse results. This means that matching, while using texture features, is dependant of the feature vector's scale. This dependence is probably caused by the entropy features, that generates high absolute values, leading to more disperse distances.

4.3.3 Fused Features

The settings from the Faster R-CNN's backbone and textural features that yielded the best results are used to fuse feature vectors. Fusion is achieved either by concatenating feature vectors, or averaging the distance between the feature vectors of both sets. Additionally, an L2 normalization is used in some concatenation cases to normalize feature vectors among different images. The different fusion combinations and their results are displayed in tables 4.6 and 4.7 (the best results for each metric are displayed in bold).

Table 4.6: Fused features models.

Name	Backbone	Texture	Fusion Method	L2
f1	Backbone(2fc)+L2	All features+ L2	Average	
f2	Backbone(2fc)+L2	All features+ L2	Concatenate	
f3	Backbone(1fc)	t2	Concatenate	x
f4	Backbone(1fc)	t3	Concatenate	x
f5	Backbone(1fc)	t2	Concatenate	
f6	Backbone(1fc)	t3	Concatenate	

Table 4.7: Fused features' results for matching.

	ROIvROI		ROIvCandidate	
Model	Top1	Top5	Top1	Top5
f1	20 of 106 (18.87%)	43 of 106 (40.57%)	70 of 106 (66.04%)	**104 of 106 (98.11%)**
f2	18 of 106 (16.98%)	**47 of 106 (44.34%)**	67 of 106 (63.21%)	101 of 106 (95.28%)
f3	**23 of 106 (21.70%)**	40 of 106 (37.74%)	53 of 106 (50.00%)	100 of 106 (94.34%)
f4	18 of 106 (16.98%)	46 of 106 (43.40%)	**73 of 106 (68.87%)**	101 of 106 (95.28%)
f5	20 of 106 (18.87%)	41 of 106 (38.68%)	49 of 106 (46.23%)	93 of 106 (87.34%)
f6	18 of 106 (16.98%)	**47 of 106 (44.34%)**)	72 of 106 (67.92%)	102 of 106 (96.23%)

Discussion

- There is some complementary information in textural and CNN features, since the best accuracy model in ROIvROI, among all feature classes (CNN, handcrafted and fusion), is a fusion model (table 4.7). However, this complementarity is possibly not optimized by using concatenation and average to fuse models, since it is not clearly visible in the remaining results. Moreover, generally, the ROIvCandidate results for fused models are worse than their backbone's results (table 4.2).

- Overall, the deep learning features achieve much better results than the handcrafted features. This can probably be due to the robustness of the deep learning model to image variations, which result in a better generalization.

- ROIvROI is the metric that evaluates the ability of the system to compare true lesions. The best the object detector is, the more important this metric is. This motivates the chosen feature extractor to have the best results in this setup, which makes the *f3* and *b3_norm* models (tables 4.7 and 4.2) the most appealing models, since the first has the best overall score in this setup and the second has a competitive accuracy in both ROIvROI and ROIvCandidates setups.

- Euclidean distance between the features of two detected candidates in different views is not necessarily an effective way of matching them. All the extracted features in this section are optimized for lesion detection, not to distinguish them (category-level similarity).

4.4 Similarity

Even if the previous feature extraction methods offer some representation of the lesions that allow matching, they are not optimized to compare matching candidates. The Faster R-CNN model was trained to detect lesions, not to match them. Thus, there is no guaranty that the deep feature extraction method proposed has optimal parameters for the matching task. In this section, we evaluate if a model specifically trained for this task can outperform the previous feature extraction methods.

All the models use a torchvision pretrained VGG-16 [45] model. Optimization is done with ADAM and the learning rate is set to $1e-05$. The objective function is the triplet loss with the margin parameter set to 0.3. All models were trained for 40 epochs. The saved model is the one where best validation accuracy is achieved - proportion of instances where the loss values is 0 ($d(q,p)+0.3 < d(q,n)$) during an epoch. Inference is done with the best model after every 10 epochs of training, using ROIvROI and ROIvCandidate Setups.

The training data is comprised of 75% (964 triplets) of the CBIS paired lesions, and the validation data is the remaining 25% (322 triplets). The triplets were generated by associating one non-matching lesion (negative), to the query (anchor) and the positive matching lesion. Two models use "*Negative Online Mining*" on the training data. This samples one random non-matching lesion as the triplet's negative, instead of using a fixed negative (i.e., different epochs may have different negatives among the triplets).

The trained models differ in the data augmentation type and were named accordingly:

- ***None***: No data augmentation;
- ***DA***: Classical - random horizontal flips and random rotations (multiples of 90º);
- ***NOM***: Negative online mining;
- ***NOM-DA***: Classical data augmentation and negative online mining.

The train and validation accuracy are displayed in figures 4.3a and 4.3b, respectively. The models' inference results – with regards to the introduced experimental setups and metrics – are displayed in tables 4.8, 4.9, 4.10 and 4.11 (in this particular set of experiments, only the best results of all sets at each metric are bolded, since these models can not be fused).

Table 4.8: *None* model inference results.

	ROIvROI		ROIvCandidate	
Epoch	Top1	Top5	Top1	Top5
10	12 of 106 (11.32%)	47 of 106 (44.34%)	54 of 106 (50.94%)	100 of 106 (94.34%)
20	12 of 106 (11.32%)	47 of 106 (44.34%)	54 of 106 (50.94%)	100 of 106 (94.34%)
30	12 of 106 (11.32%)	47 of 106 (44.34%)	54 of 106 (50.94%)	100 of 106 (94.34%)
40	20 of 106 (18.87%)	39 of 106 (36.79%)	56 of 106 (52.83%)	103 of 106 (97.17%)

Table 4.9: *DA* experiment's inference results.

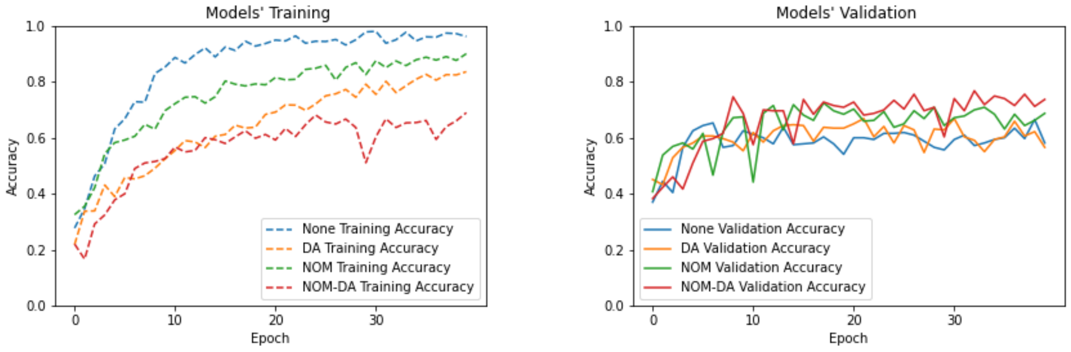
	ROIvROI		ROIvCandidate	
Epoch	Top1	Top5	Top1	Top5
10	9 of 106 (8.49%)	42 of 106 (39.62%)	51 of 106 (48.11%)	101 of 106 (95.28%)
20	9 of 106 (8.49%)	35 of 106 (33.02%)	48 of 106 (45.28%)	97 of 106 (91.51%)
30	12 of 106 (11.32%)	45 of 106 (42.45%)	**70 of 106 (66.04%)**	99 of 106 (93.39%)
40	12 of 106 (11.32%)	45 of 106 (42.45%)	**70 of 106 (66.04%)**	99 of 106 (93.39%)

Table 4.10: *NOM* experiment's inference results.

	ROIvROI		ROIvCandidate	
Epoch	Top1	Top5	Top1	Top5
10	**21 of 106 (19.81%)**	**48 of 106 (45.28%)**	59 of 106 (55.66%)	100 of 106 (94.34%)
20	19 of 106 (17.92%)	**48 of 106 (45.28%)**	59 of 106 (55.66%)	103 of 106 (97.17%)
30	19 of 106 (17.92%)	**48 of 106 (45.28%)**	59 of 106 (55.66%)	103 of 106 (97.17%)
40	19 of 106 (17.92%)	**48 of 106 (45.28%)**	59 of 106 (55.66%)	103 of 106 (97.17%)

Table 4.11: *NOM-DA* experiment's inference results.

	ROIvROI		ROIvCandidate	
Epoch	Top1	Top5	Top1	Top5
10	11 of 106 (10.38%)	38 of 106 (35.85%)	53 of 106 (50.00%)	**105 of 106 (99.06%)**
20	11 of 106 (10.38%)	38 of 106 (35.85%)	53 of 106 (50.00%)	**105 of 106 (99.06%)**
30	12 of 106 (11.32%)	47 of 106 (44.34%)	53 of 106 (50.00%)	101 of 106 (95.28%)
40	6 of 106 (5.66%)	46 of 106 (43.40%)	50 of 106 (47.17%)	100 of 106 (94.34%)



(a) Training Accuracy

(b) Validation Accuracy

Figure 4.3: Training and Validation Accuracy of each experiment.

Discussion

From the presented results, some notes can be made:

- During training, the models that have more data augmentation take longer to achieve 60% of training accuracy and present better results in validation. This means that the models are able to generalize, to a certain extent, to new data in CBIS-DDSM dataset.
- The validation accuracy can not be correlated with the inference's results. The first has a margin parameter of 0.3 as criterion, while the second, in practice, has a margin parameter of 0. Moreover, chance level during training is 50%, while the chance level at inference time is lower (1.89% in ROIvROI, 10% in ROIvCandidates).
- All experiments' models seem to overfit the training data - the less regularization it has, the more noticeable it is; the training accuracy seems to be constantly increasing, while

validation accuracy does not clearly improve from epoch 15 onwards.

- Counter intuitively, the experiments which perform best on the validation data, performs worst during inference. There are some possible hypothesis for this:

  1. The experiments with no data augmentation (*None* and *NOM*) might have learned features that are related to the orientation of the lesion.
  2. On the other hand, the data augmented experiments might have learned instead some specific features of the training data's domain (CBIS-DDSM), worsening the inference results (INbreast).

4.5 Ranking and Heuristics

4.5.1 Introduction

Deep Learning feature vectors are a valuable source of information for comparing lesions. However, the "*black-box*" paradigm does not allow to know if some domain knowledge characteristics are being used. Here we study the possibility to use some of those characteristics to improve matching between lesions. In particular: the area ratio between lesions, the absolute distance of the x position across views (Δx), and the score given by the Faster R-CNN:

- The area is known to be approximately similar for the same lesion in different views (e.g., a small lesion in the CC view will not be a large lesion in the MLO view, since it is limited to the real size of the tumor). The closer this ratio is to 1, the more similar the areas of the candidates are;

- The x-axis position of matches are known to be approximately similar (e.g., a lesion close to the nipple in the CC view will not be close to the pectoral muscle in the MLO view). Having a Δx that defines the distance between the candidates' x-axis position, the closer this value is to 0, the more similar the candidates' positions are.

- Score is the model's confidence that the candidate is a true lesion; the closer this value is to 1, the more confident the model is.

Two types of heuristics were explored:

- **Hard Heuristics**: are used as constraints, i.e., thresholds defined among different characteristics to disallow some lesion pairs;

- **Soft Heuristics**: are used as a multiplicative factor on the distance to condition the matching without restricting it.

4.5.2 Hard Heuristics

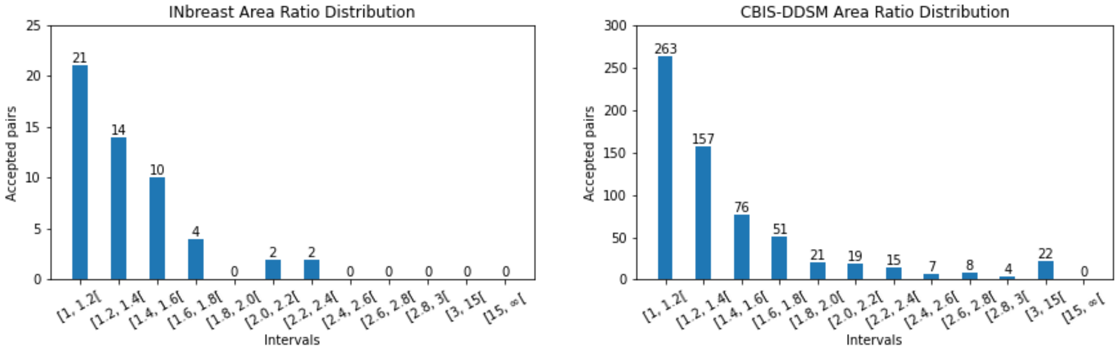
Area Ratio

Area Ratio is defined by the following expression:

$$area\_ratio = max\left\{\frac{a_{cc}}{a_{mlo}}, \frac{a_{mlo}}{a_{cc}}\right\}, \tag{4.1}$$

where a_{cc} and a_{mlo} are the area of the candidates in the CC and MLO views, respectively. For each pair of candidates, the closer the ratio is to 1, the more likely it is to be a true pair (matching lesions). In this heuristic, we define a threshold to be the maximum accepted area ratio.

The distribution of the area ratio among true pairs of lesions (using the radiologists' annotations for INbreast [47] and CBIS-DDSM [46]) are displayed in figures 4.4a and 4.4b, respectively.



(a) INbreast

(b) CBIS-DDSM

Figure 4.4: Area Ratio's distribution in annotated data.

In both datasets, the closer the an *area_ratio* interval is to 1, the more matches are contained in that interval. This reassures the proposition that matching candidates have similar areas.

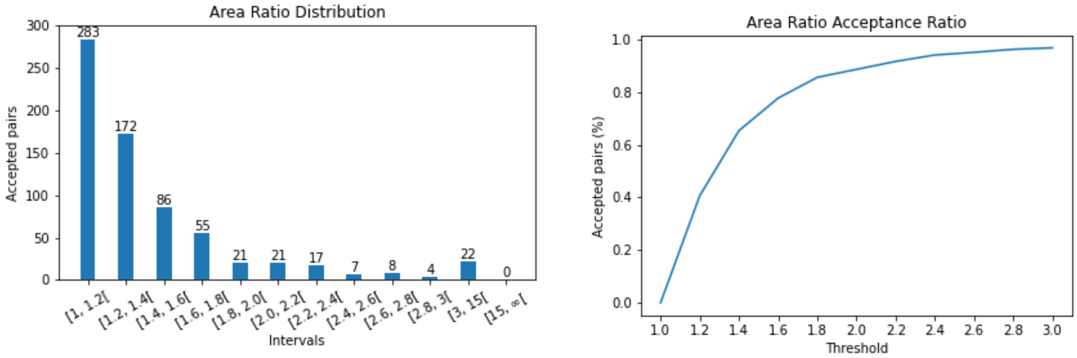
INbreast and CBIS-DDSM datasets are merged and analyzed to determine possible thresholds and heuristics. Their distribution and area ratio acceptance/threshold are displayed in figures 4.5a and 4.5b.

The distribution of the area ratio among true and false positive matches using the Faster R-CNN's candidates is also studied, and is displayed in figures 4.6a and 4.6b. False positives are the area ratios between a candidate, that is known to be a lesion and to have a pair, and a candidate that is not its match. For this distribution, only the candidates of the same breast from different views are available for matching. The area ratio between these matches is computed using the Faster R-CNN's candidates' predicted boxes.

Based on the distribution shown, the following thresholds were defined:

1. No threshold

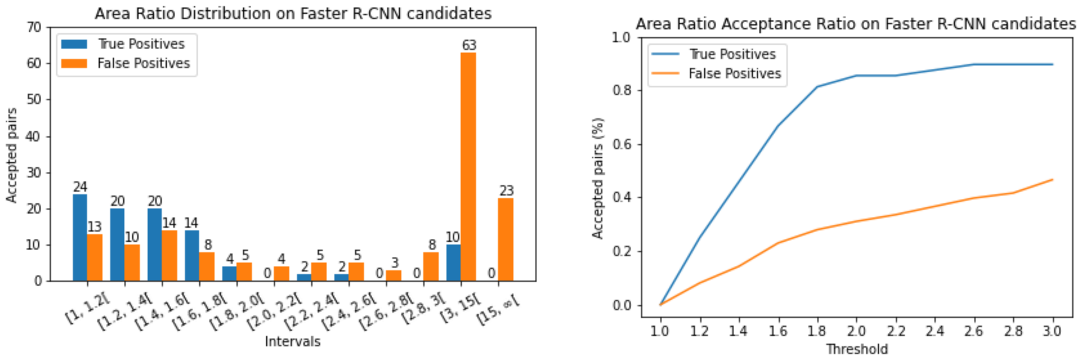
2. Mean of distribution: 1.4



(a) Distribution

(b) Acceptance ratio

Figure 4.5: Area Ratio's distribution among the merged INbreast and CBIS-DDSM annotated data.



(a) Distribution

(b) Acceptance ratio

Figure 4.6: Area Ratio's distribution among Faster R-CNN's candidates.

3. Mean + standard deviation (std) of distribution: 2.0

4. Mean + 2*std: 2.6

5. Mean +3*std: 3.0

6. Figure 4.5b's approximate point where the second derivative is 0: 1.6

7. Figure 4.6b maximum gap between True Positive and False Positive rate: 1.8

The results for each threshold value are shown in table 4.12 – using model *b3_norm* from table 4.1 –, on ROIvROI and Faster R-CNN ROIvCandidate setups (the best results for each metric are displayed in bold).

Table 4.12: Inference results for Area Ratio *Hard Heuristic*.

		ROIvROI		Faster R-CNN ROIvCandidate	
#	Threshold	Top1	Top5	Top1	Top5
1	None	21 of 106 (19.81%)	45 of 106 (42.45%)	65 of 96 (67.71%)	**96 of 96 (100.00%)**
2	1.4	21 of 106 (19.81%)	57 of 106 (53.77%)	43 of 96 (44.79%)	46 of 96 (47.92%)
3	2.0	23 of 106 (21.70%)	63 of 106 (59.43%)	75 of 96 (78.12%)	84 of 96 (87.50%)
4	2.6	21 of 106 (19.81%)	56 of 106 (52.83%)	**76 of 96 (79.17%)**	88 of 96 (91.67%)
5	3.0	21 of 106 (19.81%)	54 of 106 (50.94%)	72 of 96 (75.00%)	88 of 96 (91.67%)
6	1.6	**25 of 106 (23.58%)**	**64 of 106 (60.38%)**	60 of 96 (62.50%)	66 of 96 (68.75%)
7	1.8	**25 of 106 (23.58%)**	**64 of 106 (60.38%)**	72 of 96 (75.00%)	80 of 96 (83.33%)

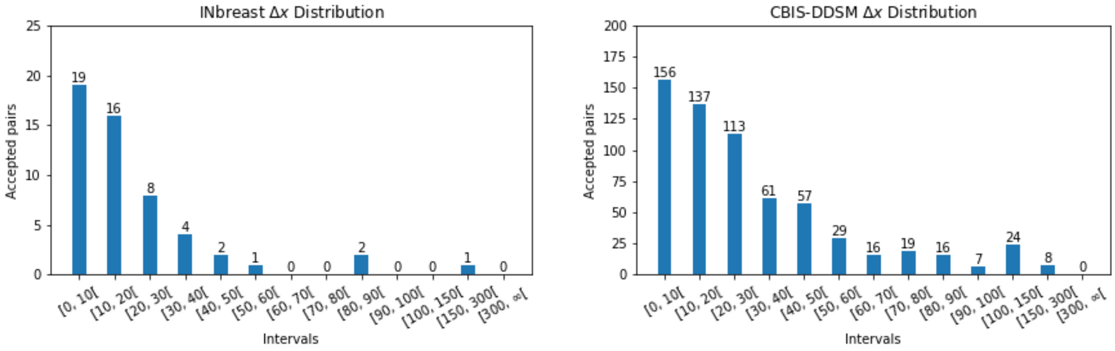
Delta x

Δx was obtained using the following expression:

$$\Delta x = |x_{cc} - x_{mlo}|, \tag{4.2}$$

where x_{cc} and x_{mlo} are the x-axis central points of the candidates' bounding boxes in the CC and MLO views, respectively. For each pair of candidates, the closer Δx is to 0, the likelier it is to be a true pair. In this heuristic, a threshold is defined as the maximum Δx value.

Similar to the "Area Ratio" heuristic, distribution and acceptance-ratio are studied, using the radiologists' annotated data for INbreast and CBIS-DDSM datasets (figure 4.7).



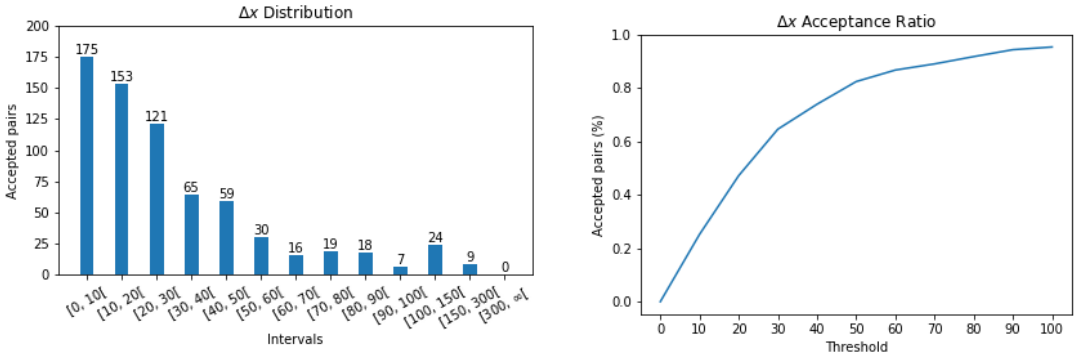
(a) INbreast

(b) CBIS-DDSM

Figure 4.7: Δx distribution in annotated data.

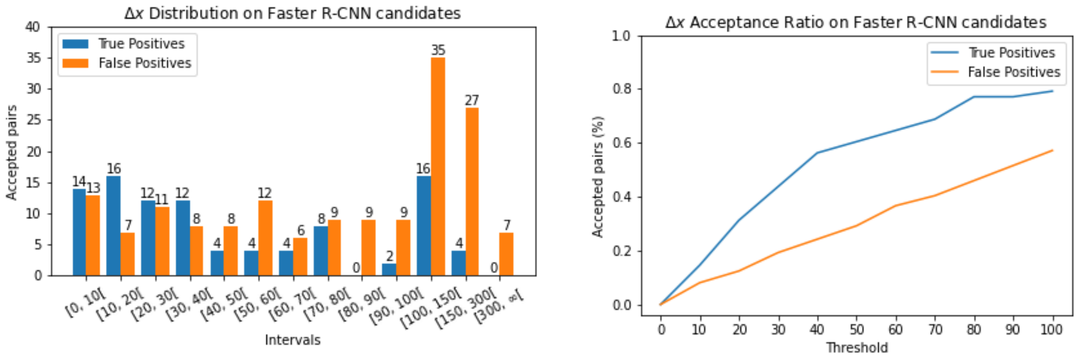
In both datasets, the closer the Δx's intervals are to 0, the most matches are contained in that interval. This reassures the proposition that matching candidates have similar positions in the x-axis.

The merged INbreast and CBIS-DDSM datasets (figure 4.8) and Faster R-CNN's candidates (figure 4.9) are analyzed, like in the "Area Ratio" heuristic, to determine possible thresholds and heuristics. In the Faster R-CNN's distribution, only the candidates of the same breast from different views are available for matching. The Δx between these matches is computed using the x-axis central point of the Faster R-CNN's candidates' predicted boxes.



(a) Distribution (b) Acceptance Ratio

Figure 4.8: Δx distribution among INbreast and CBIS-DDSM annotated data.



(a) Distribution (b) Acceptance Ratio

Figure 4.9: Δx distribution among Faster R-CNN's candidates.

From the graphics and distributions, the following thresholds are defined:

1. No threshold
2. Mean of distribution: 30
3. Mean + standard deviation (std) of distribution: 60
4. Mean + 2*std: 90
5. Figure 4.8b's approximate point where the second derivative is 0: 40
6. Figure 4.9b maximum gap between True Positive and False Positive rate: 80

The results for different thresholds are shown in table 4.13.

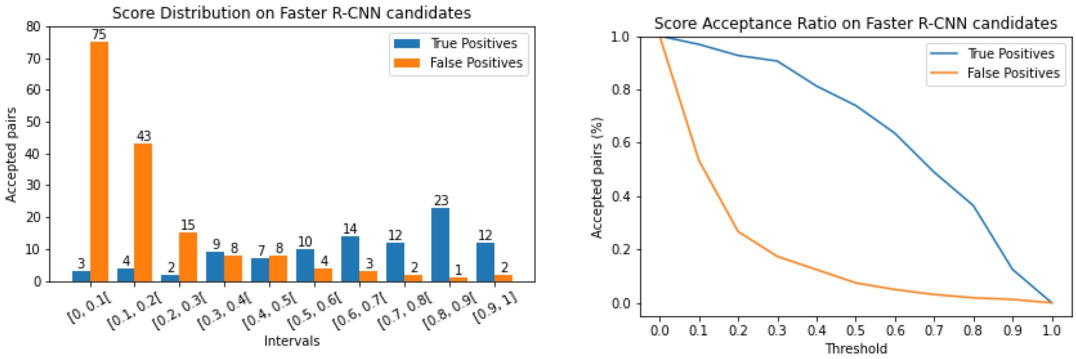
Table 4.13: Inference results for Δx Heuristic.

		ROIvROI		Faster R-CNN ROIvCandidate	
#	Threshold	Top1	Top5	Top1	Top5
1	None	21 of 106 (19.81%)	45 of 106 (42.45%)	**65 of 96 (67.71%)**	**96 of 96 (100.00%)**
2	30	29 of 106 (27.36%)	**63 of 106 (59.43%)**	37 of 96 (38.54%)	44 of 96 (45.83%)
3	60	25 of 106 (23.58%)	56 of 106 (52.83%)	52 of 96 (54.17%)	64 of 96 (66.67%)
4	90	23 of 106 (21.70%)	53 of 106 (50.00%)	61 of 96 (63.54%)	76 of 96 (79.17%)
5	40	**30 of 106 (28.30%)**	**63 of 106 (59.43%)**	49 of 96 (51.04%)	56 of 96 (58.33%)
6	80	23 of 106 (21.70%)	54 of 106 (50.94%)	61 of 96 (63.54%)	76 of 96 (79.17%)

Score

The score is obtained through the output of the Faster R-CNN. Since this model is, in essence, a detector of lesions, the score represents the model's confidence that a candidate is a true lesion. This heuristic defines that the score of the candidates must be higher than a certain threshold.

The distribution and acceptance ratio of the score are studied using the Faster R-CNN's candidates (figure 4.10).



(a) Distribution

(b) Acceptance Ratio

Figure 4.10: Score distribution among Faster R-CNN's candidates.

From the distribution, the following thresholds are defined:

1. No threshold
2. Figure 4.10b maximum gap between True Positive and False Positive rate: 0.3
3. Figure 4.10b's approximate point where the second derivative is 0: 0.5
4. Score average (average of the scores of CC and MLO views: 0.3
5. Score average: 0.5

Inference is made only for the Faster R-CNN ROIvCandidate setup and the results are displayed in table 4.14 (the best results for each metric are displayed in bold).

Table 4.14: Inference results for Score Heuristic.

		Faster R-CNN ROIvCandidate	
#	Threshold	Top1	Top5
1	None	65 of 96 (67.71%)	**96 of 96 (100.00%)**
2	0.3	**74 of 96 (77.08%)**	78 of 96 (81.25%)
3	0.5	60 of 96 (62.50%)	60 of 96 (62.50%)
4	Score avg: 0.3	69 of 96 (71.88%)	88 of 96 (91.67%)
5	Score avg: 0.5	61 of 96 (63.54%)	68 of 96 (70.83%)

Fusion

The best results from each type of heuristic are combined to achieve the best results in Faster R-CNN ROIvCandidate setup. The combinations are displayed in table 4.15.

Table 4.15: Fused heuristic combinations.

Name	Area Ratio Threshold	Δx Threshold	Score Threshold
Baseline	None	None	None
fhh1	2.0	None	0.3
fhh2	2.6	None	0.3
fhh3	1.8	None	0.3

One of the main advantages of the "hard" heuristics is the elimination of false positives. The Faster R-CNN ROIvCandidate is the only setup that has false positive detections. Since the tested heuristics in table 4.13 obtain worse results in Faster R-CNN ROIvCandidate setup than the baseline (model #1 from table 4.13), Δx is not used for fusion.

The heuristic combinations are tested in Faster R-CNN ROIvCandidate setup and their results are displayed in table 4.16 (the best results for each metric are displayed in bold).

Table 4.16: Fused heuristic combinations' Results.

	Faster R-CNN ROIvCandidate	
Threshold	Top1	Top5
Baseline	65 of 96 (67.71%)	**96 of 96 (100.00%)**
fhh1	67 of 96 (69.79%)	70 of 96 (72.92%)
fhh2	**69 of 96 (71.88%)**	72 of 96 (75.00%)
fhh3	67 of 96 (69.79%)	70 of 96 (72.92%)

Discussion

- In ROIvROI, the lesions x-axis position are very variable, since they are dependant on breast size. To have very restrictive thresholds increases significantly the results in this setup (table 4.13), possibly not only because of the position itself, but also due to the indirect exclusion of matches of breasts that are very distinct in size or by the laterality of the lesion itself.

Moreover, for the same thresholds, Faster R-CNN ROIvCandidate results are significantly worse than the other models, which can indicate that these thresholds are not for suitable for matching at a patient level analysis.

- Fused heuristics do not achieve better results than the single heuristics themselves, which indicates that the fusion of these heuristics get to restrictive, eliminating some matches that would be accepted using only one of the heuristics.

4.5.3 Soft Heuristics

Although "hard" heuristics are an effective method of decreasing false positive rate, they can also increase false negative rate. Due to this, the rules presented in section 4.5.2, instead of being used to exclude possible matches, are used to calculate α – related to the area ratio –, β – related to Δx – and s – related to scores –, which are used as multiplicative factors in the distance between feature vectors.

To obtain these parameters – (α, β, s) –, four scenarios are considered; the first two using the "Area Ratio" and Δx distributions (figures 4.5a and 4.8a), and the last two from the Faster R-CNN's scores:

- Scenario 1:

$$\alpha_1, \beta_1 = 1 - p([a,b[), i \in [a,b[, \tag{4.3}$$

where i is the $area\_ratio$ / Δx of the pair and $[a,b[$ is in one of the intervals present in figures 4.5a and 4.8a, respectively. This scenario represents a pair of candidates' probability of having its $area\_ratio$ / Δx to belong to a certain interval, given it is a correct match. The probability values are estimated, using the annotated data distribution.

- Scenario 2:

$$\alpha_2, \beta_2 = 1 - p([0,b[), i \in [a,b[, \tag{4.4}$$

where i is the $area\_ratio$ / Δx of the pair and $[a,b[$ is in one of the intervals present in figures 4.5a and 4.8a, respectively. This scenario represents a pair of candidates' probability off having its $area\_ratio$ / Δx to be lower than b, given it is a correct match. The probability values are estimated, using the annotated data distribution.

- Scenario 3:

$$s_1 = (1 - s_{cc})(1 - s_{mlo}), \tag{4.5}$$

where s_{cc} and s_{mlo} are the scores of the CC and MLO candidates, respectively.

- Scenario 4:

$$s_2 = (1 - s_{avg}), \tag{4.6}$$

where s_{avg} is the average of the CC and MLO candidates' scores.

These variables are multiplicative factors of the distance. Since the smallest distance between candidates leads to match, the variables must be the lowest when the heuristics are the most confident that the candidates are a match. In all presented scenarios, the range of the probabilities and scores is [0, 1] and the higher they are, the more confident the algorithm must be that they are a match. Since we want minimum variable values to force matches, the probabilities and scores are subtracted from 1. This way, higher probabilities and scores lead to lower variable values, which lead to smaller distances, increasing the probability of assigning the candidates as a match.

Single Soft Heuristic

Firstly, all scenarios for each variable are tested using the ROIvROI and Faster R-CNN ROIv-Candidate setups (tables 4.17, 4.18 and 4.19) (the best results for each metric are displayed in bold).

Table 4.17: Inference results for α (Area Ratio) scenarios.

	ROIvROI		Faster R-CNN ROIvCandidate	
Scenario	Top1	Top5	Top1	Top5
Baseline	21 of 106 (19.81%)	45 of 106 (42.45%)	65 of 96 (67.71%)	96 of 96 (100.00%)
1	**25 of 106 (23.58%)**	**61 of 106 (57.55%)**	77 of 96 (80.21%)	96 of 96 (100.00%)
2	21 of 106 (19.81%)	58 of 106 (54.72%)	**78 of 96 (81.25%)**	**96 of 96 (100.00%)**

Table 4.18: Inference results for $\beta(\Delta x)$ scenarios.

	ROIvROI		Faster R-CNN ROIvCandidate	
Scenario	Top1	Top5	Top1	Top5
Baseline	21 of 106 (19.81%)	45 of 106 (42.45%)	65 of 96 (67.71%)	96 of 96 (100.00%)
1	**29 of 106 (27.36%)**	58 of 106 (54.72%)	71 of 96 (73.96%)	96 of 96 (100.00%)
2	25 of 106 (23.58%)	**65 of 106 (61.32%)**	**73 of 96 (76.04%)**	**96 of 96 (100.00%)**

Table 4.19: Inference results for s (scores) scenarios.

	Faster R-CNN ROIvCandidate	
Scenario	Top1	Top5
Baseline	65 of 96 (67.71%)	96 of 96 (100.00%)
3	**82 of 96 (85.42%)**	**96 of 96 (100.00%)**
4	81 of 96 (84.38%)	96 of 96 (100.00%)

For each set of scenarios, it is considered that the best result is the one that had the highest accuracy in Faster R-CNN ROIvCandidate setup, since it is the more realistic setup.

Fused Soft Heuristic

From each variable, the scenario that presents the best results is used for fusion of the "soft" heuristics, resulting in the combinations presented in table 4.20 (the best results for each metric are displayed in bold).

Table 4.20: Fused soft heuristic combinations.

Name	α scenario	β scenario	s scenario
Baseline	None	None	None
fsh1	α_2	None	s_1
fsh2	α_2	β_2	None
fsh3	None	β_2	s_1
fsh4	α_2	β_2	s_1

Each combination is tested on ROIvROI (if s is not involved) and Faster R-CNN ROIvCandidate setups. The results are presented in table 4.21.

Table 4.21: Inference results Fused Soft Heuristics combinations.

	ROIvROI		Faster R-CNN ROIvCandidate	
Name	Top1	Top5	Top1	Top5
Baseline	21 of 106 (19.81%)	45 of 106 (42.45%)	65 of 96 (67.71%)	96 of 96 (100.00%)
fsh1	**-**	**-**	**83 of 96 (86.46%)**	**96 of 96 (100.00%)**
fsh2	23 of 106 (21.70%)	63 of 106 (59.43%)	77 of 96 (80.21%)	96 of 96 (100.00%)
fsh3	-	-	76 of 96 (79.17%)	96 of 96 (100.00%)
fsh4	-	-	81 of 96 (84.38%)	96 of 96 (100.00%)

Discussion

- The use of "soft" heuristics, generally, improved results in all scenarios, with all variables.
- The fusion of different heuristics allowed the addition of some complementary information, since the best accuracy, while using "soft" heuristics, consists in fused heuristics ($fsh1$ in table 4.21).
- Like in "hard" heuristics' results, Δx achieved the best results in ROIvROI setup. This strengthens the hypothesis that the use of this heuristics in this setup is influenced by the laterality and size of the breast. However, unlike "hard" heuristics, Faster R-CNN ROIvCandidates were also improved by this heuristic, probably due to the non-exclusion of lesions that permitted a better integration of the heuristics in the algorithm.
- The possibility of using "hard" or "soft" heuristics can be valuable, pending on the candidates generator:

  1. If the object detector is not good, it generates many false positive candidates, variable scores and lowly accurate bounding box regressions on true lesions. In this case, the system can have low confidence in the object detector, using "hard" heuristics, since they can improve accuracy by eliminating false positive lesions for matching. However, "hard" heuristics can also exclude true positive matches.

2. If the object detector is good, it generates few false positive candidates, high scores and accurate bounding box regressions on true lesions. In this case, the system can have high confidence in the object detector, using "soft" heuristics, since they can properly match lesions, without the risk of excluding true positive matches. However, "soft" heuristics can generate false positive matches (by using false positive generated candidates), since they can not exclude matches.

4.6 Final System

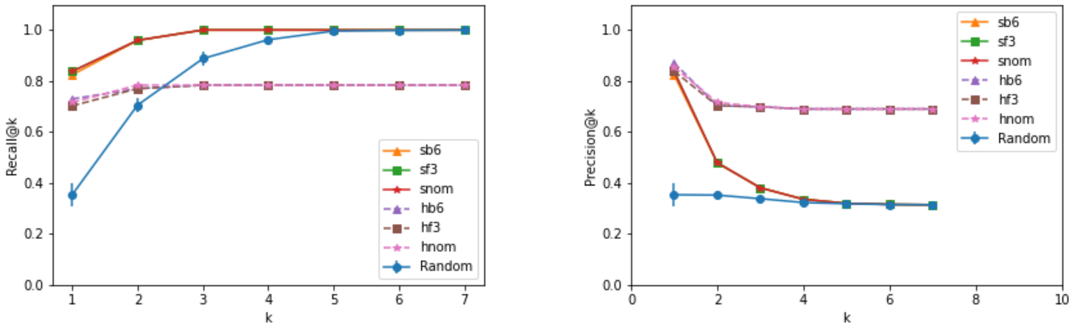
By selecting the feature extractors and heuristics that achieve the best performance, six final proposals are made (divided by two sets, using "soft" and "hard" heuristics), to find the best combination. The selected feature extractors are models *b3_nom*, *NOM* (10 epochs) and *f3*, from tables 4.1, 4.10 and 4.6, respectively.

Table 4.22: Final System combinations.

Name	Feature Extractor	Hard Heuristic	Soft Heuristic
sb3_nom	b3_nom	None	fsh1
sf3	f3	None	fsh1
snom10	NOM 10 epochs	None	fsh1
hb3_nom	b3_nom	s_1	None
hf3	f3	s_1	None
hnom10	NOM 10 epochs	s_1	None

Additionally another algorithm, which matches candidates randomly, is also evaluated, to compare performance between this model and the proposed ones.

To evaluate performance, recall@k and precision@k metrics are measured on Faster R-CNN ROIvCandidate setup. The random model evaluation is made through the mean and standard deviation of 10 experiences, since it generates different values at each iteration. The evaluation results for all models is displayed in figures 4.11.

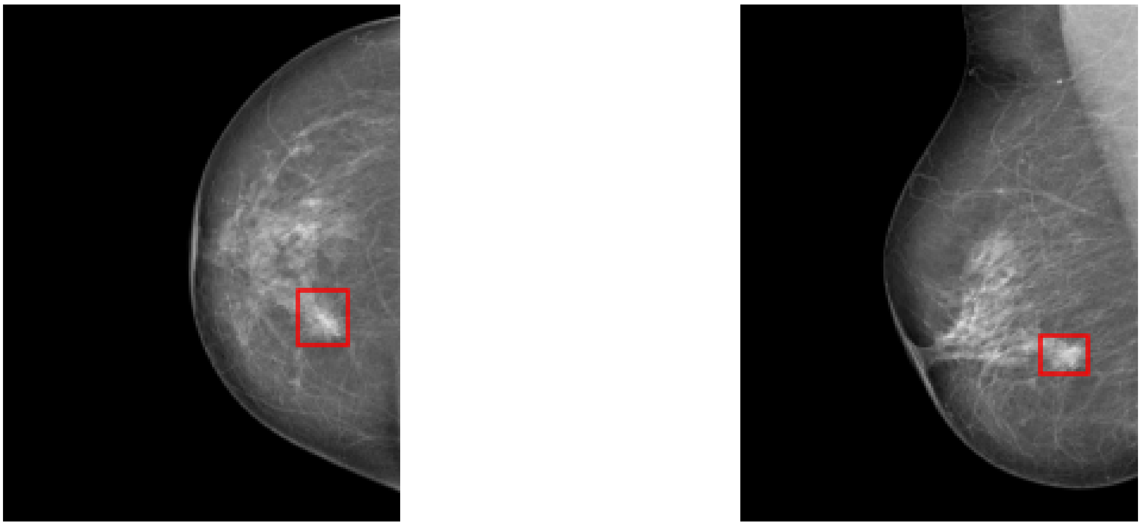


(a) Recall@k

(b) Precision@k

Figure 4.11: Evaluation of the proposed algorithms.

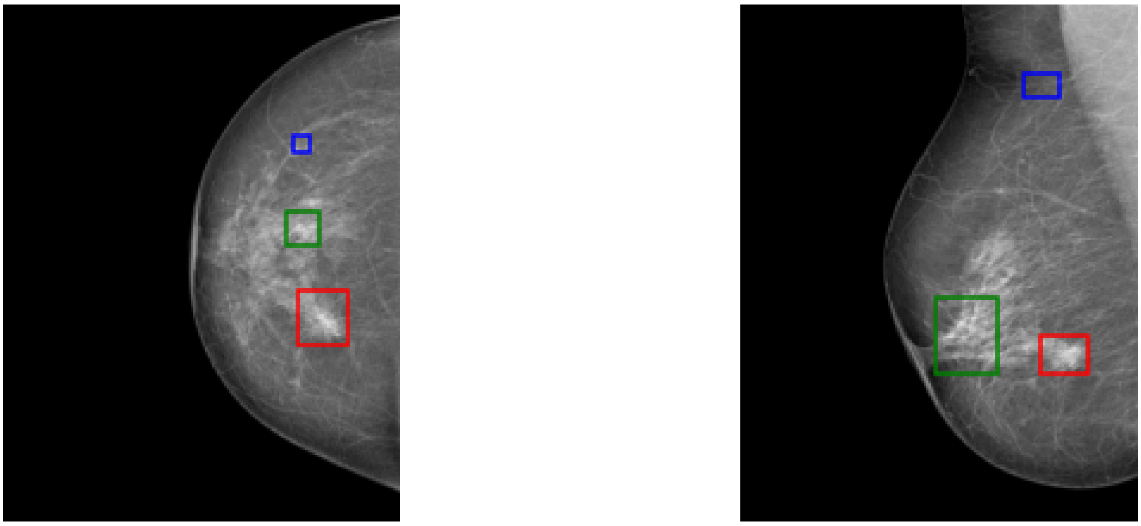
At last, the final system is tested, by generating and matching candidates, using both views of a breast. Two experiments are made, using the *snom10* and *hnom10* combinations (table 4.22). Figures 4.12 and 4.13 represents the system's results using the "hard" and "soft" heuristics, respectively.



(a) CC view

(b) MLO view

Figure 4.12: Matching system results, using "hard" heuristics. Boxes of the same color represent matches. Red boxes represent a true positive match.



(a) CC view

(b) MLO view

Figure 4.13: Matching system results, using "soft" heuristics. Boxes of the same color represent matches. Red boxes represent a true positive match.

4.6.1 Discussion

- Both "hard" and "soft" heuristics proposals achieved satisfactory results since they achieved a matching accuracy within the range of 70%-85%, for the different feature extractors hypothesis. Moreover, this accuracy is way higher than random chance probability, represented by the "Random" curve (figure 4.11a)

- Recall@k of "hard" heuristic combinations (figure 4.11a) achieved worse results than the random chance matching for $k > 2$, since the algorithms exclude some true positive matches, limiting the recall score. However, "hard" heuristics also significantly improves precision@k scores (figure 4.11b), since it removes a lot of the generated false positive lesions, which makes the precision for $k > 1$ much higher than their "soft" heuristics and random match counterparts.

- "Soft" heuristics is shown to be a relevant strategy, since it achieves better accuracy with any feature extractor than their "hard" heuristics counterparts, without the cost of excluding possible true positive lesions.

- The fact that the triplet loss model is competitive with the other feature extractors in both heuristics scenario is very relevant, since it allows to have a matching option independent from the candidates detector's backbone - e.g., the system has its Faster R-CNN replaced by a state-of-the-art object detector that makes better candidates predictions. There is no guaranty that the new backbone will achieve the same matching results as the current one, since it is trained for the lesion detection task.

- As mentioned in the heuristics discussion, the use of each heuristics scenario highly depends on the candidates detector (good detector - "soft" heuristics; bad detector - "hard" heuristics). However, since false negatives for breast cancer diagnosis are not desirable, the "soft" heuristics' models seem the more appealing systems, due to the higher accuracy and to the non-exclusion of matches, which reduce the number of false negatives.

Chapter 5

Conclusions

Of all cancer diseases, breast cancer is the most lethal among women. It has been shown that breast cancer screening programs can decrease mortality, since early detection increases the chances of survival. Usually, a pair of radiologists interpret the screening mammograms, however the process is long and exhausting. This has encouraged the development of CADx systems to replace the second radiologist making a better use of human-experts' time. But CADx systems are associated with high false positive rates, since most of them only use one view (CC or MLO) of the screening mammogram. Radiologist, on the other hand, use both views; frequently reasoning about the diagnosis by noticing differences between the two views.

The proposed system uses both views of the screening mammograms to detect lesions and compare them across views to detect matches. It presents two possible solutions for matching, using hard or soft heuristics.

Both solutions achieve >70% matching accuracy, which is way higher than chance level (30%-40%). Thus the present work is well suited to expand other CADx systems, leading to improved accuracy. Moreover, by matching lesions, our system enables more intelligent reasoning strategies for CADx systems.

However, the system has some limitations. It is built to match detected lesions on two views of the breast and returns only pairs of lesions, not considering the possibility of existing lesions that are visible in only one view. This work is also only focused on matching masses and does not consider micro-calcifications, which are important findings in the screening mammogram. Finally, the system is not able to classify the detected lesions as malignant or benign.

Due to these limitations, this system, as is, could not act as CADx system in clinical environment. However, it can be used to improve other CADx systems (most of the lesions are visible in both CC and MLO views) - e.g. as a multiplicative factor to decision making, similar to the soft heuristics in the present work.

5.1 Future Work

To make this system suitable for clinical environment we present some topics for future work:

- **State-of-the-art object detector** - to have a more robust lesion detector can decrease the number of generated false positives, increase the viability of scores and bounding box regressions. All these characteristics have a key contribution for matching in the proposed system. The better the object detector becomes, the better the matching performance can be.

- **Classify lesion matches** - the system could use the classification labels (benign or malignant), obtained through the object detector, as an heuristic. If the labels of two candidates are distinct, matching could be excluded – "hard" heuristics – or restrained by the attribution of large multiplicative factors – soft Heuristics.

- **Expand the system to detect and classify lesions that are visible in only one view** - as mentioned in discussion, this system can only detect matched lesions. To have a system that detects and classifies lesions in single view, while considering possible candidates matches in both views, can make it more suitable for being used in a clinical environment.

- **Fine-Tune system with more data** - this system is highly dependant in CBIS-DDSM and INbreast data. However, screening mammograms' images are highly dependant on the equipment that generates those images. Since most of the algorithm is developed in fairly small ammounts of data, the system is likely over-fitted to the INbreast and CBIS-DDSM datasets.

References

[1] Lei Zhen and A K Chan. An artificial intelligent algorithm for tumor detection in screening mammogram. *IEEE Transactions on Medical Imaging*, 20(7):559–567, 2001. doi:10.1109/42.932741.

[2] N Dhungel, G Carneiro, and A P Bradley. Automated Mass Detection in Mammograms Using Cascaded Deep Learning and Random Forests. In *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2015. doi:10.1109/DICTA.2015.7371234.

[3] Dezső Ribli, Anna Horváth, Zsuzsa Unger, Péter Pollner, and István Csabai. Detecting and classifying lesions in mammograms with Deep Learning. *Scientific Reports*, 8(1):4165, 2018. URL: https://doi.org/10.1038/s41598-018-22437-z, doi:10.1038/s41598-018-22437-z.

[4] Li Shen, Laurie R Margolies, Joseph H Rothstein, Eugene Fluder, Russell McBride, and Weiva Sieh. Deep Learning to Improve Breast Cancer Detection on Screening Mammography. *Scientific Reports*, 9(1):12495, 2019. URL: https://doi.org/10.1038/s41598-019-48995-4, doi:10.1038/s41598-019-48995-4.

[5] Gustavo Carneiro, Jacinto Nascimento, and Andrew P Bradley. Unregistered Multiview Mammogram Analysis with Pre-trained Deep Learning Models BT - Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pages 652–660, Cham, 2015. Springer International Publishing.

[6] Sami Dhahbi, Walid Barhoumi, and Ezzeddine Zagrouba. Breast cancer diagnosis in digitized mammograms using curvelet moments. *Computers in Biology and Medicine*, 64:79–90, 2015. URL: http://www.sciencedirect.com/science/article/pii/S0010482515002206, doi:https://doi.org/10.1016/j.compbiomed.2015.06.012.

[7] SuperDataScience. The Ultimate Guide to Artificial Neural Networks (ANN), 2018. URL: https://www.superdatascience.com/blogs/the-ultimate-guide-to-artificial-neural-networks-ann.

[8] Sagar Sharma. Activation Functions in Neural Networks, 2017. URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[9] Sumit Saha. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, 2018. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b11

[10] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, 2018. URL: https://doi.org/10.1007/s13244-018-0639-9, doi:10.1007/s13244-018-0639-9.

[11] Muneeb ul Hassan. VGG16 – Convolutional Network for Classification and Detection. 2018. URL: https://neurohive.io/en/popular-networks/vgg16/.

[12] S Ren, K He, R Girshick, and J Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. doi:10.1109/TPAMI.2016.2577031.

[13] J Wang, Yang Song, Thomas Leung, C Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Y Wu. Learning Fine-Grained Image Similarity with Deep Ranking. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[14] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 68(6):394–424, nov 2018. URL: https://doi.org/10.3322/caac.21492, doi:https://doi.org/10.3322/caac.21492.

[15] H Nasir Khan, A R Shahid, B Raza, A H Dar, and H Alquhayz. Multi-View Feature Fusion Based Four Views Model for Mammogram Classification Using Convolutional Neural Network. *IEEE Access*, 7:165724–165733, 2019. doi:10.1109/ACCESS.2019.2953318.

[16] American Cancer Society. Mammogram Basics, 2020. URL: https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/mammogram-basics.html.

[17] Ioannis Sechopoulos, Jonas Teuwen, and Ritse Mann. Artificial intelligence for breast cancer detection in mammography and digital breast tomosynthesis: State of the art. *Seminars in Cancer Biology*, (May), 2020. doi:10.1016/j.semcancer.2020.06.002.

[18] Amira Jouirou, Abir Baâzaoui, and Walid Barhoumi. Multi-view information fusion in mammograms: A comprehensive overview. *Information Fusion*, 52(March):308–321, 2019. URL: https://doi.org/10.1016/j.inffus.2019.05.001, doi:10.1016/j.inffus.2019.05.001.

[19] Zhiqiong Wang, Qixun Qu, Ge Yu, and Yan Kang. Breast tumor detection in double views mammography based on extreme learning machine. *Neural Computing and Applications*, 27:227–240, jan 2016. doi:10.1007/s00521-014-1764-0.

[20] F Lefebvre, H Benali, R Gilles, E Kahn, and R Di Paola. A fractal approach to the segmentation of microcalcifications in digital mammograms. *Medical physics*, 22(4):381–390, apr 1995. doi:10.1118/1.597473.

[21] H D Li, M Kallergi, L P Clarke, V K Jain, and R A Clark. Markov random field for tumor detection in digital mammography. *IEEE Transactions on Medical Imaging*, 14(3):565–576, 1995. doi:10.1109/42.414622.

[22] Patricia McKenzie and Michael Alder. Initializing the EM algorithm for use in Gaussian mixture modelling. In Edzard S GELSEMA, Laveen S B T Machine Intelligence KANAL, and Pattern Recognition, editors, *Pattern Recognition in Practice IV*, volume 16, pages 91–105. North-Holland, 1994. URL: http://www.sciencedirect.com/science/article/pii/B9780444818928500134, doi:https://doi.org/10.1016/B978-0-444-81892-8.50013-4.

[23] Danilo Cesar Pereira, Rodrigo Pereira Ramos, and Marcelo Zanchetta do Nascimento. Segmentation and detection of breast cancer in mammograms combining wavelet analysis and genetic algorithm. *Computer Methods and Programs in Biomedicine*, 114(1):88–101, 2014. URL: http://www.sciencedirect.com/science/article/pii/S0169260714000261, doi:https://doi.org/10.1016/j.cmpb.2014.01.014.

[24] Kamal Hammouche, Moussa Diaf, and Patrick Siarry. A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision and Image Understanding*, 109(2):163–175, 2008. URL: http://www.sciencedirect.com/science/article/pii/S1077314207001336, doi:https://doi.org/10.1016/j.cviu.2007.09.001.

[25] N Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi:10.1109/TSMC.1979.4310076.

[26] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi:10.1126/science.1127647.

[27] A P Dempster, N M Laird, and D B Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, sep 1977. URL: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x, doi:https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[28] R Girshick, J Donahue, T Darrell, and J Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. doi:10.1109/CVPR.2014.81.

[29] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. URL: https://doi.org/10.1023/A:1010933404324, doi:10.1023/A:1010933404324.

[30] Krzysztof Geras, Stacey Wolfson, S Kim, Linda Moy, and Kyunghyun Cho. High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Networks. mar 2017.

[31] Daniel Levy and Arzav Jain. Breast Mass Classification from Mammograms using Deep Convolutional Neural Networks. dec 2016.

[32] Thijs Kooi, Geert Litjens, Bram van Ginneken, Albert Gubern-Mérida, Clara I Sánchez, Ritse Mann, Ard den Heeten, and Nico Karssemeijer. Large scale deep learning for computer aided detection of mammographic lesions. *Medical Image Analysis*, 35:303–312, 2017. URL: http://www.sciencedirect.com/science/article/pii/S1361841516301244, doi:https://doi.org/10.1016/j.media.2016.07.007.

[33] Jinhua Wang, Xi Yang, Hongmin Cai, Wanchang Tan, Cangzheng Jin, and Li Li. Discrimination of Breast Cancer with Microcalcifications on Mammography by Deep Learning. *Scientific Reports*, 6(1):27327, 2016. URL: https://doi.org/10.1038/srep27327, doi:10.1038/srep27327.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[35] C Szegedy, Wei Liu, Yangqing Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi:10.1109/CVPR.2015.7298594.

[36] J R R Uijlings, K E A van de Sande, T Gevers, and A W M Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. URL: https://doi.org/10.1007/s11263-013-0620-5, doi:10.1007/s11263-013-0620-5.

[37] R Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. doi:10.1109/ICCV.2015.169.

[38] K He, G Gkioxari, P Dollár, and R Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi:10.1109/ICCV.2017.322.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context BT - Computer Vision – ECCV 2014. pages 740–755, Cham, 2014. Springer International Publishing.

[40] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1, jul 2018. doi:10.1109/TPAMI.2018.2858826.

[41] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501, 2006. URL: https://www.sciencedirect.com/science/article/pii/S0925231206000385, doi:https://doi.org/10.1016/j.neucom.2005.12.126.

[42] Sami Dhahbi, Walid Barhoumi, and Ezzeddine Zagrouba. Multi-view score fusion for content-based mammogram retrieval. In *Eighth International Conference on Machine Vision (ICMV 2015)*, volume 9875, page 987515, dec 2015. URL: https://doi.org/10.1117/12.2228614, doi:10.1117/12.2228614.

[43] Abir Baâzaoui, Marwa Abderrahim, and Walid Barhoumi. Dynamic distance learning for joint assessment of visual and semantic similarities within the framework of medical image retrieval. *Computers in Biology and Medicine*, 122:103833, 2020. URL: http://www.sciencedirect.com/science/article/pii/S0010482520301967, doi:https://doi.org/10.1016/j.compbiomed.2020.103833.

[44] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6, 2015.

[45] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*, sep 2014.

[46] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, 4(1):170177, 2017. URL: https://doi.org/10.1038/sdata.2017.177, doi:10.1038/sdata.2017.177.

[47] Inês C. Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria João Cardoso, and Jaime S. Cardoso. INbreast: Toward a Full-field Digital Mammographic Database. *Academic Radiology*, 19(2):236–248, 2012. doi:10.1016/j.acra.2011.09.014.

[48] Dezső Ribli. Computer aided detection with Faster-RCNN, 2018. URL: https://github.com/riblidezso/frcnn{_}cad.

[49] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[50] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. *Deep Metric Learning with Angular Loss*. oct 2017. doi:10.1109/ICCV.2017.283.

[51] M Heath, Kevin Bowyer, D Kopans, R Moore, and P Kegelmeyer. The Digital Database for Screening Mammography. *Proceedings of the Fourth International Workshop on Digital Mammography*, jan 2000. doi:10.1007/978-94-011-5318-8_75.

[52] American Cancer Society. Understanding Your Mammogram Report, 2019. URL: https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/understanding-your-mammogram-report.html.

[53] American College of Radiologists. ACR, 2020. URL: https://www.acraccreditation.org/mammography-saves-lives/guidelines.

[54] Google LLC. Google Colab, 2019. URL: https://colab.research.google.com/.