This is mostly for general structure and to understand what plots/tables/info can be missing. The text is expected to be expanded-detailed.

—---------- Section with comments not to miss to explain in this report —----------

- ~~Dataset~~
  - ~~Labeling:~~
    - ~~Clusters of microcalcifications. The GT are full circles and the mC inside are not annotated.~~
- Labeling and ~~metrics~~ in both detectors and ML stage (all results are with circular labeling)
- Mention why we don't use pectoral muscle suppression.
- ~~Mention the normals subset, and how we use it in all stages to compute metrics.~~
- ~~In the detectors' comparison, check how many low sensitivity images (the outliers in the boxplots) we have per detector and if there are shared images and why (e.g. 5 images with very high breast density, etc.).~~
- ~~Labeling problem in database: week labeling~~
- ~~Number of haar features: 28643~~
- Add appendix of algorithms' parameters (ex. Candidate detectors, etc.)

—----------

Figures here

—----------

Introduction

Problem

Breast cancer has the highest incidence rate amongst all cancers in women in the world [GLOBOCAN]. It is of special interest to create Computed Aided Detection (CADe) systems that properly identify lesions due to the difficulty that radiologists face distinguishing the most relevant grades of ductal carcinoma in situ (DCIS) in mammograms. Furthermore, it has been shown that there is a considerable percentage of cancers that could have been diagnosed in a previous screening examination, and specifically, early detection of microcalcifications (MCs) is key for the prevention of invasive cancer [IMPORTANCE 2018].

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.

Dataset description

We used the INbreast dataset [CITE] which includes 410 full-field digital mammograms with pixel size of 70 μm 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): calcifications, masses, clusters, spiculated regions, asymmetries, and distortions. The calcifications 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 [RADIOLOGY MANUAL] and therefore they become the primarly detection target of our detection systems. In the INbreast dataset around 87% of all labeled calcifications are linked to malignant cancer. Furthermore, around 40% of all labeled calcification ROIs were found in images tagged as suspicious of malignancy, highly suggestive of malignancy, or malignant according to the BI-RADS assessment categories.

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 background 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.

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. This allows us to tackle the problem of the high imbalance of candidates. 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-RCNN) run in a sliding window fashion over the full mammogram.

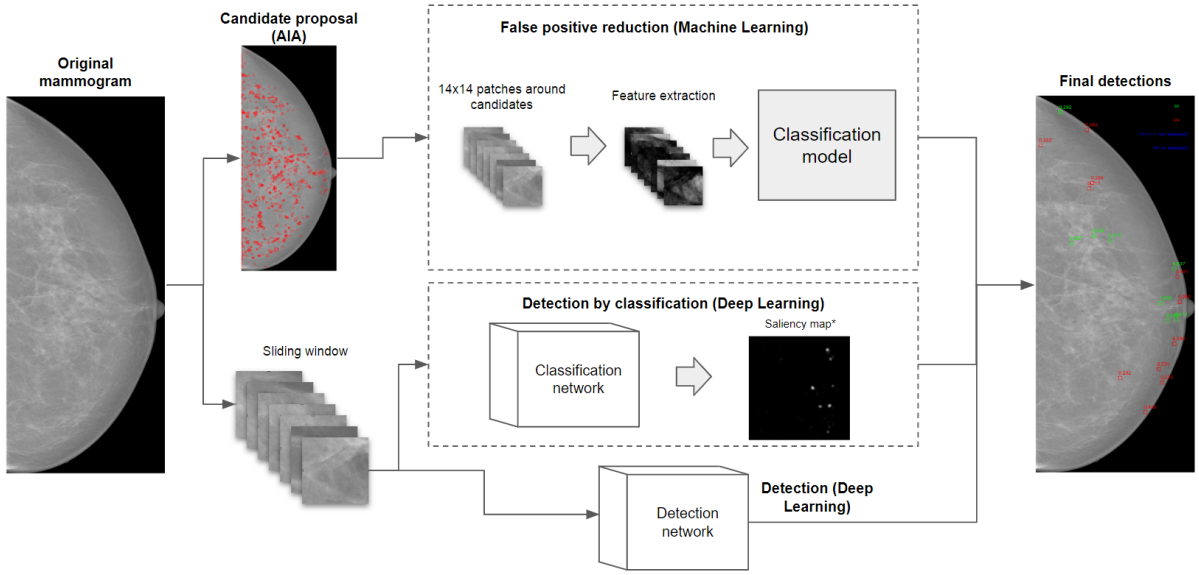


Figure 1. General diagram showing the three developed pipelines. *Zoomed in to see details.

As an initial common step applied for all pipelines, we processed 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 one 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 correctly spot several MCs that were 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, but we discarded it due to the inaccuracy and possible bias induced on the methods comparison. 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 weren't 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 labeling which allowed us to evaluate the performance with more confidence.

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. 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. This last set of images was highly relevant, mainly to have a 'control set' in which to compute the FP per image of the methods, and given that not all MCs are labeled in many MC-positive cases present in INBreast.

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).

First Pipeline: Advanced Image Analysis and Machine Learning.

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 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.

To compensate for the fact that the images didn't 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.

Hough transform (HT)

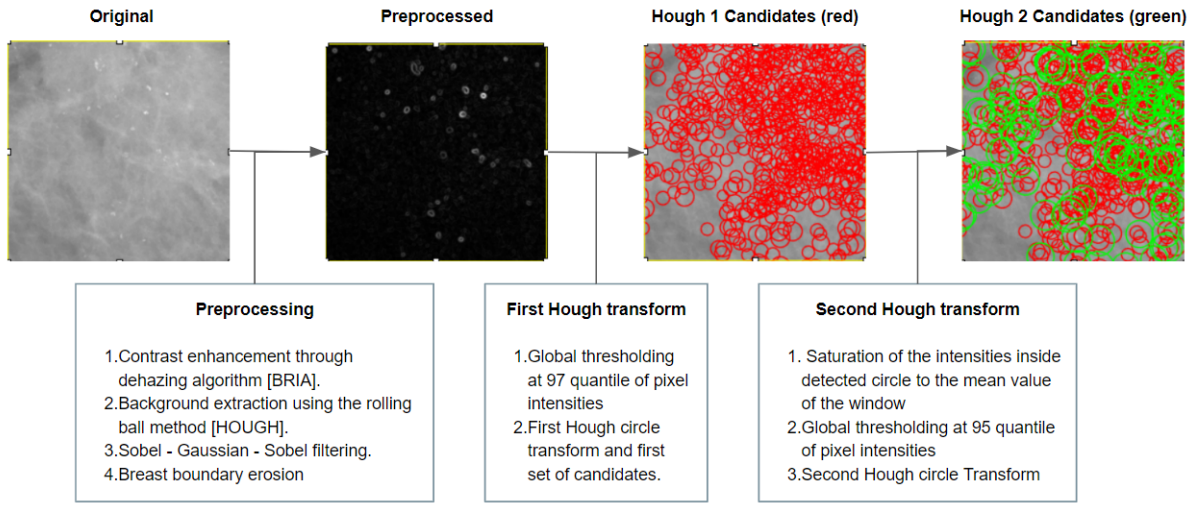


Figure 2: Hough transform candidate proposal pipeline. Plotted circles are enlarged by 10 pixels to be better appreciated.

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 most relevant from a clinical perspective and therefore the ones we were mostly interested in, have a round shape and with a proper preprocessing, they can be successfully detected.

We followed the general pipeline described in [HOUGH], 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]. 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 [Basile2019] 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.

Microcalcifications usually appear as relatively bright lesions, compared to the surrounding breast tissue or masses. However mammograms also contain a lot of fibroglandular tissue, which remains very bright even after the first preprocessing phase, thus making small microcalcifications poorly visible [Ciecholewski].

For this reason, after image preprocessing 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 [HOUGH], 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 processed patches to obtain a final set of candidates.

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

Hessian of Difference of Gaussians (HDoG)

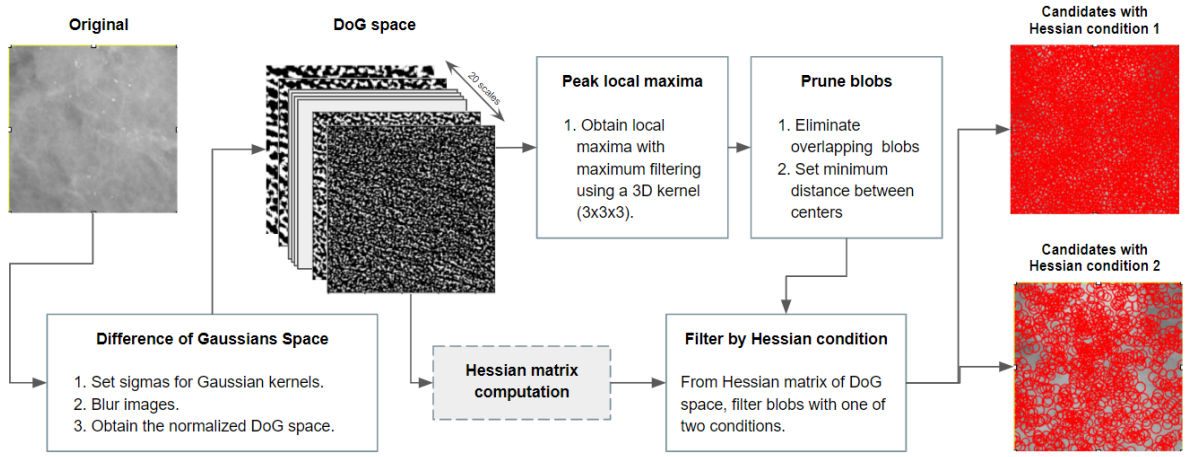


Figure 3: Hessian of Difference of Gaussian candidate proposal pipeline. Plotted circles are enlarged by 10 pixels to be better appreciated.

Following the general pipeline described in [HDOG], 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. Then, we found peak local maxima with 3x3x3 local windows around each point and then 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 candidates. The condition aims to describe MCs geometrical structure, being circular or tubular, through the eigenvalues of the Hessian matrix. The first condition [HDOG] uses the trace and determinant, and follows:

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

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

Inspired by [HDOG2], 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.

$$P_k^j(x,y) \geq T_k ,$$

$$\text{where } T_k = \frac{max(P_k^j)}{divider}, \; P_k^j = \prod_{m=j}^{j+1} \lambda_k^m \text{, and } k = 1, 2,$$

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.

Grayscale Morphology (GM)

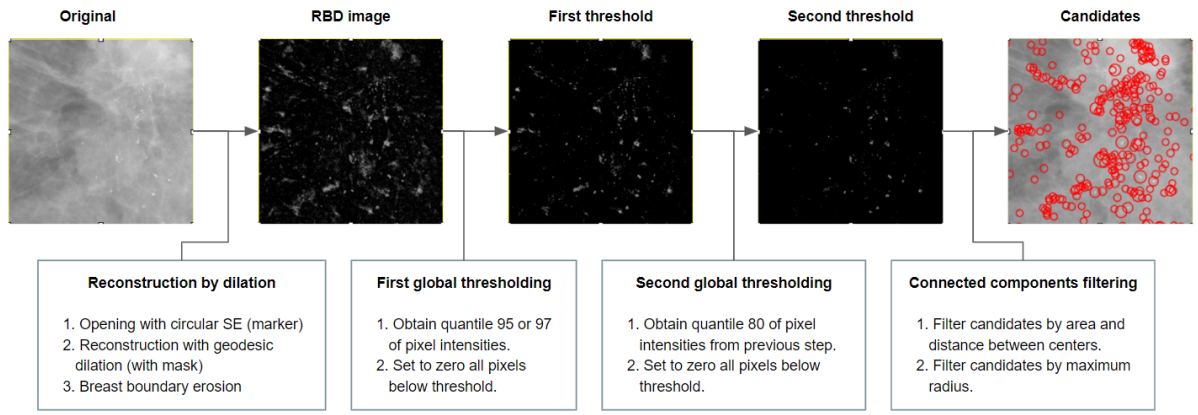


Figure 4: Grayscale Morphology candidate proposal pipeline.Plotted circles are enlarged by 10 pixels to be better appreciated.

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]. It further extends binary morphology by considering an image to be a non-binay 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:

$$[f \oplus b](x,y) = max < undercase(s,t) \in b > \{f(x+s, y+t)\}$$

Erosion:

$$[f \ominus b](x,y) = min < undercase(s,t) \in b > \{f(x+s, y+t)\}$$

Opening:

$$[f \circ b](x,y) = (f \ominus b) \oplus b$$

Closing:

$$[f \cdot b](x,y) = (f \oplus b) \ominus b$$

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 a second image, called mask. The mask acts as a constraint for the repeated dilations of *f*, the marker represents an image with high values (seeds) at the objects we want to reconstruct (MCs) and low ones elsewhere. In our case the marker was given by the opened version of the image. It flattened its intensity profile, removing MCs and placing lower 'tissue' values in their locations. We use the mask as the original image. After idempotence of reconstruction by dilation, we obtained an image where the general intensity profile is recovered, but the high flattened peaks of MCs are not. Finally, subtracting the last dilated marker from the mask produced an image that preserved these small round high-intensity structures, due to the circular kernel used during opening, while removing as much background tissue as possible.

In the same way as done in Hough Transform, we kept only detections 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 defined quantiles, 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.

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.

$$Sensitivity = \frac{TP}{TP+FN} \ , \ FPpTPpI = \frac{|FP|i}{|TP|i} \ ,$$

where $|\cdot|$ stands for cardinality

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 contains any target lesions, we label the candidate as TP, otherwise it is 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 such duplicates and all comparison is done assuming that each ground truth is matched only one time.

Moreover, for the final candidate proposal methods comparison we include two variants of the GM method (95 and 97 quantiles in the first threshold) and Hough detectors (first and second Hough transform) to give an idea of the influence of the configuration parameters on the results.

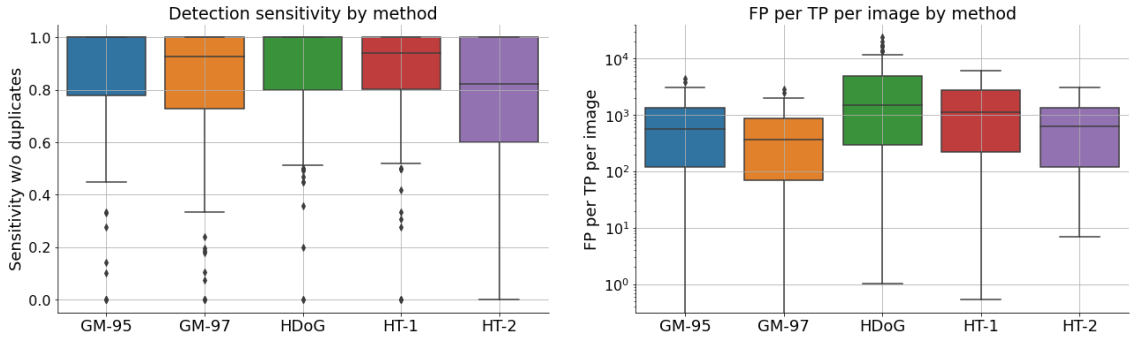


Figure 5: (Left) Sensitivity per candidate proposal method, (right) Log number of False positives per true positive per image for each candidate proposal method.

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.

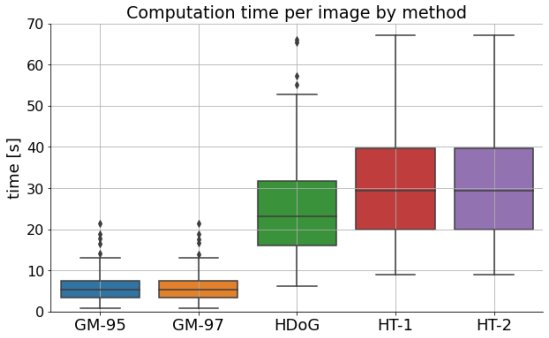


Figure 6: Computation time per image by method.

Finally, in Figure 6 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 GM-95 method was the best, as it retained one of the highest sensitivities while being the fastest and having fewer FP.

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 of the breast border,

ignored by two of the three methods, and (b) highly compromised tissue by various lesions. (See Figure 7 for examples).
Finding (a) was very interesting because it proved that it is possible for MCs to be found in the breast border, even though it is unlikely. 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 keep the boundary filtering.

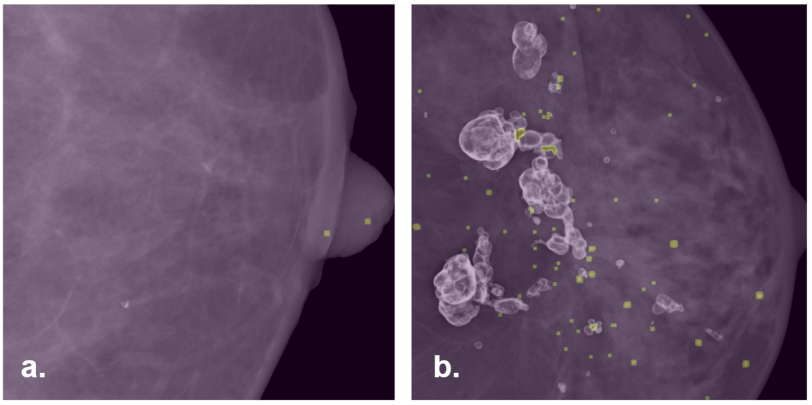


Figure 7: Example images of cases with (a) microcalcification in the nipple and (b) highly compromised tissue, with overlapped detections (green circles).

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.

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 each candidate patch, see table 1 for the complete list.

Gabor filter features: First we filtered the complete image with six different gabor kernels [GABOR KAHN]. The selection of a reduced set of features aimed to keep computation time low and the specific kernel choices (Figure 8) were meant to achieve direction invariant texture features. Then, from each candidate patch and using its bounding box coordinates, we obtain first and second order statistics. See Table 1, for the complete list of extracted features.

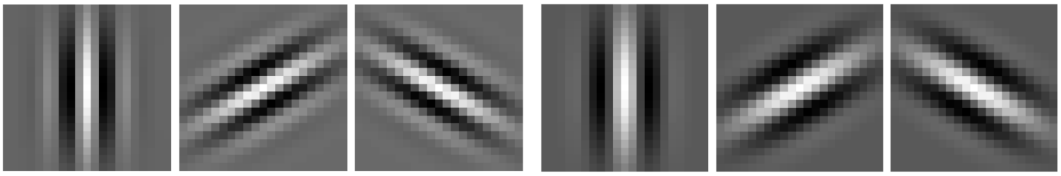


Figure 8. The six gabor kernels we use to filter the images.

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 aim to describe the local geometry of the patch in terms of scale and orientation (see Figure 9).

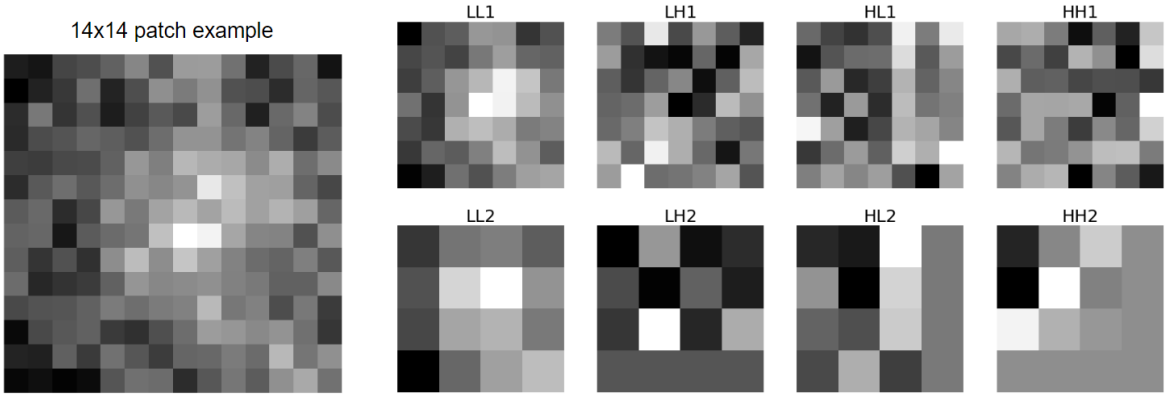


Figure 9. Decimated two-level decomposition of a patch example using 2-D discrete wavelet transform.

Following [WAVELET FANIZZI 2020], from each of the eight subpatches we extract first and second order statistics (see Table 1). Furthermore, from the first level detail features (LH1, HL1, HH1) we obtain their gray-level co-occurrence matrix (GLCM) [WAVELET FANIZZI 2019] using an offset of 2 and using angles of 0°, 45° and 90°. With this we focus to describe textural patterns spanning spatially inside the patch. From them we get statistical descriptors (see Table 1). The final set from wavelet features is of size 92.

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 kernels), compute: Energy Mean Standard deviation Skewness Kurtosis	From each of the eight wavelet decompositions compute (56 features): Mean Skewness Standard deviation Kurtosis Entropy Uniformity	Min Max 10th quantile 90th quantile Mean Median Standard deviation Energy Entropy

	Relative smoothness From LH1, HL1, and HH1 decompositions, obtain three GLCM and compute (36 features): Correlation Homogeneity Contrast Dissimilarity	Uniformity Skewness Kurtosis Interquartile range Range Mean absolute deviation Robust mean absolute deviation Root mean square

Haar-like features: Following [Bria 2014], three groups of haar-like features were extracted from every patch, see figure N. 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] (sets (a) and (b)) and the variation presented in [Lienhart 2003] for the rotated case.

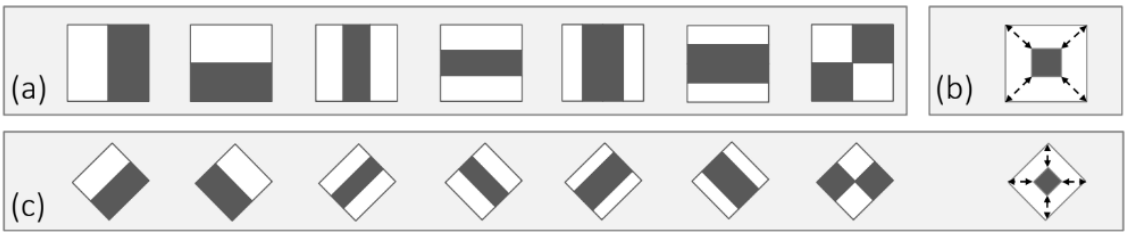


Figure 10. Three groups of Haar prototypes used. Image taken from [Bria2014].

The originally proposed version of the haar 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 of the generated data becomes 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, facing directly the curse of dimensionality.

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, while reducing the computation time and to better condition the learning problem.To select the most discriminative features we use the embedded Random Forest (RF) Classifier approach, and the feature importances computed in it by using the Gini index.

In order to handle the huge volume of data, the feature selection pipeline was the following. First we extracted all the haar 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, in order to focus our selection in the *blob-like* concentric feature (present in the group b and c), 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, in the validation set we evaluated the influence of the number of features by assessing the classification performance of a Random Forest - measured by the area under the ROC curve (AUROC) and the area under the precision recall curve (AUPRC) -. Going from 25 features up to 3000, and by doubling the features number at each step, we trained a RF model for each number of predictors with a five fold CV.

For GM and HT detected candidates the and obtained the following results:

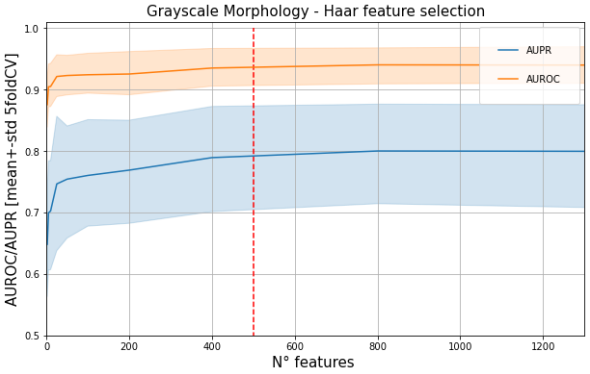


Figure 10: Effect of number of haar features on the classification performance of Random Forest in the validation set of GM candidate proposal method.

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. Also, we can see that the AUPRC curve reaches a value of ~ 0.9 at the selected threshold.
The AUROC values can be misleading sometimes due to the high imbalance in the dataset, since the false positive rate (1-Specificity) can be low in percentage terms, but because we have a very large number of truly negative samples, that small percentage could mean an unacceptable number of FP in our prediction. 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.

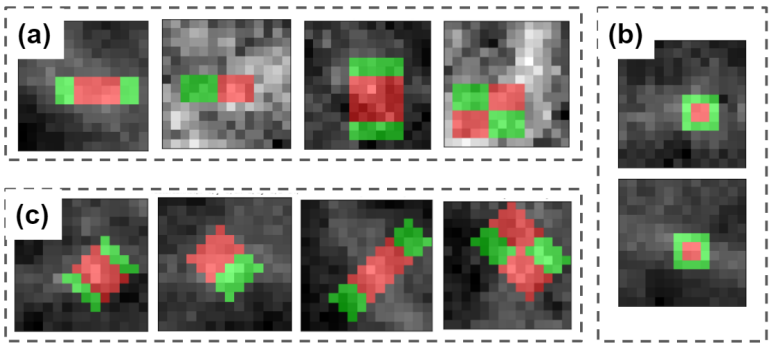


Figure 11. Sample features from the sets in Figure 10, plotted over diverse MC patches.

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 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 left each sample with their inherited 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.

Performance 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 in a clinically not useful scenario, the curves are constrained to the range [0, 50] on abscissa's axis.

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 (SVC).
XGBRF is a decision tree ensemble learning algorithm, similar to RF. Both of them 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 ensembling technique, in which each new tree that is added to the ensemble is built following an optimization procedure of the residuals values.
SVC belongs to a different class of ML approaches. It is a supervised learning algorithm that separates training samples belonging to different classes by mapping each of them to a point in the feature space in a way that maximizes the gap between points of 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.

Model	Features	AUROC	AUPR
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**
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**
SVC	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**

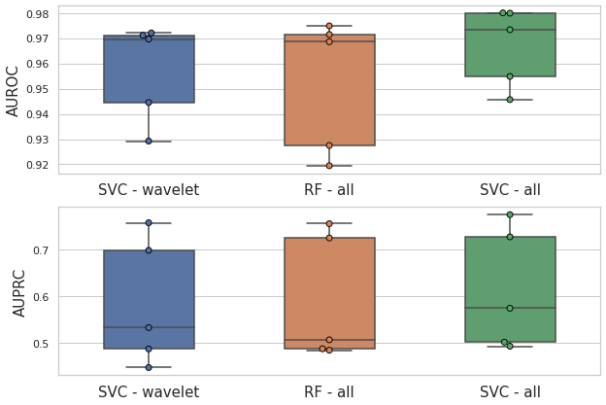


Table 2: Performance evaluation of different classifiers in the different sets of features.
Figure 12: AUROC (up) and AUPRC (down) of best classifiers from Table 2, for validation set.

From Table 2, we conclude that the results obtained for each model across all the different sets of features showed that using all features always led to the highest performance. Also, as is highlighted in yellow, the best three performances were obtained with Random Forest and SVM using the set of all features, and using SVM and only the wavelet features. Even more, 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 better performance. These results highlighted the power of SVM classifiers. A more detailed comparison of the ten fold CV results is depicted in Figure 12.
Then considering all this evidence, the SVM classifier trained with all features is chosen as the best performing one.

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 A.
Having found the best model and its best hyperparameters, we then explored the benefits of exploiting hard examples among the set of proposed candidates. Following [Bria2014], having a first stage in the pipeline that generates a big 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.

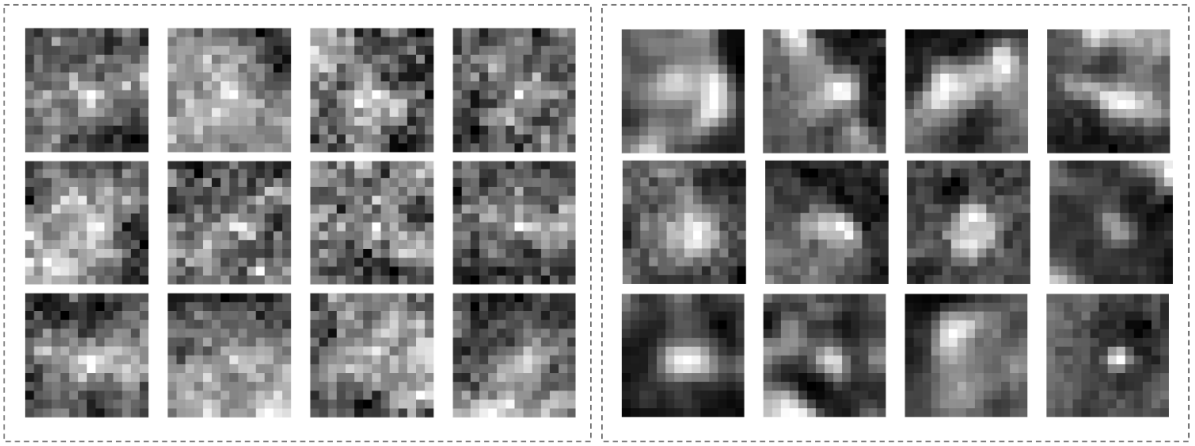


Figure 13: Examples of *easy negatives* (left), and hard negatives (right) candidates.

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 tuning it with hard negatives (*hard negative mining* approach). The second one, inspired by [Bria2014] 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 examples were identified. The second model (fine tuned version or second cascade stage) was trained on train set hard negatives to avoid data leakage.

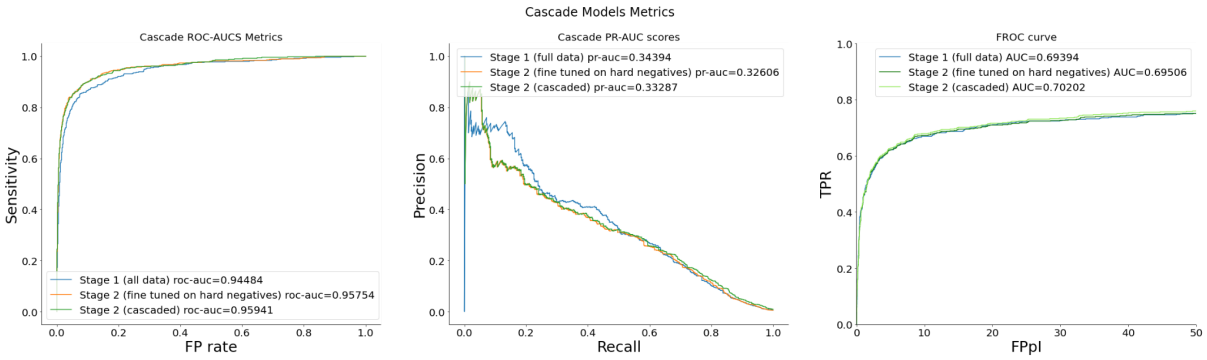


Figure 14: Results on validation set, of different ways of exploiting hard negative examples, including original model, hard negatives finetuned one and cascaded approach.

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, the FPpI were reduced while the sensitivity was preserved.

With the evidence obtained, 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.

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 [HDOG] the authors combined image analysis methods and DL models to detect MCs; with the same objective in [BRIA], the authors implemented deep learning cascaded algorithms to handle the imbalance of the problem; in [Zhang2021] the authors use 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. Later in this section we reviewed some of the methods that can be used to deal with this problem

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 and 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 tasks.
As shown in [Christos], 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.

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 obtained information 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” look 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 obtain 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 didn’t have a significant influence, so we ignored it. Thus, the final output of the pipeline were the detections with their associated boxes of 14x14 and confidence scores.

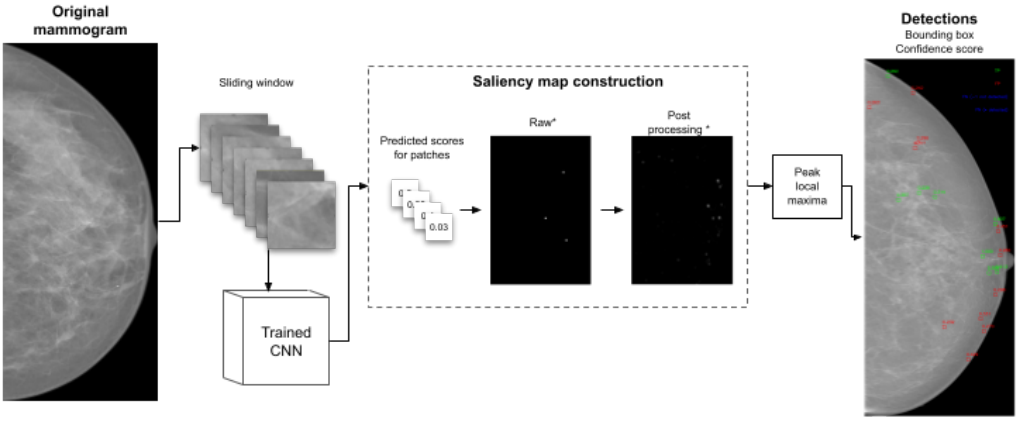


Figure N. Detection by classification pipeline. *Zoomed in for clarity

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 their results in the second deep learning pipeline and are 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. This resulted in a train set containing N patches with distribution N and a validation set with N cases with distribution N. We selected this first patch size in order to have a reasonable image size, and matching one of the standard ones in computer vision, that allowed us to 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 (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 (He et at, 2016), DenseNet121 (Huang et al, 2017), EfficientNet from B0 to B7 (Tan et al, 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 variations on the negative balancing, by downsampling a different set of negative examples in each epoch and matching the desired 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 were:

	F1-score	AUROC
resnet50_01	0.755184	0.93
resnet50_05	0.709174	0.9481
densenet121_01	0.708066	0.943

Table N: 3 Best performing models on 224x224 patches. Check Appendix B for training details.

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, didn't 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 didn't match them. None of the previously commented saliency maps variations could increase the models performance. See figure N (FIGURE OF FROC COMPARISON DET BY CLASS).

Following the literature [BRIA dl comparison with Cascade], 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 wouldn't 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 N), 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 N 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.

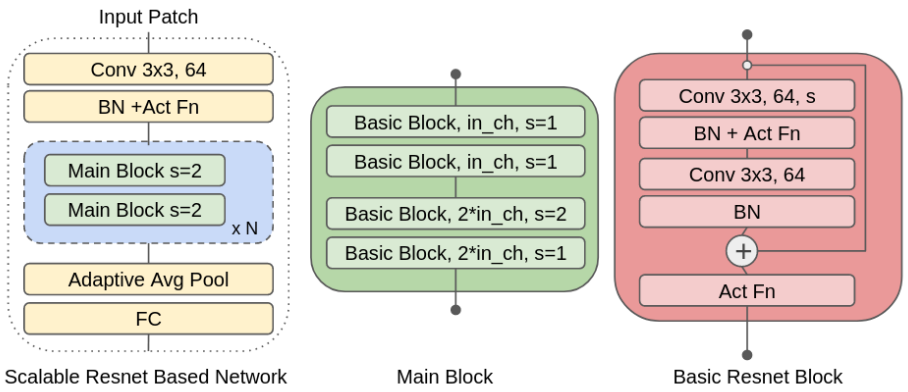


Figure N. Implemented scalable ResNet-based model, where the blue box is the customizable downsampling stage.

We trained these three networks, again 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 N.

Model	f1	auroc	avgpr
64_ResnetBased_03	0.4423	**0.993332**	**0.8477**
32_ResnetBased_05	0.4251	**0.993204**	**0.895**
16_ResnetBased_07	0.424	**0.993714**	**0.9009**

Table N. Best performing models for each patch size, see appendix B for training specification details.

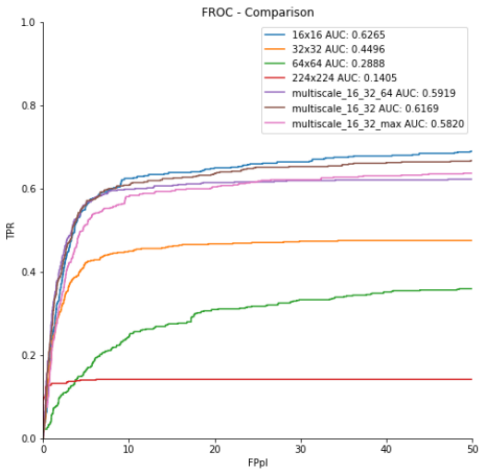


Figure N, comparison of FROC curves of the different variants of detection by classification pipeline on validation set.

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 N .

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 wasn't solved. Trying to improve these results and also inspired by [BRIA], 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 N. Finally, figure N shows the significant computation times differences that existed among the proposed variants.

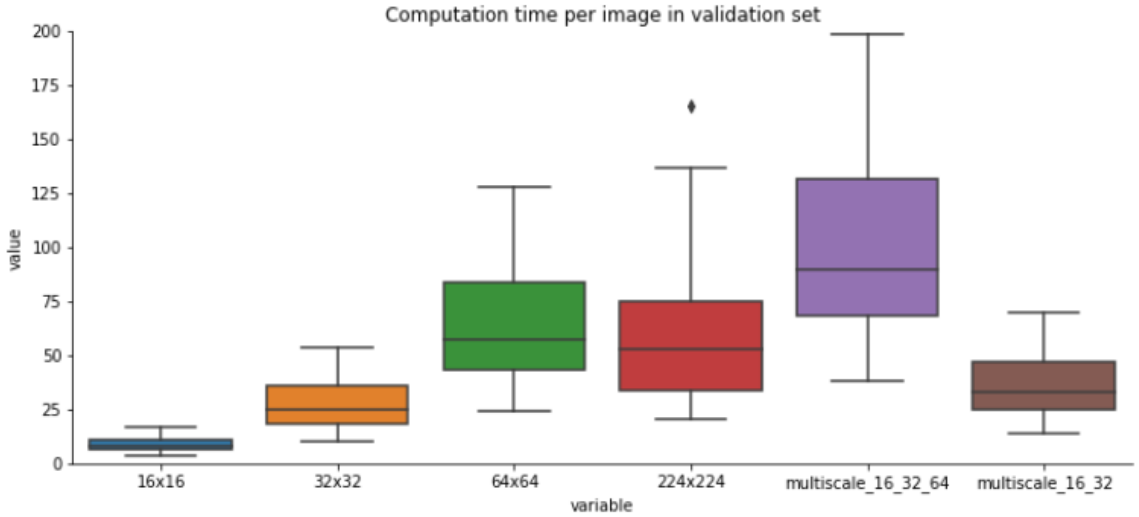


Figure N, computational time per image of different variants for detection by classification pipeline.

With all this evidence 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 9.

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) [2014], You-Only-Look-Once (YOLO) [2015], Single Shot Detector (SSD) [2015], Region-based CNNs [2013] and its follow up improved versions Fast-R-CNN [2015], Faster-R-CNN [2015], and Mask-R-CNN [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 (Girshik, 2015). 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 network will classify ROI feature maps and output class prediction scores while a separate regression network will use propagated feature maps to generate final bounding box coordinates.

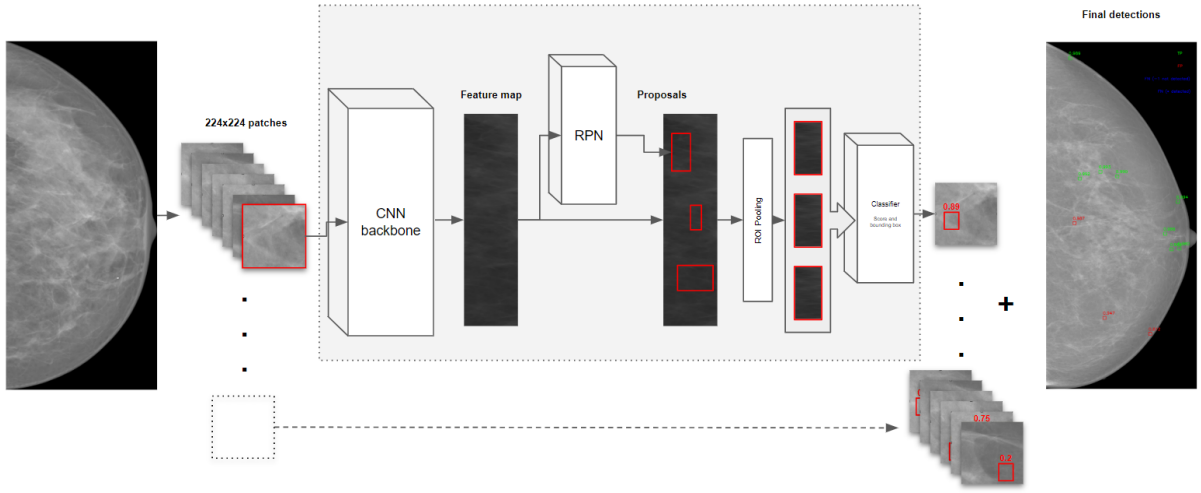


Figure N. Faster-R-CNN general architecture

Applying 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-RCNN 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, EfficientNet-B0 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. Feature map extracted from the CNN backbone and passed to RPN was always obtained from the input to the last block of FC layers.

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 (due to maximum size of lesions of interest of <1 mm) centered on each calcification, there was no need for this redundancy, and even though we explored different sizes of anchors, we sticked to the size 14x14 and the aspect ratio of 1:1.

Experiments varying Faster-R-CNN parameters and corresponding metrics can be seen in Appendix C. 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 isn't 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 NN, and the ground truth bounding boxes were defined as 14x14 squares around the center of every lesion inside the patch.

At inference time, whole image detection was performed in a sliding window manner, on patches 224x224 with a stride of 200. Having a small overlap between patches allowed the model to better account for lesions in the borders. 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:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (5)$$

where $r_1, r_2, \ldots, r_n$ 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 $r' \geq r$:

$$p_{interp}(r) = \max_{r' \geq r} p(r') \quad (4)$$

The 3 best performing models on the aforementioned metric are present in the table M below:

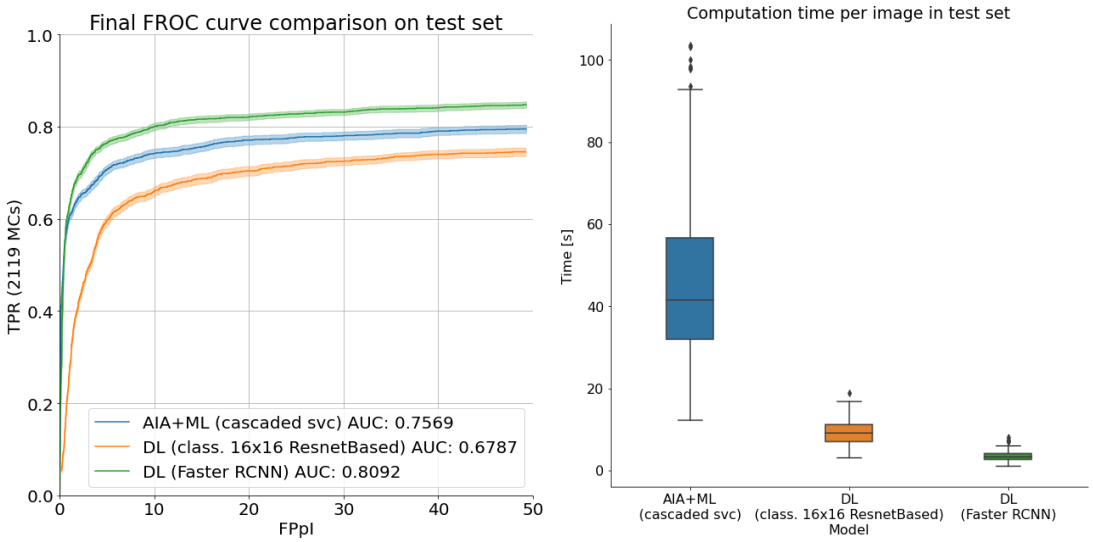
Model	AR_IoU_0.50_0.95_all_mdets_100:	AP_IoU_0.50_all	
d_resnet50_00	0.7781893004	0.913054	
d_resnet50_06	0.789787	0.905355	
dns121ptr0	0.748859	0.90434	

Table N: 3 Best performing models on 224x224 patches with Faster-R-CNN model. Check Appendix C for training details used.

Results and Discussions

In this section we compared performances of the best models described in previous sections. The comparison is done over the full test set and only after all the models were trained and tuned on train and validation sets. For the first pipeline (AIA-ML) we selected the cascaded framework with the final SVC trained on hard negatives and with discarded easy negatives. 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 ResNet50 backbone pretrained on the classification task with anchors of 14x14 pixels.

The main metric used to compare performance of aforementioned models is Area Under the FROC curve constrained to a maximum of 50 FPpI. Another important metric for us was the detection time per image, as we optimized all of our models for the highest speed of execution.

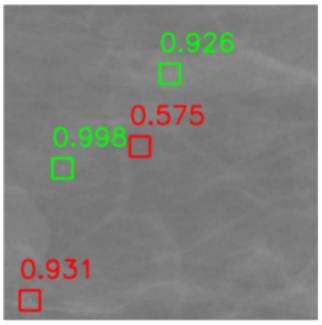


In the figure above, we can clearly see that the Faster-R-CNN model considerably outperformed other models with the FROC-AUC of 0.8124. It is due to the fact that this is a DL model specifically designed for detection tasks, that we have successfully adopted to our data. It was directly optimized with the detection objective and transfer learning was used to speed up training and improve final results, which wasn't the case for other models. All of them had separate stages in the detection process that had multiple parameters that were

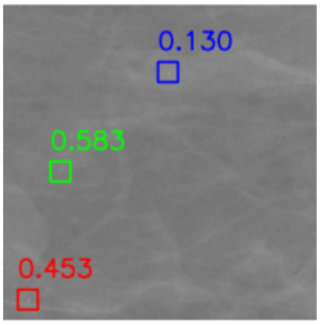
either manually adjusted or optimized separately. Faster-R-CNN was the only one that used an optimization procedure to fine tune its internal parameters for the whole detection process, from image to set of detections. Moreover, we can see that the ML approach with SVC outperformed our classification-based 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 that both DL approaches perform substantially better. The main reason for this is that the feature extraction (FE), being the most time-consuming stage of all of the models, in DL is implemented through extremely efficient convolutions, while for ML approach it is done through a large variety of methods. In addition to that, before the FE, a candidate proposal step is required, which is completely independent of the ML stage. A small difference in the detection time between two DL approaches is caused by the need of detection by classification model to use much smaller patches than Faster-R-CNN to obtain a high resolution saliency map, and by the need of additional transformation of the map into a set of detections.

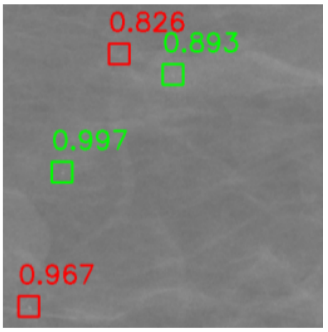
Looking at a concrete example of detections of all three models on the image N we can further expand on the differences between them.



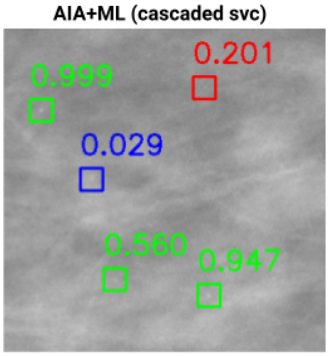
AIA+ML (cascaded svc)

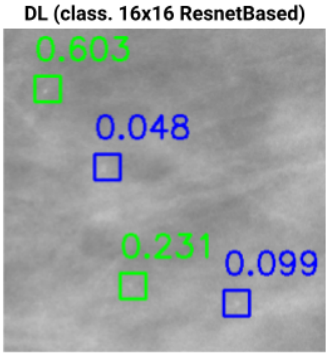


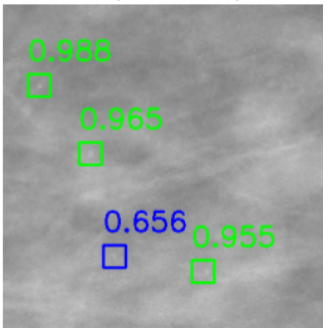
DL (class. 16x16 ResnetBased)



DL (Faster RCNN)







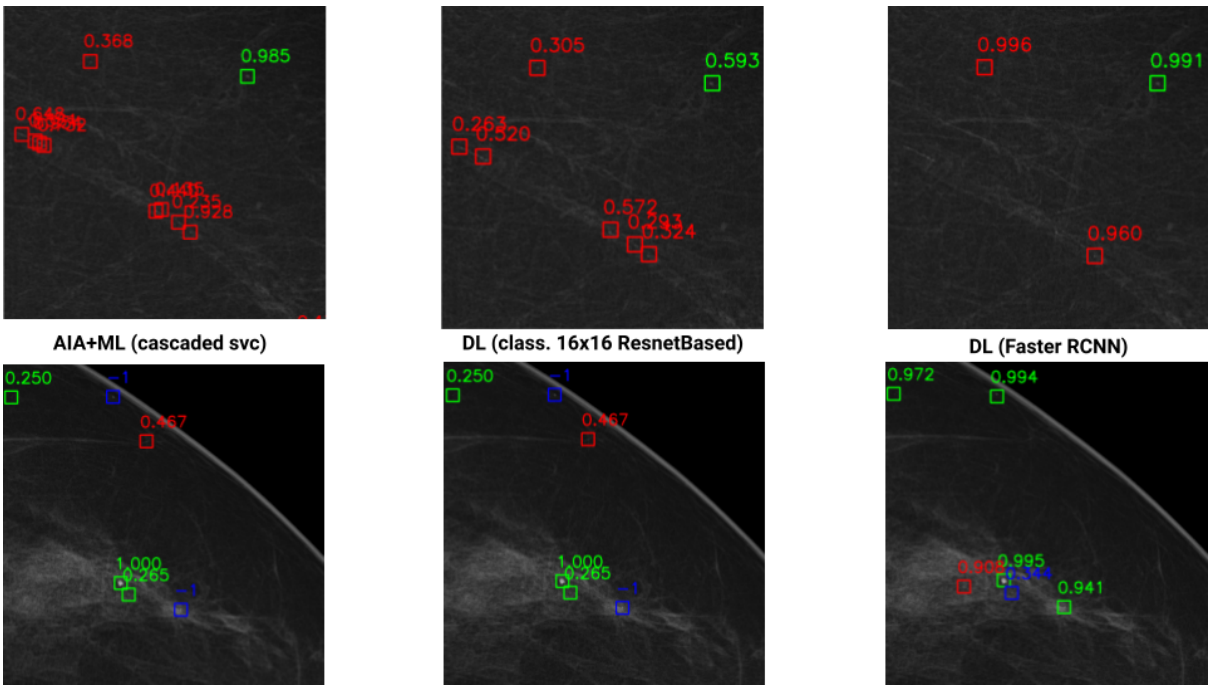
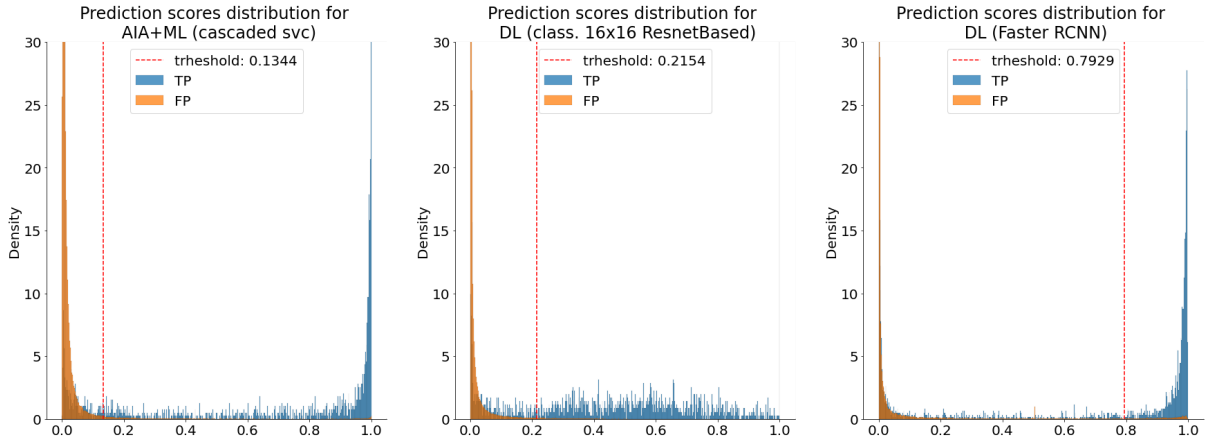


Image N.



Conclusions

In spite of the intrinsic challenges we faced for this detection problem, the high resolution of the images and small and dispersed regions of interest, we confronted many problems with the dataset. We realized the importance of having a large dataset with high quality images, but as equally important, to have a properly annotated one. This difficulty not only affects the training task in supervised algorithms but also the accurate way to evaluate their performance. We were constantly challenged to search for ways to find and maintain quality training samples for our models, but also we had to arrive at decisions that would trade-off a proper model training versus discarding some amount of erroneously labeled data.

Our results show that use of pretrained models is still beneficial even when there is an objective or data difference

A last comment on the use of the information provided in the original dataset is related to pectoral muscle that is common to all methods developed. 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 the pipeline in order to mask-out all detections in that region. Before implementing any approach, we ran several experiments using the provided masks, noticing that there wasn't any significant improvement in the methods performance by the inclusion of this filtering.

Appendix A: Final parameters for different methods

- Parameters for candidate proposal methods:
    - GM
    - Hough parameters
    - HDoG

- Parameters for SVM hypertuning using gridsearch CV

    - *C*, inversely related to the tolerance of samples entering into the decision function margin, from {1, 10, 100}

    Tree different *kernels functions*:

    - Linear
    - Polynomial
        - *Degree:* {3,5,7,10}
    - Random basis function (RBF)
        - *Gamma*: {0.01, 0.1, 1, 'scale'}

    The experiments showed that a random basis function kernel, with *gamma*=0.1 and C=10 led to the best performing model.

Appendix B: Training experiments for Detection by Classification

- Original architectures training experiments

[clean table with experiments results]

- Custom architectures training experiments

[clean table]

Appendix C: Training experiments for Detector Faster-R-CNN

[clean table with experiments results]

Partial Bibliography

Hough
https://doi.org/10.1016/j.ejmp.2019.05.022
https://www.physicamedica.com/article/S1120-1797(19)30130-9/fulltext

HDoG
https://arxiv.org/abs/2102.00754
https://doi.org/10.48550/arXiv.2102.00754

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209–249. https://doi.org/10.3322/caac.21660

Khan, S., Hussain, M., Aboalsamh, H., & Bebis, G. (2017). A comparison of different Gabor feature extraction approaches for mass classification in mammography. *Multimedia Tools and Applications*, *76*(1), 33–57. https://doi.org/10.1007/s11042-015-3017-3

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

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

Tan, M. & Le, Q.. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, in Proceedings of Machine Learning Research 97:6105-6114 Available from https://proceedings.mlr.press/v97/tan19a.html.