# Tuberculosis and Pneumonia Detection from Chest X-Ray Images Using Deep Learning Techniques

Chiara Boscarino
*Department of Electronics, Information and Bioengineering*
Politecnico di Milano, Milan, Italy
chiara.boscarino@mail.polimi.it

Federico Cavallini
*Department of Electronics, Information and Bioengineering*
Politecnico di Milano, Milan, Italy
federico.cavallini@mail.polimi.it

Filippo Castellani
*Department of Electronics, Information and Bioengineering*
Politecnico di Milano, Milan, Italy
filippo.castellani@mail.polimi.it

**Abstract** – *The purpose of this paper is to present the development of a Convolutional Neural Network specifically trained to detect pathologies on chest x-rays of patients with pneumonia and tuberculosis. The work output is a very efficient and accurate tool that can understand when to make an informed prediction and when the input is inappropriate to make statements about the patient's clinical situation. The study includes a detailed explainability analysis of the learned decision-making process to further develop trust in the tool among the final users. In addition, it is discussed how image preprocessing impacts predictions and what approaches can be used when dealing with medical images.*

**Keywords** – *X-rays processing; Automatic preprocessing; Denoising; Image filtering; Denoising Autoencoder; Deep Learning; Convolutional Neural Networks; Transfer learning; Chest X-rays; Image classification; XAI; LIME; Occlusion; Inverted occlusion; Filter visualization; Simple Activation Visualization; Confidence analysis; Prediction thresholding.*

## I. INTRODUCTION

Tuberculosis (TB) and pneumonia are among the most prevalent and deadly infectious diseases globally, and accurate and timely detection is crucial for the effective treatment and control of these disorders.

TB is caused by the bacterium Mycobacterium TB and primarily affects the lungs. Pneumonia is an infection of the lungs that can be caused by a variety of microorganisms, including bacteria, viruses, and fungi. Chest x-ray imaging is one of the most widely used diagnostic tools for these pathologies, as it can provide important information about the presence of lung lesions, consolidation, and other signs of disease. TB and pneumonia can both appear as opacities or areas of increased density on a chest x-ray. TB can cause patchy or diffuse opacities in the lung, often in the upper lobes. Pneumonia can also cause patchy or diffuse opacities, but it is more common for the opacities to be concentrated in specific lobes or segments of the lung. In advanced cases, TB and pneumonia can cause consolidation, which appears as denser, white areas on a chest x-ray. [1, 2]

Nonetheless, it is important to note that the interpretation of chest x-ray images can be challenging, particularly in resource-limited settings where there may be a shortage of radiologists or other trained professionals. Moreover, in recent years, Deep Learning (DL) techniques have shown promise in the field of medical image analysis and have been shown to achieve state-of-the-art performance in a variety of tasks, including the detection of pulmonary disorders. Therefore, AI-based tools can provide valuable support in the screening process of pulmonary disorders. However, the internal mechanism of these models is a major concern, particularly when they are used in medical applications. [3]

A model with high interpretability is desirable when something significant is at stake, like human health. On the other hand, interpretability often comes at the expense of model complexity and in most real applications problems require more entangled structures to be properly represented than simple and interpretable regression models. Therefore, especially in medical applications, explainability is an important consideration that can provide significant benefits in terms of reliability, and transparency. Indeed, it allows for a better understanding of the model's decision-making process and can help identify and address any potential biases or errors in the model. Additionally, it can provide valuable information to practitioners, patients and their families, building trust in the model and technology and it can serve to ensure responsibility and auditability in critical situations [4, 5].

In this study, we propose a DL-based framework using Convolutional Neural Networks (CNN) to classify chest x-ray images as normal, TB, or pneumonia. Given the possible use of the system in real-world applications, the proposed system has been designed to work with high accuracy and efficiency and explainable artificial intelligence (XAI) frameworks have been included in the study to investigate the decision-making process and asses its trustworthiness.

In this paper, the data sources used, along with the processing pipeline and the methodology used to develop and train the models will be presented in Chapter II. Next, in chapter III the results of these methods are presented to provide the reader with an overview of the entire project before the final discussion in chapter IV. Chapter V it is summed up the presented work.

## II. MATERIALS AND METHODS

### 1) Data Description and exploration

This study utilizes a dataset consisting of 15470 chest x-ray images sourced from 12086 individuals (some of which possess more than one image, at most two). The population represented in the dataset is diverse, including patients diagnosed with either TB or pneumonia and a control group. Being the focus of the study a supervised task, each image was accompanied by a label that classified it as one of the three classes: "Normal", "Pneumonia", or "Tuberculosis". This resulted in an unbalanced class distribution, with 60% (9354) of the images being labelled as "Normal", 27% (4250) as "Pneumonia", and 12% (1866) as "Tuberculosis".

Analyzing the dataset by visual inspection, a high variability can be noticed in terms of image content (subject's body structure, condition, medical devices in the body area, ...), grayscale of reference (traditional or inverted contrast, exposure, …) and image quality (clear or noisy – with a different type of noises, shifted and cropped, presenting blanking patches, …) (Figure 1).

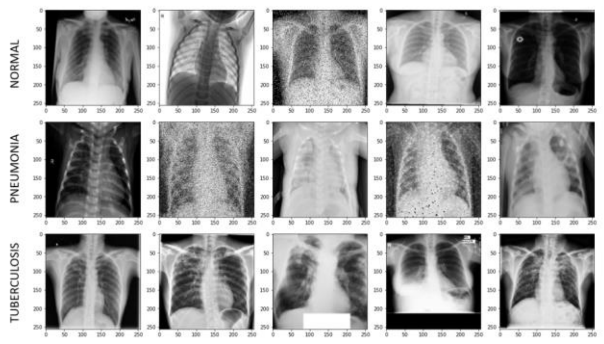


*Figure 1. A subset of the data divided by labels demonstrates a range of image heterogeneity.*

## 2) *Study Design*

The traditional approach to image classification would suggest solving the addressed problem using a neural network designed to predict for each input image the a posteriori probability of the 3 given classes. Nevertheless, considering the nature of the problem and the unbalanced dataset, it may also be reasonable for two models to be developed, one trained to distinguish between pathologic and control groups, and the other to classify between the two pathologies as an expert system (Figure 2).

In this work the two approaches have been investigated, using different architectures. Indeed, parallel to the main workflow related to the development of a single CNN for the multi-label classification problem using the original dataset, a second framework has been established for the development of two classificators: one addressing the binary problem of healthy/pathologic and the second one trained to further categorize the pathologic patients into Pneumonia and Tuberculosis. To do so, it was also necessary to re-organize the data management to have in the first case access to all images, labelled either as healthy or pathologic and in the second case to deal only with images related to pathologic patients along with the correspondent disease as a label.

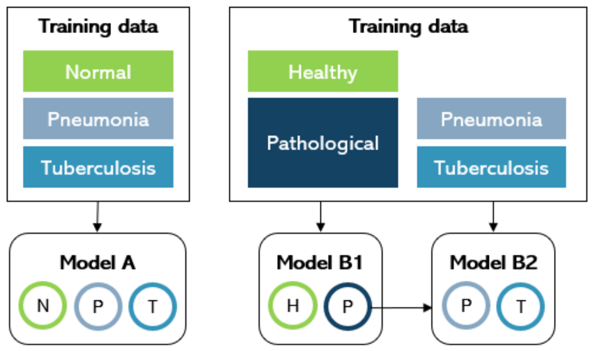


*Figure 2. The two addressed options for model development: a single multi-label classifier (left) or two binary classifiers to be used in cascade.*

The model development workflow employed in this study in both cases was structured sequentially as shown in Figure 3 (Note that also all specifications presented below regarding preprocessing and data splitting hold in both cases). Given the dataset to be used for model development and tuning, the first step was to subdivide the dataset into training, validation, and test sets according to the desired use. The image processing issue was addressed next, using firstly traditional preprocessing approaches on all images to standardize data and mitigate variability. Secondly, applying denoising techniques to increase the overall quality of the available data by removing noise where required. Using the so obtained sets, multiple architectures were trained and tuned, and finally, their performance was tested and compared, as shown below, before selecting the final model and testing it on the hidden test.

## 3) *Dataset splitting*

An important step preliminary to the model design phase, regardless of the underlying approach for the model design, is the dataset division into 3 perfectly separated sets to train, validate and test the models. In performing this operation, it is crucial to use a stratified split to ensure that each set contains a representative sample of the data, accounting for any imbalances in the class distribution. Furthermore, when working with medical data containing multiple images that belong to the same patient, it is important to keep them together in the same split to prevent evaluating the model on data that were partially already seen in the training. This can lead to an overestimation of the model's performance that must be avoided.

All these aspects were considered when splitting the original dataset in this study. As a result, a stratified splitting algorithm was used, to maintain the class distribution. Additionally, a constraint was included to ensure that all images related to the same patient were in a single set. In terms of the splitting proportion, 20% of the original dataset was retained for the final evaluation of the model. While the residual 80% was further divided into training and validation sets (90%, and 10% respectively) and used to fit the models on data. (Figure 4)

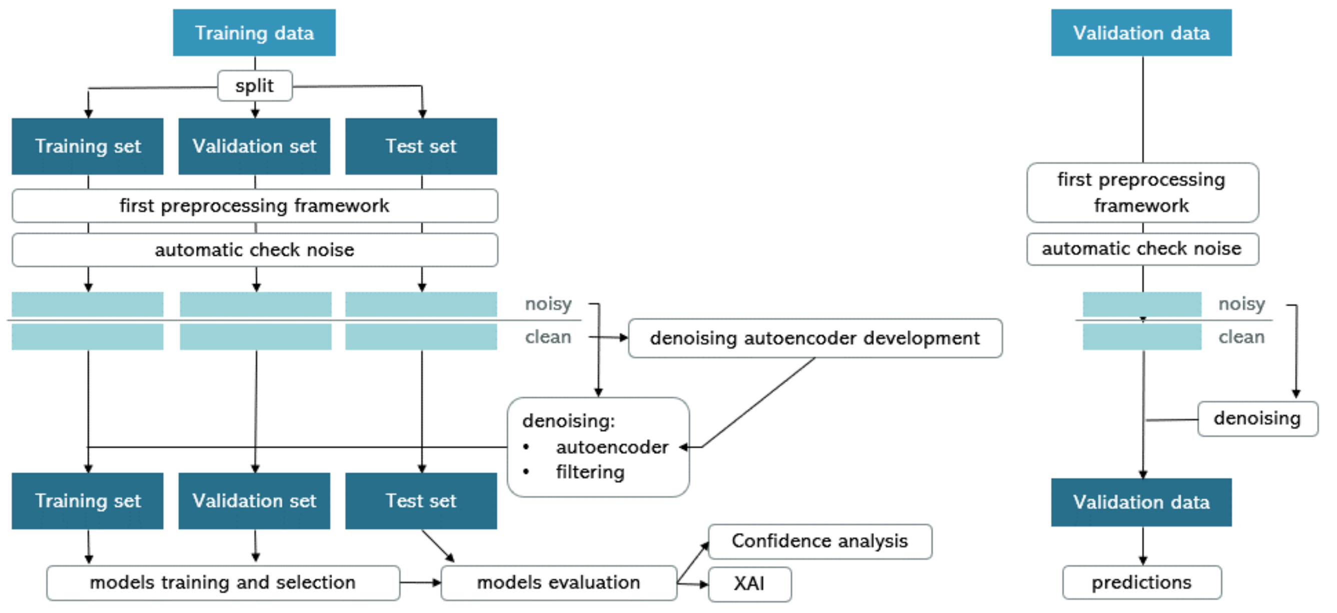


*Figure 3. The model development workflow is presented visually by its macro-steps.*

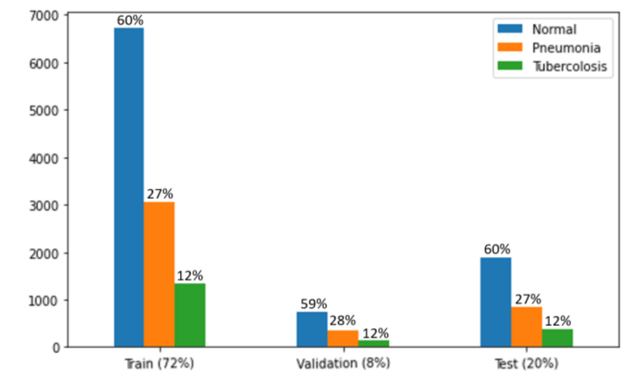


*Figure 4. The training, validation and test sets are reported with the related label distribution (only for the main workflow for the sake of conciseness). Note that the stratified split maintained the same proportions of classes among the 3 sets (with a negligible error caused by the additional constraint on patient non-overlap between sets).*

### *4) Image Processing*

Given the high variability of the images, a preprocessing framework was integrated. The framework was structured in two main steps, one including traditional preprocessing approaches to be applied on all images to standardize the data and to mitigate the impact of the image variability on the classification task, the other applying filtering and DL-based techniques to increase the overall quality of the available data by removing noise where required.

The first preprocessing framework included colour normalization and resizing. To be precise, the images were firstly resized from their non-homogenous and generally excessive dimensionality to a standard size. Then, all images passed through a function properly designed to detect whether an image was inverted and eventually to invert its grayscale. The rationale behind this automatic detection relies on the counting of pixels near black and white saturation at the image's corners (Appendix A, Figure 1 for more details).

The second framework focused on image quality enhancement and included an Automatic Check for Noise (ACN) and the subsequent application of a denoising function where required. The rationale behind the ACN module relies on the detection of the relative frequency of maximally saturated pixels, both over the whole image and concerning its central area. By including this double check, it is ensured that even low-level noise is detected, while not mistaken for noisy overexposed images or those with white patches (Appendix A, Figure 2 for more details).

Once an image was identified as noisy it was processed through a denoise module. In this study, multiple options were considered to implement this module. At first, a set of filters in cascade with different combinations was tested. To be more specific it was defined a batch of filters including the following in sequence: median, gaussian, bilateral and sharpening filter. The rationale was to include a median and a gaussian filter to remove salt-and-pepper and gaussian noise respectively, then to apply a bilateral filter to introduce a non-linear transformation to reduce noise while preserving edges [7] and finally to apply a sharpening filter to further enhance the contours and restore a clear image. Then, to increase the flexibility of the framework, the function was modified to apply the filters in different combinations and to enable the selection of a specific combination for the application considering the effective performance on data.

Starting from this premise, the next attempt was to use a Fully Convolutional Denoising Network (FCDN) to learn the optimal filters to be used. Therefore, an FCDN was specifically designed for the sequential application of filters learned with a data-driven approach, consisting of five convolutional blocks without any pooling operator.

Finally, a Denoising Autoencoder (DAE) was also designed and trained to clean up data using a DL-based approach (Figure 5). The DAE is a type of neural network that is specifically trained to learn a data representation ignoring the noise, to reconstruct a clean image from a noisy version of the input.

These two DL approaches for denoising were developed on a dedicated data framework. Since the two main kinds of noise identified in the original data are salt and pepper and gaussian noise, only these two were addressed. A new dataset for this model training was created. Starting from the non-noisy original images, the two different kinds of noise were artificially applied to them to simulate the original phenomenology. So, it was possible to train the two denoising neural network using as input the artificially noisy images, and as target the original clean images.

To define the final workflow, the output of the three considered denoising techniques was evaluated, both visually inspecting the outputs and in terms of the final performance of the models using the preprocessed images for training. Then, the most effective module was included in the final workflow. The images obtained thanks to the selected pre-processing framework were then used to train, tune and test the models designed for this study.

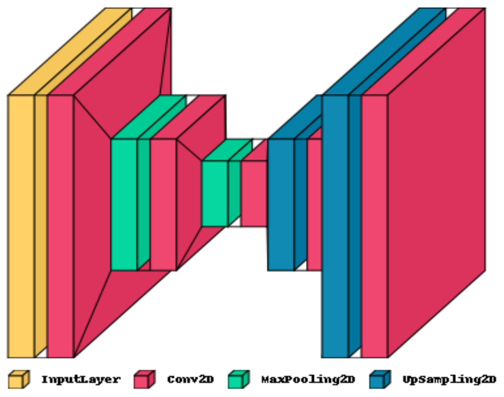


*Figure 5. The structure of the Denoising Autoencoder proposed in this study consists of two convolutional layers with MaxPooling in the contracting part and three convolutional layers with UpSampling in the expanding part.*

Also, the training of the models has been performed both on processed and non-processed images. This has been done, on one hand, to verify the effectiveness of the pre-processing in learning the data representation. On the other hand, considering the possible use of the developed model in the real case scenario, to be able to provide a lighter software eventually not including the pre-processing framework.

### 5) *Architecture definition*

The study focused on the development of DL models for image classification and included the investigation of multiple neural network design approaches.

The first architecture assessed was a traditional mid-depth CNN, specifically designed for the purpose. It was a simple structure consisting of five convolutional blocks with doubling depth (each composed of a convolutional layer followed by a MaxPooling operation) followed by a dense top with one hidden layer. For this first attempt, such an elementary architecture was selected to have a starting point of performances to assess the complexity of the problem and define a reference of accuracy.

As a second approach toward better results, it was tested another model specifically designed for the purpose which consisted of a Residual Neural Network (ResNet) incorporating an attention mechanism. The main strength of ResNet is the use of residual connections, which work as shortcut connections that allow the gradients to flow more easily through the network during training and to learn more complex functions by stacking many layers without suffering from the vanishing gradient problem. On top of this, an attention model has been introduced to focus the learning process on relevant features and improve the overall representation ability of the network. To be more specific, the implemented mechanism consisted of a "Squeeze-and-Excitation" (SE) block placed between the convolutional layers of the network aiming to adaptively recalibrate channel-wise feature responses by explicitly modelling inter-dependencies between channels.

Afterwards, on the premise that the SE block is one of the key components on which the design of the EfficientNet family of models has been based, a transfer learning approach was introduced at first using two variants of the EfficientNetV2 model, which is a version of the EfficientNet architecture that is optimized for image classification tasks. To be specific the B0 and the S versions were tested. The main difference between the two models is the trade-off between accuracy and computational resources. The larger model EfficientNetV2B0 (EffNetB0) is designed to achieve higher accuracy on image classification tasks, but it requires more computational resources. On the other hand, EfficientNetV2S has lower accuracy but is more computationally efficient. The introduction of this family of models managed to effectively improve the efficiency of CNNs, but it isn’t the only solution for this purpose.

Accordingly, it was tested also another architecture that proved to solve efficiently the problem and who has shown also promise in the literature on chest x-rays for pneumonia detection: the DenseNet [6]. DenseNet is an architecture characterized by a dense connectivity pattern among layers in a feed-forward fashion, originally designed to improve the flow of gradients and to mitigate the vanishing gradient problem in deep networks. Considering the impressive results Yang et al. achieved using this network, it was reasonable to try to design a DenseNet like the one presented in that paper, but adapted to our specific problem, slightly modifying its top (adapting the classification and the output layer to the proper number of classes to be predicted) and tuning the net's parameters on our data.

To further improve performance, the use of an ensemble approach can be effective. This method involves combining the predictions of multiple models, often of different architectures, to create a more robust and accurate final prediction. This paper presents several architectures that have been trained and tested to solve the task at hand, but a significant performance improvement was enabled over any single model by combining their predictions.

### 6) *Model training*

During this work, a standard framework was established to perform the model training, regardless of the underlying approach and the architecture, to guarantee results comparability.

First of all, a standard framework for data retrieval including augmentation techniques was defined. Data augmentation is a technique used in DL to increase the size and diversity of the training data by creating new, synthetic examples from existing ones. This is done by applying random transformations such as flipping, cropping, scaling, and rotating to the original data. The goal is to artificially increase the variability of training data, reduce overfitting, and improve the generalization performance of the model.

In this work, the selected transformations were chosen to have minimal impact on the features characterizing the label. At the same time, given the limited knowledge about chest x-rays images, three different augmentation modalities have been designed with increasing intensity to determine at training time the optimal degree of transformation concerning the model's ability in detecting pathologies. The main difference among the three modalities is the range over which the intensity of the transformation is defined, determining the augmentation's aggressiveness, or equivalently the magnitude of the introduced variability (Figure 6).

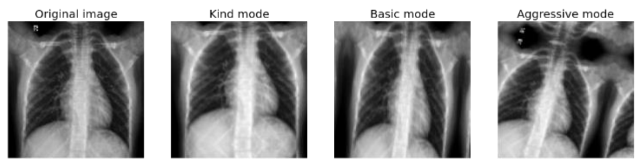


*Figure 6. The three augmentation modalities presented for increasing intensity on the same input image.*

Another important factor included in the framework was the dimension of the batch of images to be retrieved. Indeed, wanting to exploit the advantages of the Stochastic Gradient Descent (SGD) algorithm for training, it is relevant to properly determine the dimension of the batches over which the adjustment of the parameters is iteratively performed. The SGD is the iterative method used to optimize the loss function by iteratively adjusting the parameters over a randomly selected subset of the training data (batch). The randomness introduced by this approach in the updates can lead to more robust optimization and regularization. In this work, multiple dimensions have been attempted using the power of two values (16, 32, 64, 128) for memory management and computational efficiency issues.

For the training phase of each architecture the before-defined training and validation set were employed, for parameter fitting and online validating, respectively. As a general approach for each training process, a high number of epochs was set, to ensure that the model had enough iterations to learn a good data representation, but accompanied by some regularization technique to avoid overfitting. The main form of regularization employed in this study to avoid overfitting was the Early Stopping monitoring the accuracy on validation, to stop the learning procedure as soon as the next iterations were expected only to reduce the generalization power of the model. Moreover, to enable a faster and more stable convergence other two crucial factors were introduced in the training procedure: the Adam optimizer and the adaptive learning rate. The former is an optimization algorithm used to efficiently include momentum in the weights update. The latter refers to the practice of adjusting the learning rate along the process, to allow to start with a larger updating step and to reduce it as soon as required to reach the minimum on the loss function (to be precise, using a reduction of 0.5 after 3 epochs with no relevant improvements). Additionally, other forms of regularization were considered to further avoid overfitting. Indeed, some dropout layers have been included in the architectures and weight decay frameworks have been envisaged where the provided regularization would have resulted to be not enough. Parameters related to both techniques have been considered as the model's hyperparameters and properly tuned.

Furthermore, to deal with the class imbalance, the possibility to use a weighted loss function in the training phase of models was envisaged. A weighted loss is a way to correct inequality between classes by giving a higher weight to the minority class to compensate for its underrepresentation, i.e., by making the model pay more attention to less frequent instances. Similarly, to common practice in literature, in this application, these weights have been set to be inversely proportional to class frequency.

Finally, the result of each training process was evaluated using the confusion matrix, the total accuracy and other metrics (precision, recall and f1-score all computed class-wise) computed both on the training set and validation set, to also check for overfitting by comparing the two. The test set was left aside until the very end of each model training to perform on it only the final model evaluation, to avoid any overfitting on this set as well. The final comparison among models has been performed using mainly the total accuracy, the f1-score computed class-wise and the ROC curve.

### 7) *Confidence analysis*

A detailed prediction and confidence analysis has also been conducted to provide the final user with a more transparent insight into the decision-making process. In particular, it was decided to consider the class probability distribution provided by the selected model and to analyze the prediction accuracy for different confidence thresholds. This method enabled the disclosure of the correlation between accuracy and confidence in the prediction. To be precise, given a model designed to perform classification over a defined set of classes $\{C_1, \dots, C_K\}$ trained over a given dataset $\{x_1, \dots, x_N\}$, calling W the learned parameters, its output for a generic observation is the probability vector $p(x_i/W) = [p_1, \dots, p_K]$ where $p_j$ is the predicted probability for $x_i$ to belong to $C_j$. Therefore, setting a confidence threshold $T > 0.5$ below which the confidence of the model in predicting the class is considered too low to make the prediction reliable, it is possible to define a confidence interval in which the model should be trusted in the specific application. Specifically, if the model is intended to be used as a support tool for medical decisions, as in this case, providing the medical operator with a confidence threshold below which the model is to be considered not completely reliable or "not sure" can be relevant. In this study, the confidence analysis was performed by analyzing relevant metrics at an increasing confidence threshold. The following metrics were included. The number of predictions provided (PP) and not provided (NPP) with the given threshold as minimal confidence to provide the prediction:

$$PP_{rate} = \frac{Provided\ \ predictions}{N}$$

$$NPP_{rate} = \frac{Not\,provided\ \ predictions}{N}$$

The Accuracy of the provided predictions:

$$PPAcc = \frac{Correct\ predictions}{Provided\ \ predictions}$$

The Total Accuracy considering only high confidence predictions:

$$TotalAccuracy = \frac{Correct\ predictions}{N}$$

The Total Error, quantifying the overall misclassified samples with the thresholding method:

$$TotalError = \frac{Misclassified\ predictions}{N}$$

As a result:

$$TotalAccuracy + TotalError + NPP_{rate} = 100\%$$

Certainly, the ideal situation would have been to have all cases provided with a correct prediction, but in the real scenario, given the specific application, it is reasonable to prefer to have a higher $NPP_{rate}$ than a higher error rate, envisaging the possibility that low confidence can be caused by a low-quality input image. Indeed, due to the lack of prediction, users may be compelled to undertake further diagnostics to reduce uncertainty.

Therefore, in this study, we included the analysis of the performance of the model in predicting the classes varying the minimum confidence threshold to consider the prediction reliable.

### 8) *Interpretability and explainability*

In medical applications, decisions made by AI-based tools can have significant consequences for patients, and it is important to ensure that these decisions are transparent and understandable to medical professionals and patients. This is known as explainability and allows to understand how the tool arrived at a particular decision, and to assess whether the decision is appropriate in the context of the patient's unique condition. Without explainability, AI-based tools can be seen as black boxes that produce decisions that are difficult to interpret and understand. When dealing with image classification problems using DL approaches, it is even more challenging to stay within the boundaries of eXplainable Artificial Intelligence (XAI) considering the data-driven approach for feature extraction and the subsequent lack of easily interpretable features.

Visualizing the filters of a convolutional layer can be useful to understand the features that the network has learned to extract from the input data. However, it is a limited technique for two reasons. Firstly, filter visualization is only useful for shallow layers, while deeper layers in the network are responsible for detecting more complex and abstract features, and filters learned in these layers tend to be less interpretable. Secondly, small filters provide a limited view from which patterns are less recognizable. Accordingly, in the context of this study filter visualization was included in the XAI methods employed, but only for the shallowest layers, and other methods were considered in order to provide other visual explanations to understand the overall decision-making process of a model.

The main XAI methods included in this paper are the occlusion method, the Local Interpretable Model-Agnostic Explanations (LIME) technique, the Simple Activation Visualization (SAV) and the Gradient-weighted Class Activation Mapping (Grad-CAM).

The occlusion method is a technique used in explainability to understand the importance of a certain input feature to the prediction of a machine learning model. It works by occluding a portion of the input image and observing the effect on the model's prediction. This helps determine the regions of the image that the model relies on for making its prediction and provides insights into the model's understanding of the data. In this work, we tried both the traditional approach and its inverted version, in which instead of hiding a portion of the image and perform the classification on the remaining part, the inference is demanded only by providing the portion.

LIME is a technique used to explain the predictions of any classifier regardless of the underlying architecture (model agnostic) that provides locally interpretable explanations by approximating a black-box model's prediction around a given instance, and then explaining the prediction in terms of a simple linear model. This way, it explains the model's decision by highlighting the most important input features that contribute to the prediction both positively and negatively.

The SAV is a visual explanation technique that aims to explain the contribution of a specific set of neurons in a network to the prediction by visualizing the activation map of the relative convolutional layer.

Finally, the GradCAM is a technique that generates heatmap visualizations to show which regions of an input image were most important for a given prediction. It works by computing the gradient of the target class score with respect to the feature maps of a convolutional layer and then using these gradients to weight the feature maps, generating a coarse localization map highlighting the regions of the image that contribute most to the prediction. In this work, the last convolutional layer of the CNN was selected to compute the gradient. In particular, it was performed on the CNN built starting from EfficientNetB0, having the last convolutional layer output with size 7x7x1280. The obtained 7x7 GradCAM was then expanded and superposed to the original image.

Besides the immediate visual explanation given by this methodology, some analyses were also performed on the GradCAM pixel values, representing the gradient for a specific area. In particular, it was looked at the correlation between the maximal values of the 49 pixels and the prediction correctness. A too high gradient value for a pixel means that having the specific area slightly changed may bring big changes in the model output which might lead to unstable predictions because the model is not looking at features extracted from the whole image but is only focusing on a small detail.

All of these XAI methods can be combined to form a reasonable understanding of the decision-making process learned by the network. This can be useful to experts in assessing decision adequacy and developing trust in the model, and that is what this work has attempted to provide.

## III. RESULTS

### *1) Data Processing and Augmentation*

As a final result of the analysis presented before, the preprocessing framework established was composed of the data retrieval module performing resizing at standard size 224x224 [pixels], the check for inverted images and the inverter module applied where required, the ACN and the denoise module applied where required.

The three denoising techniques considered, namely the batch of filters in cascade, the DAE and the FCDN, provided quite successful results (Appendix A, Figure 3 for more details). Regarding the set of filters in cascade the optimal combination resulted to be the one including the median and bilateral filters only. The parameter chosen for such filters are 5x5 kernel size for the median filter; 15x15 neighbourhood size for the bilateral filter ($\sigma_{color} = 2,\ \sigma_{space} = 2$) (Appendix A, Figure 4 for more details).

By visually inspecting the 3 outputs the most performing denoising module appeared to be the Batch of Filters and this was further confirmed by the comparison of the final performance of the models using the cleaned images for training, as reported in Table 1.

| **Denoising module** | **Accuracy** | **F1 score** [ **N**, **P** , **T** ] |
|---|---|---|
| *None* | 0.96 | [0.96, 0.97, 0.88] |
| *Batch of filters* | 0.97 | [0.97, 0.98, 0.91] |
| *DAE* | 0.96 | [0.96, 0.97, 0.87] |
| *FCDN* | 0.96 | [0.97, 0.96, 0.89] |

*Table 1. Performance obtained with the same model (EffNetB0) using the specified denoising module in the preprocessing framework to prepare the dataset.*

As a result, the Batch of Filters was the module included in the final workflow.

Regarding augmentation, the more conservative approach turned out to be the most rewarding, since the modality that provided the best results was the least aggressive among the three. Hence this was the one included in the proposed workflow.

### *2) Model comparison*

Regarding the general classification approach, the design with two binary networks in cascade was early abandoned due to unsatisfactory results in combination with a higher computational cost. With this approach, the training time almost doubled, and the error rate resulted to be higher with respect to the end-to-end classification.

Moving to the chosen approach with a single multi-label classificator, all models developed resulted in having learned effective representations of data, as in the end, they all provided really good results. The obtained results in equal training conditions are reported in Table 2 for the following models: traditional mid-depth CNN trained with weighted loss, ResNet with SE attention blocks, EfficientNetB0, EfficientNetV2S and DenseNet, with the respective ROC curve in Figure 7 for the best four ones. In light of these results, the ResNet and the EfficientNetB0 were selected as the best models to solve the given task.

| Model | Accuracy | Recall [ N, P, T] | Precision [ N, P, T] | F1 score [ N, P, T] | AUC |
|---|---|---|---|---|---|
| *CNN* | 0.93 | [0.92, 0.98, 0.92] | [0.97, 0.97, 0.73] | [0.95, 0.97, 0.81] | [0.985, 0.998, 0.982] |
| *ResNet* | 0.95 | [0.97, 0.96, 0.82] | [0.95, 0.97, 0.89] | [0.96, 0.97, 0.85] | [0.988, 0.998, 0.984] |
| *EffNetB0* | 0.96 | [0.95, 0.97, 0.92] | [0.97, 0.97, 0.83] | [0.96, 0.97, 0.88] | [0.991, 0.996, 0.992] |
| *EffNetV2S* | 0.95 | [0.97, 0.95, 0.82] | [0.95, 0.98, 0.86] | [0.96, 0.97, 0.84] | [0.986, 0.997, 0.985] |
| *DenseNet* | 0.88 | [0.89, 0.87, 0.78] | [0.92, 0.96, 0.59] | [0.91, 0.91, 0.67] | [0.958, 0.992, 0.944] |

*Table 2. Each row in the table reports the results obtained for the correspondent model in terms of total accuracy and multiple metrics over the single classes.*

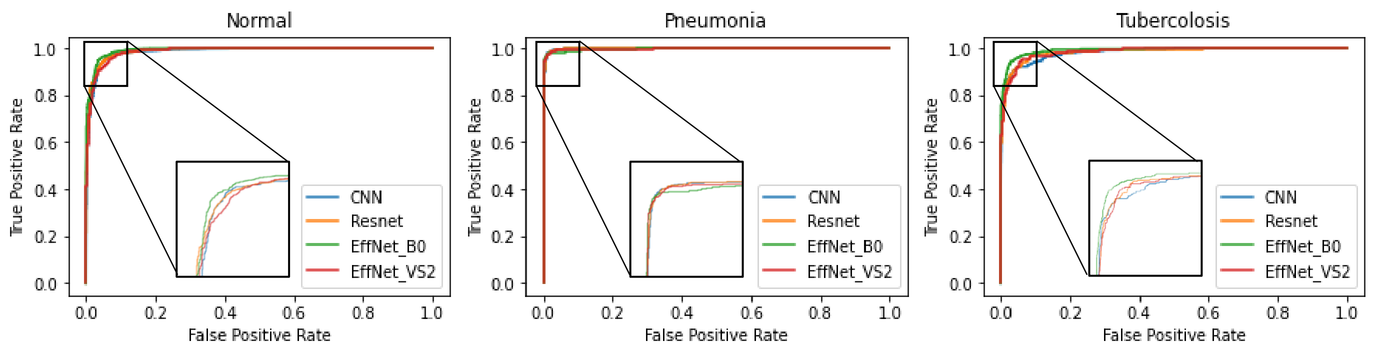


*Figure 7. The ROC curves reported for the best four models, with a zoomed view on the critical point of the curve.*

Finally, these two models (EffNetB0 and ResNet) were further tuned over the processed images and combined by averaging in an ensemble. Such aggregated model would combine the strengths and weaknesses of the individual models and increase robustness and performance, as can be noticed in Table 3 both in terms of accuracy and F1 score over the three classes.

| Model | Accuracy | F1 score [ N, P , T ] |
|---|---|---|
| *EffNetB0* | 0.966 | [0.97, 0.97, 0.91] |
| *ResNet* | 0.955 | [0.96, 0.96, 0.90] |
| *Ensemble* | 0.97 | [0.98, 0.98, 0.93] |

*Table 3. Performance obtained on the test set, by ensembling the EffNetB0 and the ResNet model, concerning the two models used independently.*

Ensembling, however, comes with the downside of increased complexity and computational cost. Nevertheless, the use case was deemed to be worth it, and the ensemble model was selected as the most appropriate model for the application.

### 3) *Confidence analysis*

Even more interesting is the confidence analysis on the final model. Analyzing the confidence of the model differentiating between the case of a correct and incorrect prediction, it emerged that the model was on average much more confident in predicting the correct class with respect to the other case. This means that when the model is not sure about any of the classes to be predicted (i.e., the class probability is more balanced among the three labels) it is more likely that the final prediction obtained as the maximal class probability will be wrong. The plot in Figure 8 depicts this behaviour of the model, which is reasonable and desirable. Correct predictions have a distribution of maximum probability very skewed towards maximum likelihood. Conversely, the misclassified observations result in having a prediction with on average a much lower maximum class probability, thus a more balanced probability distribution among the three labels.

Consequently, this pattern suggests that the use of a threshold defining the minimal level of confidence for a prediction to be reliable may result in a higher level of overall accuracy. This would lead to the absence of predictions for some observations. However, given that a medical application is involved, this might be a reasonable trade-off considering the rationale that it is better to say “I don't know" and, eventually to move to further diagnostic attempts, than to affirm something incorrect. Nonetheless, the definition of such a threshold isn’t trivial and must be properly assessed based on the model’s response.

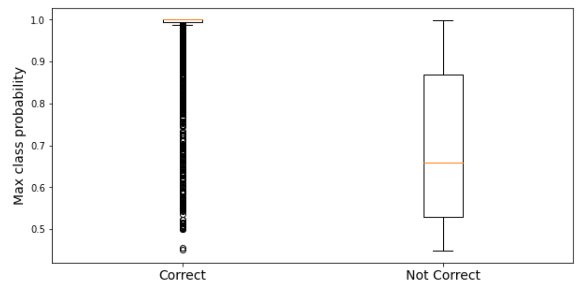


*Figure 8. The boxplot reports the distribution of the maximal class probability predicted by the model in the case of correct and incorrect final prediction (obtained as the class with the most likely one among the three according to the model's predictions).*

Considering as relevant metrics the Total Accuracy, the Total Error and the NNP, as defined before in chapter II, and analyzing the model's response varying the threshold defining the minimal confidence for a prediction to be provided, the results presented in Figure 9 were obtained. The reported graph depicts how the results change by varying the threshold. It is evident that the higher the threshold, the higher the probability for a label must be to provide the correspondent class as the final prediction (the more confident the model must be in predicting the class), thus the higher the number of observations lacking a prediction, but the lower the total error. Conversely, by progressively lowering the threshold the constraint on confidence gets less strict, thus the number of samples not predicted gets lower, but the likelihood that the provided prediction is wrong gets higher.

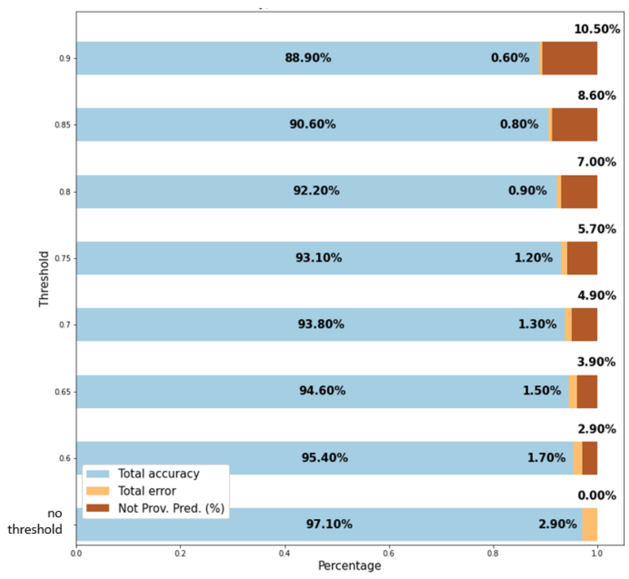


*Figure 9. Barplot reporting the values of Total Accuracy, the Total Error and the NNP, as defined before in chapter II, for different thresholds used as minimal confidence for a prediction to be provided. The bar at the bottom corresponds to the case without any threshold, namely when simply the most likely class is provided as the predicted one.*

It is worth remarking that by setting a confidence threshold to 0.8, the model is estimated to provide medical operators with more than 92% accurate predictions and <1% error rate, at the price of about 7 times over 100 "I don't know" as output.

### *4) Explainability*

As expected, the filter visualization technique provided limited results in terms of the explainability of the models, generally speaking, because they only provide insights on the features extracted by the first convolutional layers and especially in this case due to the small dimensions of the filters used in the architectures. Indeed, having at most 3x3 kernels it is difficult at first glance to interpret the filtering operations in a meaningful manner. (Figure 10)

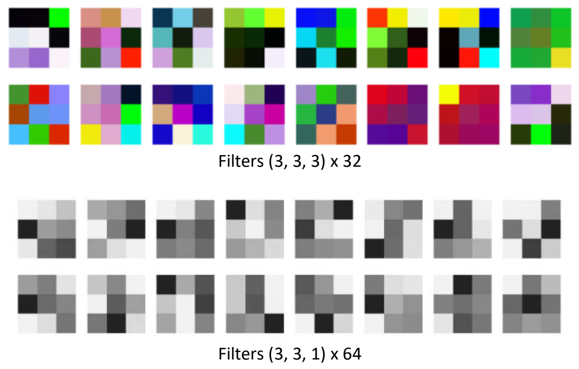


*Figure 10. The kernel of the first convolutional layer is visualized from the EffNetB0 (top) and the ResNet (bottom). Note that the EffNetB0 takes as input rgb images, thus the filters in the first conv layer have 3 channels, too. While the ResNet has been designed to handle one-channel inputs, resulting in one-channel filters. Here the learned weights of the filters are plotted proportionally to their magnitude on the respective colourmap (rgb and grayscale).*

On the other hand, the SAV technique, whose results on the ResNet are reported in Figure 12, provided a much more interpretable visualization of the features extracted by the convolutional layers. Indeed, shapes characterizing the body segments and the bones are clearly outlined. However, it resulted to be still limiting. This is mainly because the visualization of the activation results in interpretable maps only for the shallow part of the network, particularly in non-sequential models, where increasing the number of non-linearities, pooling operations and shortcut merges makes the activations more abstract. Moreover, this technique only provides a partial understanding of how a model is making predictions and does not capture the interactions between neurons in different layers. This is why it has been used in conjunction with other techniques.

LIME provided significant results in terms of explainability as evident in Figure 13. The presented maps depict the relevance of each area on the input image in a Blue-Green-Yellow colourmap. The closer to yellow the area on the map, the more important the feature in assessing the relative class probability of the input, and the opposite in blue. It is evident how according to the input image's label the model is affected the most by different areas of the image. It is also relevant to note that, as expected, the areas that are more relevant for one class (positively contribute to that class), have lower values for the other ones (negatively contribute to other classes). Moreover, when fed with a noisy image, the features used for predictions seem to become meaningless and much less defined, which is another unsurprising and reasonable behaviour shown by the model.

Also, occlusion techniques provided quite relevant insight into which part of the input image contributes the most in predicting the correct class. Indeed, as reported in Figure 14, the areas that negatively affected the most prediction confidence change according to the input label and are coherent with what is highlighted by LIME's technique. Moreover, also in the noisy case, the results as expected are chaotic and less consistent.

GradCAMs generated on EffNetB0 (considering the last convolutional layer having 1280 feature maps) for the same four images previously analyzed with different XAI methods are represented in Figure 15. The red areas are those with respect to which the output prediction gradient is bigger. From those red areas, the model extracted the most important among the 1280 features to predict the final class. The final representation is quite intuitive to interpret and it shows how important are specific areas for the prediction. By visual inspection we can say that it is working quite well in most cases, having the most important areas within the lungs. Also, it is in accordance with the aforementioned methods in most of the images.

The pixel values of GradCAMs were also analyzed, dividing the cases when a correct prediction was performed from those with a wrong output (whatever class they are).

The maximal value for each GradCAM was extracted. The distributions of the two cases are represented by the boxplots in Figure 11: it is evident that wrong predictions have a maximal pixel value bigger than correct ones. To make the distribution more symmetrical and enhance the difference between the two cases a double logarithmic transformation was applied.

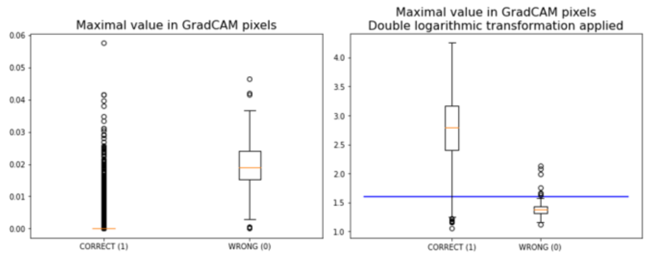


*Figure 11. Maximal values of GradCAMs divided whether the predicted class is correct or wrong. On the left side, the original distribution is represented: a strongly asymmetric tail can be identified, mainly among correct predictions. Conversely, on the right, the double logarithmic transformation is reducing this asymmetry and expanding the difference between the two distributions which can be easily separated by a threshold at 1.6, represented in blue.*

On the transformed variables a threshold (1.6) was set: this simple operation allowed to detect from GradCAMs 96% of wrong classifications, at expense of only 6% of error on correct classifications. This strategy may be used to read in a quantitative GradCAMs and may allow making the prediction even more reliable when the maximal value is over a threshold.

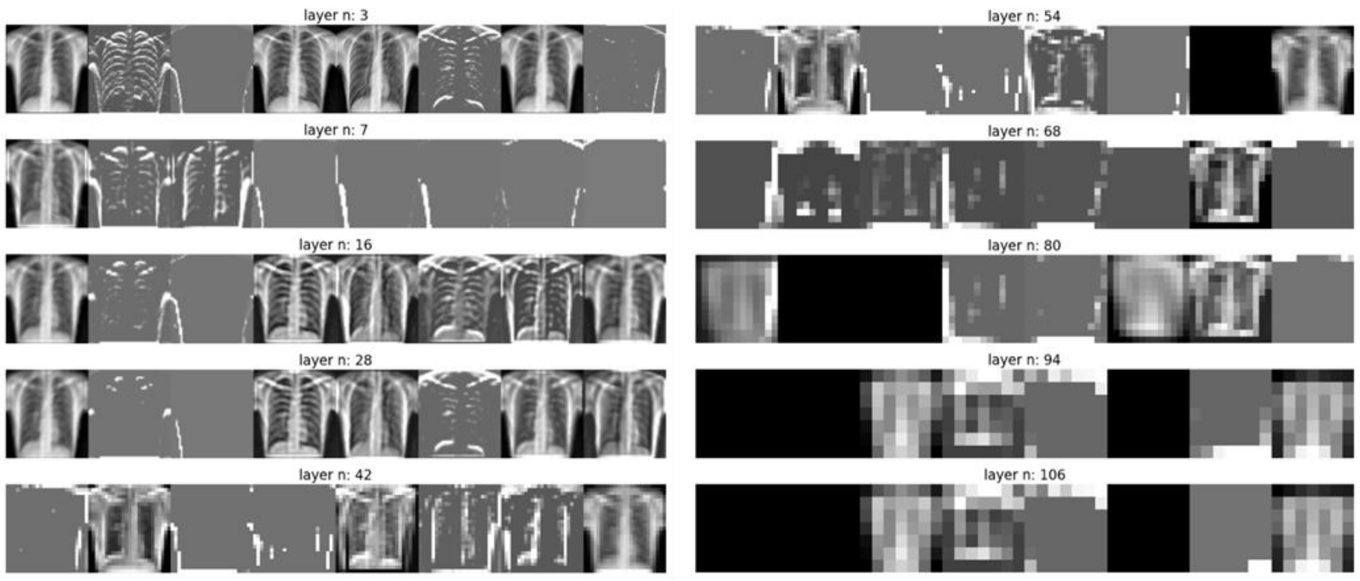


*Figure 12. The first 8 activation maps of the output volume of each residual block of the ResNet are visualized layer by layer.*

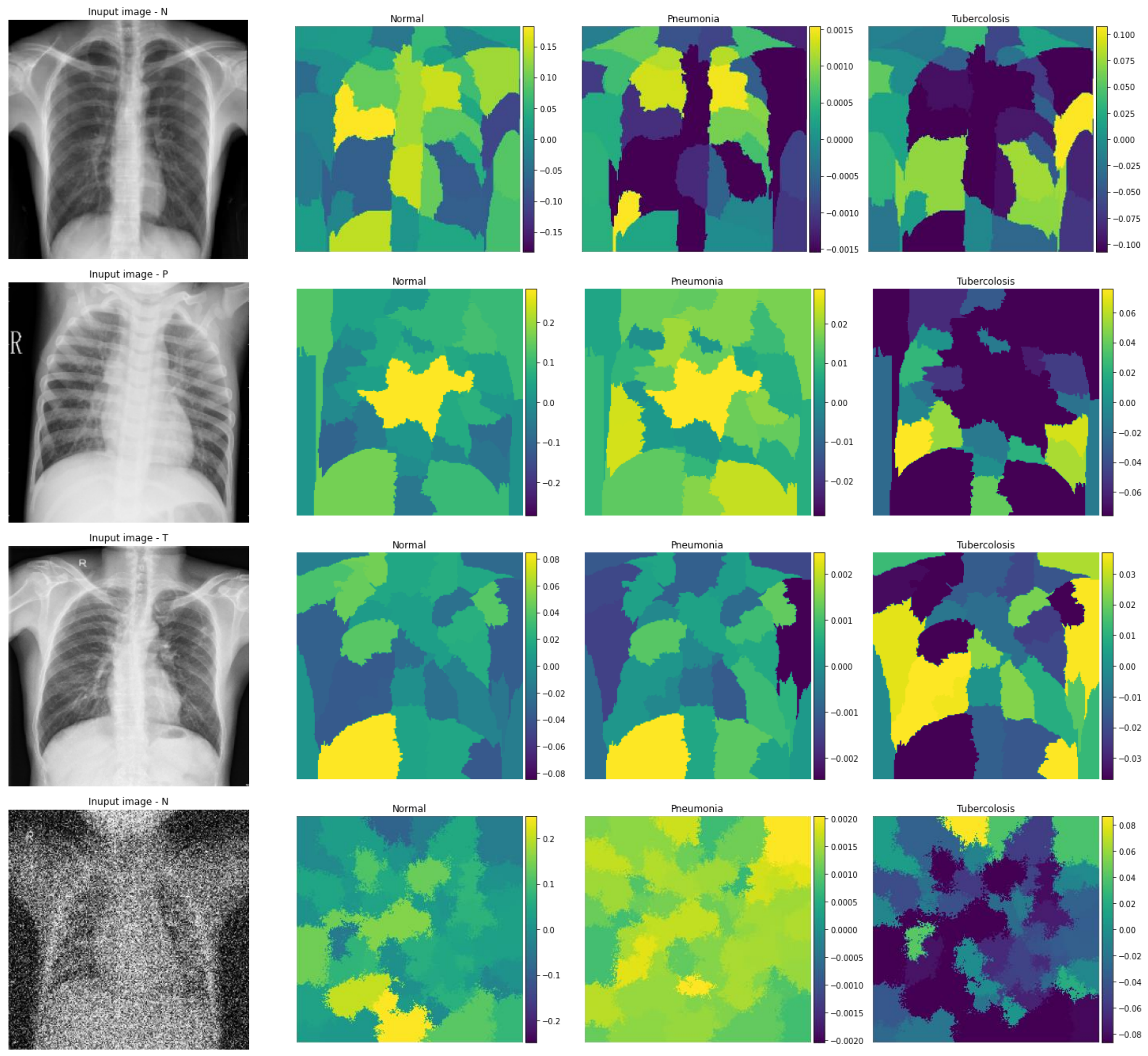


*Figure 13. Results of LIME over the EffNetB0 presented for 4 different cases: healthy subject, pneumonia subject, tuberculosis subject and healthy subject but noisy image, in the order from the top. The reported maps encode the importance of each area on the input image with the colourmap located at their sides. The closer to yellow the area on the map, the higher the relevance of that area in assessing the relative class probability of the input.*

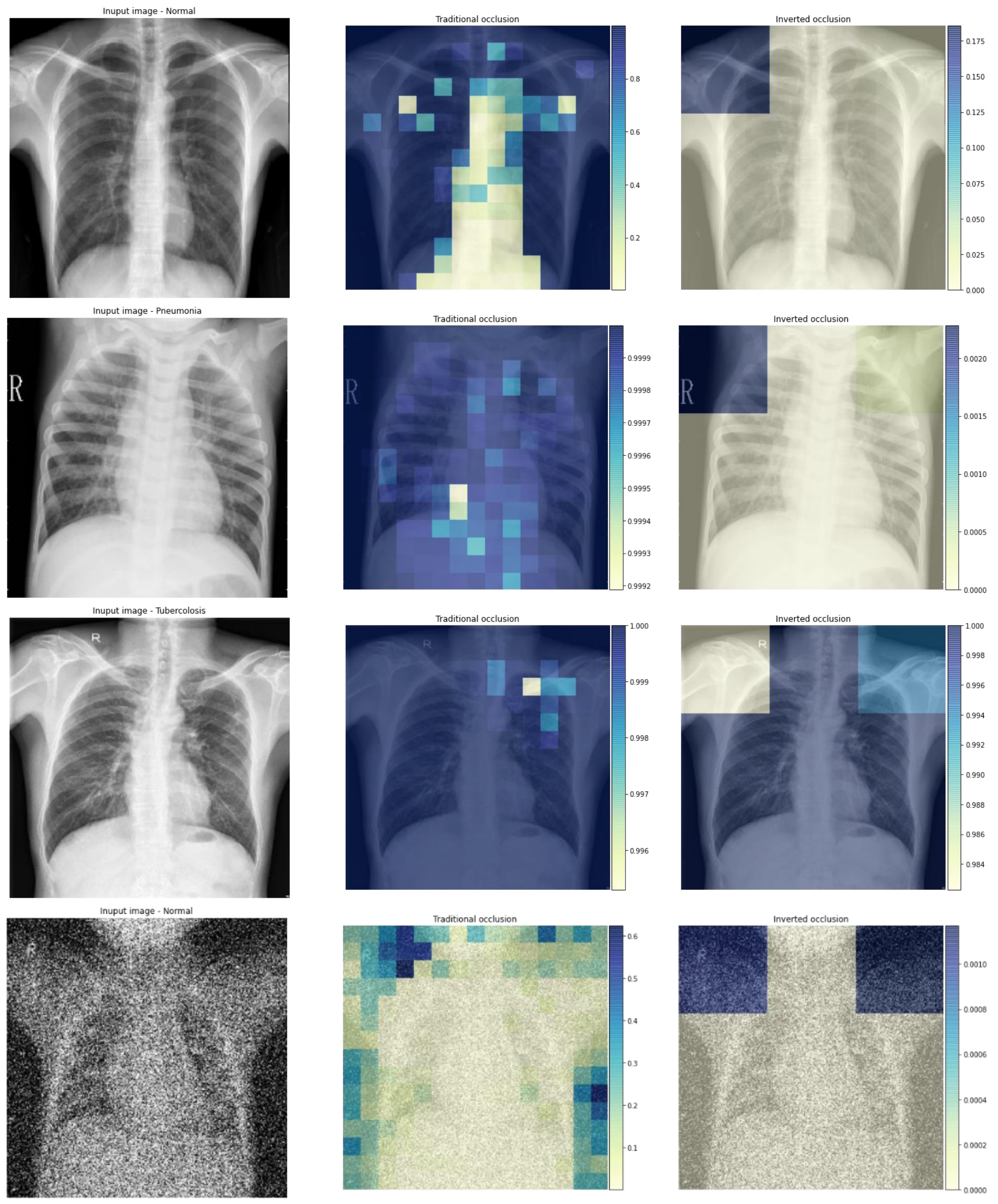
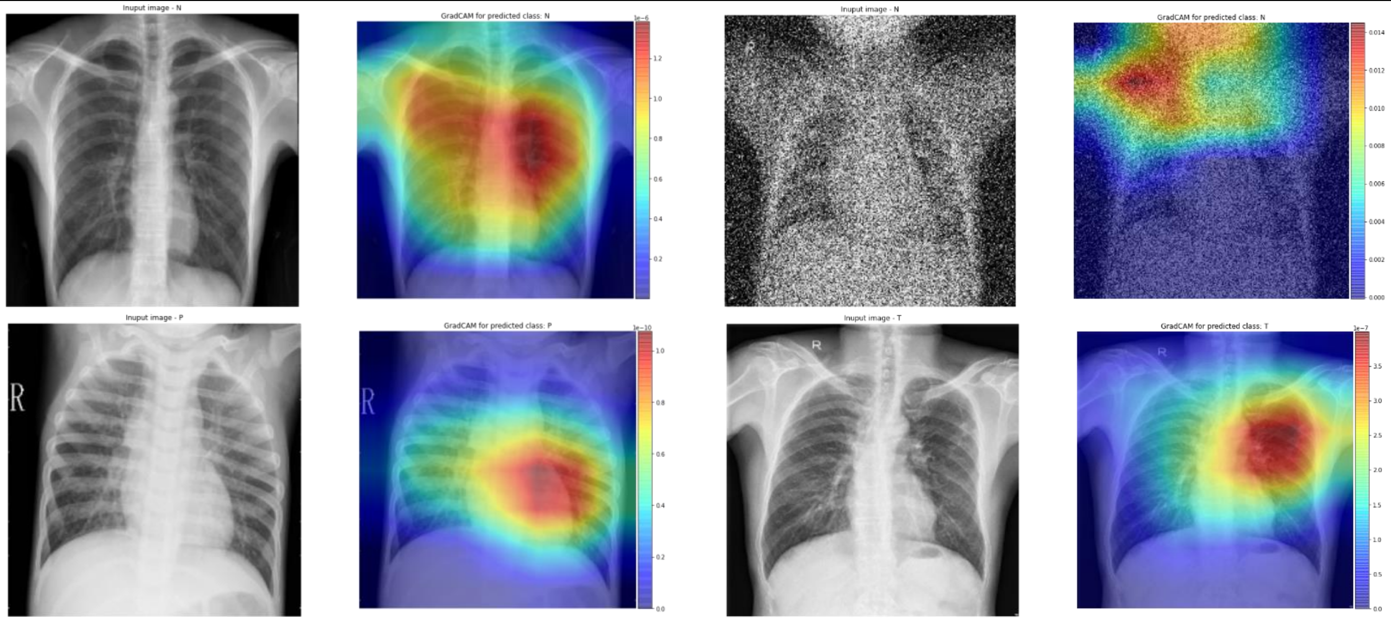


*Figure 14. Occlusion and inverted occlusion results over the EffNetB0 reported for the same 4 cases: healthy subject, pneumonia subject, tuberculosis subject and healthy subject but noisy image, in the order from the top. The reported maps for traditional occlusion encode how the absence of the relative patch affected the prediction: the lighter the patch the more significant the loss of confidence in predicting the class. Conversely, the inverted occlusion maps encode how the presence of the relative patch affected the prediction: the darker the patch the higher the confidence in predicting the class, thus the more informative the patch.*



Inuput image - N
GradCAM for predicted class: N
Inuput image - N
GradCAM for predicted class: N
Inuput image - P
GradCAM for predicted class: P
Inuput image - T
GradCAM for predicted class: T

*Figure 15. GradCAMs generated on EffNetB0 architecture for the same 4 cases: healthy subject, healthy subject but noisy image, pneumonia subject and tuberculosis subject in the order from left to right, from top to down. The values are representing the gradient of the prediction concerning the last layer of feature pixels. The higher the value the most important the features extracted from that area in predicting the output class.*

## IV. Discussion

### 1) *Final considerations*

In light of the results observed in the previous chapter, regarding the designed architecture with two binary classificators, the longer training time was not worth it, considering the unsatisfactory results. Therefore, this approach was early abandoned. The increased error rate is imputable to the double possibility of error: misclassification of the first network will inevitably spoil the performance of the second classificator. Moreover, this suggests that the original idea of combining the two diseases into a single pathological class was based on an incorrect assumption: probably the two pathologies do not share enough clinical features to make this approach effective.

The denoising procedure has emerged as a critical step in the data processing pipeline. In particular, in terms of F1 score over TB class, it was possible to observe a +5% increase for ResNet and a +3% increase for EffNetB0 by using the best strategy. It is important to remark that during the image filtering process, many details are lost, mainly in those cases where the superimposed noise is too intense to be satisfactorily removed. Depending on how each module can preserve or even enhance the information in the images, this can lead to better or worse performance of the final model. Unfortunately, this weakness is common to all denoising strategies and in extreme cases, the best option might be to simply repeat the diagnostic examination. The denoising modules that demonstrated to affect positively the performances of the models is the batch of filters, while FCDN and DAE led to no observable improvement. This could be due to an erratic identification of the intensity of the noise: it might be the case that the two deep networks were trained on images where the added noise was not completely mimicking the noise found in the original dataset. Additionally, it is worth noticing that using a known filter (such as those in the batch of filters strategy) allows being fully aware of the preprocessing operations. Nevertheless, it is also true that more complex deep architectures and other noise artificial sources, that are left for further development, might have led to better results.

A second important aspect that can be appreciated is the fact that the ensemble model proved to be more robust and performing than single models taken individually, as expected. This is probably because this model manages to bring together the strengths of the two best models and evidently mitigate their weaknesses. This is the rationale behind creating an ensemble method and the final performances are worth the increased computational cost and complexity considering the application.

To summarize the prediction analysis results, the chosen model performs well on test data and tends to be "less confident" when the prediction is at risk of error. When using the model, it would be important to exploit this property, which, combined with the use of a threshold on the confidence level, could have numerous advantages. Indeed, as seen above in Figure 9, using a confidence threshold of 80 % would lead to minimising the total number of incorrect predictions to less than 1%, which is a highly desirable condition. This brings to some non-predicted images, but it is a reasonable trade off and it would greatly increase the end user's confidence in the model's output and ensure a much lower level of uncertainty in the provided predictions.

The opportunity to understand how parts of the image and small changes in the input can affect the final output proved to be very valuable. Each of the explainability algorithms showed the ability to capture interesting aspects of the decision-making process except for the very noisy images, which remain mostly unintelligible. Moreover, thanks to explainability methods employed it can be affirmed that the proposed model bases its predictions on a reasonable rationale. Indeed the consistency in results obtained through the different methods, suggests that the model learnt to identify specific characteristics within the lung to recognize the two pathologies. Certainly, the support of an expert in the domain would provide crucial knowledge to verify if the characteristics considered by the model are consistent with clinical practice.

To sum up, the final framework proposed to detect TB and pneumonia from x-ray images in a real case scenario is presented in Figure 16. The core classificator is the ensemble model including EffNetB0 and ResNet, as it resulted to be the best performing one. When a prediction is provided, it is also implemented a confidence control to guarantee an error rate below 1%.

### 2) *Limitations*

In clinical practice, the diagnosis of tuberculosis and pneumonia is made through a multitude of different examinations that require the inspection of samples obtained from the patient. X-ray images are only a part of the diagnostic process as they are not considered to be determinant for a correct diagnosis. For a complete diagnosis, pneumologists dispose of other clinical information which is not included in the proposed model.

Second, the dataset at hand was unfortunately highly corrupted by noise, this issue was only partially solved thanks to preprocessing and confidence thresholding. Translating this model to a real-world setting hopefully eliminates this limitation as higher quality images are expected to be employed.

### 3) *Further developments*

The use of a UNet-like structure for the denoising autoencoder should be considered: it could lead to better results as the skip connections can help preserve spatial information during the upsampling process, improving the performance.

In the very last stages of the project, an important criterion for the a priori identification of noise-affected images was found. It was investigated the amount of information stored within a noisy image and it was successfully found its storage size (kilobytes) is significantly larger than clean ones (equal dimension 224x224 pixels). Therefore, an automatic thresholding on the storage size could be included to refine the existing algorithm for identifying noisy images.

Finally, considering the first limitation mentioned, the inclusion of an additional machine learning module in the decision-making process considering the whole set of diagnostic information used by clinicians might lead to further improvements and higher reliability of the final prediction.

## V. Conclusion

In conclusion, deep learning models proved to be successful techniques for solving the task of detecting TB and pneumonia from x-ray images on the available dataset. Other external datasets should be considered to further validate model performances. Some interpretations of the decision process are also provided. However, to really understand them and extract useful information from them, a collaborative framework involving clinicians with pneumology expertise is required. Work is still needed in the XAI field to increase trust in AI models, but a NN that provides only high confidence predictions and some explanations about the focus areas can be a good starting point. The diagnostic value of such a model should be tested in real clinical settings for verification of the clinical benefits that the model may potentially deliver.

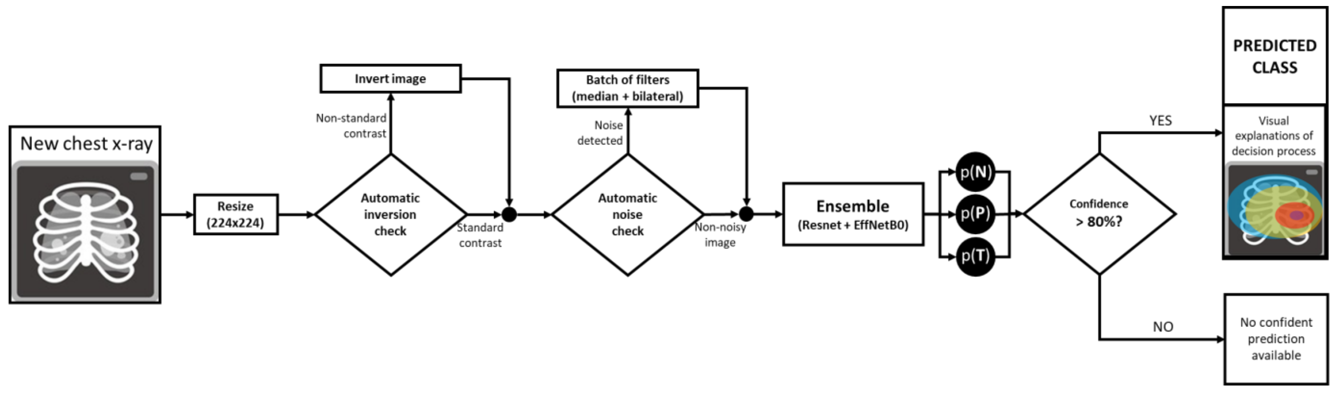


*Figure 16 Schematic representation of the final framework to be proposed in a real case scenario. When a new x-ray image is received, after the preprocessing phase, a first guess by the ensemble model is provided (resulting in 3 complementary probabilities to be in one of the three classes). If the 80% confidence level is reached, a high confidence prediction is provided, along with a visual explanation of the decision process.*

## VI. References

[1] World Health Organization. (2021). Tuberculosis. Retrieved from https://www.who.int/news-room/fact-sheets/detail/tuberculosis

[2] World Health Organization. (2021). Pneumonia. Retrieved from https://www.who.int/news-room/fact-sheets/detail/pneumonia

[3] K. Simonyan, A. Zisserman. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1135–1144.

[5] R. R. Selvaraju, A. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. (2016). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, pages 618–626.

[6] H.M. Yang, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya. (2017). CheXNet: radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225.

[7] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand (2007). A Gentle Introduction to Bilateral Filtering and its Applications. ACM SIGGRAPH 2007.

# APPENDIX A

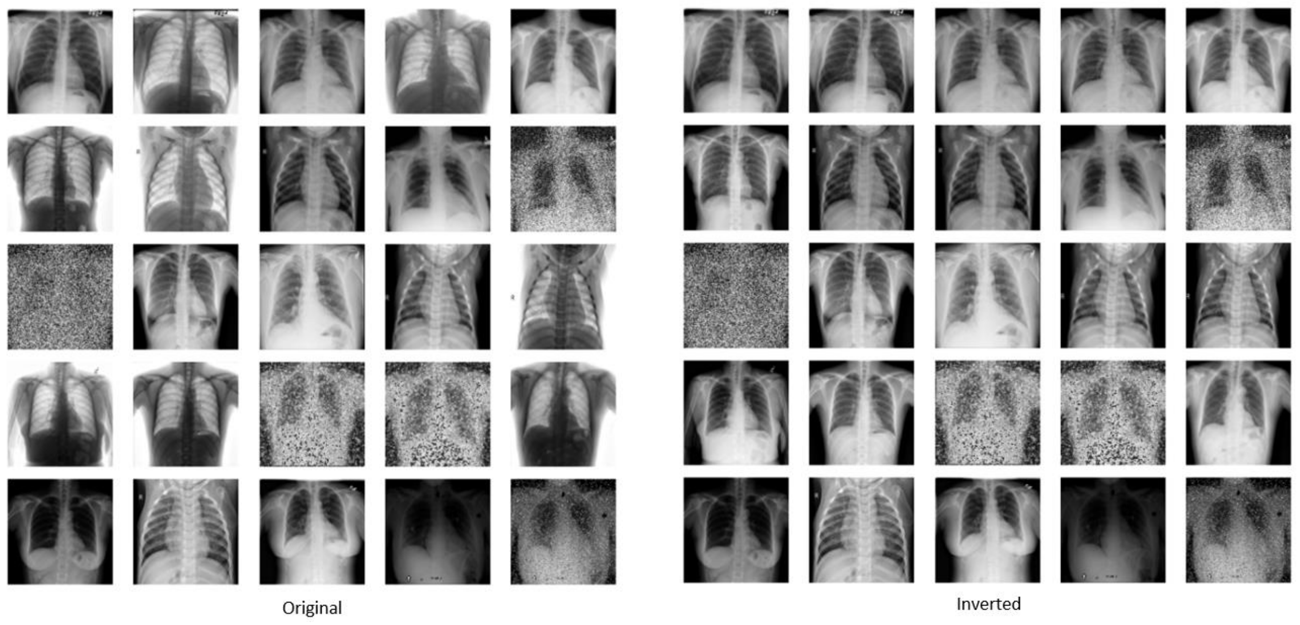


*Figure 1. The input and output of the image inverter. The detection is implemented as a functional that automatically checks the number of pixels in the image's corners [5x5] that are below 0.2 and above 0.8. Then, the prevalence of dark or light pixels determines the probability of the image to be with normal or inverted contrast. The mentioned thresholds were used instead of maximal or minimal saturation [0, 1] to consider also cases with non-optimal contrast conditions. The inversion was performed using a point operator defined as O(x,y)=1 - I(x,y).*

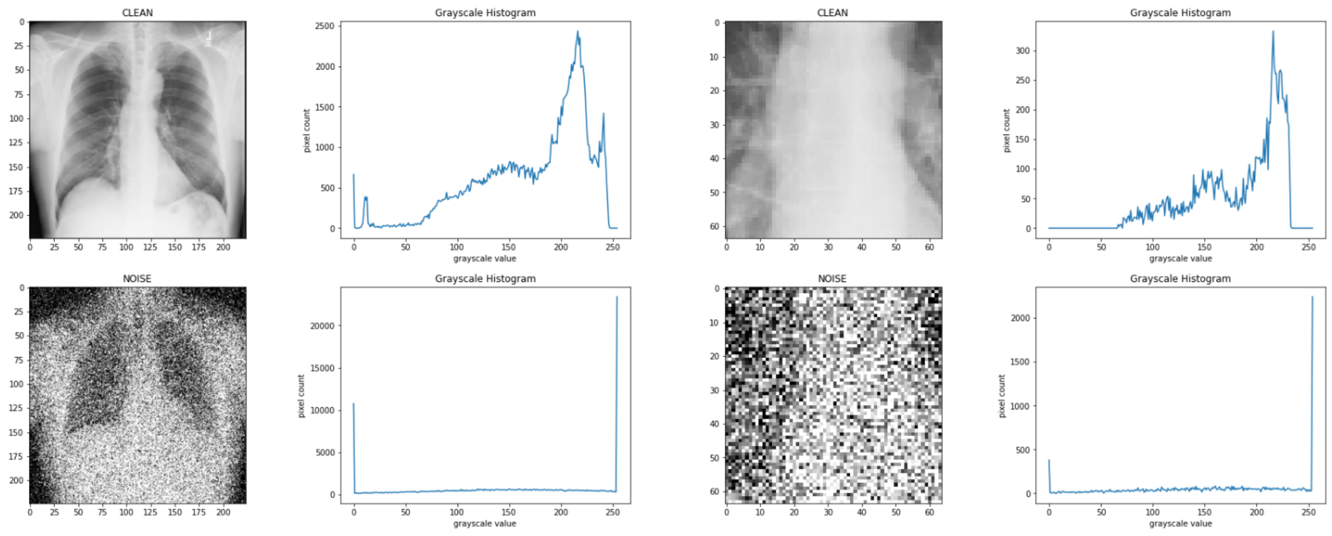


*Figure 2. The Automatic Check for Noise (ACN) considers as likely to be noisy all images with the relative frequency of pixels at maximal intensity over the whole dimension above 1%. Then it performs a second check with the same rationale, but only over the central area of the image, with a lower threshold for the relative frequency ( > 0.7% ). This second check allows to properly exclude from the set of images categorized as noisy those with white patches at the edges or with overexposed zones, while including even those with light noise. Note that a crucial pre-condition for this method to work properly is the preliminary inversion of the images and that all thresholds employed have been tuned over the histograms.*

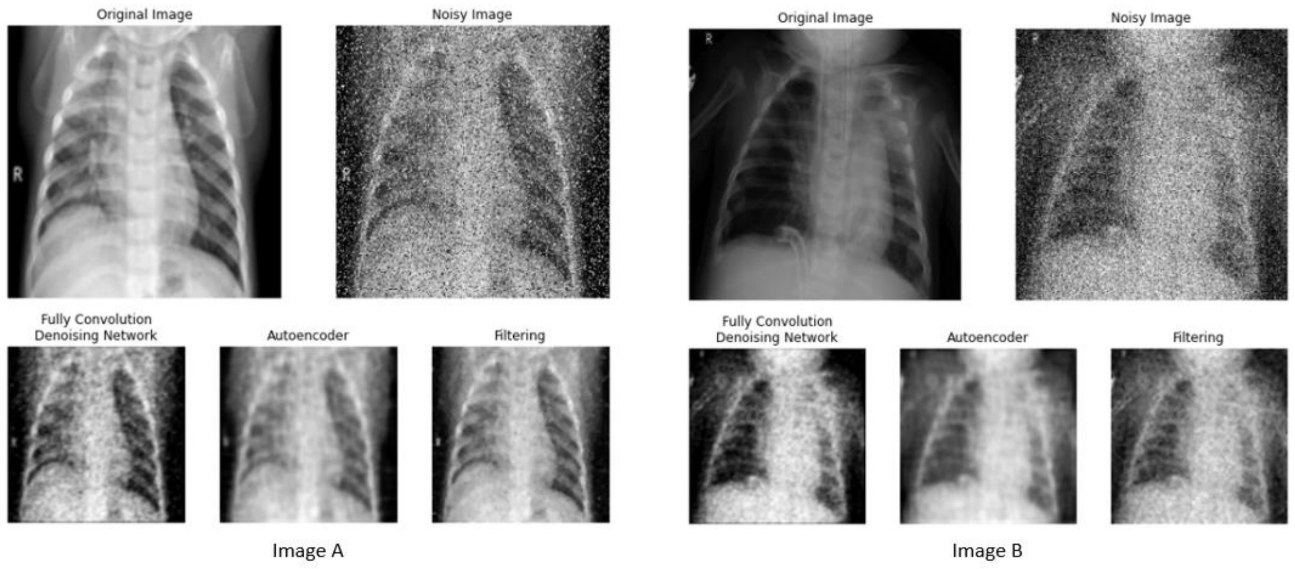


*Figure 3. The outputs of the three denoising techniques considered in this study in comparison for two different chest x-ray images. For both images, the original clean image is presented (top left), next to its noisy version as included in the provided dataset (top right), along with the three cleaned outputs of the denoising modules respectively using the Fully Convolutional Denoising Network (bottom left), the Denoising Autoencoder (bottom centre) and the batch of filters (bottom right).*

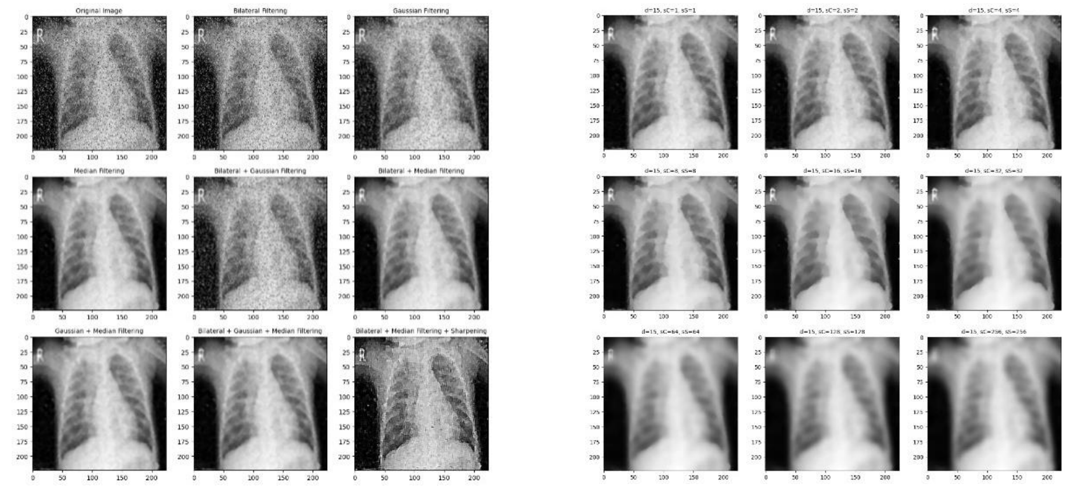


*Figure 4. On the left side, the outputs of different cascades of filters considered in this study are represented. The original noisy image is displayed (top left) along with 8 possible combinations of batch of filters. On the right side, the effect of tweaking bilateral filter parameters is reported, considering an image that was previously processed with a median filter.*