Applied AI in Biomedicine
Chest X-Rays Image Classification

ABSTRACT: in this project, several Deep Learning models have been evaluated to tackle the problem of chest X-ray images classification in the field of lung diseases, such as Tuberculosis and Pneumonia. Particular attention has been given to models' explainability since in clinical scenarios it is important to understand where the model is seeking information and to establish the confidence of each prediction.

I. INTRODUCTION

Among many lung diseases, two of the most serious ones are Pneumonia and Tuberculosis. Pneumonia is caused by a bacterial, viral, or fungal infection and can be fatal if not treated in a timely manner. Tuberculosis is an infectious disease caused by a bacterium called Mycobacterium Tuberculosis that can cause damage to the lungs and other organs in the body. Both are transmitted through respiratory droplets and can be prevented by a healthy lifestyle, personal hygiene, and vaccination.

Diagnosing tuberculosis (TB) and pneumonia using chest X-rays is a common and effective way to detect and diagnose both conditions. A chest X-ray is a quick and painless procedure that uses a small amount of radiation to take a picture of the lungs, heart, and other structures in the chest. It is used to detect abnormalities or changes in the lungs due to TB or pneumonia.

When analysing a chest X-ray for TB or pneumonia, the doctor or radiologist looks for the presence of certain characteristics in the image. In TB, the X-ray can reveal white patches or streaks that indicate the presence of TB bacteria. In pneumonia, the X-ray will show cloudy patterns in the lungs, typically in a symmetrical pattern, indicating an infection.

II. MATERIALS & METHODS

A. Dataset

The dataset used for the project consists of 15470 grayscale chest X-rays images of different sizes of several subjects, divided into healthy (N), tuberculosis (T) and pneumonia (P) patients. We resized them all to 256x256 and we stacked the grayscale channel three times to get an RGB format.

The dataset presents a strong unbalance since there is a large majority of N-class images (60.5%) and a minority of P (27.5%) and T (12%) images.

We performed a split of the provided dataset into train set, validation and test set, respectively composed by 75%, 15% and 10% of the available images.

Furthermore, while performing the above splitting we carefully checked that all the images belonging to a patient falls into the same set to avoid possible biases.

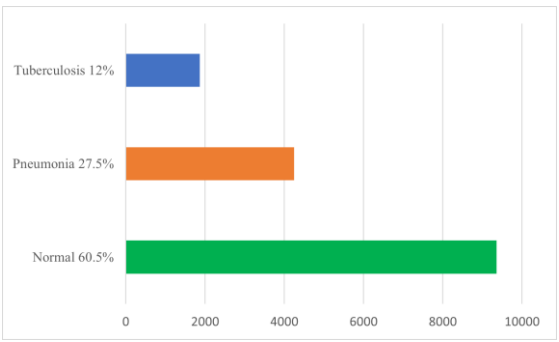

Fig.1 Images Distribution over classes

B. Data Exploration & Unbalancing

On a first visual investigation of the images, we noticed that some of them were corrupted by noise, mostly salt&pepper and random, some were cropped or shifted, and a small portion had inverted contrast.

To deal with the latter issue and to make our models invariant to this transformation we decided to create a new dataset in which the training set contained each original image and its negative. We will call this dataset DataNeg.

Additionally, to cope with unbalance three different approaches have been considered: we designed a GAN (General Adversarial Network) to obtain new images belonging to class T starting from the original ones in the dataset, we trained our models with a weighted loss to highlight samples of the least represented classes.

We also tried to use a training set in which we oversampled images of class T by performing minor augmentations, called AugDataNeg.

C. Preprocessing

To reduce noise and increase models' learning capabilities we exploit different approaches of pre-processing.

The first solution we tried was the application of noise reduction filters:

- Gaussian (G)
- Median (M)
- Bilateral (B)
- Combination of different filters

Then, we moved to Deep Learning denoising techniques designing a DAE (Denoising Auto Encoder).

Since this model needs to be fed with clean images we designed an automatic noise detection algorithm based upon the Noise Variance Score[1].

$$\sigma = \sqrt{\frac{\pi}{2}} \frac{1}{6(W-2)(H-2)} \sum_{image\ I} |I(x,y) * N| \qquad (1)$$

Noise Variance Score

Where N is the matrix:

$$N = 2(L_2 - L_1) = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad (2)$$

Images with a score lower than 1 can be considered not affected by noise, so by setting that value as our threshold we were able to extract a subset of DataNeg suitable for the DAE training.

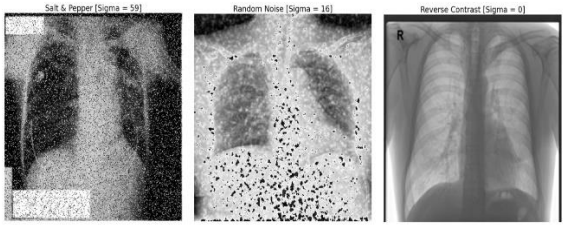

Fig.2 Examples of different types of noisy images with the associated σ score. From the left: Salt & Pepper, Random Noise, Inverted Contrast, with associated sigma-score of 59,16,0

D. Models architecture

We tried different Deep Learning models to perform classification.

The idea we followed was to gradually increase model's complexity to assess whether a relatively simple/shallow model was able to perform a correct classification.

By following this reasoning our first evaluated model was a Convolutional Neural Network built from scratch.

Indeed, we stacked together multiple convolutional blocks with an increasing number of filters, and a dense classifier. Each convolutional block is composed by a 2DConvolutional layer, a ReLU Activation layer and 2DMaxPooling layer. The last convolutional block is then followed by a GAP layer [2] that precedes the classifier, which is designed with two dense layers before the output and a Dropout layer before each of them.

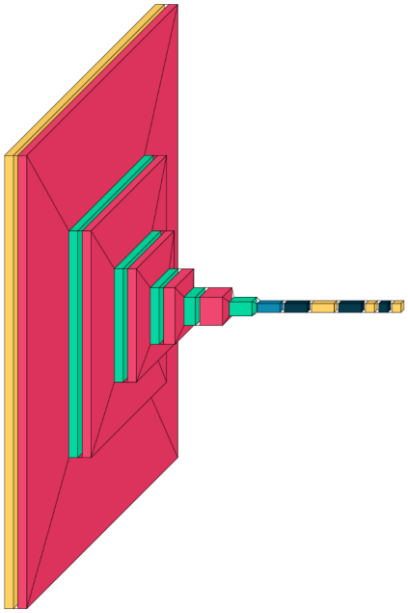

Fig.3 CNN structure

Then, we moved to pre-trained models (SuperNets) with a deeper and more complex structure, hoping to have more reliable and accurate predictions.

We only kept the feature extraction part of these models since we wanted to exploit their feature extraction capabilities. Moreover, to make models suitable for our problem we designed an ad-hoc fully connected (FC) classifier.

Initially we tried to use the same classifier configuration for all the SuperNets to perform a first SuperNet selection. This was composed by two hidden layers with respectively 256 and 128 neurons and a Dropout layer after each of them. Then a SoftMax layer with 3 neurons.

The SuperNets we tested are, in order of complexity, VGG16, DenseNet and EfficientNetB7.

E. Models training

The training of these networks is a two-step end-to-end training, composed of a first training procedure in which only the classifier's weights are updated while keeping the feature extractor's weights untrainable. Then in a second Fine Tuning step a variable number of layers of the feature extractor are trained together with the classifier's ones, to make the network adapt to the classification task.

The models were compiled with Categorical Crossentropy as loss function a learning rate of 10^{-3} for the first training procedure. In the second part (Fine Tuning), we used an adaptive learning rate starting from 10^{-4}. We also tested different batch sizes values to find the optimal balance between computational load and training variability.

To deal with overfitting we used several regularization techniques, such as Dropout, Weight Decay and Early Stopping.

Furthermore, we also included class weights, to weight our loss function as stated in Section B.

Additionally, we performed a minor image augmentation which consists in applying small transformations to the training set images before feeding them to the network.

This enhances the generalization ability of the networks by making them invariant to the applied transformations as well as reducing overfitting.

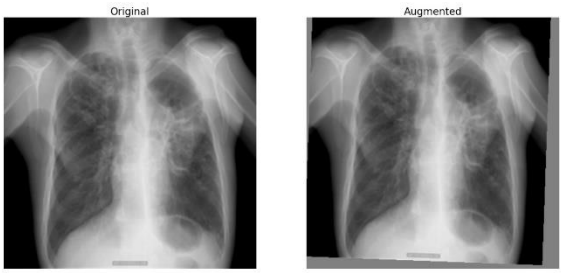

Fig.4 Comparison between original and augmented image

F. XAI

AI models are increasingly being used in decision making processes in many fields of interest. However, the reasoning behind these models' outputs is difficult to understand and interpret and this can be an issue in clinical scenarios where each decision could be critical for the patient.

This is the reason why, more and more studies have focused recently on finding a way

to explain the logic behind AI outcomes, introducing what is called eXplainable Artificial Intelligence (XAI).

XAI can, indeed, be integrated in what is called 2.0 Health Care which is characterized by the cooperation of doctors and artificial intelligence making AI decisions clearer and more interpretable by clinicians.

In this paper we explored four XAI techniques:

- Gradient-weighted Class Activation Mapping (Grad-CAM),
- Local Interpretable Model-Agnostic Explanations (LIME)
- Occlusion
- Uncertainty Quantification (Monte Carlo Dropout)

Since the explainability is a relevant part of the project we used Grad-CAMs [3] to check step-by-step how they are affected by changes in model development. Then, we applied the other XAI techniques on the final model.

F.1. GRAD-CAM

This technique consists of computing the gradient of the target class score with respect to the output feature map of a convolutional layer and then using these gradients to weight the feature maps. So, we can compute a coarse localization map highlighting the important regions in the image for predicting that specific concept.

Grad-CAMs were chosen due to their superior performance and flexibility compared to CAM. Furthermore, deriving Grad-CAMs does not require modification of the network topology.

F.2. LIME

The second technique adopted is LIME [4]. We decided to use it since it is both independent from the model and works well with images. Its basic idea is to locally approximate the behaviour of a black-box model by training a simple, interpretable model on a small subset of data close to our point of interest. First a set of perturbed samples (inputs with small variations) is generated around the point of interest. Then, the black box model is used to predict the outcomes of these perturbed samples.

Finally, LIME trains a simple and interpretable model, such as a linear model or a decision tree, on the perturbed samples and their corresponding predictions. The simple model is then used to explain the predictions of the black box model. Variation of the images are generated by removing (i.e. painting with uniform colour) region of the image with similar colour distribution.

F.3. OCCLUSION

Third technique we used is occlusion, a technique able to show the importance of a certain region of the input image in the final prediction. It works by blocking portions of the image and observing how the model's prediction is affected.

Thanks to this technique is possible to determine which areas of the image are most important for the final prediction and which are not. We also used Inverted Occlusion, in which instead of hiding a portion of the image and perform the classification on the remaining part, the inference is done only on the provided portion.

The remarkable advantage of these three techniques is that they provide visual explanations which are clear and easily interpretable also by non-expert end-users. This is key in the process of including AI in other fields, like healthcare.

F.4. UNCERTAINTY MEASUREMENTS

Besides providing visual explanations, in order to make an effective XAI, it is also important to quantify how much the model is confident about its own predictions.

We can do this by looking at the SoftMax output distribution, which expresses the probability of the image being associated to each class. If the distribution is sharp over the predicted class the model can be considered as confident.

In order to get a more refined probability distribution, hence, to better quantify the uncertainty we used Monte Carlo Dropout.

This consists of applying dropout at test time, while repeating each prediction 100÷1000 times, then averaging the SoftMax output over all the predictions. The result is the same as having a statistically relevant ensemble of models, performing the same task, from which to calculate a true realistic classification output.

We performed Monte Carlo Dropout by introducing a dropout layer, with 0.1 Dropout, after each MaxPooling layer of the VGG16.

Subsequently, we performed 100 predictions of each image while keeping the dropout active to use a slightly different network each time. Then we calculated the mean of the 100 SoftMax output and we used it as feedback of the uncertainty of the model over the predictions.

Additionally, to have a numerical value that quantifies how much a prediction is uncertain we computed the Jain's Fairness Index [5]:

$$J(x_1, x_2, \dots x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n * \sum_{i=1}^{n} x_i^2} \qquad (3)$$

Jain's fairness index. x_i is the i-th SoftMax output associated to the i-th class.

This is an index of uncertainty, in the range [1/n,1], where 1 means maximum uncertainty.

So, as result, each prediction is associated with a bar plot showing the post-Monte Carlo Dropout distribution of the SoftMax over classes and the above index quantifying how much the prediction is uncertain.

III. RESULTS

A. Model assessment

In this section we will present the results of our models, going through both feature extractor and classifier comparison. These have been trained on DataNeg, that turned out to be the optimal dataset.

Feature Extractor	Accuracy	F1-score [N, P, T]
CNN	0.89	[0.92, 0.92, 0.84]
VGG16	**0.93**	[0.94, 0.93, **0.87**]
DENSENET	0.93	[**0.95**, **0.94**, 0.80]
EFFNETB7	0.93	[0.94, 0.93, 0.86]

Table 1. Evaluation of proposed Feature Extractor on local test set

As can be seen from Table1 the CNN model built from scratch shows an accuracy of 0.89 and an F1-score of 0.84 for class T.

When it comes to SuperNets, VGG16 with an accuracy of 0.93 and an F1-score of 0.87 for class T, was according to our evaluation the best feature extraction module due to its slightly increased F1-score for class T and its shorter and less computational demanding training.

Classifier [neurons]	Accuracy	F1-score [N, P, T]
GAP + SOFTMAX	0.93	[0.94, 0.93, 0.86]
[256]	**0.93**	[**0.95**, **0.94**, **0.87**]
[256 - 128]	**0.93**	[0.94, 0.93, **0.87**]

Table 2. Evaluation of the Classifier on local test set

After selecting VGG16 as feature extraction module, we tuned the classifier architecture.

According to Table 2 the best configuration was the one with a single dense layer after the GAP layer, with 256 neurons.

B. Preprocess & dataset

The following section considers the effect of the denoising techniques on the performances.

To this extent, Table 3 shows the results for the most promising denoising techniques. Combination of Median and Gaussian as well as Bilateral filter have been excluded since they do not provide meaningful results.

Denoising	**Accuracy**	**F1-score [N, P, T]**
NONE	0.93	[0.95, 0.94, 0.87]
GAUSSIAN	0.94	[0.95, 0.95, 0.88]
MEDIAN	**0.95**	[**0.96**, **0.96**, **0.89**]
DAE	0.94	[**0.96**, 0.95, 0.87]

Table 3. Comparison between denoising techniques

Median filtering was in our evaluation the best denoising technique, both in terms of accuracy and F1-score.

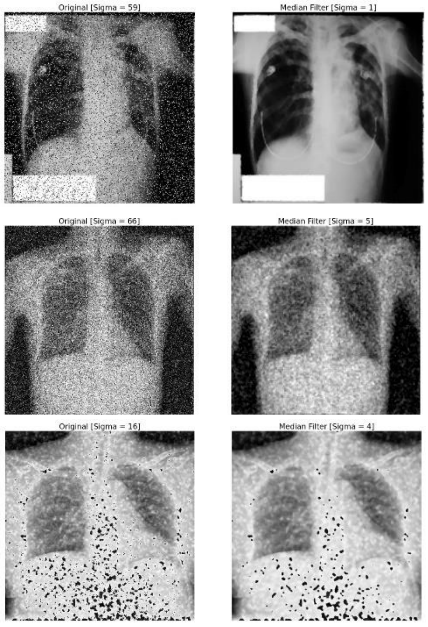

Fig.5 Effect of Median filter on images with Salt & Pepper (first two rows) and Random noise (third row) with the relative σ-score. Original on the left, filtered on the right.

C. Gan

Unfortunately, even though we only trained GAN with clean images belonging to class T, due to our limited computational capabilities, we were not able to obtain a generator able to create new plausible images.

Fig.6 GAN output images

As we can see in Fig.6 what we obtained was something that resembles the general structure of chest X-rays, black background with white patches in the centre, but not that much to be considered as different from noise.

However, even though the generator had been able to generate new truthful X-ray images we still would not have been sure about whether they belonged to class T.

So, one possible solution might be to have a physician annotate those images.

D. XAI

Here we will present how our choices in the model training procedure and denoising affect the GRAD-CAMs output, to show their validity.

Fig.7 shows the effect of performing training set augmentation.

Prior to augmentation it was common for the model to focus on areas such as the shoulders and the small letters (L, R) in the corners of the image.

We wanted to avoid this as it means that our model is classifying images by considering areas unrelated to the target pathology.

The results of the minor augmentation we used is that the model is, now, less prone to focus on these areas and more on lungs.

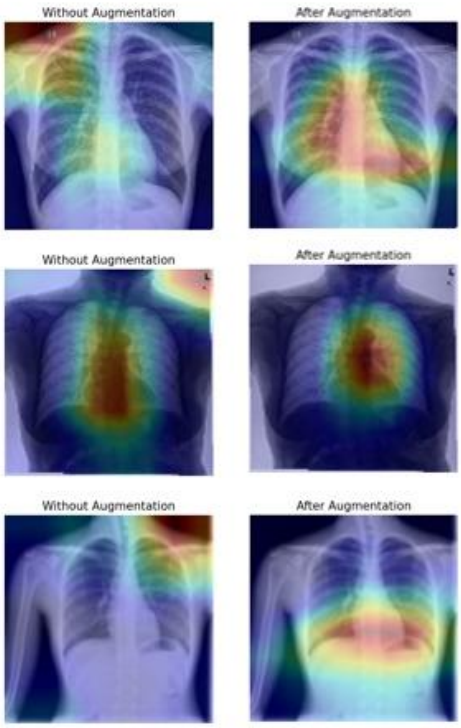

Fig.7 Grad-CAMs before (left) and after applying augmentation (right). We can see how the model correctly learns where to focus on, without taking shoulders or image artifacts into consideration.

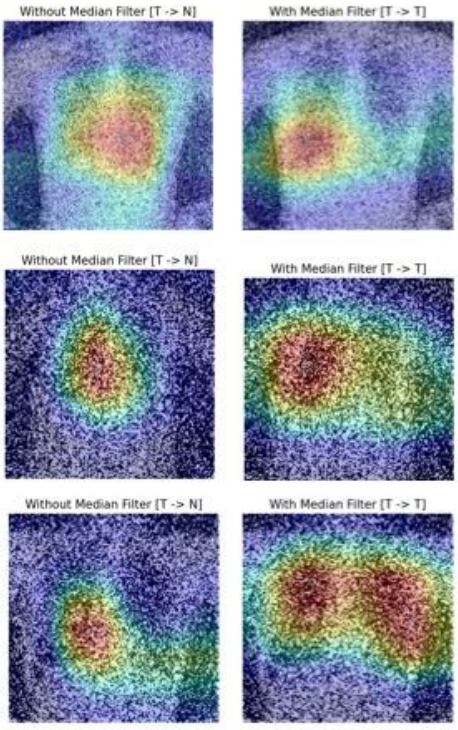

Fig.8 Grad-CAMs for different class T images before (left) and after applying a median filter (right). We can see that after the application of the median filter the heatmap is refined and the classification is correctly performed.

Another relevant comparison highlighted by GRAD-CAMs is about how Median filter affects both predictions and model's areas of focus. As shown by Fig.8 the application of the filter helps the model to correctly predict noisy images by making it focus on different areas.

In the end, considering the different XAI techniques performed on the final model, trained with augmentation and Median filtering, and summarized in Fig.9, we can say that the model is able to correctly focus on the lungs to perform the prediction.

It is important to note that these techniques are not mutually exclusive, but should all be taken into account to better understand the model behavior. For this reason, we provide them all.

E. Uncertainty measure

When calculating Jain's Fairness Index on SoftMax output we noticed that values were all around 0.33 (the lower limit of such index) in most cases. At first this might mean that the model was confident about own predictions. The problem was that this happened for wrong predictions too, and this could be an issue.

It is, indeed, preferable to have a model that is unsure of its predictions when they are wrong and therefore associates with them a high Jain's index value. In this way clinicians, for example, can decide not to consider those predictions.

When using Monte Carlo Dropout (MCD) we immediately noticed that some of the predictions were not so robust and reliable as we thought before.

In Fig.10a we can see the effect of applying Monte Carlo Dropout when the model performs a correct classification while in Fig.10b when an image is misclassified.

It is good to notice that the model stays confident of its own predictions when the prediction is correct while this is not true when the prediction is wrong.

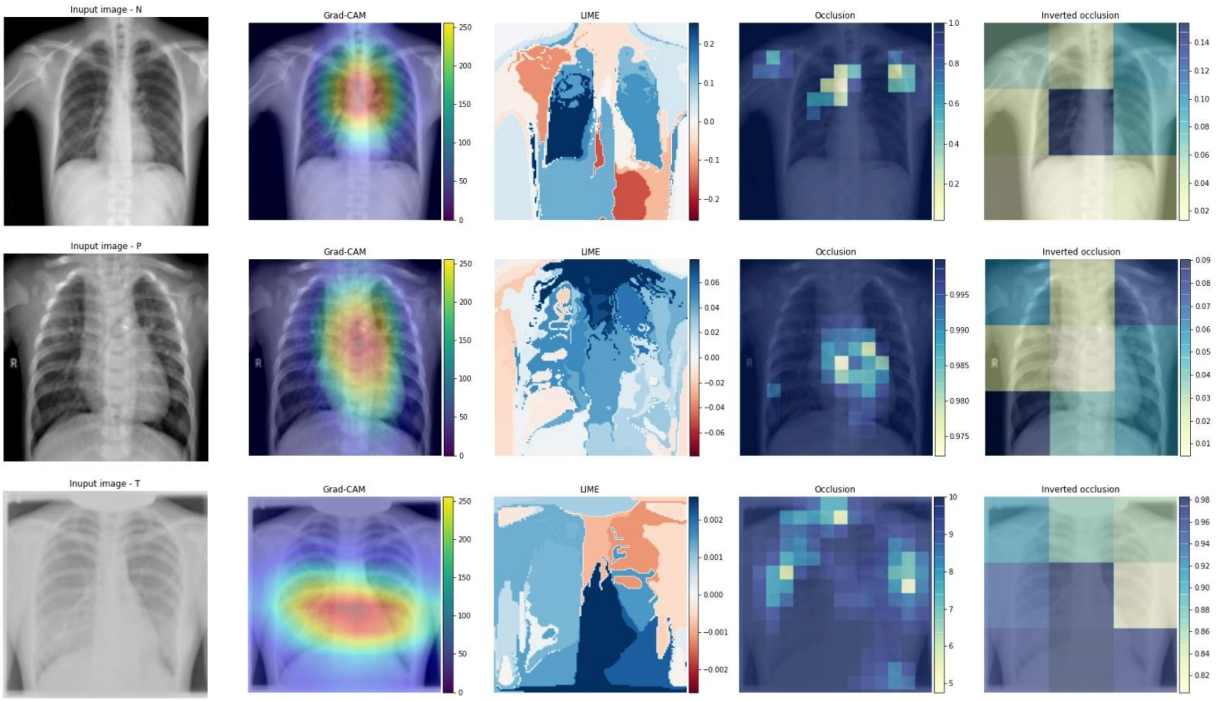

Fig.9 From the left to the right: Input Image - Grad-CAM - LIME - Occlusion - Inverted Occlusion for 3 images belonging to different classes. Images can be interpreted as follows: In Grad-CAMs the most relevant areas are the ones highlighted in yellow and red. In LIME blue areas are the ones that positively contribute to the prediction of the target class while the red are the ones that negatively contribute. In Occlusion white dots show areas that are required to perform a good prediction. In Inverted occlusion instead, areas upon which the model can correctly predict the target class are shown in dark blue.

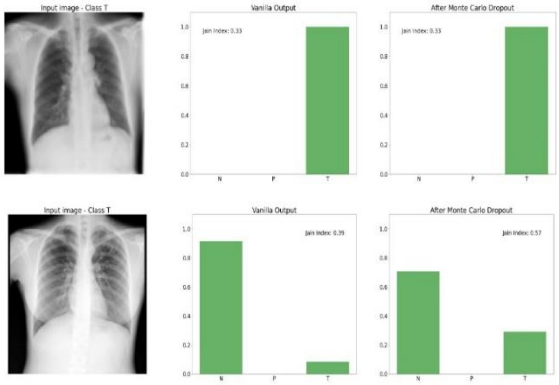

Fig.10 a) SoftMax output before (left barplot) and after MCD on a correct classification; b) SoftMax output before and after MCD on a wrong classification.

In the first case after MCD the fairness index has a constant value while in the second one we can observe an increase of J of 0.18, stating more uncertainty behind that prediction.

IV. Future developments

One of the main difficulties we faced was the presence of noisy images. Even if the techniques we exploited were able to yield improvements in the results, the issue was only partially solved. So, improving denoising by exploiting more effective techniques could be beneficial.

In a real healthcare application, our goal is to have the best performing model since the stakes are high when we are dealing with life of the patients. So, a naïve approach could be to make the model detect the images which are corrupted ($\sigma > 1$) and report them to the medical staff to repeat the X-ray if possible.

Moreover, another possible promising approach is to use Bayesian Networks to exploit fully probabilistic models' ability in quantifying model uncertainty.

V. Conclusion

In this project we have been able to explore both different models' architectures and denoising techniques as well as multiple XAI approaches. Since results, in terms of accuracy and f1-score, were good in most of the cases we heavily focused on explainability.

While doing this we understood which changes in terms of model training and denoising techniques turned out to be most beneficial for the model, allowing it to focus on meaningful areas of the image.

We chose to use multiple XAI techniques so as to provide multiple point of view from which it is possible to interpret the model.

In the end this we also tried to assess whether the model was robust and confident about its own predictions or not. By doing so we have been able to understand that our model was not so confident when it wrongly predicts an input image, as we expect it to be.

VI. References

[1] J. Immerkær, «Fast Noise Variance Estimation», *Computer Vision and Image Understanding*, vol. 64, fasc. 2, pp. 300–302, set. 1996, doi: 10.1006/cviu.1996.0060.

[2] M. Lin, Q. Chen, e S. Yan, «Network In Network». arXiv, 4 marzo 2014. doi: 10.48550/arXiv.1312.4400.

[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, e D. Batra, «Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization», *arXiv.org*, 7 ottobre 2016. https://arxiv.org/abs/1610.02391v4 (consultato 13 febbraio 2023).

[4] M. T. Ribeiro, S. Singh, e C. Guestrin, «"Why Should I Trust You?": Explaining the Predictions of Any Classifier», *arXiv.org*, 16 febbraio 2016. https://arxiv.org/abs/1602.04938v3 (consultato 13 febbraio 2023).

[5] R. Jain, D. Chiu, e W. Hawe, «A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems», *arXiv.org*, 24 settembre 1998. https://arxiv.org/abs/cs/9809099v1 (consultato 13 febbraio 2023).