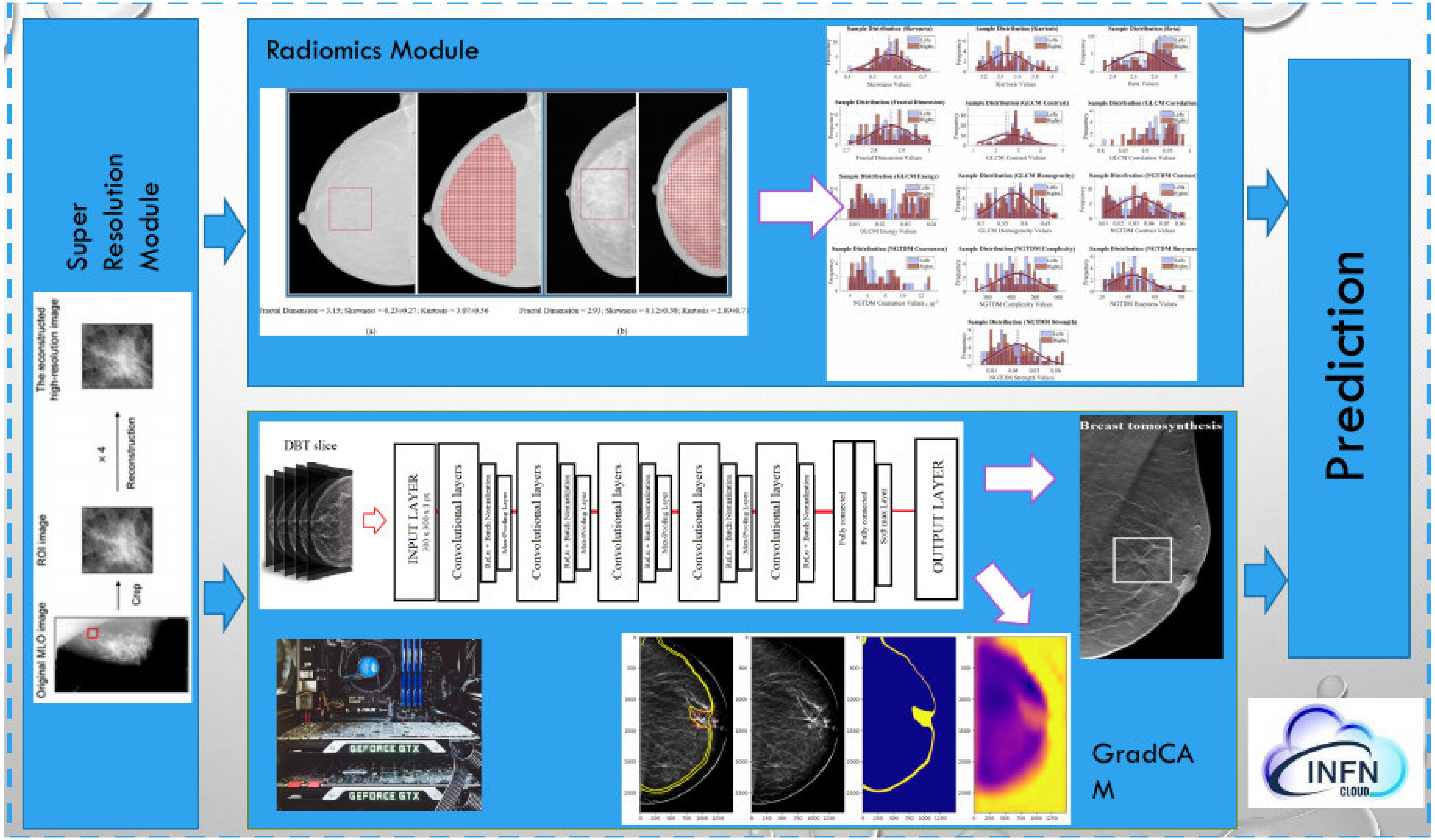
A python framework for classification of DBT slices through Deep Learning

Gianfranco Paternò
Istituto Nazionale di Fisica Nucleare
Sezione di Farrara

paterno@fe.infn.it

Radiomics in Digital Breast Tomosynthesis



DL for BDT classification

WP1 **Challenge I: no-so-big data**

Strategies for efficient learning with limited data samples.

Evaluation of robustness and reliability of trained models.

WP2 **Challenge II: explainable AI (XAI)**

Make AI results understandable to humans.

Which image/data features were relevant to make a decision?

a) Radiomics + Machine Learning

b) Deep Learning

WP3 **Applications to real-world data samples**

Practical medical data analysis use cases on available samples (public data, private collections, integration of both), where the analysis approaches a, b or both are implemented, and the challenges I, II or both are encountered.

Implementation, test and validation in collaboration with colleagues working in Clinical context

DL for BDT classification: The starting point

Physica Medica 83 (2021) 184–193

Contents lists available at ScienceDirect

Physica Medica

journal homepage: www.elsevier.com/locate/ejmp

A deep learning classifier for digital breast tomosynthesis

R. Ricciardi [a,b,1], G. Mettivier [a,b,*,1,2], M. Staffa [a,3], A. Sarno [b,4], G. Acampora [a], S. Minelli [c], A. Santoro [c], E. Antignani [c], A. Orientale [d], I.A.M. Pilotti [d], V. Santangelo [d], P. D'Andria [d], P. Russo [a,b,5]

[a] Università di Napoli Federico II, Dipartimento di Fisica "Ettore Pancini", I-80126 Napoli, Italy
[b] INFN Sezione di Napoli, I-80126 Napoli, Italy
[c] A.O.R.N. "Antonio Cardarelli", I-80131 Napoli, Italy
[d] A.O.U. "San Giovanni di Dio Ruggi D'Aragona", I-84131 Salerno, Italy

- Orignal DBT slices were pre-processed, resized to 300 x 300 px and fed independently to the CNN.
- CNN trained on the full dataset (CC), but with slices of validation patient present also in the training set.

- CNN developed and trainend in **Matlab**
- Accuracy and ROC_AUC reached about 90%.

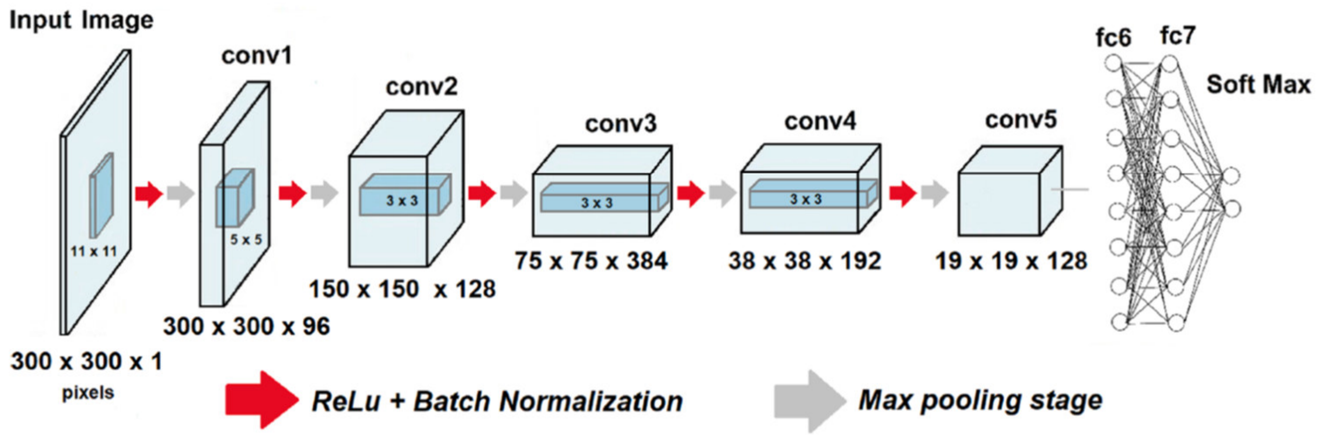


Fig. 4. Deep Convolution Neural Network architecture developed for this work. It is made up of 24 convolution levels: 1 level input, 5 convolutional levels, 2 fully connected classification levels and finally 1 softmax level immediately followed by an output level.

DL for BDT classification: The aims of the new work

- Adoption of **python** for the whole pipeline of processing, training and analysis

- **Optimization** of the consideed CNN and investigation of other models

- Test the model on **other/mixed datasets**

- Managing of **different projections** (apart from CC)

- **Extension to 3 classes classification** (negative, malign, benign)

DL for BDT classification: python implementation

We developed **a tool composed of a set files (one main script)**, to:

- Compose **properly** the datasets (it is the main limitation of the previous work)
- Define the model architecture (we used **TensorFlow** Keras) from scratch or import known ones
- Perform hyperparameter optimization and/or a k-fold cross-validation
- Train the model on the full dataset and calculate automatically a variety of metrics and produce useful plots
- Calculate, plot and export the activation (saliency) maps

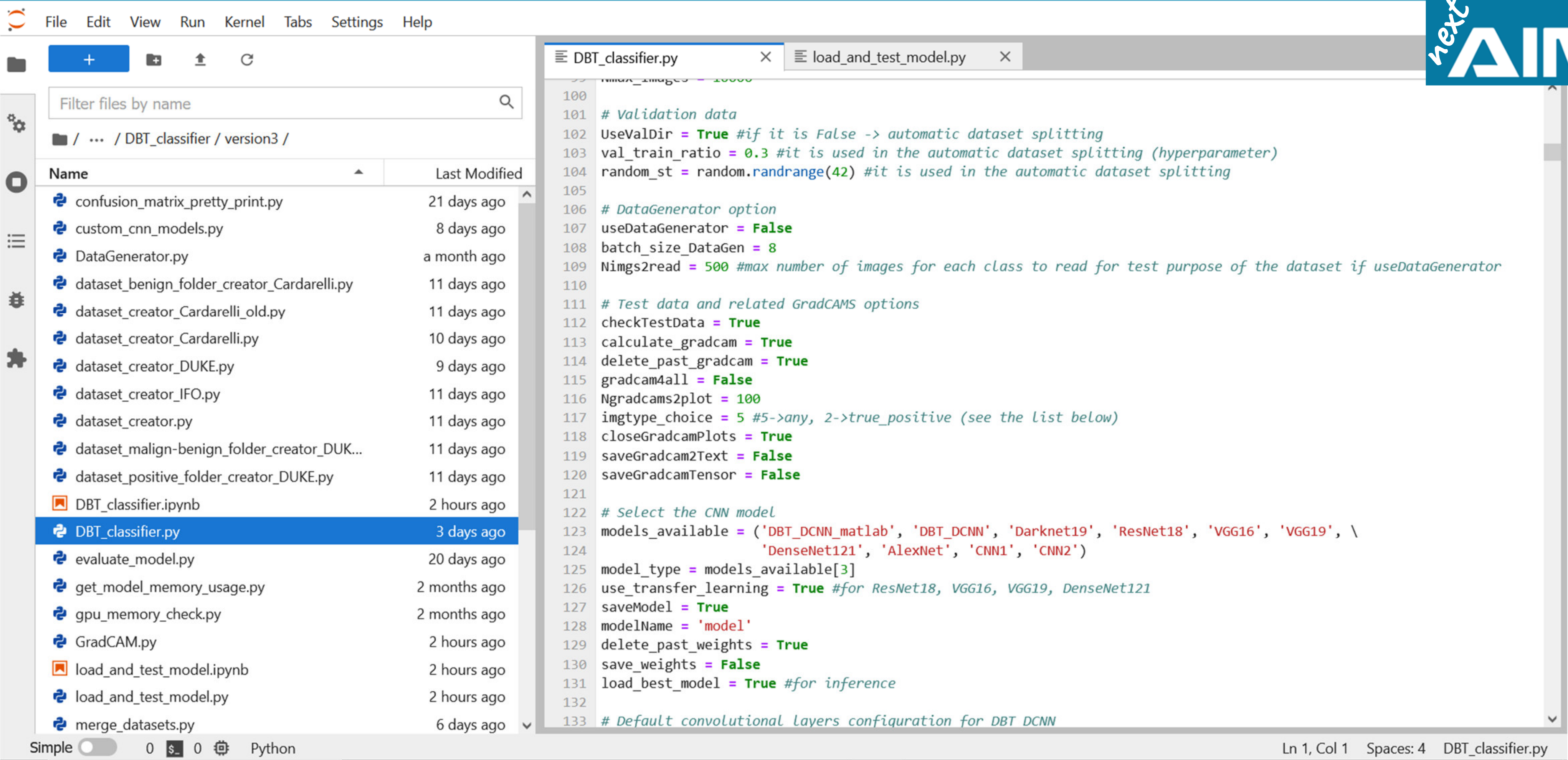
All the variables that define the beahviour of each script are grouped at the beginning («input section») to make its use simple even for non expert users.

DL for BDT classification: python implementation

Features implemented and issues solved:

- Full (and automatic) support of calssification in 3 classes
- Creation of datasets with an (automatic) selection of slices for each patient/projection
- Merging of different datasets pre-processed in the same way
- Data augmentation (flip, zoom, rotate, shift) with random sampling
- Implementation of a Data Generator (to avoid memory issues)
- Automatic split of data in train/validation or import reserved validation data
- Selection among different CNN architectures (some of them are customizable)
- Adoption of regularizers (L2) and dropout
- Transfer learning
- Hyperparameter optimization with Grid or Random Search and k-fold cross-validation
- Automatic setting of the best parameters obtained during hyperparameter optimization and/or the training on the full dataset
- Perform the training of the model on CPU, GPU or multiple GPUs by setting a couple of variables
- Calculation of metrics, confusion matrix, and ROC on train/validation/test sets
- Calculation, plot and export of the activation maps (Grad-CAMs) for all or a sample of images

DL for BDT classification: python implementation



DL for BDT classification: considered datasets

Dataset	Mammographic unit	Number of patiens	Projections	Total slices
Cardarelli	IMS Giotto Class 40000	100 (all with positive biopsies)	CC (L or R)	7496 (3970N + 3526P)
Duke	Hologic	> 1000 (<10% positive), up to now used 102 (32N + 70P)	CC, MLO (L/R)	6191 (3959N + 2232P)
IFO	IMS Giotto Class 40000	40	MLO (L/R)	4062 (2357N + 1705P)

- To **avoid overfitting**, **10 slices per projection** were selected out of the whole set and about **20% patients** were reserved for **validation/test** (the same for various pre-processing level. This was repeat 5 time with random sampling -> manual **5-fold cross validation** (automatic mode was not useful…).
- Due to decimation, about **25% of the images were used** for training (increased to 42% with data augmentation).

DL for BDT classification: image pre-processing

A dedicated tool has been developed in Matlab by **D. Esposito (INFN-NA)**

Import DICOM of DBT scan

↓

Read metadata:
- Type of projection;
- Number of rows;
- Number of columns;
- Number of frames.

↓

Actions on the first image of the image stack:
- manual crop (if requested by user);
- if it is a MLO projection, the user is asked:
  - to draw a straight line containing the pectoral muscle;
  - select the order of the polynomial (1^{st}, 2^{nd}, 3^{rd}) that best fit the pectoral muscle contour.

↓

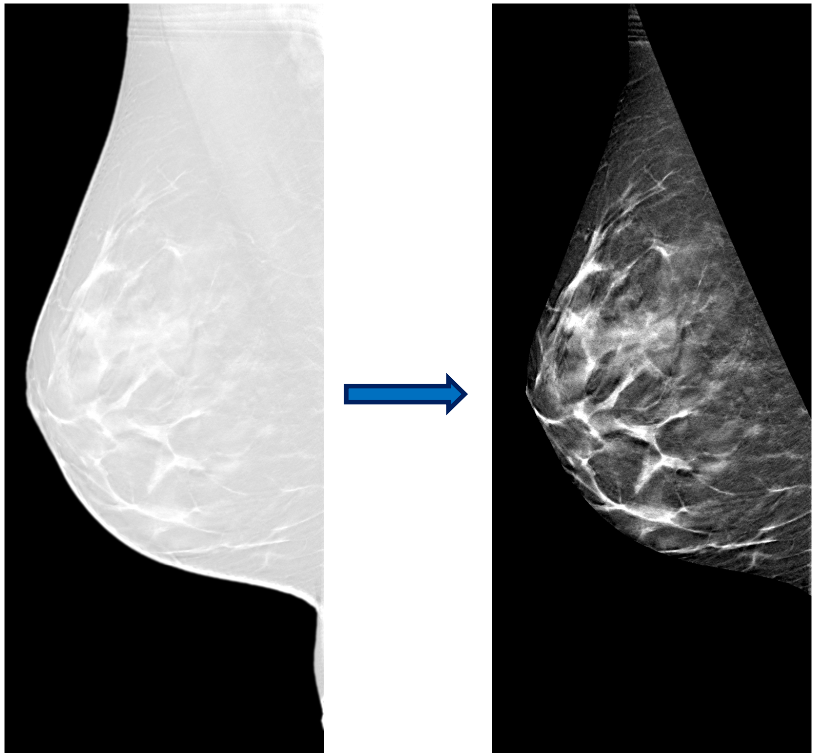
Automatic processing on the whole image stack:
- contrast enhancement;
- skin contour removal;
- pectoral muscle removal (if MLO);

↓

Saving the various image processing steps in the target folders

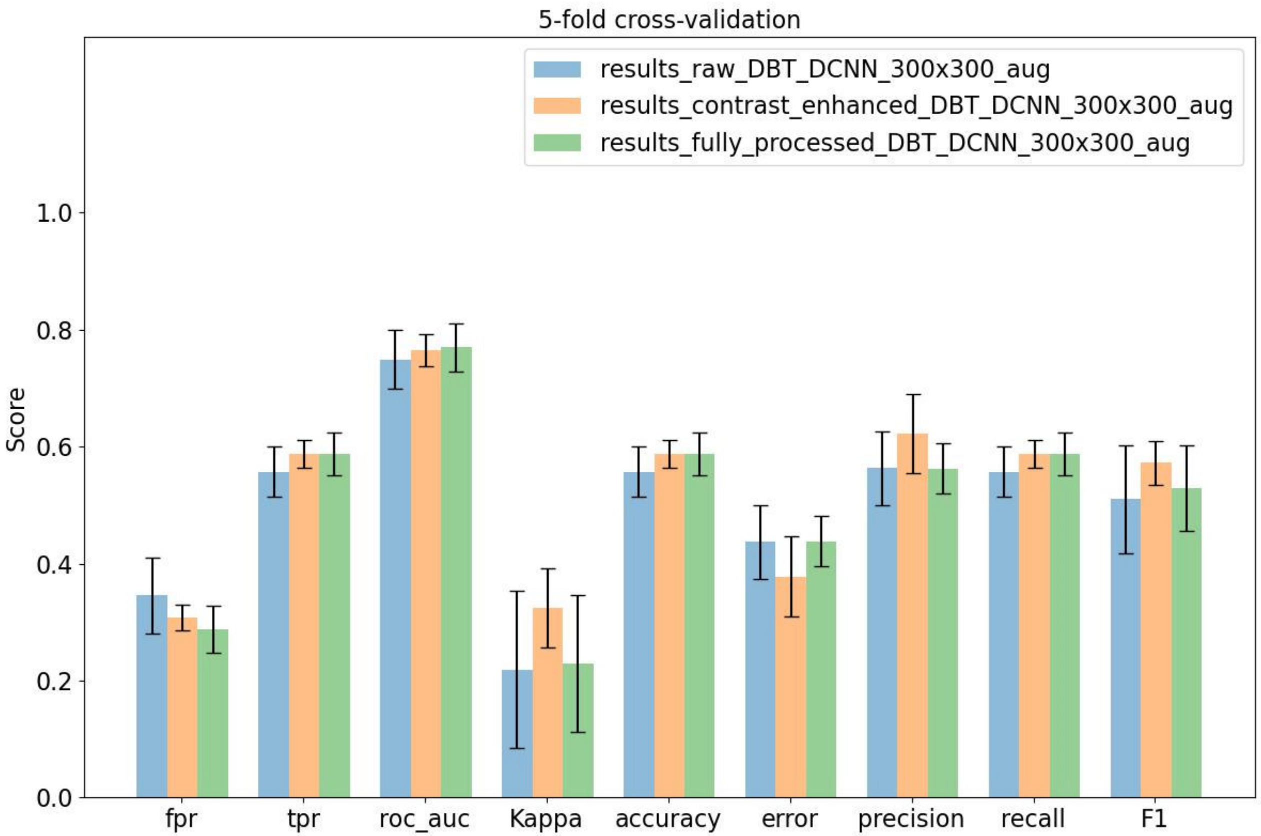
↓

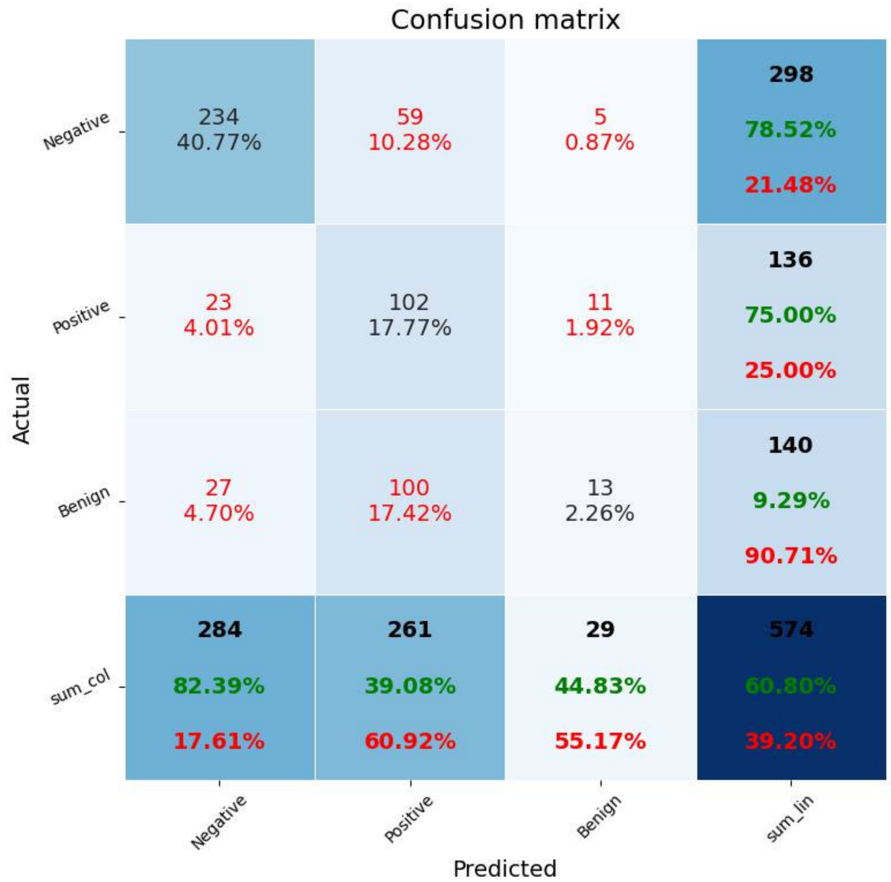
Raw	Contrast enhanced	Contrast enhanced Skin removed	Contrast enhanced Skin removed Pectoral muscle removed



DL for BDT classification: results – 3 classes CC slices

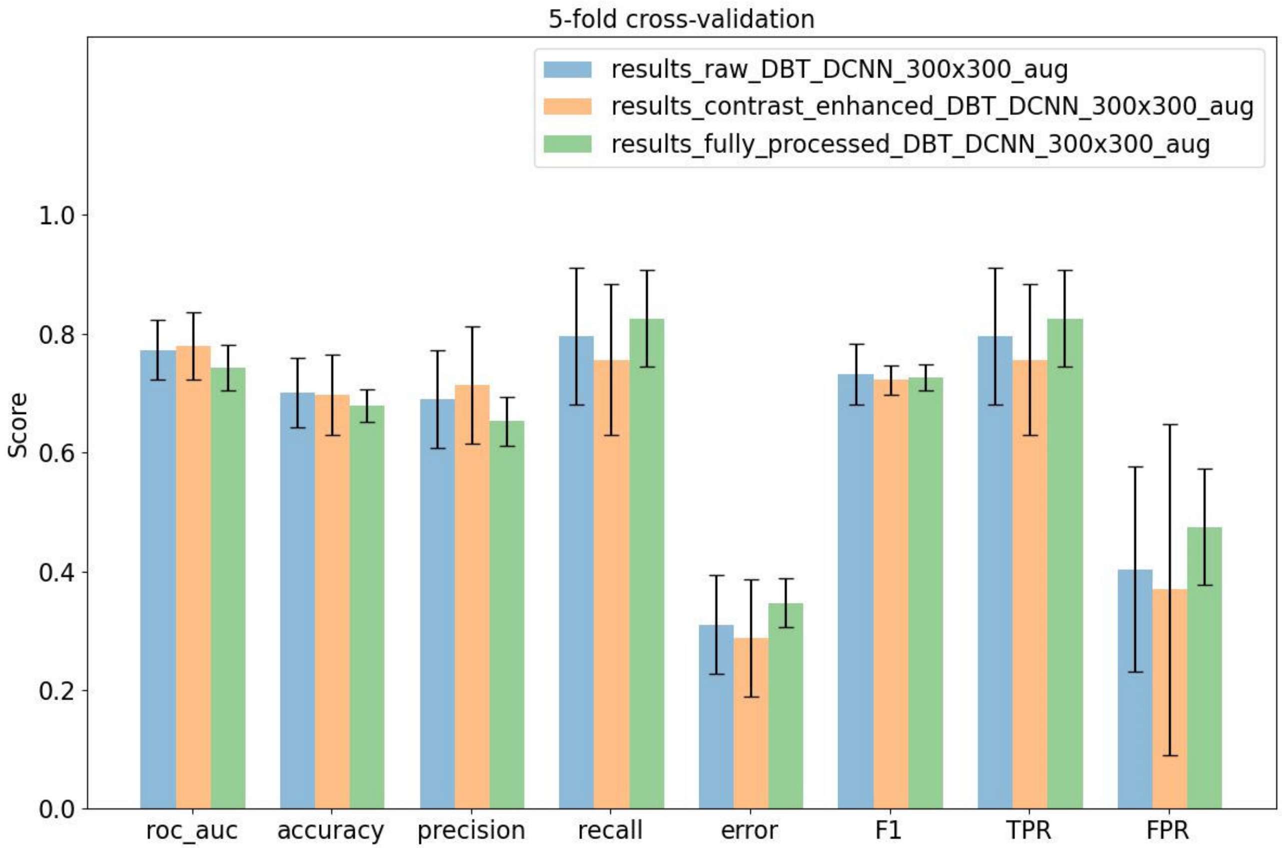
DBT_DCNN simplified: less and smaller filters of conv layers and a much smaller final dense layer (from about $2x10^8$ param to less then $1x10^6$).

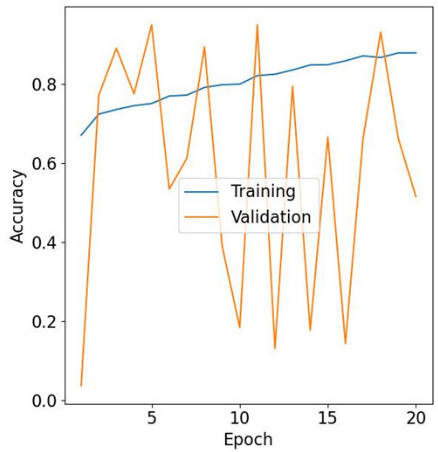


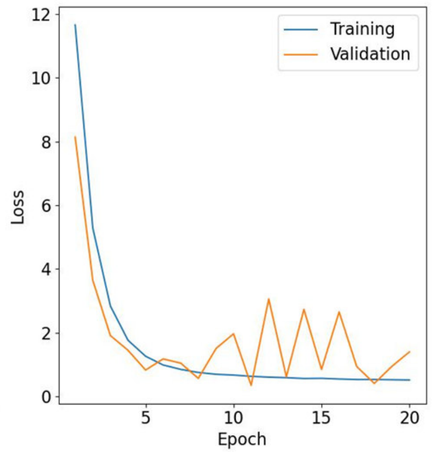


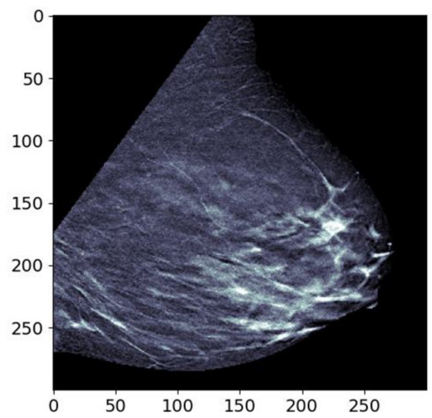
DL for BDT classification: results – 2 classes CC + MLO slices

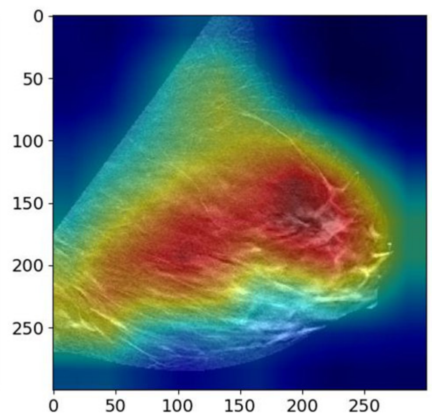
DBT_DCNN simplified: less and smaller filters of conv layers and a much smaller final dense layer (from about 2x10^8 param to less then 1x10^6).





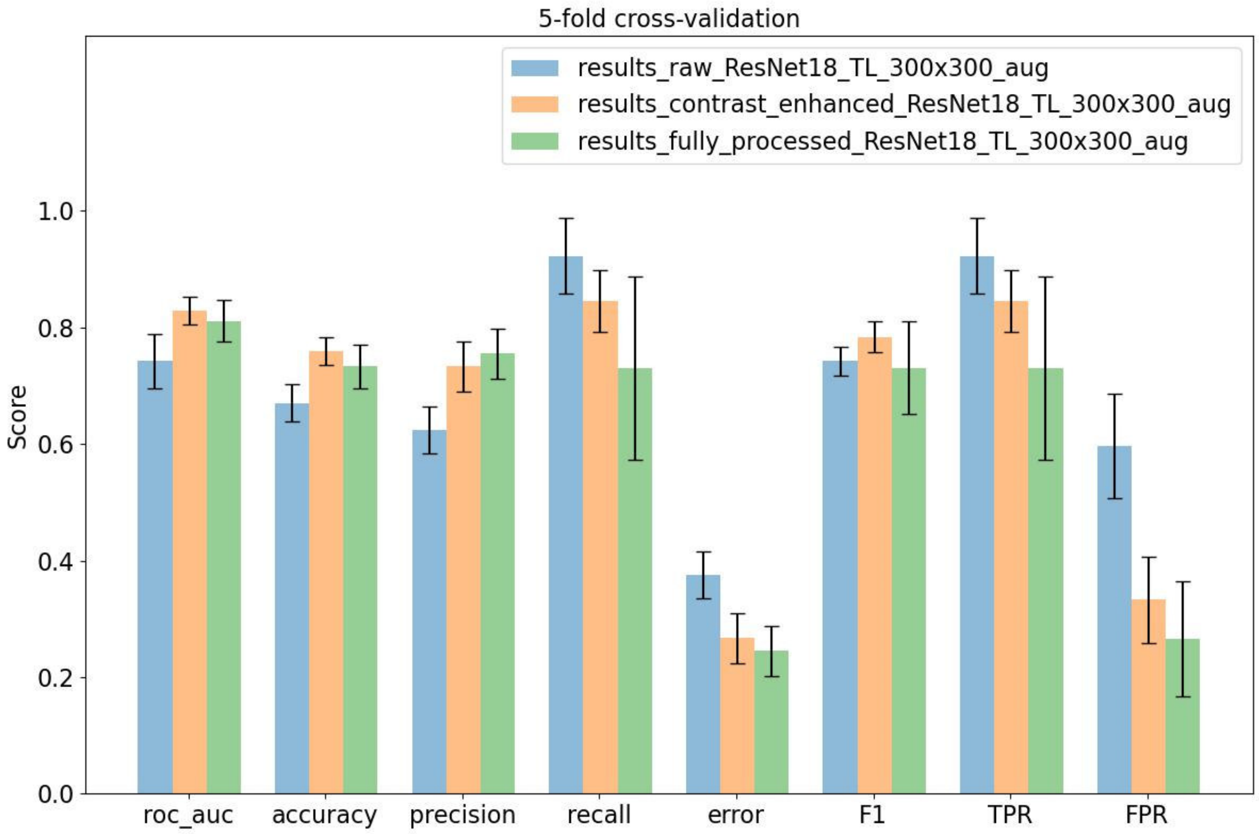


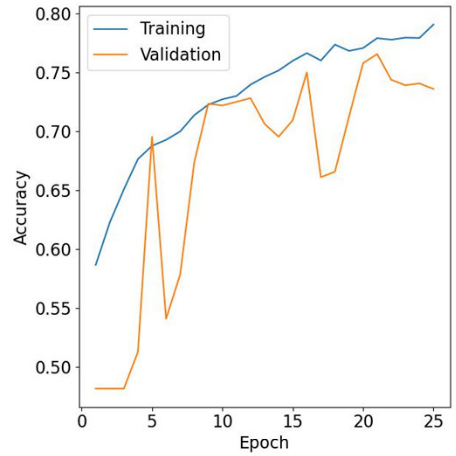


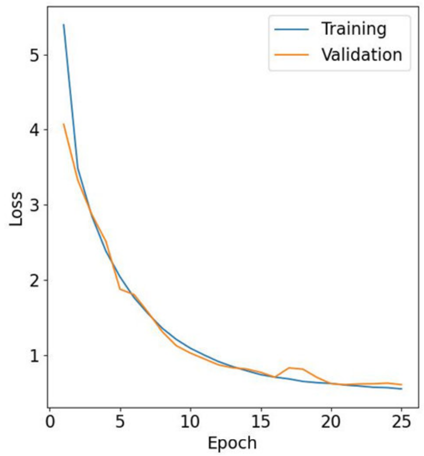


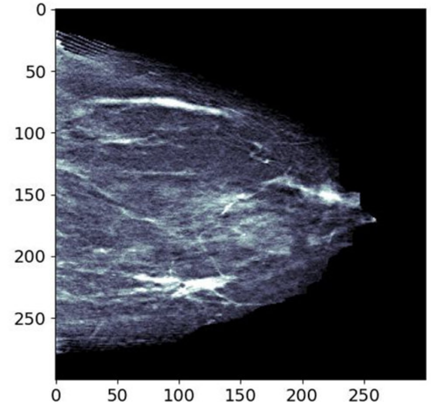
DL for BDT classification: results – 2 classes CC + MLO slices

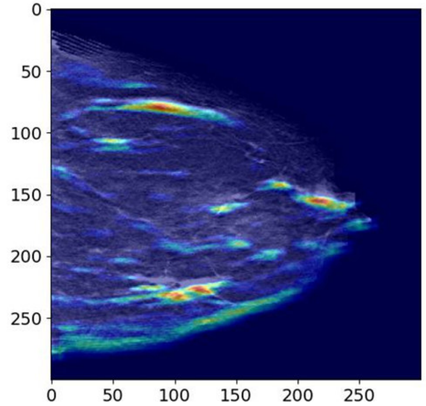
ResNet18 (Transfer learning) + 1 top conv layer (for RGB2Gray) **+ 2 bottom dense layers.**











DL for BDT classification: conclusions

- A tool for the classification of DBT slices has been developed.
- The tool could useful as a template for other medical imaging classification problems.

Open issues
- Improve the classification performance of the considered models, in particular for the 3 classes problem
- Assess the ability of the classifier to correclty recognize the position of lesions (via Grad-CAM anlysis)

Possible strategies for improving the performance and avoid overfitting
- Improve further the considered dataset with more patients (but Duke dataset contains maily negative cases -> data imbalance)
- Implement the pre-processing in python and apply a better data harmonization
- Adoption of an approach similar to the one described in

Buda M et al. 10.1001/jamanetworkopen.2021.19100, who proposed the challenge https://www.aapm.org/GrandChallenge/DBTex/ for which the Duke dataset was released. They trained the CNN to recognize a-priori the mass position…

Tank you for your attention...