"Deep Learning for Medical Image Classification: A Comprehensive Study"

Berker Senol[†], Roya Ghamari[‡]

***Abstract*—The COVID-19 pandemic has shown how crucial it is to diagnose the virus as soon as possible and accurately. Although several techniques have been proposed to use X-ray imaging to differentiate COVID-19 from non-COVID pneumonia, these techniques lack a comprehensive comparative examination. The objective of this research is to assess the performance of several deep learning and machine learning architectures in the classification of chest radiograph images into three groups: normal circumstances, non-COVID-19 pneumonia, and COVID-19. In order to maximize performance, we investigated a number of models, such as bespoke CNNs, a custom CNN with an attention mechanism, ResNet-34, ResNet-50, and DenseNet-121, using two different image preprocessing methods. In addition to demonstrating the relative efficacy of these models, our investigation looks at potential biases by comparing the algorithm's classification accuracy across various patient demographics. The results show notable performance differences, with DenseNet-121 proving to be very successful, followed by the CNN with an attention mechanism, which also demonstrated robust performance, particularly in discriminating fine details in the images. Although ResNet models in comparison to DenseNet-121 are more computationally efficient and may be preferable in resource-constrained settings, DenseNet-121 is more suited for situations where higher accuracy is paramount. Our research adds to ongoing efforts to improve COVID-19 diagnostic methods through modern imaging technology and offers valuable information about the strengths and weaknesses of machine learning techniques used in medical diagnostics.**

***Index Terms*—Lung Disease Detection, Neural Networks, Convolutional Neural Networks, DenseNet, ResNet, Image Classification.**

I. Introduction

The precise classification of chest radiograph images into categories such as COVID-19, pneumonia (non-COVID), and normal conditions is a critical challenge in medical diagnostics, intensified by the ongoing global pandemic. This study is driven by the need for robust diagnostic tools that can quickly and accurately differentiate these respiratory diseases. We engage in a comprehensive evaluation of several convolutional neural network (CNN) models, including ResNet-34, ResNet-50, and DenseNet-121, along with two custom CNN architectures, one incorporating an attention mechanism, to determine their diagnostic efficacy under various preprocessing techniques.

The urgency of developing automated diagnostic methods capable of distinguishing between COVID-19 and other forms of pneumonia has been a focus of recent research efforts. Previous studies such as those by Wang et al. [1], Cohen et al. [2], [3] have demonstrated the potential of deep learning solutions to identify COVID-19 from imaging studies. However, these studies often lack a uniform testing ground as they employ varied datasets and experimental conditions, leading to inconclusive comparative results across different models.

In this context, our work aims to fill the gap by conducting a methodical comparison of well-established and novel neural network architectures on a consistent dataset. This not only provides a direct performance comparison but also assesses each model's computational efficiency and viability for practical application. In particular, we improve DenseNet-121 with an attention mechanism, in order to refine its pattern recognition capabilities, which could be crucial to discern subtle features in X-ray images indicative of specific diseases.

The findings of this study are geared towards assisting healthcare systems in choosing suitable deep learning models for deployment in real-world clinical settings, particularly where computational resources are limited. Furthermore, our results contribute to the broader academic and professional discourse on improving AI-driven diagnostic tools, offering insights that could facilitate the development of more sophisticated and scalable solutions.

This paper is structured to first introduce the state of the art in automated chest X-ray analysis, followed by a detailed account of our methodologies including data preprocessing and model training. Subsequent sections delineate the architecture of each CNN employed, present a comprehensive evaluation of their performance, and discuss the implications of our findings. We conclude with reflections on the potential expansions of this research and its relevance to ongoing and future technological advancements in medical diagnostics.

II. Related Work

Recently, there has been a significant focus on the precise detection of lung diseases, including COVID-19 and both viral and bacterial pneumonia, utilizing sophisticated methods in computer vision and soft computing. Wang et al. (2020) [1] introduced COVID-Net, a specialized deep convolutional neural network designed to identify COVID-19 cases from chest X-ray images. They enhanced accessibility by making COVID-Net open-source and releasing a benchmark dataset containing 13,975 X-ray images. COVID-Net demonstrated superior performance compared to established models like ResNet-50 and VGG-19, with a reported accuracy of 93.3%. Similarly, Cohen et al. (2020) [3] employed the DenseNet architecture to assess COVID-19 severity using lung images,

[†]Department of Information Engineering, University of Padova, email: {berker.senol}@studenti.unipd.it

[‡]Author two affiliation, email: {roya.ghamari}@studenti.unipd.it

training its feature extraction layers on non-COVID-19 chest datasets before classifying disease severity using a smaller COVID-specific dataset. Despite numerous efforts to detect lung disease from X-ray images, comparing these results to pinpoint the optimal architecture remains challenging due to the varied techniques and datasets employed. While some researchers leveraged pre-trained models, others developed models from scratch. This paper seeks to fill this gap by evaluating well-known neural network architectures on lung disease prediction tasks to determine the most effective approach. Ismael et al. (2021) [4] enhance COVID-19 detection in chest X-rays by employing an ensemble of CNNs, integrating models like ResNet50, InceptionV3, and Xception, as detailed in their study. This approach achieves a high accuracy of 95.6% by leveraging diverse architectural strengths, though it presents challenges in terms of computational demands, highlighting a trade-off between diagnostic accuracy and deployment feasibility in resource-constrained settings.Narin et al. (2020) [5] "Automated detection of COVID-19 cases using deep neural networks with X-ray images" published in Computers in Biology and Medicine.] advanced the field of COVID-19 detection using chest X-rays by developing an automated system that leverages deep neural networks. Their methodology involves the utilization of pretrained convolutional neural networks (CNNs), specifically ResNet50, InceptionV3, and MobileNet, to distinguish between COVID-19 and other pneumonias. While their model achieved high diagnostic accuracy, they noted the importance of expanding the dataset to enhance the robustness and applicability of their tool in real-world clinical settings. Apostolopoulos and Mpesiana (2020) [6] Deep learning techniques for detecting COVID-19 using chest X-ray images" published in Applied Intelligence] leveraged deep learning models to detect COVID-19 from chest X-ray images. Utilizing transfer learning, they employed several pretrained networks such as VGG19 and MobileNetV2, achieving high diagnostic accuracy. Their research highlights the potential of deep learning in enhancing rapid COVID-19 detection. However, they acknowledge limitations due to the dataset's size and diversity, which could impact the models' generalization capabilities across different populations. Haval I. Hussein et al. (2023) [7] proposed two new lightweight CNN-based models optimized for early COVID-19 detection from X-ray images. These models offer efficient performance with reduced computational demands, making them ideal for devices with limited resources. They tested these models on a large dataset of chest X-rays, achieving high accuracy rates of 98.55% and 96.83% for binary and multiclass classifications, respectively. Their work demonstrates the potential for deploying high-accuracy, low-resource models in clinical settings to aid rapid COVID-19 diagnosis. In our work, we tried to address the gaps in current methodologies that primarily focus on single-model evaluations or limited preprocessing approaches. Our study not only identifies DenseNet-121 as the most accurate model but also highlights the practical applicability of CNNs with attention mechanisms for enhanced detail recognition, setting a new benchmark for future research in medical image analysis.

III. Processing Pipeline

In our study, we designed a comprehensive processing pipeline tailored to the classification of chest radiograph images into three distinct categories: normal, non-COVID-19 pneumonia, and COVID-19. Figure 1 shows our processing pipeline.

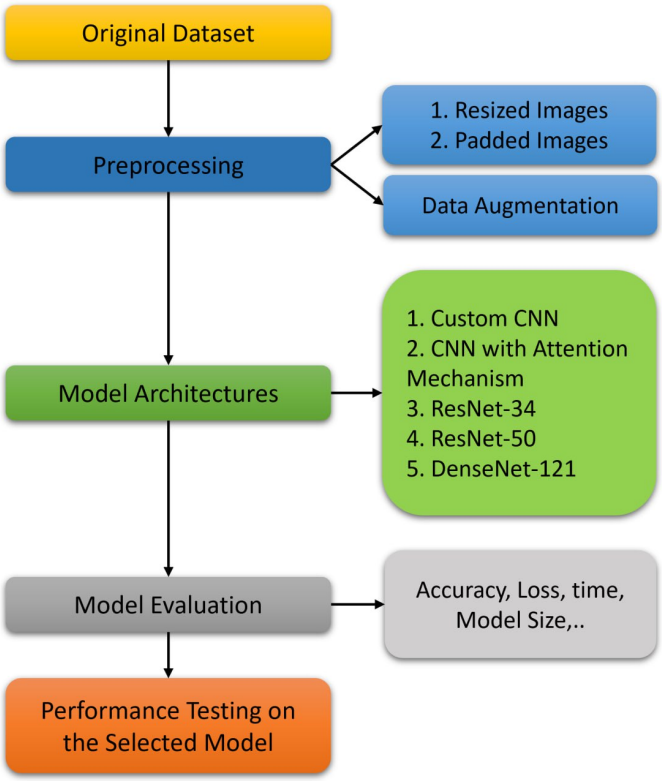

Fig. 1: Processing Pipeline

The preprocessing step involved two strategies: resizing images to a uniform dimension of 224x224 pixels or padding images to the same dimensions while maintaining their aspect ratio. Each image was normalized to scale pixel values between 0 and 1. Data augmentation techniques such as random flipping, brightness adjustment, and rotation were applied during training to enhance the model's ability to generalize from the training data to new, unseen images. This approach was crucial for developing a model that performs reliably across diverse clinical settings and imaging conditions. For the training phase, we implemented custom convolutional neural networks (CNN) alongside more sophisticated architectures like ResNet and DenseNet, which were trained separately on both resized and padded datasets. Our custom CNN model incorporated multiple convolutional and pooling layers, followed by dense layers to classify the images effectively. An attention mechanism was integrated into a second CNN model to enhance feature sensitivity in distinguishing between the classes. ResNet and DenseNet architectures were employed to leverage their advanced residual and densely connected

structures for robust feature extraction and classification. The models were evaluated using both training and validation datasets to tune the parameters and prevent overfitting. Early stopping was used to halt training once the validation loss ceased to improve, thereby ensuring optimal model performance without unnecessary computation. The best-performing model, DenseNet-121 trained on padded images, was selected based on its superior accuracy and loss metrics on the validation set. This model was subsequently used to predict the test dataset, providing a final evaluation of its diagnostic capability.

IV. Signals and Features

The dataset consists of 4575 posteroanterior chest X-ray images, evenly distributed across the three classes, with 1525 images per class. Some of the challenges with the dataset included varied image sizes and the width to height ratio differences among the classes. Figure 2 shows a sample from the dataset.

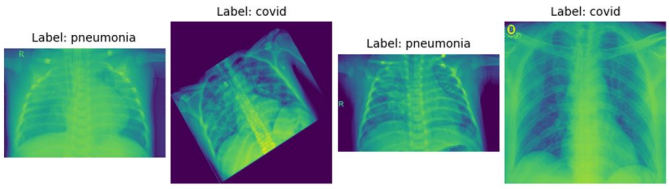

Fig. 2: Sample of the input data

1) **Resizing Images:** All images were resized to a uniform dimension of 224x224 pixels. This process helps standardize the input size for our models but has the potential drawback of distorting images.
2) **Padding Images:** another idea was to pad images in order to preserve the original aspect ratios, and at the same time to achieve a square shape. This method maintains the true proportions of the images but introduces its own form of bias with the padded areas potentially affecting the model's learning patterns.

All images were also converted to grayscale to reduce computational complexity and focus on structural features essential for effective classification. A sample of resized image and padded image is shown in Figure 3.

In addition to these preprocessing steps, we employed augmentation techniques to our train data in order to enhance the robustness of our models. These included random flipping, brightness adjustment, and rotation. Such augmentations are crucial as they introduce variability into the training process, simulating different imaging conditions and thus improving the model's ability to generalize. We utilized the train_test_split function from the sklearn.model_selection library to manage our dataset effectively. The dataset was split into training and testing subsets, with 80% of the images used for training and the remaining 20% for testing. The training set was further divided into a training subset (80% of the training set) and a validation subset (20% of the training set). This

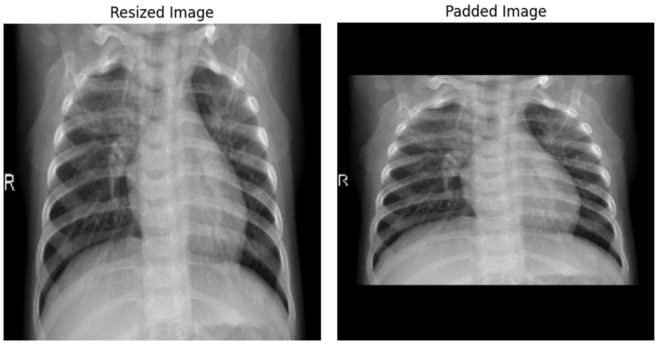

Fig. 3: Samples of resized and padded images

structure allowed for continuous evaluation of the model's performance and adjustments during the training phase to prevent overfitting and ensure the model's effectiveness on unseen data.

V. Learning Framework

In this section, we propose different model architectures that we used as our learning framework.

1) Custom CNN

Architecture: The architecture consists of three convolutional layers followed by max-pooling layers to reduce the spatial dimensions and extract significant features. The first convolutional layer applies 32 filters of size 3x3 with a ReLU activation function. This is followed by a max-pooling layer that reduces the dimensionality of the feature maps by half. Subsequent layers increase the number of filters to 64, enhancing the model's capability to capture more detailed features in the input images. Each convolutional layer is followed by a max-pooling layer and then the model flattens the feature maps to form a long feature vector that is then processed through a dense layer with 64 neurons. The output layer consists of three neurons, corresponding to the three classes, with a softmax activation function. The architecture of the model is shown in Figure 4.

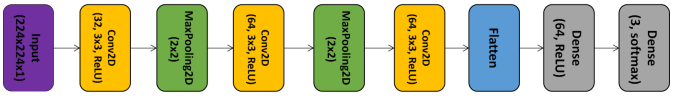

Fig. 4: Architecture of custom CNN

Learning Strategy: The model utilizes the Adam optimizer, chosen for its effective adaptive learning rate capabilities, and is set with a learning rate of 0.001. It uses sparse categorical cross entropy as the loss function, which is appropriate for our multi-class classification setup. The model is trained over a predefined maximum of 15 epochs, with the aim to continuously decrease validation loss and enhance classification accuracy throughout the training phase.

Parameter Optimization: Parameter optimization is facilitated by the implementation of an early stopping callback. This mechanism monitors the validation loss during training, and if there's no improvement observed for five consecutive

epochs, it terminates the training process. This not only prevents the model from overfitting but also ensures that the model retains the best possible weights when the validation loss was at its lowest. The model has over 11 million trainable parameters, structured to handle the complexities of medical image analysis efficiently.

2) **CNN with Attention Mechanism**

Architecture: The model starts with sequential convolutional layers: two sets of 32 and 64 filters of size 3x3 followed by max pooling, then two deeper sets of 128 filters to capture more complex features. Each convolutional step includes ReLU activation for non-linearity. The core innovation lies in the Squeeze-and-Excitation (SE) block following the last convolutional layer, which applies global average pooling to compress spatial dimensions to a single vector per channel, enhancing inter-dependencies between channels. This vector is then processed through two dense layers, transforming it into a re-scaled channel-wise attention map that gets multiplied back to the original feature map to emphasize informative features and suppress less useful ones. The modified feature map then passes through flattening and dense layers, concluding with a dropout layer for regularization before reaching a softmax output layer that predicts the three classes. Figure 5 shows the architecture of the model.

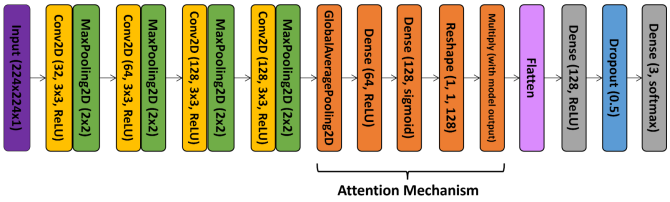

Fig. 5: Architecture of CNN with attention mechanism

Learning Strategy: The same as our custom CNN model, this model uses the Adam optimizer with a standard learning rate of 0.001 and a maximum of 15 epochs.

Parameter Optimization: The early stopping with a number of patience equal to 5, combined with a dropout rate of 50% in the final layers, helps prevent overfitting by enhancing the model's ability to generalize. The architecture totals 2,616,643 trainable parameters, providing a balance between computational efficiency and the capacity to learn complex patterns in medical imaging data.

3) **ResNet**

Architecture: The ResNet-34 model starts with a Conv2D layer with 64 filters of size 7x7, stride 2, followed by batch normalization and ReLU activation.This is followed by a MaxPooling2D layer with a pool size of 3x3 and stride 2, which helps reduce the dimensionality of the data. For Residual Blocks in stage 2 to 5, comprises multiple "basic blocks" for each stage. Each basic block implements two Conv2D layers where the first convolution might have strides set to (2, 2) for down-sampling based on the stage. Each convolution is followed by batch normalization and ReLU activation.The shortcut (or identity) connections in each block add the input to the block to its output, which helps in training deeper networks by mitigating the effects of diminishing gradients. Followed by Global Average Pooling in the deep convolutional layers to reduce each feature map to a single number by averaging out each feature. As output layer, a Dense layer with a softmax activation function to classify the input into one of three classes.

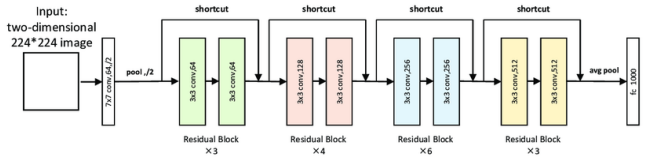

Fig. 6: Architecture of ResNet-34

The ResNet-50 model begins with a Conv2D layer with 64 filters of size 7x7, stride 2, followed by batch normalization and ReLU activation. Follows with a MaxPooling2D layer with a pool size of 3x3 and stride 2 to help reduce spatial dimensions early in the network. For Stage 2 to 5, the model utilizes a series of "convolutional blocks" and "identity blocks." Then Convolutional blocks adjust the dimensionality through strided convolutions and increase the depth using three layers (1x1, 3x3, 1x1 convolutions) where the first and last are bottleneck layers that reduce and then increase dimensions, minimizing computational load. Identity blocks do not alter the dimensions; they use three convolutional layers (as in the convolutional block) to deepen the feature maps without reducing size. Each block includes shortcut connections that bypass one or more layers.Global Average Pooling, placed near the end to reduce each feature map to a single vector per map, effectively summarizing the output of the convolutional network. As an output layer, a Dense layer with a softmax activation function that outputs probabilities for each of the three classes like in the ResNet-34 model.

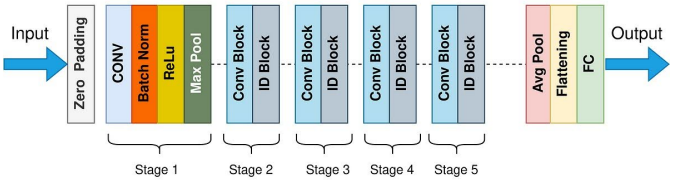

Fig. 7: Architecture of ResNet-50

As architecture-wise, ResNet-50 is deeper than ResNet-34, employing 50 layers compared to 34. This depth includes more bottleneck layers which help in managing the model's complexity and computational cost.While ResNet-34 uses two-layer blocks, ResNet-50 uses three-layer bottleneck blocks for most parts of its architecture, enhancing its ability to learn from more features with a reduced risk of overfitting on complex image classification tasks like those involving medical images.

Learning Strategy: For both models, sparse categorical crossentropy loss function has been utilized which is suitable for multi-class classification problems where the labels are provided as integers. We employed the Adam optimizer with an initial learning rate of 1e-5.Lastly, model checkpointing has been used to save the best model based on the validation accuracy parameter to ensure that the model does not lose any progress during training. We also have chosen epoch as 40 because the previous epoch size of 15 was not enough for the model to learn the fundamental information from the data.

Parameter Optimization: ResNet-34 and ResNet-50 had the total number of parameters 21,305,475 and 23,587,587 respectively. For both ResNet models during the hyperparameter tuning part, parameters such as the number of layers, size of filters, and learning rate were initially set based on common practices and literature benchmarks for similar tasks.The model‘s hyperparameters were refined iteratively based on its performance on a held-out validation dataset by early stopping with patience parameter chosen as 10, which was crucial for tuning without overfitting to the training data.

4) **DenseNet**

Architecture: Models start with before each convolutional layer, the feature maps are normalized (using Batch Normalization) to stabilize learning and accelerate convergence. Following normalization, the ReLU activation function introduces non-linearity, helping the network learn complex patterns. The core building block of DenseNet. Each dense block consists of multiple convolutional layers. In DenseNet-121, the structure of each layer in the dense block starts with a bottleneck layer (1x1 convolutions) that reduces the dimensionality, followed by a 3x3 convolution that extracts features. This arrangement helps in reducing the number of parameters. Layers within a block are connected such that each layer receives the feature maps from all preceding layers, fostering feature reuse. Transition layers placed between dense blocks, these layers control the dimensionality of the network. They typically involve a 1x1 convolution followed by 2x2 average pooling, reducing the size of the feature maps and thus the computational requirements for subsequent layers. Global Average Pooling, applied after the last dense block, transforms the feature map of each class into a single number by taking the average of each feature map. This step reduces the total number of parameters and helps in combating overfitting. Lastly, for the output layer, a dense layer with a softmax activation function computes the probability distribution over the target classes.

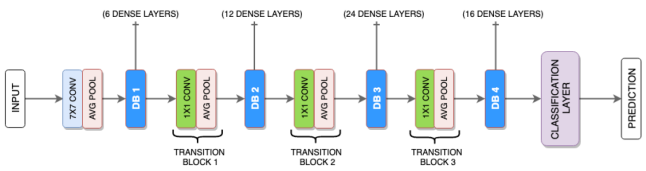

Fig. 8: Architecture of DenseNet-121

Learning Strategy: The model used ’sparse categorical crossentropy’ for loss function. Like the previous models, adam optimizer with a very low learning rate (1e-6) is employed, which adjusts the learning rate during training, providing an advantage in terms of converging faster and escaping local minima. To prevent overfitting, early stopping monitors validation loss and halts training if there‘s no improvement after a defined number of epochs (patience has been selected as 10 because of the lower learning rate selection).

Parameter Optimization: During the hyperparameter tuning part, parameters such as the number of layers, size of filters, and learning rate were initially set based on common practices and literature benchmarks for similar tasks. By having about 37,410,243 parameters, it was our most resource-intensive model. The model’s hyperparameters were fine-tuned through repeated adjustments, guided by performance metrics from validation dataset. By employing early stopping with a patience parameter set to 10, we tried to prevent overfitting during the training process.

VI. Results

A. Classification Performance Overview

As it was mentioned before, we employed two distinct preprocessing methodologies, resizing and padding. The two datasets we achieved were as input to various deep learning models including a custom CNN, a CNN enhanced with an attention mechanism, and variants of ResNet (ResNet-34, ResNet-50) along with DenseNet-121. The accuracy and loss on the validation set was used to determine the best preprocessing technique.

The results consistently showcased good overall accuracy and low loss across all architectures for both resized and padded datasets, surpassing the accuracy of 86% on validation set, indicating robust model performance. From the plots, it is evident that while the overall accuracies were satisfactory, certain fluctuations were noted. For instance, some models exhibited minor instabilities in terms of loss metrics, which were particularly evident in the initial training epochs. Especially, ResNet34 and ResNet50 architectures showed more fluctuations in accuracy and loss for both resized and padded datasets. These fluctuations stabilized as the models continued to learn, a trend observed with both the basic and advanced CNN architectures. This performance variability can be attributed to the initial learning phase where models adjust their weights significantly in response to the diversity of the training data. As the models’ weights begin to stabilize, so does their performance, leading to a consistent decrease in loss and improvement in accuracy as training progresses. Comparing all our models, DenseNet-121 represented a smaller gap between the training and validation losses and accuracies during the training process. The Figures 9 and 10 show these plots.

B. Comparative Model Evaluation

Performance and computational efficiency of our models with respect to both resized and padded data are shown in Table 1 in appendix. In terms of model performance, Custom CNNs

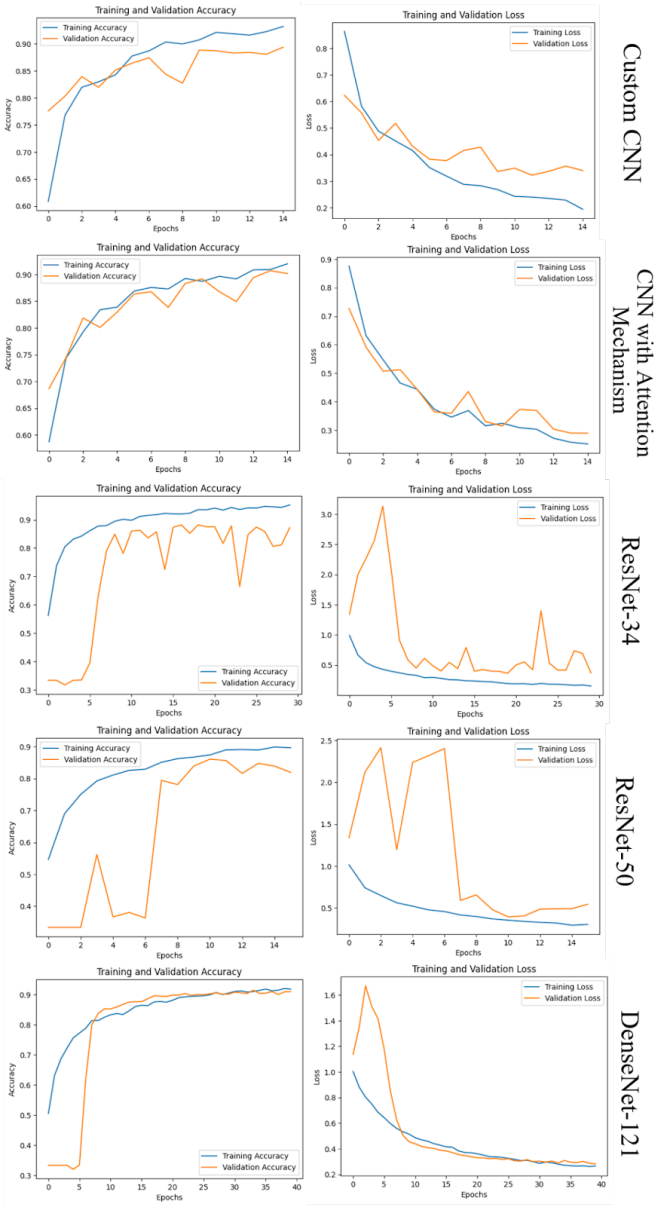

Fig. 9: Results for resized data

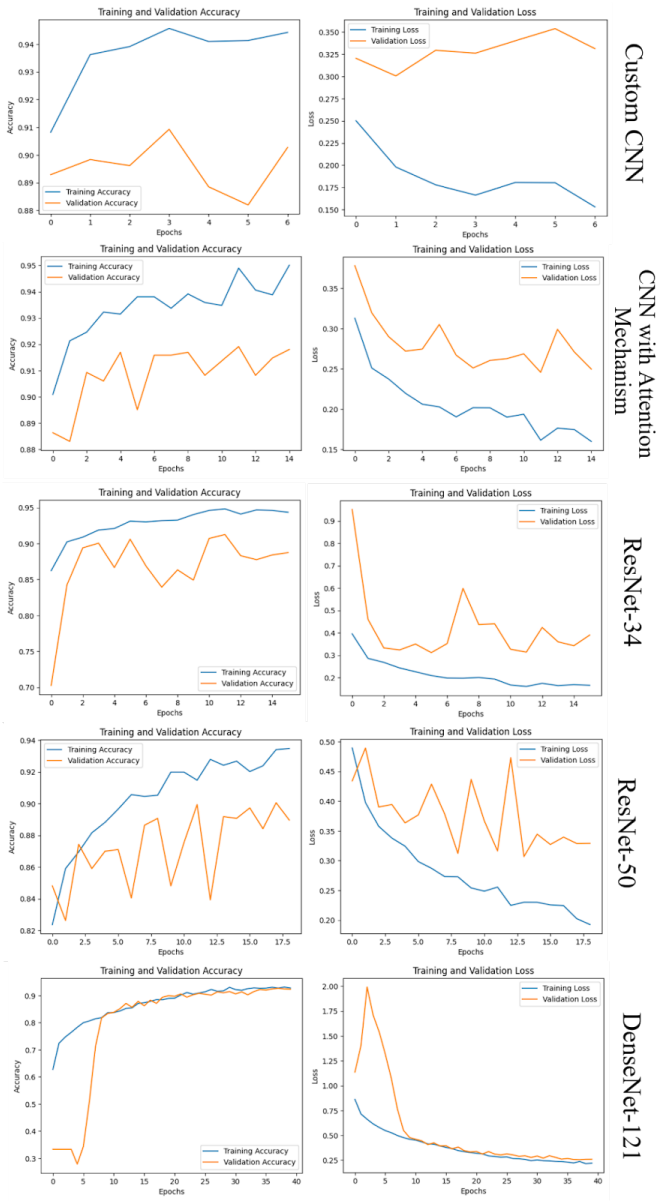

Fig. 10: Results for padded data

showed robust performance with slightly better validation accuracy and lower loss when trained on padded data compared to resized data. This indicates that maintaining the original aspect ratio might preserve critical diagnostic features in the images. CNN with Attention benefited from the padded dataset, achieving a higher validation accuracy (0.918 vs. 0.901) and lower validation loss (0.249 vs. 0.289). The attention mechanism likely enhanced the model's ability to focus on relevant features that are better preserved in padded images. ResNet Variants displayed mixed results; while ResNet-34 had a marginally better performance on padded data, ResNet-50 performed better on resized data. This suggests that the impact of image preprocessing may vary depending on the depth and complexity of the network architecture. DenseNet-121 demonstrated superior performance on the padded dataset across all metrics, reinforcing the advantage of using padded data to maintain the integrity of the original image structure in deeper networks.

In terms of computational efficiency, two important aspects of our evaluation were training time and model size. Especially, training times for ResNet-50 and DenseNet-121 were noticeably longer, especially for the scaled data, requiring more processing power. Although DenseNet-121 was the most accurate, its usefulness in resource-constrained environments may be limited because it had the greatest model size and longest training duration. The Custom CNN and CNN with Attention models were more effective, needing smaller model sizes and less training time, which made them appropriate for situations where quick diagnoses are crucial.

C. Optimal Model Selection and Testing

The DenseNet-121 model trained on padded data emerged as the superior choice based on validation accuracy and loss metrics and was selected for final testing to confirm its diagnostic accuracy under more stringent and unbiased conditions. The validation confusion matrix shown in Figure 11 for DenseNet-121 on the padded dataset using validation set illustrates effective classification across three categories. The model identified COVID and normal conditions correctly classifying 291 and 295 cases respectively, out of 305. It accurately detected 259 pneumonia cases, though there were 40 misclassifications with normal conditions. This highlights its strong performance, particularly for COVID and normal classifications, with some room for improvement in pneumonia detection.

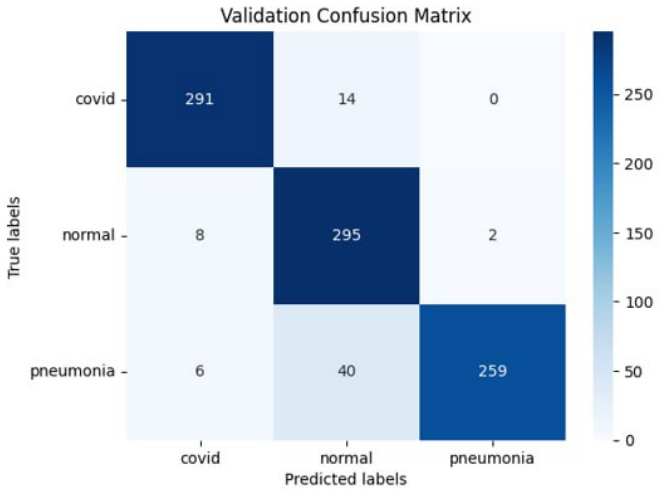

Fig. 11: Confusion Matrix of padded data for DenseNet-121

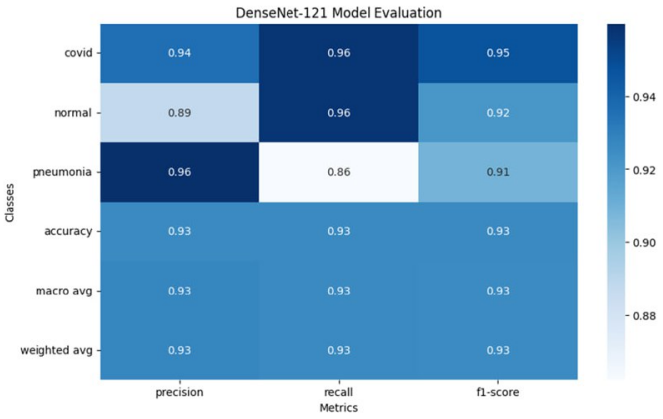

Fig. 12: Results obtained using test set

The test results confirmed the findings from the validation phase, with the model achieving high precision, recall, and F1-scores across all three categories: COVID, Normal, and Pneumonia. The results are shown in Figure 12. Overall, the DenseNet-121 model demonstrated an aggregate accuracy of 0.93 on the test dataset. The balanced scores across precision, recall, and F1 measures for each category affirm the model's comprehensive learning and predictive capabilities.

VII. Concluding Remarks

In this project, we investigated five deep learning architectures applied to two differently preprocessed datasets. These methods showcased promising accuracies.After training phase particularly DenseNet-121 showed decent performance due to its deep architecture and efficient feature learning from the preprocessed data. However, having over 37 million parameters, it was computationally much more expensive than our other models. We also observed that our small sized CNN with attention mechanism achieved the second-best performance in terms of validation accuracy and loss. Future work for this project we could involve exploring ensemble methods to combine the strengths of various individual architectures, and also implementing more sophisticated attention mechanisms to enhance feature recognition in X-ray images. We faced a challenging dataset which needed rigorous preprocessing and potentially sophisticated model architectures to enhance robustness and accuracy. During training phase, especially for finding the correct learning rate for the more complex models was one of the most difficult task that we had. We have tried different learning rate schedulers. We implemented both high and low learning rates and adaptive learning rates that changes during training phase. We have seen the effects of the different learning rates and their effects on the models and sometimes fluctuations on the validation loss and accuracy values. We saved the trained models and their related histories to use later in the demo representation.

References

[1] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, p. 19549, 2020.

[2] J. P. Cohen, P. Morrison, and L. Dao, "Covid-19 image data collection." https://doi.org/10.48550/arXiv.2003.11597, 2020. Accessed: 2024-07-03.

[3] J. P. Cohen, L. Dao, K. Roth, P. Morrison, Y. Bengio, A. F. Abbasi, B. Shen, H. K. Mahsa, M. Ghassemi, H. Li, and T. Q. Duong, "Predicting covid-19 pneumonia severity on chest x-ray with deep learning," *Cureus*, vol. 12, no. 7, p. e9448, 2020.

[4] G. Jia, H.-K. Lam, and Y. Xu, "Classification of covid-19 chest x-ray and ct images using a type of dynamic cnn modification method," *Computers in Biology and Medicine*, vol. 134, p. 104425, 2021.

[5] M. Heidari, S. Mirniaharikandehei, A. Zargari Khuzani, G. Danala, Y. Qiu, and B. Zheng, "Improving the performance of cnn to predict the likelihood of covid-19 using chest x-ray images with preprocessing algorithms," *International Journal of Medical Informatics*, vol. 144, p. 104284, 2020.

[6] S. Hira, A. Bai, and S. Hira, "An automatic approach based on cnn architecture to detect covid-19 disease from chest x-ray images," *Applied Intelligence*, 2020. Accepted: 9 October 2020.

[7] H. I. Hussein, A. O. Mohammed, M. M. Hassan, and R. J. Mstafa, "Lightweight deep cnn-based models for early detection of covid-19 patients from chest x-ray images," *Expert Systems with Applications*, vol. 213, p. 119900, 2023.

APPENDIX

TABLE 1: General evaluation of the models with datasets

Architecture and Data	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Training Time (s)	Size (MB)	Epochs to Train
Custom CNN resized	0.938	0.186	0.893	0.34	1109.13	42.46	15
Custom CNN padded	0.943	0.166	0.898	0.3	426.2	42.46	7
CNN with Attention resized	0.939	0.188	0.901	0.289	924.05	9.98	15
CNN with Attention padded	0.961	0.117	0.918	0.249	913.16	9.98	15
ResNet-34 resized	0.93	0.21	0.875	0.362	2121.52	81.27	30
ResNet-34 padded	0.904	0.255	0.906	0.311	1137.41	81.27	16
ResNet-50 resized	0.881	0.338	0.861	0.394	1526.04	89.98	16
ResNet-50 padded	0.922	0.23	0.891	0.306	1483.49	89.98	19
DenseNet-121 resized	0.931	0.223	0.91	0.281	4290.28	142.71	40
DenseNet-121 padded	0.945	0.179	0.923	0.258	4305.44	142.71	40