Analysis of COVID-19 Chest X-rays

Final Report

ALEXANDRA RANCIC, PHILIPP TRINH, PREETHA BALAKRISHNAN,
PAUL POURMOUSSAVI

1. Executive Summary

1.1. Brief overview of the project

The objective of our project is to enhance the diagnostic accuracy of X-ray image classification using machine learning and deep learning techniques. Accurate interpretation of X-ray images is crucial for timely and effective medical interventions. This project aims to build the models which will provide reliable predictions and interpretable results. We explored classical machine learning models, and compared their performance with deep learning methods. Additionally, we implemented Gradient-weighted Class Activation Mapping (Grad-CAM) for the Convolutional Neural Networks (CNN) to visualize which parts of the X-ray images influenced the model's decisions.

1.2. Key findings

- Model Performance:
    - Classical models like Random Forest. Bagging, and XGBoost achieved high accuracies, with XGBoost achieving up to 90%.
    - The CNN model outperformed the classical models with an accuracy of 95%, making it most reliable model for our task.
- Interpretability:
    - Saliency maps was successfully integrated with the CNN model, providing visual explanations of the model's predictions. This feature is crucial as it highlights the areas of the X-ray images that the model focused on when making a diagnosis.

1.3. Project implications

The deployment of a CNN model equipped with Saliency maps may have significant implications for the healthcare industry:

- Enhanced diagnostic accuracy: The high accuracy of the CNN model ensures more reliable diagnoses, reducing the chances of human error and improving patient outcomes.
- Improved efficiency: With rapid and accurate diagnoses, the model can streamline the diagnostic process, allowing for quicker medical decisions and treatment plans.
- Scalability: The model can be continuously updated and trained with new dta, ensuring its relevance and accuracy over time.

1.4. Recommendations

- Implement the CNN Model: Given its superior accuracy and interpretability, the CNN model with Saliency maps should be implemented for X-ray image classification in clinical environments. Also, we suggest showing Grad-CAM images, since they may support the model interpretability.
- Expand and refine dataset: Continuously expand the dataset with new X-ray images and refine the model to cover a broader range of diagnostic conditions.
- Regular updates: Regularly update and retrain the model with new data to maintain its performance.

2. Introduction

2.1. Background and context of the project

COVID-19 is a disease caused by the virus SARS-CoV-2, and it was discovered by the end of 2019 in Wuhan, China. This virus has spread fast around the globe, resulting in the global pandemic, which has changed our lives. The transmission of the virus occurs mainly via droplets, through coughing, sneezing, or even speaking with an infected person [1]. Initial studies have shown that the virus can survive on surfaces, which stimulates its transmission [2]. Aged people and those with chronic diseases, e.g., diabetes and cardiovascular diseases are at a higher risk of developing serious complications. The symptoms of COVID-19 vary from mild to severe. Fever, cough, shortness of breath and tiredness are considered mild symptoms. Nevertheless, in more serious cases, the virus can lead to various lung complications such as COVID-19 Acute Respiratory Distress Syndrome (CARDS) or/and pneumonia [3][4]. CARDS, a result of the fluid build-up in the alveoli in the lungs, causes difficulty in breathing and leads to severe hypoxia. Consequently, patients with CARDS require intensive medical care and even after their recovery the lung scars may stay permanently. This reduces respiratory function and further influences the quality of life [6]. Pneumonia, which manifests with changes in breathing patterns, breathlessness, chest pain, and hypoxia, in most cases also requires hospitalization and additional oxygen therapy or mechanical ventilation in intensive care units [5]. Studies have shown that along with acute lung problems, COVID-19 can cause long-term respiratory issues known as post-COVID syndrome, leaving permanent changes in the lung tissue [7].

Current methods employed to detect COVID-19 include real-time Polymerase Chain Reaction (RT-PCR), fast antigen tests, serological tests, genome sequencing, computer tomography

(CT), and radiography (X-rays) of the lungs [8]. X-rays of the lungs were indispensable during the pandemic since this method is fast, available in hospitals, and provides valuable information on patients' conditions. X-rays are of great importance for monitoring the severity of the disease and its progress. This method is used to diagnose characteristics and signs of pneumonia and gives information about lung tissue damage [9].

We live in the era of Artificial intelligence (AI) and Machine Learning (ML), and the application of these technologies has already generated revolutionary changes in medicine. One of the fields in which AI may have a strong influence is disease diagnostics - especially for the detection of COVID-19 using radiography. Diagnosis of lung diseases via X-rays largely relies on the experience of the radiologist and may be subjective depending on the workload of the medical staff and their available time for analyses. Thus, AI and ML give the possibility of automated, fast, and precise analysis of X-rays while, at the same time, increasing the accuracy of diagnosis and decreasing the time needed for decision [10]. This project focusses on building a deep learning model that recognizes characteristic lung patterns (such as pneumonia and lung opacity) of COVID-19 patients. Once trained, we hope that it can automatically analyses new X-rays with high accuracy, helping medical staff to identify infected patients faster and with more efficacy. Consequently, this association between AI and X-rays of the lungs to detect COVID-19 improves diagnosis and reduces the workload of healthcare workers. Early detection of COVD-19 is essential as it may reduce the burden on the healthcare system, burden of the disease, and may even help to ameliorate the life-threatening respiratory complications.

2.2. Problems and opportunity

2.2.1. Problems

The COVID-19 pandemic has posed several significant challenges to healthcare systems worldwide:

- Overburdened Healthcare Systems: The rapid spread of COVID-19 led to and unprecedented number of hospitalizations, straining healthcare resources and personnel.
- Diagnostic delays: Traditional diagnostic methods like RT-PCR, though accurate, can be time-consuming and resource-intensive, causing delays in diagnosis and treatment.
- Subjectivity in Radiographic Analysis: The interpretation of chest X-rays is heavily dependent on the expertise and experience of radiologists. Variability in interpretations can lead to inconsistent diagnoses, especially under high workload conditions.

- Limited Resources: In many regions, the availability of trained radiologists and diagnostic tools is limited, making timely and accurate diagnosis challenging.

2.2.2. Opportunities

The challenges presented by the COVID-19 pandemic have also highlighted significant opportunities to advance healthcare through innovative technologies and methodologies. Addressing these opportunities can not only improve the management of the current pandemic but also support the healthcare system's resilience against future outbreaks.

- The advent of AI and machine learning presents a transformative opportunity to enhance diagnostic accuracy and speed. These technologies can be trained to recognize complex patterns in chest X-rays that may be indicative of COVID-19, pneumonia, and other lung conditions.
- AI-driven diagnostic tools can be rapidly deployed and scaled to meet the demands of a pandemic. Unlike traditional methods that require extensive laboratory infrastructure, AI models can be implemented in various healthcare settings, from large hospitals to small clinics. Additionally, AI systems can continuously learn and improve from new data.
- Early detection of COVID-19 is crucial for effective treatment. AI-powered diagnostic tools can identify changes in lung patterns that might be missed by the human eye, enabling earlier intervention. Furthermore, these tools can be used for ongoing monitoring of patients, tracking the progression of the disease and the effectiveness of treatments.
- AI and machine learning can help optimize the allocation of medical resources. By providing accurate and rapid diagnostics, these technologies ensure that patients receive appropriate care without unnecessary delays. This optimization is particularly important in overburdened healthcare systems, where efficient resource management can save lives.
- The implementation of AI diagnostic tools offers an opportunity for training and education within the medical community.
- AI-driven diagnostics have the potential to bridge gaps in global health equity.

In summary, the integration of AI and machine learning in COVID-19 diagnostics represents a significant opportunity to revolutionize healthcare delivery. These technologies can enhance

diagnostic accuracy, speed, and consistency, while also addressing resource constraints and variability in human interpretation.

2.3. Objectives of the project

The main goal of this project is to develop a deep-learning model for the detection of COVID-19 based on the X-ray images of the lungs. Through this process, we aim to develop a robust tool, which will help in faster and more precise diagnosis of COVID-19. This will support medical experts to use this as proof (initially along with their medical expertise) to go ahead with treatment within a short span of time. Earlier diagnosis and treatment alleviate symptom severity providing ample time for the body to fight against the virus. Such a model reduces the burden on the healthcare system in many ways. When combined with telemedicine and virtual consulting, it could aid in reducing the number of hospitalizations and the number of resources utilized.

3. Methodology

X-ray image data required to train and test the model were obtained from Kaggle [11], a large machine learning and data science online platform that helps people to build their skills with various data-related challenges. It contains '.png' images of volunteers/patients grouped into four categories: normal (i.e. images from healthy volunteers), viral pneumonia, lung opacity, and COVID-19. Our analysis was divided into two main parts- data exploration, data pre-processing and model development.

3.1. Data exploration

To get an understanding about the data at hand, we studied raw images, masks and metadata at length. Aspects such as number of images, their format, image size, color/grey scale images, mean and standard deviation of pixel intensities as well as lung area were explored, and corresponding visualizations were created. This served as the starting point to work on our pre-processing strategy.

3.2. Data pre-processing

Image pre-processing is a crucial step that affects the performance of the models developed. It generally includes selecting specific regions of interest from the image, weighting data, resizing images, filtering the noise and normalizing pixel data. In our project we worked on classical classifications models and compared their performance with deep learning models.

To this end, we divided our pre-processing into two parts each specific for the type of models we develop in the following step.

3.2.1. Image pre-processing for Classification models

3.2.1.1. Class weights

During the data exploration phase, we observed that the data we obtained was not equally weighted for each group/class. Training a model on an imbalanced dataset is surely possible. However, the learning becomes biased towards the majority classes. We tried to balance our data using standard undersampling and oversampling techniques. Undersampling, reduced the number of samples per class and the accuracy of the model was not that great (data not shown in report but present in notebook). Oversampling on the other hand had a high run time. Therefore, we chose to use class weights to balance our data. Statistical or class weighting assigns different weights to the classes in the dataset. These weights influence the loss function during training, giving higher importance to minority classes. In our project we used class weights for classification and DL models.

3.2.1.2. Resizing images

For classification models all raw X-ray images or their corresponding region of interests(roi) that focusses only on the lung region were 256 x 256 pixels in size. The X-ray images originally were 299 x 299 pixels in size. In such cases, the images were resized to the target size (256 x 256).

3.2.1.3. Selecting the lung area as the region of interest (ROI)

The Corona virus primarily resides in the lungs and leads to complications in this region. For this reason, we decided to test images with just the lung as the roi along with whole X-ray images. To obtain this, we used resized X-ray images and corresponding masks of the same size and extracted just the lung area of the X-ray of each image. This procedure was followed for every image from each category (Figure 3.1, shows a sample image from each category). This roi in grayscale was used in the subsequent pre-processing steps.

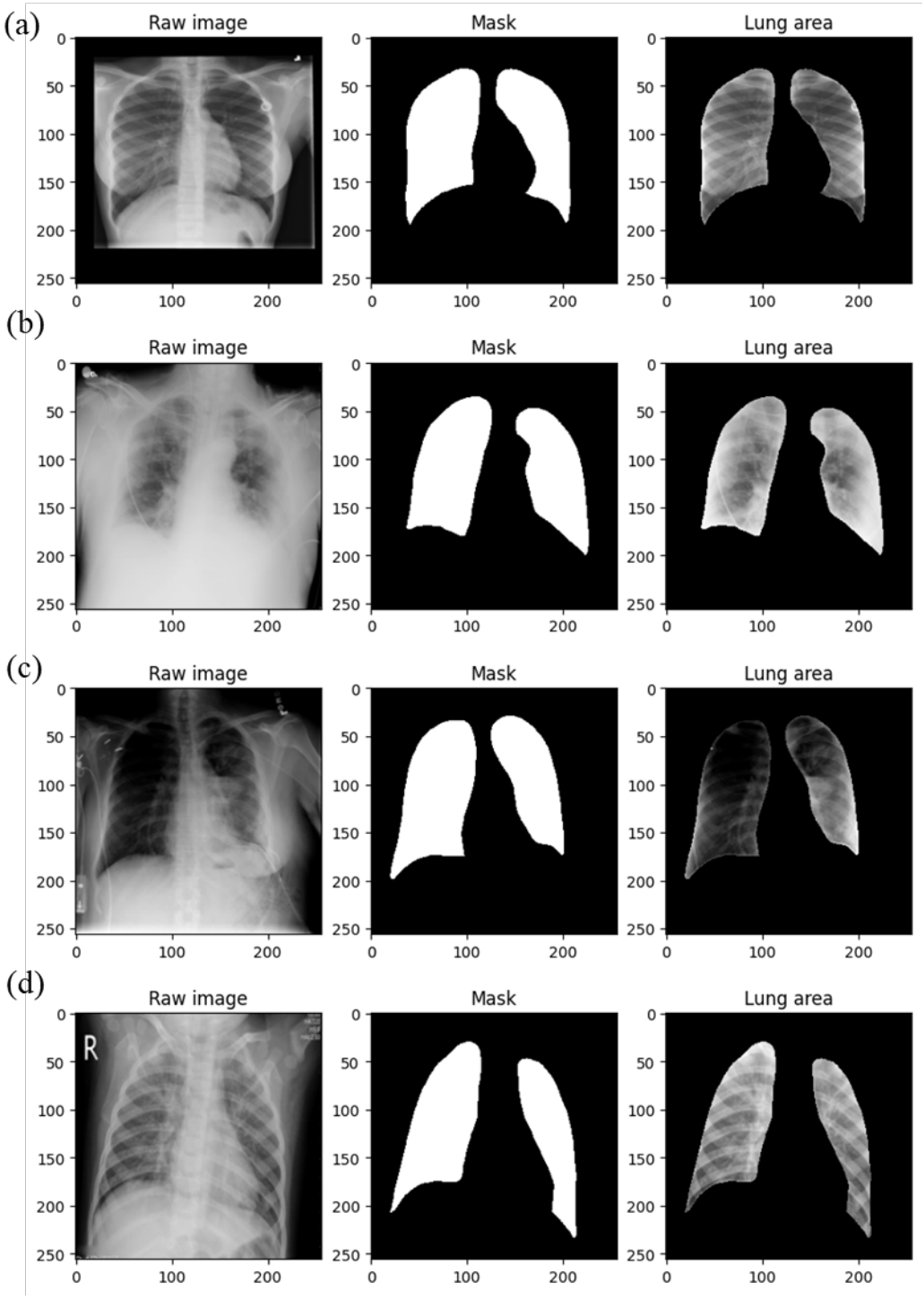


Figure 3.1: Selecting the lung region of the chest X-ray (a) Representative X-ray of a normal volunteer (b) X-ray image of a COVID-19 positive patient (c) X-ray image of a patient with an opaque lung (d) X-ray image of a patient with pneumonia. The scale signifies the width and height of the image in pixels.

3.2.1.4. Image filtering

Raw image data has ample noise that could unnecessarily delay the ML learning time that in future could be costly for hospitals and diagnostic centers. It could also decrease the precision and accuracy of the model in question. To reduce noise, we tested different filtering techniques in the OpenCv library such as median blur, Gaussian blur, erosion, Laplacian filter along with Canny edge detection. After extensively studying all filters (data presented in the pre-processing and modeling report), we decided to consider only one filter for modelling and analysis. Best results and a good variation between normal and COVID images were observed with Gaussian smoothening coupled with Canny edge detection. It has also been a technique that is widely used for smoothening X-ray images [12]. Therefore, we used this filtering technique to reduce noise in our roi.

The Gaussian blur is a smooth blurring technique resembling that of viewing the image through a translucent screen. It uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. Values from this distribution are used to build a convolution matrix which is applied to the original image. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighbouring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters.

The Canny edge detection algorithm was developed by John F. Canny in 1986. It is a multi-step process consisting of:

- Applying Gaussian smoothing (or other filters in our case) to the image to help reduce noise
- Computing the image gradients using the Sobel kernel
- Applying non-maxima suppression to keep only the local maxima of gradient magnitude pixels that are pointing in the direction of the gradient
- Defining and applying the minimum and maximum thresholds for Hysteresis thresholding

Together, a Gaussian blur and Canny edge detection visually improved the quality of images.

3.2.1.5. Data normalization

Standardizing the images ensures an uniform scale, or in the other words, that each pixel value has a similar range. It helps to mitigate the effect of different lighting conditions and shadows, making the model more robust to variation in image capture conditions. This may be particularly important for X-ray datasets. Standard scaler also reduces the influence of outliers (extreme pixels values), which may distort the learning process.

Data from images used for training classical models were normalized using the 'StandardScaler' function from the sklearn preprocessing library.

3.2.1.6. Data splitting

Chest X-rays, raw roi and filtered roi's were used to train each classical model separately. In each case, data was split into train and test sets using the train_test_split method of sklearn library. The test size used was 20% and the rest of the data was used for training classification models.

3.2.2. Image pre-processing for DL models

CNNs are commonly designed to handle images and therefore feature extraction is already a part of it. For this reason, we did not filter images. Image pre-processing for CNNs included the following:

3.2.2.1. Class weights

Class weights for training data was calculated using 'compute_class_weight' function from the sklearn utilities module. See section 3.2.1.1 for more details about class weights and how they function.

3.2.2.2. Resizing images

For the VGG16 model, the whole images and roi's were resized to 224 x 224 pixels. On the other hand, for LMAP3 model whole images and roi's were resized to 256 x 256 pixels.

3.2.2.3. Selecting the lung area as the region of interest (ROI)

As specified in section 3.2.1.3, the lung area was chosen as the roi. These images along with the original whole chest X-rays, with or without data augmentation was subsequently used to train our CNN models.

3.2.2.4. Data augmentation

Data augmentation is a critical technique used to increase the diversity and size of the training dataset without collecting new data. It commonly includes techniques like rotation, flipping, zooming in and out, and noise addition that enhances the variability of the existing dataset. It reduces overfitting and improves the accuracy of the model by exposing the model to such a varied training set.

To augment the data at hand, we chose 3 transformations: rotation, zooming and flipping. The images are transformed with a random rotation of +/-36°, a random zoom of +/-10% and a chance to be horizontally flipped (Figure 3.2). We tested data with and without augmentation in our CNNs to compare how each model works.

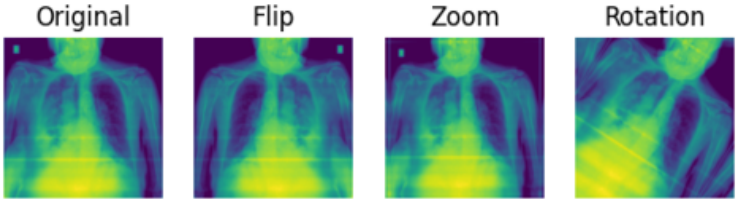


Figure 3.2. Data augmentation of a sample image where each image represents a specific type of augmentation compared to the original image (extreme left)

3.2.2.5. Normalizing data

To normalize data for CNNs, batch normalization was used. Refer section 3.2.1.5 for more details on normalization and why it is performed.

3.2.2.6. Data splitting

Data was split into 3 parts- training (80%), validation (10%) and test (10%) dataset for all CNN models.

3.3. Overview of models

To detect COVID-19, viral pneumonia and/ lung opacity from chest X-rays, we decided to train two different types of models- classification and deep learning models. Even though we work with images, with the aim to go into deep learning, running classical models is important for the sake of learning, and showing why deep learning techniques, such as CNN, are more suitable.

3.3.1. Classification models

Classification is a type of supervised ML where the goal is to predict which categories or classes new data falls into based on predefined categories or classes. The classical models tested in this project on whole images are Linear Regression, Support Vector Machine, Random Forest, KNeighbors Classifier (KNN), Decision Tree, Bagging, AdaBoost, XGBoost, and Voting. The parameters for Classification models are given in Table 3.1.

Table 3.1. Parameters used for Classical models

Model	Parameter	Value
Logistic Regression	max_iter	2000
	class_weight	class_weights_dict
Support Vector Machine	kernel	'linear'
	class_weight	class_weights_dict
K-Nearest Neighbors	n_neighbors	7
Decision Tree	class_weight	class_weights_dict
	random_state	123
AdaBoost	base_estimator	DecisionTreeClassifier()
	n_estimators	50
	learning_rate	1.0
	random_state	123
Voting Classifier	estimators	[('rf', RandomForestClassifier), ('bagging', BaggingClassifier), ('xgboost', XGBClassifier)]
	voting	'hard'

Based on the accuracy and training time, we chose the top 3 models (Random Forest, Bagging and Boosting) to run the ROI data with and without the Gaussian filter.

Random forest is a commonly used ensemble learning algorithm, trademarked by Leo Breiman and Adele Cutler. It combines the output of multiple decision trees after training to reach a single result [13]. Bagging is yet another ensemble learning technique designed to improve the accuracy and robustness of ML models, particularly those that are prone to high variance. It

involves creating multiple subsets of the original dataset using a technique called bootstrapping. Each subset (bootstrap sample) is used to train a separate model (typically the same type of model). The final prediction is made by combining the predictions of these individual models, usually through majority voting for classification [14]. XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library that implements ML algorithms under the Gradient Boosting framework designed for speed and performance. It is widely used for structured/tabular data [15]. XGBoost builds a predictive model by combining the predictions of multiple individual models, often decision trees, in an iterative manner.
The parameters chosen for each of these models are summarized in Table 3.2

Table 3.2. Parameters for Random Forest, Bagging and XGBoost

Model	Parameter	Value
Random Forest	n_estimators	100
	random_state	123
Bagging	estimator	RandomForestClassifier()
	n_estimators	100
	random_state	123
XGBoost	objective	'multi:softmax'
	num_class	4
	scale_pos_weight	class_weights

We would like to emphasize that for each model, except for KNN and XGBoost, classical weighting was included and models were balanced. Other models such as KNN and XGBoost do not support class weights. KNN is a model with distance-based voting, and classifies a sample based on the majority vote of its k-nearest neighbors determined by distance metrics. Since this approach relies purely on proximity, it does not have mechanism to incorporate class weight. Additionally, introducing class weights into the distance calculations or the voting process would complicate the algorithm significantly. XGBoost focuses on sample weights

through the boosting process. Gradient Boosting Mechanism optimizes the model by adding trees to minimize a loss function based on gradients, and it does not support class weights.
Furthermore, Bagging does not support class weights either, since involves generating multiple subsets of the training data by random sampling with replacement (bootstrap sampling). However, we have handled the class imbalance in Bagging by choosing RandomForestClassifier as an estimator.

3.3.2. Convolutional Neural Networks (CNN)

Deep Learning (DL) is a subset of machine learning that involves neural networks with many layers (deep networks). It is characterized by its ability to automatically learn hierarchical representations of data. The evolution of DL has significantly advanced the field of image processing, enabling machines to perform tasks such as image classification, object detection, and image generation with high accuracy.
Neural Networks are one of the many methods of DL and excel at handling topological data such as images. Neural networks are computational models inspired by the human brain, composed of layers of interconnected nodes (neurons). These networks transform input data through a series of weighted connections and activation functions to produce an output.
CNNs are specifically designed to handle grid-like data structures. They consist of multiple layers, primarily including convolutional layers, pooling layers, and fully connected layers. Lower layers capture basic features like edges and textures, while deeper layers capture more complex patterns and object parts. By using small filters, CNNs exploit spatially local correlations in data, reducing the number of parameters compared to fully connected networks. This makes CNNs computationally efficient and suitable for large-scale image processing tasks. Scalability of CNNs is given by adapting the depth (number of layers) which allows the model to handle increasingly complex functions.
The core functional layers of a CNN encompass Convolutional, Pooling and Dense Layers:

- Convolutional Layers:
  - Apply convolution operations to the input image using filters (typically 3x3).
  - Filters slide over the image to produce feature maps that highlight specific patterns like edges, textures, and shapes.
- Pooling Layers:
  - Reduce the spatial dimensions of feature maps, preserving the most important information while reducing computational complexity.

○ Common pooling techniques include max pooling (taking the max of a striding window) and average (taking the mean of a striding window) pooling.

- Fully Connected Layers (Dense Layers):

○ After several convolutional and pooling layers, the high-level reasoning in the neural network is done via fully connected layers.

○ These layers flatten the feature maps and connect every neuron to every neuron in the next layer, similar to traditional neural networks.

- Activation Functions:

○ Introduce non-linearity into the model, allowing it to learn complex patterns.

○ ReLU (Rectified Linear Unit) is the most commonly used activation function in CNNs. For the final Dense layer that predicts the classification a softmax or sigmoid function is applied.

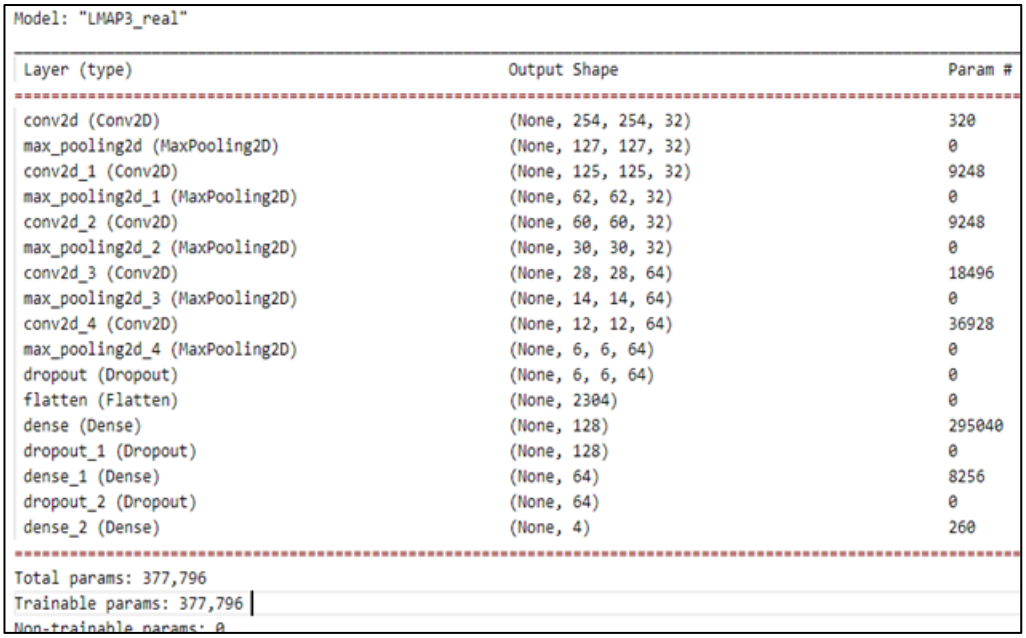
In our project, we used two different CNN models- VGGNet and LMAP3. VGGNet is a convolutional neural network architecture that was developed by the Visual Geometry Group (VGG) at the University of Oxford in 2014. It is known for its simplicity and depth in network architecture design. It uses very small (3x3) convolution filters stacked in deep layers. The approach of using small convolution filters in deeper networks allows for more complex features to be learned without a dramatic increase in the number of parameters.
The model description for VGG16 used in our project is shown in Figure 3.3.

```
Model: "VGG19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 512)               20024384

 dense (Dense)               (None, 4)                 2052

=================================================================
Total params: 20,026,436
Trainable params: 20,026,436
Non-trainable params: 0
```

Figure 3.3. Model description for a pre-trained VGG16 showing the number of parameters used

Local Mean Activity Pattern 3 (LMAP3) CNN is a special architecture designed to address certain challenges in image processing and computer vision tasks. At its core it is a repeated sequence of Convolutional, MaxPooling and Dropout layers (5 repetitions) with an increasing

number of filters, i.e., doubling the amount of filters per convolutional layer. Due to the MaxPooling layer the effective image size is being halved in both dimensions per layer cycle. The model description for an LMAP3 model used in our project is shown below.



Model: "LMAP3_real"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	320
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_3 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_4 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout (Dropout)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 128)	295040
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 4)	260

Total params: 377,796
Trainable params: 377,796
Non-trainable params: 0

Figure 3.4. LMAP3 model description showing the various layers and parameters used ín each layer

The parameters used to compile these models are tabulated below.

Table 3.3. Parameters used for compiling CNNs

Optimizer	Adam
Loss function	Categorical Crossentropy
Metrics	Accuracy
Epochs	25
Batch size	32
Class weights	As shown in section 4, Figure 4.1 (right)

3.4. Model Evaluation

In the evaluation of ML models, especially those designed for medical image classification tasks such as COVID-19 detection from X-ray images, it is crucial to use appropriate metrics. These metrics help determine the performance of the models in distinguishing between normal

images, lung opacity, viral pneumonia, and COVID-19. We are not going deeper in the explanation of the used metrics, since we already did it in the previous reports, but we will only repeat once again what are the metrics we were focused on.
The metrics we used to determine what is the model that suits the best to our dataset are accuracy, precision or positive predictive value, recall, F1-score, Area under the Receiver Operating Characteristic Curve (AUC-ROC), and confusion matrices.

Evaluating ML models using a combination of the mentioned metrics, provides a comprehensive understanding of their performance. In the context of medical image classification for COVID-19 detection, these metrics ensure that the models are not only accurate but reliable in identifying true cases and minimizing false positives and negatives. This approach to model evaluation may be critical in developing effective diagnostic tools.

The important metric we used for DL model are Saliency maps. Saliency maps highlight which parts of an X-ray image contribute most to the CNN's decision-making process. They provide a visual explanation of the model's focus areas, and may help in understanding why a particular diagnosis was made. By examining saliency maps, we can identify if the CNN is making decisions based on incorrect or misleading features, which is essential for refining the model and improving its performance.
In the next chapters, we will show and interpret our results obtained with Classical models and with DL models.

3.5. Interpretability

Model interpretability in deep learning involves understanding how a model makes predictions by identifying which input features or data patterns influence its decisions. Interpretability is important for analysts and scientists to be able to understand the workings of the model. For this purpose, we looked into saliency maps. Saliency maps in deep learning are essentially heatmaps that highlight the most important regions of an input image with respect to a specific output of a neural network. They are used to visualize which parts of an input image contribute the most to a network's decision-making process.
Following is a step-by-step explanation of how a basic saliency map is computed:

- Forward Pass: Run the input image through the neural network to get the output prediction.

- Gradient Calculation: Compute the gradient of the output with respect to the input image. This is typically done using backpropagation.
- Absolute Values: Take the absolute value of the gradient to capture the magnitude of change.
- Heatmap Generation: Convert the gradient magnitudes into a heatmap, with higher values indicating more important regions.

3.6. Software and libraries

We studied the data and developed our model using Python and worked with Jupyter notebooks. Various libraries and their versions used are listed in the 'requirements.txt' file in our Github repository (may24_bds_int_covid_xray).

4. Results

In this section we will discuss the results we obtained while training our ML and DL models.

4.1. Results of classical models (RF, Bagging, XGBoost) on whole images

4.1.1. Class weights

The important characteristic of our dataset is that originally it was not balanced, since the distribution of images were not the same in each category. To overcome this, we decided to focus first on statistical weighting and perform the classification models on statistically weighted dataset. The distribution of images in training set (80% of all data), and corresponding computed class weights are shown in Figure 4.1.

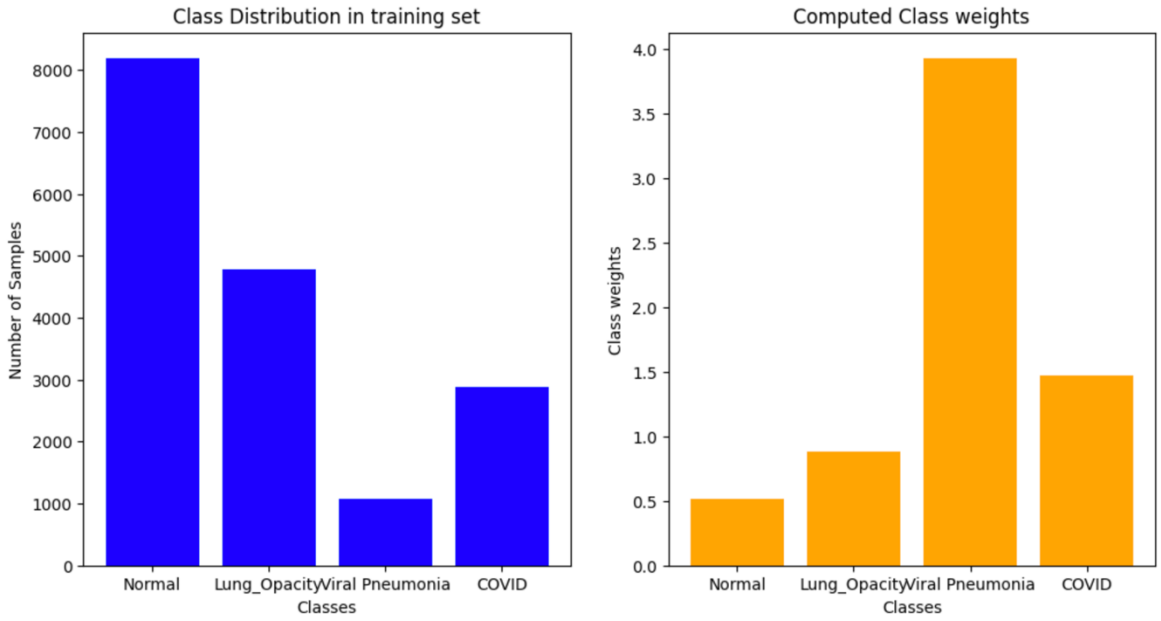


Figure 4.1. The distribution of images in training set (left, blue) and computed class weights (right, orange)

After scaling the data, we performed classification models with: Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), K-Neighbors (KNN), Decision Tree (DT), Bagging with RF as an estimator, Boosting, XG Boost, and Voting Classifiers. The obtained accuracies of all the models are shown in Figure 4.2.

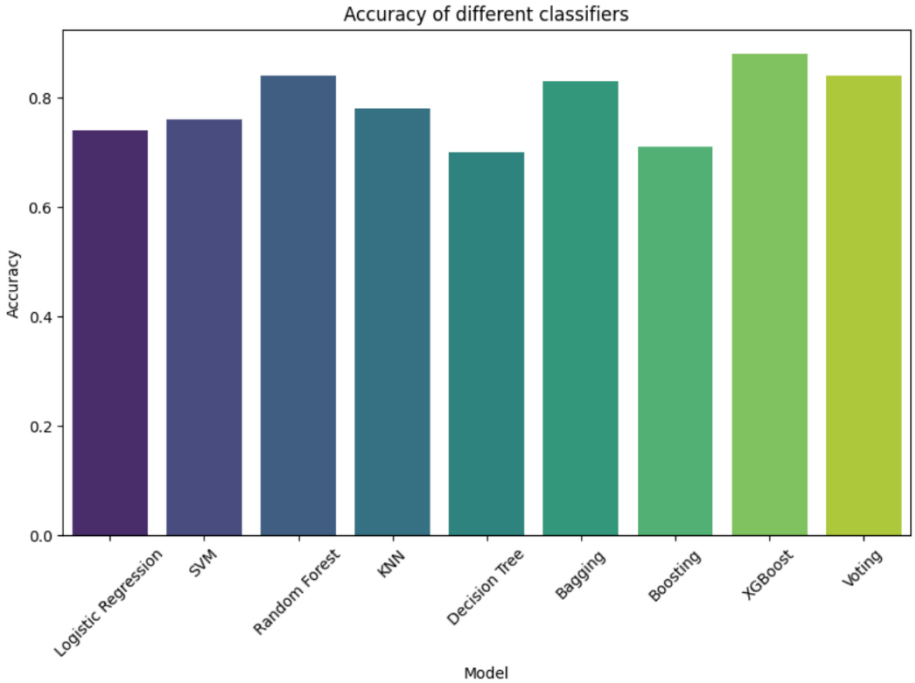


Figure 4.2. Accuracies of different Classification models trained on the whole X-rays

The best accuracies had RF, Bagging with RF, XG Boost, and Voting. Since Voting classifier took almost 800 min to train, we decided not to further use it in our analyses. The total training times for each model are shown in Figure 4.3.

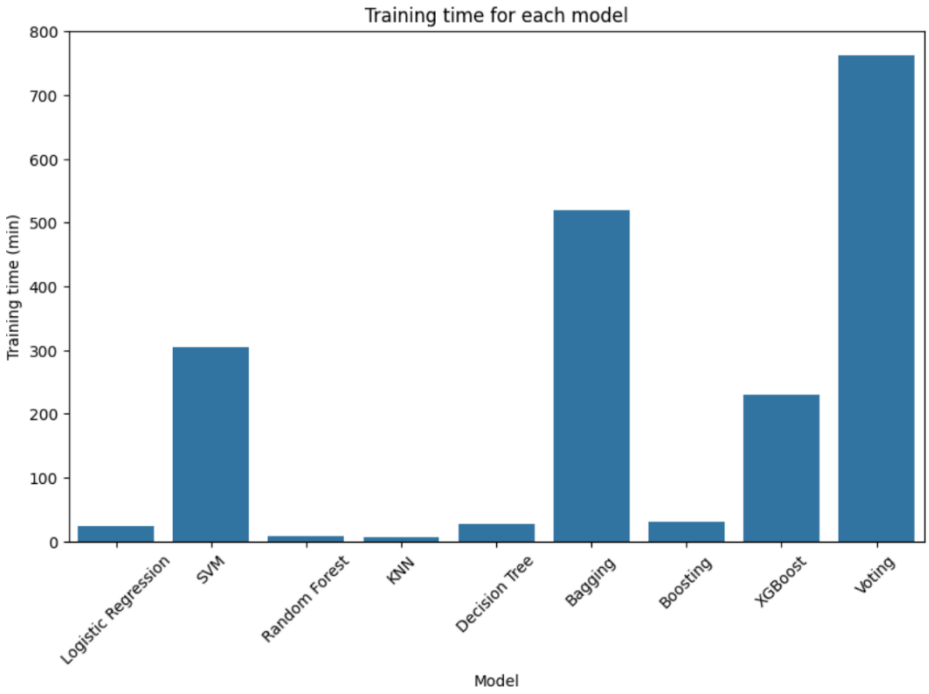


Figure 4.3. Total training time for each model trained on the whole X-rays (run on the same computer)

It is important to emphasize that models KNN and XGBoost do not support class weighting. Thus, they were run on an imbalanced dataset. Classifier Bagging also does not support class weighting. However, we have overcome this problem by using RF classifier as an estimator and here we included statistical weighting.

In KNN and XGBoost, sample imbalance can be handled by oversampling or undersampling. When including the RandomUnderSampler function from the "imblearn" library on our dataset, the resulting accuracy scores dropped significantly which is why we did not continue to use random undersampling. The summarized data are given in Table 4.1.

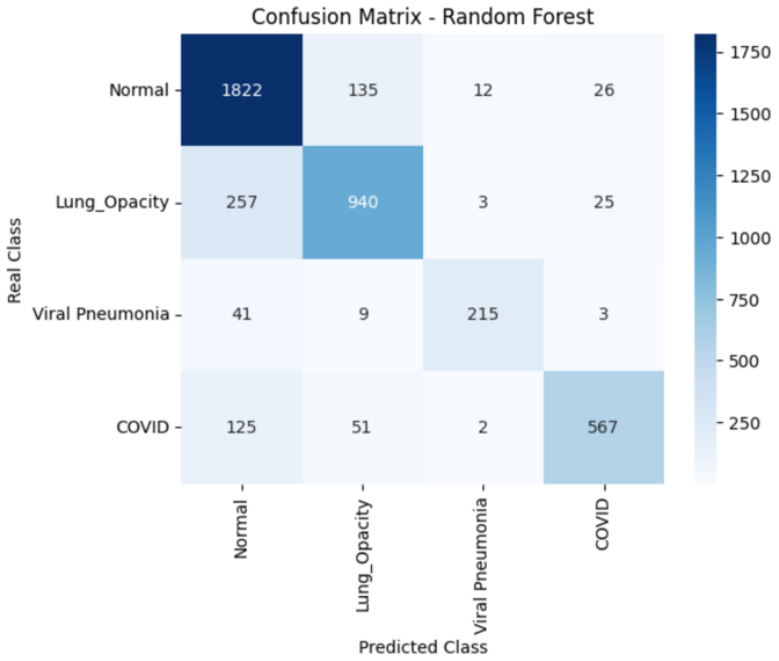
We also attempted to use random oversampling and SMOTE. Unfortunately, when applying the RandomOverSampling function to the dataset, the kernel crashed after several hours, and due to time issues, we did not follow this strategy of balancing the dataset any further.

Table 4.1. Accuracy scores with undersampled training data

Classifier	Accuracy Score
RandomForest	0.79
AdaBoost	0.52
LogisticRegression	0.72
KNeighbors	0.71
SVC	0.78

At this point, we will continue presenting only results for RF, Bagging with RF, and XGBoost run on dataset statistically weighted in the case for the first two models. The results for all the other models could be found in jupyter notebooks.

Confusion matrices for RF, Bagging with RF, and XGBoost are shown in Figure 4.4.



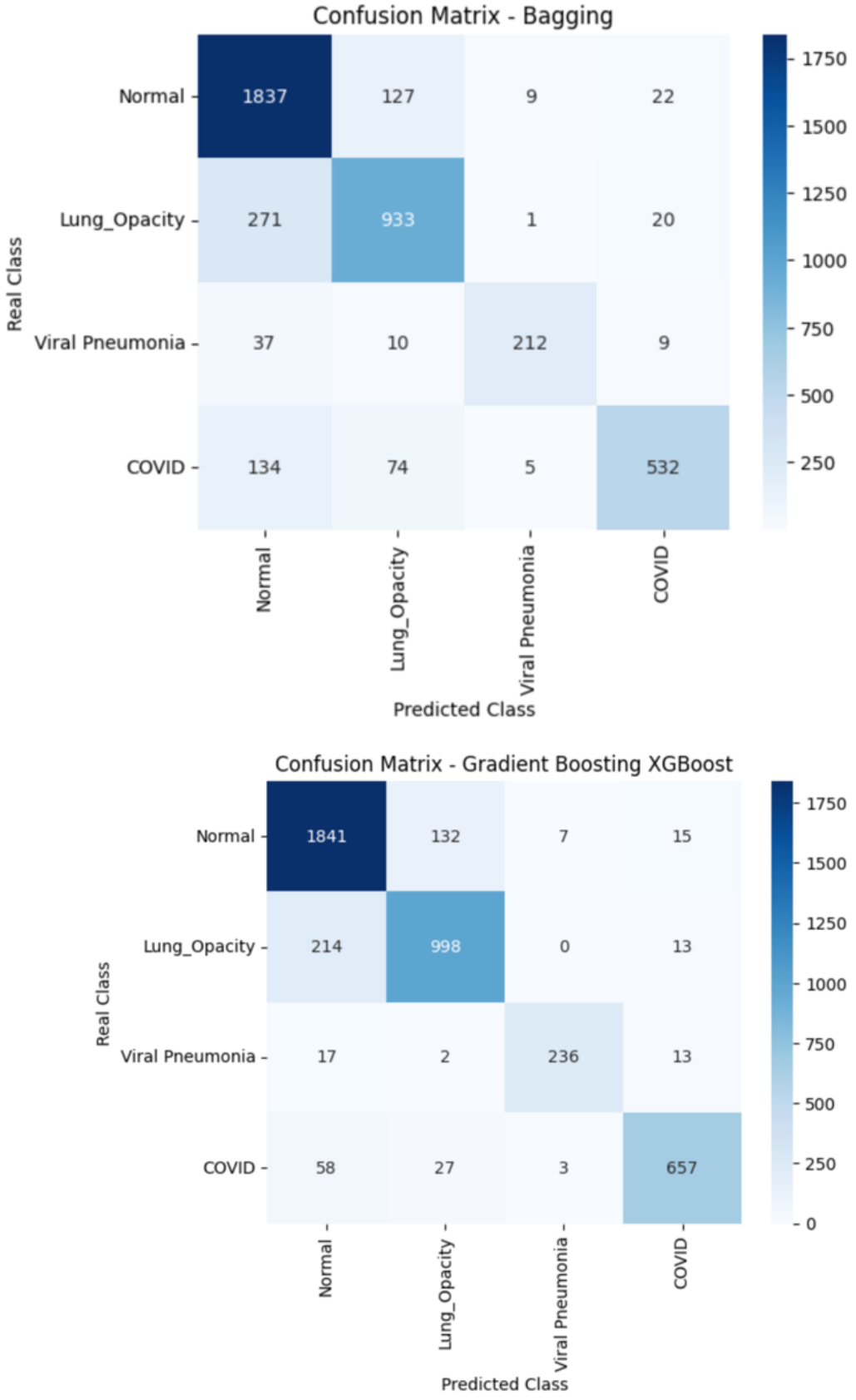
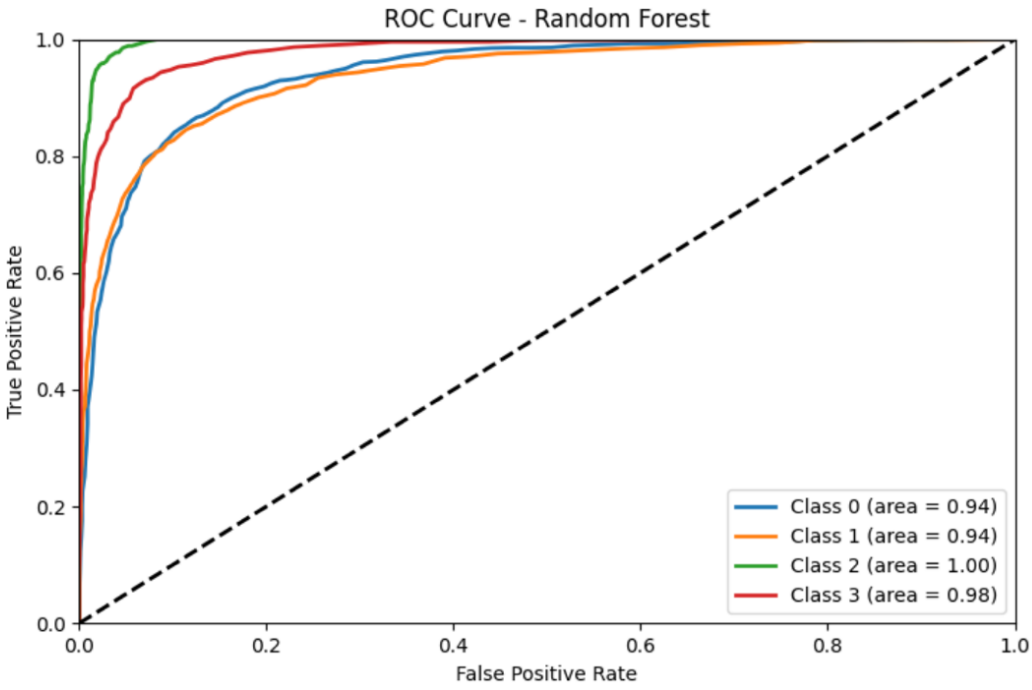


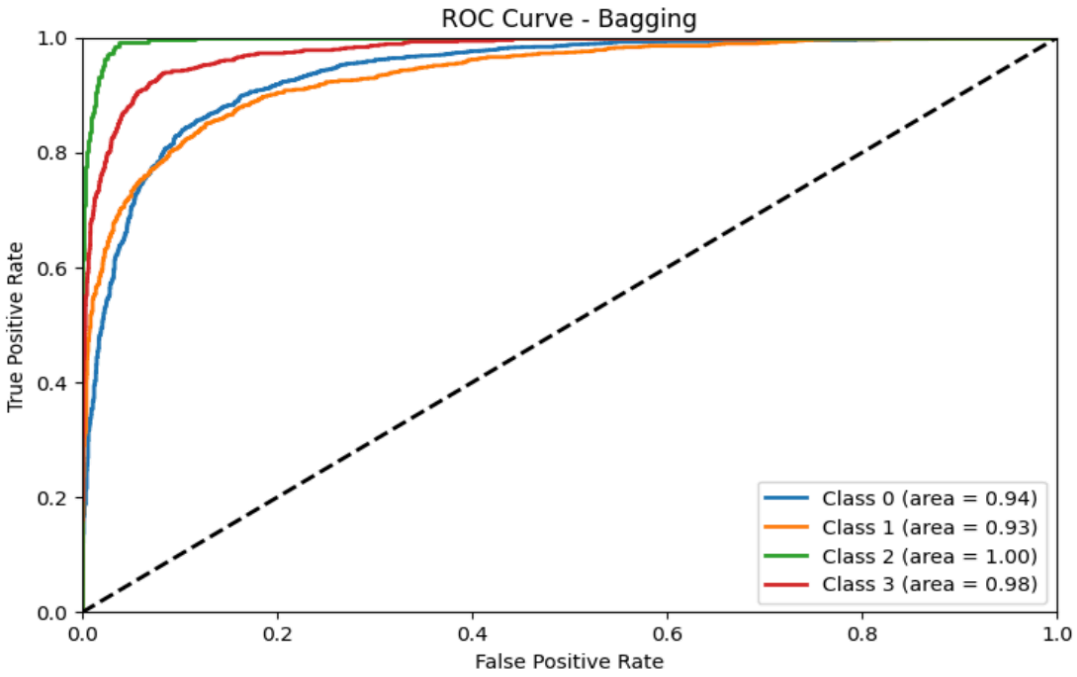
Figure 4.4. Confusion Matrices for different Classification models: Random Forest, Bagging and XGBoost

From the presented confusion matrices, we can conclude that in the case of all presented models, the biggest number of true positives always had category ***Normal***. On the other side, the lowest number of true positives always had category ***Viral Pneumonia***. This may be related to the fact that different categories have different number of samples, even though class weights were included in the case of RF, and Bagging with RF. Also, we can conclude that, in the case of XGBoost, that even though the dataset was not class weighted the results are similar to the previous to models (RF and Bagging) and that they are always following the same trend.

The important parameter to be considered when choosing the right ML model is Area Under the Receiver Operating Characteristic Curve (AUC-ROC). An AUC-ROC of 1 indicates a perfect classifier, while and AUC-ROC of 0.5 indicates a model no better than random guessing.

ROC curves together with calculated AUC-ROC for RF, Bagging with RF, and XGBoost are shown in Figure 4.5.





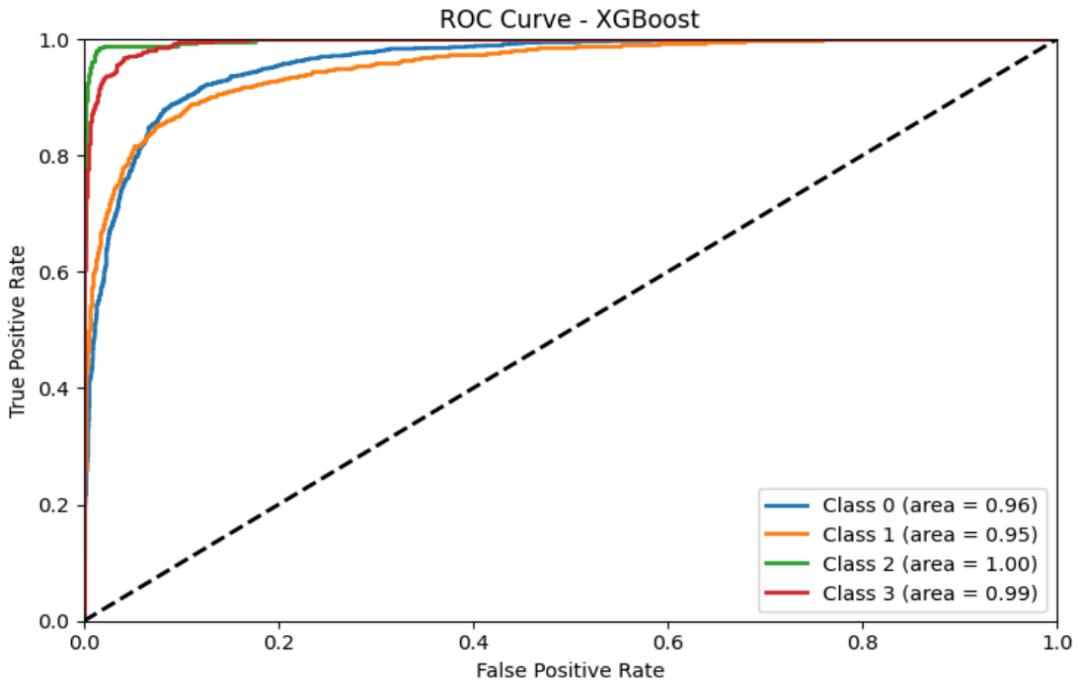


Figure 4.5. ROC Curves for all categories (Normal – 0, Lung Opacity – 1, Viral Pneumonia – 2, COVID – 3) obtained in Random Forest, Bagging and XGBoost

We can observe that for all categories in these three models, AUC-ROC are higher than 0.9 and close to 1, suggesting that all three classifiers are suitable. Also, as can be seen in figures shown, for high true positive rate, the false positive rate is low.

For all three models presented, RF, Bagging with RF, and XGBoost, and for all categories we have obtained Precision-, Recall-, and F1-Scores 0.8 and higher. This indicates good model predictability. Also, these results imply that the model is not only able to correctly identify most of true positives, but also has a low rate of false positives and false negatives.

The common misclassified images for all classical models performed are shown in Figure 4.6.

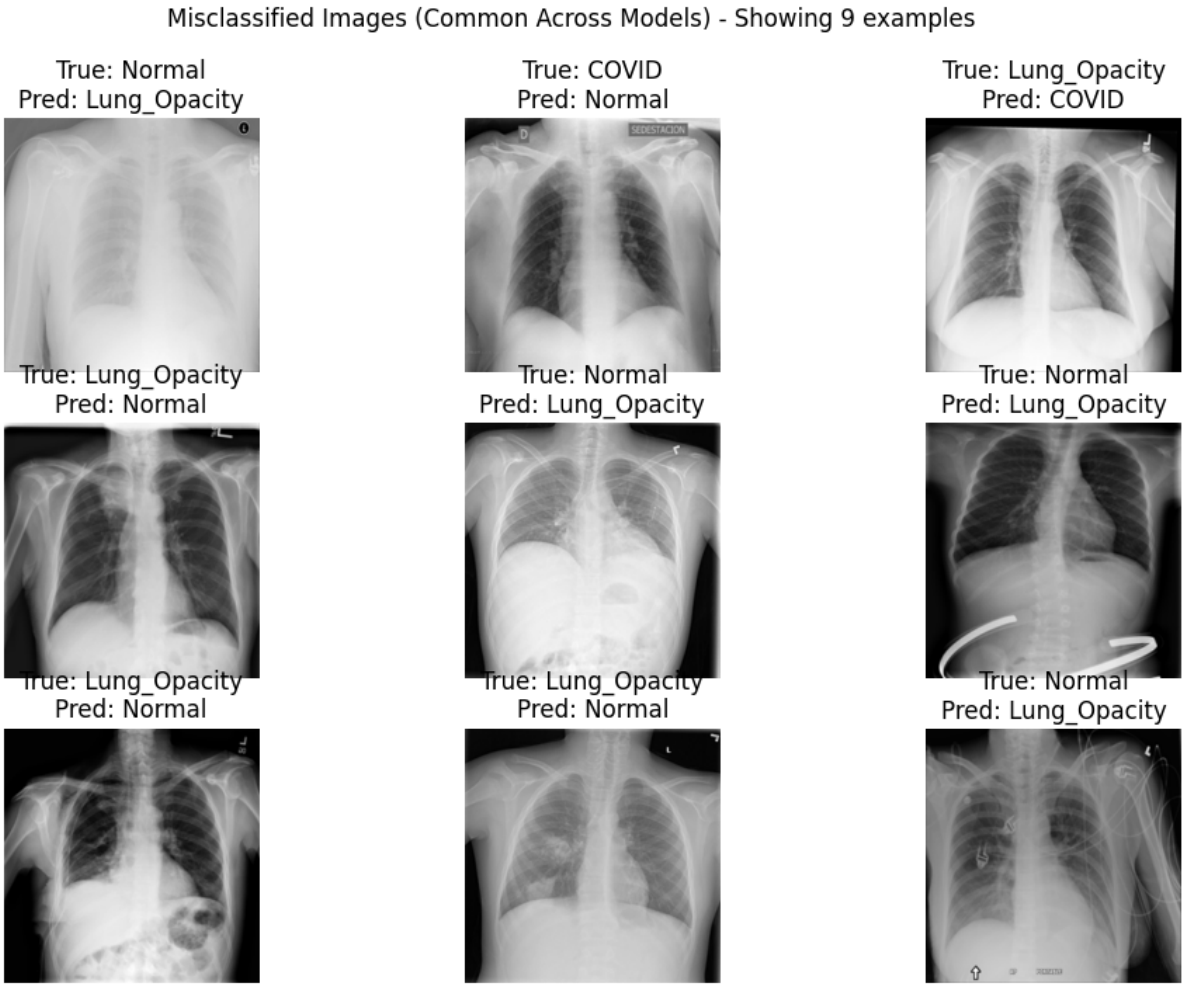


Figure 4.6. Common misclassified images across all performed models

As seen in the above presented, the models with highest accuracies, best performances, and acceptable training times are RandomForest, Bagging and XGBoost. Thus, we tried to optimize these three models before running them on the preprocessed data. However, the codes have never finished, since one optimization took more than two days. This is the reason why we decided to skip this step and just continue with running the three best models on two datasets: 1) dataset of images presenting Region of Interest (ROI), and 2) dataset of images presenting

ROI but with filters. The datasets were statistically weighted to avoid disbalance and class weight was included in Random Forest and Bagging, while it was not possible for XGBoost.

4.2. Results of Classification models on preprocessing images

4.2.1. Classification models on raw ROI

Before performing image processing, we cropped the images and prepared the dataset containing only roi, as explained in section 3.2.1.3. The classification models, RF, Bagging with RF, and XGBoost were trained on the ROI dataset and results are presented below. Model accuracies for raw ROI (only lungs) are shown in Figure 4.7.

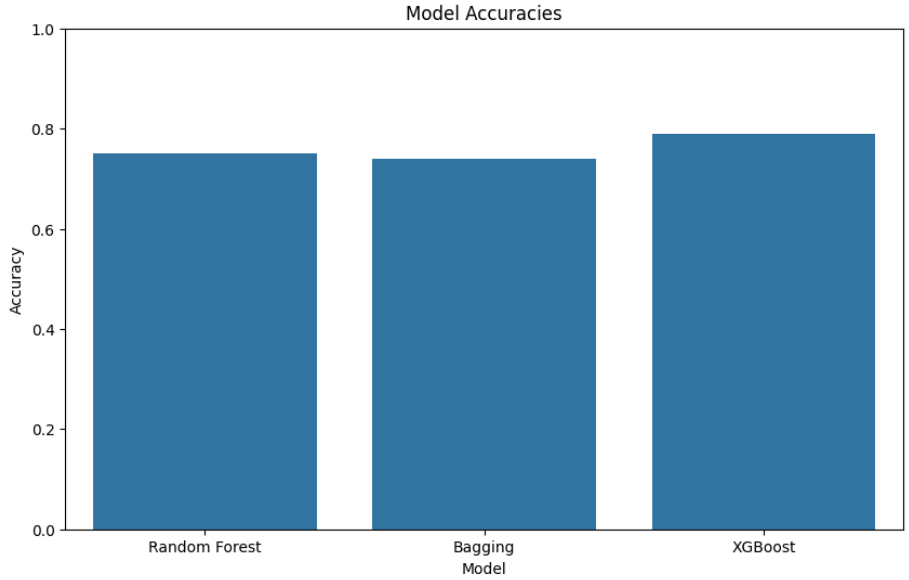


Figure 4.7. Accuracies of Random Forest, Bagging and XGBoost trained on ROI (lung area)

Even though, the accuracies are still high, they are slightly lower than it is the case when the same models were trained on the raw dataset, Figure 4.2.

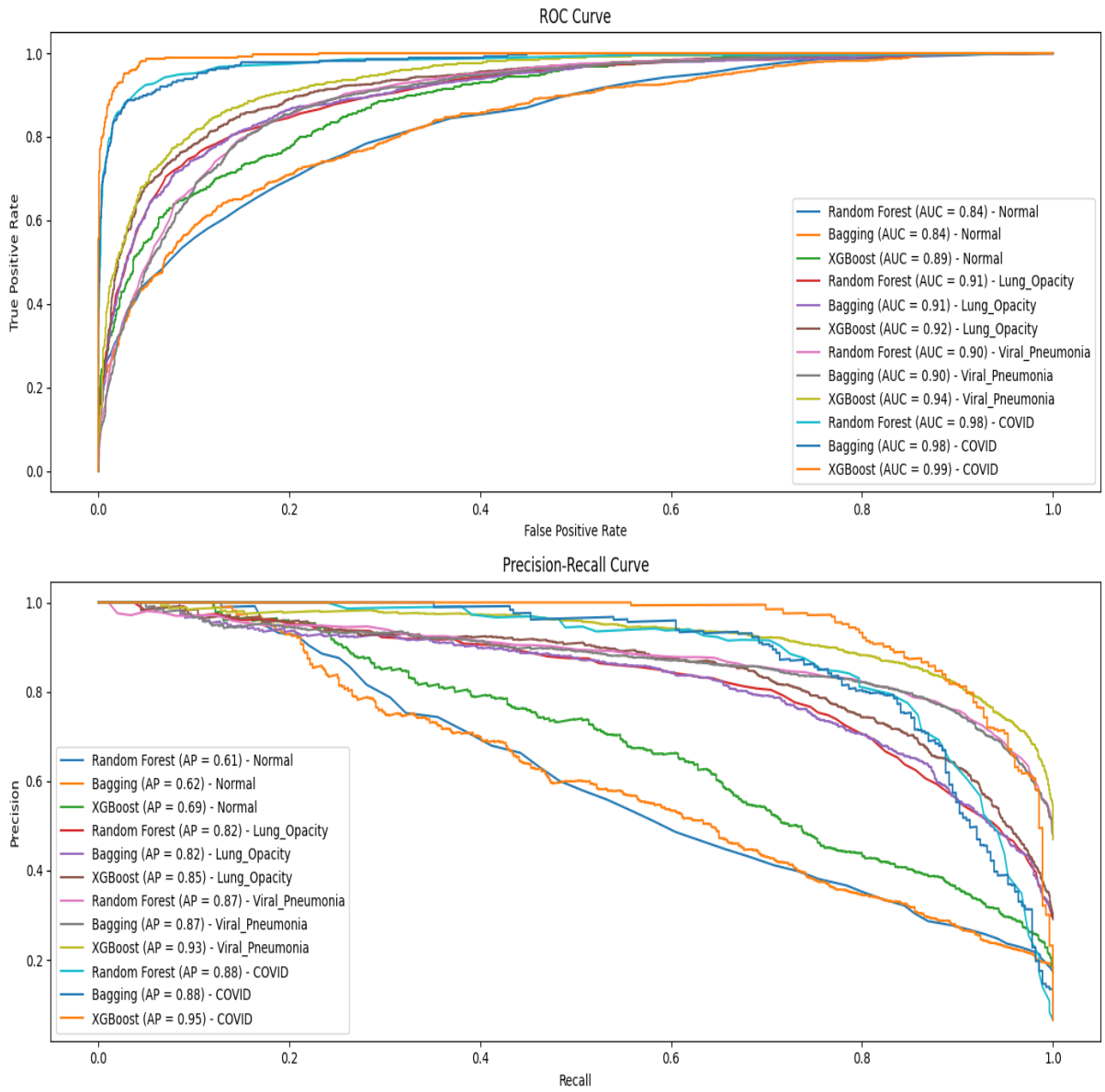


Figure 4.8. ROC Curve and PR Curve for each category obtained with Random Forest, Bagging and XGBoost on ROI (lung area)

From the ROC and PR Curves shown in Figure 4.8., we can conclude that, for the most of the models and categories, when the true positive rate is high the false positive rate is low. However, this ratio is not as good as in the case of whole images (X-rays) shown in the previous chapter. Furthermore, the AUC-ROC are slightly lower for ROI than it is the case for the whole images. From the PR curve shown in Figure 4.8., we may conclude that all the models show fluctuations and that AP values are lower compared to the whole images (data could be seen in notebooks and in the previous reports).

4.2.2. Gaussian blurring and Canny edge detection

The application of Gaussian blur to X-ray images revealed that increasing kernel sizes significantly impacted image clarity. For normal lung X-rays, an unfiltered image maintained clear structural details. However, as the kernel size increased from (3, 3) to (7, 7), the images showed progressively more blurring, with the (3, 3) kernel providing the best balance between noise reduction and detail preservation. In contrast, for COVID-19 lung X-rays, the unfiltered images distinctly highlighted hazy regions indicative of infection.

The (3, 3) kernel maintained these critical details while reducing noise. However, larger kernels like (5, 5) and (7, 7) excessively blurred the images, obscuring important diagnostic features.

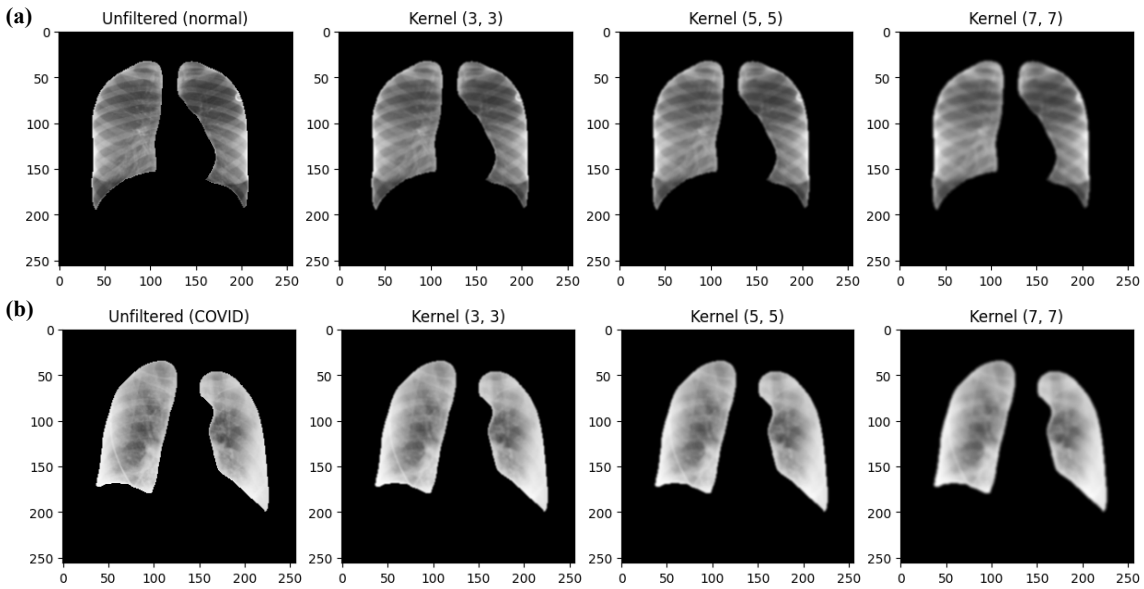


Figure 4.9. Gaussian blur of X-ray images (a) Representative lung area of a normal volunteer filtered with a kernel size (3, 3), (5, 5) or (7, 7) (b) Representative lung area of a COVID patient filtered with a kernel size (3, 3), (5, 5) or (7, 7). The scale signifies the width and height of the image in pixels.

The edge detection analysis further emphasized the effectiveness of the (3, 3) Gaussian blur in conjunction with Canny edge detection. In healthy individuals, this combination effectively highlighted lung structures while reducing noise. For COVID-19 patients, it successfully delineated both normal lung structures and infection-related hazy regions. This dual approach of Gaussian blurring followed by edge detection provided optimal for enhancing diagnostic features and differentiating between normal and COVID-19 affected lungs.

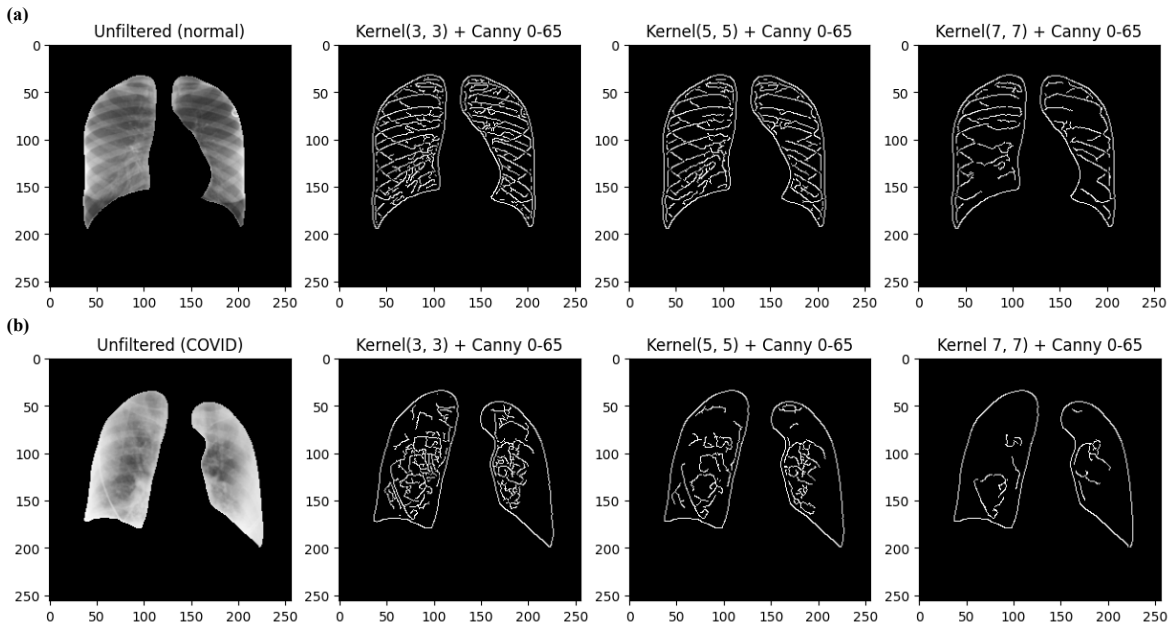


Figure 4.10. Gaussian blur and Canny edge detection (a) Gaussian filtered X-ray image of a healthy individual followed by a Canny edge detection (threshold - min: 0, max: 65) (b) Gaussian filtered COVID X-ray image followed by a Canny edge detection (threshold - min: 0, max: 65)

4.2.3. Classification models on filtered ROI

We further checked whether the application of the Gaussian blur and Canny edge detection would improve the accuracies obtained with RF, Bagging with RF, and XGBoost on ROI. Accuracies for these models are shown in Figure 4.11.

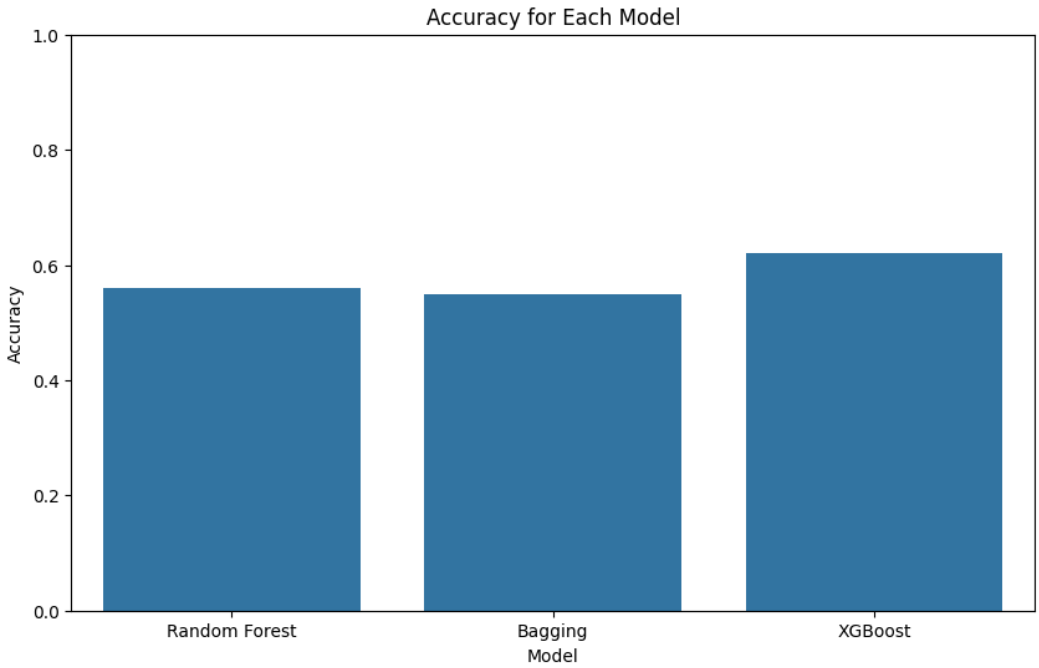


Figure 4.11. Accuracies of Random Forest, Bagging and XGBoost trained on filtered ROI (lung area)

As can be seen from Figure 4.11., accuracies are significantly lower for all three models trained on ROI with filters. The corresponding ROC curves with AUC-ROC, and corresponding PR curves with AP are shown in Figure 4.12. As it is the case with accuracies, AUC-ROC are lower than it is the case with ROI without filters, and significantly lower than it is the case with whole X-rays. It is the same with AP calculated from PR curves shown in Figure 4.12.

Thus, we can conclude that our classification models work good and are suitable for the whole images (X-rays), and that the models are not suitable at all for ROI with filters.

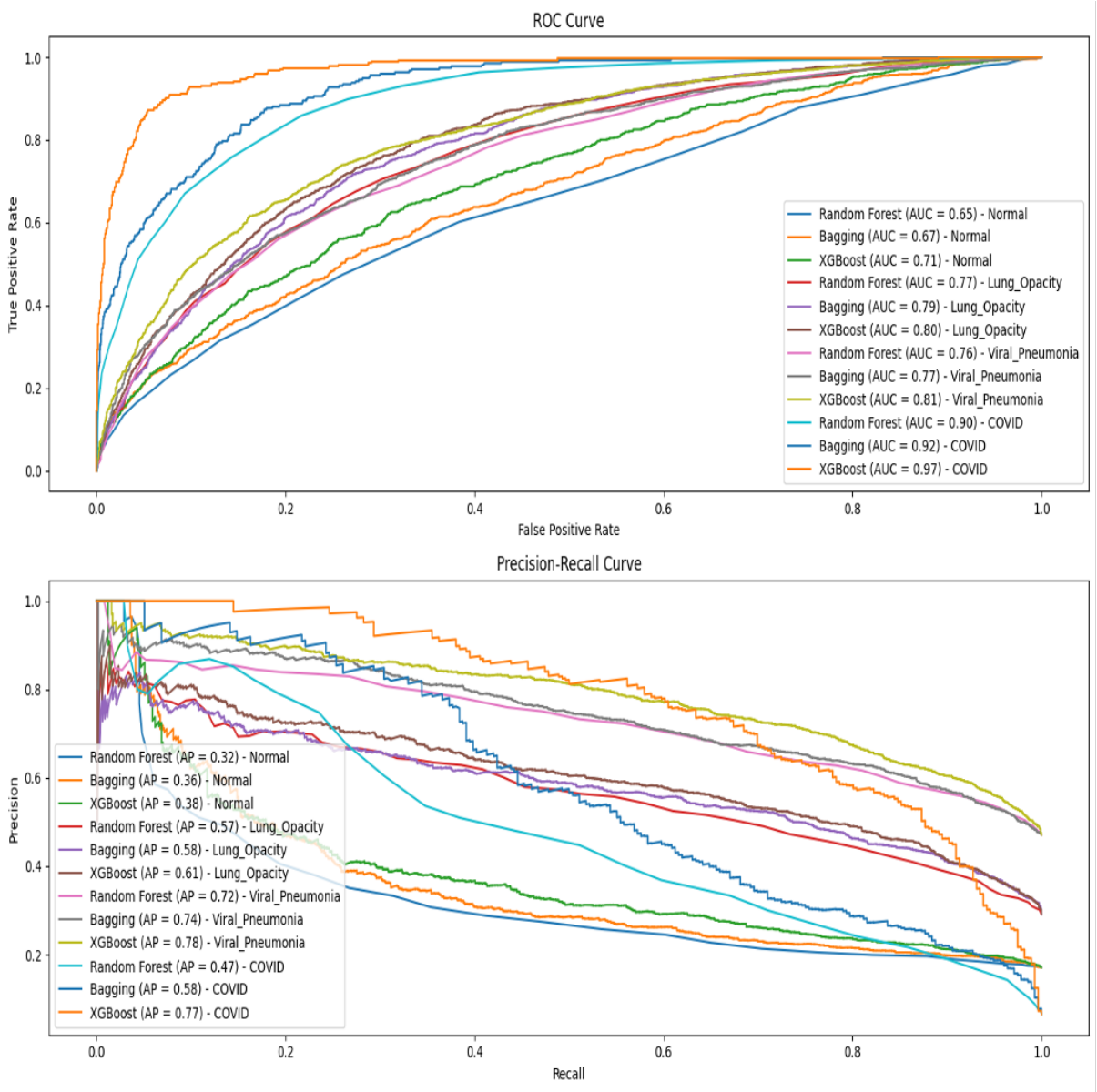


Figure 4.12. ROC Curve and PR Curve for each category obtained with Random Forest, Bagging and XGBoost on filtered ROI (lung area)

4.3. Results of the deep learning model (CNN) and interpretation

The training performance of both VGG19 and LMAP3 models, as illustrated in Figures 4.13. and 4.14., indicates a significant disparity in both accuracy and computation time. LMAP3 not only achieves a higher accuracy but also does so more efficiently compared to VGG19. Consequently, LMAP3 has been chosen as the preferred model for further analysis.

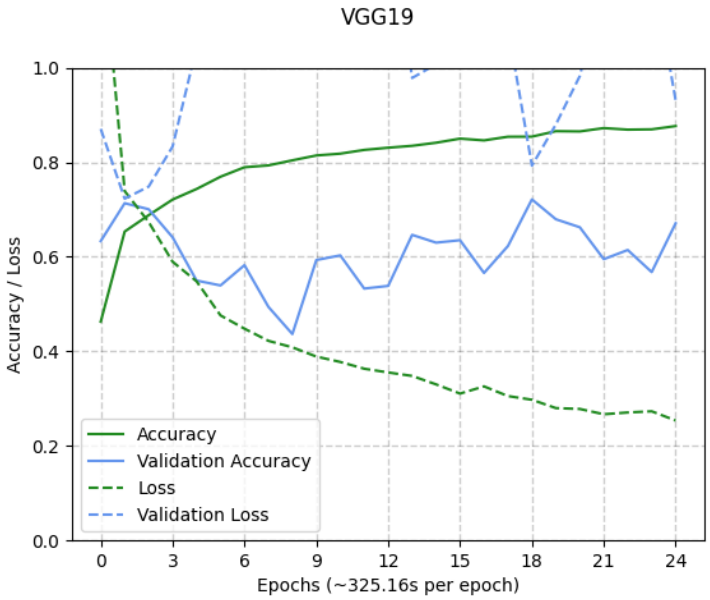


Figure 4.13. The training performance of VGG19

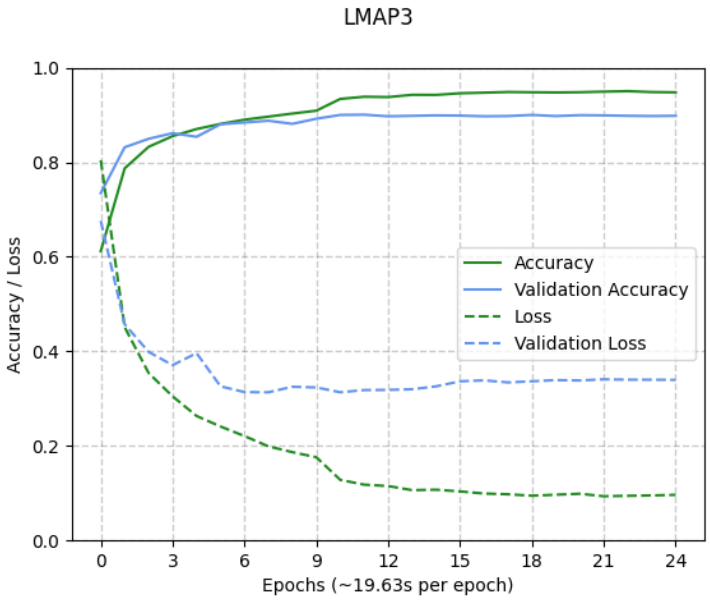


Figure 4.14. The training performance of LMAP3

In the subsequent analysis, the impact of masking the lungs and applying data augmentation techniques on model performance was investigated through four different trainings scenarios:

1. Full image without data augmentation
2. Full image with data augmentation
3. ROI image without data augmentation
4. ROI image with data augmentation

The observed results are summarized in Table 4.2.

Table 4.2. Accuracies and Epoch times for different training scenarios

Training	Accuracy	Epoch Time
Full image without data augmentation	0.90	19.63 s
Full image with data augmentation	0.65	36.09 s
ROI image without data augmentation	0.84	20.42 s
ROI image with data augmentation	0.74	37.38 s

The scenarios involving data augmentation (B and D) resulted in lower accuracy (0.65 and 0.74, respectively) and nearly doubled the computation time compared to their counterparts without data augmentation (A and C). This suggests that data augmentation may not be beneficial in this context, likely due to the nature of X-ray images where augmentations could distort crucial diagnostic features.

Interestingly, the full image scenarios (A and B) outperformed the ROI scenarios (C and D) despite the expectation that only the lung areas (ROI) would contain relevant diagnostic information. This discrepancy indicates that the full images may contain subtle contextual clues outside the lung regions that the model leverages for classification, though from a medical standpoint, this requires cautious interpretation.

The confusion matrix shown in Figure 4.15 of the best-performing model (scenario A) reveals that most misclassifications occur between the "Normal" and "Lung_Opacity" categories. This indicates that while the model is generally accurate, it sometimes confuses these two classes, possibly due to overlapping radiographic features.

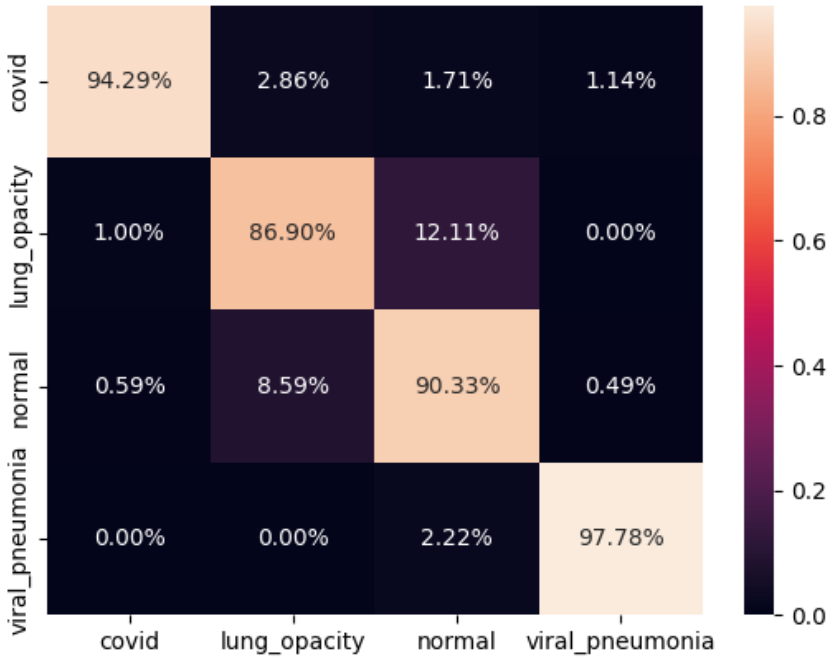


Figure 4.15. Confusion matrix of the best performing CNN model

To further enhance our understanding and trust in the model, we will proceed with the model interpretability phase in the next chapters. This involves using saliency maps to visualize which parts of the X-ray images the model focuses on when making its predictions. By doing so, we can verify if the model's decision-making aligns with medical knowledge and ensure that it is not relying on irrelevant artifacts.

4.4. Interpretability

The saliency maps generated for a random sample of each of the four classes reveal key insights into the model's focus and accuracy. In the saliency maps, brighter pixels denote regions that significantly contribute to the classification. Notably, higher intensities are predominantly observed in the lung areas, indicating that the model effectively captures important regions relevant to the classification task. On the other side, areas outside the lungs, including those covered by the heart, exhibit lower activity on the saliency maps. This observation is consistent with medical knowledge, suggesting that the model accurately identifies and prioritizes medically relevant areas.

The analysis of the original (images in the middle) and overlay images (right) further prove these findings (Figure 4.16). The original X-ray images provide a baseline visual context, while the overlay images, which combine the saliency maps alone (left) and the X-rays (middle),

clearly demonstrate that the salient regions align well with the anatomical structures of the lungs. This coherence between the saliency maps and medical expectations supports the model's reliability in focusing on critical areas for diagnosis, ensuring that its predictions are grounded in relevant medical features.

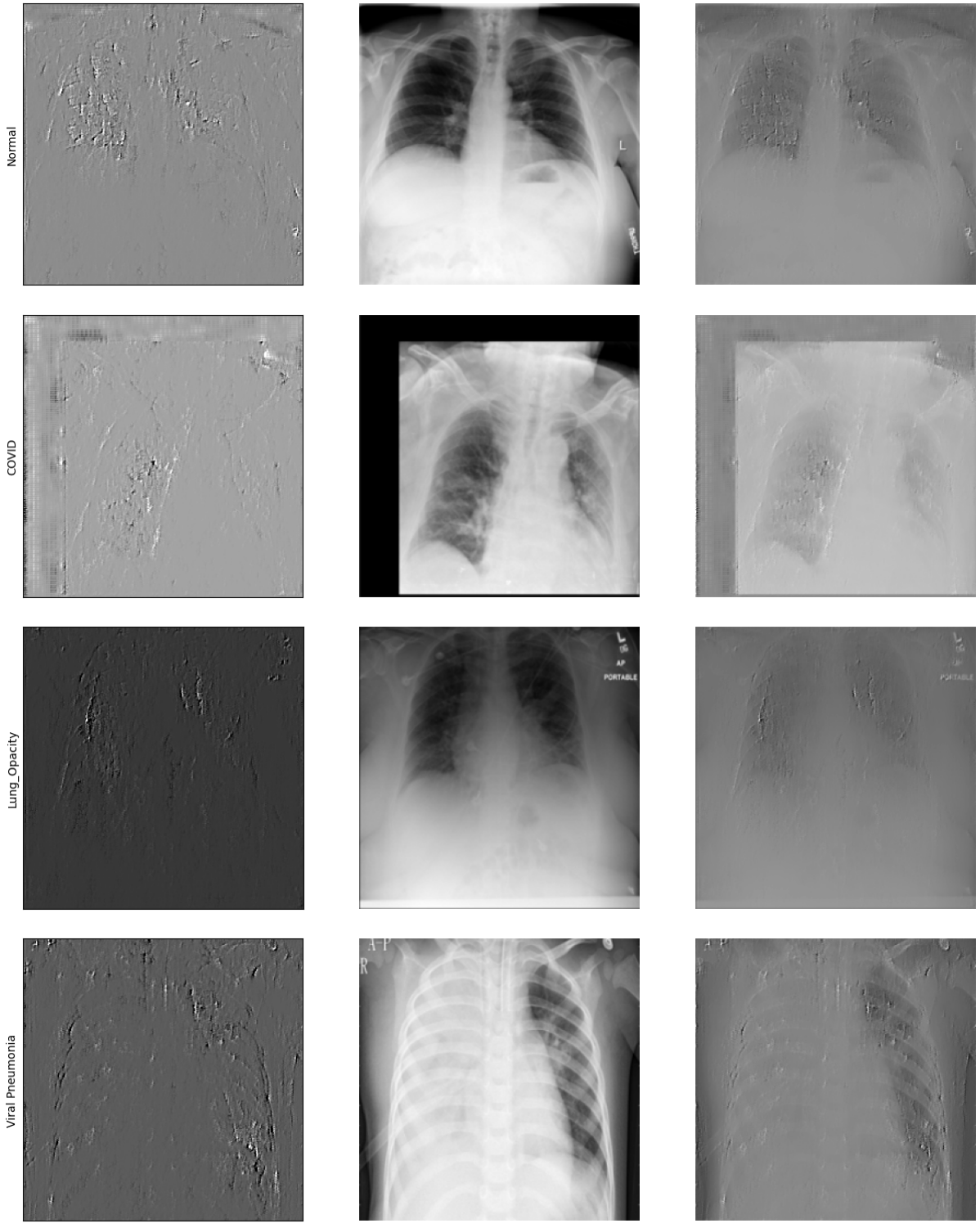


Figure 4.16. Saliency maps of representative images from each class.

While the saliency maps have proven effective, the implementation of GradCAM presented some challenges. GradCAM was intended to offer deeper insights into which kernels and features are most influential in the classification decisions by targeting the last convolutional layer of the CNN and calculating the gradient between that layer and the classification output layer. However, the results obtained were inconsistent and unreliable, indicating potential issues in the implementation. Despite these challenges, the initial promise of GradCAM suggests that with further refinement and testing, it could provide valuable additional insights into model interpretability.

In conclusion, the results from the saliency maps confirm that the model focuses on relevant lung regions, validating its reliability for medical imaging classification tasks. The consistency between the saliency maps and medical expectations reinforces the model's trustworthiness and alignment with medical expertise. Although the GradCAM implementation needs further improvement, saliency maps have demonstrated their efficacy in highlighting critical areas for diagnosis. Enhancing GradCAM could potentially offer even more detailed insights into the model's decision-making processes, further augmenting its interpretability and clinical applicability.

4.5. Comparison of model performances

When comparing the performance of classical machine learning models to convolutional neural network (CNN) models, distinct differences in efficiency and adaptability emerge. Classical models like Random Forest, Bagging with RF, and XGBoost have demonstrated solid performance with high accuracy and robust ROC curves, reflecting their capability to handle the classification task effectively. These models benefit from well-understood algorithms and established methods for dealing with class imbalance through techniques such as statistical weighting. However, their performance can be heavily dependent on the pre-processing steps and they often require more computational resources and time, especially when handling large and complex datasets.

In contrast, CNN models, particularly the LMAP3 architecture, have shown remarkable efficiency and adaptability in processing lung X-ray images. CNNs naturally excel in image-based tasks due to their ability to automatically learn and extract relevant features from raw data without extensive pre-processing. The LMAP3 model, in particular, outperforms classical

models in terms of training speed and handles various input scenarios, including full images and regions of interest (ROI), with greater flexibility. Despite the complexity of CNNs, their end-to-end learning capability and reduced need for manual feature extraction make them more practical and scalable for real-world applications in medical image classification. This advantage positions CNNs, and specifically LMAP3, as a more suitable choice for tasks requiring rapid and accurate image analysis.

5. Discussion

5.1. Why CNNs were chosen despite high accuracies in Classical models?

Despite the impressive accuracies achieved by classical machine learning models, the decision to pursue deep learning convolutional neural networks (CNNs) was driven by several compelling factors. Classical models like Random Forest, Bagging with RF, and XGBoost have demonstrated solid performance metrics, particularly when statistical weighting is employed to address class imbalances. These models are well-established and provide transparent, interpretable results, making them a reliable choice for many traditional classification tasks.

However, the nature of medical imaging, especially in diagnosing conditions from lung X-rays, often involves complex patterns and subtle variations that classical models might not fully capture. CNNs, with their ability to learn hierarchical features through multiple layers, are particularly suited for this task. They excel in automatically extracting relevant features from raw pixel data, reducing the need for manual feature engineering and potentially capturing more nuanced information than classical models. This inherent strength of CNNs in handling high-dimensional image data is a primary reason for their selection in this context.

Deep learning CNNs were chosen for this project despite the high accuracies achieved by classical models due to their superior ability to handle the intricacies of image data and their potential for scalability and adaptability. Firstly, CNNs can learn and generalize complex patterns and textures in medical images that might be missed by classical models, which rely heavily on predefined features. This ability to capture subtle variations and intricate details in X-ray images is crucial for accurately diagnosing conditions like COVID-19, where minor differences in lung patterns can be significant.

Moreover, CNNs are highly scalable and can be fine-tuned with additional data, potentially improving their performance as more annotated images become available. This adaptability is particularly beneficial in a rapidly evolving field like medical diagnostics, where new data and imaging techniques continuously emerge. Furthermore, CNNs can leverage transfer learning, where pre-trained models on large datasets can be fine-tuned for specific tasks, enhancing their efficiency and effectiveness. This approach not only reduces the computational resources required for training but also allows the models to benefit from prior knowledge, further improving their diagnostic accuracy and reliability. Overall, the choice of deep learning CNNs aligns with the goal of developing a robust, scalable, and precise diagnostic tool for COVID-19 detection.

6. Conclusion

6.1. Summary of findings

We started this project with the aim of using DL to guide COVID-19 diagnosis. To this end, we not only trained specialized neural networks, but we also compared it with traditional classification models. In general, both models had an accuracy>= 90% when whole X-ray images were considered for analysis, but when we focused only on the lungs (ROI) with or without additional filtering, the accuracy of the models dropped considerably (75-80% in case of classical models and 74 84% in case of CNNs). These results highlight that simple X-ray images are sufficient to run complex models in our specific use case and one need not perform complex filtering to reduce the noise and improve the performance of the model. This ideally reduces data-preprocessing time, which is extremely beneficial in the context of diagnosing a disease like COVID-19 under limiting time periods.

Interpretability using saliency maps showed that CNNs mainly covered the lung region of X-rays, which is exactly what a medical expert would focus on. As discussed in Chapter 5, CNNs are specially designed to handle images and extract features via supervised learning. It is not just robust but is highly scalable and can be improved over time. Due to the aforementioned reasons, we conclude that CNNs are the best models to work on a classification problem using X-ray images.

6.2. Future Outlook and improvements

This project can be further improved by exploring advanced techniques and addressing crucial areas of model performance and efficiency.

- We have succeeded to fine-tune only one model (KNN with grid search). However, we suggest further hyperparameter optimization for classical models of interest.
- Considering experimenting with more advanced ensemble techniques beyond Voting classifier, may provide better results. Although we have tried to run Stacking, this code was not finished since it took more than 2 days.
- Performing a thorough error analysis to understand where models are making mistakes can lead to better conclusions. This can provide insights into which features are problematic and help with improvements.
- For dataset with class imbalance, techniques like SMOTE can generate synthetic samples to balance the class distribution. This may potentially improve model performances on minority of classes.
- To reduce training times and allow for the exploration of more complex models and larger datasets, we recommend leveraging more powerful hardware with GPUs or utilizing GoogleColab's TPU.
- Improving CNN performance may include using deeper architectures, and optimizing the models by using different optimizers (Adam, AdaDelta,…) and by changing their parameters.
- Interpretability of models is of great value when pitching a business idea such as this one. It helps clients understand how the model functions and gain trust in the model for future use. Although we did achieve acceptable results with Saliency maps, one could dive deeper and inspect how Grad-CAM and other interpretability tools along with heat map's function, and go over random images in the entire dataset to be sure if the model considers the right features for class prediction.

7. References

[1] M. Ciotti, M. Ciccozzi, A. Terrinoni, W.-C. Jiang, C.-B. Wang, S. Bernardini, The COVID-19 pandemic, Crit. Rev. Clin. Lab. Sci. 57 (2020) 365–388. https://doi.org/10.1080/10408363.2020.1783198.

[2] A. Spena, L. Palombi, M. Carestia, V.A. Spena, F. Biso, SARS-CoV-2 Survival on Surfaces. Measurements Optimisation for an Enthalpy-Based Assessment of the Risk, Int. J. Environ. Res. Public. Health 20 (2023) 6169. https://doi.org/10.3390/ijerph20126169.

[3] M.A. Matthay, A. Leligdowicz, K.D. Liu, Biological Mechanisms of COVID-19 Acute Respiratory Distress Syndrome, Am. J. Respir. Crit. Care Med. 202 (2020) 1489–1491. https://doi.org/10.1164/rccm.202009-3629ED.

[4] L. Gattinoni, D. Chiumello, S. Rossi, COVID-19 pneumonia: ARDS or not?, Crit. Care 24 (2020) 154. https://doi.org/10.1186/s13054-020-02880-z.

[5] J. Aranda, I. Oriol, M. Martín, L. Feria, N. Vázquez, N. Rhyman, E. Vall-Llosera, N. Pallarés, A. Coloma, M. Pestaña, J. Loureiro, E. Güell, B. Borjabad, E. León, E. Franz, A. Domènech, S. Pintado, A. Contra, M. del S. Cortés, I. Chivite, R. Clivillé, M. Vacas, L.M. Ceresuela, J. Carratalà, Long-term impact of COVID-19 associated acute respiratory distress syndrome, J. Infect. 83 (2021) 581–588. https://doi.org/10.1016/j.jinf.2021.08.018.

[6] C. Scelfo, M. Fontana, E. Casalini, F. Menzella, R. Piro, A. Zerbini, L. Spaggiari, L. Ghidorsi, G. Ghidoni, N.C. Facciolongo, A Dangerous Consequence of the Recent Pandemic: Early Lung Fibrosis Following COVID-19 Pneumonia – Case Reports, Ther. Clin. Risk Manag. 16 (2020) 1039–1046. https://doi.org/10.2147/TCRM.S275779.

[7] J.-M. Anaya, M. Rojas, M.L. Salinas, Y. Rodríguez, G. Roa, M. Lozano, M. Rodríguez-Jiménez, N. Montoya, E. Zapata, D.M. Monsalve, Y. Acosta-Ampudia, C. Ramírez-Santana, Post-COVID syndrome. A case series and comprehensive review, Autoimmun. Rev. 20 (2021) 102947. https://doi.org/10.1016/j.autrev.2021.102947.

[8] O. Filchakova, D. Dossym, A. Ilyas, T. Kuanysheva, A. Abdizhamil, R. Bukasov, Review of COVID-19 testing and diagnostic methods, Talanta 244 (2022) 123409. https://doi.org/10.1016/j.talanta.2022.123409.

[9] V. Nikolaou, S. Massaro, M. Fakhimi, L. Stergioulas, W. Garn, COVID-19 diagnosis from chest x-rays: developing a simple, fast, and accurate neural network, Health Inf. Sci. Syst. 9 (2021) 36. https://doi.org/10.1007/s13755-021-00166-4.

[10] A.A. Borkowski, N.A. Viswanadhan, L.B. Thomas, R.D. Guzman, L.A. Deland, S.M. Mastorides, Using Artificial Intelligence for COVID-19 Chest X-ray Diagnosis, Fed. Pract. 37 (2020) 398–404. https://doi.org/10.12788/fp.0045.

[11] https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data

[12] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0265949

[13] https://scikit-learn.org/stable/modules/ensemble.html#random-forests-and-other-randomized-tree-ensembles

[14] https://scikit-learn.org/stable/modules/ensemble.html#forest

[15] https://xgboost.readthedocs.io/en/stable/tutorials/model.html

8. Appendices

Code snippets with detailed technical information on Github

may24_bds_int_covid_xray/notebooks/drafts and exploration/