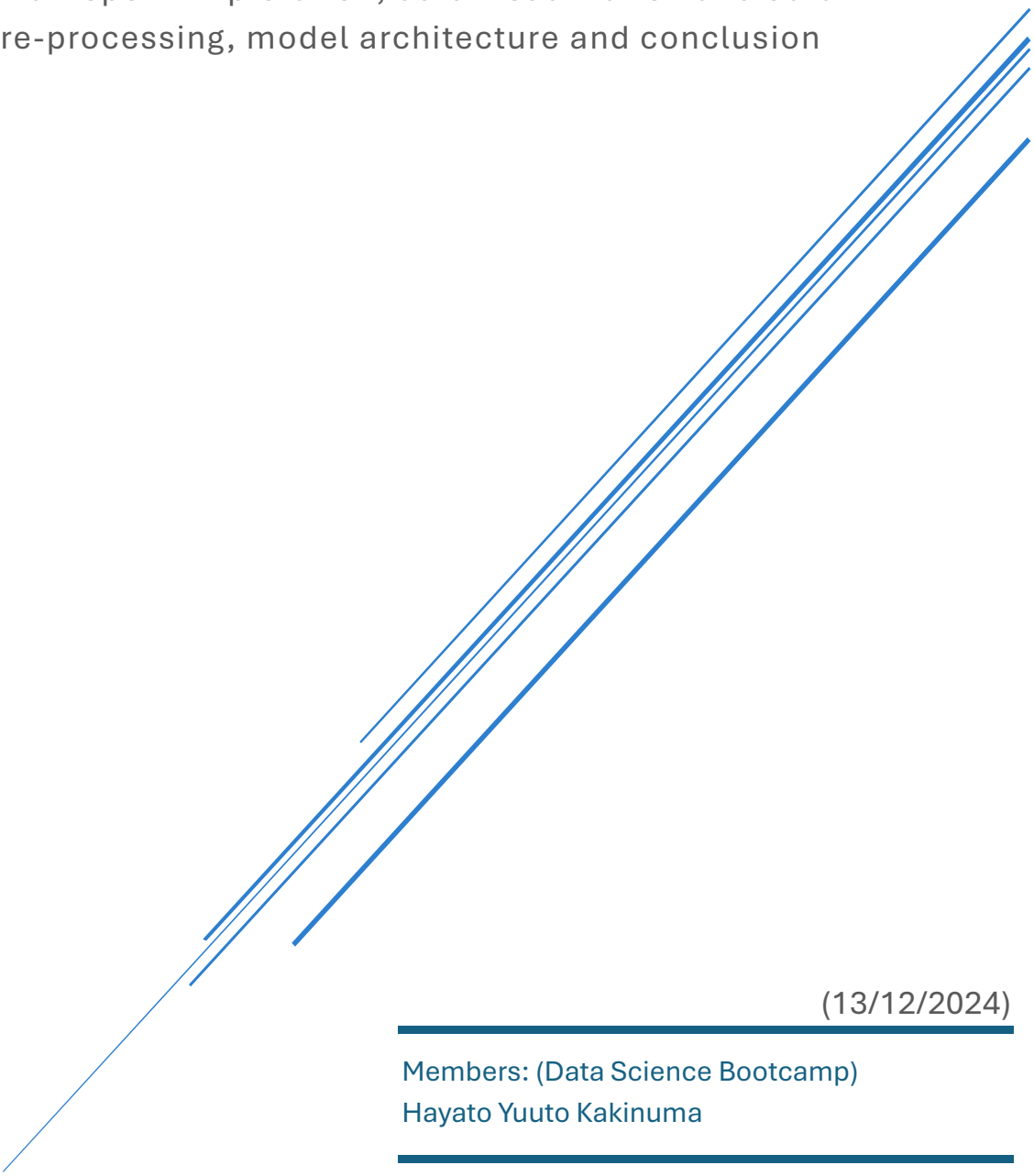
Project: Exploring the Feasibility of Using Chest X-ray Data for Diagnosis of COVID-19 Patients

Final report: Exploration, data visualization and data pre-processing, model architecture and conclusion



(13/12/2024)

Members: (Data Science Bootcamp)
Hayato Yuuto Kakinuma

Table of Contents

Executive Summary

Background

The COVID-19 pandemic has highlighted the need for efficient and scalable diagnostic tools. While RT-PCR tests are the gold standard, they can be time-consuming and resource-intensive. This project explores the feasibility of using chest X-ray data as a supplementary diagnostic approach to distinguish COVID-19 patients from individuals with other respiratory conditions or healthy individuals.

Objective

To develop and evaluate models capable of diagnosing COVID-19 using X-ray images. The project employs:

1. **Machine Learning (ML):** XGBoost for feature-based classification.
2. **Deep Learning (DL):** Convolutional Neural Networks (CNNs) for end-to-end image classification.

Methods and Data

The analysis is based on the COVID-QU-Ex dataset, comprising 33,920 chest X-rays categorized into:

- COVID-19 positive cases
- Non-COVID infections
- Normal/healthy individuals

Key steps included:

- Image preprocessing and augmentation to enhance dataset diversity and model robustness.
- Model implementation, training, and comparative evaluation of ML and DL approaches.

Key Insights

Initial results highlight the effectiveness of CNNs in capturing complex patterns in X-ray images, while XGBoost provides competitive results with extracted features. This dual approach underscores the potential of integrating traditional ML and modern DL techniques for improved diagnostic accuracy.

Impact and Relevance

By leveraging widely available X-ray technology, this work contributes to developing rapid, scalable, and interpretable diagnostic solutions, particularly valuable in resource-limited healthcare settings.

Introduction:

1. Context

The COVID-19 pandemic has posed a significant global health crisis, overwhelming healthcare systems and prompting the development of rapid diagnostic tools. Traditional methods, such as PCR tests, although effective, can be resource-intensive and time-consuming. Medical imaging, specifically chest X-rays, has emerged as a potential supplementary diagnostic tool. X-rays are cost-effective, widely available, and may reveal characteristic lung changes associated with COVID-19, providing a faster diagnostic alternative.

This project focuses on exploring the feasibility of using chest X-ray data to distinguish COVID-19 patients from individuals with other respiratory conditions or healthy subjects using deep learning and machine learning techniques.

2. Objectives

The primary objective of this project is to develop a diagnostic model that can:

a) Differentiate between COVID-19 positive, non-COVID lung infections (such as pneumonia), and healthy patients using chest X-ray images.

b) Improve the interpretability of the model's predictions through visualizations and explainability techniques, ensuring that decisions are made based on relevant patterns in the X-ray data.

c) Provide a comparative analysis of traditional machine learning techniques and deep learning-based convolutional neural networks (CNNs) to find the most effective diagnostic tool.

Additionally, the project aims to provide a robust, scalable solution by leveraging data augmentation, pre-processing techniques, and performance evaluation to ensure accuracy and generalizability.

3. Framework

Data Collection:
The dataset used for this project is the **COVID-QU-Ex dataset** from Kaggle, which consists of **33,920 chest X-ray images**. The images are divided into three main categories:

- **11,263 Non-COVID infections** (including viral or bacterial pneumonia)
- **10,701 Normal/Healthy patients**
- **11,956 COVID-19 positive patients**

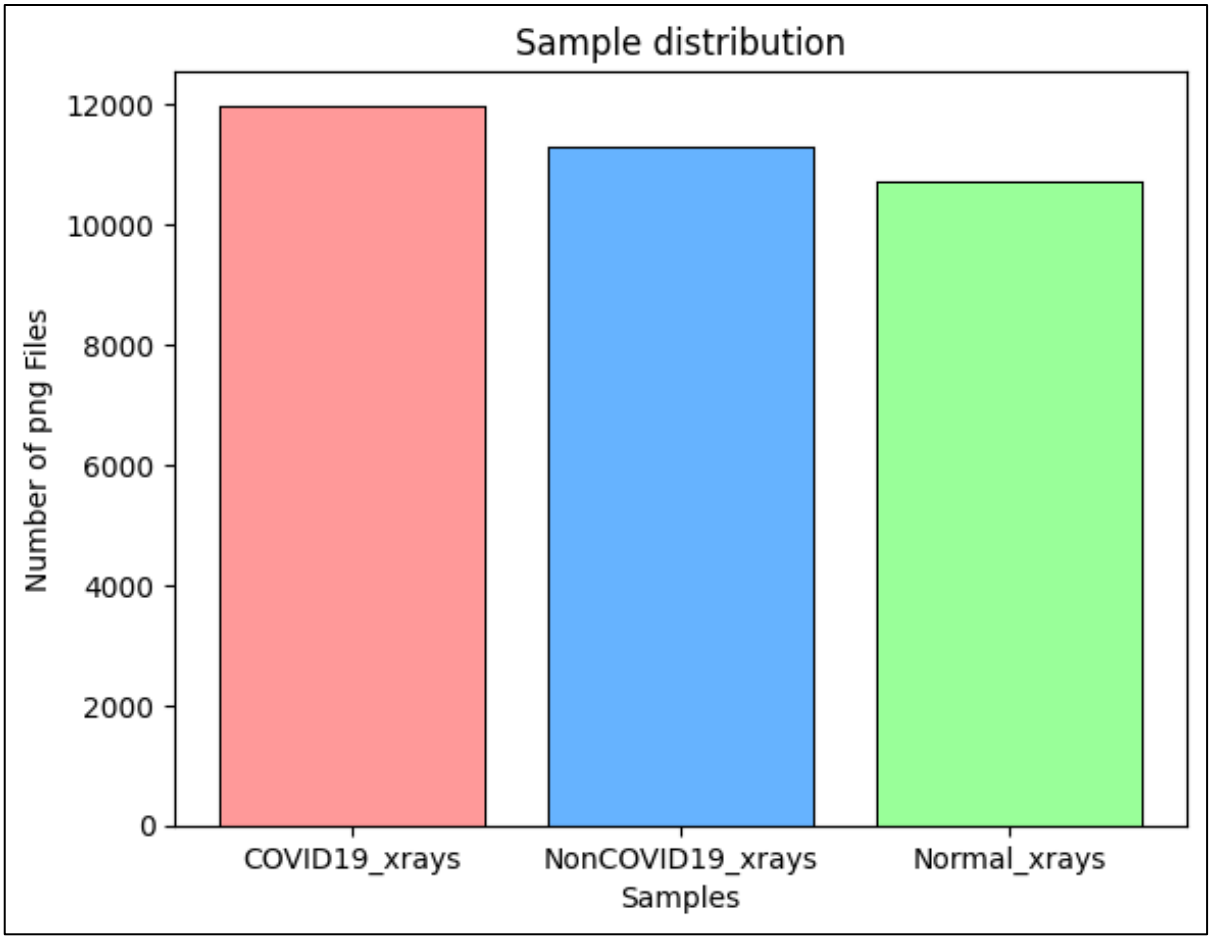


Figure 1.Distribution of X-ray Samples:
COVID-19 Patients, Non-COVID Viral Pneumonia (Non-COVID) Patients, and Healthy Individuals (Normal)

Data Preprocessing will involve:

- **Normalising, resizing, and applying colour mapping** to standardize the images for input into machine learning models. This process may involve contrast enhancement, noise reduction, and cropping to improve the clarity and focus of the images.

- **Data augmentation** techniques, such as rotation, flipping, zooming, and colour mapping variations, are used to increase the diversity of the training images. These techniques help reduce overfitting and enhance the model's robustness by introducing different visual perspectives and colour representations.

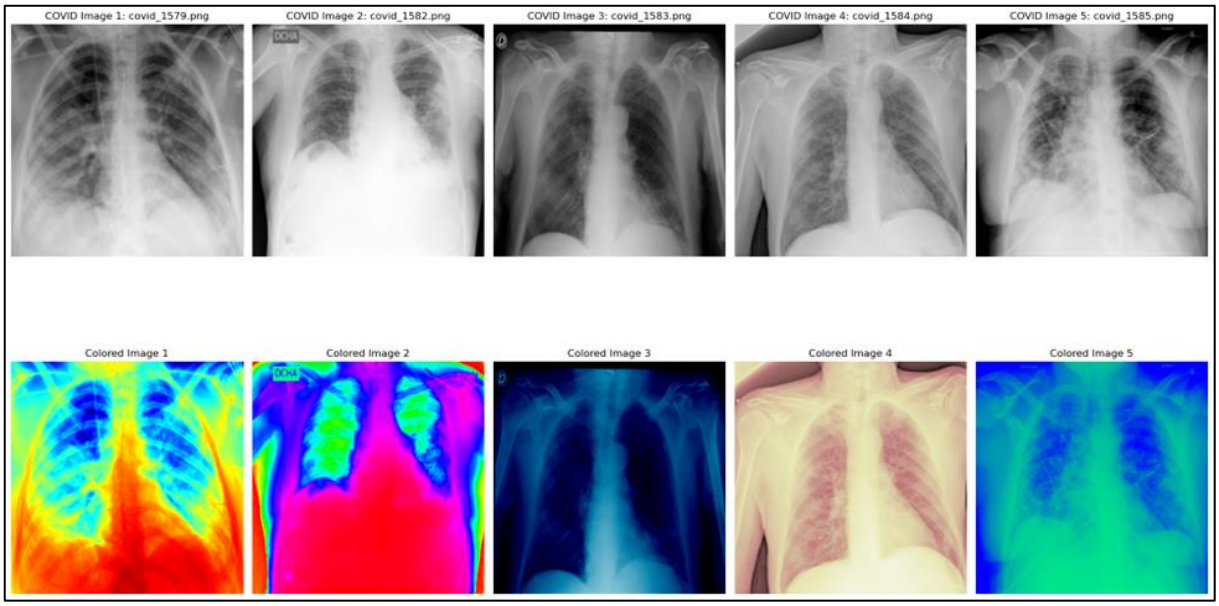


Figure 2. Sample X-ray images from COVID-19 positive patients (top five).
Different colour maps: COLOURMAP_JET, COLOURMAP_HSV, COLOURMAP_OCEAN, COLOURMAP_PINK, COLOURMAP_WINTER (bottom five).

4. Relevance

The relevance of this project lies in its potential to assist healthcare systems in diagnosing COVID-19 quickly and accurately using a readily available diagnostic tool—chest X-rays. This would allow medical professionals to assess patients even in resource-limited settings where PCR tests or more advanced imaging techniques are not readily available.

By leveraging machine learning models, the project aims to:

a) **Speed up the diagnostic process** by automating the detection of COVID-19 patterns in chest X-ray images.

b) **Improve diagnostic accuracy** by incorporating deep learning models trained on a large dataset of X-rays, fine-tuned with transfer learning.

c) **Contribute to explainability and trust** in AI-driven diagnoses through techniques like Grad-CAM and SHAP, which will provide visual and statistical explanations of the model's predictions.

5. Preprocessing and Feature Engineering

The preprocessing phase of this project is critical to ensuring that the data fed into the models is clean, consistent, and informative. The following steps are part of the preprocessing strategy:

a) Image Preprocessing

The X-ray images, initially provided in PNG format, were first converted to greyscale to ensure uniformity and to facilitate subsequent image analysis. Each lung X-ray image was accompanied by its corresponding ground truth mask, essential for segmentation and accurate feature extraction. To maintain consistency across the dataset, it was necessary to ensure that the dimensions of the images and their respective masks were perfectly aligned. After verifying size compatibility, the images were further processed by converting them into 8-bit unsigned integers, optimizing them for computational efficiency and reducing the overall file size while preserving critical pixel intensity information for further analysis.

Each image will be resized to a standard dimension of 224x224 pixels. This resizing ensures that all input images match the expected dimensions of popular convolutional neural networks (CNNs). Resizing will maintain the aspect ratio where possible or apply interpolation methods to prevent significant distortion.

b) Data Augmentation

Data augmentation will be used to artificially expand the size and diversity of a training dataset by applying various transformations to the existing data. This process helps improve the performance and generalization of machine learning models. It will include the methods like:

- **Rotating**, **flipping**, and **zooming** images to create variations.
- **Brightness adjustment** to simulate different X-ray conditions.

c) Feature Extraction

In addition to CNNs, traditional machine learning techniques will be tested. For these methods, feature extraction is crucial. Techniques like histogram of oriented gradients (HOG) and grey-level co-occurrence matrices (GLCM) will be used to extract key features from the images. These features will serve as input for traditional classifiers like SVM or Random Forest.

6. Visualisations and Statistics

The KDE plots below reveal that while there is overlap between the categories; normal X-rays are distinguishable by lower opacity levels, while COVID-19 and non-COVID viral pneumonia X-rays share similarities, though COVID-19 X-rays exhibit a slightly broader range of opacity levels, potentially due to more severe or varied lung changes (Figure 3).

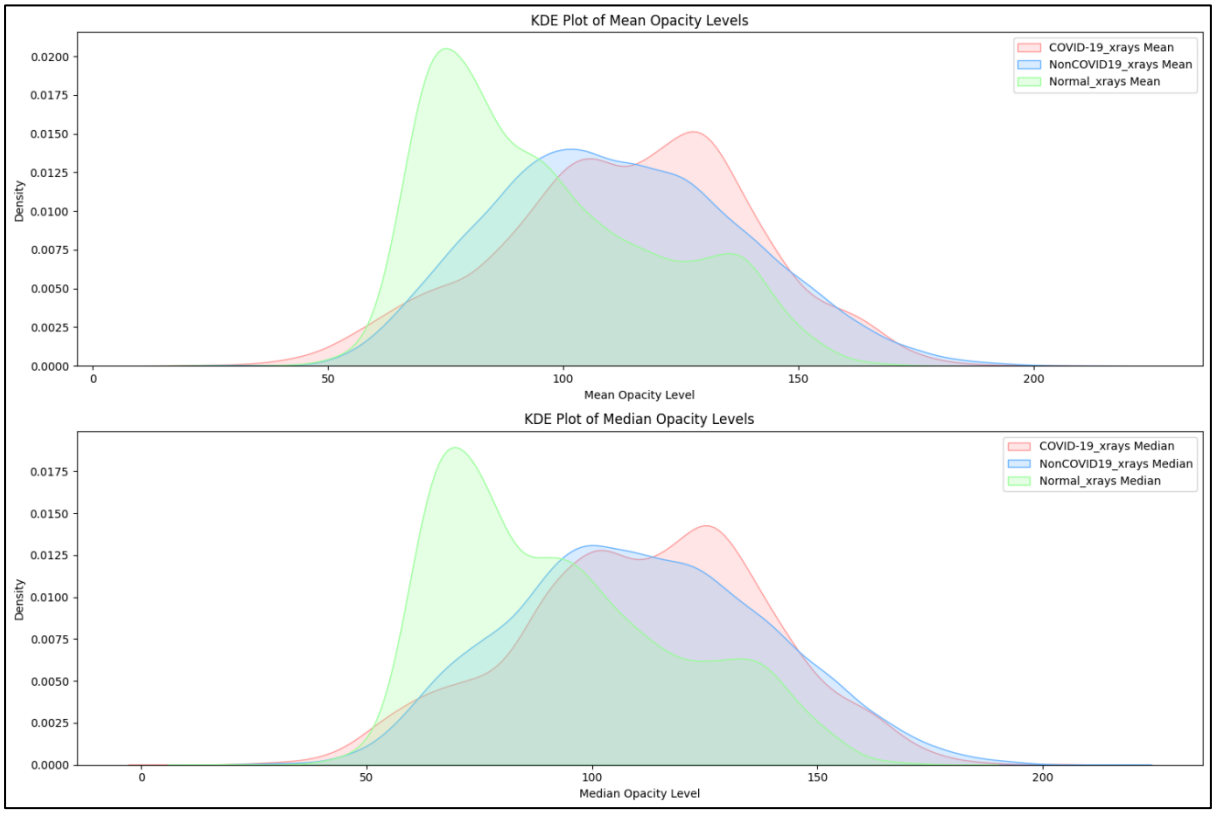


Figure 3. KDE Plot of Mean and Median Opacity Levels: COVID-19 Patients, Non-COVID Patients, and Normal X-rays.

Looking into their standard deviations (STD):

- COVID-19 and Non-COVID-19 viral pneumonia X-rays exhibit higher standard deviations in opacity levels, indicating greater variability within the lungs due to disease, which may reflect pathological features like consolidations and ground-glass opacities.
- Normal X-rays have a significantly lower standard deviation, reflecting the more homogenous nature of healthy lungs with fewer opacities and clearer structures.
- The numerous outliers in the diseased categories (COVID-19 and Non-COVID-19) suggest that some cases exhibit extreme opacity variations, which might correlate with severe disease manifestations.

The box plot highlights the more consistent and lower opacity variability in healthy lungs compared to the increased and more variable opacity seen in both COVID-19 and other viral pneumonia cases (figure 4).

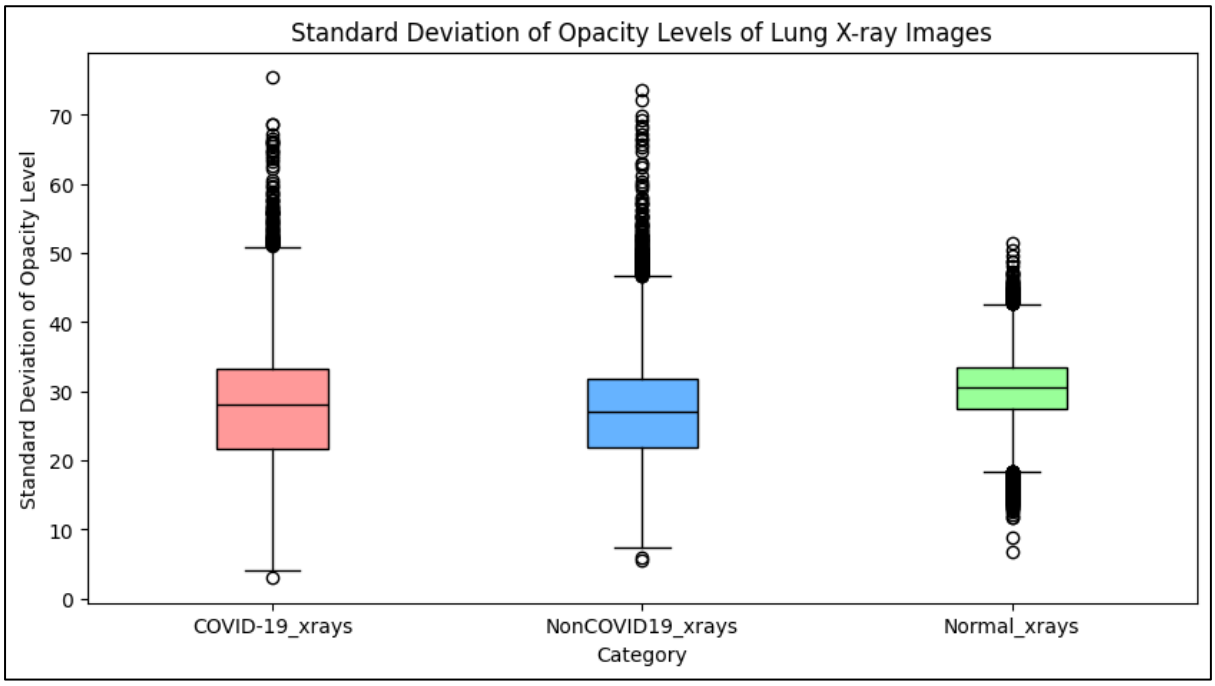


Figure 4. Box plot of STD of Opacity Levels: COVID-19 Patients, Non-COVID Patients, and Normal X-rays

In a scatter plot, we can better observe the relationship between the standard deviation of opacity level and the median opacity level of X-ray images (figure 5). We can visualise better how narrow the normal X-ray STD and median range compared to diseased X-rays.

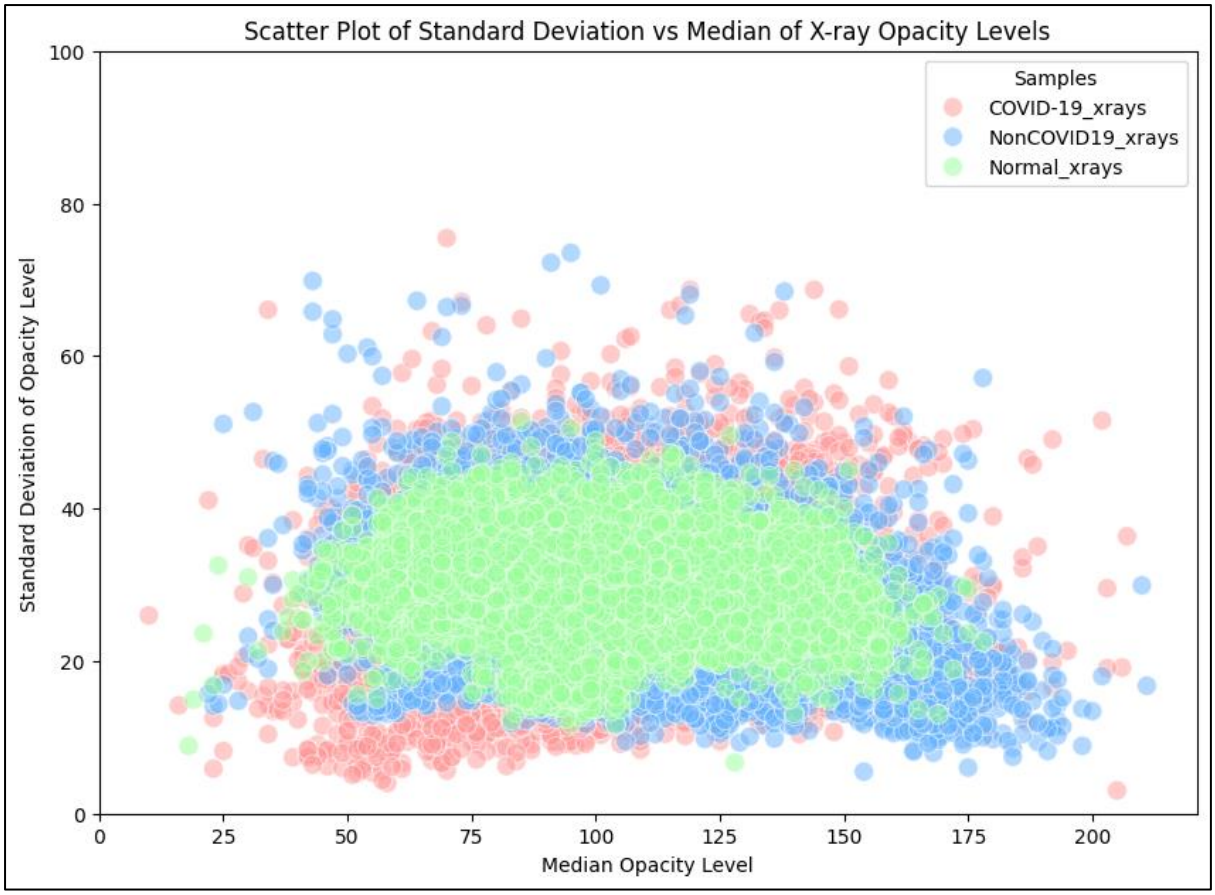


Figure 5. Scatter plot of STD against Median Opacity Levels: : COVID-19 Patients, Non-COVID Patients, and Normal X-rays

Modelling Report

1. Classification of the problem

Type of Machine Learning Problem

This project addresses a **classification problem**. Specifically, **multi-class image classification problem** - the goal is to classify chest X-ray images into three distinct categories:

- **COVID-19 Positive**
- **Non-COVID Infections** (such as viral or bacterial pneumonia)
- **Normal/Healthy Individuals**

The use of deep learning models like ResNet-50 and traditional machine learning models such as XGBoost clearly frames this as a supervised learning classification task.

Related Task

The project relates to **medical image classification** and aid in **disease diagnosis**. The task aims to develop models that can assist in diagnosing COVID-19 by identifying characteristic patterns in chest X-rays. This approach can serve as a supplementary diagnostic tool to the standard RT-PCR tests, providing rapid screening, especially in resource-limited healthcare settings.

Main Performance Metric

The primary performance metric used for evaluating and comparing models in this project is the **F1-score**. The F1-score is the harmonic mean of precision and recall and is particularly suitable for this project due to the following reasons:

1. **Imbalanced Classes**: The dataset may have an uneven distribution of COVID-19, non-COVID, and normal X-rays. The F1-score helps balance the consideration of both false positives (precision) and false negatives (recall).
2. **Medical Context**: In diagnosing diseases, both precision (avoiding false positives) and recall (avoiding false negatives) are critical, as misdiagnosis can lead to serious consequences.

3. **Comprehensive Assessment**: Unlike accuracy, the F1-score provides a more nuanced view of model performance when dealing with multiple classes.

Additional Performance Metrics

In addition to the F1-score, the following metrics were also considered:

1. **Accuracy**: The proportion of correctly classified instances over the total number of instances. Provides a general measure of performance. However, in the presence of class imbalance, it may not be fully reliable as a standalone metric.
2. **Precision (Positive Predictive Value)**: The proportion of correctly identified positive cases out of all predicted positive cases. Important for ensuring that a COVID-19 diagnosis is correct and minimising false positives.
3. **Recall (Sensitivity or True Positive Rate)**: The proportion of actual positive cases correctly identified by the model. Critical for ensuring that COVID-19 cases are not missed (false negatives), which is essential in medical diagnostics.
4. **Confusion Matrix**: Offers a detailed breakdown of true positives, false positives, true negatives, and false negatives for each class, aiding in model interpretability.

2. Model choice and optimization

Algorithms Tried

1. **Convolutional Neural Network (CNN)**:
    - **TensorFlow with Transfer-Learning ResNet-50**: Initially attempted with pre-trained weights on ImageNet to expedite training. Incompatibility with Colab's native libraries prevented full utilisation of the NVIDIA L4 GPU.
    - **PyTorch with Non-Transfer Learning ResNet-50**: Switched to PyTorch for better GPU support. The non-transfer learning approach allowed training the model from scratch, avoiding dependency conflicts.
2. **Random Forest**: A bagging ensemble method known for handling high-dimensional data and reducing overfitting.

3. **Support Vector Machine (SVM)**: Effective for creating decision boundaries in feature-based image classification.
4. **XGBoost**: A gradient-boosting method known for its efficiency and performance with structured data.
5. **k-Nearest Neighbours (k-NN)**: A simple instance-based learning algorithm used for baseline comparison.

Selected Models and Rationale

1. **PyTorch CNN (ResNet-50)**:

- Rich Feature Learning: ResNet-50 captures complex patterns in X-rays without manual feature extraction.
- Compatibility: Switching to PyTorch allowed full utilisation of the NVIDIA L4 GPU in Colab.
- Flexibility: Training from scratch (non-transfer learning) helped avoid library conflicts and provided full control over the learning process.

2. **XGBoost**:

- Performance: Known for handling tabular data effectively.
- Interpretability: Easier to understand and interpret compared to deep learning models.

Parameter Optimization Techniques

1. **Grid Search** - Applied to Random Forest, SVM, XGBoost, and k-NN to identify optimal hyperparameters. Key parameters tuned:

- **Random Forest**: Number of trees, max depth.
- **SVM**: Kernel type, regularisation parameter.
- **XGBoost**: Learning rate, number of estimators, max depth.
- **k-NN**: Number of neighbours, distance metric.

2. **Cross-Validation** - 5-Fold Cross-Validation was used to ensure robustness and reduce overfitting across all models.

Evaluation Metrics

1. **Classification Report**: Provides detailed performance metrics.
   - **Precision**: The proportion of true positives among all predicted positives.
   - **Recall (Sensitivity)**: The proportion of true positives among actual positives.
   - **F1-Score**: The harmonic mean of precision and recall, balancing the trade-off between false positives and false negatives.
2. **Confusion Matrix**: Offers a detailed breakdown of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Useful for identifying which classes are misclassified and understanding model performance in detail.

Advanced Models Tested

1. **Bagging**:
   - **Random Forest** was tested to evaluate ensemble methods that reduce variance by combining multiple decision trees.
2. **Boosting**:
   - **XGBoost** was implemented to leverage sequential learning and handle class imbalance effectively.
3. **Deep Learning**:
   - **ResNet-50** CNN model trained using PyTorch for direct image classification.

Why These Methods?

- **Deep Learning (CNN)**: Ideal for automatically learning complex features from images.
- **Boosting (XGBoost)**: Effective for structured data and feature-based methods.
- **Bagging (Random Forest)**: Provides a reliable ensemble baseline.
- **Traditional ML (SVM, k-NN)**: Useful for comparative analysis and baseline performance.

3. Interpretation of Results

Error Analysis

To better understand model performance, we conducted a thorough error analysis using the confusion matrix and classification reports for each of the models. This analysis helped identify specific patterns in the errors, such as:

1. **Misclassifications Between COVID-19 and Non-COVID Infections**: Errors were common between these two classes due to the similarity in lung patterns (e.g., ground-glass opacities and consolidations). This insight led to tweaking the class weights, additional data augmentations and adjusting the learning rate to enhance model generalisation and improve the model's ability to distinguish between these categories.
2. **False Negatives in COVID-19 Detection**: False negatives (COVID-19 cases classified as normal or non-COVID) were particularly concerning. To mitigate this, we focused on balancing the dataset and adjusting class weights in the CNN and XGBoost models. This improved the model's sensitivity (recall) for COVID-19 detection.

Model Improvement from Error Analysis

The insights gained from error analysis contributed to the following improvements:

1. **Enhanced Data Augmentation** - Applied horizontal flipping, brightness adjustment, and slight rotations to increase dataset diversity. This helped reduce overfitting and improved the generalisation of the **CNN model**.
2. **Class Weight Adjustments** - In the CNN and XGBoost models, we adjusted class weights to penalise misclassification of minority classes (e.g., COVID-19 cases). This reduced false negatives and improved overall recall for COVID-19 detection.
3. **Feature Extraction Refinements** - For traditional models (SVM, Random Forest, XGBoost, k-NN), refining HOG feature extraction parameters (e.g., cell size and block size) improved feature quality and, subsequently, model accuracy.

Interpretability Techniques

To interpret the models and enhance transparency, we employed the following techniques:

1. **Grad-CAM (Gradient-weighted Class Activation Mapping)**:

For CNN (ResNet-50), Grad-CAM was used to generate heatmaps highlighting regions in the X-ray images that contributed most to the model's predictions (Figure 6). This allowed us to verify that the model focused on relevant lung regions (e.g., areas showing opacities or consolidations). Insights from Grad-CAM helped confirm that the CNN was learning meaningful patterns rather than relying on spurious correlations.

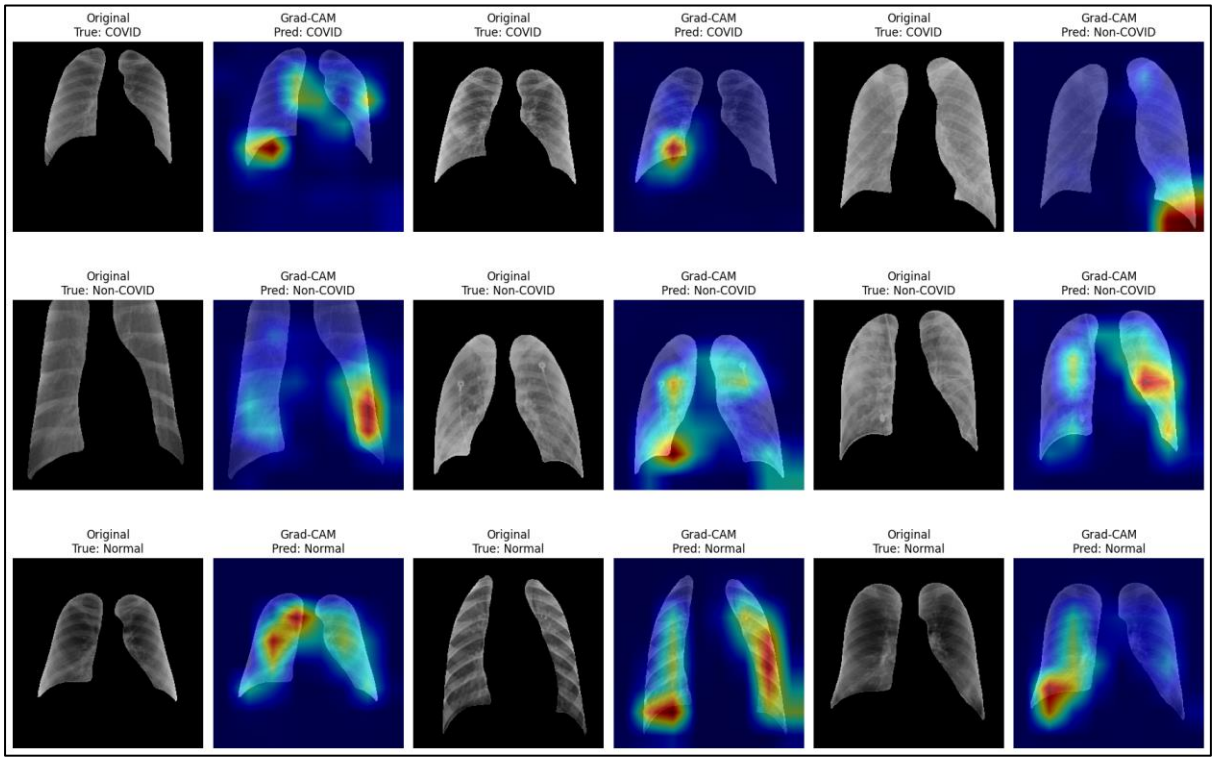


Figure 6. Grad-CAM Visualisation for CNN ResNet-50 Model Predictions on Chest X-rays.
This figure shows Grad-CAM visualisations of the CNN ResNet-50 model's predictions on chest X-ray images. The heatmaps highlight the regions of the lungs that contributed most to the model's decisions. The rows represent three categories: COVID-19, Non-COVID infections, and Normal cases. Each pair of images consists of the original X-ray (left) and the corresponding Grad-CAM heatmap (right). The heatmaps demonstrate that the model focuses on relevant areas of lung abnormalities, such as opacities and consolidations, when making predictions.

2. **Confusion Matrix** is used across all models to identify specific misclassification patterns and quantify the types of errors (false positives, false negatives). The matrix (figure 7) shows that the model correctly classified:

    - 1,353 COVID-19 cases, with 218 misclassified as Non-COVID and 199 as Normal.
    - 1,334 Non-COVID cases, with 92 misclassified as COVID-19 and 254 as Normal.
    - 1,476 Normal cases, with 49 misclassified as COVID-19 and 113 as Non-COVID

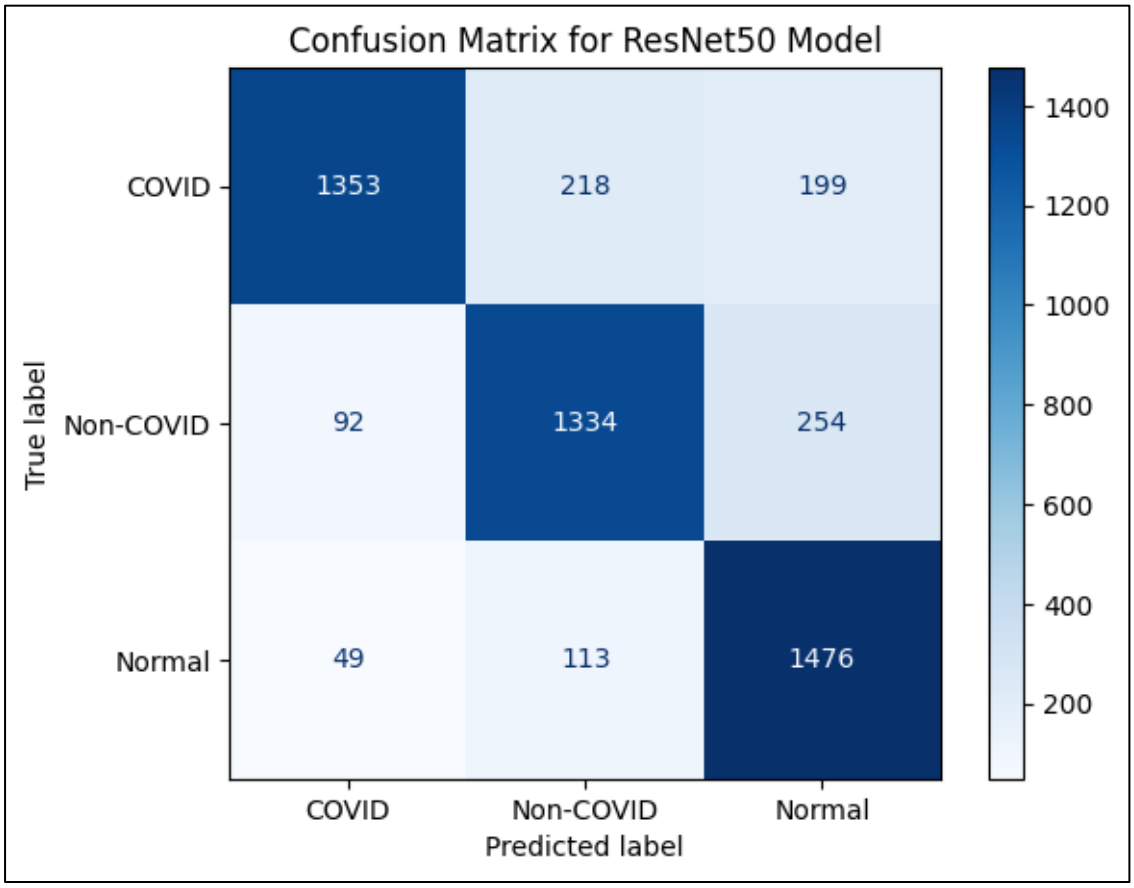


Figure 7. Confusion Matrix for ResNet-50 CNN Model Predictions.
This confusion matrix illustrates the performance of the ResNet-50 CNN model on the test dataset for classifying chest X-rays into COVID-19, Non-COVID infections, and Normal cases. The rows represent the true labels, and the columns represent the predicted labels.

3. **Classification Report** provided detailed metrics for precision, recall, and F1-score, helping to identify weaknesses in model performance.

Impact on Performance

The following approaches significantly improved model performance:

1. **Switching to PyTorch for CNN**:
    - Enabled full utilisation of the NVIDIA L4 GPU in Colab, improving training speed and model accuracy.
2. **Grad-CAM Visualisations**:
    - Helped refine the CNN model by ensuring it focused on appropriate lung regions, leading to more reliable predictions.
3. **Data Augmentation and Class Balancing**:
    - Reduced overfitting and improved recall, especially for detecting COVID-19 cases.

What Did Not Generate Significant Improvement

1. **Transfer Learning (TensorFlow ResNet-50)**:
    - Due to compatibility issues with Colab's GPU libraries, transfer learning with TensorFlow ResNet-50 did not perform optimally.
    - Switching to PyTorch with non-transfer learning proved more effective.
2. **k-Nearest Neighbours (k-NN)**:
    - The k-NN model did not perform well on high-dimensional HOG features and was computationally expensive.
    - It was retained as a baseline but did not contribute significantly to final performance.

Result and Conclusion

1. Difficulties encountered during the project

Main Scientific Obstacles

The primary scientific challenge was the difficulty in distinguishing between COVID-19 and non-COVID lung infections due to their similar radiological patterns (perforations or scar formations). This required extensive data augmentation, careful model tuning, and interpretability techniques to improve performance.

Several tasks took longer than anticipated due to a combination of technical and logistical challenges:

1. **Model Training and Optimization**: Training deep learning models, particularly ResNet-50, involved multiple iterations and fine-tuning, which was slowed down by GPU availability issues and Colab's limitations. Computational limitations on the Colab platform necessitated the use of 1/10 of the dataset (i.e., XGBoost) for training due to memory constraints.
2. **Trial-and-Error Learning in Colab**: Without an official tutorial from Datascientest on using Colab Pro, there was a steep learning curve. Each test run incurred costs as computing units were purchased with personal funds, adding pressure to optimise runs.

Datasets

1. **Acquisition and Transfer**: The dataset was 1.2GB, and transferring it from a personal laptop to Colab took considerable time. Initially, the dataset was uploaded to Google Drive and mounted to Colab, but this approach led to significant I/O bottlenecks, further delaying model training.
2. **Solution**: The breakthrough came when the dataset was zipped and uploaded directly to Colab. Unzipping it in Colab reduced the I/O bottleneck and allowed full utilisation of the GPU (applied in the final CNN model).

Technical/Theoretical Skills

1. **Colab Learning Curve**: The lack of a GPU on the personal laptop necessitated the use of Colab Pro. Learning to navigate Colab's environment, manage libraries, and optimise GPU use involved trial-and-error testing, which was costly due to paid computing units.
2. **Library Compatibility**: Libraries used in VSCode were not always available in Colab. This required adjustments to the code and, in some cases, switching between different frameworks (e.g., TensorFlow to PyTorch).
3. **Software library switching** - The initial use of TensorFlow with transfer-learning ResNet-50 was hindered by compatibility issues with Colab's NVIDIA L4 GPU. Switching to PyTorch and training ResNet-50 from scratch improved performance but required additional time to adapt the codebase.

IT Challenges

1. **Lack of Personal GPU**: The absence of a GPU on the personal laptop made local training infeasible. Relying on Colab Pro introduced challenges, including learning to manage cloud-based resources efficiently and dealing with trial-and-error costs.
2. **I/O Bottlenecks**: Initially mounting the dataset from Google Drive caused severe I/O bottlenecks, slowing down training. Discovering the method of zipping and unzipping the dataset in Colab was crucial for overcoming these bottlenecks and utilising the GPU effectively.

Other Challenges

1. **Cost Management**: Each test run consumed paid computing units, adding financial pressure to optimise processes quickly.
2. **Library Dependencies**: Ensuring the correct libraries were installed and compatible with Colab's environment added troubleshooting time.

2. Reporting Results

Throughout this project, I independently handled all aspects, from data preprocessing and model selection to error analysis and interpretation. We were a group of three initially up to the submission of the first report which was derived from my initial data mining and data analysis.

My primary contribution was implementing and evaluating multiple machine learning and deep learning models to classify chest X-ray images into COVID-19, non-COVID infections, and normal categories. Specifically, I explored Convolutional Neural Networks (CNNs), Random Forest, Support Vector Machines (SVM), XGBoost, and k-Nearest Neighbours/k-NN (see Table 1). Initially, I attempted using TensorFlow with a transfer-learning ResNet-50 model. However, due to compatibility issues with Colab's native libraries, which hindered the full utilisation of the NVIDIA L4 GPU, I switched to PyTorch and employed a non-transfer learning approach with ResNet-50. This transition allowed me to effectively train the model and achieve optimal performance.

Table 1. Observed validation accuracy and F1-score for each model in the initial pipeline. Simultaneous run with 1/10 of the dataset.

Model	Accuracy	F1-Score
CNN	70%	0.68
XGBoost	65%	0.64
Random Forest	63%	0.63
SVM	61%	0.61
k-NN	57%	0.56

In addition to model development, I designed a comprehensive image preprocessing pipeline that involved resizing images, applying lung masks, and normalising the data. To enhance model generalisation and reduce overfitting, I implemented various data augmentation techniques such as horizontal flipping, elastic transformation, contrast and brightness adjustments, and rotations. These efforts improved the robustness of the CNN model, which ultimately achieved a test accuracy of 81.82% and a macro-average F1-score of 82%. The CNN performed particularly well for the COVID and Normal classes, both achieving an F1-score of 83%, while the Non-COVID class achieved an F1-score of 80% (Table2).

Table 2. Classification report for the new CNN model (Pytorch ResNet50)

Class	Precision	Recall	F1-Score	Support
COVID	0.91	0.76	0.83	1770
Non-COVID	0.80	0.79	0.80	1680
Normal	0.77	0.90	0.83	1638

Among the traditional models, XGBoost showed the best performance, achieving a test accuracy of 80.22% and a macro-average F1-score of 80%. The detailed performance for XGBoost included an F1-score of 83% for the COVID class, 82% for the Non-COVID class, and 76% for the Normal class. This demonstrated that XGBoost was competitive with the CNN model, particularly in terms of precision for detecting the Non-COVID class (Table 3). The Random Forest, SVM, and k-NN models performed less effectively, achieving validation accuracies of 63%, 61%, and 57%, respectively, with corresponding F1-scores reflecting similar trends.

Table 3. Classification report for the XGBoost model with 1000 images from each class

Class	Precision	Recall	F1-Score	Support
COVID	0.81	0.85	0.83	150
Non-COVID	0.94	0.73	0.82	150
Normal	0.71	0.82	0.76	150

3. Conclusion

In this project, I successfully developed and evaluated multiple machine learning and deep learning models to classify chest X-ray images into COVID-19, non-COVID infections, and normal cases. Through detailed error analysis using confusion matrices and classification reports, I identified the key challenge of distinguishing between COVID-19 and non-COVID infections due to overlapping radiological features. To mitigate this, I balanced the dataset and adjusted class weights in the CNN and XGBoost models, reducing false negatives. The use of Grad-CAM provided critical insights into the CNN model's predictions, ensuring the model focused on relevant lung regions and enhancing its reliability for medical diagnosis.

Despite technical challenges, such as the lack of a GPU on my personal laptop, the steep learning curve with Colab Pro, and dataset transfer delays due to I/O bottlenecks,

I was able to overcome these obstacles. By zipping and unzipping the dataset directly in Colab, I fully utilised the NVIDIA L4 GPU to train and optimise my models effectively.

Among all the models tested, the PyTorch CNN ResNet-50 demonstrated the greatest potential, achieving strong performance and interpretability. Given more time, I would explore transfer learning and hyperparameter tuning for the PyTorch ResNet-50 model, similar to the approach attempted with the TensorFlow model, which faced compatibility issues. These enhancements could further improve accuracy and efficiency, making the model even more robust.

The resulting models are well-suited for deployment in rapid screening systems, telemedicine platforms, and triage systems to support COVID-19 diagnosis in resource-limited settings. This project underscores the power of deep learning for medical imaging and highlights practical solutions for overcoming computational and logistical challenges, paving the way for more accessible and efficient diagnostic tools.

4. Continuation of the Project

To improve model performance, expanding the dataset with more diverse X-rays from various populations and imaging conditions would enhance generalisation and reduce bias. Revisiting **transfer learning** with PyTorch using pre-trained models like **DenseNet** or **EfficientNet** could potentially boost accuracy and reduce training time. More sophisticated data augmentation, such as **GAN-based techniques**, can synthetically expand the dataset. Combining predictions from **CNN** and traditional models like **XGBoost** could leverage the strengths of both approaches through ensemble methods. Further hyperparameter tuning with **Bayesian Optimisation** or **Random Search** may refine model performance. Additionally, employing interpretability techniques such as **SHAP** or **LIME** would offer deeper insights into model predictions.

This project contributes to scientific knowledge by providing a comparative analysis of deep learning and traditional machine learning for COVID-19 diagnosis. It addresses real-world challenges like **GPU limitations**, **dataset transfer bottlenecks**, and **cost constraints**, offering practical solutions for researchers. The use of **Grad-CAM** highlights the importance of interpretability in medical AI, ensuring reliable and trustworthy model predictions. These insights can inform the development of more efficient and accessible diagnostic tools for resource-limited settings.

5. Bibliography

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp.5-32. Available at: https://doi.org/10.1023/A:1010933404324.

Chen, T. and Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, USA: ACM, pp.785-794. Available at: https://doi.org/10.1145/2939672.2939785.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, USA: IEEE, pp.1251-1258.

Chollet, F., 2024. Keras image data loading API. Available at: https://keras.io/api/data_loading/image/.

Cohen, J.P., Morrison, P., Dao, L., Roth, K., Duong, T.Q. and Ghassemi, M., 2020. COVID-19 Image Data Collection: Prospective Predictions are the Future. *arXiv preprint*. Available at: https://arxiv.org/abs/2006.11988.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20(3), pp.273-297. Available at: https://doi.org/10.1007/BF00994018.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, USA: IEEE, pp.770-778. Available at: https://doi.org/10.1109/CVPR.2016.90.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Identity mappings in deep residual networks. *arXiv preprint*. Available at: https://arxiv.org/abs/1610.02391.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), pp.90-95. Available at: https://doi.org/10.1109/MCSE.2007.55.

Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint*. Available at: https://arxiv.org/abs/1412.6980.

Oliphant, T.E., 2007. Python for scientific computing. *Computing in Science & Engineering*, 9(3), pp.10-20. Available at: https://doi.org/10.1109/MCSE.2007.58.

OpenAI, 2024. ChatGPT [computer program]. Available at: https://chat.openai.com/

Paszke, A., Gross, S., Massa, F., et al., 2019. PyTorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems (NeurIPS)*, 32, pp.8024-8035.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al., 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pp.2825-2830.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, pp.618-626. Available at: https://doi.org/10.1109/ICCV.2017.74.

Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), p.60. Available at: https://doi.org/10.1186/s40537-019-0197-0.

StatQuest with Josh Starmer, 2019. Introduction to Machine Learning [video]. Available at: https://www.youtube.com/watch?v=JcU72smpLJk.

StatQuest with Josh Starmer, 2020. Gradient Boosting and XGBoost [video]. Available at: https://www.youtube.com/watch?v=m2lYtgmHsbk.

Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B. and Liang, J., 2016. Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5), pp.1299-1312. Available at: https://doi.org/10.1109/TMI.2016.2535302.

Tschandl, P., Codella, N. and Akay, B.N., 2020. Impact of class imbalance on deep learning for medical image classification. *The Lancet Digital Health*, 2(6), pp.e315-e323. Available at: https://doi.org/10.1016/S2589-7500(20)30114-8.

Waskom, M.L., 2021. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), p.3021. Available at: https://doi.org/10.21105/joss.03021.