Advanced Image Analysis - Machine and Deep Learning (2022), 1–30

ARTICLE

Calcification detection in mammograms: Project report

Alejandro Cortina Uribe, Joaquin Oscar Seia, and Vladyslav Zalevskyi

Abstract

Due to the high incidence of breast cancer amongst women around the world, the development of CADe systems that can help radiologists to early and correctly diagnose the illness becomes crucial. In this report we present the development of three automatic algorithms to detect microcalcifications in digital mammograms, lesions that are important to identify and characterize in the early stages of breast cancer. We explored advanced image analysis algorithms, machine learning models and deep learning models, obtaining relevant results in each of the implemented pipelines. The first pipeline, which consisted in a candidate proposals generation with a grayscale morphology algorithm, followed by a taylored feature extraction and a SVM-cascaded classification algorithm achieved an $AUFROC_{50}$ of 0.7569 ± 0.0184. The second one involved a sliding window patch-wise classification model using an ad-hoc ResNet-based CNN to generate a probability map, from which detections are obtained and reached an $AUFROC_{50}$ of 0.6787 ± 0.0188. The last one implied a sliding window patch-wise microcalcification detection using Faster-R-CNN, showed $AUFROC_{50}$ of 0.8092 ± 0.0141. We confirmed the ability of deep convolutional neural networks to fastly extract meaningful features, in contrast with the highly sensitive and detailed handcrafting of candidate and feature extraction methods from the first pipeline.

Keywords: Calcification detection; Mammography; Machine learning; Deep learning; Advanced image analysis

1. Introduction

1.1 Problem

Breast cancer has the highest incidence rate amongst all cancers in women around the world (Sung et al. 2021). Different studies have shown that screening programs can result in a significant reduction -between 20 to 30%- in the mortality rates associated with this disease (O'Grady and Morgan 2018, Monticciolo 2020, Dibden et al. 2020). There are many features that can be assessed in a mammography image in order to suspect the presence of breast cancer, one of the most common ones is the presence of very small deposits of calcium called microcalcifications. In the permanent examination of the benefits of screening programs, some studies have found that between 30 to 50% of non palpable tumors could be found in screening mammography only by the presence of microcalcifications (Venkatesan et al. 2009, Gulsun, Demirkazik, and Ariyurek 2003).

Microcalcifications might be one of the first detectable signs of malignancy, been associated with pre-malignant and proliferative stages of breast cancer, therefore detecting them on time can allow the prevention of a highly bad prognosis invasive stage (Logullo et al. 2022, Mordang et al. 2018). However, the relevance of accurately detecting this lesions is even higher than just determining their presence. Their shape and distribution pattern inside the breast constitute important biomarkers linked to the malignancy, prognosis, genetic and molecular characteristics of the tumor (Tot et al. 2021).

Nevertheless, interpreting mammograms represents a difficult task even for experienced radiologists. Assuring the presence and nature of microcalcifications might even need the use of extra magnification techniques or modalities (Logullo et al. 2022). In this scenario, it results of special interest to develop Computed Aided Detection (CADe) systems that precisely identify this lesions. Several attempts have been made to obtain well performing automatic systems to aid radiologist in their labour (Savelli et al. 2020). However, there is still plenty of room to improve these algorithms performances, which in general produce several more false detections per true ones than their humans counterparts. Further work needs to be done in order to increase the precision of the systems without the avoiding radiologist having to check many false positive predictions, but without loosing their sensitivity in order to contribute to the early detection of the pathology.

Trying to understand the intrinsic challenges that this medical image modality and problem might present, this report summarizes the work done in order to implement a fully automatic microcalcification detection algorithm. To achieve this goal, we studied and compared three different approaches. The first one involved the use of advanced image analysis and machine learning classifiers, whereas the other two were deep learning based. Across this study, we present the dataset used, provide detailed description of each of the three pipelines to finally compare the results obtained with them.

1.2 Dataset description

We used the INbreast dataset (Moreira et al. 2012) which includes 410 full-field digital mammograms with pixel size of 70 μ and 14-bit contrast resolution. The image dimensions are 3328x4084 or 2560x3328 pixels, depending on the compression plate used in the study.

The dataset includes mammograms with masses, microcalcifications, clusters of microcalcifications, architectural distortions, and asymmetries. Each image in the dataset can contain either single or multiple lesions of different kinds. Images are annotated by distinguishing different regions of interest (ROI): microcalcifications, masses, clusters, spiculated regions, asymmetries, and distortions. Microcalcifications are the most prevalent ROIs around all images (301 out of 410 images), and in most of the cases (271 out of 410 images) one image contains more than one MC.

Around 16% of all calcification ROIs are present in extremely dense breast tissue mammograms (level 4 of ACR), making them very difficult to detect. Calcifications smaller than 1 mm in diameter have higher chances of being malignant (Dähnert 2011) and therefore they became the primarily detection target of our detection systems. In the INbreast dataset around 87% of all labeled calcifications are linked to malignant cancer. In this way, the algorithms wedeveloped using this database, even though we did not concentrate only on malignant lesions, can be thought to be well conditioned to detect these clinically critical lesions.

The problem and the database implied specific challenges that were needed to be addressed by the developed methods. The first one being high imbalance of data, as detection of MCs involves finding very small lesions/structures (14 pixels in diameter) in a very large image containing mostly normal breast tissue. In addition, we dealt with three different types of ground truths for MC: detailed contour segmentation, an ellipse enclosing an entire cluster of MCs or point (pixel) annotations; the latest being around 40% of all available annotations. This lack of regularity in the labeling increased the difficulty of the problem, reducing the information of most of the ground truth detections to only location without any data on its extension or shape.

2. Methods

As stated before, in our work we developed three different detection pipelines (Figure 1). The first one involved the use of *advanced image analysis* techniques as a first stage in order to generate MC candidate proposals, followed by the use of *machine learning* models to classify the candidates in true positive (TP) or false positive (FP) to achieve a false positive reduction. The second and third pipelines

consisted in the use of *deep convolutional neural networks* to detect MCs. The second one involved the use of patch-wise *classification* models in a sliding window fashion to generate saliency maps that we later thresholded to obtain detections. The third pipeline used a patch-wise *detection* model (Faster-R-CNN) run in a sliding window fashion over the full mammogram. The complete code of this project is publicly available at https://github.com/joaco18/calc-det.

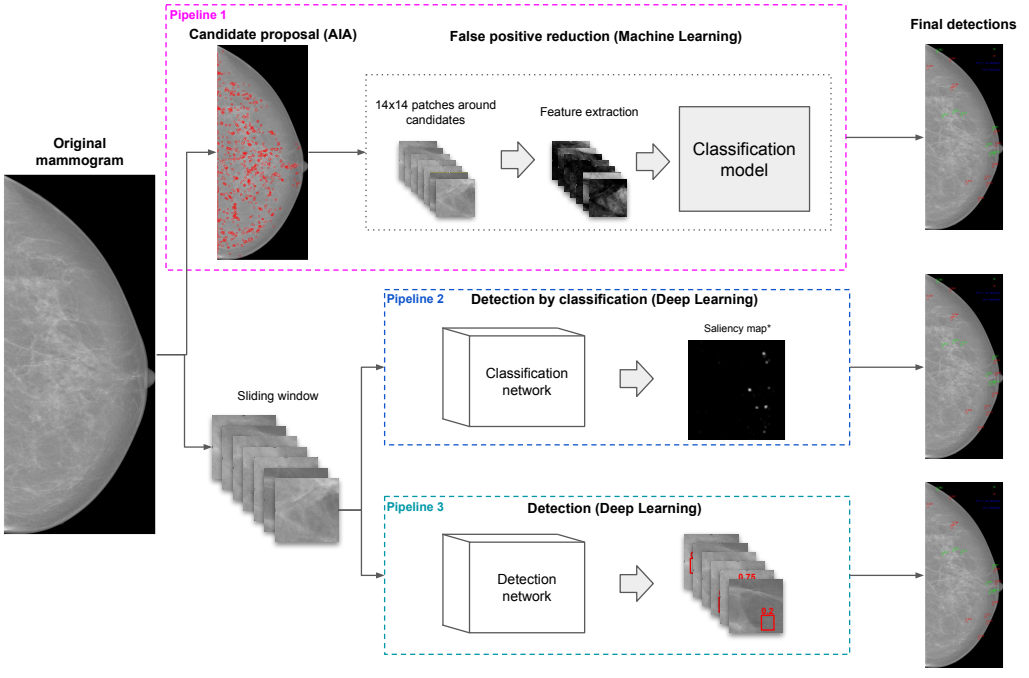

Figure 1. General diagram showing the three developed pipelines.

As an initial common step applied for all pipelines, we preprocessed the original images from the dataset by cropping only the breast region and flipping the right ones to the left orientation. With the first action we reduced the size of the data in disk, in memory and the extension of the region to process. The second procedure was mainly meant to increase the regularity of the data fed to the algorithms.

For the ground truths, we discarded all lesion labels except those from MCs and clusters of MCs. We used the original database labels and not the dilated mask-version provided in the project presentation.

Furthermore, not all labeled lesions were used to evaluate the pipelines. In the case of clustered lesions, the ellipse labeling represented a problem in assessing models' performance. The algorithms could have correctly spotted several MCs that were later unfairly counted as a single lesion, and on the contrary, false positive detections generated by the algorithms in normal tissue inside the ellipse were wrongly considered a true positive, thus difficulting the training of learning based approaches. In addition, after inspection of the labeled MCs present in INBreast, we found out that the MCs with diameter larger than 1mm were mainly calcified ducts and benign pop-corn like MCs. This represented a problem for our image analysis methods that were constructed based on the assumption that MCs (or at least the most clinically relevant ones that we are interested in) have 1mm in diameter or less, and therefore they were not able to spot larger lesions.

To handle the MC clusters, we explored a weak labeling approach based on label refinement with image processing; however, it was later discarded due to the inaccuracy and possible bias induced

towards one of the compared methods. In the end, to deal with these two labeling challenges, we decided to ignore these labels for the performance evaluation, by not considering the regions that included those lesions. In other words, if the generated detections matched a ground truth label coming from these problematic cases, they were not counted as a true positive. In the same sense, if no lesion was spotted on the location of these ground truths, it was not counted as false negative. By doing this, we ignored labeled information but gained certainty on the used ground truth which allowed us to have a reliable performance evaluation.

In order to train and evaluate the models properly, the complete set of images was splitted into train, validation, and test sets, containing 33%, 17% and 50% of the cases respectively. We did the splitting at a patient level using the case ID ensuring no data leakage occurred between the sets. We obtained the partitions in a stratified fashion considering the proportion of cases with MCs vs cases without MCs. The cases without MCs, from now on also referred to as *MC-negative* cases, were specifically identified since they played a key role in the detection metrics used to evaluate the methods. Given that not all MCs are labeled in many *MC-positive* cases present in INBreast, the last set of images was highly relevant since they served as a 'control set' in which to compute the number of FP per image generated by the methods.

The partitioning resulted in a train set containing 40 cases (148 images), a validation set containing 15 cases (62 images), and a test set including 53 cases (200 images).

2.1 First pipeline: Advanced Image Analysis and Machine Learning

2.1.1 Advanced image analysis for candidate proposal

The main objective of this stage was to obtain a set of MC candidates, each described as a blob with a center (two coordinates) and a radius, that led to a high sensitivity value at the lowest possible number of false positives per image. To do so, we implemented three candidate proposal techniques to be applied on full images that were based on: Hough Transform, Hessian of Difference of Gaussians (HDoG), and Grayscale Morphology (GM). We chose all methods' operating parameters by considering the trade-off between sensitivity and number of FP per image.

To compensate for the fact that the images did not span the whole range defined by the uint16 original data type, all techniques required the mammograms intensity to be normalized to the range [0,1] and casted to float32.

2.1.2 Hough transform (HT)

The Hough circle transform is a particularly useful technique that localizes circular structures in an image. Though microcalcifications are not generally described as perfect circular structures, malignant MCs, which are the ones we were mostly interested in, are known to have a round shape and with a proper preprocessing, they can be successfully detected.

We followed the general pipeline described in (Basile et al. 2019), while changing certain steps since they were not clearly described or reasoned in the original paper (Figure 2). First, we preprocessed the mammograms to remove noise and enhance small, contrasted structures. For that, we applied a dehazing method which used dark channel prior, as it has been shown to enhance local contrast of MCs in (Bria et al. 2017). After that, we used a rolling ball background subtraction algorithm to even further enhance contrast of small, bright MCs, by removing smooth continuous backgrounds from images. Following that up, we used a sequence of Sobel - Gaussian - Sobel filtering that highlights the boundaries of enhanced MCs and ensures that we preserve the edges of the structures of interest without amplifying the noise. Finally, we used an eroded version of the breast mask to filter false detections originated in the high contrast region between breast and the image background (breast border). As stated in Basile et al. 2019 there are only a few MCs in this region of the breast, and it is safe to ignore it. In this way, we obtained an isolation of possible MCs with delineated circular contours.

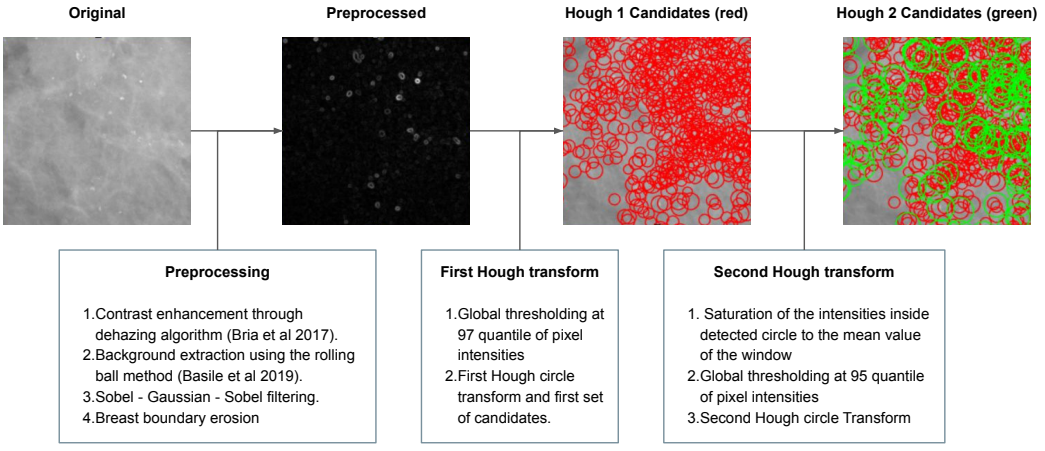

Figure 2. Hough transform candidate proposal pipeline. Plotted circles are enlarged by 10 pixels to be better appreciated.

Microcalcifications usually appear as relatively bright lesions, compared to the surrounding breast tissue or masses. However mammograms also contain large quantities of fibroglandular tissue, which remained very bright even after the first preprocessing phase, thus making small microcalcifications poorly visible (Morris 2003).

For this reason, after preprocessing the image we applied the Hough circle transform in two consecutive steps. We obtained the first set of circle candidates over the binarized image using a global thresholding of the 97 quantile of the preprocessed image pixel intensities, thus removing the dispersed low-intensity structures and leaving mostly MCs edges. Following Basile et al. 2019, in order to reduce the number of false positives candidates, we apply a second transform in a patch around each candidate from the first step. We saturated the intensities of pixels in the 200x200 patch around each candidate from the first transform to the mean value of the patch, and then we used a second, less conservative thresholding of 95 quantiles to further eliminate low contrast structures. At last, the second Hough circle transform was consequently applied on the processed patches to obtain a final set of candidates.

We conducted experiments to obtain the most suitable operating parameters and the full list of them can be found in Appendix 1.

2.1.3 Hessian of Difference of Gaussian (HDoG)

Following the general pipeline described in Marasinou et al. 2021, we implemented a blob segmentation algorithm using the Hessian matrix of a difference of Gaussians (DoG) space (Figure 3). First, we computed the DoG space by subtracting consecutive Gaussian blurred versions of the mammogram using increasing sigma smoothing kernels. This resulted in a 3D array containing different high pass filtered versions of the image at different scales, that highlighted the borders of MCs. Then, we found peak local maxima using 3x3x3 local windows around each point. As a next step, we filtered the overlapping blobs and the ones with centers closer than a threshold, thus getting a first set of candidates. After that, we computed the Hessian matrix of each scale in the DoG space and applied a final Hessian filtering condition over all preselected candidates. The condition aims to describe MCs geometrical structure, being circular or tubular, through the eigenvalues of the Hessian matrix. The first condition Marasinou et al. 2021 uses the trace and determinant, and follows (Eq. 1):

$$(tr(H) < 0) \wedge \left(det(H) < 0 \vee \frac{|det(H)|}{tr(H)^2} \leq h_{thr} \right), \tag{1}$$

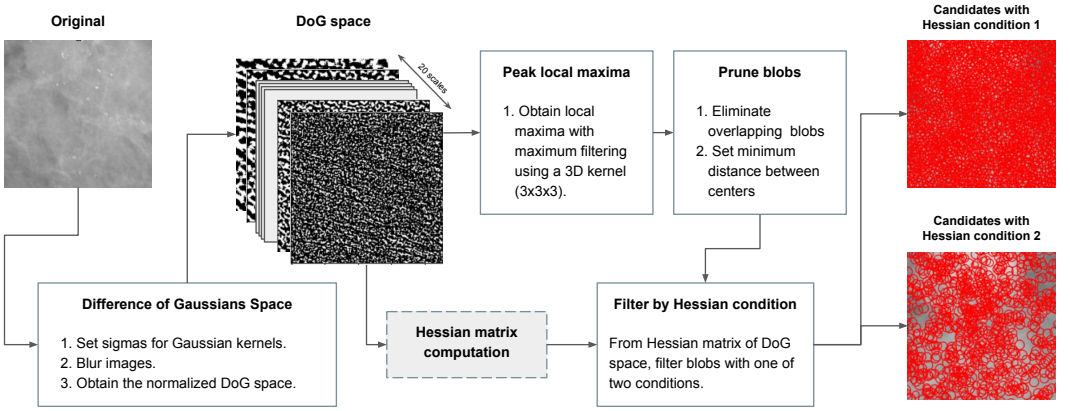

Figure 3. Hessian of Difference of Gaussian candidate proposal pipeline. Plotted circles are enlarged by 10 pixels to be better appreciated.

where H is the 3D array of Hessians of the DoG space of the image.

Inspired by Muthuvel, Thangaraju, and Chinnasamy 2017, we explored an alternative condition using directly the eigenvalues of the Hessian matrix. Given the presence of a MC circular structure in the image, the Hessian of the DoG space will show large negative values for the two eigenvalues on consecutive scales. On this basis, we applied a threshold condition on the multiscale product of eigenvalues, further increasing the signal where the MC is present (Eq. 2 and 3) .

$$P_k^j(x, y) \geq T_k, \tag{2}$$

$$\text{where } T_k = \frac{max(P_k^j)}{divider}, P_k^j = \prod_{m=j}^{j+1} \lambda_k^m, \tag{3}$$

with j being a DoG space scale and λ_1, λ_2 are the the first and second eigenvalues of the Hessian matrix.

After several experiments, we decided to use the second condition since it provided many less false positives at the same sensitivity values.

2.1.4 Grayscale morphology (GM)

Grayscale mathematical morphology is a powerful image analysis set of methods that has been proven to be effective in the enhancement of certain image aspects (Diaz-Huerta, Felipe-Riveron, and Montaño-Zetina 2014). It further extends binary morphology by considering an image to be a non-binary function f, that represents the pixel intensities at given spatial locations. For a given intensity function f and a function b it defines few important operations:

Dilation:

$$[f \oplus b](x, y) = max_{(s,t)\in b}[f(x + s, y + t)] \tag{4}$$

Erosion:

$$[f \ominus b](x, y) = min_{(s,t)\in b}[f(x + s, y + t)] \tag{5}$$

Opening:

$$[f \circ b](x, y) = (f \ominus b) \oplus b \tag{6}$$

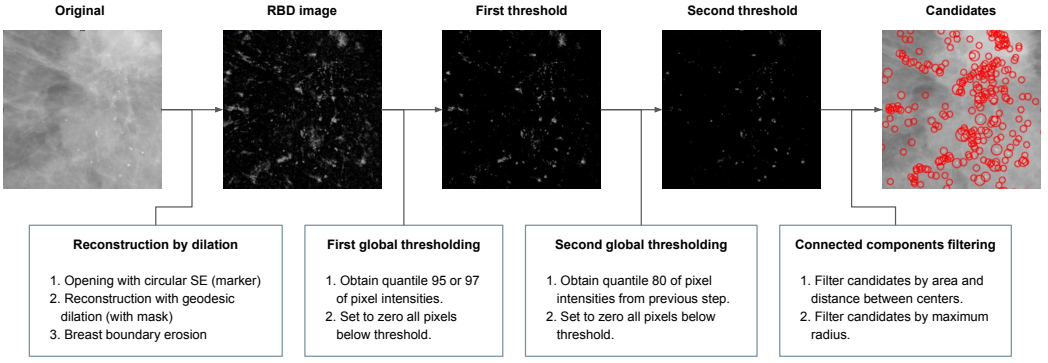

Figure 4. Grayscale Morphology candidate proposal pipeline.Plotted circles are enlarged by 10 pixels to be better appreciated.

Closing:

$$[f \cdot b](x, y) = (f \oplus b) \ominus b \tag{7}$$

We implemented a candidate proposal method using reconstruction by dilation (RBD) algorithm (Figure 4). The main principle of RBD is to repeat dilations of an image, called marker, until its intensity profile fits under the one of a second image, called mask. The mask acts as a constraint for the repeated dilations of the marker, and in our case was the original image. The marker represents an image with high values (seeds) at the objects we want to reconstruct and low ones elsewhere. In our case the marker was given by the opened version of the image. This produced a truncation of high intensity peaks relative to their surroundings, replacing MCs, noise peaks and other high density small structues with lower 'normal tissue' values in their place. After convergence of the iterative geodesic dilation process, we obtained an image where the general intensity profile is recovered, but the high peaks of MCs surrounded by normal tissue are not. Finally, subtracting the last dilated marker from the mask produced an image that preserved small round high-intensity structures while removing as much background tissue as possible.

In the same way as done in Hough Transform, we kept only detections from inside the breast by filtering the ones present in the breast border thus reducing the number of FP. We applied two consecutive global thresholds based on predefined quantiles of 95 and 80, in order to remove low intensity noise and preserve MC-like bright structures. After binarizing the image and obtaining the connected components, we filtered them based on their area and left only candidates separated by more than a minimum distance. Finally, we filtered the candidate blobs by maximum radius of the minimum enclosing circle.

2.1.5 Candidate proposal comparison

Performance Evaluation

To evaluate the performance of the different candidate proposal methods we use sensitivity and false positives per true positive per image (FPpTPpI) metrics, as well as the computation time per image. The higher the sensitivity, the lower the FPpTPpI and the less time, the better the method.

$$Sensitivity = \frac{TP}{TP + FN}, \; FPpTPpI = \frac{nFP_i}{nTP_i}, \; \forall i \in Images\ Dataset \tag{8}$$

To decide whether each candidate was a TP or a FP, we labeled them based on the intersection of a circle with diameter of 14 pixels centered on the candidate blob center with the ground truth mask. This allows us to consider all sizes of the target MCs (< 1 mm diameter) and account for imperfect

single-pixel annotations that sometimes do not correspond to the actual center of the MC. If this intersection contained any target lesions, we labeled the candidate as TP, otherwise it was considered as FP.

It is important to mention that in this candidate proposal stage we might have multiple patches matching the same ground truth. In order not to be misguided by this inflated number of TP, we dropped all candidates matching the ground truth except one.

For the final candidate proposal methods comparison present here we included two variants of the GM method (95 and 97 quantiles in the first threshold) and Hough detectors (first and second Hough transform) to picture the influence of some of the configuration parameters on the results.

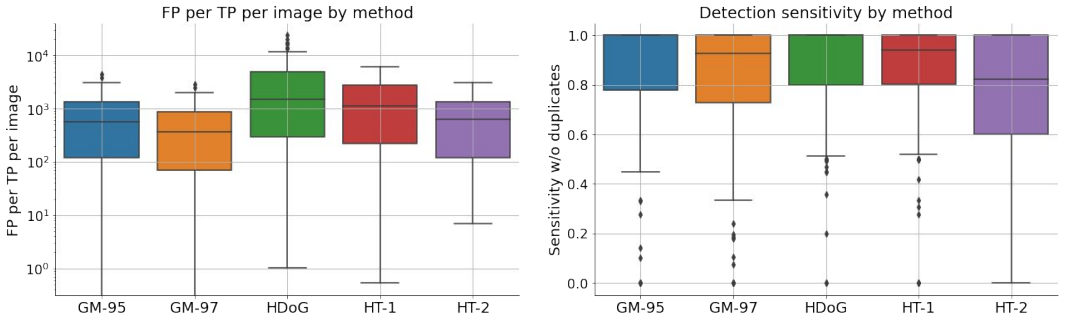

Figure 5. (Left) Sensitivity per candidate proposal method, (right) Log number of False positives per true positive per image for each candidate proposal method.

As we can see in Figure 5 (left), the GM detector with the 95 quantile threshold (GM-95) and HDoG detector showed the best sensitivity per image, achieving a detection sensitivity of 1 for more than half of images. At the same time, from Figure 5 (right) we can see both GM methods showed the lowest FPpTPpI, compared to all other methods.

Finally, in Figure 6 (left) we can see a significant difference in the computation time between methods, where GM ones were the fastest.

Considering all the covered aspects, we concluded that the *Grayscale Morphology with 95 quantil threshold was the best one*, as it achieved one of the highest sensitivities while being the fastest and having fewer FP.

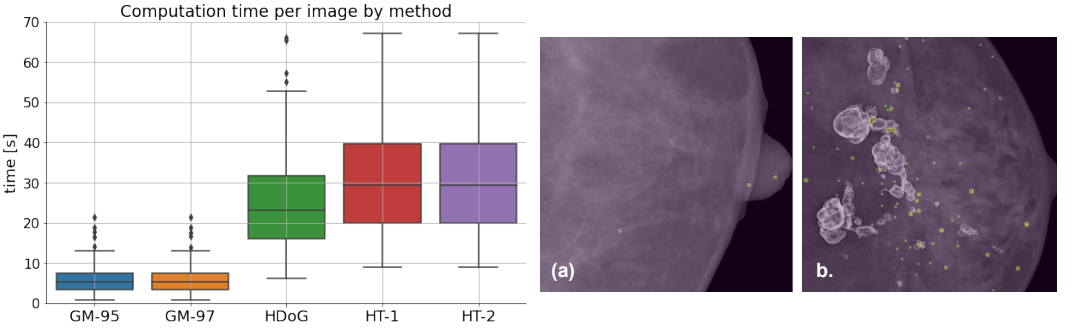

Figure 6. (Left) Computation time per image by method. (Right) Example images of cases with (a) microcalcification in the nipple and (b) highly compromised tissue, with overlapped detections (green circles).

In-depth analysis of the outputed candidates

Analyzing the cases with low sensitivity, we noticed that they were mostly repeated among the different methods. There were 39 images in total (19%) for which at least one of our candidate proposal methods had a sensitivity of less than 0.5. We found out that the most common reasons for this were: (a) Calcification inside the nipple or on the breast border, ignored by two of the three methods, and (b) highly compromised tissue by various lesions (see Figure 7 for examples).

It was interesting to notice finding (a) because it proved that the presence of MCs in the breast border is unlikely but not negligible, as the literature suggested. However, considering the trade-off between the number of missed MCs and the number of false positives added by keeping the breast border region, we decided to still retain the boundary filtering.

2.2 Machine learning for false positive reduction

In order to reduce the number of false positives from the previous stage, we developed a patch classification pipeline in which a set of features is extracted from the 14x14 patch around each candidate. Each sample is then fed into a classification model predicting whether a candidate patch is a TP or a FP.

2.2.1 Feature extraction

We extracted four general sets of features: first order statistics, Gabor-filters-based features, wavelet-decompositions-based features and Haar-like features. With these features we aimed to characterize the patch textural information, thus extracting the discriminative information contained in MCs and its surroundings.

First order statistic features: These features were computed directly from the intensity values from each candidate patch, see table 1 for the complete list.

Gabor filter features: First we filtered the complete image with six different Gabor kernels (Khan et al. 2017). The selection of a reduced set of features aimed to keep computation time low and the specific kernel choices (Figure 7) were meant to achieve 45° rotational invariant texture features. Then, from each candidate patch and using its bounding box coordinates, we obtain first and second order statistics over the Gabor-filtered image. See Table 1, for the complete list of extracted features.

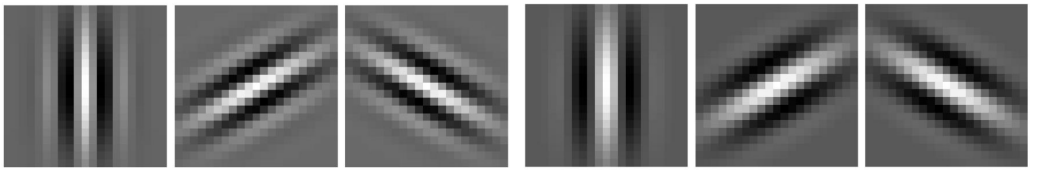

Figure 7. The six Gabor kernels we used to filter the images.

Wavelet features: We first decomposed each patch using the haar two-level wavelet decomposition, getting eight sub-patches: LL1, LH1, HL1, HH1, LL2, LH2, HL2, and HH2. LL contains the approximation coefficients or low frequency features; and LH,HL,HH contain the detail coefficients or horizontal, vertical, and diagonal high frequency features, respectively. With this, we aimed we aim to describe the local information content differentiated by scale and orientation (see Figure 8).

Following Fanizzi et al. 2020, from each of the eight subpatches we extracted first and second order statistics. Furthermore, from the first level detail features (LH1, HL1, HH1) we obtained their gray-level co-occurrence matrix (GLCM) (Fanizzi et al. 2019) using positional operators with offset

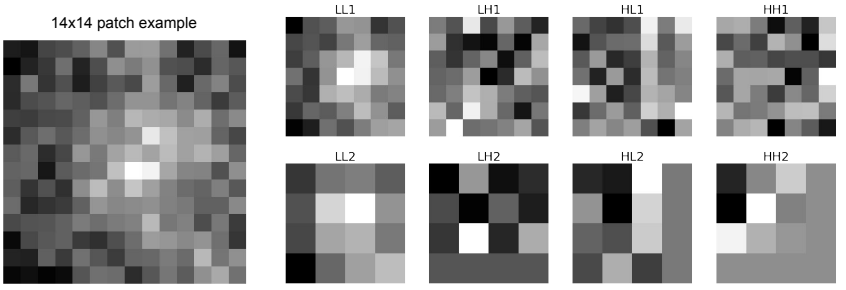

Figure 8. Decimated two-level decomposition of a patch example using 2-D discrete wavelet transform.

of 2 and angles of $0°$, $45°$ and $90°$. With this we focused in describing textural patterns spanning spatially inside the patch. Finally, we summarized the GLCM content using a set of statistical descriptors (see Table 1).

Table 1. Extracted features: Gabor filter features, wavelet features, and first order statistics.

Gabor filter features	**Wavelet features**	**First order statistics**
30 features	92 features	17 features
From each gabor filtered image (6 kernes), compute: • Energy • Mean • Standard deviation • Skewness • Kurtosis	From each of the eight wavelet decompositions, compute (56 features): • Mean • Skewness • Standard deviation • Kurtosis • Entropy • Uniformity • Relative smoothness From LH1, HL1, and HH1 decompositions, obtain three GLCM and compute (36 features): • Correlation • Homogeneity • Contrast • Dissimilarity	• Minimum value • Maximum value • 10^{th} quantile • 90^{th} quantile • Mean • Median • Standard deviation • Energy • Entropy • Uniformity • Skewness • Kurtosis • Interquartile range • Range • Mean absolute deviation • Robust mean absolut deviation • Root mean square

Haar-like features: Following Bria, Karssemeijer, and Tortorella 2014, three groups of Haar-like features were extracted from every patch, see Figure 9. The first two groups (a) and (b) were calculated as the difference between the sum of pixels belonging to adjacent rectangular regions, being concentric regions in the second case. The third group was formed by 45-degree-rotated versions of the features in the first two groups. All of these features were fastly computed taking advantage of the integral image method proposed by (Viola and Jones 2001) (sets (a) and (b)) and the variation presented in (Lienhart, Kuranov, and Pisarevsky 2003)for the rotated case.

The originally proposed version of the Haar-like features involved the extraction inside the interest region/image of all the scaled and translated possible combinations of the prototypes. Applying this approach to our 14x14 pixels patches led to a total of 28643 features for each patch. Taking into account that during training an average of 1685 patches could be extracted per image, the final size

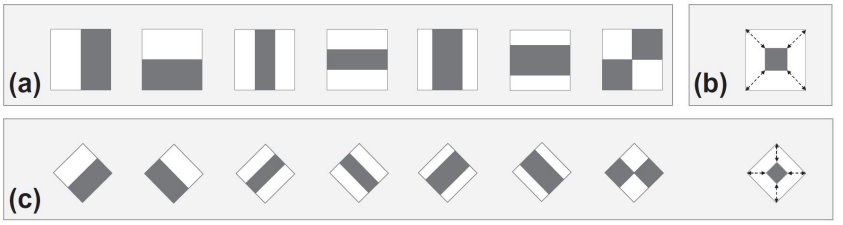

Figure 9. Three groups of Haar prototypes used. Image taken from Bria, Karssemeijer, and Tortorella 2014

of the generated data became a problem. The high dimensionality of the feature space represented a problem not only from a hardware-constraint point of view (fitting all the examples at the same time in memory), but also from a learning point of view, risking overfitting or the inclusion of useless non-discriminative features.

In order to reduce the feature space, we performed feature selection over the complete set of Haar features. We aimed to retain the most discriminative set of features, reducing the computation time and better condition the learning problem. To select the most discriminative features we use and embedded approach with a Random Forest (RF) Classifier, exploiting the Gini index feature importances computed in it.

In order to handle the enormous volume of data, the feature selection was done in several steps. First we extracted all the Haar-like features from all the candidate patches from the first 100 images in our train set. We then reduced the size of the dataset by doing downsampling balancing to reach a TP-FP ratio of 1:10. To reduce the computational load even further, we divided the dataset into two subsets of features, the ones coming from set (a) and the ones coming from set (b) and (c).

With these two sets of features we trained two separate RF models using five fold cross validation (CV). We selected 1000 and 2000 features from the first and second RF models respectively, based on the averaged feature importance across the five folds. Finally, we merged the two high importance subsets into a single 3000 features set, trained a new RF model again with five fold CV, and obtained each feature's importance.

To get the final set of features, we evaluated in the validation set the influence of the number of features by assessing the classification performance of a Random Forest model, measuring it by the area under the ROC curve (AUROC) and the area under the precision recall curve (AUPRC). Going from 25 features up to 3000, doubling the features number at each step, we trained a RF model for each number of predictors with a five fold CV.

Classifying Grayscale Morphology's detected candidates we obtained the following results:

In Figure 10 we can see that AUROC and AUPRC curves reached a plateau after ∼500 features, so we picked this threshold as our final number of selected Haar features. Considering also the AUPRC, and more specifically the precision, we get a broader picture of the model's performance since it gives a better sense of the amount of FP we were including relative to the number of TP we actually obtained.

A sample of the shared most significant features for each set is presented in Figure 11. Visually inspecting these, and the rest of the selected features, we noted that center features were always selected and mostly the wide-center-striped ones (horizontal, vertical and rotated). This was consistent with the information that we would expect to be discriminative when considering 14x14 patches centered in the candidate, mainly focusing on center versus surrounding contrast.

2.2.2 Classification

Once we obtained our final set of selected features, we carried out a series of experiments with three different classifiers. The experiments were conducted with two objectives: search for the best

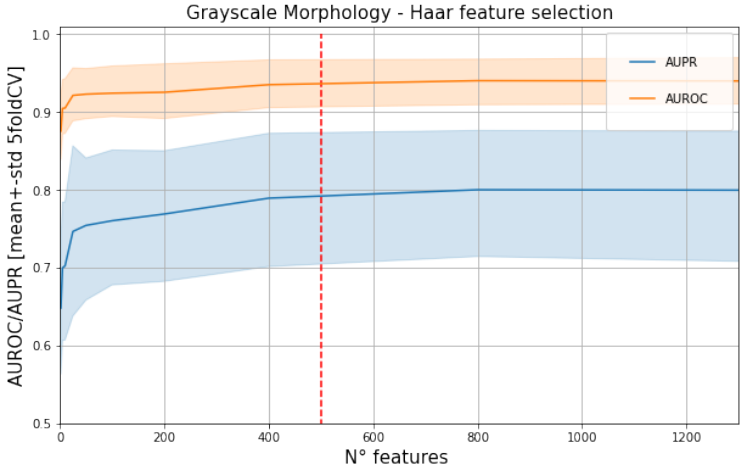

Figure 10. Effect of number of Haar features on the classification performance of Random Forest in the validation set of GM candidate proposal method.

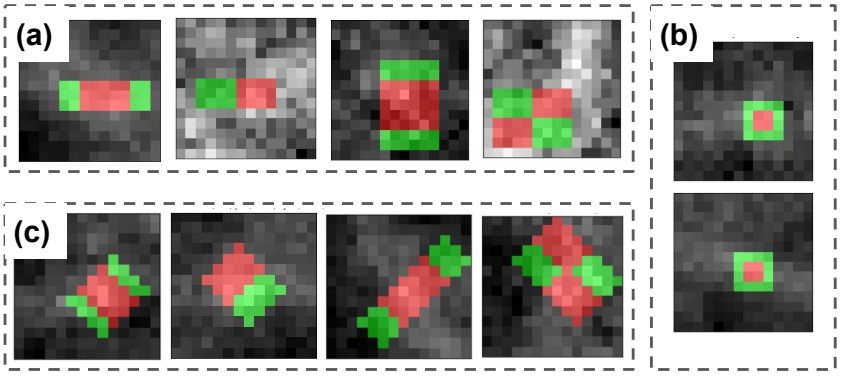

Figure 11. Sample features from the sets in Figure 9, plotted over diverse MC patches.

performing classifier, and evaluate their performance on each of the four types of features. Considering these sets of features individually and all together in a single one allowed us to understand their relative contribution.

In order to train, evaluate and carry out model selection, we used the partitioning of the database presented at the beginning of the methods section. Since the feature extraction –and therefore the classifiers– worked at a patch level, we ran the candidate proposal algorithm (stage one) on each of the images in the train and validation set, extracted their features (and normalized them) and made each sample inherit their original partitions. After labeling each candidate, we ended up with 233491 false positives and 2877 (1.2%) true positives in the train set and 116936 false positives and 528 true positives (0.4%) in the validation set.

To further ensure no data leakage with the validation set, and to have a better estimation of the performance of the final models on the test set, the model training and the hyperparameter selection experiments were carried out using case-wise 10-fold cross validation over only the train set. Leaving in this way the actual validation set just for final evaluation. All training procedures were conducted balancing the samples by downsampling with a ratio of 1:10 for TP and FP.

2.2.3 Perfomance evaluation

We used the AUROC and the AUPRC to evaluate the classifiers performance. Every performance measure was computed preserving the original prevalence of the classes.

The final model performance, measured over the validation set, was evaluated using the Free-Response Receiver Operating Characteristic Curve (FROC). This curve plots the True Positive Ratio (Sensitivity or Recall) versus the average False Positives per Image (FPpI). In our particular problem, in order to get rid of the bias of non-labeled MCs in MC-positive cases, the FROC curve is computed using the average False Positives per normal image, understanding *normal* as MC-negative cases. Furthermore, since having more than 50 FP per normal image results not useful scenario from a clinical perspective, the curves are constrained to the range [0, 50] on abscissa's axis.

2.2.4 Training the models

We evaluated five features subsets: first order statistics, Gabor filter features, wavelet features, Haar-like features, and all features, with the classifiers Extreme Gradient Boosting Random Forest classifier (XGBRF), normal RF, and Support Vector Machine Classifier (SVMc).

XGBRF is a decision tree ensemble learning algorithm, similar to RF. Both of them are supervised lerning methods that combine multiple weak learners - decision trees (DT) - to obtain a better model. RF uses bagging to build a set of DT from bootstrapped samples of the data set. Whereas gradient boosting uses gradient descent guided ensemble technique, in which each new tree that is added to the ensemble is built following an optimization procedure of the residuals values.

SVMc belongs to a different class of ML approaches. It is a supervised learning algorithm that in its kernel-trick version, maps each sample into a higher dimensional feature space in which a linear hyperplane can be found that maximizes the margins between the points of the two different classes.

At this step, we use the default hyperparameters for the models, without detailed tuning, just to have a general picture on the performance of each model.

Table 2. Performance evaluation of different classifiers with different sets of features

Model	Features	AUROC	AUPRC
XGBRF	fos	0.946±0.034	0.353±0.136
	Gabor	0.903±0.078	0.419±0.236
	wavelet	0.939±0.036	0.493±0.178
	Haar	0.922±0.058	0.475±0.175
	all	**0.955±0.025**	**0.567±0.148**t
RF	fos	0.941±0.038	0.356±0.136
	Gabor	0.908±0.080	0.428±0.241
	wavelet	0.940±0.041	0.515±0.201
	Haar	0.935±0.046	0.492±0.172
	all	**0.962±0.024**	**0.553±0.173**
SVM	fos	0.952±0.035	0.391±0.174
	Gabor	0.925±0.061	0.437±0.236
	wavelet	0.962±0.021	0.566±0.151
	Haar	0.933±0.048	0.519±0.152
	all	**0.973±0.016**	**0.584±0.148**

From the results presented in Table 2, we conclude that the highest performance in all models was achieved when using all the available features. Also, as is highlighted in yellow, the top three performances were obtained with Random Forest and SVM using the set of all features, and using

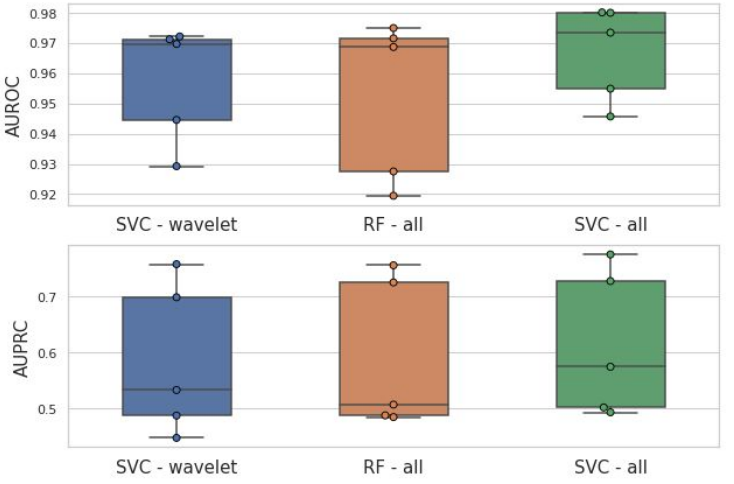

Figure 12. AUROC (up) and AUPRC (down) of best classifiers from Table 2, for validation set.

SVM and only the wavelet features. It is important to notice that the difference between SVM on wavelet features and RF using all features is practically null, but using all features and SVM gave us a slightly higher performance, as can be better seen in Figure 12. These results highlighted the power of SVM classifiers.

Considering all this evidence, *the SVM classifier trained with all features was chosen as the best performing one*.

2.2.5 Fine tuning and hard negatives exploitation

In order to further increase the performance of the classification stage of this first pipeline we implemented two additional experiments.

First, we performed a grid search hyperparameter tuning for the SVM classifier using a 5 fold cross validation approach. The grid of hyperparameters explored can be checked in Appendix 1.

Having found the best model and its best hyperparameters, we then explored the benefits of exploiting hard examples among the set of proposed candidates. Having a first stage in the pipeline that generates a large quantity of candidates (1000 FP per TP), it was expected that many of those FP candidates were going to be correctly classified by the classifier (easy negatives, non-MC-like structures or background tissue with a low prediction score), while others were going to represent hard negatives (MC-like structures) difficult to identify. See Figure 13 for examples of each case.

In order to use this in our favor, we explored two strategies. The first one involved training the selected SVM on all candidates and then fine tune it with hard negatives (hard negative mining approach). The second one, inspired by Bria, Karssemeijer, and Tortorella 2014, consisted in applying two SVMs in a *cascaded approach*. This last idea involved training two different SVMs, one on all available candidates to be used as a first stage and aimed to detect easy negatives; and a second one trained on the samples classified as positive by the first stage (hard negatives + true positives).

In both approaches the first stage was shared, training of the SVM on all cases, this was done over the complete candidates train set. Then, using the validation set, we defined a threshold to binarize the model's predictions that let us retain almost every TP (sensitivity of 98%), while eliminating approximately half of the FP (easy negatives). Once that model was trained, the predictions of candidates of both train and validation set were binarized using the selected threshold, and hard

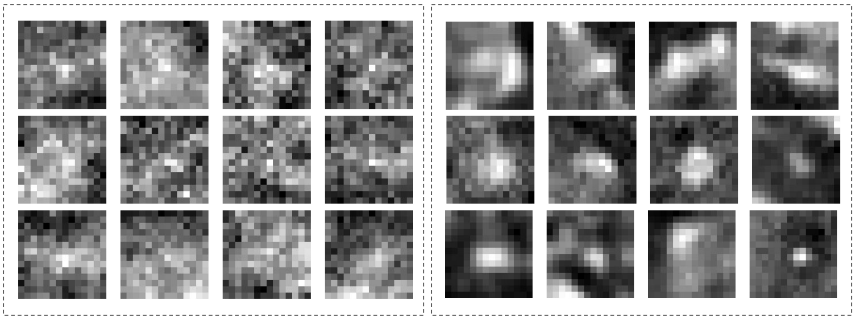

Figure 13. Examples of easy negatives (left), and hard negatives (right) candidates.

examples were identified. The second model (fine tuned version or second cascade stage) was trained on first stage's positives coming from the train set.

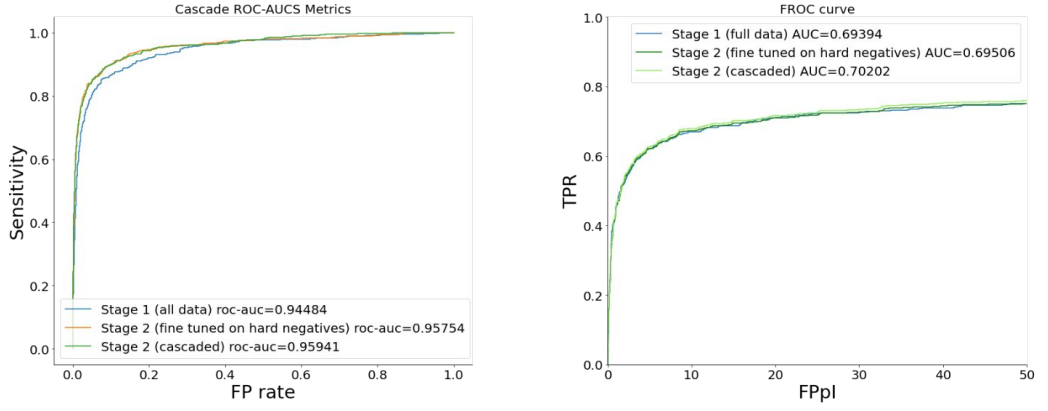

Figure 14. Results on validation set, of different ways of exploiting hard negative examples, including original model, hard negatives fine tuned one and cascaded approach.

Given the results presented in Figure 14, according to the AUFROC, the cascaded approach resulted in a slightly better performance, this is mainly due to a left-shifting of the curve, since by discarding half of the FP and very few TP, then FPpI were reduced while the sensitivity was preserved.

With the obtained results, we concluded that for the advanced image analysis and machine learning the best pipeline consisted of: **candidate proposal using GM; first order statistics, wavelet, Gabor, and selected Haar features extraction on 14x14 candidate centered patches; and two stages cascaded RBF-SVM based classification model.**

Lastly, it is worthy to make a comment on the use of the information provided in the original dataset related to pectoral muscle. INBreast originally included medio-lateral-oblique (MLO) views pectoral muscle segmentation masks. Considering that this high density region of the images has very low probability of containing MCs, we evaluated the benefit of including a pectoral muscle segmentation stage in this first pipeline in order to mask-out all detections in that region. Before implementing any automatic segmentation method of our own, we ran several experiments using the provided masks, noticing that there was not a significant improvement in the methods performance by the inclusion of this filtering. Therefore, this strategy was discarded.

2.3 Deep Learning

The state of the art for MC detection involves the use of Deep Learning (DL) algorithms. Just to mention some of the articles published in this field: in Marasinou et al. 2021 the authors combined image analysis methods and DL models to detect MCs; with the same objective in Bria et al. 2016, the authors implemented deep learning cascaded algorithms to handle the imbalance of the problem; in Zhang et al. 2019 the authors used generative models to take advantage of the high imbalance of the problem to model normal tissue and identify MC by their outlying condition.

However, there are unsolved challenges that researchers share while working with mammograms. These are mainly related to the high data imbalance manifested in the form of high resolution of images with very small objects of interest in them.

Apart from the rampant increase in performance in almost any computer vision task, one of the aspects that have spread the use of deep learning models across many fields and problems is the possibility to use pre-trained models. These models are trained on very large datasets have learnt to recognize generic descriptive image features and have constructed hierarchical feature representations of the image content. All this knowledge can be reutilized for similar or different computer vision tasks.

As shown in Matsoukas et al. 2022, transfer learning provides substantial increase in model's performance even when the task and images used to obtain pre-trained models differs considerably from the new objective and data. Previously learnt representations speed the convergence and allow using smaller datasets.

Deep learning has shown outstanding results in several medical image processing tasks. However, there are still many problems unsolved in this field and specific knowledge and creativity is still needed to overcome problem specific limitations that impede the straightforward transfer of general image computer vision solutions to medical images. Some examples of these challenges are the lack of publicly available models trained with very large images as mammograms, or optimized for the detection or classification of small and sparse objects in the image.

In this context, we decided to explore two different deep learning approaches to try to overcome the mentioned difficulties when trying to achieve automatic MCs detection in mammography images. The two of them dealt with the high resolution on mammography images by applying CNN models in a sliding window fashion. The first one used CNNs for patch classification while the second one used a CNN-based detection model to directly localize the desired lesions.

2.3.1 Second pipeline: Detection by Classification

The development of this approach implied two steps, first patch-wise training of classification models, and then the development of a method to apply the classifier in a sliding window combining the classification results in order to generate MC detections. We start for the second step for more clarity.

The idea behind the pipeline was to generate a smooth saliency map over the whole mammography image that indicated the probability of a MC to be located at each pixel. To do so (independently of the used patch size) we applied the classification model in a sliding window fashion with a stride smaller than the patch size. Even when we evaluated different strategies to combine all the predictions in a single saliency map, the best performing one involved transferring back the patch classification score to the original region in the image and then blending the overlapped areas by averaging. Imposed by hardware constraints, we used a non-minimal stride during the sliding window, which induced a 'blocky' aspect in the resulting saliency map. To overcome the problems that these induced in finding local peaks, the map was Gaussian smoothed. Finally, we normalized the whole map to obtain a probability map of in the range $[0, 1]$.

In order to get detections from the resulting map, we obtained local maxima points using a window of 14x14 pixels filtering the peaks below a desired confidence threshold. We considered non-maximal suppression as a final step to avoid overlapping detections but this did not have a

significant influence, so we ended up ignoring it. Thus, the final output of the pipeline were the detections with their associated boxes of 14x14 pixels size and confidence scores (Figure 15).

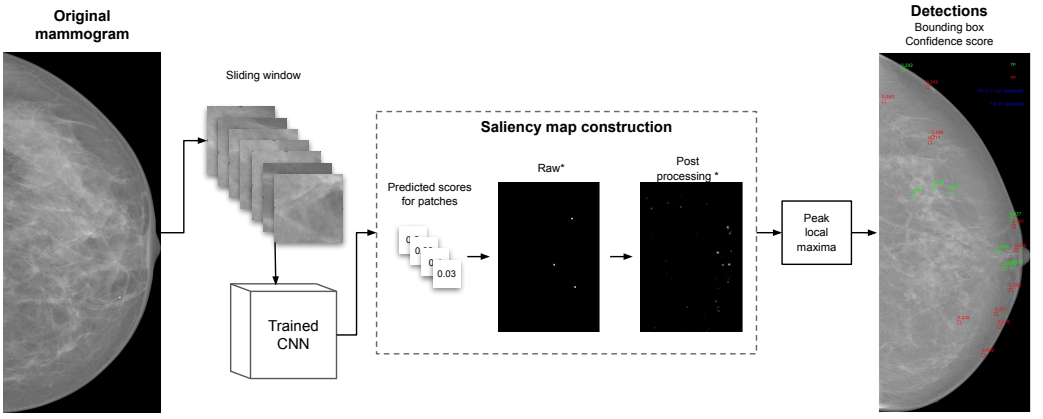

Figure 15. Detection by classification pipeline. *Zoomed in for clarity.

With a clear picture of how the prediction of the classification results were used, now we introduce the model training experiments we performed. Even though not all the experiments were successful, we used the results of some of them in the second deep learning pipeline and are therefore worthy to be presented.

First we opted to train classifiers on 224x224 patches not centered at MCs. We extracted all patches that could be fitted in each mammography image using a stride of 100 pixels. If the patch contained any MC then it was labeled as positive. We selected this first patch size in order to have a reasonable amount of information in the patch, and because it matched one of the standard sizes used in computer vision, allowing us to easily take advantage of pretrained models. In order to further avoid overfitting, a set of data augmentation techniques were applied on the dataset including: contrast and brightness jittering, affine transformations, rotations, resized cropping, horizontal and vertical flipping; were applied randomly one at the time with a tuned probability.

We explored several training strategies and models. In all cases, given the unbalanced nature of the dataset, we used two metrics to monitor the model's performance during training and validation: F1 score, by binarizing the predictions with the threshold maximizing Youden's index (Youden 1950), and AUROC. We studied the use of different pretrained standard CNN architectures, such as ResNet from 18 to 152 (Kaiming et al. 2015), DenseNet121 (Huang, Liu, and Weinberger 2016), EfficientNet from B0 to B7 (Tan and Le 2019).

For ResNets we first tried leaving the pretrained weights freezed and used the network as a feature extractor by just replacing the last fully connected (FC) layers. We tried different FC configurations, activation functions, percentages of dropout but this approach never led to a good performance. This was consistent with the fact that the information content of mammography images highly differs from natural images.

Then, we explored fine tuning the different architectures given that we had a considerable number of samples. We tried several configurations of the FC layers from one up to 3 layers of 500 neurons. We explored the use of different activation functions as ReLU and Leaky ReLU; drop out of 0.2, 0.4 and 0.5 in between the FC layers; Adam (lr=10-4) and SGD (lr 10-4, and momentum of 0.9) optimizers; learning rate schedulers (Reduce on Plateau or Step LR scheduler); and batch sizes of 32, 64, 128. Also, we considered different numbers of epochs and probabilities of data augmentation from 0, 0.2, 0.4. In order to take advantage of the large dataset, we explored varying the classes balance, by downsampling a different set of negative examples in each epoch to match the desired

P:N ratio. In this way, the model was still exposed to the complete set of negative samples during training. After several training experiments the best 3 performing models are presented in Table 7.

Table 3. Three best performing models on classification of 224x224 patches.

Model	F1-score	AUROC
ResNet50_01	**0.755184**	0.93
ResNet50_05	0.709174	**0.993204**
Densenet121_01	0.708066	0.943

However, even when the models achieved good results in the classification task, the saliency map approach with a stride of 24, the minimum size that was able to be handled by hardware constraints, did not give us good results. The main problem was that the saliency map was very coarse, and in many cases the generated detections were close to actual lesions but did not match them, dramatically lowering the overall performance (see Figure 17)

Following the literature Savelli et al. 2020, to improve the results we decided to change the patch size and the models. To explore the influence of the size of the patch being classified, we trained CNN classifiers using 16x16, 32x32 and 64x64 patches.

The main problem related to the classification of these small patches is the depth of the used CNN models. Mainly for 16x16 and 32x32, the conventional architectures used for the 224x224 cases were very deep, causing that the deepest convolutional layers would not work on feature masks but on single pixels. To overcome this, we generated a ResNet based architecture tailored specifically for our problem. The network has the same ResNet blocks as in the original architecture (depicted in Figure 16), but we introduced two main differences. The first one was to avoid the first "severe" downsampling performed in the first layer of the original ResNet, to prevent the resolution loss and its impact in detecting very small lesions sizes. The second main modification was an adaptive tuning of the network depth according to the size of the patch. In figure 16 the full architecture can be appreciated. For images of size 16x16 only N=2 downsampling stages were used (blue box), whereas 3 and 4 of them were used in 32x32 and 64x64 respectively.

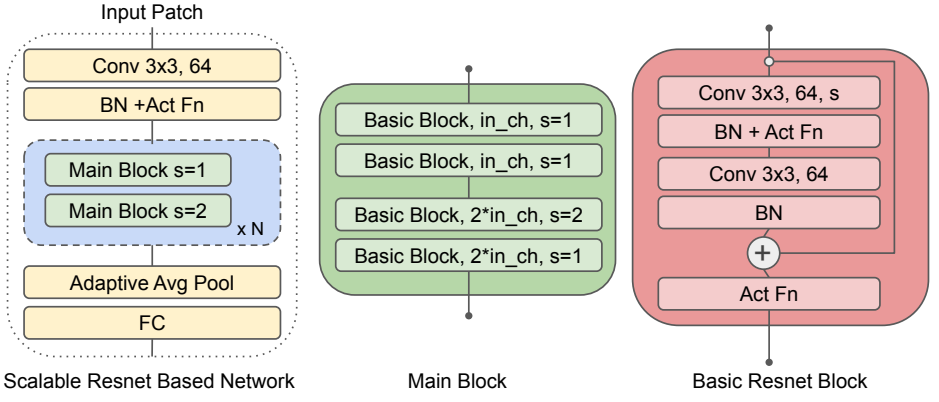

Figure 16. Implemented scalable ResNet-based model, where the blue box is the customizable downsampling stage.

We trained these three networks, varying the same hyperparameters commented before, but this time positive samples were centered on the MCs. The performance of the best models can be seen in Table 4.

Table 4. Best performing models for each patch size and model

Model	F1-score	AUROC	avgPR
64_ResNetBased_03	0.4423	0.993332	0.8477
32_ResNetBased_05	0.4251	0.993204	0.8950
16_ResNetBased_07	0.4240	0.993714	0.9009

See Appendix 2 for training specification details.

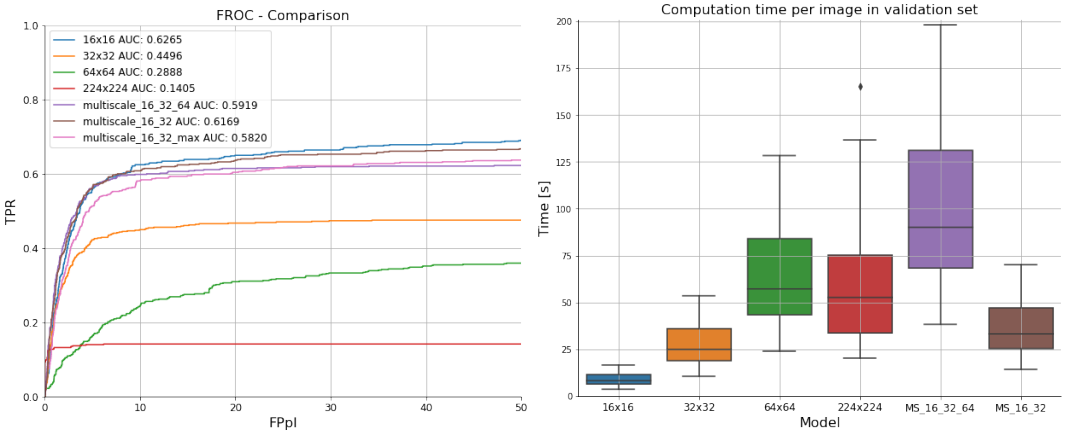

Figure 17. (Left) Comparison of FROC curves of the different variants of Detection by Classification pipeline, on validation set. (Right) Computation time per image of different variants for Detection by Classification pipeline.

Having trained these models, we tried each of them in the detection pipeline. For each model the stride was chosen considering the trade off between the computational time and the smoothness of the fine detail of the generated saliency map. In the same sense the kernel size of the gaussian filter was adapted. The resulting FROC curves can be seen in Figure 17 (left).

After studying the generated detections in more detail, the main problem of the approach was the saliency map generation. The classifiers performed very well according to the validation binary classification metrics, but the localization of lesions was not solved. Trying to improve these results and also inspired by Savelli et al. 2020, we tried a multiscale version of this pipeline, which combined the saliency maps obtained using different patches scales by averaging them. In this way we took advantage of the good classification performance of the different models while at the same time combining the higher context information available in bigger patches and the high resolution of smaller scales. All the models' performance comparison can be seen also in Figure 17. Finally, Figure 17 (right) shows the significant differences in computation times that existed among the proposed variants.

Even further discussion on this approach is done in the upcoming sections, with the results here presented we concluded that the best performing variant for this pipeline was: **16x16 patch wise classification with our specifically tailored ResNet based network applied in a sliding window fashion with stride of 8 and Gaussian smoothing of the saliency map with kernel size of 9.**

2.3.2 Third pipeline: Detection using Faster-R-CNN

The results obtained after the analysis of the Detection by Classification approach led to two important conclusions: patch-wise classification task can be successfully solved using standard CNNs, while the

transformation of saliency maps to a set of detections described by 14x14 bounding boxes is not trivial, and it becomes a bottleneck in achieving the best possible detection performance. Therefore, we decided to try another detection pipeline, one that would exploit the benefit of trained classification models while performing an automatic detection, without the need for a handcrafted post-processing.

Currently, many high-performance DL models have been developed for object detection tasks with a constraint of real-time processing: Spatial Pyramid Pooling (SPP-net) (He et al. 2014), You-Only-Look-Once (YOLO) (Redmon et al. 2015), Single Shot Detector (SSD) (Liu et al. 2015), Region-based CNNs (Girshick et al. 2013) and its follow up improved versions Fast-R-CNN (Girshick 2015), Faster-R-CNN (Ren et al. 2016), and Mask-R-CNN (He et al. 2017). We chose Faster-R-CNN model since it is easily available in Pytorch library, and it focuses purely on detection task in which we can use our pretrained CNNs as backbones, therefore satisfying two conditions given at the beginning of this section.

Faster-R-CNN is an extension of Fast-R-CNN. It introduces a fully convolutional network called region proposal network (RPN) that replaces the Selective Search stage in Fast-R-CNN and R-CNN, and is used to generate detection proposals (ROIs) with different scales and aspect ratios. It takes the convolution feature map generated by the backbone model as input and applies the anchors at each location. It generates the maximum number of k- anchor boxes, which after filtering by IoU with the ground truth box, are represented by an objectness score (probability of belonging to the foreground) and bounding box coordinates. The output from the RPN is fed to the ROI pooling layer that transforms different sized ROIs into a fixed sized feature map. A final classification head will classify ROI feature maps and output class prediction scores while a separate regression head will use propagated feature maps to adjust the anchor box to the final bounding box coordinates (Figure 18

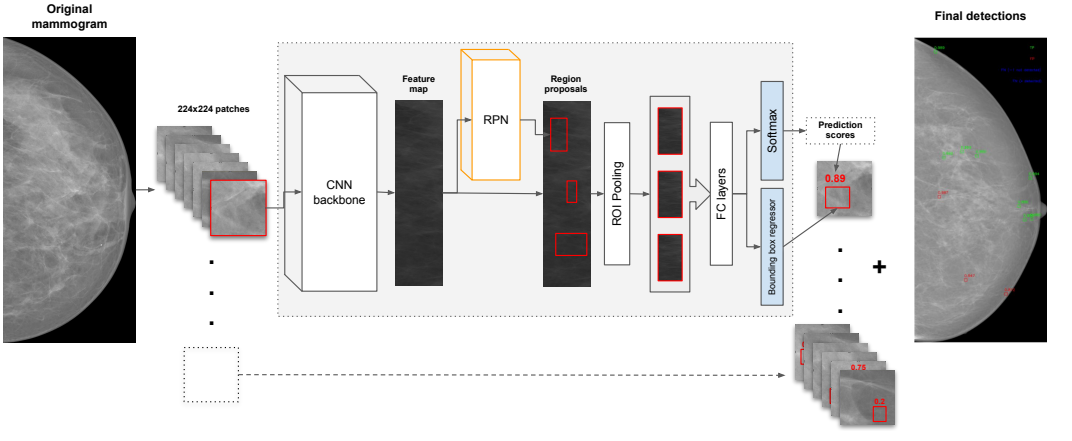

Figure 18. General detection pipeline with Faster-R-CNN architecture.

Transferring detection models coming from natural image processing to MC detections in mammography is not straightforward. As said before, the size of the object to localize and their sparseness impedes the direct application of these models on the whole image. In order to both take advantage of previously trained models for detection by classification pipeline and mitigate the impact of the resolution, we decided to apply Faster-R-CNN in a sliding window fashion over 224x224 patches across the image. Adapting Faster-R-CNN to this task required us to make few modifications compared to the original architecture and parameters. First, we used our best performing pretrained networks -namely ResNet50, EfficientNet-B3 and DenseNet121 as backbone models. This allowed us to transfer patch representations learned by these models, as they are more relevant to the detection task of MCs than those obtained from weights after training on the ImageNet dataset. However, it

also imposed a limitation, since now, to fully benefit from the pre-trained backbones, we had to use the same patch size they were trained on (224x224) for classification.

About the anchor boxes' aspect-ratios and scales, the original network used 3 different aspect-ratios and 3 different scales for anchors to achieve some scale invariance. However, since ground truths for the third DL approach consisted of bounding boxes of 14x14 pixels centered on each microcalcification, there was no need for this redundancy, and even though we explored different sizes of anchors, we kept the 14x14 size and the aspect ratio of 1:1.

As in the previous pipeline, we explored the fine tuning with different architectures. Among the shared parameters for all of the experiments we can highlight the usage of Adam optimizer with a learning rate of 0.0001, step learning scheduler with step size 3, gamma 0.1, and early stopping activated when a minimal difference of 0.0001 was not achieved on a validation set loss for more than 3 epochs. In all cases the model was trained on 224x224 patches, extracted from the original image using a stride of 100 with only patches containing MCs used for training. Every patch was separately z-score standardized before passing to the network, and the ground truth bounding boxes were defined as 14x14 squares around the center of every lesion inside the patch.

Furthermore, with the trained model, we performed the detection at inference time of the whole image with a sliding window fashion, with patches 224x224 and stride of 200. This small overlap between the patches allowed the model to better account for lesions in the borders. Finally, to avoid overdetection, NMS was applied as a final stage for all predicted candidates with IoU greater than 0.5.

We used Average Recall (AR) IoU and Average Precision (AP) IoU as metrics to evaluate the detectors performance. From the standard metrics used to evaluate object detection architectures, we used the average AR in the range of IoUs between prediction and ground truth from 0.5 to 0.95 with steps of 0.05, while AP was calculated for a IoU threshold of 0.5. This last metric was used to decide on the best performing model. AP can be defined as the area under the interpolated precision-recall curve, which can be calculated using following formula:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}), \tag{9}$$

where $r_1, r_2, ..., r_n$ is the recall levels (in an ascending order) at which the precision is first interpolated. The interpolated precision p_{interp} at a certain recall level r is defined as the highest precision found for any recall level $r' \geq r$:

$$p_{interp}(r) = \max_{r' \geq r} p(r') \tag{10}$$

The 3 best performing models on the aforementioned metric calculated on validation set are present in the Table 5:

Table 5. Three Best performing models on 224x224 patches with Faster-R-CNN model

Model	AR (IoU 0.50 - 0.95)	AP (IoU 0.50)	FROCAUC
FasterRCNN(ResNet50_00)	**0.778189**	**0.91305**	0.8865
FasterRCNN(EfficientNet_B3_00)	0.640479	0.82104	0.8798
FasterRCNN(DenseNet121_00)	0.748859	0.90434	**0.8946**

See Appendix 3 for training details. AR - Average Recall, AP - Average Precision (eq.9)

Taking into account the results obtained, we concluded that the best model in the third pipeline was **Faster-RCNN with DenseNet 121, applied in sliding window fashion on patches of**

224x224 with stride of 200 fusing all detections and finally applying non-maxima suppression with 0.5 IoU threshold.

3. Test set results and discussion

In this section we compared performances of the best models described in previous sections selected based on their performance on the validation set. The comparison is done over the full test set and only after all the models were trained and tuned. Recapitulating, for the first pipeline (AIA-ML) we selected the cascaded framework with the second stage SVMc trained on hard samples. For the Detection by Classification pipeline we selected a 16x16 patch-wise classification model with specifically tailored ResNet based classifier network. Finally, for the Detection pipeline which uses Faster-R-CNN, we selected a DenseNet121 backbone pretrained on the classification task with anchors of 14x14 pixels.

The main metric used to compare performance of aforementioned models was Area Under the FROC curve constrained to a maximum of 50 FPpI. Having put a great deal of effort into optimising all of our models for the highest speed of execution, we also compared the detection time per image.

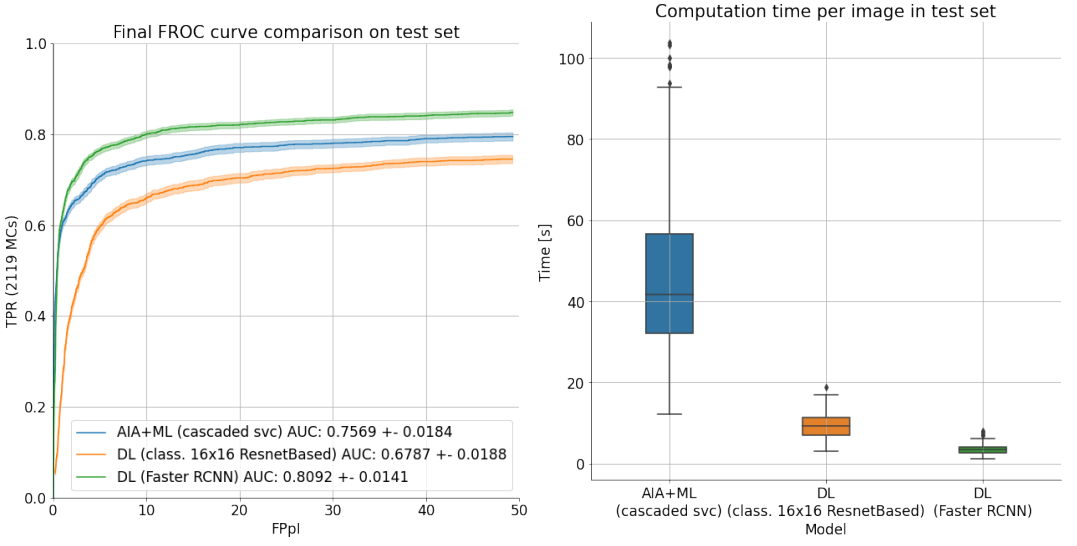

Figure 19. (Left) FROC curves with bootstrapping, and AUC of our three pipelines, on the test set. (Right) Computation time per image of each pipeline

In the Figure 19, we can clearly see that the Faster-R-CNN model considerably outperformed other models with a mean FROC-AUC of 0.8092 (std±0.0141). It is due to the fact that this is a deep learning model specifically designed for detection tasks, that we have could successfully adopted it in our problem. It was directly optimized with the detection objective and transfer learning was used to speed up convergence and improve final results, which was not the case for other models. All of them had separate stages in the detection process that had multiple parameters that were either manually adjusted or optimized separately. Faster-R-CNN was the only one that by means of gradient descent and backpropagation fine tuned its internal parameters for the whole detection process, from image to set of detections. Moreover, we can see that the machine learning approach with support vector machine classifier outperformed our deep learning classification-based detection approach. It shows that despite all our efforts in solving the saliency map-to-detection transformation problem, it remains a bottleneck for that pipeline.

Considering the detection time per image, it is clearly visible from figure 19 (right) that both DL approaches perform substantially better. However, this comparison is not completely fair since the hardware constraints are not shared among all the methods. While advanced image analysis and machine learning pipeline results were obtained in a machine running on a 11th Gen Intel®i7-11390H@ 3.40GHz x 8 processor and 16GB of RAM, the deep learning models were trained and evaluated using a Google Colab Pro instance with 2 vCPU, 24GB of RAM, and a Tesla-P100(16GB) NVIDIA Graphics card. Having said so, there are still valid reasons to state that the deep learning pipelines are faster at inference time. In the first place, the feature extraction (FE), being the most time-consuming stage of all of the models, in deep learning is implemented through highly efficient convolutions, while for machine learning approach it is done through a large variety of methods. In addition to that, before the feature extraction, a candidate proposal step is required, which is completely independent of the machine learning stage. A small difference in the detection time between two deep learning approaches is caused by the need of detection by classification model to use greater overlapping among the patches than Faster-R-CNN in order to obtain a high resolution saliency map, and by the need of additional transformation of the map into a set of detections.

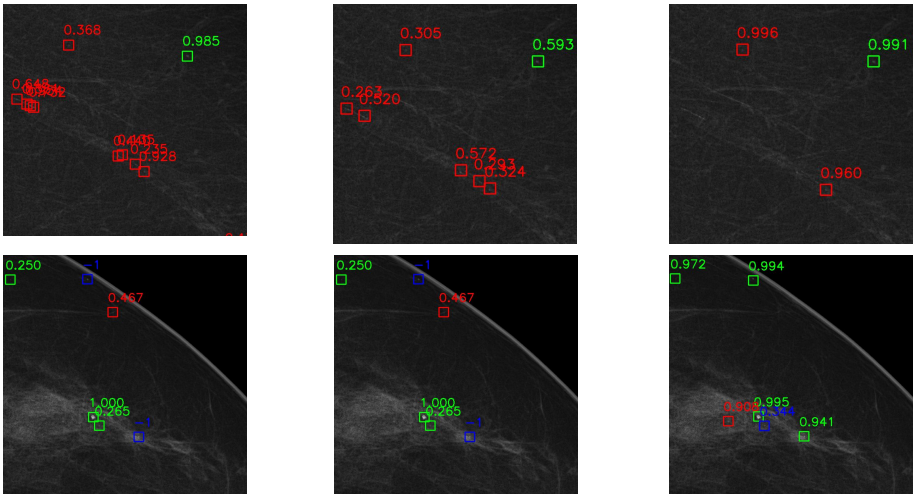

Figure 20. Detection examples on two patches of a mammogram from the test set: **(a)** AIA + ML (cascaded SVMc), **(b)** DL (class 16x16 ResNet-based), **(c)** DL (Faster-R-CNN). Note: dehazing is used to enhance the contrast for better visualization.

Looking at a concrete example of detections of all three models on the Figure 20 we can further expand on the differences between them. In the first row, we can see that the three pipelines output consequently less FP, with Faster-R-CNN having the smallest number. It is also clear that first two models suffer from over-detection in the regions with calcified ducts, while Faster-R-CNN was able to learn to ignore lesions in that area. More generally speaking, we have noticed that many of the FP with high confidence scores may correspond to actual lesions that were not annotated by radiologists (see Appendix 4 - Figure 22 for detailed examples) and could also be of medical interest. From the second row of the image we can also see that Faster-R-CNN had a higher sensitivity, being able to detect lesions closer to the border of the breast and in high density regions, which is more complicated for other algorithms.

To understand better the difference between detections of proposed pipelines we looked at the distribution of confidences by label for each of the models, which can be seen in the figure 21 below.

Both machine learning detection pipeline and Faster-R-CNN showed a clear bimodal distribution

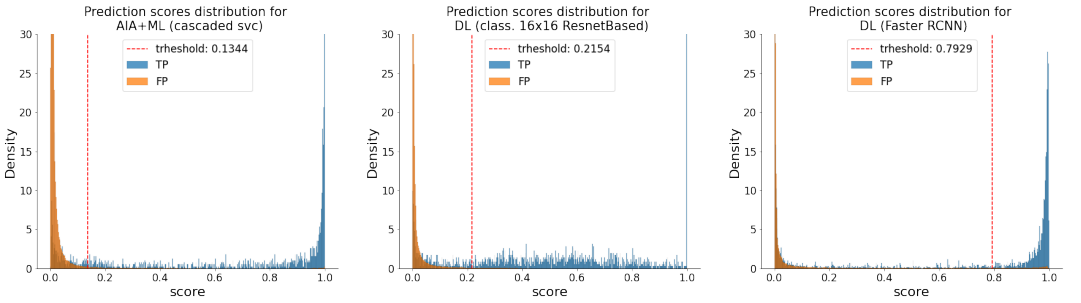

Figure 21. Scores distribution based on detection label per pipeline, in the test set. Red vertical line shows the optimized operative point obtained from the FROC curve for each of the models.

of scores, with peaks at 0 and 1, which correspond to correctly detected and rejected candidates. However, Faster-R-CNN had a much clearer gap between the two modes, which implies that the model output more confident detections and learned the target distribution of lesions better.

At the same time, for the detection by classification pipeline, there is not a clear peak of the detection scores for TP candidates, as they are more uniformly distributed over the whole range of probabilities. Nevertheless, we can see a clear peak in the neighborhood of 0, which means that the pipeline is able to filter out FP detections. This was consistent with our conclusions on saliency maps, and showed again that they are relevant and efficient at distinguishing background breast tissue and calcifications, even though accurate localization of the latest is challenging, at least within the framework and objectives we had set.

4. Conclusions

Three different pipelines for automatic calcification detection were presented in this work. The first one consisted in a first stage of candidate proposals generation with a grayscale morphology algorithm followed by a tailored feature extraction to finally classify them using a SVM-cascaded algorithm. The second one involved a sliding window patch-wise classification using an ad-hoc ResNet-based CNN in order to generate a microcalcification probability map over the mammogram which is then processed to obtain final detections. The last one implied a sliding window patch-wise microcalcification detection using Faster-R-CNN, and then combining the detections obtained in all patches.

For each pipeline several experiments were realized, always trying to extract the most of their different stages and supporting methodologies. As shown in previous sections, we thoroughly evaluated the performance of these methods, comparing intra-pipeline variants and finally all the pipelines in an independent hold out test set. Concisely, the results showed that Faster-R-CNN with a DenseNet121 backbone (pretrained on microcalcification patches classification), applied in a sliding window over the image with stride of 200, led to the best performance compared with the other two pipelines, achieving a FROC-AUC of 0.8092 ± 0.0141. Not ignoring the disclaimers previously made on the time comparison, we could also see that the third pipeline achieved the fastest inference time.

Our work highlights some of the key aspects of deep convolutional neural networks: their ability to extract meaningful features and hierarchical representations useful for different computer vision tasks. In our work we successfully trained CNNs in order to classify patches containing microcalcifications, forcing the network to adjust its internal weights in order to identify the discriminative characteristics of the very distinctive images that mammograms are. Even more, the learnt representations not only resulted useful for classification but also for detection. By reutilizing the pretrained models as backbone of the Faster-R-CNN model, we were able to help the model convergence and achieve better results, proving pretrained weights were still useful even with

different objective tasks. This well contrasts with the highly sensitive and detailed handcrafting of candidate extraction and feature extraction pipelines that were needed to get decent results in machine learning.

Across the developed experiments, the importance of having a good dataset has been permanently reinforced. The chance to have a large dataset of high quality images was definitely important in our success, but equally important was to curate it carefully in order to overcome the challenges the irregular annotations represented. We were challenged to find the proper techniques to deal with the tradeoff between reliability on the ground truths and their quantity. This was a key element to consider, not only because it affected the training of the chosen supervised approaches, but also because it was directly related with the accuracy of the models' performance evaluations.

Finally, our results and its comparison with the available literature showed us that the detection of microcalcifications in mammography images is a problem still waiting to be solved. Nevertheless, the huge advances done in the last decades, reinforce our willingness to contribute to finding clever, robust and efficient solutions. Our results appear to us very appealing and encouraging, even more by considering this being our first encounter with this particular task. Among the several ideas that we would like to explore, would be to keep experimenting with detection networks, considering the use of state of the art promising architectures based on attention mechanisms as Transformers.

References

Basile, T.M.A., A. Fanizzi, L. Losurdo, R. Bellotti, U. Bottigli, R. Dentamaro, V. Didonna, et al. 2019. Microcalcification detection in full-field digital mammograms: A fully automated computer-aided system [in en]. *Physica Medica* 64 (August): 1–9. ISSN: 11201797, accessed June 13, 2022. https://doi.org/10.1016/j.ejmp.2019.05.022. https://linkinghub.elsevier.com/retrieve/pii/S1120179719301309.

Bria, Alessandro, Claudio Marrocco, Adrian Galdran, Aurélio Campilho, Agnese Marchesi, Jan-Jurre Mordang, Nico Karssemeijer, Mario Molinara, and Francesco Tortorella. 2017. Spatial enhancement by dehazing for detection of microcalcifications with convolutional nets. In *Image analysis and processing - iciap 2017,* edited by Sebastiano Battiato, Giovanni Gallo, Raimondo Schettini, and Filippo Stanco, 288–298. Cham: Springer International Publishing. ISBN: 978-3-319-68548-9.

Bria, Alessandro, Claudio Marrocco, Nico Karssemeijer, Mario Molinara, and Francesco Tortorella. 2016. Deep Cascade Classifiers to Detect Clusters of Microcalcifications [in en]. In *Breast Imaging,* edited by Anders Tingberg, Kristina Lång, and Pontus Timberg, 9699:415–422. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing. ISBN: 978-3-319-41545-1 978-3-319-41546-8, accessed June 13, 2022. https://doi.org/10.1007/978-3-319-41546-8_52. http://link.springer.com/10.1007/978-3-319-41546-8_52.

Bria, Alessandro., Nico. Karssemeijer, and Francesco Tortorella. 2014. Learning from unbalanced data: A cascade-based approach for detecting clustered microcalcifications [in en]. *Medical Image Analysis* 18, no. 2 (February): 241–252. ISSN: 13618415, accessed June 26, 2022. https://doi.org/10.1016/j.media.2013.10.014. https://linkinghub.elsevier.com/retrieve/pii/S1361841513001588.

Dähnert, W. 2011. *Radiology review manual.* Wolters Kluwer Health. ISBN: 9781451153644. https://books.google.it/books?id=N87SCi%5C_ZJJYC.

Diaz-Huerta, C.C., E.M. Felipe-Riveron, and L.M. Montaño-Zetina. 2014. Quantitative analysis of morphological techniques for automatic classification of micro-calcifications in digitized mammograms [in en]. *Expert Systems with Applications* 41, no. 16 (November): 7361–7369. ISSN: 09574174, accessed June 24, 2022. https://doi.org/10.1016/j.eswa.2014.05.051. https://linkinghub.elsevier.com/retrieve/pii/S0957417414003467.

Dibden, Amanda, Judith Offman, Stephen W. Duffy, and Rhian Gabe. 2020. Worldwide review and meta-analysis of cohort studies measuring the effect of mammography screening programmes on incidence-based breast cancer mortality. *Cancers* 12 (4). ISSN: 2072-6694. https://doi.org/10.3390/cancers12040976. https://www.mdpi.com/2072-6694/12/4/976.

Fanizzi, Annarita, Teresa M. A. Basile, Liliana Losurdo, Roberto Bellotti, Ubaldo Bottigli, Rosalba Dentamaro, Vittorio Didonna, et al. 2020. A machine learning approach on multiscale texture analysis for breast microcalcification diagnosis [in en]. *BMC Bioinformatics* 21, no. S2 (March): 91. ISSN: 1471-2105, accessed June 13, 2022. https://doi.org/10.1186/s12859-020-3358-4. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-3358-4.

Fanizzi, Annarita, Teresa Maria Basile, Liliana Losurdo, Roberto Bellotti, Ubaldo Bottigli, Francesco Campobasso, Vittorio Didonna, et al. 2019. Ensemble Discrete Wavelet Transform and Gray-Level Co-Occurrence Matrix for Microcalcification Cluster Classification in Digital Mammography [in en]. *Applied Sciences* 9, no. 24 (December): 5388. ISSN: 2076-3417, accessed June 17, 2022. https://doi.org/10.3390/app9245388. https://www.mdpi.com/2076-3417/9/24/5388.

Girshick, Ross B. 2015. Fast R-CNN. *CoRR* abs/1504.08083. arXiv: 1504.08083. http://arxiv.org/abs/1504.08083.

Girshick, Ross B., Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* abs/1311.2524. arXiv: 1311.2524. http://arxiv.org/abs/1311.2524.

Gulsun, Meltem, Figen Basaran Demirkazik, and Macit Ariyurek. 2003. Evaluation of breast microcalcifications according to breast imaging reporting and data system criteria and le gal's classification. *European Journal of Radiology* 47 (3): 227–231. https://doi.org/https://doi.org/10.1016/S0720-048X(02)00181-X.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *CoRR* abs/1703.06870. arXiv: 1703.06870. http://arxiv.org/abs/1703.06870.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR* abs/1406.4729. arXiv: 1406.4729. http://arxiv.org/abs/1406.4729.

Huang, Gao, Zhuang Liu, and Kilian Q. Weinberger. 2016. Densely connected convolutional networks. *CoRR* abs/1608.06993. arXiv: 1608.06993. http://arxiv.org/abs/1608.06993.

Kaiming, He, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. 2015. *Deep residual learning for image recognition.* https://doi.org/10.48550/ARXIV.1512.03385. https://arxiv.org/abs/1512.03385.

Khan, Salabat, Muhammad Hussain, Hatim Aboalsamh, and George Bebis. 2017. A comparison of different Gabor feature extraction approaches for mass classification in mammography [in en]. *Multimed Tools Appl* 76, no. 1 (January): 33–57. ISSN: 1380-7501, 1573-7721, accessed June 13, 2022. https://doi.org/10.1007/s11042-015-3017-3. http://link.springer.com/10.1007/s11042-015-3017-3.

Lienhart, Rainer, Alexander Kuranov, and Vadim Pisarevsky. 2003. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection [in en]. In *Pattern Recognition,* edited by Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Bernd Michaelis, and Gerald Krell, 2781:297–304. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-40861-1 978-3-540-45243-0, accessed June 27, 2022. https://doi.org/10.1007/978-3-540-45243-0_39. http://link.springer.com/10.1007/978-3-540-45243-0_39.

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2015. SSD: single shot multibox detector. *CoRR* abs/1512.02325. arXiv: 1512.02325. http://arxiv.org/abs/1512.02325.

Logullo, Angela, Karla Pregenzi, Cristiane Nimir, Andreia Franco, and Mario Campos. 2022. Breast microcalcifications: past, present and future (review). *Molecular and Clinical Oncology* 16 (4): 81. https://doi.org/10.3892/mco.2022.2514.

Marasinou, Chrysostomos, Bo Li, Jeremy Paige, Akinyinka Omigbodun, Noor Nakhaei, Anne Hoyt, and William Hsu. 2021. *Segmentation of Breast Microcalcifications: A Multi-Scale Approach* [in en]. Number: arXiv:2102.00754 arXiv:2102.00754 [eess], February. Accessed June 13, 2022. http://arxiv.org/abs/2102.00754.

Matsoukas, Christos, Johan Fredin Haslum, Moein Sorkhei, Magnus Soderberg, and Kevin Smith. 2022. What Makes Transfer Learning Work for Medical Images: Feature Reuse & Other Factors [in en], 10.

Monticciolo, Debra L. 2020. Current guidelines and gaps in breast cancer screening. Focus on Data, Distilled, *Journal of the American College of Radiology* 17 (10): 1269–1275. ISSN: 1546-1440. https://doi.org/https://doi.org/10.1016/j.jacr.2020.05.002. https://www.sciencedirect.com/science/article/pii/S1546144020305147.

Mordang, J. J., A. Gubern-Mérida, A. Bria, F. Tortorella, R. M. Mann, M. J. M. Broeders, G. J. den Heeten, and N. Karssemeijer. 2018. The importance of early detection of calcifications associated with breast cancer in screening [in en]. *Breast Cancer Res Treat* 167, no. 2 (January): 451–458. ISSN: 0167-6806, 1573-7217, accessed June 13, 2022. https://doi.org/10.1007/s10549-017-4527-7. http://link.springer.com/10.1007/s10549-017-4527-7.

Moreira, Inês C., Igor Amaral, Inês Domingues, António Cardoso, Maria João Cardoso, and Jaime S. Cardoso. 2012. INbreast [in en]. *Academic Radiology* 19, no. 2 (February): 236–248. ISSN: 10766332, accessed June 13, 2022. https://doi.org/10.1016/j.acra.2011.09.014. https://linkinghub.elsevier.com/retrieve/pii/S107663321100451X.

Morris, Ben. 2003. The components of the Wired Spanning Forest are recurrent [in en]. *Probab Theory Relat Fields* 125, no. 2 (February): 259–265. ISSN: 0178-8051, 1432-2064, accessed June 27, 2022. https://doi.org/10.1007/s00440-002-0236-0. https://link.springer.com/10.1007/s00440-002-0236-0.

Muthuvel, Marimuthu, Balakumaran Thangaraju, and Gowrishankar Chinnasamy. 2017. Microcalcification cluster detection using multiscale products based Hessian matrix via the Tsallis thresholding scheme [in en]. *Pattern Recognition Letters* 94 (July): 127–133. ISSN: 01678655, accessed June 13, 2022. https://doi.org/10.1016/j.patrec.2017.05.002. https://linkinghub.elsevier.com/retrieve/pii/S0167865517301447.

O'Grady, S., and M.P. Morgan. 2018. Microcalcifications in breast cancer: from pathophysiology to diagnosis and prognosis. *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer* 1869 (2): 310–320. ISSN: 0304-419X. https://doi.org/https://doi.org/10.1016/j.bbcan.2018.04.006. https://www.sciencedirect.com/science/article/pii/S0304419X18300593.

Redmon, Joseph, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You only look once: unified, real-time object detection. *CoRR* abs/1506.02640. arXiv: 1506.02640. http://arxiv.org/abs/1506.02640.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2016. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* [in en]. Number: arXiv:1506.01497 arXiv:1506.01497 [cs], January. Accessed June 24, 2022. http://arxiv.org/abs/1506.01497.

Savelli, B., A. Bria, M. Molinara, C. Marrocco, and F. Tortorella. 2020. A multi-context cnn ensemble for small lesion detection. *Artificial Intelligence in Medicine* 103:101749. ISSN: 0933-3657. https://doi.org/https://doi.org/10.1016/j.artmed.2019.101749. https://www.sciencedirect.com/science/article/pii/S0933365719303082.

Sung, Hyuna, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. 2021. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries [in en]. *CA A Cancer J Clin* 71, no. 3 (May): 209–249. ISSN: 0007-9235, 1542-4863, accessed June 13, 2022. https://doi.org/10.3322/caac.21660. https://onlinelibrary.wiley.com/doi/10.3322/caac.21660.

Tan, Mingxing, and Quoc V. Le. 2019. Efficientnet: rethinking model scaling for convolutional neural networks. *CoRR* abs/1905.11946. arXiv: 1905.11946. http://arxiv.org/abs/1905.11946.

Tot, Tibor, Maria Gere, Syster Hofmeyer, Annette Bauer, and Ulrika Pellas. 2021. The clinical value of detecting microcalcifications on a mammogram. Precision Medicine in Breast Cancer, *Seminars in Cancer Biology* 72:165–174. ISSN: 1044-579X. https://doi.org/https://doi.org/10.1016/j.semcancer.2019.10.024. https://www.sciencedirect.com/science/article/pii/S1044579X19303566.

Venkatesan, Aruna, Philip Chu, Karla Kerlikowske, Edward A. Sickles, and Rebecca Smith-Bindman. 2009. Positive predictive value of specific mammographic findings according to reader and patient variables. PMID: 19164116, *Radiology* 250 (3): 648–657. https://doi.org/10.1148/radiol.2503080541. eprint: https://doi.org/10.1148/radiol.2503080541. https://doi.org/10.1148/radiol.2503080541.

Viola, P., and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 ieee computer society conference on computer vision and pattern recognition. cvpr 2001,* 1:I–I. December. https://doi.org/10.1109/CVPR.2001.990517.

Youden, W. J. 1950. Index for rating diagnostic tests. *Cancer* 3 (1): 32–35. https://doi.org/https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.

Zhang, Fandong, Ling Luo, Xinwei Sun, Zhen Zhou, Xiuli Li, Yizhou Yu, and Yizhou Wang. 2019. Cascaded Generative and Discriminative Learning for Microcalcification Detection in Breast Mammograms [in en]. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 12570–12578. Long Beach, CA, USA: IEEE, June. ISBN: 978-1-72813-293-8, accessed June 27, 2022. https://doi.org/10.1109/CVPR.2019.01286. https://ieeexplore.ieee.org/document/8953247/.

Appendix 1. Final parameters for different methods

- **Hough Transform**
  - Dehazing parameters = { 'omega': 0.9, 'window_size': 11, 'radius': 40, 'eps': 1e-5 }
  - For OpenCV method *HoughCircles*. hough1_params = {'method': cv2.HOUGH_GRADIENT, 'Default accumulator resolution': 1, 'minDist': 10, 'Higher threshold for Canny detector': 300, 'Accumulator threshold': 5, 'minRadius': 2, 'maxRadius': 10}
    hough2_params = {'method': cv2.HOUGH_GRADIENT, 'Default accumulator resolution': 1, 'minDist': 10, 'Higher threshold for Canny detector': 300, 'Accumulator threshold': 3, 'minRadius': 2, 'maxRadius': 10 }

  - Extra parameters = {'back_ext_radius (rolling ball)': 50,'erosion_iter': 20, 'erosion_size': 5,'min_hough2_distance': 6, 'gaussianf_sigma': 2, 'first global threshold': 0.97, 'second global threshold': 0.95.}

- **Hessian of Difference of Gaussians**
  - dog_parameters = {'min_sigma': 1, 'max_sigma': 3, 'n_scales': 20, 'sigma_ratio': 1.05, 'dog_blob_th': 0.006, 'dog_overlap': 0.2, 'dog_min_dist': 6}
  - hessian_parameters = {'method': 'eigenval' | 'marsinou', 'hessian_threshold': 1.4, 'hessian_th_divider': 300 }

- **Grayscale Morphology**
  - General params = {'threshold (first quantile)': 0.95, 'threshold (second quantile)': 0.80, 'min_distance': 6, 'area': 14, 'SE rectangle (RBD geo dilation)': 3, 'SE circle (RBD opening)': 14, 'max_radius': 10 }

- **SVC hypertuning using gridsearch CV**, best experiment: RBF, gamma = 0.1, and $C = 10$.
  - **C**, inversely related to the tolerance of samples entering into the decision function margin = {1, 10, 100}
  - Kernel functions:
    - Linear
    - Polynomial = {'degree': 3, 5, 7, 10}
    - Random basis function (RBF) = {'gamma': 0.01, 0.1, 1, 'scale'}

Appendix 2. Training experiments for Detection by Classification

Training configuration that led us to the 3 best performing methods for classification on 224x224 patches.

Table 6. Three best performing models on classification of 224x224 patches. N: Negatives P: Positives

Exp. name	FC Layers	N:P	Batch Size	LR sched.	Data Aug. prob.	Its. per epoch	Epochs
ResNet50_01	3	5	128	StepLR	0.2	All	15
ResNet50_05	1	All	128	ReduceOnPlateau	0.4	400	30
Densenet121_00	1	All	32	StepLR	0.4	500	15

Among these three models, all shared:

- Leaky ReLU activation function on the fully connected layers.
- Initialized with their pretrained weights on Image net.
- Z-score normalization.
- Adam optimizer, with lr=0.0001.
- Early stopping after 3 epochs with no change in validation loss greater than 0.0001.
- BCEWithLogits
- Dropout of 0.5 beteween FC layers.

Training configuration that led us to the best performing methods for classification on 16x16, 32x32, 64x64 patches.

Among these three models, all shared:

- GELU activation function on the fully connected layers.
- Initialized with their pretrained weights on Image net.
- Z-score normalization.

Table 7. Three best performing models on classification of 16x16, 32x32, 64x64 patches. N: Negatives P: Positives

Exp. name	MainBlocks	N:P.	Data Aug. prob.	Iterations per epoch	Epochs
16_ResNetBased_07	2	All	0.8	500	30
32_ResNetBased_05	3	All	0	300	30
64_ResNetBased_03	4	10	0	200	30

- Batch size: 128
- Adam (AdamW in 16x16 case) optimizer, with lr=0.0001.
- StepLR Scheduler, Step size=7 gamma=0.1
- Early stopping after 3 epochs with no change in validation loss greater than 0.0001.
- BCEWithLogits
- Dropout of 0.4 beteween FC layers.

Appendix 3. Training experiments for Detector Faster-R-CNN

Training configuration that led us to the best performing models.

- Backbone network: the one indicated in the model/experiment name.
- Epochs: The one with Desnet backbone was trained for 10 epochs the other two for 15.
- Anchors size = 14, ratio = 1
- Initialized weights on the pretrained model for classification of MC patches task.
- Z-score normalization.
- Batch size: 6
- Adam optimizer, lr=0.0001.
- StepLR Scheduler, Step size=3 gamma=0.1
- Early stopping after 3 epochs with no change in validation loss greater than 0.0001.

Appendix 4. Detection examples

- In figure 22:

  A True positives with low confidence - lesions that we have detected but that fell below the operative point threshold (finally being FN).
  B True positives with high detection confidence.
  C False positives with low detection confidence.
  D False positives with high detection confidence.
  E False negatives - lesions that were not detected.

In the first row (a), the three pipelines, we see detected structures that are compatible with the actual MCs labeled in the ground truth. Is interesting to notice that all the structures are very subtle, either very small or with very low contrast with the background. In some cases of the false positives with low confidence (c), the algorithms have detected structures consistent with the appearance of microcalcifications, even though again they are very subtle as in the previous case. Both AIA+ML and detection by classification pipelines have false positives with high confidence (d) coming from calcified ducts, whereas Faster-R-CNN pipeline, detects lesions that are probably unlabeled real microcalcifications. The false negatives in detection by classification are very evident, showing the actual lack of performance of this approach. In the case of AIA+ML as expected, border lesions are ignored (by design), but this situation is interestingly happening also in Faster-R-CNN suggesting that different training strategies paying more attention to the borders of the best might be helpful.

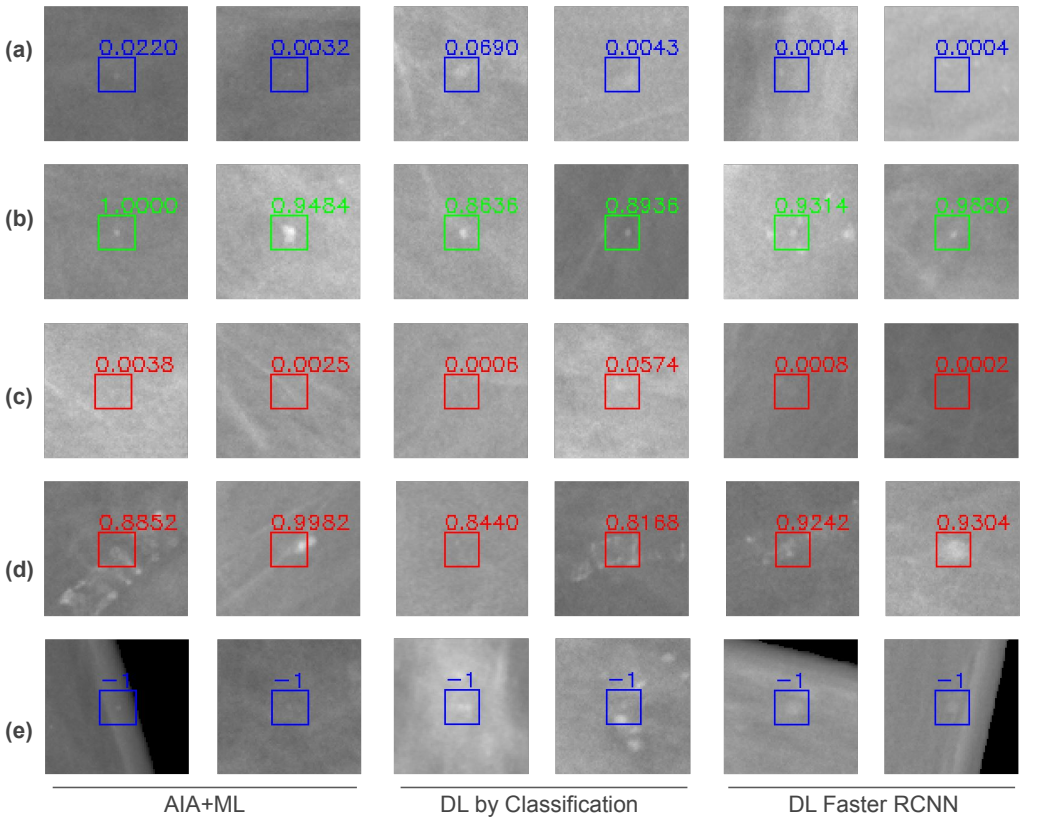

Figure 22. Examples of detections of the three models of various labels and confidences. Green: TP, red: FP, blue: FN (-1 when not detected, * when detected)