Progress Report

COVID-19 Detection from X-rays using CNN

Kseniia Lipikhin 219 273 127

1 Introduction

COVID-19 is a respiratory disease caused by the SARS-CoV-2 virus, which was first identified in December 2019 with more than 7 million confirmed cases worldwide (as of 11 January 2025) (REF1). Detecting COVID-19 early is important to stop the virus from spreading and to provide the right treatment for patients. The most common way to test for COVID-19 is using PCR (Polymerase Chain Reaction)(REF1).

X-ray imaging is a fast and widely available method for detecting lung infections, including COVID-19 (REF2). Although X-rays usually cost more than PCR tests, they offer many benefits. While PCR tests only show if a patient is positive, X-rays reveal how severely the lungs are affected. Additionally, once an X-ray is performed (it takes only 15 seconds, while results for PCR tests can take around 12-24 hours), it can help diagnose other diseases as well (REF7).

However, reading X-ray images manually takes time and requires expert radiologists, and there is a shortage of these professionals, leading to delays in diagnosis (REF5). To address this, researchers are developing machine learning models that automatically detect COVID-19 from X-rays (REF3). Together with AI, X-rays can provide a fast and effective way to diagnose patients when time is critical.

In this project, we aim to develop a model that can classify X-ray images into 4 categories (Covid-19, Lung Opacity (Non-Covid infection), Viral Pneumonia and Normal).

2 Dataset Description

Researchers from Qatar University, the University of Dhaka, and their collaborators in Pakistan and Malaysia, in partnership with medical professionals, have developed a chest X-ray image database (REF8).

The dataset consists of:

- 3616 COVID-19 X-ray images
- 6012 Lung Opacity (Non-COVID lung infection) images
- 1345 Viral Pneumonia images
- 10102 Normal X-ray images

Examples of these X-ray images are shown in Figure 1. All images are in PNG file format and the resolution is 299x299 pixels.

References for each image are provided in the metadata.

No demographic information (such as gender, age, or race) of the individuals is provided, which could introduce bias into the model if the dataset is severely unbalanced.

((a)) COVID-19 X-ray

((b)) Pneumonia X-ray

((c)) Normal X-ray

((d)) Lung Opacity X-ray

Figure 1: Example X-ray images for COVID-19, Pneumonia, Lung Opacity and Normal cases.

3 Literature Review

An AI model developed by Warwick, King's College London, and several NHS sites can diagnose X-rays as accurately as or even better than doctors, correctly identifying 37 conditions with 94% accuracy(REF4).

This impressive performance is achieved using deep learning, specifically through Convolutional Neural Networks (CNNs), which work by breaking images into small pieces and finding patterns (REF6).

CNNs have also been used by researchers for COVID-19 detection on chest X-rays. In one study(REF3), eight CNN models were evaluated for chest X-Ray image classification, using both shallow networks (MobileNetv2, SqueezeNet, ResNet18) and deeper networks (Inceptionv3, ResNet101, CheXNet, VGG19, DenseNet201).The study investigated the performance of models with and without image augmentation and compared their accuracy in classifying two-class (normal and COVID-19 pneumonia) and three-class (normal, viral and COVID-19 pneumonia) classification. In the two-class problem, CheXNet, pre-trained on X-ray images, performed the best, achieving 99.4% accuracy without augmentation and 99.7% with it, slightly outperforming other models. In the three-class problem, CheXNet achieved the highest accuracy without augmentation at 97.7%, while DenseNet201 had the highest accuracy with augmentation at 97.9%.

4 Introduction to the Model

4.1 Neural Networks

A Neural Network (NN) is a computational model inspired by the human brain. It is structured as a sequence of layers: an input layer, one or more hidden layers, and an output layer. Each layer contains neurons, which are small processing units that help transform the data as it moves through the network.

Each neuron contains parameters (weights), which determine how input signals are scaled before being passed on. These weights are used to transform input values (such as numbers, images, or sounds) into meaningful outputs (such as predictions or classifications). The input layer receives raw data, which is then processed by the hidden layers (each performing a series of weighted computations) before reaching the output layer, where the final decision or prediction is produced. (REF15)

To enable neural networks to learn non-linear relationships, each neuron also applies an activation function after computing its weighted input. Without

activation functions, the network would act as a simple linear model, regardless of depth. We used a popular activation function ReLU (Rectified Linear Unit), which outputs zero for negative inputs and passes positive values unchanged.

When training begins, the network's weights are typically initialized with random values. The model then processes the input data and makes predictions. As these early predictions are often inaccurate, a loss function is used to quantify the error between the predicted outputs and the actual answers. This loss guides the network in adjusting its weights over time to improve accuracy. (REF15)

In our project, we used the categorical cross-entropy loss function, which is well-suited for multi-class classification problems. It measures the dissimilarity between the predicted probability distribution and the true class labels.

$$\text{Loss} = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i)$$

In this formula: C is the total number of classes, y_i is 1 if the input belongs to class i, and 0 otherwise. And $\hat{y}_i$ is the model's predicted probability for class i.

This equation ensures that the model is penalized when it assigns low probability to the correct class. A correct prediction with high confidence results in a small loss, while an incorrect or uncertain prediction results in a higher loss.

To minimize error over time, the training process uses a method called back-propagation to reduce the loss. In this process, we compute the gradient (i.e., partial derivative) of the loss function with respect to each weight. This gradient tells us how to adjust each weight to reduce the loss. An optimization algorithm such as Gradient Descent is then used to update the weights. (REF15)

The gradient of categorical cross-entropy with respect to the model's weights W is given by:

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial}{\partial W}\left(-\sum_{i=1}^{C} y_i \log(\hat{y}_i)\right)$$

Once the gradient is computed, each weight is adjusted by a small amount in the direction that reduces the loss, and this change is scaled by a value called the learning rate.

$$W^{(t+1)} = W^{(t)} - \alpha \frac{\partial \mathcal{L}}{\partial W}$$

$W^{(t)}$ represents the weights at iteration t, α is the learning rate(controls the step size of the update), and $\frac{\partial \mathcal{L}}{\partial W}$ is the gradient of the loss function with respect to the weights.

This iterative process continues until the loss converges to a minimal value, meaning the model has learned an optimal set of weights for classification.

The entire cycle of making predictions, calculating loss, updating weights, and repeating is known as an epoch. Training consists of many epochs. Over time, the network gradually improves its accuracy as it sees more data and fine-tunes its internal parameters.

4.2 Convolutional Neural Networks (CNNs)

While traditional neural networks are powerful, they are not ideal for image data. This is because they treat images as flat vectors, ignoring the spatial relationships between pixels. In images, neighboring pixels often have meaningful relationships, and the order in which pixels appear matters.

Convolutional Neural Networks (CNNs) are a special type of neural network designed specifically to handle images. Rather than flattening images into one-dimensional inputs, CNNs preserve spatial structure using layered operations that identify edges, textures, patterns, and shapes. (REF16)

4.3 Layers in a CNN

The first layer in a CNN is usually a convolutional layer. This layer applies small filters across the image, performing convolution operations to extract important features like edges and corners. Each filter is trained to detect specific visual patterns. The output of this layer is a set of feature maps that highlight where in the image those patterns occur. (REF16)

Following the convolutional layers are pooling layers, which reduce the spatial dimensions of the feature maps. A common pooling method is max pooling, which selects the highest value from a region (e.g., a 2×2 block) in the feature map. This reduces the size of the data while retaining the most important information. Pooling also improves computational efficiency and makes the model more robust to small shifts and distortions in the image. (REF16)

After multiple convolutional and pooling layers, CNNs use fully connected layers to interpret the features and make predictions. These layers flatten the final feature maps into a one-dimensional vector, which is passed through one or more dense layers (traditional Neural Network architecture). The last layer uses

a another activation function. We used Softmax, which converts raw scores into a probability distribution by exponentiating the scores and normalizing them, to output a probability distribution over classes for classification tasks. (REF16)

4.4 Transfer Learning

Transfer learning is a technique in deep learning where a model trained on one large dataset (such as ImageNet (REF13) in our project) is reused as the starting point for a new but related task. These pre-trained models have already learned to detect general features like edges, shapes, and textures, which are useful for many different image-related problems.

Instead of training a CNN from scratch, which requires a large amount of data and computational power, transfer learning allows us to fine-tune an existing model. Usually, the early layers of the model are frozen, preserving their learned representations, while the later layers are retrained on the new dataset. This process adapts the model to the specific task while retaining the general knowledge it has acquired. (REF16)

There are several advantages to using transfer learning. It significantly speeds up the training process because fewer parameters need to be learned. It also performs well with smaller datasets, which is particularly valuable when labeled data is limited. Also, pre-trained models tend to achieve higher accuracy since they have already developed a strong understanding of visual features. (REF16)

5 Model

5.1 Introduction

In this project, we focus on the task of classifying chest X-ray images into four diagnostic categories. To accomplish this, we use Convolutional Neural Networks. However, CNNs typically require large amounts of data and computational power due to their many parameters. To address this challenge, we apply transfer learning, which allows us to build on a pre-trained model instead of training a new one from scratch. We use a pre-trained model initialized with ImageNet weights (a large dataset of natural images). We unfreeze the last 20 layers of the base model to allow partial retraining. This provides a balance: unfreezing too many layers may cause the model to overfit, while unfreezing too few may limit the model's ability to adapt to the specific characteristics of chest X-ray images. We will first select a transfer learning model with a basic (common architecture and hyperparameters) fully connected neural network. After identifying the best-performing model, we will experiment with different architectures and hyperparameters for the fully connected layers to further optimize performance.

5.2 Base Models

We experimented with several popular CNN architectures that are known for their effectiveness and efficiency in image classification tasks, particularly in the medical imaging domain:

5.2.1 DenseNet121

DenseNet121 is designed using a feed-forward architecture, where each layer receives input not just from the previous layer, but from all earlier layers. This dense connectivity allows feature reuse and improves the flow of information and gradients throughout the network, making it easier to train as you continuously refer back to all previously learned features. Despite its depth, this model remains efficient, requires fewer parameters and maintains high accuracy. (REF10)

5.2.2 MobileNetV2

MobileNetV2 is optimized for efficiency and is particularly well-suited for scenarios where computational resources are limited. It uses two key innovations:

1. Depthwise Separable Convolutions: Instead of processing all channels of an image at once, this technique separates the filtering and combining steps. First, each channel is filtered individually (depthwise convolution), and then the results are merged using a pointwise (1x1) convolution. This greatly reduces computation without sacrificing much accuracy. (REF12)

2. Inverted Residual Blocks: These blocks expand the data dimensions before applying transformations and then compress them back down. This allows the network to learn rich representations efficiently while keeping the overall model size small. (REF12)

5.2.3 InceptionV3

InceptionV3 enhances its learning ability by applying parallel convolutional filters of different sizes on the same input. This architecture enables the model to capture both small details and broader patterns within an image. For example, small filters focus on fine-grained features like edges, while larger filters detect overall structures. The results from these multiple paths are then combined, giving the model a richer and more comprehensive understanding of the image. (REF 11)

This parallel processing helps reduce overfitting and enhances generalization, which is particularly important in medical image classification where precision is critical.

5.3 Summary of Proposed Architecture

After conducting thorough experimentation, this represents the finalized model architecture.

CNN Part (Feature Extractor)

- **Base Model:** A pre-trained CNN DenseNet121 with ImageNet weights. The base model processes input images and extracts meaningful features.
- **Layer Freezing Strategy:** The first layers of the base model are frozen and the last 20 layers are trainable for fine-tuning.
- **Batch Normalization:** Applied to stabilize learning and speed up training.

Fully Connected Neural Network (Classifier)

- **Fully Connected Layers:**
  - A dense layer with 256 neurons and ReLU activation.
  - L2 regularization is applied to prevent overfitting.
- **Dropout Layer:** Set at 0.3 to reduce overfitting by randomly disabling neurons during training.
- **Output Layer:** A softmax layer with four neurons (one for each class) to produce classification probabilities.

Model Training

- **Optimizer:** Adamax optimizer with a learning rate of 0.0001.
- **Loss Function:** Categorical cross-entropy loss is used, with a weighted loss function to handle class imbalance.

6 Preliminary Results

Accuracy and recall was used to evaluate the three different deep learning models (MobileNetV2, InceptionV3, and DenseNet121). Accuracy shows how often the model makes correct predictions overall. However, in medical diagnosis, recall (sensitivity) is even more important because it measures how well the model identifies actual disease cases. High recall helps ensure that fewer cases are missed, which is critical for conditions like COVID-19, where missing a case could delay treatment.

The deep learning models demonstrated promising results, with DenseNet121 achieving the highest validation accuracy at 89% (Fig 5a). MobileNetV2 and InceptionV3 had validation accuracies of 86% (Fig5 b and c) Training and validation loss steadily decreased over epochs, indicating effective learning (Fig2a, Fig3a, Fig4a). However, additional epochs were required to achieve further performance gains (models were still learning since the loss was still decreasing).

However, class-wise recall varied significantly. The models exhibited strong performance in classifying Normal cases, with recall ranging from 95% to 98%, suggesting their ability to accurately distinguish healthy lungs (Fig 5). The lung opacity cases also performed well, with a recall between 80% and 85% (Fig 5). The recall of viral pneumonia ranged from 80% to 87% (Fig 5), with some misclassifications occurring, but at a lower rate compared to COVID. The COVID class had the lowest recall, varying between 65% and 77%(Fig5) with DenseNet121 having the highest (77%), indicating the challenge to distinguish it from other conditions, particularly Lung Opacity(as seen in Confusion Matrices Fig 2c, Fig 3c, Fig 4c). The most common misclassification occurred between COVID and Lung Opacity, suggesting that the models struggle to differentiate between these conditions due to similar visual features (See Confusion Matrices for each model in Fig 2c, Fig 3c, Fig 4c).

7 Final Results

Since DenseNet121 has shown the most promise it was chosen for further optimization. After refining the model, it achieved a final accuracy of 92% (Fig 7) and an improved COVID recall of 89% (Fig 7), a significant gain over the preliminary results.

8 PCR tests vs X-rays for COVID Detection

PCR tests are considered the gold standard due to their high sensitivity (98.5%) (REF 14), but they can be costly (especially if the hospital need to outsource the laboratory testing) and slow to return results. X-rays, on the other hand, offer faster turnaround times and lower costs (if using AI to interpret results), though with potentially lower sensitivity (92% with our model) . To determine when X-rays are more beneficial than PCR tests, we can use an optimization approach that balances the cost of testing, the value of early detection, and the sensitivity of each method.

We define the net benefit of each method as the value gained from true positive detections minus the cost of testing.

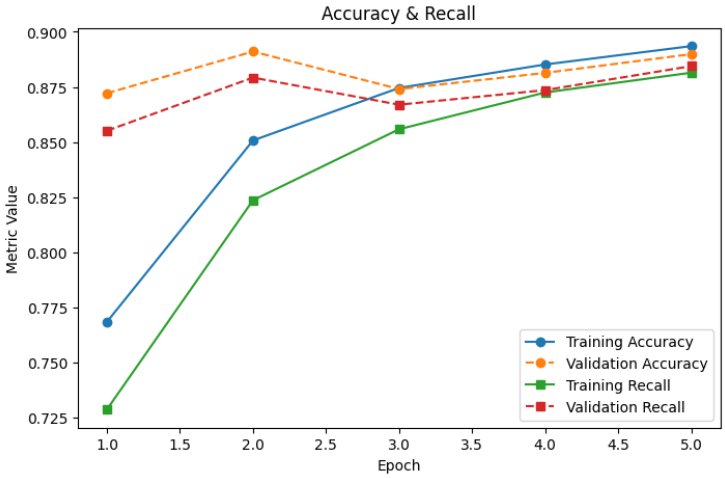
$$\text{Net Benefit} = P \cdot S \cdot V - A \cdot C$$

P is the disease prevalence, S the sensitivity, V the value of detecting a true case early, A the importance of cost, C the cost per test. P (disease prevalence) reflects the probability that a randomly selected patient has the disease. It is crucial because only true positive detections contribute value; the higher the prevalence, the more likely it is that a test yields a meaningful result.

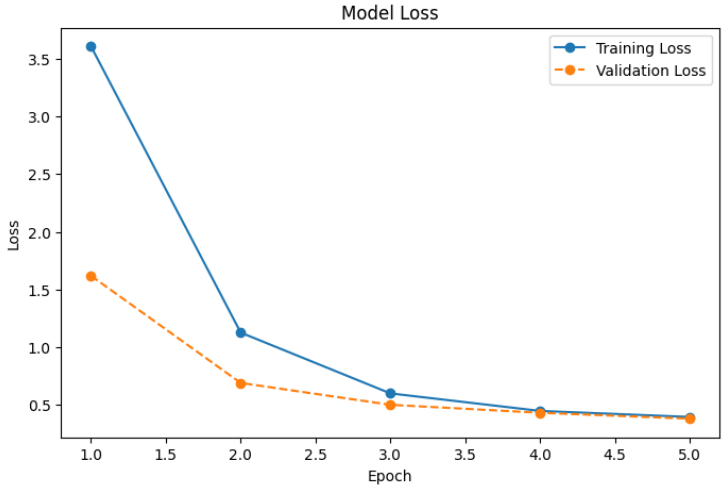
The value V should be chosen depending on how early detection of a true case is important. The higher the value, the greater the benefit of identifying the disease early. Similarly, the higher the value of A, the more weight is given to keeping the test cost low. X-ray testing is more beneficial than PCR when the net benefit from using X-rays exceeds that of PCR tests. Simplifying, this occurs when:

$$P \cdot V \cdot (S_{\text{xray}} - S_{\text{pcr}}) > A \cdot (C_{\text{xray}} - C_{\text{pcr}})$$

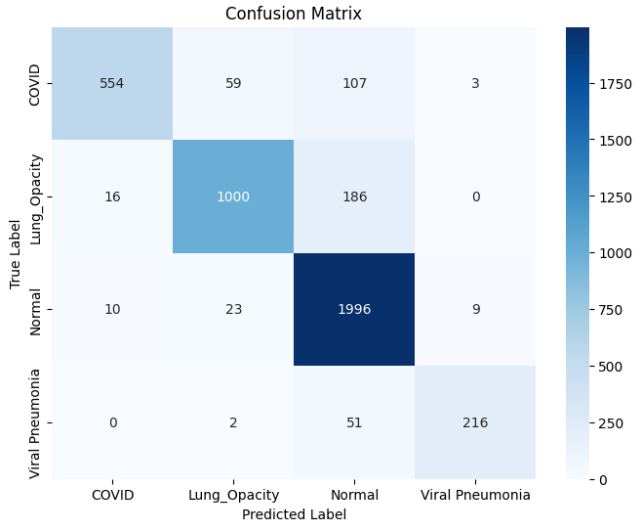
This inequality shows that X-rays are preferable when the disease is relatively prevalent, the value of early diagnosis is high (V), and the cost savings from using X-rays (weighted by the importance of cost A) outweigh the reduction in sensitivity.



((a)) Accuracy and Recall over epochs - DenseNet121

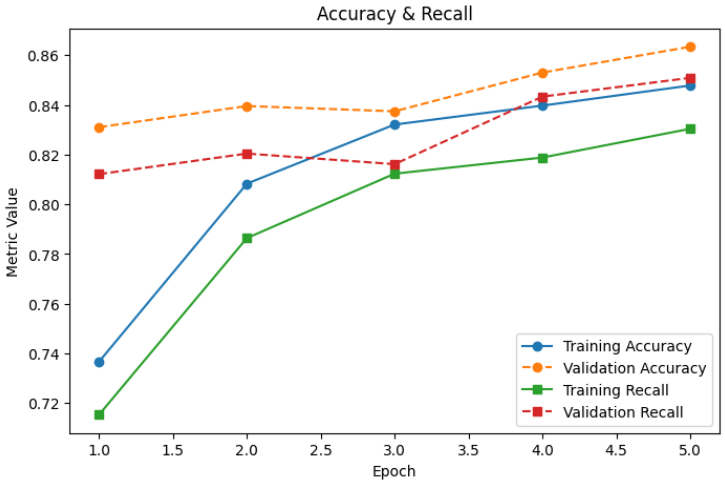


((b)) Loss over epochs - DenseNet121

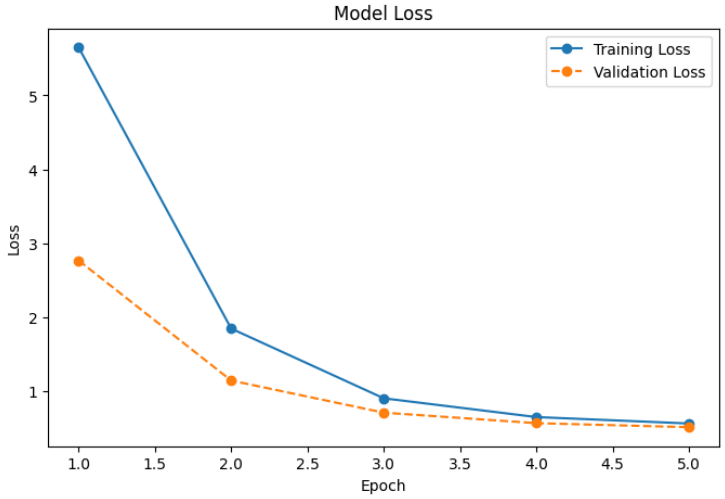


((c)) Confusion Matrix - DenseNet121

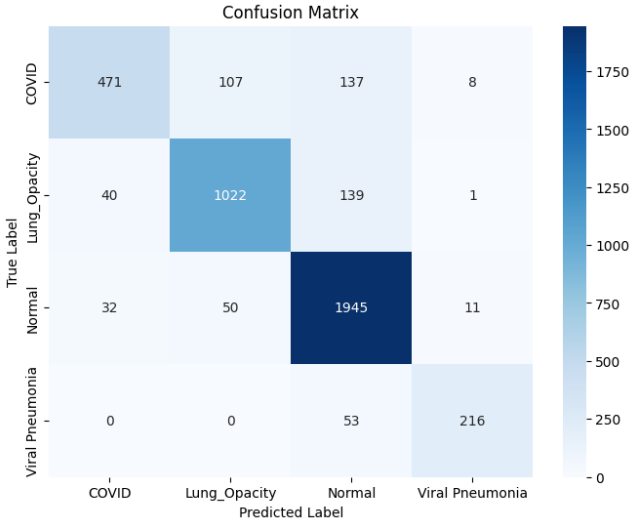
Figure 2: DenseNet121 plots (preliminary).



((a)) Accuracy and Recall over epochs - InceptionV3

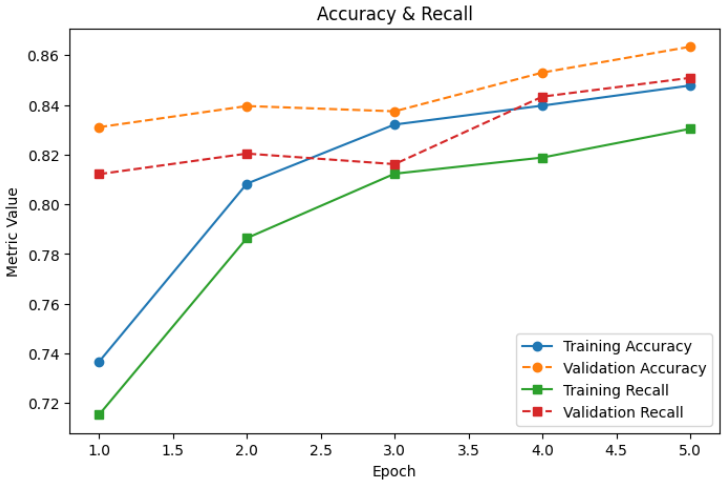


((b)) Loss over epochs - InceptionV3

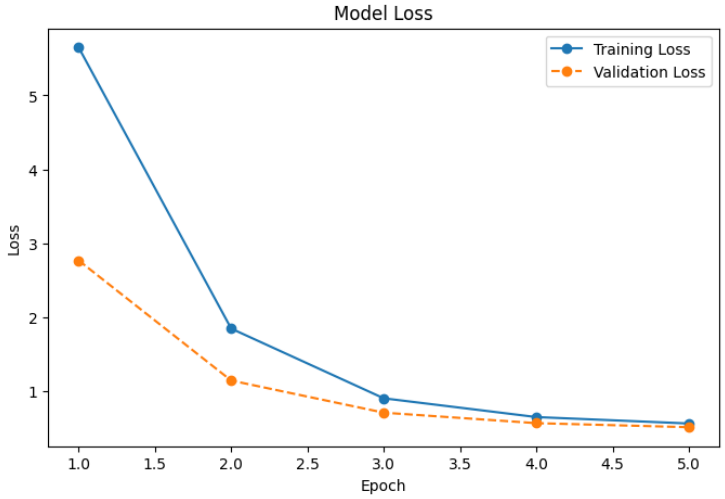


((c)) Confusion Matrix - InceptionV3

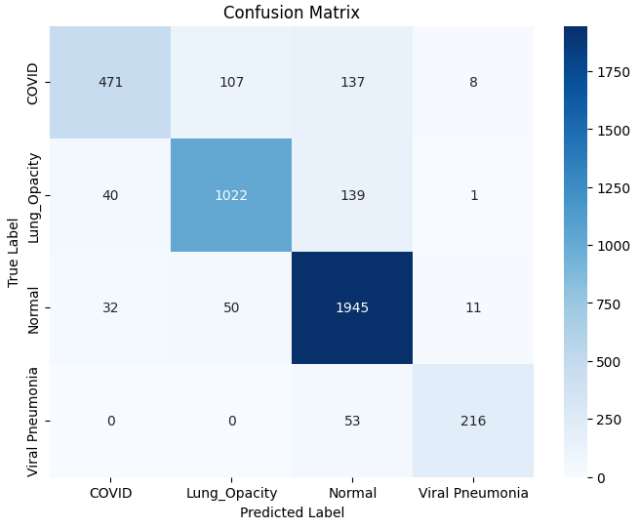
Figure 3: InceptionV3 plots (preliminary).



((a)) Accuracy and Recall over epochs - MobileNetV2



((b)) Loss over epochs - MobileNetV2



((c)) Confusion Matrix - MobileNetV2

Figure 4: MobileNetV2 plots (rpeliminary).

	precision	recall	f1-score	support
COVID	0.96	0.77	0.85	723
Lung_Opacity	0.92	0.83	0.87	1202
Normal	0.85	0.98	0.91	2038
Viral Pneumonia	0.95	0.80	0.87	269
accuracy			0.89	4232
macro avg	0.92	0.85	0.88	4232
weighted avg	0.90	0.89	0.89	4232

((a)) DenseNet121

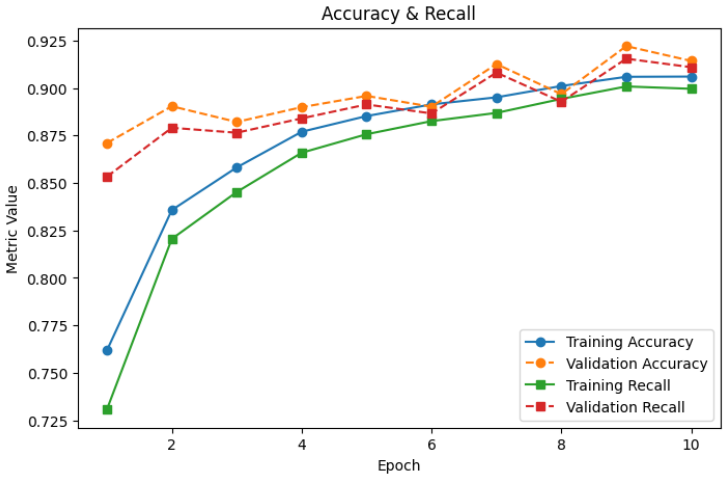
	precision	recall	f1-score	support
COVID	0.87	0.65	0.74	723
Lung_Opacity	0.87	0.85	0.86	1202
Normal	0.86	0.95	0.90	2038
Viral Pneumonia	0.92	0.80	0.86	269
accuracy			0.86	4232
macro avg	0.88	0.81	0.84	4232

((b)) InceptionV3

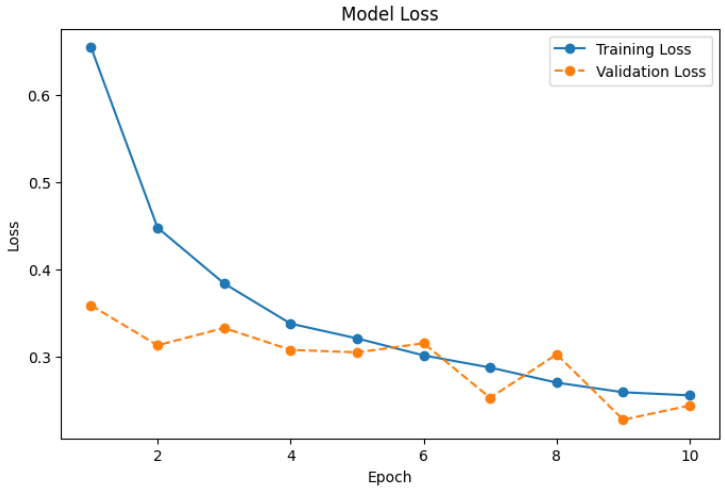
	precision	recall	f1-score	support
COVID	0.87	0.65	0.74	723
Lung_Opacity	0.87	0.85	0.86	1202
Normal	0.86	0.95	0.90	2038
Viral Pneumonia	0.92	0.80	0.86	269
accuracy			0.86	4232
macro avg	0.88	0.81	0.84	4232
weighted avg	0.86	0.86	0.86	4232

((c)) MobileNetV2

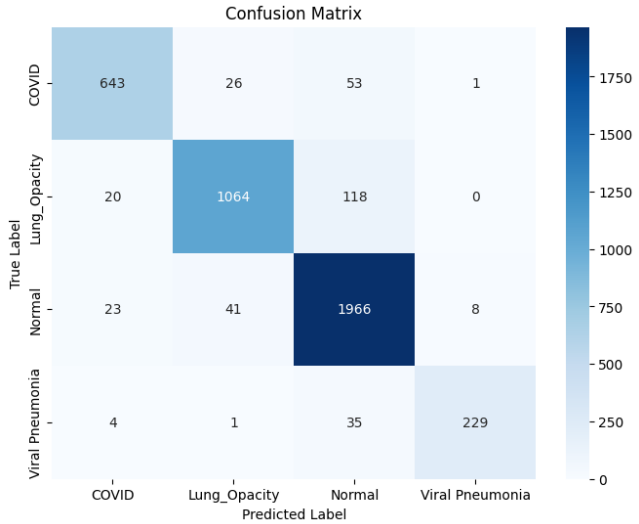
Figure 5: Preliminary Evaluation Metrics for each base model.



((a)) Accuracy and Recall over epochs - DenseNet121 - Final



((b)) Loss over epochs - DenseNet121 - Final



((c)) Confusion Matrix - DenseNet121 - Final

Figure 6: Final DenseNet121 plots.

	precision	recall	f1-score	support
COVID	0.93	0.89	0.91	723
Lung_Opacity	0.94	0.89	0.91	1202
Normal	0.91	0.96	0.93	2038
Viral Pneumonia	0.96	0.85	0.90	269
accuracy			0.92	4232
macro avg	0.93	0.90	0.91	4232
weighted avg	0.92	0.92	0.92	4232

Figure 7: DenseNet121 - Final Metrics

9 References

References

[1] World Health Organization. *Coronavirus Disease (COVID-19) Pandemic*. 2024. https://www.who.int/news-room/fact-sheets/detail/coronavirus-disease-(covid-19).

[2] E. Saez de Gordoa, A. Portella, J. M. Escudero-Fernández, and J. Andreu Soriano. *Usefulness of chest X-rays for detecting COVID 19 pneumonia during the SARS-CoV-2 pandemic*. National Library of Medecine, 2022. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC9401621/.

[3] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam.*Can AI Help in Screening Viral and COVID-19 Pneumonia?*. IEEE Access, Vol. 8, 2020. Available at: https://ieeexplore.ieee.org/document/9144185.

[4] Vicky Goh. *AI trained on X-rays can diagnose medical issues as accurately as doctors*. King's College London 2023. Available at: https://www.kcl.ac.uk/news/ai-trained-on-x-rays-can-diagnose-medical-issues-as-accurately-as-doctors.

[5] Elizabeth Y. Rula *Radiology Workforce Shortage and Growing Demand Something Has to Give*. Bulletin 2024. Available at: https://www.acr.org/clinical-resources/publications-and-research/acr-bulletin/Radiology-Workforce-Shortage-and-Growing-Demand-Something-Has-to-Give.

[6] Mohd Sanad Zaki Rizvi *Image Classification Using CNN*. Analytics Vidhya 2024. Available at: https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/.

[7] Tabik S, Gomez-Rios A, Martin-Rodriguez JL, Sevillano-Garcia I, Rey-Area M, Charte D, et al.*Image Classification Using CNN*. IEEJour0al of Biomedical and Health Informatics 2020. Available at: https://ieeexplore.ieee.org/document/9254002.

[8] Muhammad E. H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar.*COVID-19 Radiography Database*. Kaggle 2022. Available at: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

[9] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger.*Densely Connected Convolutional Networks*. Cornell University 2018.

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger.*Densely Connected Convolutional Networks*. Cornell University 2018.

[11] Christian Szegedy; Wei Liu; Yangqing Jia; Pierre Sermanet; Scott Reed; Dragomir Anguelov.*Going deeper with convolutions*. IEEE 2015.

[12] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam.*Searching for MobileNetV3*. Cornell University 2019.

[13] ImageNet. *A large-scale hierarchical image database*. Available at: `https://www.image-net.org/index.php`

[14] Mirella Cristine Oliveira, Karoleen Oswald Scharan, Bruna Isadora Thomés, Rafaella Stradiotto Bernardelli, Fernanda Baeumle Reese, Amanda Christina Kozesinski-Nakatani, Cintia Cristina Martins, Suzana Margareth Ajeje Lobo, Álvaro Réa-Neto. *Diagnostic accuracy of a set of clinical and radiological criteria for screening of COVID-19 using RT-PCR as the reference standard*. BMC Pulmonary Medicine, 2023. Available at: `https://bmcpulmmed.biomedcentral.com/articles/10.1186/s12890-023-02369-9`

[15] Tariq Rashid. *Make Your Own Neural Network*. CreateSpace Independent Publishing Platform; Illustrated edition (March 31, 2016).

[16] Saban Ozturk (Ed.). *Convolutional Neural Networks for Medical Image Processing Applications*. CRC Press, October 8, 2024. ISBN 9781032104010.