Applied AI in biomedicine: X-Ray images classification

Francisco Dario Sanchez
DEIB
Politecnico di Milano
Milan, Italy
franciscodario.sanchez@mail.polimi.it

Ali Shadman Yazdi
DEIB
Politecnico di Milano
Milan, Italy
ali.shadman@mail.polimi.it

Siavash Taleghani
DEIB
Politecnico di Milano
Milan, Italy
siavash.taleghani@mail.polimi.it

Manon Alexandra Vignier
DEIB
Politecnico di Milano
Milan, Italy
manonalexandra.vignier@mail.polimi.it

***Abstract*—The scope for this project was to design and train a multiclass classifier for rib cage radiographic images able to classify image into Normal, Pneumonia or Tuberculosis. Steps involved data exploration, pre-processing, train and the test of different models. This project showcases a classification task carried out by radiologists and pneumologists approached with deep learning methods field and highlights model explainabity within said task.**

***Keywords*—deep learning, image processing, chest X-Ray images, pulmonary; tuberculosis; pneumonia**

I. Introduction

According to the World Health Organization (WHO), pneumonia and tuberculosis are two of the world's leading causes of death and diseases. In 2019, pneumonia was estimated to be responsible for approximately 11% of all global deaths and was considered the top cause of death among infectious diseases. On an annual basis, over 9 million new cases of pneumonia and over 10 million new cases of tuberculosis are reported. Pneumonia and tuberculosis are two diseases that can be diagnosed with thorax radiographies. Tuberculosis is caused by a bacterial infection, namely the mycobacterium species. This bacterium affects mainly lungs but also the whole body after multiplying. From X-ray images it is possible to highlight abnormalities that are consistent with tuberculosis. Such as opaque clusters typically found on the lung's lower lobe. However, these clusters can be easily confused by alterations due to other pulmonary pathologies, introducing a significant intra and interobserver variation.

Pneumonia is also an infection of the lungs caused by viruses or bacteria which inflames the sacs of the lungs called alveoli [1]. They get filled with fluid or pus so that it makes it hard for gas exchange to take place. Consequently, X-rays are suitable for detecting it as it will induce difference of density and will attenuate the X-rays – and so the image intensity – passing through the lungs.
Both diseases can be hard to distinguish, thus an AI support decision could be useful for the practitioner.

II. Material and Methods

A. Data exploration

1) Target distribution

A dataset with 15470 annotated chest X-ray images was used to develop a multiclass classifier. The images were labelled: "N" for healthy subjects, "P" for pneumonia cases, and "T" for tuberculosis cases. The distribution of these classes was analysed, as shown in *Fig1*. Most of the images corresponded to healthy subjects, making up more than two-thirds of the data. Pneumonia cases were less frequent, appearing at approximately half the frequency of healthy subjects. Tuberculosis cases were the least common, appearing at only a quarter of the frequency of healthy subjects.

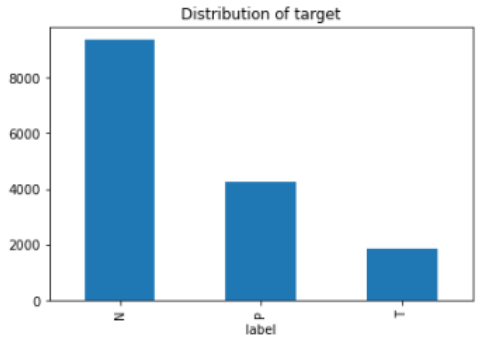


Fig.1. Class distribution of the data set.

2) Stratified splitting

With a closer examination to the dataset, some radiographies belonged to the same patient, which could potentially skew the results of the classifier. To mitigate this, a stratified splitting was performed to the dataset while images from the same patients were placed in the same subset. Thus, 75% of the images were designated for training, 15% for testing and 10% for validation while ensuring class proportion.

Despite the stratified splitting, the dataset remains unbalanced, with a greater number of images of healthy subjects. To counteract this discrepancy, class weights were computed to give greater importance to classes with fewer samples and are reported in Fig 2.

	Class N	Class T	Class P
Weights	0.5527	1.2079	2.7668

Fig.2. Class weights computed and applied during training.

3) Data augmentation

Due to the nature of the images the algorithm would likely focus on the anatomical differences between normal and pathological states, relying on pixel intensity distribution directly linked to tissue density to effectively discern among healthy and non-healthy patients. Thus, it was considered that adding augmented data would create examples that may not correlated to real images.

4) Data quality

After a visual inspection, it was observed that the images can be discerned. Examples are reported in Fig 3. the output class can be determined by looking up for specific morphological indicators, the baseline on a healthy patient lung will be dark, filled with air and branching structures can be observed close to the heart sprouting from the heart (not actual anatomy, but give the ionization nature of the images it's hard to discern among different tissues one behind another), heart will be slightly pointing towards the left. For pneumonia an important liquid quantity accumulates within the lungs, making the lungs appear denser (pixel intensity) and finally tuberculosis appears as opaque nodules within the lungs usually placed on the inferior lung lobe, close to the heart.

Fig.3. Example of data provided for class N (on the left), class P (in the middle), class T (on the right)

Furthermore, some images were found to be clear and of good quality, see Fig 3, while others were heavily noise contaminated or missed chunks. Salt and pepper noise was observed. Some images, even though they might be clear, may have implants, catheters and or lead markers. To address this, pre-processing was performed with caution to preserve crucial information, particularly when dealing with non-noisy data. Selecting an appropriate image processing pipeline is critical for ensuring data integrity.

It was observed that some images are non-standard, white pixels for air and, backgrounds, and dark pixels for dense structures, like bones. It was decided not to address this situation since deep learning models tend to learn complex nonlinear functions fed on image variations rather than just magnitude, image's main source of variability is spatial representation given by transitions on pixel intensities. Several images were zoomed in, mainly centering the rib cage, while others were occluded. Refer to Fig.4.

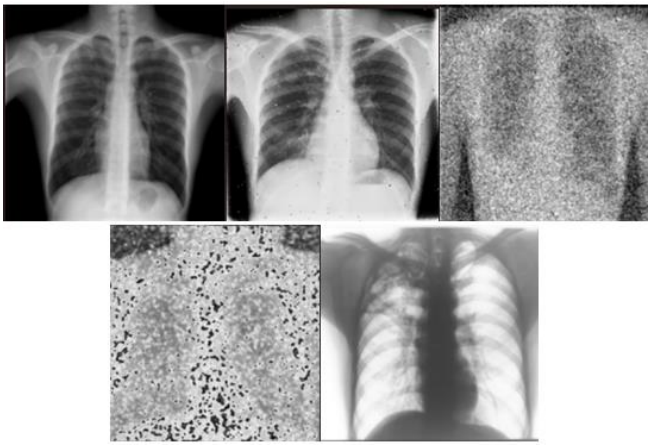


Fig.4. Investigation on data quality with clear image (top right), salt and paper noise (top middle), random noise (top left), damage and zoomed image (bottom left) and negative image (bottom right).

B. Pre-processing

1) Auto-encoder

Having a fair size dataset means that there is a lot of information at hand, to extract a lot of information may be seen as redundant so implementing an autoencoder who learns how to reduce the information content, such as omitting noise could be implemented. GE architecture takes an input image projects it on a higher dimensional representation and up sample it back to the previous dimension. This approach learns how to reconstruct the images by compressing it and then working a lower information content point. After observing its performance images lost details and sharp transitions, they showed significant morphological deformations. As it can be appreciated on an image noise was removed and an image with no noise present. Additionally, it has been determined that the decoder presents a significant computational burden,

thus it has been decided to not allocate further resources towards attempting to improve its result.

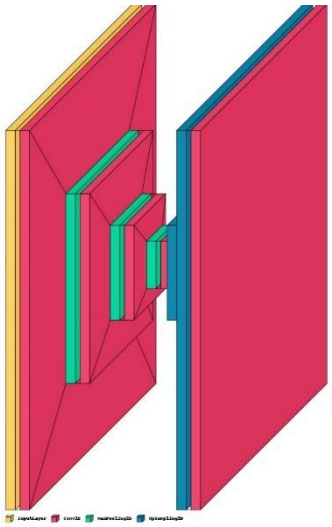


Fig.5. Architecture of the autoencoder

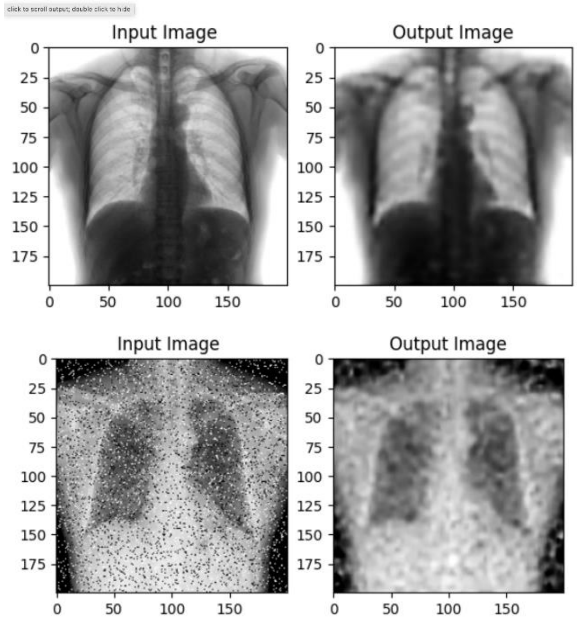


Fig.6. Example of images obtained after processing through the autoencoder.

The classical autoencoder architecture can be observed on Fig5, a dimensional reduction with a geometric grow on filters and a mirrored architecture to come back at the original dimensions. Fig 6 presents two images that are representative of the effect the autoencoder has on the dataset. For the first it's evident a blurry effect took place, and some morphological changes are to be clearly observed. While on the second one added noise leaves traces.

2) Filtering

To enhance dataset quality, it was opted to use a median filter with a 3x3 kernel followed by a Gaussian filter with a standard deviation of 1. The median filter is used to deal with salt and pepper noise, it preserves edges in images and is not sensitive to outliers, thus do not distort image contents. The Gaussian filter, on the other hand, was selected to produce a smooth output and reduce other types of noise while smoothing edges of the image. Finally, the images were resized to 200x200 to reduce the computational burden, as they were originally larger. Each image was processed, and time required for model training was reduced. Examples are presented in Fig7. For few images the quality was unchanged except a slight smoothing effect, as for the images on the first row in Fig 7.

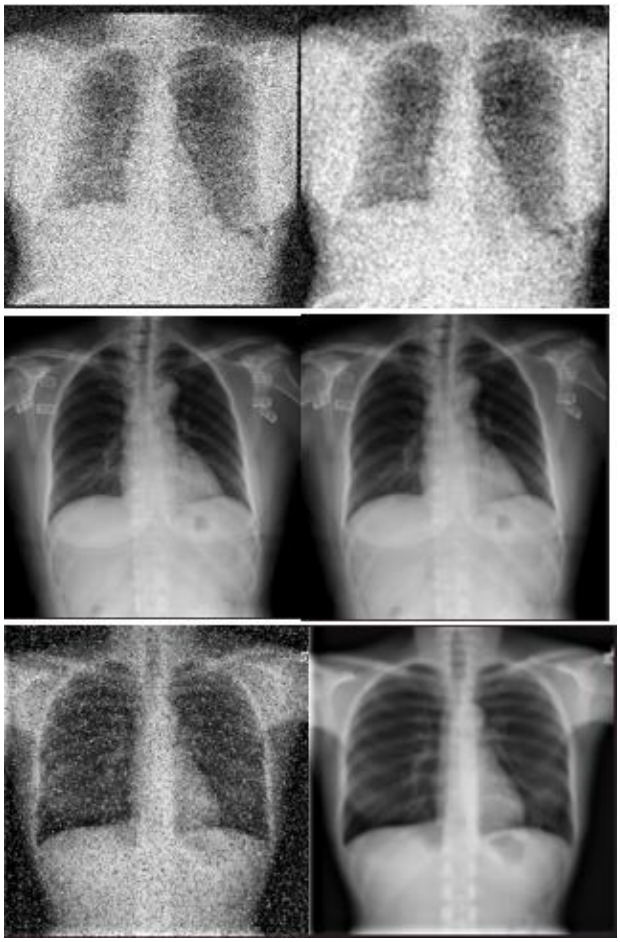


Fig.7. Example of images obtained after filtering, on the right the original images and on the left the filtered one.

3) Variance analysis

Filtering and preprocessing data is a common practice to reduce dataset complexity to a datum level. This might lead to different results depending on each image's own nature, given the preprocessing pipeline that's been proposed. Since

the data that was made available is over 15000 samples a variance spatial distribution over the preset split datasets was performed for both non-filtered and filtered images. The subset considered are: test, validation and training. One subset for filtered and non-filtered images. Giving a total of 6 subsets. The variance over all images present on each were used to calculate within its respective subset and heatmaps are presented in Fig.8.

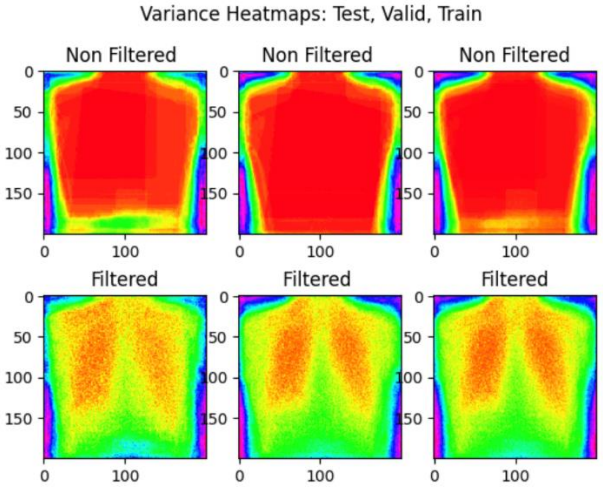


Fig.8. Variance heatmaps on test (left), validation (middle) and train (right) set before (top) and after (bottom) filtering.

Note that the upper maps present great variability on the rib cage and diaphragm section, indicating that it is the region of interest for this given task. Even if heat distribution indicates there's great information contained, a learning algorithm will likely focus on that whole region extension with a similar importance. After feeding the first dataset to the processing pipeline the variance over each image for given split was calculated. It is evident that hot point areas are placed within the lungs internal structures can be observed now, just after degrading the data.

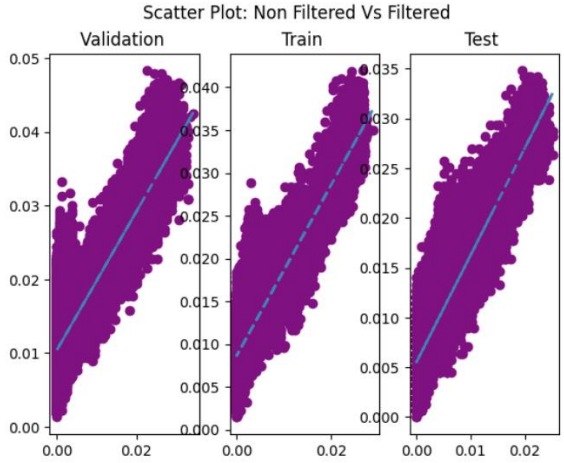


Fig.9. Scatterplot of validation, train, and test data: non processed data in function of processed data

The pair of heatmaps, in Fig 9., corresponding for each split were flattened and plotted, one against the other to visualize how they relate. A linear relationship can be observed, with some outlying points. Supporting this visual inspection, the correlation coefficient for each split pair was computed and overall, it was observed in Fig10. at least an 85% correlation for each data set. Which was deemed acceptable. It indicates at least 86% of the variance contained on the original dataset was preserved.

Data Split	Correlation Coefficient
Train	0.914
Validation	0.867
Test	0.900

Fig.10. Correlation coefficient of filtered and non-filtered images

C. Models

1) KNN

KNN is not usually a go to method for image classification, it's an unsupervised learning method, when it comes to high dimensionality problems, such as the feature extraction phase for image classification tasks. However, this type of models is useful for both a classification and explanatory purposes. It creates clusters and fit them to data distribution.

It is suitable for tracing nonlinear boundaries, if the data is distributed in a favourable way. No prior assumption is made about the distribution data, and it is fast to perform real-time classification, so we thought it might be worth trying it. KNN was performed with the dual intention of data exploration and classification, images are easily distinguishable in practice by a trained specialist's eyes, KNN on this application could be pertinent.

Layer type	Quantity
Input	1
Conv2d	9
Batch Normalization	9
Max Pooling2d	4
Flatten	1
Gap	1
Concatenate	1
Dense	3
Dropout	1
Output	1

Fig.11. Layers on homemade CNN

2) Homemade CNN

A convolutional neural network was implemented from scratch to extract features from data whose output was fed to a classifier 3 layer fully connected classifier, to provide a model that offered better explain ability. The architecture consists of the layers presented on Fig11.

A 3-layer classifier with a Convolutional Neural Network for feature extraction was proposed as represented in Fig11. The classifier having 7,864,929 parameters while the CNN has 559,872. This means that around 7% of the parameters are used to extract features while 93% for classifying images. Images are fed to the CNN part of the network, and the highest number of filters used was 256; while the last conv layer outputs a tensor of the form (12,12,16) that is subsequently flattened and concatenated with the average value from each of the filters on the 256 filters layer. Said vector is fed to a classifier composed by 3 fully connected layers, 2048, 1024 and 512 units. Throughout the feature extraction part of the network batch normalization layers were used to avoid vanishing gradient. While the idea of the concatenation of features plus average filters came to further expand the data been fed to the classifier without adding computational burden. ReLU functions were used for all layers but on the output, there it was opted to use SoftMax.

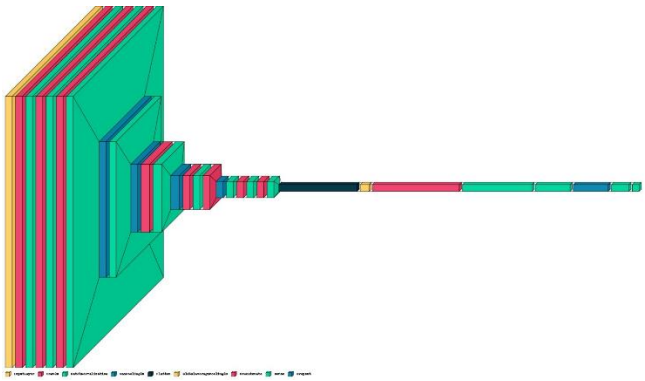


Fig.11. Architecture of the final CNN with convolutional layers (pink), dense layers(green), average pooling layers (yellow), dropout layer (second blue), and flatten layer (dark blue)

3) VGG

The Visual Geometry Group (VGG) is a network that can extract important geometric features and has been shown to be effective in processing X-ray images [1]. There are two versions of the VGG: VGG16 and VGG19, which have 16 and 19 layers respectively. The VGG16 has five convolutional blocks and a fully connected network, as depicted in Fig12. It includes a non-linear activation function called the ReLU, which provides non-linearity. The VGG19 has a similar architecture, including 5 blocks but having 19 layers of convolution instead of 16.

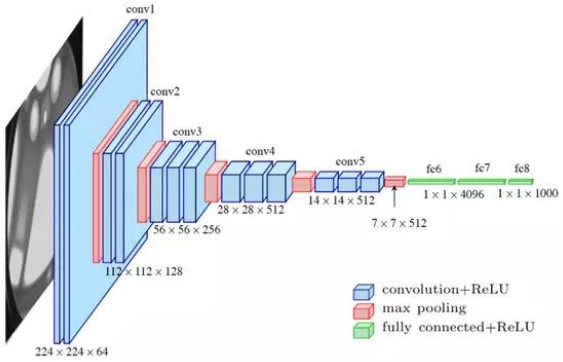


Fig.12. VGG16 architecture for an image input of default size 224x224

VGG network from the Keras library was applied on the dataset to enhance the accuracy of the classifier. The VGG network, which had already been trained on the ImageNet dataset, was integrated with a two-layer classifier. The output of the VGG network was processed through a flattening operation, which was then fed into a 512-unit dense layer, equipped with a Dropout regularization technique to prevent overfitting. Finally, the output of the dense layer was inputted into a 3-unit dense layer with a SoftMax activation function, as illustrated in Fig.13. Furthermore, fine-tuning was performed to optimize the feature extraction to fit out dataset in a more efficient way.

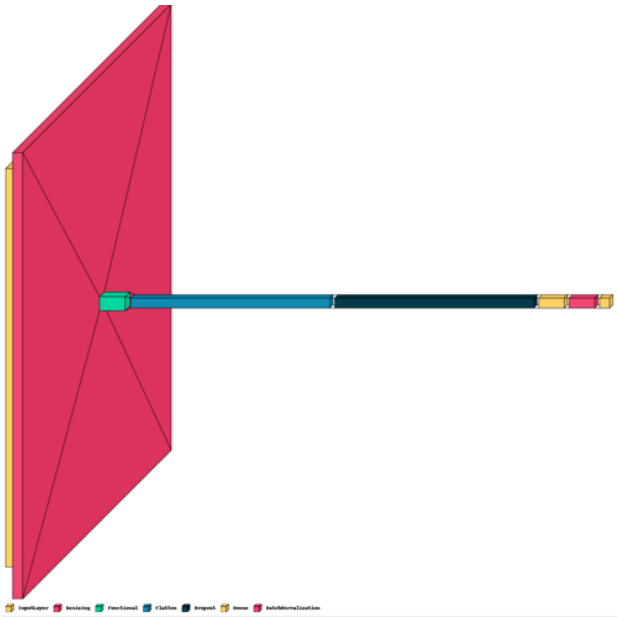


Fig.13. Architecture of the VGG transfer learning and fine tuning: input layer (yellow), VGG network (pink), flatten layer (blue), dropout layer (dark blue), dense layer (yellow) and batch normalization (pink).

The performance result of transfer learning is documented in the results section. Despite initial low performance and overfitting issues, it was employed as a preliminary step prior to the fine-tuning stage. Both VGG16 and VGG19 were

experimented with, and various layers unfrozen and fine-tuned pretrained classifier. Meaningful trials are reported in the results section. The optimal models were then combined to take advantage of their generalization capability and to minimize any potential weaknesses. The training procedure was executed on the training set, with continuous monitoring of overfitting on the test set. The training process was stopped when the loss function for unseen data started to decrease for several epochs. To optimize performance, all input images were resized to the standard VGG size of 224x224, resulting in a 3% improvement in accuracy score.

D. Explainability tools

Artificial intelligence models have garnered significant attention due to their ability to provide powerful support for decision-making processes. Due to the learning algorithm beign heuristic in nature, they lack trasparency, and their behavior is often perceived as a black box, making it difficult to understand the learning process that takes place. Despite they are powerful tools to support decision-making, but they can not be concoluded to be completely reliable. Explainability tools are important not only for entrusting and investigating the models behavious, but as well as providing crucial information to practitioners that can lead to the generation of new knowledge or even opening of new research possibilities. In this section, we will explore various methods for promoting explainability in AI and discuss their applications on the models we have trained. The results of these applications will be presented and analyzed in the results section.

1) LIME

LIME, the Local Interpretable Model-agnostic Explanation [3], is a tool designed to provide insights on the behaviour of a classifiers. It presents three main key features. It is model independent, interpretable, and local as it approximates local linear behaviour of a model. When applied to image classification, LIME takes as input a single image and generates a series of similar samples by turning on and off different pixels of the image. It then predicts the class of each artificial datapoint and calculates the weights of each artificial data point to measure its importance in order to fit a linear classifier to explain the most important feature.

2) Grad-CAM

Grad-CAM[4] is a gradient based explanation method for deep learning used especially for convolutional neural networks to make them more transparent and explainable. Grad-CAM stands for gradient-weighted class activation mapping. It uses gradient flowing from the final convolutional layer in order to visually validate where the network is looking, verifying it is indeed looking at the correct patterns. This information in retrieved as a heatmap.

III. RESULTS

A. General results

To evaluate the performance of the various approaches, the F1 score was chosen instead of accuracy as the evaluation metric. The F1 score considers both precision and recall, making it a more appropriate choice for multi-class image classification problems with imbalanced classes. Besides, it can be misleading in such scenarios, as it does not account for the proportion of false negatives and false positives.

All models were evaluated using the same validation set.

Model	Accuracy /F1 score (micro average)	F1-score		
		Class N	Class P	Class T
KNN	0.859	0.88	0.90	0.47
CNN First Version	0.909	0.93	0.94	0.68
CNN Second version	0.924	0.94	0.96	0.73
VGG16 transfer learning	0.966	0.97	0.98	0.90
VGG16 fine tuning with 6 layers	0.972	0.98	0.98	0.91
VGG16 fine tuning with all layers	0.969	0.97	0.99	0.89
VGG19 transfer learning	0.957	0.97	0.98	0.87
VGG19 fine tuning with 6 layers	0.975	0.98	0.99	0.92
VGG19 fine tuning with all layers	0.964	0.97	0.98	0.89

Fig.14. Results of the different models implemented.

From the results, we can assess a few models clearly do not fit for this task, such as KNN, while others seem to perform well. In order to discriminate them and evaluate them more efficiently, further analysis was carried out.

B. KNN

The KNN algorithm assesses a similarly within the dataset, when it comes to first and second class (Class1 being Normal, class2 being Pneumonia). However, a significant discrepancy is observed for the performance of the third class (tuberculosis) as indicated in the F1-score. This is due to an elevated number of false positive predictions, as

evidenced by the confusion matrix in Fig.15. But an unsupervised obtaining such results strongly indicates that the dataset is easy to classify, for class one and two, corresponding to Normal and Pneumonia. While a fair number of data is reported as false positives for class one, suggesting that among class 1 and 3(Class 3 being tuberculosis) is harder to split.

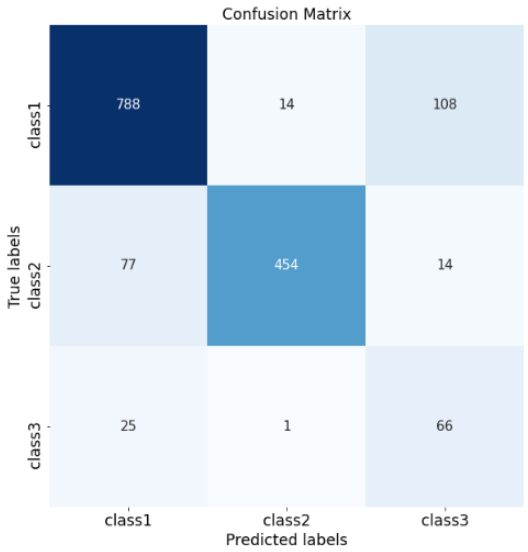
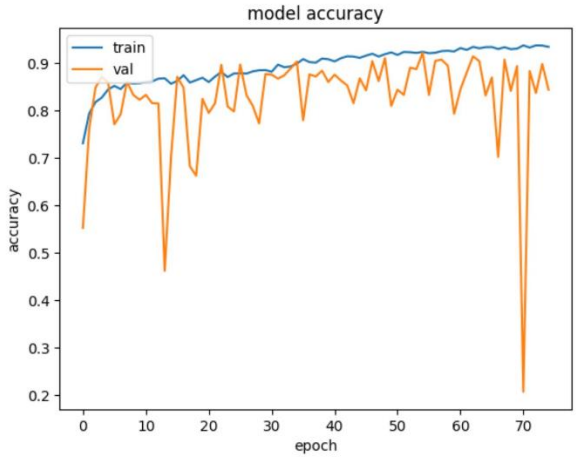


Fig.15. Confusion matrix of the prediction of the KNN (Class1 being Normal, class2 being Pneumonia, and class3 being Tuberculosis).

C. Homemade CNN

1) Metrics

In Fig16. is represented the behaviour of the CNN trained from scratch along the training, showing the accuracy and the loss function in function of the number of epochs for both training and validation dataset. Except from some oscillations in the accuracy plot, the model generalizes. In addition, it does not present overfitting before epoch 50. The best epoch is the one when both accuracies met.



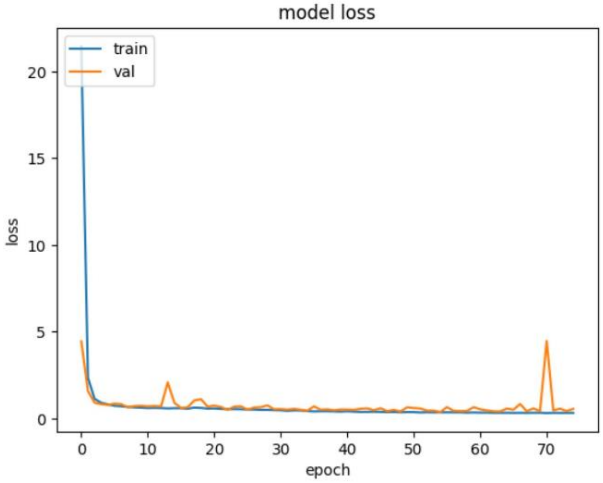


Fig.16. Evolution of accuracy and loss along the training of the CNN

Besides, considering the low F1-score associated with the third class, the precision and recall were also computed and reported in Fig17. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while Recall (also known as sensitivity) is the fraction of the total amount of relevant instances that were retrieved.

	Class N	Class P	Class T
Precision	0.94	0.96	0.67
Recall	0.93	0.95	0.80
F1 score	0.94	0.96	0.73

Fig17. Different metrics obtained with the CNN

The corresponding confusion matrix is shown in Fig18. Along with the previous metrics, it highlights the tendency of the model to predict false positives. It happens more frequently than false negatives, which, for medical purposes, is deemed acceptable rather than the other way around. Besides, this model prediction, if provided with some explanatory features, can help the practitioner to take a decision, allowing him to focus on some region of interest that influenced the model.

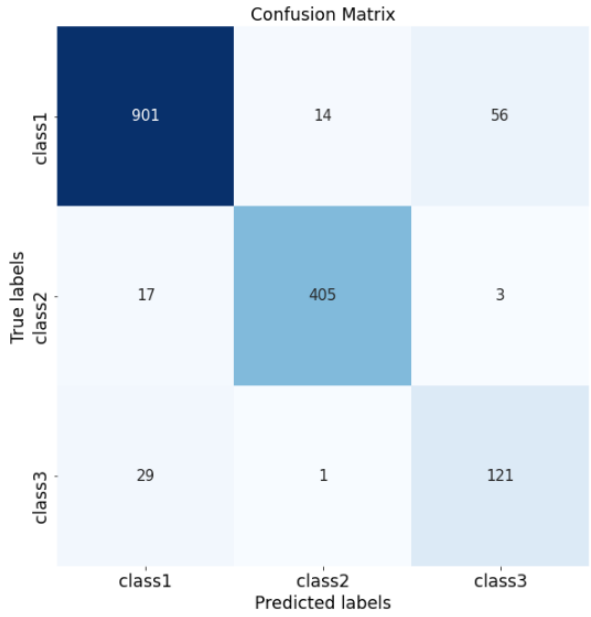


Fig.18. Confusion matrix for the CNN trained from scratch (Class1 being Normal, class2 being Pneumonia, and class3 being Tuberculosis)

2) Grad-CAM

Some samples from the testing set underwent a grad-CAM analysis, generating a heatmap for one of the last convolutional layers where location was still meaningful. Two results of each class are presented. In Fig.19. shows two healthy patient radiographies where the regions chosen by the network as important to classify were highlighted. From them, it was clear the model was extracting information from the bone structure and in general high-density organs like the heart, and not from the lungs themselves, the structures that surrounded the lungs and their physical orientations.

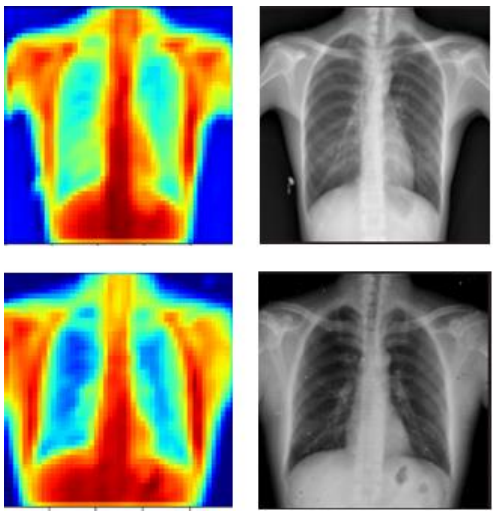


Fig.19. Heatmaps obtained through grad-CAM corresponding to healthy patients

Heatmaps of patients suffering from pneumonia in Fig.20. and patients suffering from tuberculosis in Fig.21. Showcases how the network differentiates healthy and non-healthy patients. For images of class P, lungs are in general filled with liquid, which implies a denser lung where less air is present. Lungs, its contents and surroundings, are deemed region of interest by the model. For images of class T, the difference is subtle sometimes to a normal patient. Portions of the lung present opaque nodules, predominantly on the inferior lobes, damage due to the presence of bacteria are focalized but the functional and structural integrity of the lungs are not compromised as the one observed on Pneumonia. The first case on fig 20, clearly shows a patient with enlarged heart (in red) taking a considerable space on the rig cage, while lungs (in yellow) are clearly detected.

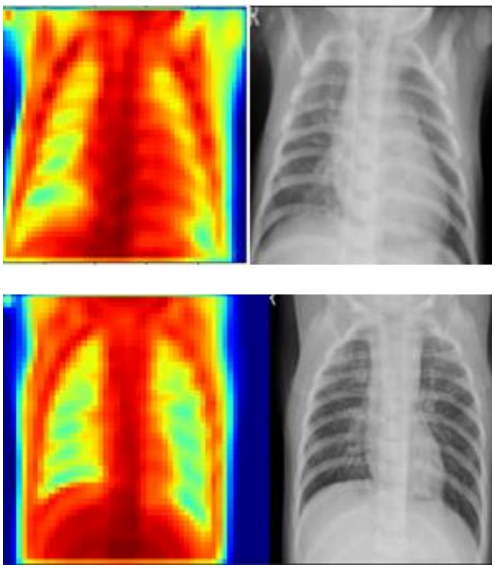


Fig.20. Heatmaps obtained through grad-CAM corresponding to patients suffering from pneumonia.

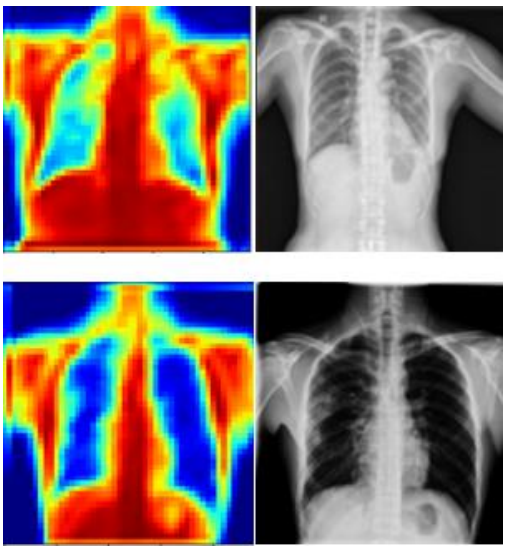


Fig.21. Heatmaps obtained through grad-CAM corresponding to patients suffering from tuberculosis.

The convolutional network is consequently implemented in a way it extracts irregularities from the tissues, with different densities, to determine if a disease is present or not. The type of non-linearity (presence of air or not and so difference of intensity) allows to determine which disease.
Moreover, it is possible to understand why sometimes the network has difficulties differentiating a healthy patient from a patient suffering from tuberculosis. Indeed, irregularities in this case are present mainly in areas visually close to the sternum and spine, but also shoulders. This can lead to confusion due to the presence of the heart near the spine which can present physiological variation which can be considered an abnormality detected by the network. These are considerations a practitioner should fully understand, allowing a full comprehension of the network decision even when he does not agree.

D. VGG

The accuracy and loss function behavior of the classifier training of the VGG architecture is shown in Fig.22. The results indicate overfitting, particularly in the loss function, and the training process was terminated at the optimal epoch (indicated with blue dashed line) to prevent further discrepancy between the training and validation performance. Despite this initial overfitting, further fine-tuning of the model's layers successfully addressed the issue.

1) Transfer learning

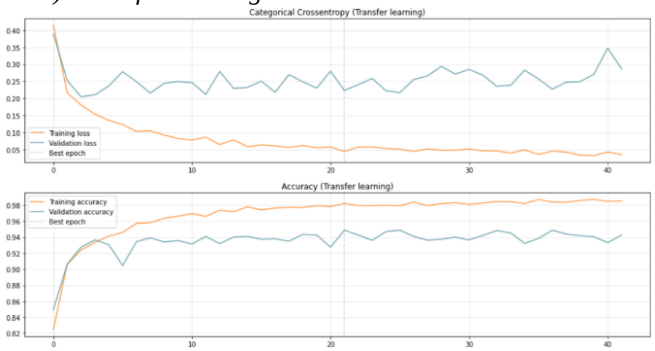


Fig.22 Evolution of accuracy and loss along the training of the transfer learning for VGG16

Despite the occurrence of overfitting in the training phase, the results of validation accuracy obtained were superior to those obtained from KNN and the CNN. As documented in Fig.14, the utilization of transfer learning with the VGG16 classifier demonstrated one of the highest accuracy rates in predicting class3 (tuberculosis), without undergoing any fine-tuning processes. The confusion matrix of the VGG16 transfer learning can be viewed in Fig.23. However, improvements can be achieved by fine tuning to achieve better results.

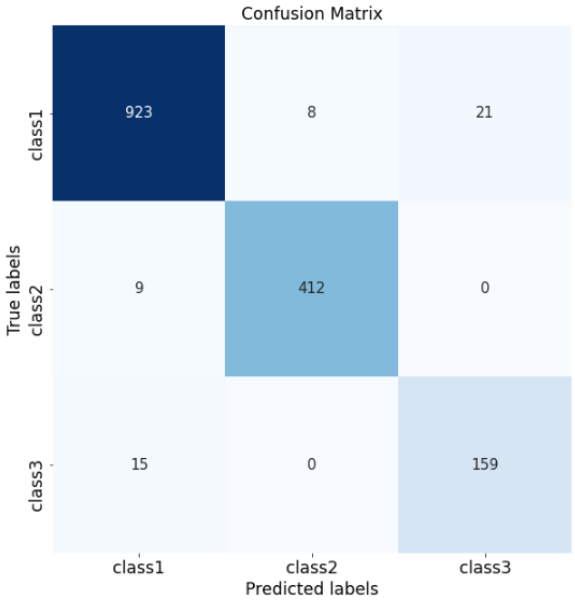


Fig.23 Confusion matrix VGG16 transfler learning (Class1 being Normal, class2 being Pneumonia, and class3 being Tuberculosis)

2) Fine-tuning of the VGG architucture

To optimize the performance of the VGG architectures fine-tuning was applied on selected layers. At first approach, fine-tuning was applied on the last six layers and then all the layers of the architecture. Fig.24 showcases the accuracy and loss function evolution during the fine-tuning process. It can be concluded that there is no evidence of overfitting, except for some peaks in the curve can be attributed to variations in the learning rate or batch size parameters.

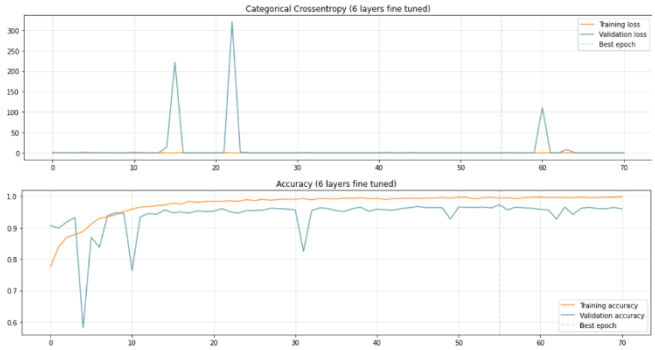


Fig.24 Evolution of accuracy and loss along fine-tuning six layer of the VGG architecture

The analysis of the results obtained varied based on the number of layers fine-tuned. The analysis on the confusion matrix revealed that when only six layers were fine-tuned, it led to a substantial improvement for class1 (Normal) and a slight improvement for class2 (Pneumonia), with no improvement observed for class3 (Tuberculosis). On the other hand, when all layers were fine-tuned, class3 experienced a minor improvement in performance, but class1

experienced a decline. The confusion matrices depicting the results of the VGG19 architecture under both six layer and all layer fine-tuning can be seen in Figures 25a and 25b, respectively. It is worth noting that the configuration with six-layer fine-tuning resulted in the best overall accuracy among the tested models.

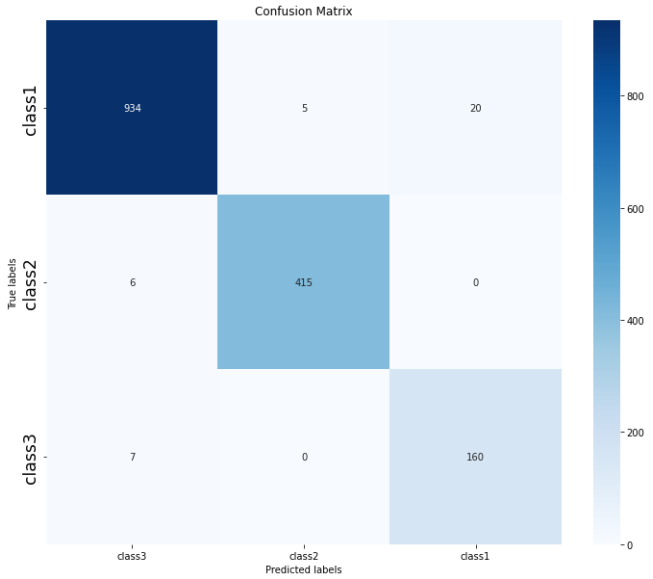


Fig.25a Confusion matrix VGG19 six layers fine-tuned (Class1 being Normal, class2 being Pneumonia, and class3 being Tuberculosis)

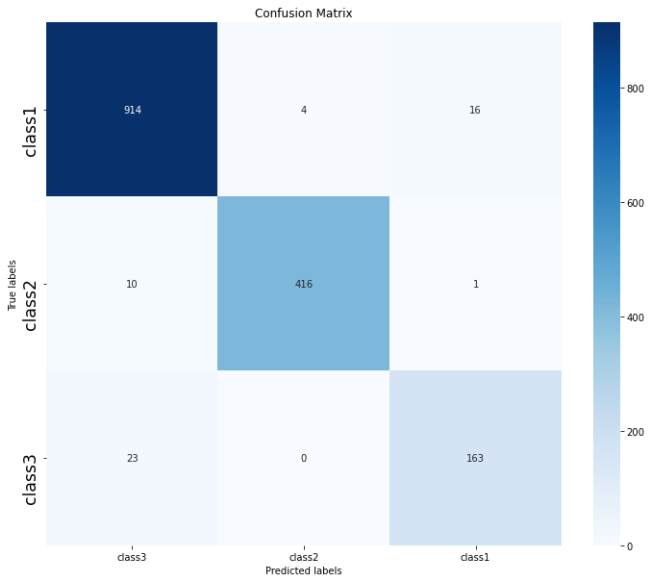


Fig.25b Confusion matrix VGG19 all layers fine-tuned (Class1 being Normal, class2 being Pneumonia, and class3 being Tuberculosis)

3) Lime

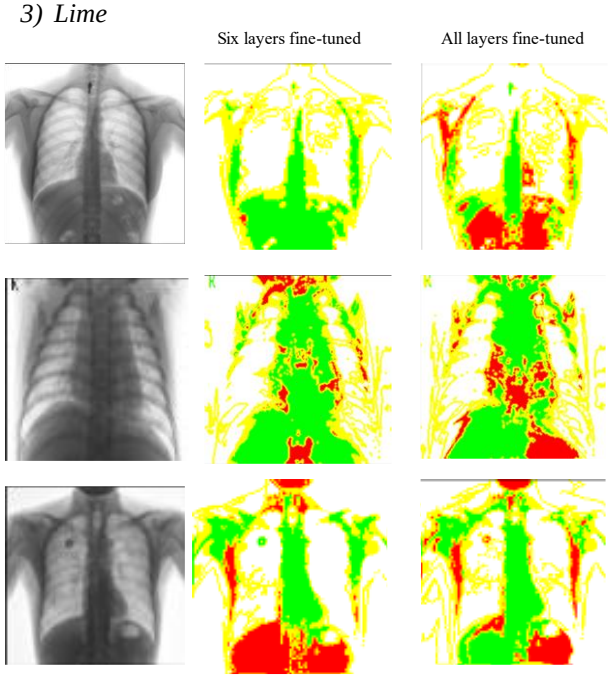


Fig.26 VGG19 six layer fine-tuned and VGG19 all Layers Lime maps

As presented above on Fig.26 Lime reports back a contour image with regions in green and red, green being the part on the image that was coherent with the class that it was predicted, while red means the parts from the image that root against the predicted class. It is a fuzzy yet visual representation on how the neural network operates. For the column in the middle the first two classes were better classified while the third one was not well generalized. On the third column the last image, for tuberculosis, was better generalized. Focusing on the last six layers fine-tuned model (second column), first class (Normal), it's clear that it considers the tissues around the lung as being on a normal state. Since almost no red is present, this suggests that the model bases its decision on the rib cage morphology. Whereas for the second-class (Pneumonia) is more of a complex case. The algorithm understands that the heart is enlarged, and the lungs reduced in size associating that to a patient with pneumonia. While for the third class(Tuberculosis) we can observe a foreign circular object at the lungs, the algorithm is sure it is part of the opaque clusters that characterize tuberculosis, which is not correct.

Passing to the model with all layers fine-tuned (third column), first class is fuzzier that it's counterpart, spine and heart superposition central column at the Rib Cage still is deemed as important for predicting a class one, but the diaphragm and the left heart ventricle are rooting against this class, showing a less desirable generalization for class one in contrast with the model that only six layers were fine-tuned. Same story on second class as for the first model, there is a clear detection of a heart enlargement but tissue on the spine heart

superposition is now prevalently rooting against it, also poorly generalizing. However, the third class for the model with all layers fine-tuned, detects the circle as something against the classification, which ideally was expected to be just ignored by the algorithm since it is likely a lead marker or an implant. Non the less a better generalization is observed, much more confident prediction rule. It's worth mentioning that on the last example both algorithms are ignoring a less dense mass on the diaphragm, suggesting it understand the that for given task the interaction of the pulmonary pleura and its surrounding tissues was important rather than the outer tissue in contact with it. However, this is just a weak observation given the fact that the diaphragm itself is fuzzy. Furthermore, it is possible to analyze and explain the underlying reasons for the superior performance of the model with six layer fine-tuned on class1 and the improved accuracy of class3 in the model with all layers fine-tuned.

IV. Conclusion

A. *Future improvements*

Using an autoencoder turned out to not be effective for data dimensionality reduction, better results were achieved by traditional image filtering. An autoencoder consists of an encoder and decoder part, said parts can be used as building blocks for creating a neural network that learned the data set and was able to generate auto validated randomly generated images. Generative Adversarial Network can be used for generating more examples. For the CNN proposed a further improvement would be to add extra dense layers and use more dropout layers for further normalization. Meanwhile for the data processing maybe running models with a bigger dimension to assess if better classification between normal state and tuberculosis is achieved. Additionally, ensemble models can be trained to average the VGG models to have a better result among all classes.

B. *Final Discussion*

Different models were analysed with different explainability tools, ultimately showcasing that those models follow a logic and rules on a common region of interest, the rib cage and more specially regions within the lungs that are not filled with air, as well as dense tissues surrounding the lungs, concluding that the presented models learned in a similar fashion. The pre-processing pipeline implemented, started with an end in mind, enhancing the ribcage region to facilitate learning. However, all models understood there's a pleural space and structures around the lungs interact with them by means of it. Given pathological conditions do shape pleura, and by extent those structures around. If a model must be proposed for explanability on a given task a CNN train from scratch may generate more coherent results, as the one obtained on this report at the cost of having a lower generalization.

References

[1] “Pneumonia,” *World Health Organization*. [Online]. Available: https://www.who.int/health-topics/pneumonia#tab=tab_1. [Accessed: 11-Feb-2023].

[2] ‘Pre-trained VGG-16 with CNN Architecture to classify X-Rays images into Normal or Pneumonia’, P.Naveen, B.Diwan, 2021, IEEE.

[3] https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/

[4] “’Why should I Trust You?’ Explaining the predictions of any classifier”, M.T.Ribeiro, S.Singh, C.Guestrin, 2016 (available here: https://arxiv.org/pdf/1602.04938.pdf)

[5] “Grad-cam: Visual Explanations from Deep Networks via Gradient-based Localization”, R.R.Selvaraju and al. 2019 (available here: https://arxiv.org/pdf/1610.02391.pdf)

[6] “Tuberculosis,” *World Health Organization*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/tuberculosis. [Accessed: 12-Feb-2023].