Applied AI in Biomedicine
Chest X-Rays Image Classification

Mattia Cazzolla
ID: 10601164
mattia.cazzolla@mail.polimi.it

Sara Ghezzi
ID: 10867210
sara1.ghezzi@mail.polimi.it

Stefano Vannoni
ID: 10863418
stefano.vannoni@mail.polimi.it

***Abstract*—The aim of this project was to develop and compare different Machine Learning and Deep Learning models built to classify Chest X-Rays images into healthy subjects, or subjects affected with either Pneumonia or Tuberculosis. A robust preprocessing pipeline was implemented to deal with noisy and inverted images. A fine tuned EfficientnetB2 model allowed us to reach an average F1-score of 0.958. Finally, multiple XAI techniques such as Grad-CAM, LIME and Occlusion analysis were used to interpret the model predictions.**

I. Introduction

Among the various diseases that can affect the lungs there are Pneumonia and Tuberculosis.

Tuberculosis is a serious pathology caused by *Mycobacterium tuberculosis* which represents the 13th leading cause of death worldwide and the second leading infectious killer after COVID-19. Tuberculosis is rather easy to cure, however fast intervention is a key factor to reduce significantly the patient's probability of dying. Symptoms, which include cough, fever, night sweats, and weight loss, can remain mild for months leading to delays in seeking care and causing other people infection [1].

Pneumonia, instead, is an acute respiratory infection caused by several agents including viruses, like *Streptococcus Pneumoniae*, bacteria, and fungi. Pneumonia is one of the biggest causes of death among children worldwide. Its incidence is higher in underdeveloped countries where pollution and unhygienic living conditions are high. Furthermore, in these areas, medical infrastructures are often quite inadequate. An infected subject generally faces difficulties in breathing due to inflammation of air sacs that leads to pus formation and liquid diffusion in the pleura. Other common symptoms are coughing and sneezing, mechanisms also exploited by pathogens to infect other subjects [2].

Even though techniques such as CT or MRI are more powerful in detecting these pathologies [3, 4], usually, X-Rays are preferred due to their lower cost and radiation dosage. Nevertheless, the diagnosis could suffer from modest specificity, subjectivity, and difficult interpretation, especially for tuberculosis. To obtain a more accurate diagnosis, it may be useful to have support from Artificial Intelligence (AI).

In this report, we are presenting the approach we implemented to classify Chest X-Rays images into three categories: *Normal*, *Pneumonia*, and *Tuberculosis*. To this purpose, we exploited Artificial Intelligence techniques such as Machine Learning and Deep Learning models. Eventually, we tried to interpret the models' behaviour through explainable AI (XAI) techniques.

II. Materials and Methods

A. Provided Data

The data we were provided consisted of 15470 chest X-Rays images in different sizes and formats (JPEG and PNG) and a CSV file containing, for each image file, the associated label among the three possible ones: N (*Normal*), P (*Pneumonia*) and T (*Tuberculosis*).
In particular, each image's file was named following the structure '*Pxxxxx_n*', therefore we interpreted the 5 digits after the letter *P* as the patient's unique code or ID.

B. Data Exploration

By counting how many images there were per each label we realized that the dataset was unbalanced, with *Normal* being the majority class and *Tuberculosis* being the minority class, as shown in Figure 1.
We evaluated if there were any subjects with more than one image. This is an important aspect to keep in mind when subdividing the dataset into training, validation, and test sets because images from the same subject should be kept in the same set. Almost 30% of the subjects had more than one image associated but none of them had more than two.

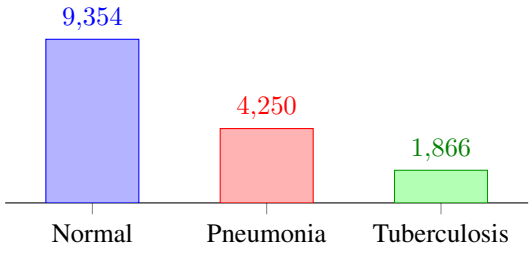
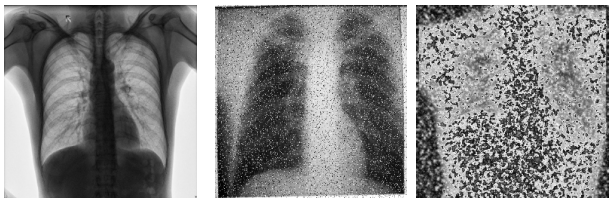


Fig. 1. *Distributions of labels associated to each image*

Finally, we visually inspected the images: all of them were posterior-anterior (PA) projections of the chest, however, they appeared to be quite heterogeneous in terms of geometric and photo-metric properties with some of them being shifted, scaled, and with intensity variations.

It is worth noticing that some images did not contain the whole lungs which were wrongly cut at the apex or at the bases. Furthermore, a significant number of images exhibited peculiar characteristics such as inverted colours or severe corruption with different forms of additive noise (e.g. Gaussian, Salt & Pepper) as depicted in Figure 2.



(a) Color Negative (b) Salt & Pepper (c) Random noise

Fig. 2. *Examples of noisy images. a) Image with inverted colours with respect to standard X-Rays. b) Image with Salt & Pepper noise. c) Image with random noise*

To evaluate how many images were corrupted, we manually labeled them as no-noise, negative, and random noise. Due to the difficulty in distinguishing Salt & Pepper noise from random noise at glance, both cases were collapsed into a single label (Random). The distribution of these labels is shown in Table I from which it is possible to see that a significant amount of images (39.55%) deviate from standard X-rays criteria.

TABLE I
NOISE DISTRIBUTION

No-Noise	Negative	Random
9351 (60.45%)	2752 (17.8%)	3367 (21.75%)

C. Pre-processing

The aim of the implemented pre-processing pipeline was to remove noise (when possible) and improve the quality of the images.
All the strategies explained ahead have been applied after resizing the images to homogeneous dimensions of 256×256. These dimensions were chosen by evaluating the highest among all images (400×400) and reducing it to the closest power of two.
The first step in our preprocessing pipeline consisted in binarizing the images by means of Otsu thresholding technique. The idea behind binarization was to produce homogeneous regions that are similar between intra-class, but different between inter-class images. Figure 3 shows binarized images for the three classes: no-noise, negative image, and random noise.
From each image the following ROI $[216:256, 88:168]$ was extracted (Figure 4a). This region was considered to be the most suitable to determine the image noise class according to these considerations:

- No noise images: most of the pixels in the ROI are white
- Negative images: most of the pixels in the ROI are black
- Random noise images: pixels randomly change between black and white, as in Figure 4b.

(a) Standard (b) Negative (c) Random noise

Fig. 3. *Examples of images after binarization. a) Standard X-Rays image binarized. b) Image with inverted colours binarized. c) Image with random noise binarized*

Making use of the extracted ROIs, two binary classifiers were developed: one to discriminate between negative and non-negative images, and the other to distinguish noisy images from those unaffected by noise.
The first classifier computes the probability of the image *i* to be negative as:

$$P_{neg}(ROI_i) = 1 - \%\, white\ pixels = \%\, black\ pixels$$

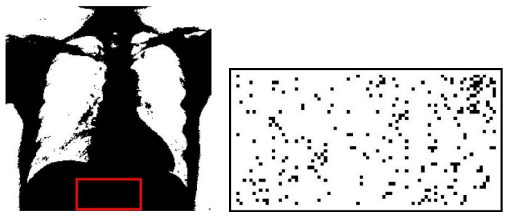
The second classifier first computes the number of flips in the ROI, where a flip is defined as the variation of pixel intensity from $0 \rightarrow 1$ or $1 \rightarrow 0$ between consecutive pixels (e.g. $0111001 \rightarrow 3$ flips). Then, it computes the probability of an image *i* to be a random noise image as follows:

$$P_{rand}(ROI_i) = \frac{flips_i}{ROI_i\ pixels}$$

For both classifiers, the threshold considered for the final classification was the one that yielded the maximum F1-score. According to the classifier results, images classified as negative are inverted in the following way:

$$Processed\ Image = 255 - Image$$

On the other hand, images classified as random noise are processed by applying a median filter with a kernel size of 5×5. The main idea behind median filtering is to fully recover images with Salt & Pepper noise, while, at the same time, trying to smooth those images containing more severe forms of noise.
Finally, we applied histogram equalization in order to improve contrast. We did not apply any normalization since it is already performed by the models themselves.



(a) ROI (b) Random image crop

Fig. 4. *Examples of ROIs. a) Red box representing ROI position in an image b) Extracted ROI from an image with random noise*

D. Classification models and training

To address the classification task we decided to develop and evaluate different models. We initially tested a Machine Learning model, in particular a Support Vector Machine (SVM). Then, we moved to Deep Learning using Convolutional Neural Networks (CNN). More precisely, we trained a CNN completely built from scratch and we made use of Transfer Learning strategy to exploit pre-trained models.

In order to fit and evaluate our model, the dataset was divided into training, validation, and test set (70%, 15%, and 15% respectively) paying attention to keeping images of the same patient in the same set of data, as well as maintaining the same proportions among classes in all sets.

SVM Classifier

SVM model was trained considering the Histogram of Oriented Gradients (HOG) of each image as its features. HOG algorithm works by first dividing the image into blocks, that are further divided into cells of arbitrary size. For each cell, the magnitude and direction of the gradient of pixel intensities are calculated. Gradients of different cells are then concatenated into a single feature vector that represents the image content. HOG exhibits useful proprieties such as invariance to photometric and geometric transformations.

CNN from scratch

The developed scratch model consists of six convolutional blocks, designed for feature extraction, and a classifier. Each convolutional block is composed of: a convolutional layer with kernel size 3×3, a batch normalization layer, a *Relu* activation layer and a Max Pooling layer with pool size 2×2 and stride 2. In each block the number of filters doubles, starting from 16 and ending up to 512, while the spatial extension of the feature maps is halved, it starts from 256×256 ending up to 8×8.

The classification portion of the network is made of a Global Max Pooling layer, a Fully Connected layer with 128 neurons, a dropout layer (p = 0.5) and an output layer with softmax activation function.

Weights in layers with *Relu* activation function were initialized with *He* uniform distribution [5] while weights in last layer were initialized with *Glorot* uniform distribution [6].

The model was trained with *Adam* optimizer with a starting learning rate of 0.0003 which was reduced by a factor of 0.1 every time the F1-score on the validation set plateaued for at least 10 epochs. To address class unbalance, weighted cross entropy was used as loss function. Early stopping was implemented to avoid overfitting.

Transfer Learning and Fine Tuning

We also tested different pre-trained models, hoping they could better address the problem thanks to their higher complexity. In particular, we evaluated the following architectures: Densenet121, EfficientNetB2, EfficientNetB3 and VGG16.

For all of these models, we maintained *ImageNet* pre-trained weights for the convolutional portion (convolutional blocks) to which we appended our own classifier constituted by Global Average Pooling (GAP), two hidden layers with 512 and 128 neurons each, and a final output layer with 3 units and softmax activation function. Each dense layer is followed by a *ReLU* activation function (weights initialized with *He* distribution) and is masked with Dropout (p = 0.3). To reduce overfitting, we also made use of the Early Stopping technique.

Initially, only the classifier was trained, while all layers in convolutional portion of the network were kept frozen. Later, we proceeded by fine-tuning the models allowing the training to update also the weights of convolutional layers.

We trained our networks using stochastic gradient descent with a batch size of 32, Categorical Cross-Entropy as loss function, and an initial learning rate of 0.001 that would decay automatically (factor 0.1) during training.

E. XAI

Explainable Artificial Intelligence (XAI), consists of a series of methods and processes used to better understand results obtained from Machine Learning and Deep Learning models. In medicine, it is extremely important to understand how a model makes its prediction and where it could fail. Since Artificial Intelligence could be a key factor in improving clinical outcomes, it is important to make clinicians trust this approach.

In this regard, we tested different techniques on our best model:

- Fairness methods: Top-two difference and Uncertainty Score.
- t-SNE visualization of latent representation
- Saliency methods: Grad-CAM, Occlusion Analysis, and LIME.

Fairness methods

These techniques aim at quantifying the uncertainty degree of a model when making predictions. In the Top-two difference technique, the histogram of the difference between the two highest probability classes is computed. If uncertainty is low, the difference is expected to be near 1 and the histogram to follow a Power Law distribution.

The Uncertainty Score (US), instead, is a metric defined as:

$$US = 1 - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n \cdot \sum_{i=1}^{n} y_i^2}$$

where n is the number of classes and y_i the probability of class i. If all the classes are equally likely, the US assumes the value 0. Therefore, in a good model, the majority of values are expected to be different from zero, following a distribution that resembles a Power Law.

t-SNE visualization

Convolutional networks can be interpreted as models that gradually transform images into a representation where classes can be separated using a linear classifier. This final representation is usually called latent space. An idea regarding the topology of this space can be obtained by embedding it into two

dimensions. In this way, the low-dimensional representation will show approximately equal distances with respect to the high-dimensional representation [7]. t-SNE method can be used for the embedding. [8]. The results of t-SNE should give us an idea of how easy it is to separate different classes in latent space. Hence, we could associate this easiness with the quality of features extracted by our models.

Grad-CAM

Gradient-Weighted Class Activation Mapping (Grad-CAM) is a technique used to produce "visual explanations" of the decisions made by CNN models. It allows the identification of the regions in an image that had the greatest impact on a specific prediction made by the network. Heatmaps are generated by the linear combination between the activations of a target layer and weights. The latters are computed as the average pixel gradient with respect to a selected class logit, underlining the importance of each feature [9].

Occlusion Analysis

In image classification, a natural question is whether the model is truly identifying the location of the object in the image, or if it is just using the surrounding context to make its predictions. An attempt to answer this question consists in systematically occluding different portions of the input image with a black square (i.e., setting pixels to 0) and monitoring the output of the classifier. The outputs can then be re-mapped into an image, using the coordinates of the center of the black square. Outputs can also be color-encoded, resulting in a heatmap that highlights the effect that occluding pixels, in a specific location, have on the output provided by the model [10].

LIME

Another common technique used in the context of XAI is the Local Interpretable Model-Agnostic Explanations (LIME). This method is able to explain the predictions of any classifier or regressor (agnosticism), approximating it into an interpretable model.

With LIME, a model is asked to predict the class of an image with some super-pixels (i.e., a set of connected pixels) turned on and off. Furthermore, a weight is computed in order to measure the importance of each super-pixel. The more the similarity between the original image and the artificial image, the bigger the weight. In the end, a linear regression model, estimating the importance of the super-pixel in the final prediction, is fitted using computed weights [11].

III. Results and Discussion

A. Pre-processing

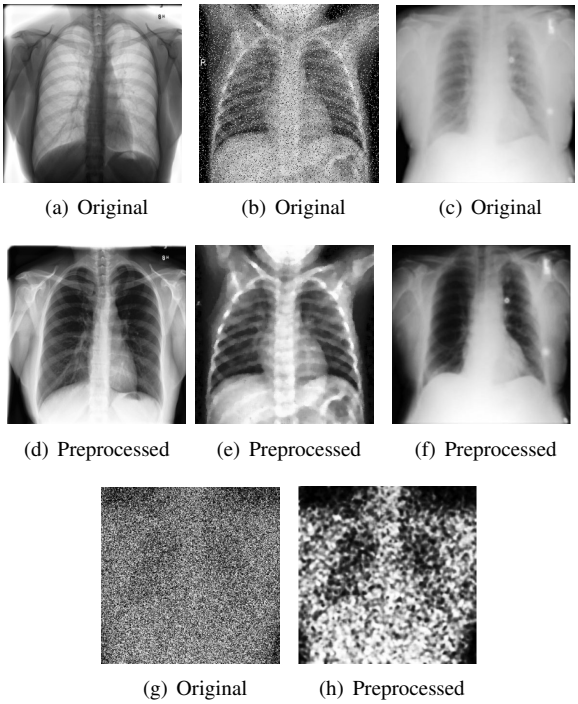
Results of our pre-processing pipeline which included inversion of negative images, median filtering of noisy ones, and histogram equalization are shown in Figure 5. As mentioned above, to evaluate if an image was inverted or corrupted by random noise we used two binary classifiers whose performances are presented in Table II. For the negative and for the random classifier the thresholds chosen were 0.99 and 0.02, respectively.

TABLE II
Binary Classifiers Results

Classifier	***Accuracy***	***F1-Score***	***AUROC***	***AUPRC***
Negative	0.990	0.983	0.99	0.97
Random	0.994	0.990	0.99	0.99

Classifiers results are particularly good meaning that almost all negative and noisy images are correctly identified. However, a small margin of error remains.

Looking at pre-processed images depicted in Figure 5, negative ones are correctly transformed into standard X-Rays images. Regarding Salt & Pepper noise, it is possible to state that it is completely removed after median filtering. However, images highly corrupted by noise remain invariant to the pre-processing pipeline which is not able to recover most of the lost information. Some low-frequency information, however, is retrieved, e.g., overall shape, as we can see in Figure 5h.



(a) Original (b) Original (c) Original

(d) Preprocessed (e) Preprocessed (f) Preprocessed

(g) Original (h) Preprocessed

Fig. 5. *Examples of Pre-processing: a) Negative Image, b) Salt & Pepper Image, c) Poor contrast Image, d) Inverted and Equalized Image, e) Filtered and Equalized Image, f) Equalized Image, g) Severe Noise Image, h) Preprocessed Image*

B. Models results

F1-scores obtained evaluating all of our models on validation set are represented in Table III. Due to the unbalanced nature of the dataset, we decided to consider F1-score as comparison metric instead of accuracy since the latter has problems in representing real performances of classification models with such datasets. F1-score also implicitly provides information about the precision and recall of a model.

SVM Classifier is the worst performing model. However it produced overall good results on *Normal* and *Pneumonia*

classes.
The CNN trained from scratch also performs well on the first two classes, outclassing SVM for *Tuberculosis* class. The comparison underlines the need for more complex models to properly handle high-dimensional data like images.
All pre-trained models, with the exception of DenseNet121, have very good results on all the classes, including *Tuberculosis*, which is the most challenging to predict correctly since it is undersampled in the dataset.

TABLE III
MODEL RESULTS (VALIDATION SET)

Model Architecture	***F1-Score* Normal**	***F1-Score* Pneumonia**	***F1-Score* Tubercolosis**
SVM	0.927	0.953	0.771
CNN Scratch	0.969	0.980	0.882
EfficientNetB2	**0.982**	0.981	**0.936**
EfficientNetB3	0.978	0.982	0.921
DenseNet121	0.968	0.976	0.883
VGG16	0.974	**0.983**	0.900

From the results obtained on validation set, we decided to choose as our best model *EfficientNetB2* which had the best F1-score on both *Normal* and *Tuberculosis* class, while coming in third place on the *Pneumonia* class. In Table IV, results provided by such model on test set are shown. We can notice how these results do not differ by a significant amount from the ones obtained on validation set, suggesting that our model has good generalization capabilities. Confusion matrix of the test set is shown in Figure 6, where we can see the model obtaining a comparable accuracy in all the classes.

TABLE IV
BEST MODEL RESULTS (TEST SET)

Model Architecture	***F1-Score* Normal**	***F1-Score* Pneumonia**	***F1-Score* Tubercolosis**
EfficientNetB2	0.975	0.977	0.921

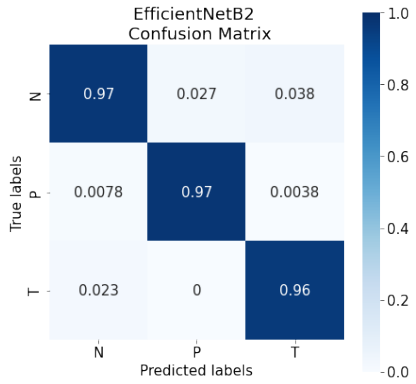
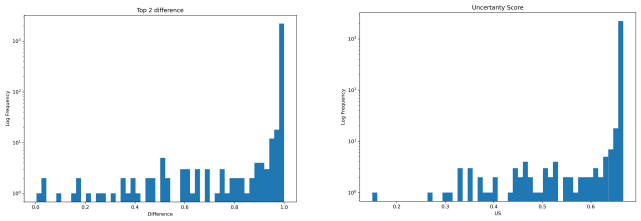


Fig. 6. *Final Model Confusion Matrix obtained for Test Set images*

C. XAI

Fairness methods

Histograms in log scale of the Top-two difference and of Uncertainty Score are depicted in Figure 7. Both histograms have a peak near 1 and 0.66 respectively, suggesting that the model has a low degree of uncertainty and is rarely unsure about its predictions.

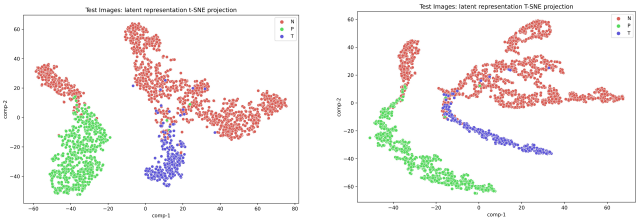


(a) Top-2 Difference (b) Uncertainty Score

Fig. 7. *Histograms in log scale of a) Top-2 difference and b) US*

t-SNE visualization

In Figure 8 we can see t-SNE embedding of test-set images latent space. Figure 8a considers the latent space of the features extracted after GAP layer: it only evaluates the quality of convolutional portion of the network. Figure 8b, instead, considers also the processing made by the classifier portion.
In both embeddings, we can notice that *Pneumonia* and *Tuberculosis* clusters merge with the *Normal* class cluster. However, they do not merge with each other. This kind of topology in latent space suggests that the features extracted by the model make it easy to distinguish between those two classes, as we can also confirm from the results of the confusion matrix in Figure 6.



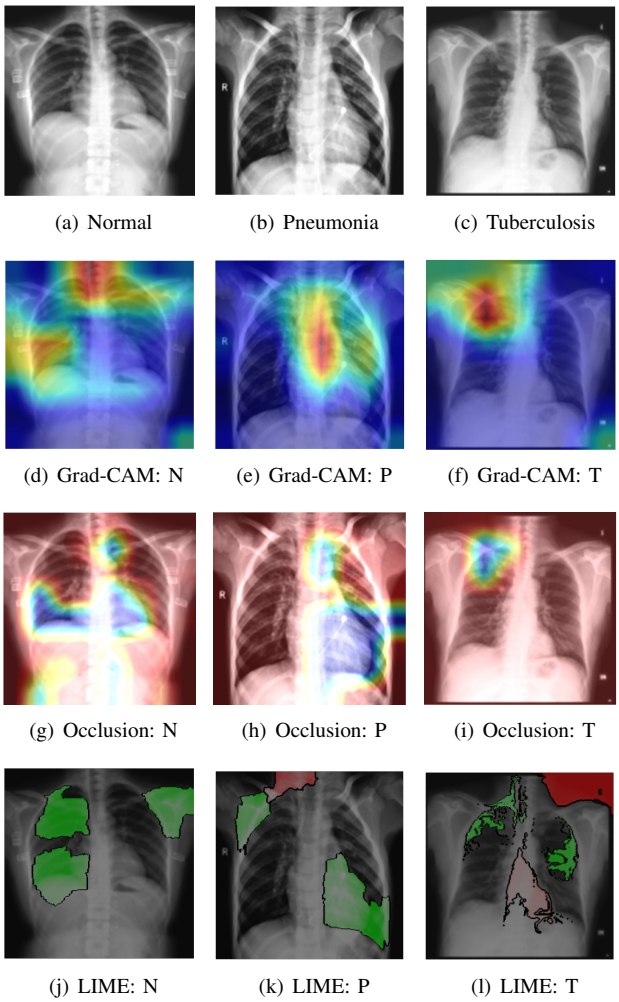
(a) t-SNE: GAP (b) t-SNE: last fully connected

Fig. 8. *t-SNE of the latent space from a) GAP and b) last fully connected layer. Normal, Pneumonia and Tuberculosis are depicted in Red, Green and Blue respectively*

Saliency Methods

In this paragraph we incorporate results obtained with the three saliency methods implemented: Occlusion analysis, Grad-CAM and LIME. In Figure 9 an example for each class and method is shown.
Starting with *Pneumonia*, similarly to what radiologists evaluate [12], the model focuses its attention on regions in the lungs characterized by the presence of opacities which are usually caused by infiltrations and effusions. Other regions that should be considered are the lungs base and profile, which can appear

less defined in case of pneumonia when compared to a healthy subject. Grad-CAM and Occlusion analysis (Figures 9e and 9d) show that the model indeed takes into consideration those regions. Also LIME (Figure 9k) underlines the importance of left lung base and profile in the classification.



(a) Normal (b) Pneumonia (c) Tuberculosis

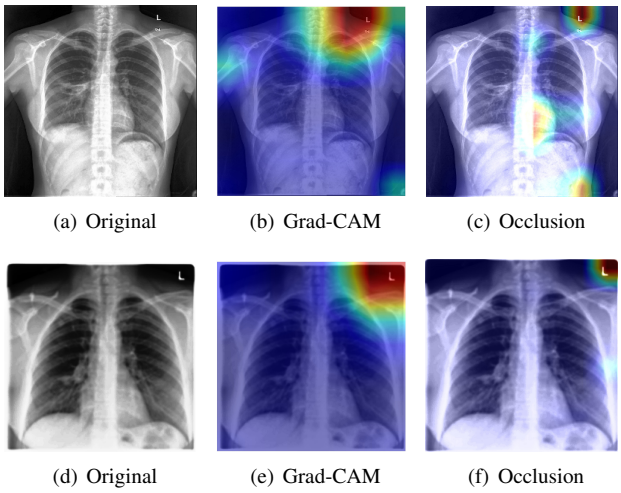
(d) Grad-CAM: N (e) Grad-CAM: P (f) Grad-CAM: T

(g) Occlusion: N (h) Occlusion: P (i) Occlusion: T

(j) LIME: N (k) LIME: P (l) LIME: T

Fig. 9. *Examples of different XAI techniques results on images correctly classified: a) Normal Chest X-Rays Image, b) Pneumonia Chest X-Rays Image, c) Tuberculosis Chest X-Rays Image. Grad-CAM result for d) Normal Image, e) Pneumonia Image, f) Tuberculosis Image. Occlusion Analysis for g) Normal Image, h) Pneumonia Image, i) Tuberculosis Image. LIME result for j) Normal Image, k) Pneumonia Image, l) Tuberculosis Image*

Regarding the evaluation of *Tuberculosis* from Chest X-Rays, typical regions where the illness manifests itself are the middle lobe, the basal segment of lower lobe and the anterior segments of upper lobe of both lungs [13]. The main radiological evidence is a combination of calcified lung parenchyma and the presence of nodules [12].
Our model seems to be consistent with these guidelines since it mainly focuses on the lungs bases and apexes, as it happens in the example in Figures 9f and 9i.
In LIME explanation, Figure 9l, the upper lobe of right lung is also considered along with a region in the middle left lobe. The reason why, in Figure 9l, LIME considers the top-right region to reduce the probability of *Tuberculosis* will be explained in the following paragraph.
Finally, regarding the explainability of *Normal* subjects' images, we expect the model to look at the same regions it has evaluated for the other two classes, but without finding any sign of illnesses. As a matter of fact, in first column of Figure 9 we can see that the model mainly considers upper and lower lungs regions to make its prediction.
In general, we can state that all the explainability models seem to be, for the most part, consistent among each other, even though some differences are present. This is mostly the case for LIME technique which, however, has its results characterized by a certain amount of stochasticity due to the randomness in the process of turning on and off the pixels. This means that evaluating a prediction on the same image twice, LIME generates slightly different results.
A critical flaw we discovered is that, in presence of letter *L* in the upper right corner, used by technicians to make the radiologist know which is patient's left side, the model may focus its entire attention on it, predicting the image as *Normal*, as it is shown in Figure 10. This behaviour was further investigated by making the model predict completely black images with the only exception of a white *L* in the top-right corner. Every time the model would predict the image to be *Normal* with a probability near 1. The model clearly considers the *L* to be a characteristic feature of *Normal* class. A possible solution to this flaw could be to occlude letter *L*, and also *R*, from all the images in training set and train the model again.



(a) Original (b) Grad-CAM (c) Occlusion

(d) Original (e) Grad-CAM (f) Occlusion

Fig. 10. *Examples of two Tuberculosis images predicted as Normal. Both Grad-CAM and Occlusion techniques highlight a written text (letter 'L') in upper right corner of these X-Rays images.*

IV. Conclusions

To face the problem of classifying Chest X-Rays images into *Normal*, *Pneumonia*, and *Tuberculosis*, we first explored the dataset evaluating labels distribution, then counted how many images were associated to the same subject, and finally recognized images corrupted by noise. At this point, we split the dataset into training, validation, and test set assuring that

labels maintained the same distribution among the three sets and that images belonging to the same subject were in the same set. After resizing the images (256×256), we implemented a pre-processing pipeline developing two binary classifiers able to identify images to be inverted or to be filtered with median filter. After pre-processing the images (inversion, filtering, equalization, and rescaling), we trained different models to address the task. The best model, *EfficientNetB2*, was chosen by looking at F1-scores obtained on validation set. Calculating the performances on the test set, we obtained F1-scores of 0.975, 0.977, and 0.921 on the *Normal*, *Pneumonia* and *Tuberculosis* class respectively.

Finally, we applied different XAI techniques to our best model in order to understand better its behaviour. We used fairness methods (Top-two difference and US score), t-SNE visualization, and saliency methods (Grad-CAM, Occlusion analysis, and LIME). Explainability techniques revealed to be quite consistent among each other and also with what a radiologist typically looks at in these cases.

The overall pathway followed in this project is outlined in Figure 11.

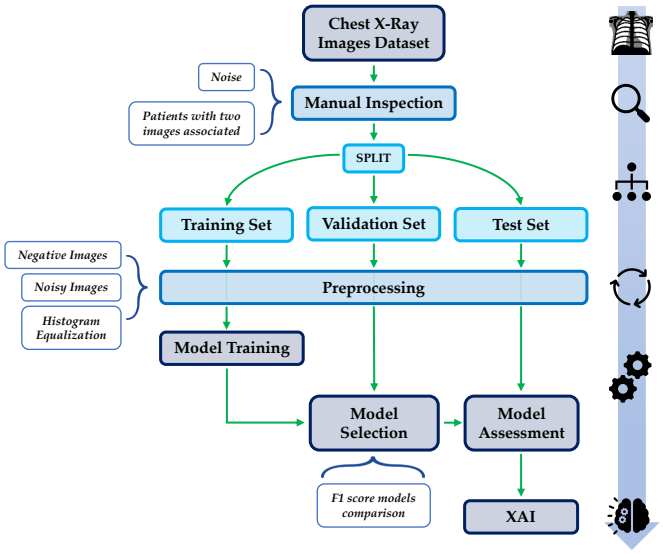


Fig. 11. *Schema of the pipeline we followed to address the task of Chest X-Rays Image Classification*

References

[1] WHO Tuberculosis. *World Health Organization.* (2023), https://www.who.int/news-room/fact-sheets/detail/tuberculosis

[2] WHO Pneumonia. *World Health Organization.* (2023), https://www.who.int/news-room/fact-sheets/detail/pneumonia

[3] Santosh KC, Allu S, Rajaraman S, Antani S., "Advances in Deep Learning for Tuberculosis Screening using Chest X-rays: The Last 5 Years Review." *J Med Syst.*, 2022

[4] Kundu R, Das R, Geem ZW, Han GT, Sarkar R., "Pneumonia detection in chest X-ray images using an ensemble of deep learning models." *PLoS One*, 2021

[5] He, K., Zhang, X., Ren, S., & Sun, J. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In *Proceedings of the IEEE international conference on computer vision*, 2015.

[6] Glorot X., Bengio Y, "Understanding the difficulty of training deep feedforward neural networks.", *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[7] Stanford, CS231n Convolutional Neural Networks for Visual Recognition (2016). https://cs231n.github.io/understanding-cnn/

[8] L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*, 2008

[9] Ramprasaath R. Selvaraju, Micheal Cogswell, Abhishek Das, Ramakrishma Vedantam, Devi Parikh, Dhruv Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization." *ArXiv.org*, 2016

[10] Zeiler, M. D., & Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV*, 2014

[11] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier.", *ArXiv.org*, 2016

[12] Katz DS, Leung AN. Radiology of pneumonia. *Clin Chest Med.*, 1999

[13] Bhalla AS, Goyal A, Guleria R, Gupta AK. Chest tuberculosis: Radiological review and imaging recommendations. *Indian J Radiol Imaging*, 2015

External Link

Code available here: GitHub Repository