Neural Networks Project Report

Cicero Edoardo, Giunta Daniele

Based on: "Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening" (https://arxiv.org/abs/1903.08297)

Introduction

In previous years the task of breast cancer detection was executed by radiologists looking at patients screenings checking if there was a finding or not and classifying it as malignant/benignant with further analysis. Neural networks have been introduced in this field as a kind of "second mammography reader" for the radiologists, but they should not work separately: in fact, it has been seen that a hybrid use of the neural network prediction and the radiologist's one can give more accurate predictions about the findings.

The goal of our project is to build a neural network model based on a CNN that from breast images associated to selected patients predicts the presence or absence of malignant or benign cancer on those screens. We try different model architectures and inputs, introducing also heatmaps generated from the images themselves as supplementary training data. Our work is based on the paper mentioned above, and some scripts for preprocessing phase and the one for the ResNet-22 structure have been taken from the researchers' github repository at (https://github.com/nyukat/breast_cancer_classifier) since they performed a specific work of image manipulation to generate the input of their models. (Our work and code is also available here https://github.com/EdoardoCicero/Neural-Networks-Project)

Data

Data related to medical field are not provided in such an easy way: indeed this kind of information is based on real patients, hence many datasets may not be public for privacy reasons (the NYU dataset, the one used by the researchers who worked on the paper is not available for public use for example). In addition, many of the public ones are not well distributed in terms of classification as needed for this particular task: datasets like DDSM and CBIS-DDSM (a subset of DDSM) contain only few samples of data with a classification for every single breast image for every patient, and most of the times there are less images than needed. In our case the dataset we want needs 2 images of a breast per view, namely MLO (mediolateral oblique) and CC (craniocaudal) for a total of **4** images per patient. DDSM dataset is labelled per patient and not per screen, and this is the reason why we discarded it. Instead, all the data that have been used for training and

validation comes from the INBreast Dataset, which contains a total of 410 total images (in dicom format) from 108 patients with a number of images per patient between 2 and 8; only **73** among them have been selected in the end, because they had exactly 1 image per view, for a total of 4 distinct images per patient (i.e. with no duplicates); compared to the NYU Dataset, it can be easily spotted how much they differ: in fact, the NYU Dataset includes 229,426 digital screening mammography exams (for a total of 1,001,093 images) taken from 141,473 patients. From the chosen dataset the images were converted in png format with a bit-depth value of 12.

While the authors relied to pathology reports from biopsies to extract labels indicating whether each breast of the patient had benign and/or malignant findings, we exploit the BIRADS classification for each mammography view; the way in which we extracted labels from them was empiric: in fact, we decided to map the BIRADS label in the following way:

- birad 1 (no cancer) → label 1
- birad 2,3 (benign, probably benign) → label 2
- birad 4,5,6 (low/medium/suspicious malignant, very high probability of malignant, known malignant) → label 3

Data distribution of all images is shown below.

	Left	Right	ToT
Benign	77	94	171
Malignant	44	38	82

In the end, we manage the output of the network in such a way that we can represent the presence/absence of a benign finding for the left/right breast and a label that can represent a presence/absence of a malignant finding for the left/right breast, that leads to an output of this kind:

[L_Benign, L_Malignant, R_Benign, R_Malignant]

Our goal is to produce four predictions corresponding to the four labels for each exam. As input, we take four high resolution images corresponding to the four standard screening mammography views. We crop each image to a fixed size of 2677 × 1942 pixels for CC views and 2974 × 1748 pixels for MLO views.

Preprocessing

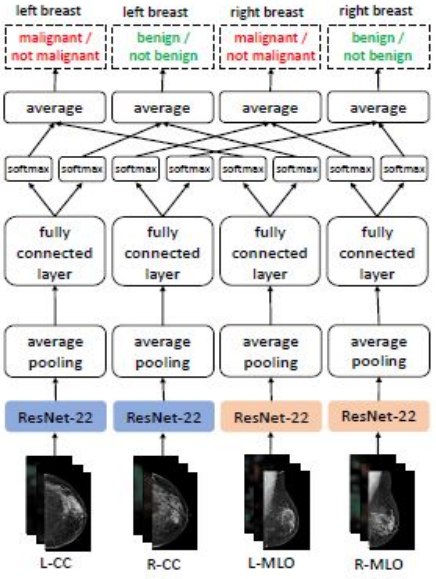
Preprocessing is an important step in the processes of building and training a neural network because it is used to clean, extract and/or highlight some features from data that can help the neural network to better generalize the information passed to it (making it noisy for example) or to focus the attention on just part of it. In this work we use the researchers' scripts to realize both of them: the images are initially cropped, and some informations are acquired from them; in particular, before passing the images to the net, the images are augmented, namely the optimal center of each image has been computed in order to add some random noise in a certain window localized around that point, then the images are cropped (again), flipped

and normalized. The flipping occurs only on left breast images to let the net see all screens from the same side.

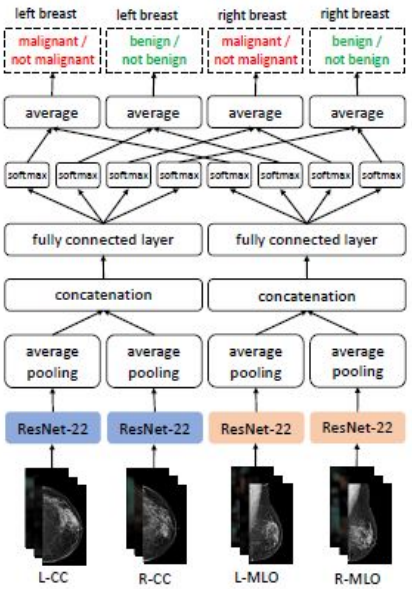
Models architecture

The models implemented share the same core structure as described in the paper: four columns, each based on the ResNet architecture that outputs a fixed-dimension hidden representation for each mammography view, and two fully connected layers to map from the computed hidden representations to the output predictions. We use four ResNet-22 columns to compute a 256-dimension hidden representation vector of each view. The columns applied to L-CC/R-CC views share their weights, so/exactly as L-MLO/R-MLO. We concatenate the L-CC and R-CC representations into a 512-dimension vector, and apply two fully connected layers to generate predictions for the four outputs. We do the same for the L-MLO and R-MLO views. We average the probabilities predicted by the CC and MLO branches of the model to obtain our final predictions.

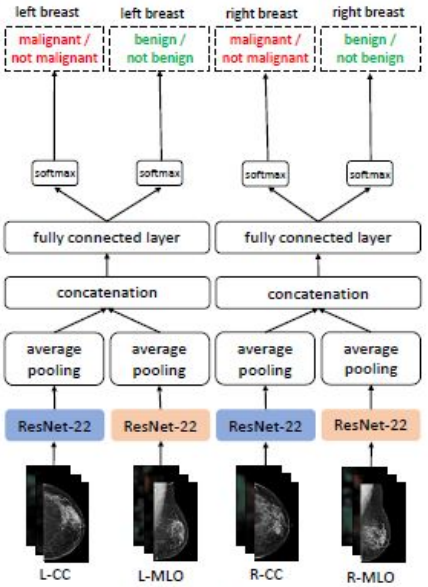
Other models have been realized with the precedent structure: they are the so-called "image-wise", "breast-wise" and "joint" models. Still, there are some differences among all the models: the "image-wise" model has a branch for each image, the "breast-wise" model has separate branches per breast (left/right), while the "joint" model has a single branch. The architecture of each model is reported below.



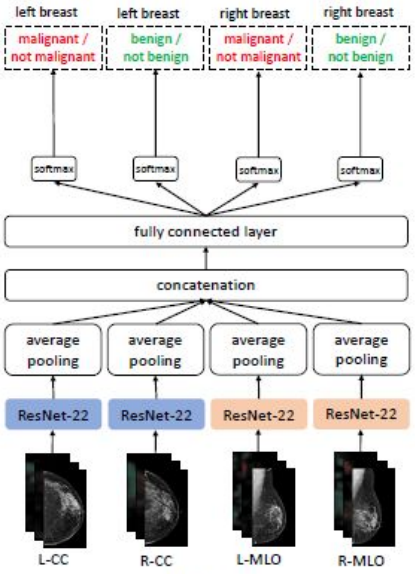
(b) image-wise



(a) view-wise



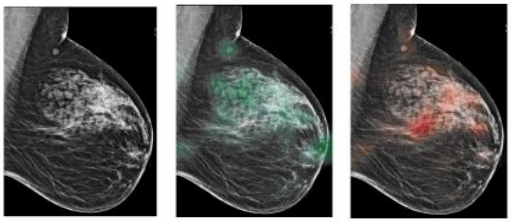
(c) breast-wise



(d) joint

As it has been said before, the inputs of each model are 4 images, one per view (1 for L-CC, 1 for R-CC, 1 for L-MLO, 1 for R-MLO); in order to improve the models performances, the authors decided to add other 2 images per view for each patient: the benign heatmap over the original image and the malignant heatmap over the original image, each one of 256x256 pixels. In this way, they had a total of 3 images per view.

The heatmap images are taken from the original ones: in them, it is highlighted the location of the benign or malignant tumoral mass, in green or red respectively. An example of heatmap images is given below (from left to right, original image, benign heatmap, malignant heatmap).



Images: benign heatmap (green) and malign heatmap (red) over breasts

The loss function that we use is a sparse-categorical crossentropy averaged across all the four output labels; the L2-regularizer has been applied to our model weights with a coefficient of $10^{-4.5}$. The stochastic gradient descent with the Adam optimization algorithm has been used.

The parameters used to train all the models (both for "image-only" and "image-and-heatmaps" modes) have been chosen as follow:

- learning rate: 10^{-5}
- batch size: 2 (half of the authors', due to low computational resources)
- epochs: 20
- validation split: 0.1
- metrics: loss, accuracy, AUC for benign case, AUC for malignant case.

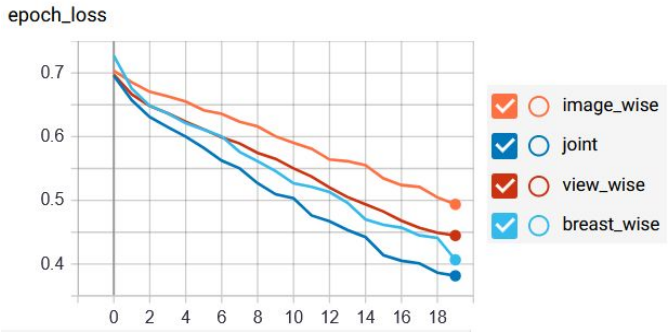
The cross entropy has been used as loss, averaged for all four labels, while the AUC (Area under ROC curve) metric has been used to evaluate the performances of the main model and of its variants.

The AUC, also known as "area under the roc curve", is a commonly used metrics in the medicine field; it tells how much a model is capable of distinguishing between classes. In other words, the higher the AUC, the better the model is at predicting 0s as 0s (that is, patients with no disease) and 1s as 1s (that is, patients with disease); it is a value that goes from 0 (bad performances of the model) to 1 (excellent performances of the model).

Results and conclusions

In our implementations, the thresholds used to compute the ROC curve have been set to a very high value to compensate the imbalance among benign and malignant images (to be more precise, the threshold values have been set from 0.55 to 0.95, with a step of 0.5). We tried leaving the threshold values as default but results were not so significant given that the two AUC were really close to each other, and this is not representative of the model performance because the data is unbalanced. Indeed, as mentioned in Data section, we got a dataset with more benign than malignant images, having many labels like [0,1,0,1] or [0,1,0,0] producing predictions with low probability in mostly cases in the first and third position of the output vector that classify left malignant/not malignant and right malignant/not malignant. Our classifier is then able to better classify benign findings than malignant ones as shown in the table below. (AUC B: AUC for benign, AUC M: AUC for malignant)

	AUC B	AUC M
image-only (view-wise)	0.795	0.675
image-only (image-wise)	0.773	0.618
image-only (breast-wise)	0.825	0.723
image-only (joint)	0.842	0.746
image-and-heatmaps (view-wise)	0.857	0.762
image-and-heatmaps (image-wise)	0.875	0.680
image-and-heatmaps (breast-wise)	0.871	0.821
image-and-heatmaps (joint)	0.873	0.825



The models have been trained with all the samples we had, and the results were not so good in terms of loss (reaching a minimum of 38% on joint model trained on images and heatmaps) mostly due to fact that the implemented architectures are very complex (dealing with Deep-NN with CNN) so the training phase was based only on 20 epochs per each model due to low computational resources.

Some parts of the models architecture have been borrowed from the github code of the researchers; in particular, we took from their work the script to convert the images from .dcm format to the .png one, the part of code in which the optimal center is computed for each image, the code used to crop the images, and finally the scripts that build the ResNet-22 that we reimplemented. We chose tensorflow, in particular tf.keras, to implement the feeding sequence of input data and reimplement the models described in the paper, and since the original code was written with the pytorch framework (with few scripts in tensorflow), we reimplemented it in tensorflow.keras.