Multiclass Classification of Chest X-ray Images Using Convolutional Neural Networks

Sumaiya Khanam[1†], Sirazum Munira[1†], Riya Saha[1†]

[1*]Department of Computer Science and Engineering, Premier University, Chattogram, 4000, Bangladesh.

Contributing authors: sumaiya.khanam01@gmail.com; 1133sirazum.munira@gmail.com; riyasaha1124@gmail.com;
[†]These authors contributed equally to this work.

Abstract

This work introduces a VGG16 based CNN architecture for multiclass classification of chest X-ray images. It has three clinically significant classes: Normal, Lung Opacity and Viral Pneumonia. The dataset is preprocessed using standardized scaling, normalization and stratified splitting after being aggregated from publically accessible radiography repositories to guarantee the robustness of model training.A baseline VGG16-style network architecture was trained end-to-end and comprehensive experimentation was conducted regarding the influence of key hyperparameters (such as learning rate, dropout rate, batch size, optimizer choice, input resolution, activation functions, padding strategies, and stride variations) on convergence stability and generalization. Wider and deeper architectural variants are also investigated to quantify performance improvements arising from increased feature extraction capability.These results show that properly tailored CNN architectures can produce strong multiclass performance for automated chest X-ray screening, even while classes with similar visual appearance, such as Lung Opacity and Viral Pneumonia, remain difficult. This work highlights the significance of carefully selecting hyperparameters and architecture in medical image analysis, and it serves as a foundation for future advancements utilizing more sophisticated models and bigger, more varied datasets.

Keywords: Lung X-ray Classification, VGG16 Architecture, Deep Learning, Medical Image Analysis

1 Introduction

One of the most popular diagnostic instruments in clinical practice, a chest X-ray often abbreviated as CXR assists in the diagnosis of a number of lung disorders. Deep learning, particularly with CNNs, has emerged as a successful technique for automated medical image classification thanks to increased medical imaging data and advancements in artificial intelligence. Radiologists will benefit from automated chest X-ray reading, which will also lessen the diagnostic burden and aid in the early detection of disease.

This work focusses on using CNN-based deep learning models to classify chest X-ray images into multiple categories. Lung Opacity, Normal Lung, and Viral Pneumonia are the three most crucial classification categories. These categories were selected because they are clinically significant, visually similar, and therefore difficult and crucial to distinguish between automatically.

The high visual similarity between Lung Opacity and Viral Pneumonia, low contrast and overlapping anatomical structures in CXR images, and the sensitivity of CNN models to hyperparameter choices and architectural variations make it challenging to achieve reliable multiclass performance, despite CNNs' strong potential for classifying chest X-ray images. In order to comprehend how various hyperparameters, activation functions, stride settings, and model structural modifications (deeper or wider designs) affect CNN performance in classifying images of lung opacity, viral pneumonia, and normal lung, a systematic investigation is necessary. By examining the impact of these variables and determining configurations that enhance discrimination between visually similar CXR categories, this study seeks to address these issues.

To improve model performance, a detailed comparison of different CNN architectures and training configurations is included in the study. Batch size, learning rate, optimiser selection, dropout rate, early stopping, input resolution, padding strategy, activation functions, and stride settings are among the significant hyperparameters whose effects on accuracy, convergence stability, and generalisation capacity are examined. Several activation functions, such as Tanh, Leaky ReLU, and ReLU, are examined in addition to standard settings to ascertain which nonlinear transformation supports superior feature extraction and gradient flow.

Furthermore, structural changes to the CNN are analysed using wider models that add more filters to enhance spatial feature representation and deeper models that add more convolutional layers for richer hierarchical feature learning. These variations are compared to the base model in order to evaluate improvements in learning behaviour and class-specific discrimination.

By carefully analysing the effects of different hyperparameters and architectural variations, this work seeks to identify model configurations that achieve robust, reliable performance for chest X-ray multiclass classification. The final findings contribute to the creation of automated screening tools that can make it easier to accurately and successfully identify lung-related conditions in real-world healthcare settings.

2 Related work

Some literature surveys for multiclass classification of chest X-ray images using other methods are given below.

Shamrat [1] implemented a fine-tuned MobileLungNetV2 model from another DCNN transfer-learning model called MobileNetV2, which was utilized for the multiclass classification of chest X-ray14 images. This MobileLungNetV2 model extracted better features and successfully identified abnormalities in the lungs. The experimental results showed that this fine-tuned model achieved a higher classification accuracy of an average AUC of 91.6However, this model had time complexity issues with high computational demands.

Nahiduzzaman [2] proposed a convolutional neural network with an Extreme Learning Machine (CNN-ELM) method for multiclass categorization of CXR images. The implemented method is integrated with the lightweight parallel capability of CNN and ELM classification. As a result, this integrated CNNELM method was successfully achieved and the 17 kind of lung diseases are identified. Nonetheless, after the integration with multiple Chest X-ray datasets, some categories of images suffered in the CNN-ELM method due to an insufficient quantity of training data.

Mann [3] introduced the Multi-Modal Fusion of Deep Transfer Learning (MMF-DTL) method employed to classify CXR images into six classes. This model used 3 DL specifically VGG16, Inception v3, and ResNet 50 for extensive feature extraction and softmax as the classifier. The outcomes established that the MMF-DTL method suffered from less training data especially while using ResNet-152, which is prone to overfitting.

Chen [4] developed a pyramidal convolution module and shuffle attention module (PCSANet) based on a residual network model for 14 thorax disease classification. Pyramid convolution was utilized to extract features from images and shuffle attention enabled a focus on pathological features. This PCSANet model efficiently increased the thoracic disease classification performance. Acquiring additional multi-scale discriminative features of pathological abnormality improves the overall classification efficiency. Nonetheless, the AUC score of infiltration was identified as quite low.

Moreover, Xu [5] proposed a Group-Wise Spatial Attention (GWSA) and Label Co-Occurrence Dependency (LCD) module for the classification of multiple diseases. The GWSA increased spatial features inside dissimilar groups and the LCD module did correlations between different thoracic abnormalities. This implemented method was employed with a DCNN called DenseNet. Due to low specificity, this implemented method had less performance in correctly identifying a negative class.

Wang [6] developed a Thorax-Net model for 14 thorax diseases by exploiting the Chest X-Ray14 dataset. This approach contained the ResNet-152 Received: May 15, 2024. Revised: July 1, 2024. 26 International Journal of Intelligent Engineering and Systems, Vol.17, No.6, 2024 DOI: 10.22266/ijies2024.1231.03 classification block and performed feature extraction, supported by the attention branch. The ResNet was exploited as a classification branch assisted by Gradient-weight Class Activation Maps (Grad-CAM). The performance of this ResNet-152-based ThoraxNet model was affected by overfitting due to imbalanced disease class distribution in the Chest XRay14 dataset.

3 Dataset Overview

The dataset employed here includes three clinically meaningful classifications of chest X-ray images: Normal lung, Lung Opacity, and Viral Pneumonia. These groups are common thoracic disorders that are usually difficult to identify with the unassisted eye and frequently appear on diagnostic imaging. The images are placed in different folders for each class, keeping a standard structure for a deep-learning dataset.

3.1 Dataset Source

It is compiled from openly available chest radiography collections and is widely used in medical image classification research. In this project, the dataset was obtained from Sugianto, Dwi (2025), "Chest X-Ray", Mendeley Data. To assure that there is a wide range in the representation of patient demographics, imaging quality, and pathological variance, the photos were gathered from prestigious clinical archives and research grade databases.

3.2 Dataset Structure and Sample Characteristics

The Lung Disease Dataset consists of 3,475 chest X-ray images from three clinically prominent classes: Normal, Lung Opacity, and Viral Pneumonia. This dataset was collected from Kaggle and focused on frontal chest X-ray pictures suitable for multiclass classification tasks in computer vision and medical imaging applications.To support supervised learning workflows and support supervised learning workflows,
the images are organised in a standard directory structure with separate train, validation, and test folders.Although the original chest X-ray images are naturally grayscale, they are stored and loaded in RGBA format during processing and visualization . Most training pipelines don't require additional resizing or format conversion because the images have undergone consistent preprocessing and can usually be used directly by deep learning models. The following classes are included in the dataset

Normal: 1,250 images
Lung Opacity: 1,125 images
Viral Pneumonia: 1,100 images

In accompanying notebooks or overviews on the dataset page, sample images from each class are usually displayed, offering distinct illustrative examples of both healthy lungs and the distinctive opacities and infiltrates linked to the pathological conditions. These illustrations show the minute distinctions that make automated differentiation difficult but useful for practical diagnostic assistance.

Fig. 1: Sample images from the dataset showing different chest X-ray image of Lung Opacity

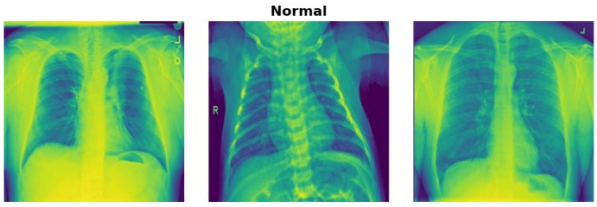

Fig. 2: Sample images from the dataset showing different chest X-ray image of Normal class

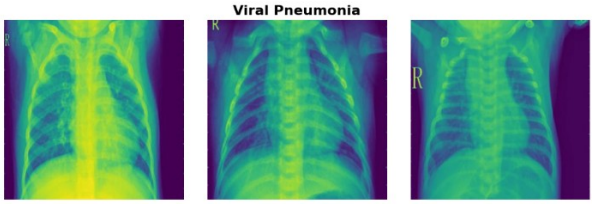

Fig. 3: Sample images from the dataset showing different chest X-ray image of Viral Pneumonia

3.3 Exploratory Data Analysis (EDA)

Preliminary exploratory analysis was done to ascertain the quality and distribution of the dataset.

Class Distribution: There is a clear class imbalance in the dataset, with a significant difference between numbers in different categories. This reflects in model learning and is, therefore, compensated for by class-weighting and augmentation strategies later on.

Image Integrity: A scan for corrupted images was performed, and most samples were intact, while a few corrupted or unreadable files have been removed.The dimensions of the imaging are of different sizes and aspects, reflecting the clinical variety in the

real-world. it confirms that sizes were not uniform, thus requiring resizing during preprocessing.
Visual Inspection: Random samples from each class were examined in order to ensure that the labels provided were indeed consistent with the visual pathology. Each class exhibited a typical radiological pattern; however, some subtle variations within the samples might explain misclassifications.

3.4 Preprocessing

The images were prepared for CNN-based training using a consistent preprocessing pipeline setup:

- Rescaling / Normalization: To ensure uniform pixel intensity and improve model convergence, the raw pixel values of all chest X-ray images were normalized. This was done by applying a rescaling factor of 1./255, which transform pixel values from the range 0–255 to 0–1.
- Image Resizing: All images were resized to a fixed dimension of 224 × 224 pixels. This standardization ensures consistent input size for the neural network and reduces computational complexity
- Train/Validation/Test Split: To assure appropriate model training and evaluation, the dataset was split into three subsets: 70% of the images were assigned to the training set, 15% to the validation set, and the remaining 15% to the test set. This stratified split allows for unbiased hyperparameter tuning and early stopping through validation, supports efficient model training, preserves class balance across all subsets, and offers a trustworthy, unseen test set for the final performance evaluation.
- One-hot Encoding: The class labels were transformed into a categorical (one-hot encoded) format using class mode='categorical'. This allows the model to handle multi-class classification effectively.
- Shuffling (Training Set Only): To prevent the model from learning patterns based on image order, only the training data was shuffled. This ensures better generalization and reduces overfitting

4 Methodology

In this work, a deep convolutional neural network is implemented based on the VGG16 architecture, which is designed from scratch and trained for three-class chest X-ray classification, namely, Normal Lung, Lung Opacity, and Viral Pneumonia. It includes the design of the model, preprocessing, the choice of hyperparameters, the training procedure, and the fine tuning steps.

4.1 Model Architecture

The model design is based on the classic VGG16 architecture, Which has been characterized by small 3×3 convolutional kernels, uniform padding, and deep features

extracted through periodic convolutional blocks. Altogether, there are five major convolutional blocks that comprise this network, followed by max-pooling for spatial downsampling.Its architecture can be summarized as follows:

- **Block 1 architecture:**

  Conv(64 filters, 3×3, ReLU)
  Conv(64 filters, 3×3, ReLU)
  MaxPooling(2×2)

Block 1 comprises two convolutional layers, each with 64 filters of size 3×3 and ReLU activation, followed by a 2×2 max-pooling layer with stride 2. This initial block extracts low-level features such as edges and textures while reducing the spatial resolution by half.

- **Block 2 architecture:**

  Conv(128 filters, 3×3, ReLU)
  Conv(128 filters, 3×3, ReLU)
  MaxPooling(2×2)

Block 2 enhance the depth by using 128 filters. It contains two consecutive 3×3 convolutional layers. Like the block 1 activation function is same,ReLU followed by another 2×2 max-pooling operation. This block starts capturing intricate patterns and mixture of the basic features learned earlier.

- **Block 3 architecture:**

  Conv(256 filters, 3×3, ReLU)
  Conv(256 filters, 3×3, ReLU)
  Conv(256 filters, 3×3, ReLU)
  MaxPooling(2×2)

Block 3 further amplify representational capacity with 256 filters. It includes three convolutional layers, each using 3×3 kernels and ReLU activation, followed by max-pooling. The additional convolutional layer in this block allows the network to learn affluent mid-level features critical for distinguishing subtle lung abnormalities.

- **Block 4 architecture:**

  Conv(512 filters, 3×3, ReLU)
  Conv(512 filters, 3×3, ReLU)
  Conv(512 filters, 3×3, ReLU)
  MaxPooling(2×2)

Block 4 Block 4 employs 512 filters across three convolutional layers with 3×3 kernels and with the Activation function ReLU, again followed by max-pooling. At this stage, the network extracts high-level semantic features that are more unchanging to small conversion and better suited for complex pattern acknowledgment in medical images.

- **Block 5 architecture:**

  Conv(512 filters, 3×3, ReLU)
  Conv(512 filters, 3×3, ReLU)
  Conv(512 filters, 3×3, ReLU)
  MaxPooling(2×2)

Block 5 mirrors Block 4 in structure, consisting of three convolutional layers with 512 filters, 3×3 kernels, and ReLU activation, terminated with a final 2×2 max-pooling layer. Before the classification head, this deepest block generates intensely abstract feature portrayals.

- **Fully Connected Classifier:**

  Flatten layer
  Dense(4096, ReLU)
  Dropout(0.5)
  Dense(4096, ReLU)
  Dropout(0.5)
  Dense(3, Softmax) — outputs probabilities for the three target classes.

The classifier head begins with a flatten operation, followed by two fully connected layers of 4096 units each with ReLU activation and 50% dropout for regularization. The final layer is a dense layer with 3 units and softmax activation, producing probability distributions over the Normal, Lung Opacity, and Viral Pneumonia classes.

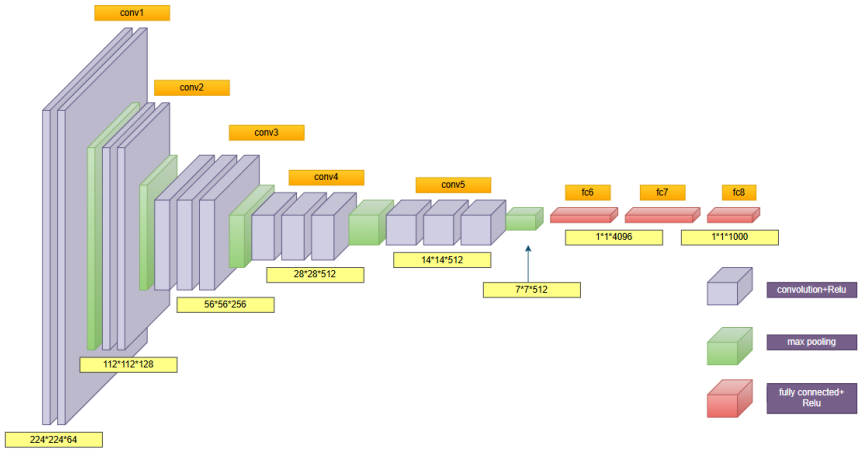

Fig. 4: VGG16 Model Architecture

This custom VGG16-style implementation retains the original depth and progressive increase in filter count while being lightweight enough to train end-to-end on the chest X-ray dataset, providing a strong baseline for subsequent hyperparameter and architectural experiments.

4.2 Hyperparameters

Hyperparameters that used in the base model are:

1. **Learning Rate:** The learning rate is a key hyperparameter in neural networks that controls how quickly the model learns during training.In base model, learning rate of 0.0001 is used.
2. **Optimiser:** An optimizer improves the model by adjusting its parameters (weights and biases) to minimize the loss function value.the Adam optimizer was used to train the base VGG16-style model. This prevented overly aggressive updates from destabilising training on this comparatively small medical dataset.
3. **Batch Size:** Batch size refers to the number of samples used in each of these smaller batches during training.A batch size of 32 was chosen. To create a clean reference performance.
4. **Input Size:** The amount of features or dimensions in each data sample given into a neural network is referred to as the input size.Images were resized to a uniform input resolution of 224x224 pixels and normalised by rescaling pixel values to the range [0,1] using ImageDataGenerator. No additional data augmentation was used in the baseline configuration.
5. **Epoch:** An epoch refers to one complete pass through the entire training dataset where every data sample is passed through the model and its parameters are updated based on the calculated error. The model was trained for a maximum of 20 epochs, but in order to prevent overfitting and guarantee that the optimal checkpoint was kept.
6. **Early Stopping:** Early stopping is a regularization strategy that monitors the model's performance on a validation set during training. EarlyStopping with a patience of 3 and ModelCheckpoint were used.

4.3 Training and Fine-Tuning

Base Training: All convolutional and dense layers were trained from random configuration.A controlled learning rate provided convergence in a stable manner.

Hyperparameter Tuning: Batch sizes, Learning rates, Activation function, Dropout rate, Padding, Stride, Optimizer, Early Stopping patience rate, Model size varied to find the most viable configuration for training. The training curves were supervised to measure the convergence speed and generalization behavior.

Regularization & Model Refinement: Dropout and data augmentation were applied to reduce overfitting. The best performing model was chosen based on the accuracy of the validation, the loss of validation, and the performance metrics per-class.

5 Training Procedure

5.1 Loss Function

In every experiment, the objective function was categorical cross-entropy loss. This loss effectively penalises incorrect class probability distributions generated by the softmax output layer and promotes confident, accurate predictions, making it a good fit for the three-class (Normal, Lung Opacity, Viral Pneumonia) classification problem.

5.2 Optimizers

An optimizer improves the model by adjusting its parameters (weights and biases) to minimize the loss function value. . We employed the Adam optimizer in our base model because it combines momentum and adaptive learning rates to produce robust generalization and faster, more stable convergence without requiring a lot of adjustment. We also evaluate RMSprop and SGD with momentum (momentum = 0.9), keeping all other hyperparameters equal for fairness, in order to methodically analyze the effects of various optimization algorithms on convergence speed, training stability, and final performance.

5.3 Random Seeds

A fixed random seed was not used in the base model.As it has some noteworthy advantages for training reliability and evaluation. In the absence of a seed, each run of the model is given different initial weights, distinct data shuffling patterns and several augmentation outputs. Since a strong model maintains consistent accuracy even under changing training conditions, this inherent randomness help out to show how stable and generalisable the model actually is. Not using a seed can reduce overfitting. A fixed seed can lead to results that are deceptively optimistic because it forces the same initialization, shuffle order and augmentation sequence every time. In order to prevent the model from overfitting to a single configuration and instead provide better average-case performance, removing the seed automatically generates multiple training scenarios. hyperparameter evaluation becomes more practical. Certain hyperparameters may appear to be effective by chance when using a fixed seed. The models tend to learn nearly identical features when they share the same seed, which reduces diversity and degrades ensemble performance. Because every model learns differently when there is no seed, there is greater variability and a discernible improvement in ensemble accuracy.

5.4 Training Environment and Hardware Configuration

All experiments for this work were executed in the Kaggle Notebook environment. We execute a provided code on a cloud GPU runtime that is enabled with an accelerator of NVIDIA Tesla P100. Using Kaggle guaranteed a consistent software stack that included a pre-configured Python environment with popularly used machine learning libraries, and reliable GPU resources for model training and evaluation. Data

preprocessing, code development and report writing have also been executed on a personal laptop besides the cloud environment. The extensive hardware configuration of this machine is summarized in Table. This kind of specification is sufficient to handle the non-GPU-intensive parts of the workflow data handling, result analysis and documentation-while computationally expensive steps of training have been offloaded to the Kaggle GPU environment

Table 1: Hardware Configuration

Component	Model
Processor	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz (8 CPUs)
RAM	16 GB
Storage	1 TB
Operating System	Windows 11 (64-bit System)
GPU	Intel Iris XE Internal GPU
Webcam	Not required

6 Results

The base model in this study serves as an initial benchmark for classifying chest X-ray images into three categories: Normal Lung, Viral Pneumonia, and Lung Opacity. The dataset would, therefore, be divided among the training, validation, and testing subsets, where class-specific files are stored in different folders; all images are resized to 224×224×3. This VGG-style CNN architecture consists of certain blocks of convolutional layers with ReLU activation and then subsequent max-pooling operations to reduce spatial dimensions while retaining important features. Finally, after the convolutional blocks that generate these features, the output is flattened and fed into fully connected layers, ending in a softmax output layer for multiclass prediction. It is also considered that the model was trained on standard hyperparameters defined in the notebook, with Adam optimization, with a fixed learning rate, and callback functions such as early stopping and model checkpointing to prevent overfitting and keep optimal weights. Generally, the architecture provides a clean and stable base on which further experiments could be compared-for instance, varying batch sizes, tuning learning rates, or changing the architecture-to evaluate the improvement in performance.

Here, the confusion matrix and evaluation curves-accuracy and loss-for the base model are shown for clear visualization of its classification performance on three target classes.

Over the course of the 20 epochs, the accuracy/loss curves for training and validation show consistent, smooth behaviour. Accuracy increases rapidly in the first five epochs (validation accuracy rises from about 77% to about 90%), then gradually increases to 98.39% on validation and 98.27% on test, with nearly no difference between the training and validation curves after epoch 8. Loss curves show good convergence and efficient control of overfitting, declining smoothly from 0.537 to 0.048 on validation without oscillations or divergence.

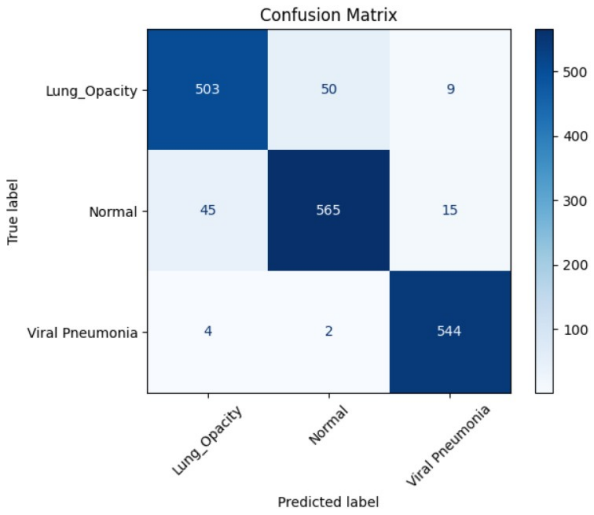

Fig. 5: Confusion Matrix of Base Model

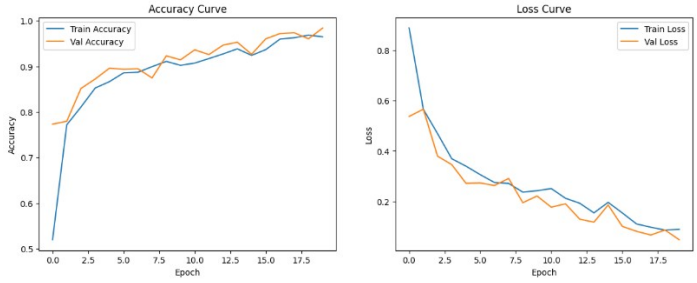

Fig. 6: Training and validation accuracy curves (left) and loss curves (right) of the Base model

To assess how each parameter influences model behaviour and performance in comparison to the base model, a thorough comparison is conducted across various training configurations. Important variables are compared to the baseline architecture, including input size, learning rate, batch size, dropout rate, early stopping settings, optimizer selection, model size, padding type, activation functions, and stride values. This comparison shows how changes in these components affect graph variations, convergence patterns, and stability, giving a better idea of how sensitive the model is to various design and training decisions.
The following sections present each comparison in a structured and sequential manner.

6.1 Learning Rate Comparison

When Learning Rate = 0.0001 is set in the Base Model, the model trains slowly and smoothly. The model successfully learns features from data,As shown in Figure 6,

accuracy increases with each epoch, and loss steadily declines. Both the training and validation performances improved as expected, and the training curves remain stable. The model can converge at a safe, regulated rate thanks to this learning rate.
After changing the learning rate to 0.1, a visual comparison of the results obtained.

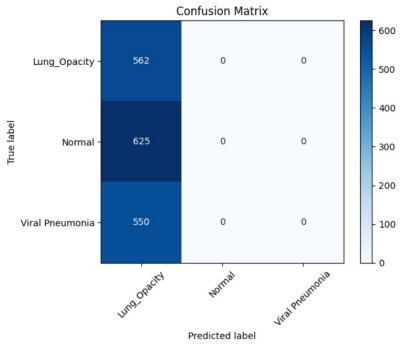

Fig. 7: Confusion matrix of the model with LR=0.1

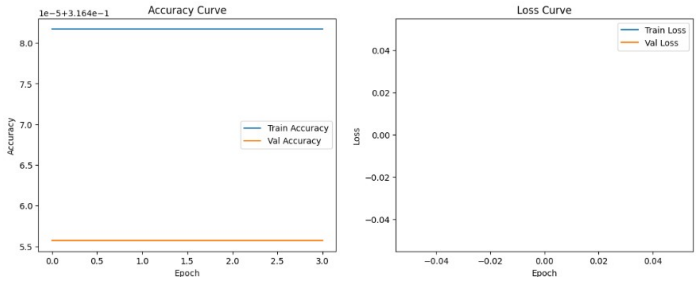

Fig. 8: training and validation accuracy curves (left) and loss curves (right) of the model with LR=0.1

The model gets unstable. Training fails to converge, accuracy does not improve and loss fluctuates. Because the learning steps are too big and the model overshoots the ideal weights. This variant is unable to learn in comparison to the base model. In general, divergence rather than learning results from the high Learning Rate.
After changing the learning rate(0.000001) a visual comparison of the results obtained. This comparison clearly illustrates how the modified setting influences the model's training behaviour.

The model becomes too sluggish to learn when LR = 0.000001. Accuracy hardly increases, loss declines very slowly, and the model advances very little.This version underfits in comparison to the base model since the revisions are too little to significantly change the weights.Although it learns far too slowly, it is stable.

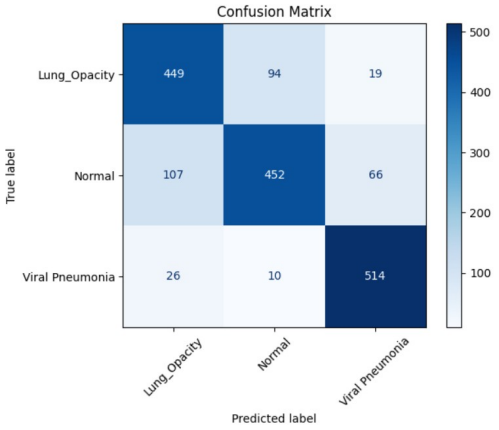

Fig. 9: Confusion matrix of the model with LR=0.000001

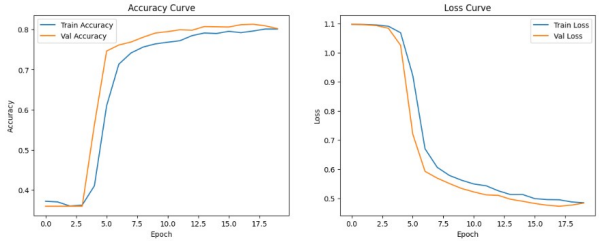

Fig. 10: Training and validation accuracy curves (left) and loss curves (right) of the model with LR=0.000001

6.2 Dropout Rates Comparison

With the dropout rate set to 0.5 in the base configuration, the model showed a proper balance in learning and stability. This was clearly reflected in the above training curves in 6 both training and validation losses went down smoothly without obvious divergence. The accuracy went up steadily across epochs, demonstrating that the regularization strength was strong enough to avoid overfitting but weak enough not to prevent the model from learning good representations. All in all, the dropout rate of 0.5 struck an effective balance in retaining useful information while suppressing co-adaptation among neurons.

By reducing the dropout rate to 0.2, the model picked up the learning sooner, as shown by the quicker increase in training accuracy. However, the weaker regularization increased its susceptibility to overfitting, resulting in less stable validation performance over time. Although the model benefited from retaining more active connections during each update, the reduced dropout constraint allowed the training accuracy to outpace validation accuracy more quickly, indicating a decline in generalization compared to the base configuration.

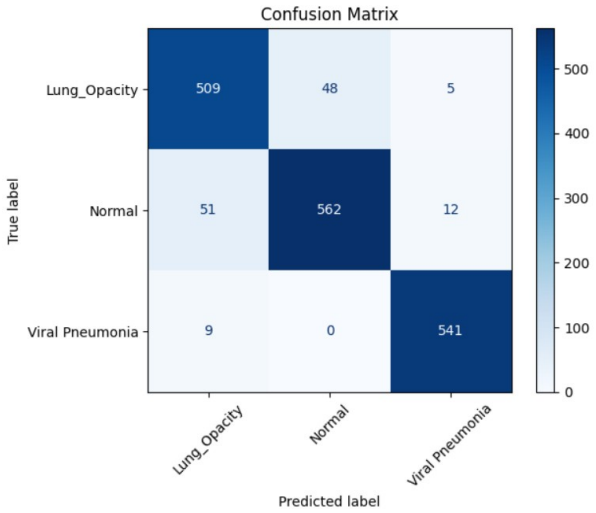

Fig. 11: confusion matrix of the model with Dropout rate = 0.2

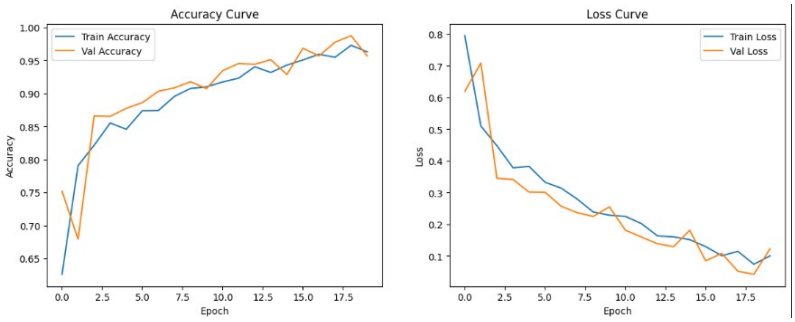

Fig. 12: Training and validation accuracy curves (left) and loss curves (right) of the model with Dropout rate = 0.2

Meanwhile, for a dropout rate of 0.7, the network learned markedly more slowly and was less accurate overall. This stronger regularization forced most of the network's activations to be dropped at any given time, preventing the model from developing strong feature representations. The convergence speed was reduced and the peak performance was lower compared to the base model, which was represented in both the accuracy and the loss curves. Excessive dropout prevented the model from retaining enough information to learn effectively and, therefore, underfit.

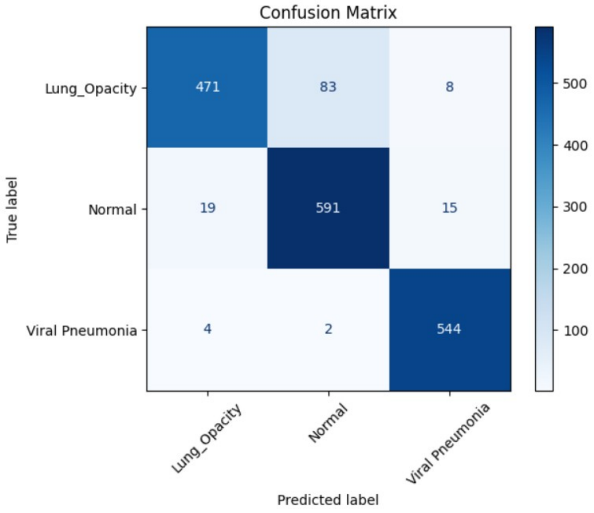

Fig. 13: Confusion matrix of the model with Dropout rate = 0.7

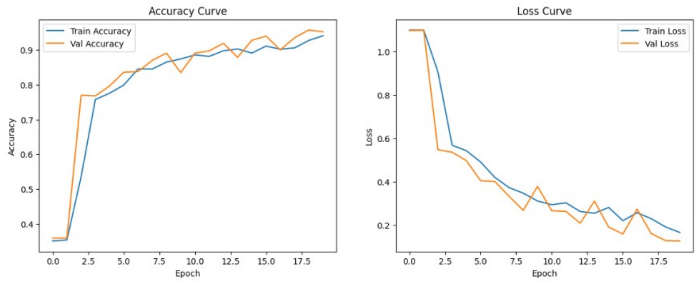

Fig. 14: Training and validation accuracy curves (left) and loss curves (right) of the model with Dropout rate = 0.7

6.3 Batch Size Comparison

For Batch Size = 32 in base model, it is shown in 6 that the training process is quite stable and smooth, providing a fairly good balance between computation efficiency and gradient stability. Over successive epochs, the model's well-averaged gradients offer consistency in loss reduction and dependability in accuracy increases.
In contrast to the Base Model, Batch Size = 8 The training curves become noticeably noisy because of the increased gradient variance. This configuration generally leads to increased loss and accuracy oscillations as well as delayed convergence. Smaller batches can collect more precise gradient information, but overall training efficiency and stability are lower than with the base model.

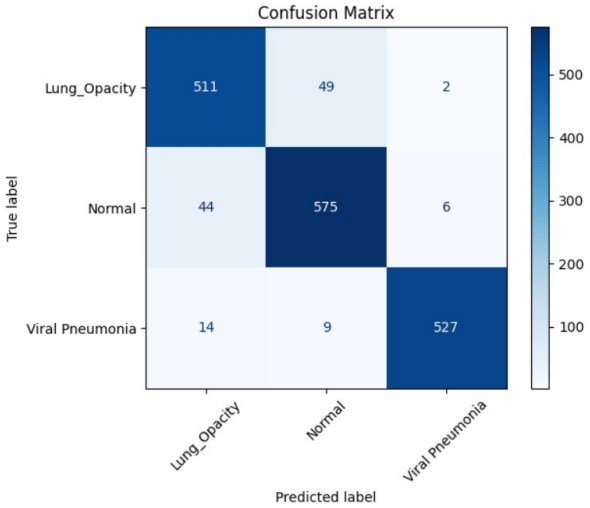

Fig. 15: Confusion matrix of the model with Batch Size = 8

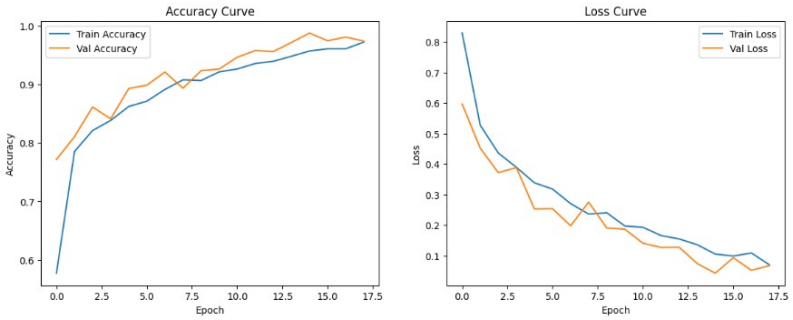

Fig. 16: Training and validation accuracy curves (left) and loss curves (right) of the model with Batch Size = 8

By increasing the batch size to 16, the training becomes moderately stable, the curves are smoother than with a batch size of 8 but not as consistent as with 32. It converges more slowly than the base configuration and presents minor fluctuations in validation performance. This suggests a balance between noise in the gradients and computational cost, although still not optimal when compared to the base model

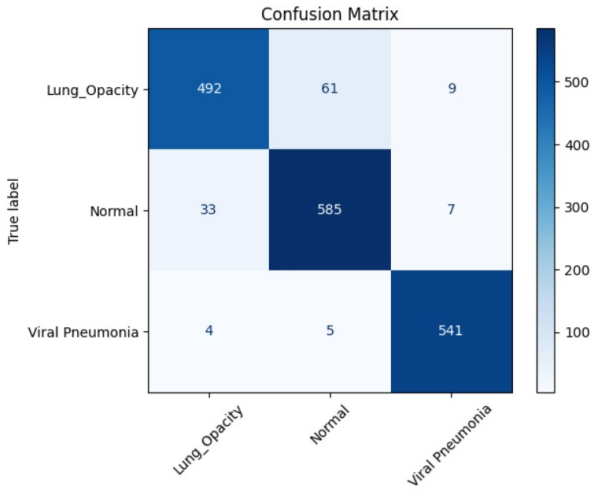

Fig. 17: Confusion matrix of the model with Batch Size = 16

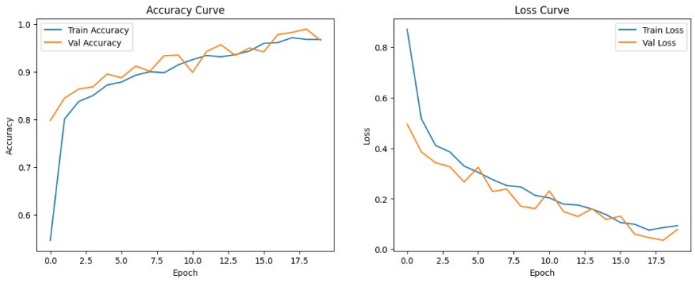

Fig. 18: Training and validation accuracy curves (left) and loss curves (right) of the model with Batch Size = 16

6.4 Comparison of Early Stopping Patience Values

The base model exhibited smooth and well-controlled convergence with early stopping patience set to 3. In the training curves, the training and validation losses were very consistent for most of the epochs, indicating stable updates of the parameters and low overfitting. Accuracy rose smoothly and learning progress attained a good balance with the intervention of early stopping. In that case, patience = 3 did allow ample time for recovery from transient performance deterioration and thereby reliable generalization without superfluous cycles of training.

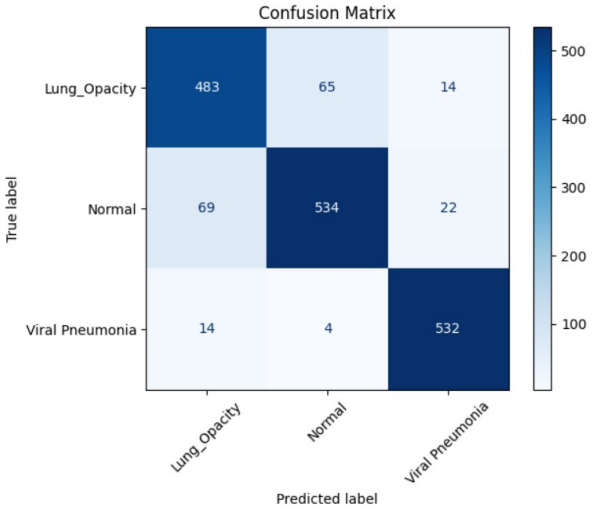

Fig. 19: Confusion matrix of the model with Early stopping: patience=2

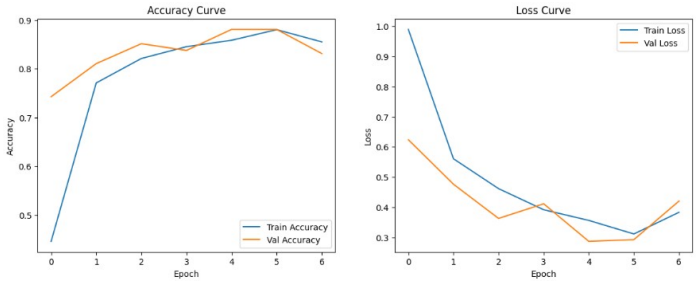

Fig. 20: Training and validation accuracy curves (left) and loss curves (right) of the model with Early stopping: patience=2

With the patience reduced to 2, the model became much more sensitive to short-term fluctuations in validation loss. Training terminated sooner, usually well before the validation accuracy reached its optimal region. The consequence of this behavior was a shorter training but slightly weaker convergence, since the model had fewer opportunities to refine the parameters after transient variations. Although the reduced patience minimized overfitting, it caused mild underfitting with marginally higher validation loss and less stability in accuracy compared to the base configuration.

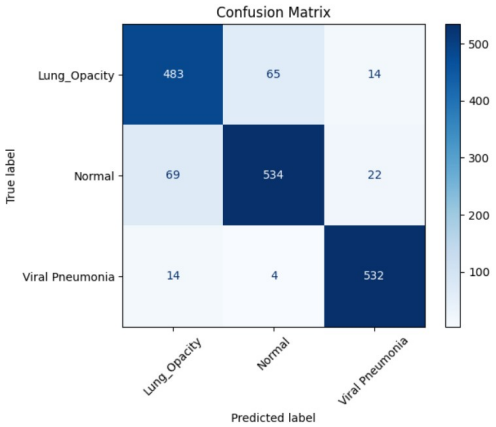

Fig. 21: Confusion matrix of the model with Early stopping: patience=5

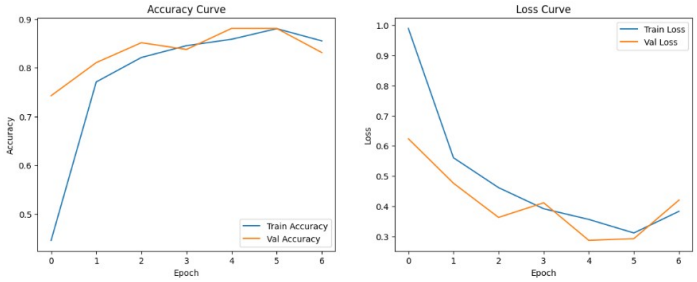

Fig. 22: Training and validation accuracy curves (left) and loss curves (right) of the model with Early stopping: patience=5

By contrast, the training with a patience value of 5 could run through multiple non-improving epochs. With this extended tolerance, the network was able to capture more refinements and sometimes reach higher peak accuracy compared to settings with shorter patience. Training and validation losses remained reasonably aligned, although slight divergence did appear at the later stages in model saturation. Nevertheless, generalization remained strong, with overall behavior indicative of better refinement, albeit at the cost of a marginally longer training period.

6.5 Optimizer Comparison

In the initial epochs, the Adam-based training showed in 6 a sharp increase in accuracy, suggesting quick data adaptation. With only slight variations, the training and validation curves maintained a close alignment and demonstrated steady generalization over time. Adam was able to maintain balanced updates across parameters, as evidenced by the loss values decreasing smoothly.Consistent saturation was observed

in the last epochs, with accuracy peaking and loss steadily decreasing. All things considered, the Adam setup produced rapid convergence and steady behaviour, attaining excellent performance with little variation between training and validation patterns.

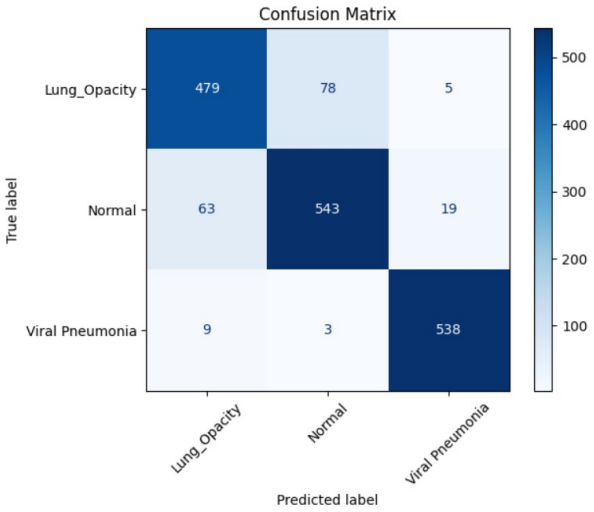

Fig. 23: Confusion matrix of the model with Optimizer: RMSProp

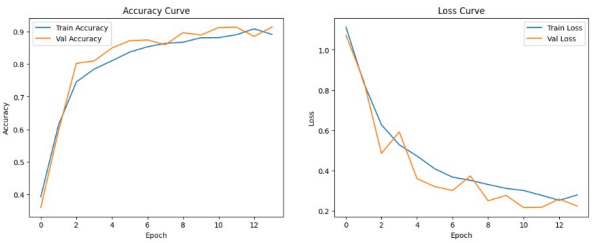

Fig. 24: Training and validation accuracy curves (left) and loss curves (right) of the model with Optimizer: RMSProp

The RMSprop run demonstrated a slower but more consistent increase in accuracy when compared to the Adam-based training. The model achieved a comparable performance level by later epochs, despite early learning progressing more slowly. Despite minor variations brought on by RMSprop's more sensitive update pattern, training and validation losses remained largely in line. There was very little misclassification between "Lung Opacity" and "Normal," according to the confusion matrix, which showed stable class separation. All things considered, RMSprop produced robust generalization with a more cautious convergence path, achieving similar accuracy albeit with somewhat more pronounced mid-epoch oscillations.

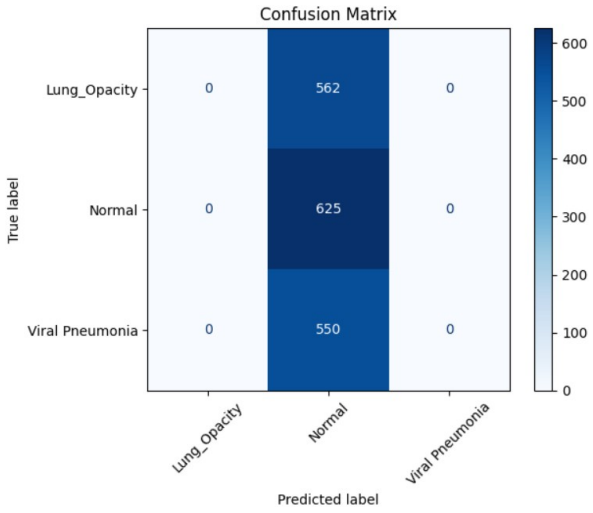

Fig. 25: Confusion matrix of the model with Optimizer: SGD with momentum = 0.9

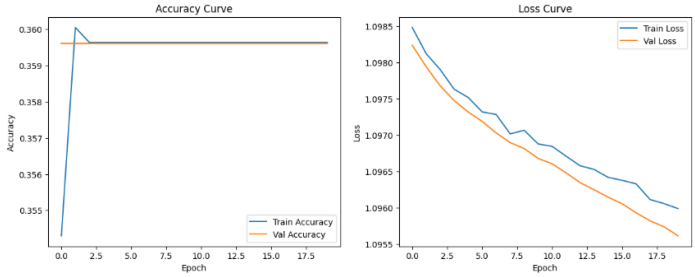

Fig. 26: Training and validation accuracy curves (left) and loss curves (right) of the model with Optimizer: SGD with momentum = 0.9

The run using SGD with momentum 0.9 demonstrated a notably slower initial improvement than the Adam-based training, which was indicative of the optimizer's more cautious update pattern. While Adam produced a sharp early rise in accuracy, SGD progressed more gradually before stabilizing. Because SGD relies on momentum-driven updates rather than adaptive learning rates, the loss curves under SGD showed somewhat larger mid-epoch fluctuations. The training and validation trends, however, continued to be fairly aligned, suggesting that generalisation was still intact. Although SGD with momentum ultimately produced a stable result, it took more training epochs to achieve the same level of refinement as Adam had earlier.

6.6 Input size comparison

Base Model (224 x 224) With accuracy increasing sharply in the early epochs and loss decreasing smoothly, the 224×224 input size resulted in quick and stable convergence.As shown in 6 The training and validation curves did not significantly diverge, indicating consistent learning. The model was able to capture finer spatial details.

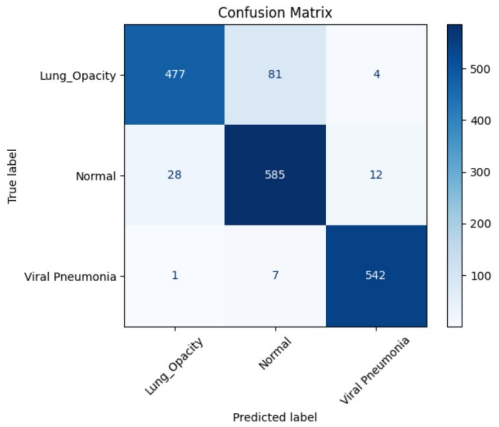

Fig. 27: Confusion matrix of the model with Input size 128*128

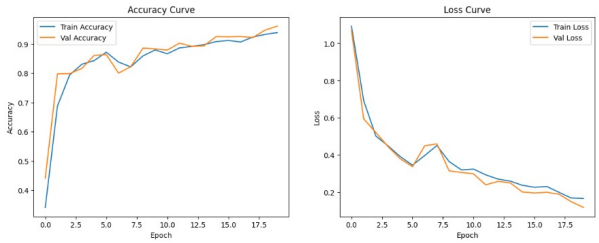

Fig. 28: Training and validation accuracy curves (left) and loss curves (right) of the model with Input size 128*128

Compared To 128×128, the model displayed somewhat slower early improvement and slight increases in fluctuation when 128×128 input was used. Class boundaries were slightly less accurate than in the 224×224 setup, but overall accuracy was still high. Although there was a slight decrease in refinement due to the decreased resolution's ability to provide fewer visual details, overall behaviour remained consistent and dependable across all classes.
Compared to 256 x 256, here epochs showed a slightly stronger refinement when using a 256x256 input, which was supported by the higher resolution's increased spatial detail. Stable generalization was indicated by a steady increase in accuracy and

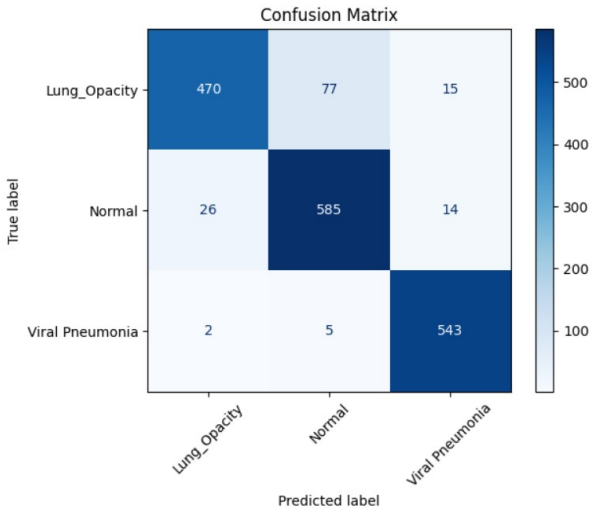

Fig. 29: Confusion matrix of the model with Input size 256*256

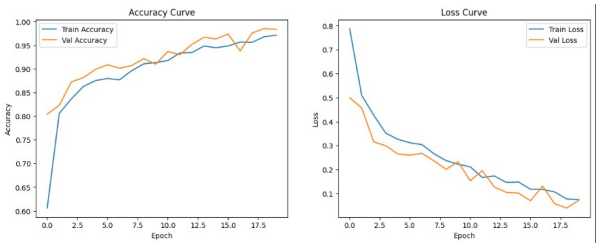

Fig. 30: Training and validation accuracy curves (left) and loss curves (right) of the model with Input size 256*256

close alignment between the training and validation curves. The larger input size increased feature complexity, causing slight oscillations, but overall performance was stable. Clearer class separation was suggested by the confusion matrix, demonstrating how the extra resolution enabled the model to capture more intricate patterns while maintaining reliability across classes.

6.7 Wider and Deeper model comparison

During training, the base VGG-style model demonstrated consistent learning. As the model adjusted to lung X-ray features, accuracy gradually increased in the early epochs, and the training and validation curves continued to closely match, suggesting good generalisation. In the later epochs, the model clearly reached a convergence point and loss decreased smoothly. In general, both datasets showed consistent behavior and balanced learning for the base model.

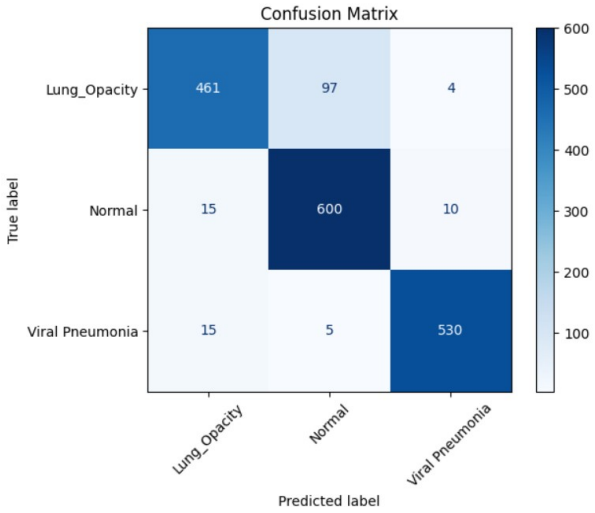

Fig. 31: Confusion matrix of the Deeper model

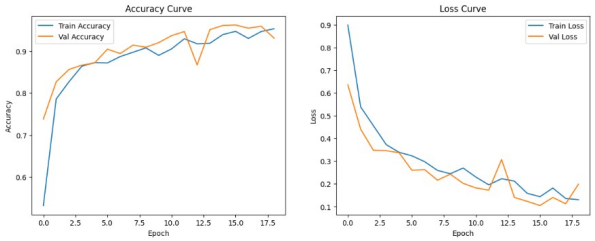

Fig. 32: Training and validation accuracy curves (left) and loss curves (right) of the Deeper model

With more convolutional layers, the deeper model adapted more slowly at first but improved steadily once the deeper layers stabilized. Despite minor mid-epoch fluctuations brought on by the deeper depth, its training and validation curves remained in line. Although it took longer to achieve a more refined convergence than the base model, the deeper architecture was able to capture more intricate patterns, particularly in "Lung Opacity" images.
With more filters per layer, the wider model outperformed the deeper model in terms of early accuracy improvement, but it was marginally slower than the base model. With only slight variations brought on by the higher number of parameters, its curves were generally smooth. Strong class separation and good alignment between training and validation trends were attained by mid-training. The wider model performed on par with or marginally better than the base model in the last epochs, delivering high accuracy and stable loss.

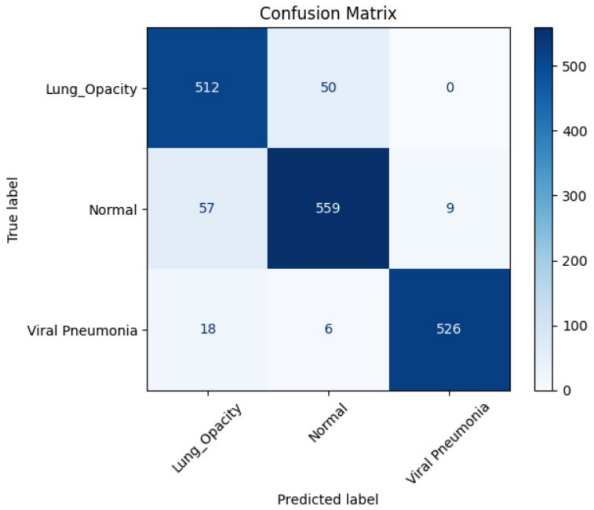

Fig. 33: Confusion matrix of Wider model

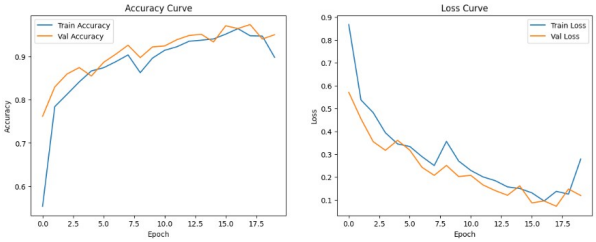

Fig. 34: Training and validation accuracy curves (left) and loss curves (right) of Wider model

6.8 Activation Function Comparison:

Training and validation curves for the base model using ReLU advanced steadily over epochs, demonstrating smooth and stable convergence. The losses stayed closely aligned, suggesting minimal overfitting, and ReLU enabled quick learning without significant gradient problems. The confusion matrix showed balanced class performance in 6, with only slight confusion between Lung Opacity and Viral Pneumonia, and accuracy increased steadily. All things considered, the ReLU-based baseline produced consistent training behaviour and clear evaluation curves.

Early on in training, the model demonstrated greater flexibility and learn a little bit more quickly with Leaky ReLU. A slight improvement in validation accuracy resulted from the presence of a small negative slope that prevented dead neurons and maintained continuous gradient flow. The confusion matrix showed fewer false negatives, particularly for viral pneumonia, and the curves fluctuated slightly but stayed steady.

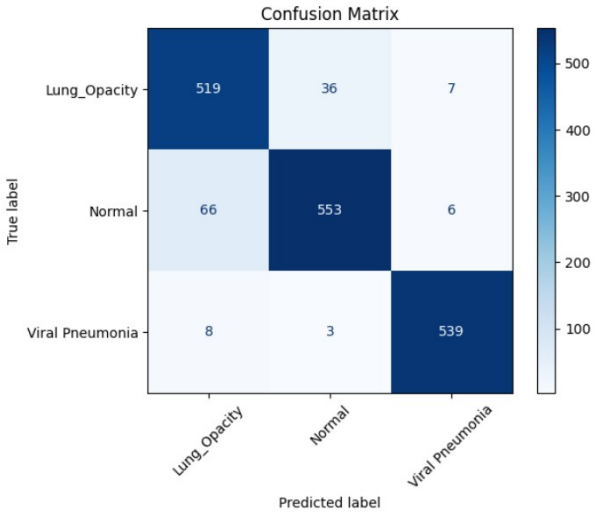

Fig. 35: Confusion matrix of the model with Activation Function : leaky ReLU

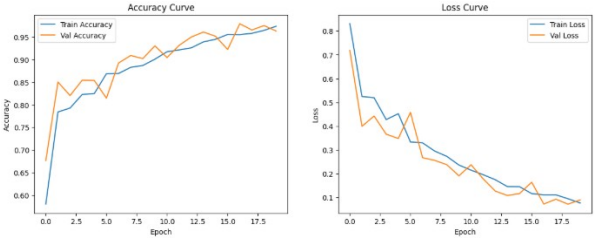

Fig. 36: Training and validation accuracy curves (left) and loss curves (right) of the model with Activation Function : leaky ReLU

Compared to the base model, this activation provided slightly stronger generalization and better feature extraction.
Because of gradient saturation in deeper layers, the tanh model trained more slowly and had smoother but flatter learning curves. Tanh decreased overfitting, but it also made it harder for the model to recognise intricate X-ray patterns, which led to a lower peak accuracy when compared to the ReLU variations. More misclassifications were found in the confusion matrix, especially between viral pneumonia and normal lungs. Tanh generally produced learning behaviour that was stable but less expressive.

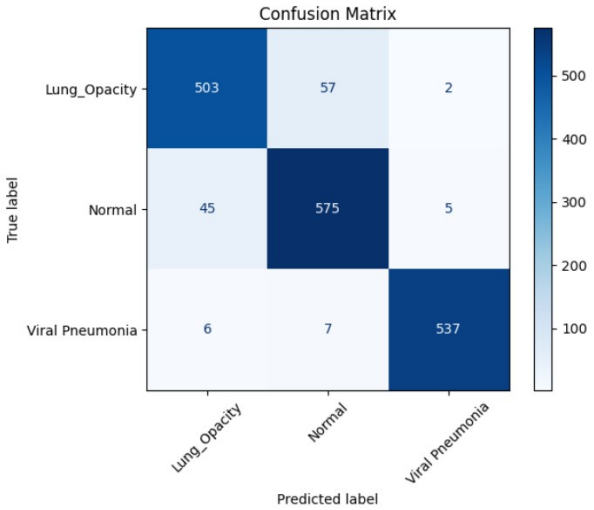

Fig. 37: Confusion matrix of the model with Activation Function : Tanh

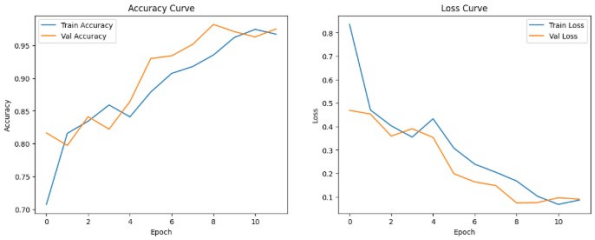

Fig. 38: Training and validation accuracy curves (left) and loss curves (right) of the model with Activation Function : Tanh

6.9 Padding Comparison:

Throughout training, the base model—which employed the same padding—showed balanced and seamless convergence. The network was able to learn stable representations from the chest X-ray images because the feature maps retained more edge and texture information due to the same padding preserving spatial dimensions. Consistent gradient flow and little overfitting were indicated by the training and validation curves' close alignment and lack of abrupt fluctuations. The confusion matrix showed good class separation, particularly between Normal and Lung Opacity, and accuracy steadily increased. Overall, clean learning behaviour with dependable generalization was produced by the basic "same padding" configuration.

Because the spatial dimensions were reduced after each convolution, the model behaved differently with valid padding. The network focused on more central and high-contrast areas of the chest X-ray as a result of the quicker feature compression. Consequently, convergence became somewhat more sensitive, especially in the early

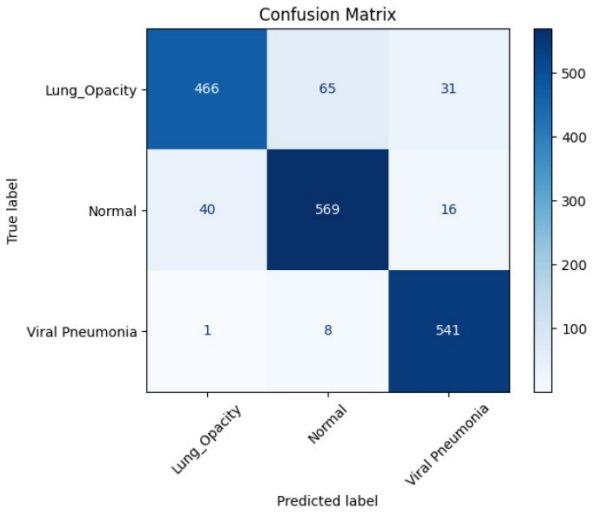

Fig. 39: Confusion matrix of the model with Padding: valid

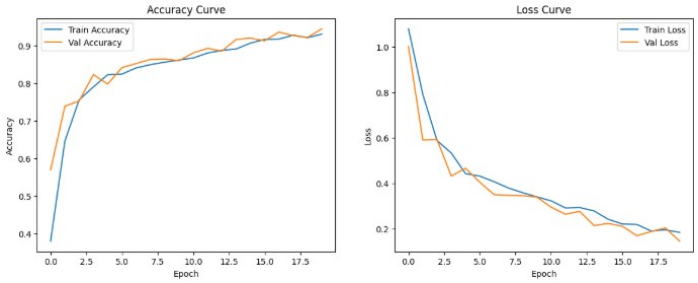

Fig. 40: Training and validation accuracy curves (left) and loss curves (right) of the model with Padding: valid

epochs, and the training curves displayed slightly sharper changes. Because smaller feature maps provide less contextual information close to the borders, validation accuracy occasionally varied even though the model trained effectively. Although there were a few more misclassifications in borderline cases, the confusion matrix showed good performance. In comparison to the base model, valid padding generally promoted compact feature extraction but resulted in more variable generalization.

The model that was trained using the same and valid padding once more showed the most stable behaviour, preserving uniform spatial dimensions and avoiding excessive feature loss in deep layers. Compared to the valid-padding version, the training and validation curves were smoother, with a steady increase in accuracy and a controlled decrease in loss. The same padding aided in maintaining edge-level details that are crucial for identifying minute lung abnormalities like mild opacities. Better retention of spatial context was indicated by the confusion matrix, which displayed more distinct

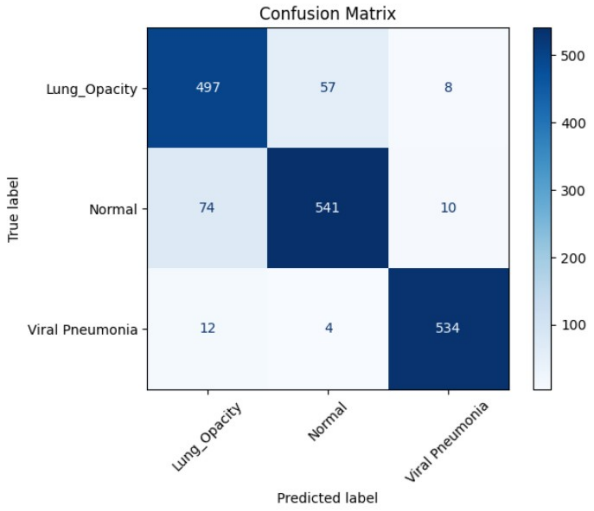

Fig. 41: Confusion matrix of the model with Padding: valid and same

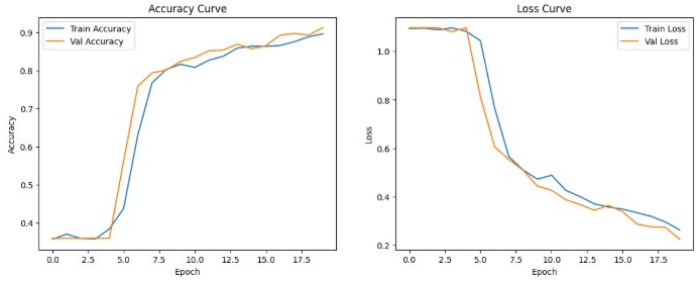

Fig. 42: Training and validation accuracy curves (left) and loss curves (right) of the model with Padding: valid and same

boundaries between the three classes. Among the compared padding methods, same padding generally yielded the most reliable and broadly applicable results.

6.10 Stride Comparison:

The base VGG style model displayed smooth and stable learning all throughtout training. From the accuracy graph, both the training and validation accuracy increased gradually and remained closely aligned across nearly all epochs. This tight overlap point out strong generalization and confirms that the model was not overfitting. The loss curve also shows a steady decline in both training and validation loss, with no major fluctuations or divergence between the two. The model merged smoothly in the later epochs, illustrating consistent feature extraction and balanced optimization. Overall, the base model established a consistent performance baseline with stable curves, clear convergence, and no signs of instability.

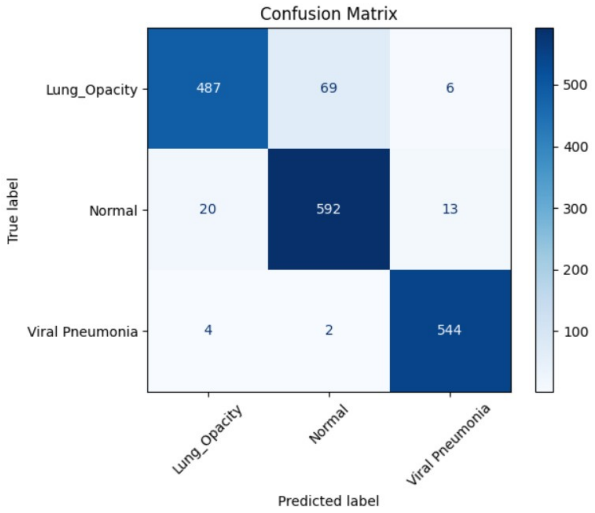

Fig. 43: Confusion matrix of the model with Stride = 3

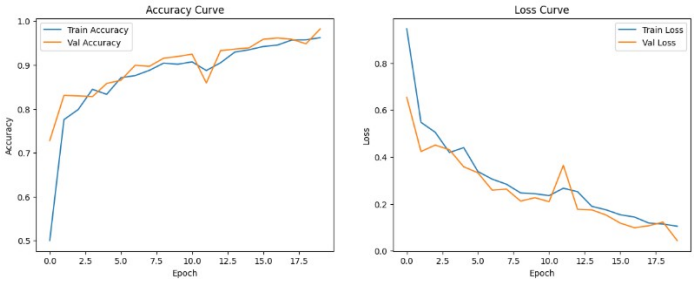

Fig. 44: Training and validation accuracy curves (left) and loss curves (right) of the model with Stride = 3

Compared to the base model, the stride 3 variant introduced a controlled mid-level down sampling effect. While the base model preserved full spatial detail longer, the stride 3 model reduced feature map resolution earlier in Block 3, increasing the receptive field. Training curves for the stride 3 model typically become slightly smoother in the mid-epochs because the reduced spatial complexity makes optimization easier. Validation accuracy trends also tend to stabilize earlier, since the model processes broader contextual information sooner than the base architecture. Despite this difference, the general behavior remains close to the base model: both maintain parallel training and validation curves, though the stride 3 version may converge with slightly reduced fluctuation due to its more efficient feature compression.
The stride 2 model differs from the base architecture by shifting most of the down sampling into the convolutional layers, making the reduction in spatial resolution learned rather than fixed. While the base model relies on Max Pooling for dimension reduction,

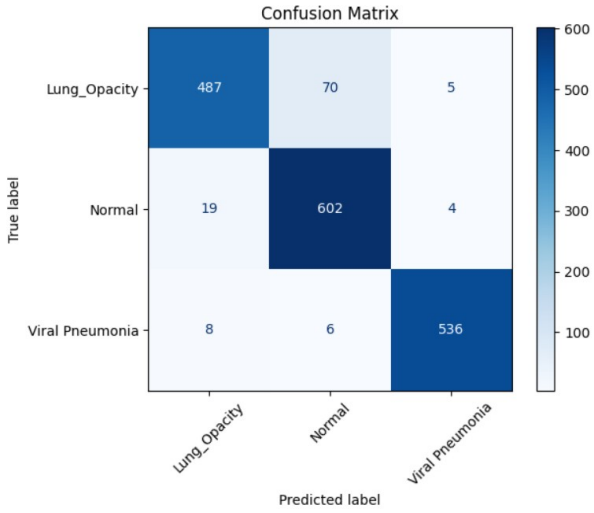

Fig. 45: Confusion matrix of the model with Stride = 2

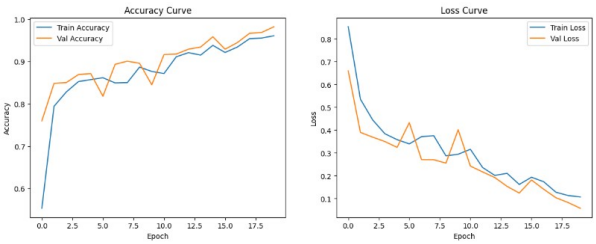

Fig. 46: Training and validation accuracy curves (left) and loss curves (right) of the model with Stride = 2

this model uses stride 2 convolutions to gradually shrink feature maps while preserving more fine-grained detail.Training curves for this version generally show slower early learning than the base model, since the model needs a few additional epochs to stabilise after each stride 2 reduction. However, once stabilised, accuracy begins increasing more rapidly and validation curves remain tightly aligned with training curves, indicating strong generalization.
Compared to the base model, this design shows better retention of subtle lung textures and avoids overly aggressive pooling-based shrinkage, resulting in stable convergence and improved detail preservation.

6.11 Complete Parameter Evaluation Table

Table 2: Comprehensive Comparison of All Model Variations

Category	Base Setting	Variant	Observed Effect
Learning Rate	0.0001	0.1	Diverged; unstable learning
		0.000001	Very slow; underfitting
Dropout Rate	0.5	0.2	Faster learning; slight overfitting
		0.7	Heavy regularization; underfitting
Batch Size	32	8	Noisy gradients; unstable
		16	Moderate stability; slower than base
Early Stopping	3	2	Training stops too early; mild underfitting
		5	Allows refinement; longer training
Optimizer	Adam	RMSprop	Slower early learning; stable
		SGD (0.9)	Slow improvement; more epochs needed
Input Size	224×224	128×128	Lower detail; slight accuracy drop
		256×256	Higher detail; minor improvement
Model Size	VGG (base)	Deeper	Slower start; better fine features
		Wider	Faster learning; similar or improved accuracy
Activation	ReLU	Leaky ReLU	Slightly better generalization
		Tanh	Saturation; lower accuracy
Padding	Same	Valid	More fluctuations; reduced context
		Mixed	Stable; similar to same padding

7 Discussion

The results of the present paper indicate that CNN-based models can classify the images of the chest X-ray with a high degree of accuracy in case the architecture and hyperparameters are balanced. The original VGG-style model was shown to be stable in convergence and different learning rates, dropout, batch size, optimizer, activation function, and input size were used to validate the sensitivity of medical image classification to configuration. Bigger and deeper models offered further enhancements with more features (including detailed ones) and a Leaky ReLU activation function facilitated gradient flow compared to Tanh. On the whole, in the context of the experiments, it should be noted that the well-selected parameters are a prerequisite of reliable classification of the chest X-rays and not an architectural change. However, it is important to mention that there exist a number of limitations. The data, though clinically relevant, has only three classes and is not diverse. There are even classes that overlap in visual and this may still be a problem to the model like Lung Opacity and Viral

Pneumonia. The data is also obtained in the publicly available repositories, which implies that there might be differences in the quality of the images, positioning of the patient, and the equipment that could create latent bias. It utilizes experiments on one architecture family (VGG-style) and does not allow to compare it to other modern architectures such as EfficientNet or DenseNet. Automated classification entails the use of medical images, which renders significant ethical issues. Models that are trained on small or imbalanced datasets have a high risk of biased preference towards particular groups of patients and acting as diagnostic errors. Clinical judgment should not undergo automated predictions. On the contrary, it should be used as decision-support systems by radiologists. Medical images should be properly anonymized and handled in a secure way to ensure that patient data is not abused. Lastly, implementation of any model into a clinical setting must be preceded by strict validation, disclosure of model weaknesses and ongoing observation to prevent damaging effects.

8 Conclusion and future work

In this study, we implement a CNN model based on VGG16 to classify chest X-ray images. Common classifications are Normal, Viral Pneumonia and Lung Opacity. According to the experiments, the model learned patterns in data set effectively and received right results without extreme overtraining. By varying hyperparameters, such as learning rate, dropouts, batch size, optimizers (or learning strategies), activation functions, input sizes and so on, we can see clearly how each choice affects the behaviour of our model. Summary In general terms, wider and deeper models capture more useful features while requiring more time to converge.As the model excelled in various applications, these two cases still gave it no end of trouble because of the similarity in their visual appearance. Formidable, is the dataset size and imbalance in classes that could hamper how good the model performs with wholly new images. For further enhancements, even larger and more balanced data sets would prepare the model to generalize better. Further pre-processing techniques (eg, image enhancement or more sophisticated augmentation) may allow the model to recognize these subtle differences between similar diseases. Recent architectures such as attention-based models or CNN-Transformer hybrids may also be experimented to further increase the performance.

We suggest in future work that a wider variety of data and significantly larger scale data collections will be needed to assure broad environmental robustness. Wrap-around approaches with advanced augmentation techniques, contrast normalization and region-directed preprocessing can be used to enhance discrimination among visually similar disease. Towards the future, the introduction of attention mechanisms, hybrid CNN-Transformer models or lightweight architectures could offer even greater contextual grounding while paying attention to issues of computational efficiency. Ability is necessary to let the model interpretable we show to the model in a simplified form.We ultimately have to deploy and verify the model in actual.

References

[1] Shamrat: Fine-tuned mobilelungnetv2 for multi-class chest x-ray classification. International Journal of Intelligent Engineering and Systems (2022). Implements MobileLungNetV2 with transfer learning and achieves AUC 91.6 but suffers from high computational complexity

[2] Nahiduzzaman: Cnn-elm based multiclass classification of chest x-ray images. International Journal of Intelligent Engineering and Systems (2022). Proposes CNN-ELM for identifying 17 lung diseases; performance limited by insufficient training data in some classes

[3] Mann: Mmf-dtl: Multi-modal fusion of deep transfer learning for cxr image classification. International Journal of Intelligent Engineering and Systems (2023). Uses VGG16, InceptionV3, and ResNet50; suffers from overfitting especially with ResNet152 due to limited training data

[4] Chen: Pcsanet: Pyramidal convolution and shuffle attention network for thorax disease classification. International Journal of Intelligent Engineering and Systems (2023). PCSANet enhances multi-scale feature extraction but shows low AUC for infiltration

[5] Xu: Group-wise spatial attention and label co-occurrence dependency for thoracic disease classification. International Journal of Intelligent Engineering and Systems (2023). Uses DenseNet with GWSA and LCD modules; performs well with sufficient data but low specificity affects negative class accuracy

[6] Wang: Thorax-net: Resnet152-based attention-assisted network for thoracic disease detection. International Journal of Intelligent Engineering and Systems **17**(6), 26 (2024) https://doi.org/10.22266/ijies2024.1231.03 . Affected by overfitting due to imbalanced class distribution in ChestX-ray14 dataset