**COVID-19 Radiography Analysis**

**Final report**

# Executive Summary

## 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.

## 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:
  + Grad-CAM 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.

## Project implications

The deployment of a CNN model equipped with Grad-CAM 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.

## Recommendations

* Implement the CNN Model: Given its superior accuracy and interpretability, the CNN model with Grad-CAM should be implemented for X\_ray image classification in clinical environments.
* 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.

# Introduction

## 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 (DL) 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.

## Problems and opportunity

### **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 an 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.

### **Opportunity**

## Objective 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.

# 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.

## 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.

## 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.

### **Image pre-processing for Classification models**

#### 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.

#### 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).

#### 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. The roi in grayscale was used in the subsequent steps.

#### 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.

#### 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.

#### 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.

### **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:

#### 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.

#### 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.

#### 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.

#### 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. 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 tot he original image(extreme left) |

#### 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.

#### Data splitting

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

## 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.

### **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 following table provides the parameters chosen for each model

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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.

### **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 3.1.

|  |
| --- |
|  |
| **Figure 3.1.** 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.

|  |
| --- |
|  |
| **Figure 3.2.** 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. |

## 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 is Gradient-weighted Class Activation Mapping (Grad-CAM)

## 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 visualise 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.

## 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).

* 1. **Results of classical models (RF, Bagging, XGBoost) and interpretation**
  2. **Results of the deep learning model (CNN) and interpretation**
  3. **Comparison of model performances**

1. **Insights and Discussion**
   1. **Why deep learning (CNN) was chosen despite high accuracies in classical models**
   2. **Importance of model interpretability and robustness (e.g., using Grad-CAM for X-rays)**
   3. **Potential impacts**
2. **Conclusion**
   1. **Summary of findings**
   2. **Future work and improvements**
3. **References**

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

1. **Appendices**
   1. **Code snippets with detailed technical information**