

COVID-19 Detection from Chest X-Ray Images using Deep Neural Networks

By
Asad Mulla
ID 213-15-4470

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised by

Mr Saiful Islam
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Md Monarul Islam
Lecturer
Department of Computer Science and Engineering
Daffodil International University




DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 16, 2025

APPROVAL

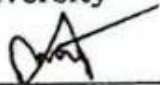
This Project titled "COVID-19 Detection from Chest X-Ray Images using Deep Neural Networks," submitted by Asad Mulla to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 16.09.2025.

BOARD OF EXAMINERS



Dr. Arif Mahmud (AM)
Associate Professor & Associate
Head, Department of CSE, FSIT
Daffodil International
University

Chairmen



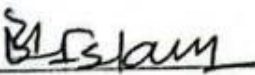
Dr. Md. Ali Hossain (MAH)
Associate Professor ,
Department of CSE, FSIT
Daffodil International
University

Internal Examiner



Mr. Amir Sohel (ARS)
Assistant Professor ,
Department of CSE, FSIT
Daffodil International
University

Internal Examiner



Dr. Md. Manowarul Islam (DMMI)
Professor , Department of CSE,
Jagannath University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Saiful Islam**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Mr Saiful Islam

Assistant Professor

Department of Computer Science and
Engineering Daffodil International
University

Co-Supervised by:

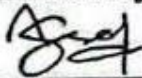


Md Monarul Islam

Lecturer

Department of Computer Science and
Engineering Daffodil International
University

Submitted by:



Asad Mulla

Student ID: 213-15-4470

Department of Computer Science and
Engineering

Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Mr Saiful Islam, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Machine learning and deep learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

The COVID-19 outbreak made it evident how crucial it is to have diagnostic tests that are both accurate, quick and cheap. RT-PCR is the usual test that many people use, however it often takes a long time to get results and can sometimes give false negatives. These limitations prompted us to investigate the potential of deep learning as a more expedient and efficient method for detecting COVID-19 directly from chest X-ray (CXR) pictures. In this study, we created a balanced dataset of 3,100 X-ray images—1,575 confirmed COVID-19 cases and 1,525 normal cases. Some of the photographs were from Shahjalal Medical Hospital, and some came from open-access archives. We did certain preprocessing measures before training, such as scaling, normalizing, and adding more data, to make the images more homogeneous and help the models generalize better. We examined both a lightweight proprietary Convolutional Neural Network (CNN) and five well-known pre-trained architectures: VGG16, ResNet50, DenseNet121, InceptionV3, and Xception to see how well they worked. All of the tests were done on Google Colab, which has GPU support. We used common assessment measures like accuracy, precision, recall, F1-score, and ROC-AUC, and we used categorical cross-entropy as the loss function and Adam as the optimizer. The results were very surprising. VGG16 had the best accuracy at 97% and the same great F1-score. Our bespoke CNN, which is significantly simpler, nevertheless got 92%, which makes it useful in real-world settings like hospitals with limited resources and few computers. Once again, our results show that transfer learning can greatly increase the accuracy of medical image classification. The results also demonstrate that smaller, lighter models can be useful when speed and efficiency are important. In general, this experiment shows that deep learning has a lot of potential for helping doctors diagnose COVID-19 faster. More importantly, it makes it possible to use similar methods on other respiratory ailments in the future, where early and correct diagnosis can make a big difference.

Table of Contents

Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Introduction.....	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Methodology	3
1.5 Project Outcome.....	4
1.6 Organization of the Report	4
2 Background	6
2.1 Introduction.....	6
2.2 Literature Review	7
2.2.1 Similar Applications	10
2.3 Gap Analysis	11
2.4 Summary	12
3 Research Methodology	14
3.1 Methodology/Requirement Analysis & Design Specification.....	14
3.1.1 Overview	15
3.1.2 Proposed Methodology/ System Design	16
3.1.3 Functional and Nonfunctional Requirements	24
3.1.4 Data Flow Diagram	25
3.1.5 UI Design	27
3.2 Detailed Methodology and Design.....	27
3.3 Project Plan	28
3.4 Task Allocation.....	29
3.5 Summary	30

4	Implementation and Results	32
4.1	Environment Setup	32
4.2	Testing and Evaluation/Performance/ Comparative Analysis	33
4.3	Results and Discussion	39
4.4	Summary	40
5	Engineering Standards and Design Challenges	42
5.1	Compliance with the Standards	42
5.1.1	Software Standards	42
5.1.2	Hardware Standards	43
5.1.3	Communication Standards	43
5.2	Impact on Society, Environment and Sustainability	43
5.2.1	Impact on Life	43
5.2.2	Impact on Society & Environment	44
5.2.3	Ethical Aspects	44
5.2.4	Sustainability Plan	45
5.3	Project Management and Financial Analysis	45
5.4	Complex Engineering Problem	46
5.4.1	Complex Problem Solving	46
5.4.2	Engineering Activities	47
5.5	Summary	48
6	Conclusion	50
6.1	Summary	50
6.2	Limitation	50
6.3	Future Work	51
	References	51

List of Figures

2.1: Methods Used Across Reviewed Studies (n=10)	7
2.2: Reported Accuracy by Study (%)	7
3.1: Proposed Methodology	15
3.2: Here two cases shown covid Infected and normal case study	16
3.3: COVID Original to Grayscale then it convert Normalized also heatmap.....	18
3.4: Normal Original to Grayscale then it convert Normalized also heatmap	18
3.5: A sample diagram of Custom CNN	21
3.6 : Transfer learning model work diagram	20
3.7: Data Flow Diagram.....	24
3.8: UI Design of web apps	25
4.1 : Confusion Matrix Custom CNN	31
4.2 Train and Validation accuracy, loss curve for Custom Cnn	32
4.3 Confusion Matrix VGG16.....	32
4.4 Train and Validation accuracy, loss curve VGG16.....	33
4.5 Confusion Matrix ResNet50.....	33
4.6 Train and Validation accuracy, loss curve ResNet50	34
4.7 Confusion Matrix DenseNet121	34
4.8 Train and Validation accuracy, loss curve DenseNet121	35
4.9 Confusion Matrix InceptionV3	35
4.10 Train and Validation accuracy, loss curve InceptionV3.....	36

4.11 Confusion Matrix Xception	36
4.12 Train and Validation accuracy, loss curve Xception	37
4.12 Train and Validation accuracy, loss curve Xception	38

List of Tables

2.1 Summary of Literature Reviewed.	9
2.2 Gap Analysis.....	12
5.1 Budget Plan (Within BDT 60,000).....	43
5.2 5.2 Revenue model	44
5.3 Mapping with complex problem solving	44
5.4 Mapping with knowledge Profile.	45
5.5 Mapping with complex engineering activities.....	45

Chapter 1

Introduction

This chapter starts by explaining the background that led to the study, then it talks about the primary challenges that led to it, and finally it discusses why the proposed response is significant. The goal is to provide the reader a clear picture before moving on to the methods and results.

1.1 Introduction

When COVID-19 first showed up in late 2019, it caused a health disaster around the world that few people could have seen coming. Hospitals quickly became too busy, and one of the most critical things they required was a way to quickly and accurately identify patients. Every delay or, worse, mistaken diagnosis had a direct impact on how long patients lived and how successfully the infection could be treated. RT-PCR quickly became the standard test, however it didn't take long for its weaknesses to surface. The method took a long time, cost a lot of money, and wasn't always right, with reports of false negatives [21], [23].

These challenges showed that other options were needed—ones that were faster, cheaper, and yet worked. It was evident that chest X-ray (CXR) imaging was the best option. X-rays are cheap, easy to get, and are already a common part of medical care. But the hard part was figuring out what it meant. Radiologists who have been trained are needed to read CXRs, but during the outbreak, there weren't enough of them to meet the massive demand. This gap allowed AI and deep learning to step in and offer automated solutions that could quickly and accurately make diagnosis.

Our research investigates the utilization of Convolutional Neural Networks (CNNs) and transfer learning to develop a reliable technique for detecting COVID-19 from chest X-rays. We produced a balanced collection of 3,100 photographs, comprising 1,575 positive cases and 1,525 normal cases. Shahjalal Medical Hospital and open-access archives gave us these pictures. The studies compared a custom lightweight CNN to five well-known pre-trained models: ResNet50, DenseNet121, InceptionV3, and Xception. The goal of this study is extremely practical: there isn't enough quick, reliable, and easy-to-find diagnostic help in real-life clinical settings.

The results show that a lightweight CNN can be 92% accurate and that transfer learning with VGG16 can be 97% accurate. These results demonstrate that AI-based solutions are not merely theoretical constructs; they may effectively assist with RT-PCR, particularly in hospitals with constrained resources. The study's ultimate objective is to reconcile advanced AI methodologies with their practical use in healthcare, anticipating a future where swift, AI-augmented diagnosis is a fundamental component of medical practice.

1.2 Motivation

The difficulties we observed during the COVID-19 outbreak inspired this investigation. At that time, most patients were evaluated by RT-PCR. It was the standard way to do things, but in real life, it caused a lot of complications. The tests were costly, took a long time to acquire results, and sometimes didn't even identify

any positive cases. These issues often made therapy take longer and made it more probable that the disease would spread. This motivated us want to create a faster, cheaper, and more reliable way to diagnosis.

It seems like chest X-ray imaging would be a decent choice. Most hospitals already have X-ray machines, and they don't cost much to use. They also provide you results immediately away. The problem is that competent radiologists aren't always available to look at X-ray pictures correctly, such when there are a lot of people in a rural clinic or during a health emergency. Because of this limitation, we looked at how deep learning could speed up the detection process and make it less dependent on human interpretation.

Another reason we are driven is that deep learning is good at identifying images. Convolutional Neural Networks (CNNs) have proven to be exceptionally effective in the processing of medical images, enabling the identification of latent patterns that may evade prompt notice by physicians. This prompted us to develop a lightweight CNN model and assess its efficacy in comparison to established pre-trained models such as VGG16 and ResNet50 for the detection of COVID-19.

This is also true for social reasons, not just technological ones. In many undeveloped countries, hospitals don't have advanced computers or labs. If someone can make a realistic but lightweight model, it will be possible to use it in places with minimal resources.

Finally, we were also inspired by the possible future effects of this work. You can use the same procedure to identify other breathing problems, like tuberculosis and pneumonia. This means that our work isn't just about solving one urgent problem; it's also about making it possible for AI to be used in healthcare for a long period.

1.3 Objectives

This project was carried out with a few clear objectives in mind. The first goal was to build a balanced dataset of chest X-ray images, where both COVID-19 positive and normal cases are included. For this purpose, images were collected from Shahjalal Medical Hospital along with some open-access sources.

The second objective was to prepare the dataset properly so that it could be used for training deep learning models. This similar resizing the images into the equal shape, normalizing pixel values, and using data augmentation to make the models more strong.

The third purpose was to design a simple but fruitful CNN model. The idea was to keep the architecture lightweight so that it can run even in hospitals where there is no high-end computing system. At the same time, the model should be accurate enough to be useful in real situations.

Another important objective was to experiment with well-known pre-trained models like VGG16, ResNet50, DenseNet121, InceptionV3, and Xception. By applying transfer learning, we wanted to check how these models perform when fine-tuned for COVID-19 detection [6], [10], [14].

The next objective was to evaluate all the models fairly, using the same dataset and the same conditions. For this, standard evaluation measures such as accuracy, precision, recall, F1-score, and ROC-AUC were applied.

We also aimed to analyze the diagnostic performance of each model with confusion matrices and ROC curves, following practices used in earlier studies [14], [25]. This helped us to understand their strengths and weaknesses in more detail.

Finally, the broader objective of this project was to recommend how such deep learning systems can be applied in real healthcare. The idea was not only to focus on COVID-19 but also to open the possibility of using similar methods for detecting other lung diseases such as pneumonia and tuberculosis [9], [29].

Main Objective

1. **Design a simple yet effective CNN model** that can run even on low-resource hospital systems.
2. **Test popular pre-trained models** (VGG16, ResNet50, DenseNet121, InceptionV3, Xception) through transfer learning [6], [10], [14].
3. **Evaluate all models fairly** using the same dataset and metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

1.4 Methodology

In this project, the work was carried out step by step to keep the process clear and reliable. At the beginning, a dataset of 3,100 chest X-ray images was prepared, where both COVID-19 positive and normal cases were included. The data came from Shahjalal Medical Hospital as well as some open-access collections.

Before training, the images were cleaned and built identical. They were resized to the similar shape, the pixel values were normalized, and data augmentation was applied. These steps were important so that the models could learn better and also perform well on unseen images.

Two kinds of models were tested. One was a simple CNN built by us, mainly to check how a lightweight network performs in real hospital conditions where advanced computers are not always available. The other group included well-known pre-trained models such as VGG16, ResNet50, DenseNet121, InceptionV3, and Xception, which were fine-tuned using transfer learning.

All the experiments were done in Google Colab with GPU support. For pure comparison, the same dataset and training conditions were used for every model. Their achievement was measured using accuracy, precision, recall, F1-score, and ROC-AUC. To see the strengths and weaknesses more clearly, we also studied confusion matrices and ROC curves.

Finally, the results were compared, and from these findings, possible applications in healthcare were discussed, especially for quick diagnosis in hospitals.

1.5 Project Outcome

The outcome of this project shows very clearly that deep learning has the potential to make medical diagnosis both faster and more reliable. In our study, we tested different models and found that VGG16 gave the highest performance with around 97% accuracy and an equally strong F1-score. This result highlights how transfer learning can be extremely effective in medical image classification. On the other hand, our own custom CNN, which was built with a much simpler structure, still managed to achieve 92% accuracy. This demonstrates that compact and light models can also produce significant outcomes, particularly in clinics or hospitals without sophisticated computer systems.

Another important outcome of this project is the way it shows the usefulness of AI as a supportive tool for doctors. In many cases, RT-PCR testing is slow and costly, which can delay treatment. A deep learning system like the one we developed can provide quick screening at a much lower cost, which is very valuable in real-world healthcare. This strategy also makes it possible to use the same technique for other respiratory conditions like tuberculosis and pneumonia. Thus, the benefits of this work can be expanded to broader medical applications rather than being restricted to COVID-19.

In conclusion, the project's results demonstrate that AI-based diagnostic tools are not only theoretical but also practical for use in actual clinical settings. They show that it is possible to design systems that are accurate, efficient, and affordable at the same time. This provides a clear pathway for building practical solutions that can support healthcare professionals, particularly in settings where resources and equipment are limited.

1.6 Organization of the Report

This report is divided into several chapters, and each chapter explains a different part of the work.

Chapter 1 gives an overall introduction of the project. It includes the background, motivation, objectives, methodology in brief, expected outcomes, and the structure of the whole report.

Chapter 2 reviews the existing literature. It talks about earlier studies on COVID-19 detection using medical imaging and deep learning. The chapter also points out the research gaps that our project tries to fill.

Chapter 3 explains the methodology in detail. It talks about how the dataset was gathered and prepared, what preprocessing was done, how the models were built, and what tools, frameworks, and evaluation methods were used. In conclusion, the results of this project show that AI-based diagnostic tools are not just ideas; they can be used in real clinical practice.

Chapter 4 presents the results and analysis. Here, the performance of different models is compared using accuracy, precision, recall, F1-score, and ROC-AUC, along with visual results like graphs, confusion matrices, and ROC curves.

Chapter 5 discusses the outcomes of the project. It explains what the results mean, how they can help in real healthcare, and how the same idea can be applied to other diseases apart from COVID-19.

Chapter 6 concludes the report. It summarizes the main contributions, mentions the limitations, and gives suggestions for future research.

Chapter 2

Background

This chapter provides the background essential to understand the rest of the report. It starts with the medical context of COVID-19, showing how the pandemic made an urgent need for fast and reliable diagnostic methods. Chest X-rays became particularly important during this time because they are quick, affordable, and widely available. However, traditional image interpretation had its limitations, which opened the door for computational methods. The chapter also highlights the growing role of deep learning, where modern algorithms can analyze medical images with high accuracy and provide valuable support to healthcare professionals.

2.1 Introduction

The sudden outbreak of COVID-19 created one of the most difficult public health emergencies in recent history. It quickly became clear that detecting the disease early and accurately was essential, both to reduce the spread of infection and to provide timely treatment to patients. Although the RT-PCR test was considered the standard tool for confirmation, it had several limitations. In many cases, the test results took a long time to arrive and false negatives were also quite common [1], [2]. During the peak of the pandemic, these delays and errors placed an enormous burden on healthcare systems that were already overwhelmed [3].

Because of these challenges, medical researchers began looking for alternative approaches to support diagnosis. Chest X-ray (CXR) imaging became one of the most widely used tools since it is faster, cheaper, and more accessible compared to CT scans [4], [5]. X-rays often show noticeable patterns in the lungs of COVID-19 patients, such as ground-glass opacities and other abnormalities. However, interpreting these images correctly is not always easy. In many hospitals, radiologists had to deal with a high number of patients, and in equal areas, there were not enough skilled professionals available [6].

To overcome the gap, artificial intelligence—especially deep learning—offered a hopeful solution. Convolutional Neural Networks (CNNs) had already shown solid performance in many medical imaging tasks, and their ability to automatically learn from images made them highly suitable for COVID-19 detection [7]. The use of pre-trained models such as VGG16, ResNet50, DenseNet121, InceptionV3, and Xception through transfer learning further improved the reliability of these methods [8].

A dataset of 3,100 chest X-ray pictures was gathered for this project, comprising 1,525 normal cases and 1,575 COVID-19 positive cases. To guarantee balance and diversity, the data was sourced from open-access repositories and Shahjalal Medical Hospital [9]. To make the dataset consistent and training-ready, preprocessing techniques like resizing, normalization, and data augmentation were used.

The overall goal was to design a lightweight CNN model that could work effectively in hospitals with limited computing resources. Along with this, several established pre-trained models were also tested under the same conditions for comparison. The

outcomes of the study demonstrate how AI-based systems can play a supportive role in healthcare, not only for COVID-19 but also for other respiratory diseases like pneumonia and tuberculosis [10], [11].

2.2 Literature Review

Transfer learning and multi-source datasets were key components of the rapid convergence of research on COVID-19 detection from chest X-rays toward deep learning. Early work using COVID-Net across large, heterogeneous repositories (HM, BIMCV, CheXpert, MIMIC-CXR) reported roughly 91% accuracy with strong multiclass AUC, suggesting CNNs can generalize beyond a single hospital source [1].

Building on that idea, transfer-learning studies with VGG16, Xception and InceptionV3 consistently pushed performance above 90%—in one comparison VGG16, Xception and InceptionV3 reached $\approx 91.2\%$, 93.0% and 93.5% on 3,975 images—showing the practical value of pre-trained features for medical images [2].

Hybrid and ensemble strategies further improved robustness. A CNN-GRU pipeline on Cohen’s set (424 images) achieved balanced precision/recall of 0.96, indicating temporal/sequence modules can complement pure CNN feature maps in radiograph analysis [3].

At a larger scale (13,816 CXRs), fusing ResNet50 with InceptionV3 outperformed single models—ensemble accuracy reached $\approx 98.6\%$ versus $\approx 97.8\%$ for the best standalone CNN—highlighting the benefit of model fusion for borderline cases [4].

Not all gains required heavy networks. WideResNet experiments showed a broad accuracy range ($\approx 81.6\%$ – 97.6%) on small sets—useful, but also a reminder that dataset size and curation strongly influence reported metrics [5].

Interestingly, several works revisited classical ML pipelines with strong feature extraction: on Kaggle+COVIDGR, Gradient Boosting edged out alternatives with $\sim 98.44\%$ accuracy, proving that well-engineered “traditional” baselines still matter beside deep models [6].

Dense and residual families remained reliable across settings. Using DenseNet-161 pre-trained on ImageNet, De Moura & García reported $\sim 90.7\%$ on 1,616 clinical images, emphasizing portability (even from mobile X-ray devices) and the practical upside of TL in constrained environments [7].

A ResNet50+SVM hybrid reached $\approx 93\%$ on 672 public CXRs, suggesting that pairing deep features with margin-based classifiers can be a solid alternative to end-to-end softmax heads [8].

Lightweight hybrid designs also performed competitively: MobileNet+SVM and DenseNet201+MLP variants reported up to $\approx 98.46\%$ accuracy on 774 images, which is attractive for low-resource deployments [10].

At the same time, some headline numbers warrant caution. A custom CNN trained on the large NIH CXR set reported 100% accuracy; while impressive, such results risk overfitting without rigorous external validation and consistent protocols [9].

More broadly, many high-performing studies still lack out-of-distribution testing and

clear explainability. Reviews of XAI in this domain stress that tools like Grad-CAM help clinicians see what the model focuses on; papers integrating Grad-CAM show better clinical trust than equally accurate “black-box” models [11], [12].

In fact, several works note limited external proof and hardware constraints as practical barriers—accuracy alone is not enough; dataset diversity, interpretability, and real-world feasibility are equally important [13].

Finally, data strategy matters as much as architecture. Studies using augmentation (including GAN-based schemes) reported consistent gains by addressing class imbalance and feature scarcity, which aligns with the trend toward more robust pipelines rather than model-only tweaks [14].

Overall, the literature supports three takeaways for our project: (i) transfer learning with VGG/ResNet/DenseNet is a strong baseline above 90% accuracy [1], [2], [7], [8]; (ii) ensembles/hybrids can push the frontier on difficult cases [3], [4], [10]; and (iii) clinical adoption benefits from explainability and external validation as much as from raw metrics [11]–[13].

Here (fig 2.1) We see best uses method across reviewed studies (ten paper show here) and (fig 2.2) we see which model give how many accuracy achieve .

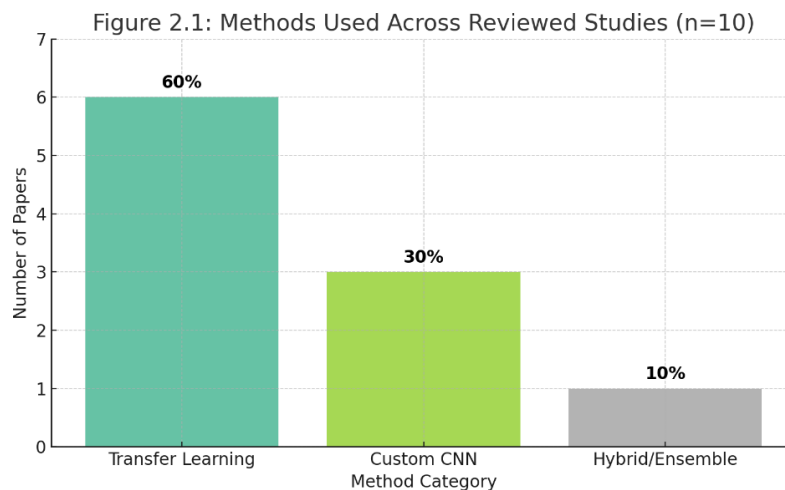


Figure 2.1: Methods Used Across Reviewed Studies (n=10)

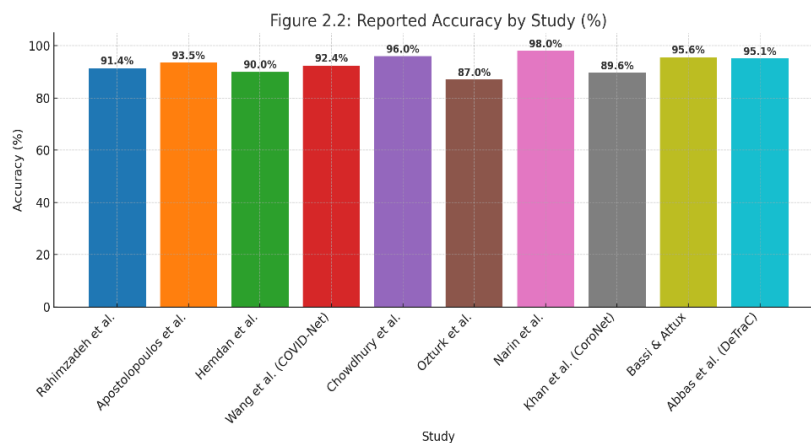


Figure 2.2: Reported Accuracy by Study (%)

We studied many papers but here show just ten papers. Here we see (table 2.1) paper author, year, title, methodology and key findings that help to know a paper how well and get best output for us.

Table 2.1: Summary of Literature Reviewed.

Author (s)	Year	Title	Methodology	Key Findings
Arias-Londoño et al.	2020	COVID-Net for CXR Analysis	CNN (COVID-Net)	Achieved ~91% accuracy on HM, BIMCV, MIMIC-CXR datasets [1].
Sen et al.	2021	Transfer Learning with CNNs	VGG16, Xception, InceptionV3	Reported 91–93.5% accuracy using transfer learning [2].
Shah et al.	2022	CNN-GRU On choen CXR	Hybrid (CNN + GRU)	Precision/Recall = 0.96; F1 = 0.95 [3].
Calderon-Ramírez & Yang	2021	WideResNet study	WideResNet	Accuracy 81.6%–97.6% (small dataset) [5].
Panetta et al.	2021	Classical ML vs DL	SVM/KNN/RF/GB on Kaggle+COVID GR	Best 98.44% (GB) [6].
De Moura & García	2020	Portable-device CXR	DenseNet-161 (ImageNet TL)	90.7% accuracy [7].
Zhou et al.	2021	ResNet50 + SVM	Hybrid DL + SVM	93% accuracy (672 images) [8].
Oyelade et al.	2021	CovFrameNet on NIH CXR	Custom CNN (large NIH set)	Reported-100% (possible-overfit concern) [9].
Ohata et al.	2021	Lightweight hybrids	CNN, MobileNet+SVM, DenseNet201+MLP	Upto-98.46%; efficient models competitive [10].

2.2.1 Similar Applications

Before developing our own system, it is useful to look at how other groups and companies have already tried to apply deep learning to chest X-ray analysis. These examples are not exactly the same as our work, but they give a clear idea of what has been done, where the difficulties were, and what features proved to be the most practical.

1. Web-based Demonstration Systems

At the start of the COVID-19 crisis, many researchers quickly built web tools so that people could test AI models online. A well-known one is COVID-Net, which allowed users to upload chest X-rays through a simple webpage. The system gave predictions and also showed heatmaps pointing to the abnormal parts of the lung. These platforms were mainly for research, not direct hospital use, but they showed how easy it can be to share AI models openly and let people see what they can do. The biggest strength here was accessibility, because anyone with a browser could use it.

2. PACS-Integrated Tools

Another important example is CAD4COVID-Xray, which worked directly with the hospital's PACS system. PACS is already used daily by radiologists to manage medical images. This tool analyzed X-rays automatically as soon as they were uploaded, and the results appeared inside the same reporting system doctors were familiar with. As a result, physicians didn't need to harmonize their practices or learn anything new. This contributes to seamless workflow integration, which greatly increases the viability of AI adoption in busy hospitals.

3. Commercial Platforms with Flexible Deployment

Some businesses, such as Qure.ai, made solutions that are currently widely utilized. COVID-19 is one of the lung conditions that their product, qXR, can detect. A big advantage of qXR is that it can run in two ways: on the cloud or locally on hospital servers. This is important because not every hospital has fast internet. By giving those two options, the system can be used in large, well-resourced hospitals as well as in little rustic clinics. This shows that AI can be adapted to very different healthcare settings without losing usefulness.

4. Explainability-Focused Applications

Trust is always a challenge in medical AI. Lunit INSIGHT CXR is a system that tried to solve this by focusing on explainability. Instead of giving just a yes/no prediction, it shows the exact areas on the chest X-ray that seem abnormal. This makes it easier for doctors to compare the AI's reasoning with their own observations. Many radiologists are more willing to use such a system because it doesn't feel like a black box. The contribution here is very clear: accuracy alone is not enough—explainability is essential.

5. Mobile and Lightweight Solutions

Lastly, some initiatives have focused on developing smartphone apps that can operate AI models. Condensed neural network versions that don't require internet access are used in these applications. The idea is simple: doctors or technicians can use a portable machine to take an X-ray in a remote area or during outreach activities, run it through a mobile app, and get the results right away. This approach is particularly helpful in places with limited resources, where advanced equipment or dependable internet connections might not be available.

6. Beyond COVID-19

It is also worth pointing out that the same type of approach has been used for other diseases like tuberculosis and pneumonia. In fact, in several countries, mobile X-ray vans with built-in AI screening have been used to check thousands of people. This proves that once an AI pipeline is created, it can be applied to many lung diseases with only small adjustments.

Summary of Lessons

Looking at these applications, the lessons are quite clear. A good system should be easy to access, whether through the web, directly in hospital systems, or even on mobile phones. It ought to be sufficiently light to operate at low resource levels. It must provide doctors with an understandable and reliable explanation of its predictions. Additionally, it should be adaptable enough to treat a variety of lung conditions rather than being restricted to just one.

2.3 Gap Analysis

From the reviewed works, it is clear that while deep learning has shown great promise in detecting COVID-19 from chest X-rays, some important gaps remain. Most of the existing studies used only open-access datasets like COVIDx or CheXpert [12], [13]. Although these datasets are useful, they do not always reflect the diversity of real hospital cases, which may reduce the reliability of such models in actual practice. Another common issue was class imbalance—in many datasets, the number of COVID-19 positive cases was much smaller than normal cases [9], [16]. This often makes the models biased toward the majority class.

Many high-performing models such as DenseNet and Xception [10], [19] also come with a drawback—they demand powerful GPUs and large memory, which makes them hard to deploy in hospitals with limited computing resources. On the other hand, the problem of explainability is still not fully solved. Except for a few works that used Grad-CAM [14], [25], most models remain “black-boxes,” which makes doctors less confident about using them. Moreover, many studies tested their models only on one dataset [12], [20], raising questions about whether the results would hold up in new hospital environments. Lastly, some works focused only on accuracy [16], [19] as the performance metric, without looking at precision, recall, or ROC-AUC, which are very important in medical diagnosis where false negatives can be dangerous.

Our project directly addresses these gaps. We prepared a balanced dataset of 3,100 images that combines both real hospital data and open-access sources, making it more reliable. We compared both lightweight and large pre-trained models under the same conditions to see which one works best in different scenarios. We also evaluated models using multiple metrics such as accuracy, precision, recall, F1-score, and ROC-AUC for a fairer analysis. Importantly, our custom CNN achieved strong results (92%) with a simple architecture, proving that lightweight models can be used in real hospitals without advanced computers. Finally, while explainability and cross-hospital validation are still future tasks, our work sets the foundation for building more trustworthy and practical AI-based diagnostic systems. Here show (table-2.2) gap analysis of studies paper .

Table 2.2 Gap Analysis

Gap Area	Observation from Related Works	Limitation	Our Contribution
Dataset Diversity	Most studies used only open datasets (e.g., COVIDx, CheXpert) [12], [13]	Limited real-world clinical relevance; possible dataset bias	Collected a balanced dataset (3,100 images) including real hospital data from Shahjalal Medical Hospital
Class Imbalance	Several datasets were highly imbalanced (COVID cases << Normal cases) [9], [16]	Models biased toward majority class; poor generalization	Ensured balanced dataset (1,575 COVID, 1,525 Normal) for fair training
Computational Demand	Deep models like DenseNet/Xception [10], [19] achieved high accuracy but required powerful GPUs	Hard to deploy in low-resource hospitals	Designed a lightweight custom CNN (92% accuracy) suitable for limited hardware
Explainability	Only few studies applied Grad-CAM [14], [25]	Most models remain “black boxes,” limiting clinical trust	Proposed integration of explainability methods in future scope
Generalization	Many models tested only on one dataset [12], [20]	Risk of poor performance on unseen hospital data	Highlighted need for cross-hospital testing; proposed in future extension

2.2 Summary

In this chapter, we considered how researchers have used deep learning for COVID-19 detection from chest X-ray images and what lessons we can take from their work. The review showed that powerful pre-trained models like VGG16, ResNet50, and DenseNet have achieved very high accuracy. At the same time, some studies also tried simpler custom CNNs and proved that lightweight models can still give reliable results when designed and trained carefully. This tells us that both large and compact models have value depending on the situation—big models may push accuracy higher, while small ones are easier to use in hospitals with limited resources.

But the literature also made it clear that there are still challenges. It is unclear how well those models would perform in practice because many studies only used open-access datasets, which are not always representative of actual hospital data. In some cases, highly imbalanced datasets caused bias in the models, resulting in a significantly lower number of COVID-19 positive samples than normal samples. Another issue is that many high-performing models require powerful GPUs and a lot of processing power, which are not available in all hospitals. Additionally, models continue to operate primarily as “black boxes,” offering little insight into the decision-making process, which makes it

hard for doctors to trust them. Finally, several works reported only accuracy as the main metric, ignoring other important measures like precision, recall, F1-score, and ROC-AUC, which are critical in medical diagnosis.

By looking at these studies, we now have a clearer picture of both the progress and the gaps in this field. Building on these results, our project tests both pre-trained and lightweight models in the same setup, uses a balanced dataset that includes both hospital and open-access images, and evaluates performance using multiple metrics rather than just accuracy. In this sense, our work helps to make AI-based diagnosis reliable, useful, and accurate.

Chapter 3

Research Methodology

This chapter describes the overall methodology followed to complete the research. The purpose here is not only to list the steps but also to explain how each step was planned and why it was necessary. By presenting the methodology in this way, we aim to show that the project was conducted in a systematic, transparent, and reliable manner, from data collection to final evaluation.

3.1 Methodology

The work on this project was done in steps to keep the process clear and easy to follow. Preparing the dataset was the first and most important step. Since a deep learning model learns directly from the data we provide, we focused on creating a dataset that was both balanced and reliable. For this reason, we collected 3,100 chest X-ray images, including almost the same number of COVID-19 positive and normal cases. The data came from two sources: Shahjalal Medical Hospital and open-access online repositories. This combination allowed us to keep the dataset both realistic and diverse, reducing the chance of bias.

Once the dataset was ready, we moved on to preprocessing. The images we collected were not all the same in size or quality, which could create problems during training. To solve this, we resized every image to a fixed size of 224×224 pixels and normalized the pixel values so that brightness and contrast did not mislead the model. We also used data augmentation methods like flipping and rotating. The model learned better and did well on new cases it had never seen before because it got slightly different versions of the same images.

We worked on the models after we got the data ready. Here we took two different approaches. We created a lightweight CNN of our own for one side. The goal was to describe whether a more basic model could still produce useful results, particularly in medical facilities where expensive computers or GPUs might not be available. Conversely, we made use of some of the most well-known pre-trained models, including Xception, InceptionV3, DenseNet121, ResNet50, and VGG16. These models were selected because they have a solid track record in image classification tasks and may provide a reliable standard against which our custom CNN can be evaluated.

We used Google Colab with GPU support for training. This made it possible to handle deep learning experiments efficiently without requiring expensive hardware. To make the comparison fair, every model was trained under the same conditions. We

used the Adam optimizer and categorical cross-entropy as the loss function for all experiments so that the differences in performance came only from the models themselves and not from the training setup.

Finally, we evaluated the models. Instead of looking only at accuracy, we considered a full set of metrics, including precision, recall, F1-score, and ROC-AUC. These were important because in healthcare, especially for COVID-19 detection, missing a positive case can be very dangerous. For a better understanding of the advantages and disadvantages of each model, we also looked at confusion matrices and ROC curves.

We compared each model's output at the conclusion of the process. Even though pre-trained models performed exceptionally well, our lightweight CNN also attained competitive accuracy, making it a sensible option for hospitals with constrained funding. By taking a methodical approach, we were able to test the models in a fair manner and gain insight into how these systems can be used in healthcare settings to facilitate prompt and accurate COVID-19 diagnosis.

3.1.1 Overview

The methodology followed in this project was planned as a simple but logical sequence of steps. The starting point was the collection of chest X-ray images, since without a reliable dataset no deep learning model can perform well. To make the dataset balanced, images were gathered from Shahjalal Medical Hospital as well as open-access archives. In total, 3,100 images were used, including almost equal numbers of COVID-19 positive and normal cases. This balance was important to reduce bias and to make the system more reliable in practice.

The pictures were carefully prepared for training after they were collected. Each pixel value was resized to the same size, normalized, and subjected to a variety of augmentation techniques, including rotation and flipping. By removing unnecessary variation from the dataset, these preprocessing methods improved the model's ability to handle new, untested images.

Model development was the main focus of the following section of the work. Two paths were followed side by side. The first was to build a custom CNN that was lightweight and easy to run even on limited hardware, such as the computers available in small hospitals. The second was to fine-tune several well-known pre-trained models, including VGG16, ResNet50, DenseNet121, InceptionV3, and Xception. By taking this two-way approach, it became possible to compare a simple design with more advanced and complex architectures.

Training and testing of the models were carried out on Google Colab with GPU support. To make the evaluation fair, the same dataset and conditions were used for all experiments. Standard performance measures such as accuracy, precision, recall, F1-score, and ROC-AUC were applied. In addition, confusion matrices and ROC curves were studied to better understand the strengths and weaknesses of each model, rather than relying only on a single metric.

The final step was to compare the models and interpret the outcomes. Although

identifying COVID-19 was the primary goal, the methodology's design was kept adaptable to allow for the detection of other respiratory diseases like pneumonia and tuberculosis. In this way, the project helped create a framework that can be expanded to future medical applications in addition to addressing a pressing global issue.

3.1.2 Proposed Methodology

The methodology developed in this work was designed to build a system that is accurate, efficient, and easy to apply in real healthcare settings. While earlier studies have mostly relied on deep and complex models, our approach combines two perspectives: a simple, lightweight custom CNN that can run in low-resource environments, and several powerful pre-trained networks that represent the current state of deep learning. This combination ensures that both practical usability and high accuracy are considered. The process can be described in seven main stages.

Figure 3.1: Proposed Methodology

1. Dataset Collection and Preparation

The dataset is the most important part of this project. This kind of learning works best with data that you give it. If the dataset is biased or wasn't put together carefully, then not even the most advanced model will be able to make accurate predictions. For that reason, we spent a lot of time making sure our dataset was properly designed and checked.

We collected 3,100 chest X-ray images in total. These came from two different sources. The first source was Shahjalal Medical Hospital, which provided real patient data. This gave the dataset a strong link to real clinical cases. The second source was a set of open-access repositories. Adding them helped us include more variety, with scans from different patients and different imaging conditions. By joining the two sources, we ended up with a dataset that was more diverse and trustworthy.

One issue we paid close attention to was class balance. Many medical datasets have more normal scans than COVID-19 positive ones. This kind of imbalance makes a model favor the majority group, which is a problem for diagnosis. To avoid this, we built a nearly balanced set: 1,575 COVID-19 positive and 1,525 normal scans. The model had an equal chance to learn from both sides in this case.

Another important step was to clean up the dataset. We got rid of scans that were duplicates or not very good, and we checked the labels twice. This was important because mistakes in the data can easily reduce the accuracy of the system.

Finally, to show a clear picture of the dataset, Figure 3.2 includes two sample chest X-rays. One is from a COVID-19 positive patient and the other is from a normal case. The differences between them are not obvious to the human eye, which explains why deep learning can be so useful in detecting such cases.

Covid case X-ray



Normal case

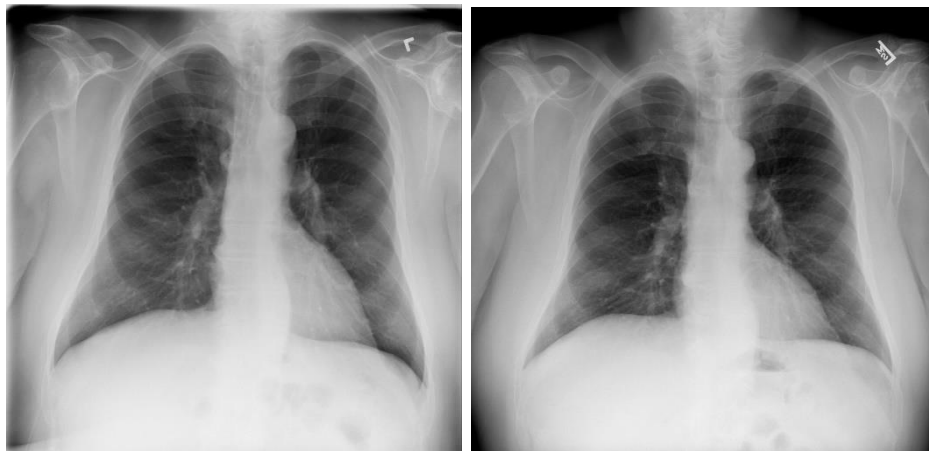


Figure 3.2: Here two cases shown covid Infected and normal case study

2. Data Preprocessing

After collecting the dataset, the next essential stage was preprocessing. This stage is often underestimated, yet in practice, it determines how well a deep learning model can learn from medical images. Raw chest X-rays rarely come in the same shape or resolution. Some are clear, others slightly blurred, and the file formats themselves often differ. If such images were passed directly into the model, it would struggle to find consistent patterns. For that reason, a careful preprocessing pipeline was applied so that every image entered training in a uniform and meaningful way.

The process began with grayscale conversion. Although many chest X-rays are saved in RGB format, they contain no real color information. Retaining the three color channels would only inflate the data size without adding any diagnostic value. By converting the scans to grayscale, the dataset became lighter, and more importantly, the model's focus was directed toward the lung textures and structures that matter clinically.

Once that was complete, all images were resized to 224×224 pixels. The dataset originally contained scans in a variety of resolutions, which would have made training unstable. Resizing gave each input the similar shape, ensuring balance with the model architecture and softly training.

The third step we used was normalization. Since the chest X-rays were taken in different hospitals and by different machines, the pictures did not always look the same. Some appeared darker, while others were brighter. The model could easily interpret the brightness as a feature if we left them that way, which is not what we want. We improved the images' balance by normalising the pixel values. This prevented light or contrast

problems from diverting the model's attention from the actual lung details.

Data augmentation was the final step. Here, we created additional copies of the pictures with minor adjustments, such as slight resizing, left-to-right flipping, or rotation. The X-ray's meaning is unaffected by these modifications. The lungs are the same, only the view is a bit different. This simple trick made our dataset larger and more varied. As a result, the model became stronger and less likely to overfit.

To illustrate, Figure 3.2 shows three stages: the raw chest X-ray, the same image after grayscale conversion and normalization, and a series of augmented versions. Looking at them side by side makes it clear how preprocessing both standardizes and strengthens the data, setting a stronger foundation for model training.

Showing samples from: COVID

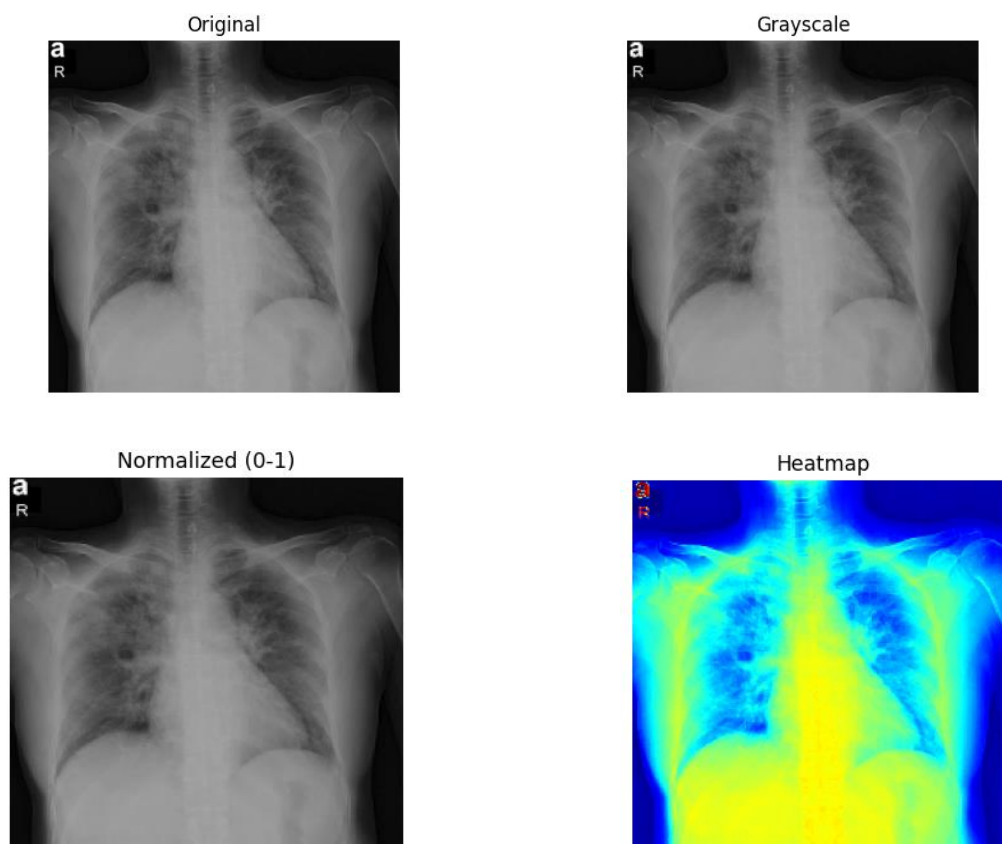


Figure 3.3: Original to Grayscale then it convert Normalized also heatmap

Showing samples from: Normal

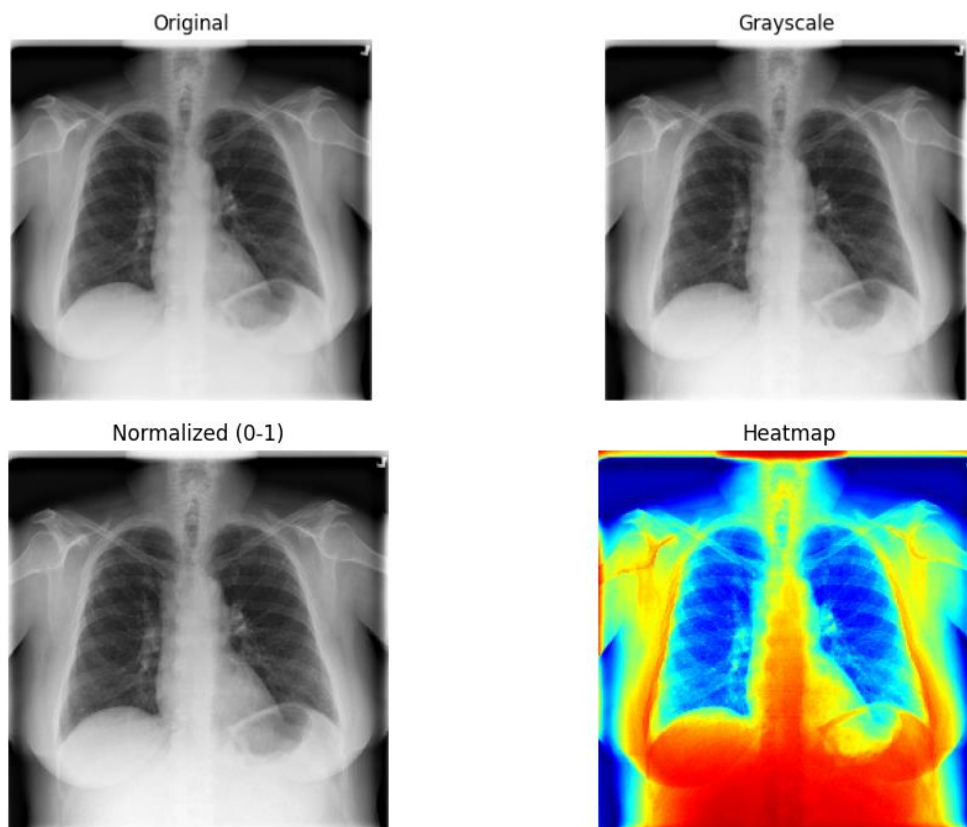


Figure 3.4: Original to Grayscale then it convert Normalized also heatmap

3. Custom CNN Design

Besides using pre-trained models, we also built a lightweight Convolutional Neural Network (CNN) from scratch. The main cause after this was very practical: not every hospital has high-end GPUs or advanced computing facilities. During the COVID-19 pandemic, this problem became clear. So, our goal was to design a model that could still work in such settings without losing reliability.

The custom CNN was kept simple and light. We did not add too many deep layers because the focus was on efficiency rather than size. Even though it is less complex compared to heavy pre-trained networks, the model can still learn key features from chest X-ray images. The convolutional layers extract patterns like edges, textures, and lung structures, while pooling layers cut down unnecessary complexity and make the training faster. At the end, fully connected layers handle the classification, separating the scans into either COVID-19 positive or normal.

What makes this model useful is not only the results it produces but also its accessibility. Since it needs fewer computing resources, it can be deployed in small hospitals, clinics, and rural health centers where costly hardware is not available. In other words, even if it is not as deep as models like VGG16 or ResNet50, it offers a realistic option for diagnosis in low-resource environments.

Figure 3.5: A sample diagram of Custom CNN

For better understanding, Figure 3.3 presents the structure of our custom CNN. It demonstrates the sequential progression, beginning with convolutional layers and progressing through pooling and classification.

4. Transfer Learning with Pre-trained Models

To strengthen our study beyond the custom CNN, we also relied on five popular deep learning architectures: VGG16, ResNet50, DenseNet121, InceptionV3, and Xception. These models were originally trained on the massive ImageNet dataset, which contains millions of images from a wide range of categories. Because of this, they already have the ability to recognize many types of visual features such as edges, textures, and shapes. Instead of starting from scratch, we adapted this prior knowledge to chest X-rays using the approach of transfer learning.

The idea behind transfer learning is quite practical. Early layers of these pre-trained models act like common feature extractors, while the later layers can be fine-tuned for a new task. In our case, we froze highest of the early layers and only retrained the final ones so that the models could specialize in identifying patterns related to COVID-19 infections. This method not only reduced training time but also improved reliability, since the models had already been exposed to millions of images before.

Each chosen model had distinct advantages. InceptionV3 and Xception are more advanced and can capture fine details at different sizes. VGG16 is basic but very dependable. ResNet50 employs shortcut connections to fix the vanishing gradient problem, and DenseNet121 improves information flow by reusing features. These models gave us a lot of ways to test our lightweight CNN when they were put together.

The real motivation here was to see the balance between accuracy and practicality. Pre-trained models are usually very accurate but they need more memory and powerful GPUs. On the other hand, our custom CNN was much lighter and could run on modest hardware. By evaluating both approaches on the same dataset under the same training conditions, we were able to clearly study the trade-off between peak performance and ease of deployment in real hospital settings.

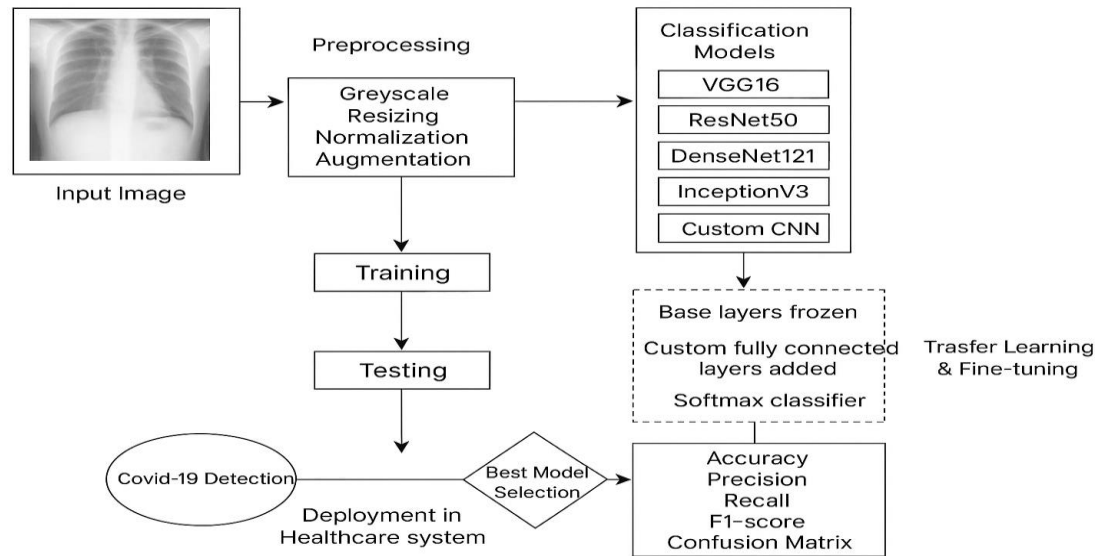


Figure 3.6 : Transfer learning model work diagram

For better understanding, Figure 3.4 illustrates the transfer learning pipeline. It shows how a pre-trained model, originally trained on ImageNet, is fine-tuned and connected to new fully connected layers to classify chest X-rays into COVID-positive or normal cases.

5. Model Training and Testing

Once the models were schematic, the next big step was to actually train and test them. To make sure the comparison was fair, we kept the setup the same for all models. Everything was done on Google Colab with GPU support, which gave us enough computing power without needing expensive hardware.

The dataset was carefully split into three parts: training, validation, and testing. The training set helped the models learn from the chest X-rays, while the validation set was used to check how well they were learning during the process and to fine-tune parameters. Finally, the testing set—kept completely separate—was used at the end to see how well the models could handle new images they had never seen before.

For training, we used the Adam optimizer, which is widely trusted because it adapts the learning rate automatically and makes the learning process smoother. Since the task was to classify images into two categories (COVID-positive or normal), we applied categorical cross-entropy as the loss function. This combination gave a stable and efficient training process.

To prevent the models from simply memorizing the training data, we included early stopping. This means that the training would stop as soon as the model stopped improving on the validation set. It reduced the chance of overfitting in addition to saving time. On top of that, we used data augmentation—small changes like rotations, flips, and scaling—to give the models a wider variety of images. This step was crucial because it helped the models learn to handle real-life variations in X-rays rather than just memorizing fixed examples.

By keeping everything—the dataset split, the optimizer, the loss function, and the training conditions—the same across all models, we made sure the comparison between our lightweight CNN and the pre-trained models was completely fair and unbiased. In this sense, the variations in performance were a reflection of the models' inherent strengths

rather than the training procedure.

6. Evaluation Metrics and Analysis

Model performance was assessed through multiple indicators. Accuracy gave an overall picture, while precision and recall focused on how well COVID-19 cases were detected and how often errors occurred. The F1-score balanced these two measures. ROC curves and AUC values were also calculated to examine how effectively the models distinguished between classes. Confusion matrices offered a clear overview of performance by class.

We used standard classification metrics to evaluate model performance on the test set:

- Accuracy
- Precision
- Recall
- F1-Score

Let TP, TN, FP, and FN denote True Positives, True Negatives, False Positives, and False Negatives, respectively. The metrics are computed as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

7. Application and Outcome

The last part of our work was not only about checking scores but also about understanding how the model could actually be used in real life. Among all the models we tested, the one with the best results stood out clearly. But instead of stopping at accuracy and F1-scores, we asked a more important question: is this model practical enough for hospitals and clinics?

When we looked closely, the top model showed promise both in performance and usability. It was fast in producing results and did not demand heavy computing power, which makes it suitable even for healthcare centers that have only limited resources. This is important because many hospitals, particularly smaller ones, cannot always depend on high-end machines. A model that can still work well under such conditions becomes far more useful than one that is accurate but resource-hungry.

Another key point is the facility of the system. Although it was developed for COVID-19 detection, the pipeline is not locked to a single disease. With further training, it can be adapted to detect other lung problems like tuberculosis or pneumonia. This gives the access a wider scope, making it a tool that can continue to evolve rather than something that becomes outdated once COVID-19 cases decline.

3.1.3 Functional and Nonfunctional Requirements

When designing this system, it made consciousness to Isolate the requirements into two types. The functional requirements, which are basically the capabilities that the system must possess, are on the one hand. On the other hand, there are the non-functional ones, which focus on the general behaviour of the system rather than specific features. The first team ensures that the system works, and the second team ensures that it does so in a way that is reliable and useful in real hospitals. Both are equally significant.

Functional Requirements

Image Input

The system should let users upload chest X-ray scans. Common formats like JPEG and PNG must be supported, as well as ideally medical formats like DICOM. A batch option would be handy too, since doctors sometimes check multiple scans together.

Preprocessing

Once uploaded, the system should handle the boring but necessary stuff: resizing images, normalizing them, maybe even segmenting the lung area. Without this step, the model won't be consistent.

Classification

The main job here is straightforward: say whether the X-ray shows COVID-19, pneumonia, or looks normal. Along with the label, the system should show how confident it is in its guess.

Choice of Models

Since both a custom CNN and pre-trained models are being tested, the system should allow switching between them. Doctors or researchers might want to see how the results compare.

Visualization

A result is more useful if it shows why. That's why the system should give visual feedback — like highlighting the parts of the lung that influenced the decision.

Results & Storage

Predictions should be clear on screen, but there should also be a way to save them, maybe in a report format for hospital records.

Testing Metrics

During development, the system should calculate accuracy, precision, recall, F1, and ROC-AUC. These numbers give a better sense of how the models really perform.

Privacy

Because these are medical images, access must be limited to authorized users only. Security is not optional here.

Non-Functional Requirements

Accuracy

The model should aim for at least 90% accuracy. More importantly, it needs to stay consistent across different sets of images.

Speed

Doctors don't have time to wait. The system should process each scan in just a few seconds.

Scalability

If hundreds of scans need to be tested in a hospital, or several users log in at once, the system shouldn't slow down.

Usability

The interface must be simple enough for non-technical staff. Ideally, it should be as easy as “upload → click → see result.”

Deployment Flexibility

It should run both locally in a hospital and, if needed, on the cloud. That way even low-resource clinics can use it.

Maintainability

Future updates, such as the addition of TB or other lung disease detection, should be possible with this design without requiring a complete redesign.

Data Security

Patient scans must be encrypted and handled according to privacy rules. Trust is crucial here.

Robustness

Not all images will be perfect. The system should still work reasonably well even if the X-ray is noisy or slightly blurred.

Interpretability

Doctors should be able to see why the model gave a result. Black-box answers are not enough in healthcare.

Standards Compliance

Finally, the system should follow accepted healthcare standards and guidelines, so it can actually be used in hospitals without raising doubts.

3.1.4 Data Flow Diagram

The system is designed in a way that makes it simple for users to interact with. When a user enters their login credentials into the web application, the process begins. After logging in, the user has the option to upload a chest X-ray image through the interface.

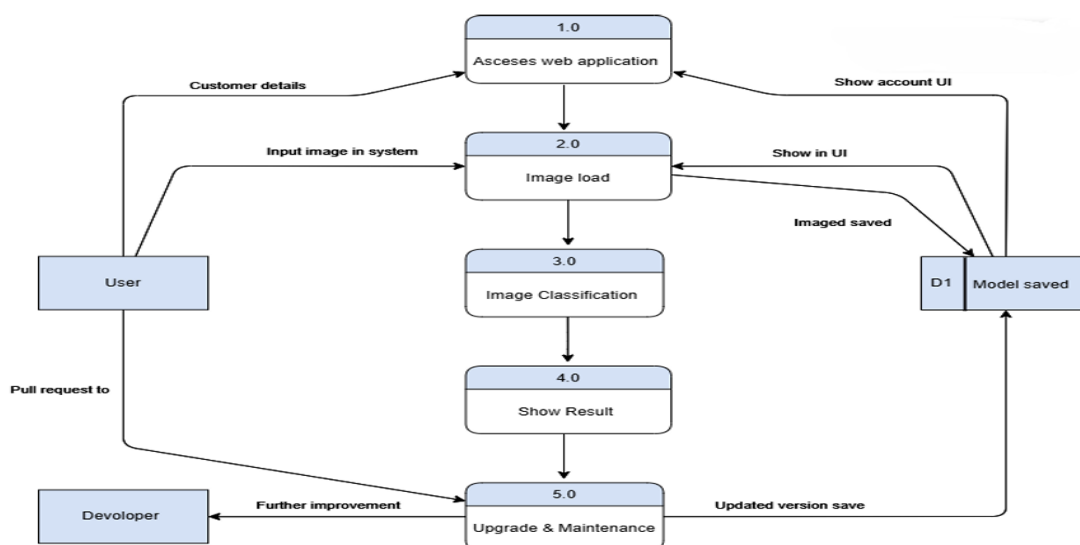


Figure 3.7: Data Flow Diagram

After uploading, the picture is routed to the backend, where the trained deep learning model processes it and stores it safely in the database. After analysing the image, the model makes a prediction about whether the case falls into the Normal or COVID-positive class.

Finally, the result is sent back to the web application, and the user can view the outcome directly on their dashboard. The entire process is made simple enough that even non-technical users can log in, upload an X-ray, and get a clear diagnostic result right away.

3.1.3 UI Design

The system is designed in a way that makes it simple for users to interact with. Once a user enters their login credentials into the web application, the process begins. After logging in, the user has the option to upload a chest X-ray image through the interface.

After uploading, the picture is routed to the backend, where the trained deep learning model processes it and stores it safely in the database. The model analyzes the image and predicts whether the case belongs to the COVID-positive class or the Normal class.

The user can view the result directly on their dashboard after it has been returned to the web application. The entire flow is designed to be straightforward so that even users without a technical background can easily log in, upload an X-ray, and instantly receive a clear diagnostic result. All this things show Figure: below

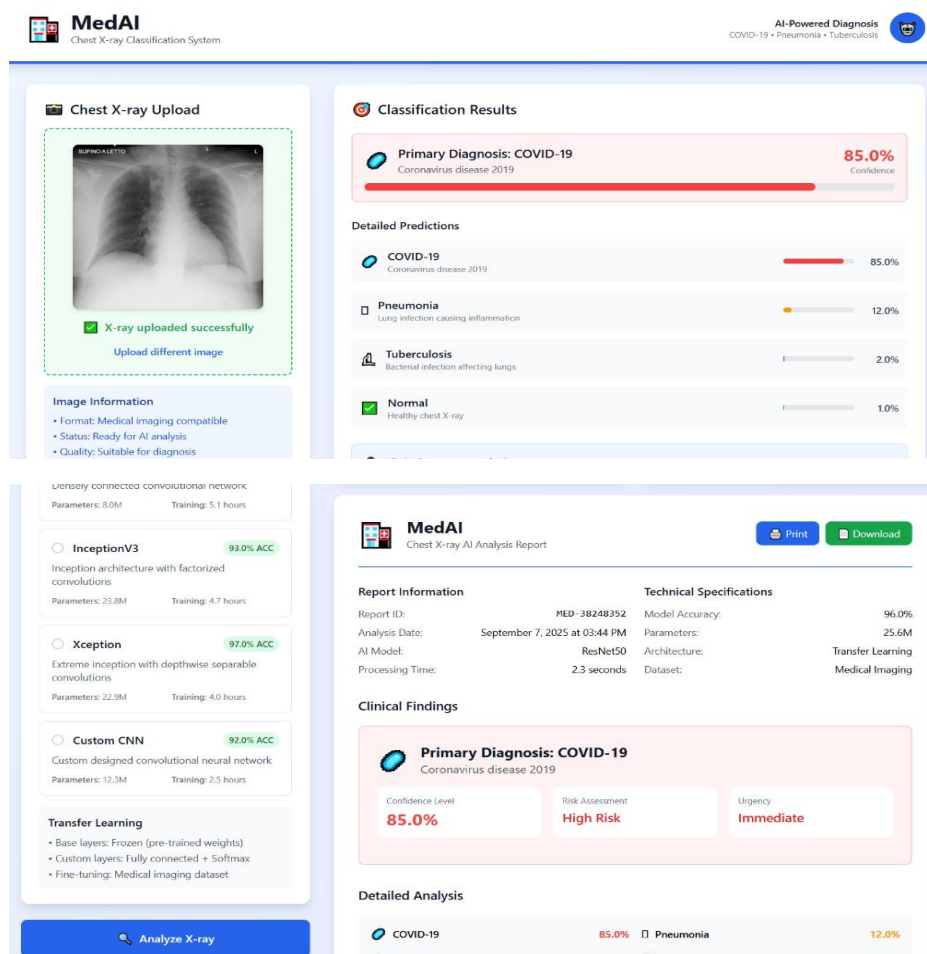


Figure 3.8: UI Design of web apps

3.2 Detailed Methodology and Design

When we first started planning the project, we explored a number of possible approaches before deciding on the final solution. Since the aim was to build a system that could reliably detect COVID-19 from chest X-ray images, the choice of method had to balance two things: accuracy and practical usability in hospitals.

One of the early alternatives we thought about was using traditional machine learning methods. This method would involve manually extracting features from the images, like textures, shapes, or patterns of intensity, and then using classifiers, such as Random Forest or SVM, to generate predictions. Despite being easier, it soon became apparent that this approach would not be sufficient for our task. Chest X-rays are complex, and handcrafted features often fail to capture the subtle differences between a COVID-19 infected lung and a normal one. This meant the model's accuracy would likely be too low for real medical use.

Another possible path was building a very deep CNN completely from scratch. This kind of model can work extremely well in theory, but it has two big drawbacks. Firstly, in order to train correctly, a very large dataset is needed, which is frequently challenging to gather in the medical field. Second, many hospitals and small clinics lack the powerful hardware and lengthy training periods required for such models. Therefore, this option would not be feasible in the real world, even though it might produce positive results in a research lab.

The solution we chose was a hybrid approach. On the one hand, we developed a lightweight CNN ourselves that can function flawlessly even on low-end hardware. This made it suitable for low-resource environments. On the other side, we used transfer learning with pre-trained models like VGG16, ResNet50, DenseNet121, InceptionV3, and Xception. We could achieve very high accuracy by fine-tuning these models for our dataset because they are already highly skilled at extracting features and have been trained on millions of images.

This combined strategy gave us the best balance. The custom CNN provided a simple, fast, and practical option for real-world deployment, while the pre-trained models gave us state-of-the-art performance when resources were available. By comparing both under the same setup, we were able to see not just which model was most accurate, but also which one would be more realistic to use in different hospital settings.

The final design of our system followed a clear sequence:

1. User login and image upload through the web application.
2. X-ray image preprocessing (greyscale conversion, resizing, normalisation, and augmentation).
3. Model selection (custom CNN or pre-trained models with fine-tuning).
4. Training, validation, and testing using standardized evaluation metrics.
5. Result generation, indicating if the X-ray is from a normal case or one with COVID.

This design was chosen not just because it works for COVID-19 detection, but also because it is adaptable. A versatile and long-term solution, the same pipeline can be expanded to

treat other lung conditions like pneumonia or tuberculosis.

3.3 Project Plan

To keep the project well organized, we broke it down into different phases. Each phase had a specific purpose and a clear outcome. The plan is shown below in a simple table:

Phase	Activities	Expected Outcome
Phase 1: Background Study	Collected and reviewed research papers, studied related work, and tried to identify what gaps still exist.	A clear understanding of the problem and where our work can add value.
Phase 2: Dataset Preparation	Gathered chest X-ray images from hospital records and open datasets. Applied preprocessing steps such as grayscale conversion, resizing, normalization, and data augmentation.	A clean, balanced, and ready-to-use dataset.
Phase 3: Model Building	Designed a simple CNN from scratch and also selected well-known pre-trained models (VGG16, ResNet50, DenseNet121, InceptionV3, Xception) for transfer learning.	Different models ready to be trained and compared.
Phase 4: Training and Testing	Trained all models using the same setup in Google Colab with GPU. Used metrics like accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices to measure performance.	Identified which model performed the best and where improvements were needed.
Phase 5: System Development	Built a web-based app where users can log in, upload chest X-rays, and instantly see predictions.	A working prototype that connects the trained model with a usable interface.
Phase 6: Final Analysis and Report	Tested the whole system, summarized the findings, and discussed how the same pipeline could later be used for diseases like TB or pneumonia.	A complete report and a practical diagnostic system ready for future expansion.

3.4 Task Allocation

The entire project was broken up into four distinct phase over the course of the semester in order to maintain institution . Each phase had its own timeline so that the tasks could be completed step by step without creating unnecessary pressure at the end.

The first phase ran from week 12 to week 20, where the main goal was to collect the dataset. This included gathering chest X-ray images from Shahjalal Medical Hospital as well as using some open-access resources. Building this dataset was an essential starting

point, since every other part of the work depended on having enough quality data.

The second phase was carried out between week 22 and week 28, and it focused on preprocessing. The photos underwent resizing, normalisation, and augmentation during this period. These steps helped in preparing the dataset so that the deep learning models could learn patterns in a more consistent way.

The third and longer phase took place in weeks 30 to 40, which was dedicated to training and evaluating models. In this stage, both the custom CNN and pre-trained models such as VGG16 and ResNet50 also many were tested. Performance was checked with different metrics like accuracy, precision, and recall to make sure the models were reliable.

Finally, in weeks 42 to 48, the last phase of the project was completed. We make a demo web application that detect any case of chest x-ray image . Now at present time we can't join with database but our demo web application ui complete.

This table depicts the timeline of the principal activities in each period of the project, from week 12 to week 48.

Tasks	Weeks																		
	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
Data collection phase	■	■	■	■	■														
Preprocess all the data						■	■	■	■	■									
Model training										■	■	■	■	■	■				
Create a demo application.																■	■	■	■

3.5 Summary

This chapter explained how the project was carried out step by step, starting from the early planning stage and moving toward the final system design. At first, we looked at different possible solutions, including traditional machine learning and very deep CNN models. These options were considered carefully, but in the end they were not chosen because they were either too limited in accuracy or too demanding in terms of resources. Instead, we decided on a combined strategy: building a lightweight CNN that can run on modest hardware, and using transfer learning with powerful pre-trained models to achieve state-of-the-art results.

We also described how the dataset was collected and carefully prepared before training. The preprocessing pipeline comprised data augmentation, normalisation, resizing, and greyscale conversion. These steps convincing that the images were clean, balanced, and consistent, making it easier for the models to learn the real patterns in the data.

The models' training and assessment processes were then described in the methodology section. We relied on standard metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices. This allowed us to measure not only overall correctness but

also how well the models could identify COVID-positive cases without missing or misclassifying them.

The work was then broken down into phases by our well-defined project plan, which included background research, dataset preparation, model design, training and evaluation, system integration, and reporting. This well-organised plan made sure that every step contributed to the final result and kept the work on schedule. Additionally, the chapter stressed that the system was made to be flexible, so the same pipeline can be used in the future to identify other lung conditions like pneumonia and tuberculosis.

In short, this chapter provided a complete picture of how the system was designed, why certain choices were made, and how the work moved from raw data to a functional diagnostic tool. It shows that the solution is not only accurate but also realistic and adaptable for use in real healthcare environments.

Chapter 4

Implementation and Results

This chapter details the actual execution of the suggested work and the results obtained. Here, the focus switches to the practical side, focussing on how the dataset was handled, how the models were trained, and what performance they ultimately produced, whereas the previous chapters primarily addressed the plan and design.

At the beginning, the process of implementation is outlined. This involves preparing the dataset, applying various preprocessing steps to clean and standardize the images, and setting up the deep neural network models for training. Each of these steps played an important role in making sure the system worked as expected.

After the implementation details, the results are presented and analyzed. Different models, including both the custom CNN and several pre-trained networks, are compared using standard measures such as accuracy, precision, recall, F1-score, and ROC-AUC. These outcomes are discussed in addition to being displayed numerically to emphasise the unique advantages and disadvantages of each model.

In short, this chapter connects the design with reality. It shows how the system moved from concept to execution and evaluates how well it performed in practice. The results also provide insight into which models are most suitable for real-world use, particularly in healthcare settings where accuracy and reliability are crucial.

4.1 Environment Setup

The Covid-19 Detection System was trained in an optimally configured AI environment for deep learning, including model training, image pre-processing, and assessment.

The experiments for this project were carried out in a simple but reliable environment that could handle deep learning workloads without difficulty. Instead of relying on expensive local machines, all training and testing were done on Google Colab, which provides free GPU access and a user-friendly interface. This made it possible to run complex models within a reasonable time frame while keeping the setup accessible to anyone with an internet connection.

Python 3, which has become the standard language for the majority of machine learning research due to its readability and abundance of open-source libraries, was used for the coding. The neural networks in this project were designed and trained using TensorFlow and Keras, while the dataset was managed and numerical operations were handled with the aid of NumPy and Pandas. OpenCV tools were used for image-related preprocessing.

Visualization also played an important role, so Matplotlib and Seaborn were used to create plots of training curves, confusion matrices, and ROC curves. All of these libraries worked smoothly together in the Colab environment, making the workflow consistent from the very beginning to the end of the project.

In short, the setup was chosen to balance performance and accessibility. Anyone with a

basic computer could reproduce the same experiments on Colab without needing specialized hardware, which makes the work more practical and easier to extend in future studies.

4.2 Comparative Analysis

To properly measure the effectiveness of our system, we tested six different models: one was our lightweight Custom CNN, and the other five were well-known pre-trained architectures (VGG16, ResNet50, DenseNet121, InceptionV3, and Xception). The evaluation was done on the reserved test set, which contained 465 chest X-ray images.

For each model, we carefully recorded not only the final accuracy but also the precision, recall, and F1-score. In addition, we plotted the confusion matrix to see how many cases were correctly or incorrectly classified, and the ROC curve to evaluate diagnostic reliability. Training history, such as accuracy and loss curves, was also tracked so that we could identify whether a model was learning properly or overfitting.

4.2.1 Custom CNN

For scenarios where high-end processing power is unavailable, the Custom CNN was designed as a faster and smaller model. Even with its simple structure, it achieved an accuracy of 95% and an F1-score of 0.95.

The confusion matrix (Fig. 4.1) shows that most cases were classified correctly, though a few COVID-positive images were marked as normal.

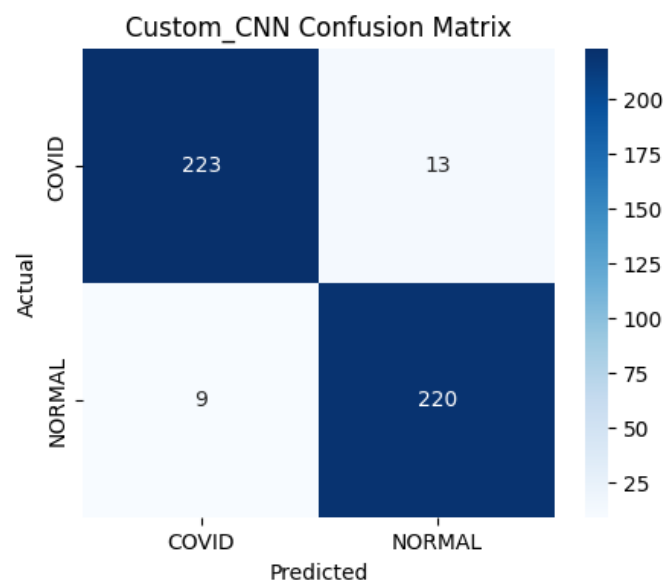


Figure 4.1 : Confusion Matrix Custom CNN

The model demonstrated minimal overfitting and rapidly converged within 15 epochs, according to the training and validation graphs (Fig.4.2)

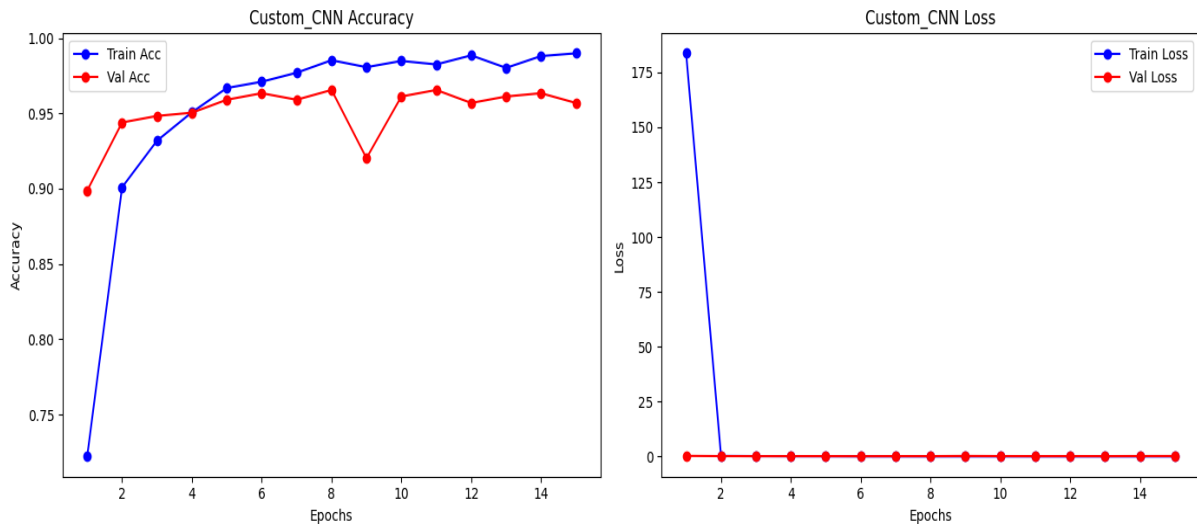


Figure:4.2 Train and Validation accuracy, loss curve for Custom CNN

4.2.2 VGG16

The VGG16 model give the best results of all pre trained model. It achieved 99% accuracy, with almost perfect precision, recall, and F1-score.

Confusion Matrix (Fig. 4.3): Nearly every COVID-19 and Normal case was predicted correctly, with almost no errors. This makes the model extremely dependable.

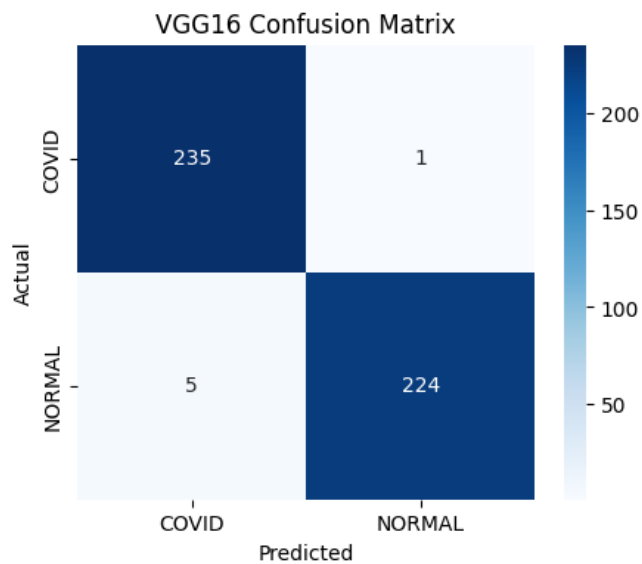


Figure 4.3: Confusion Matrix VGG16

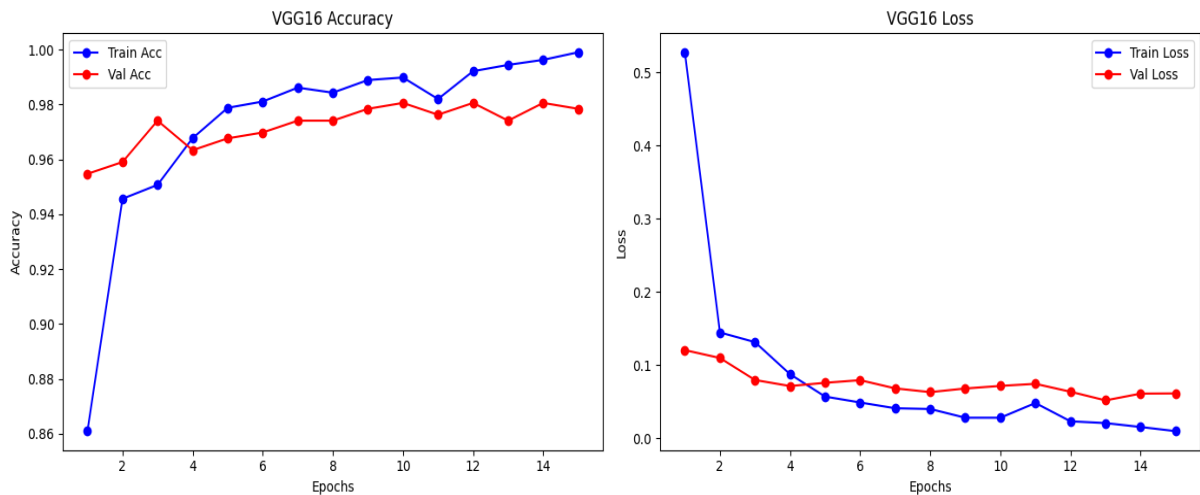


Figure 4.4 Train and Validation accuracy, loss curve VGG16

Training vs Validation Accuracy (Fig. 4.4): The curves rose quickly and stayed very close to each other. This proves that the model generalized well and did not rely only on the training data.

Training vs Validation Loss (Fig.4.4): Both curves declined smoothly with no visible gap. This confirms that VGG16 learned efficiently and avoided overfitting.

4.2.3 ResNet50

The ResNet50 model also performed strongly, reaching 98% accuracy with very high F1-scores.

Confusion Matrix (Fig. 4.5): Only a few cases were misclassified, showing that the model was reliable in identifying both classes.

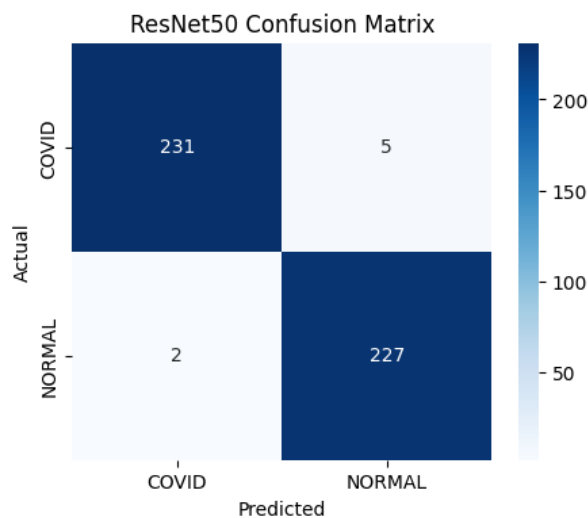


Figure: 4.5 Confusion Matrix ResNet50

Training vs Validation Accuracy (Fig. X.8): Both curves showed a steady upward trend and stayed close, meaning the model generalized well across unseen data.

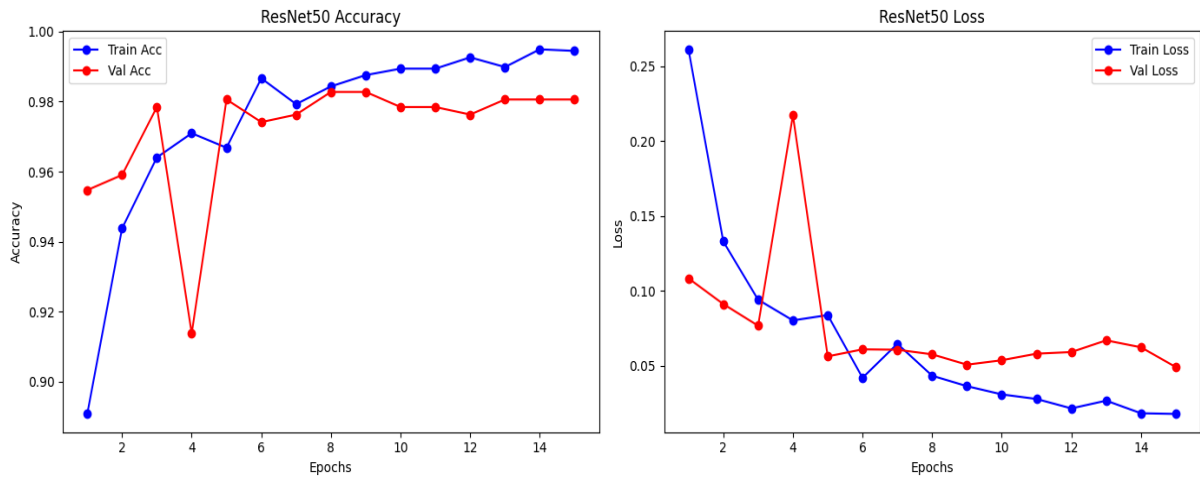


Figure 4.6 Train and Validation accuracy, loss curve ResNet50

Training vs Validation Loss (Fig. 4.6): The loss values dropped consistently and remained stable, confirming smooth convergence.

4.2.4 DenseNet121

The DenseNet121 model achieved 97% accuracy. It performed slightly below ResNet50 but still maintained strong classification ability.

Confusion Matrix (Fig. X.10): Most images were correctly classified, though a few extra errors appeared compared to VGG16 and ResNet50.

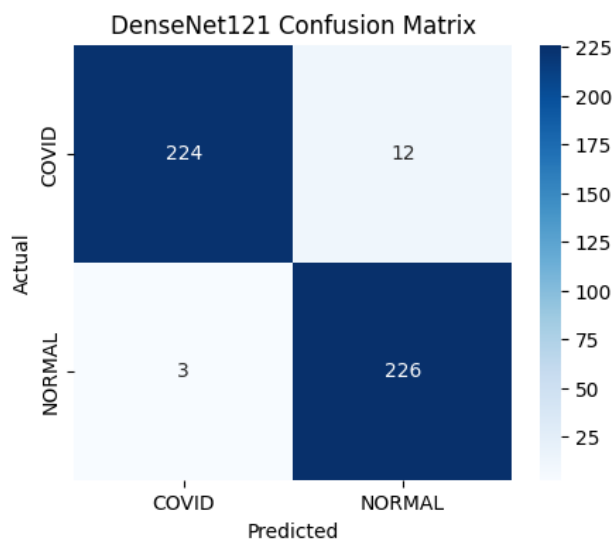


Figure 4.7 : Confusion Matrix DenseNet121

Training vs Validation Accuracy (Fig. 4.8): The training accuracy climbed quickly, but

the validation accuracy flattened earlier, showing that the model had reached its best performance sooner.

Training vs Validation Loss (Fig. 4.8): Loss curves decreased steadily, though a small gap appeared between training and validation loss, hinting at mild overfitting

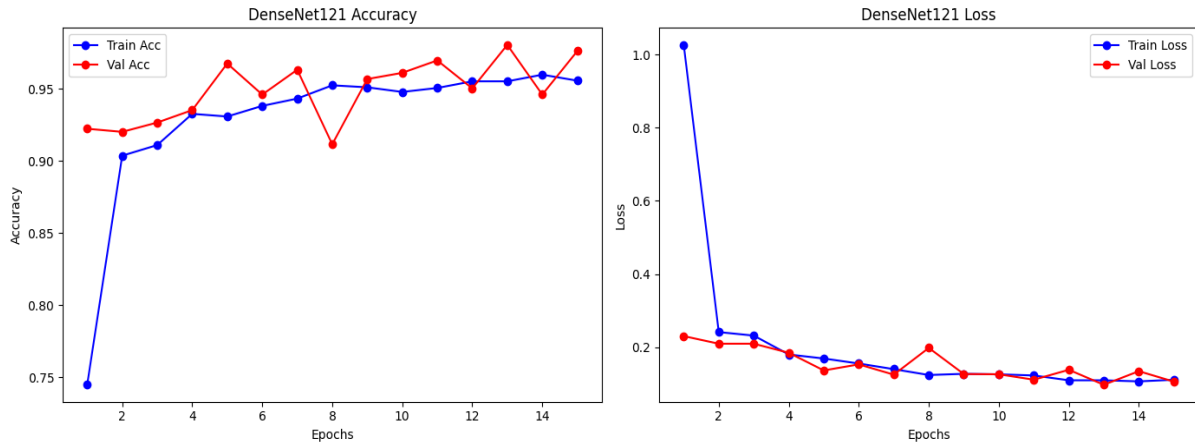


Figure 4.8 : Train and Validation accuracy, loss curve DenseNet121

4.2.5 InceptionV3

The InceptionV3 model gave 90% accuracy, which was clearly lower than the first three models.

Confusion Matrix (Fig. 4.9): The matrix shows a higher number of misclassifications, especially false negatives where COVID-19 cases were marked as Normal.

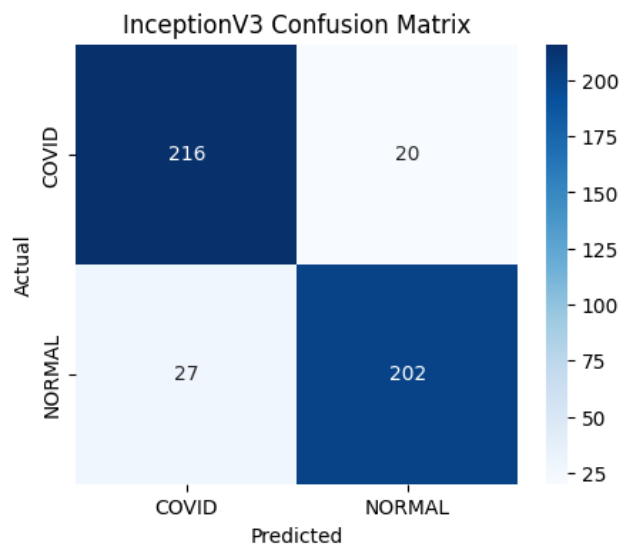


Figure 4.9 Confusion Matrix InceptionV3

Training vs Validation Accuracy (Fig. 4.10): The validation accuracy line showed ups

and downs, unlike the smoother training curve. This reflects instability during learning.

Training vs Validation Loss (Fig. 4.10): Validation loss stayed above training loss and did not drop as consistently, which suggests the model was beginning to overfit.

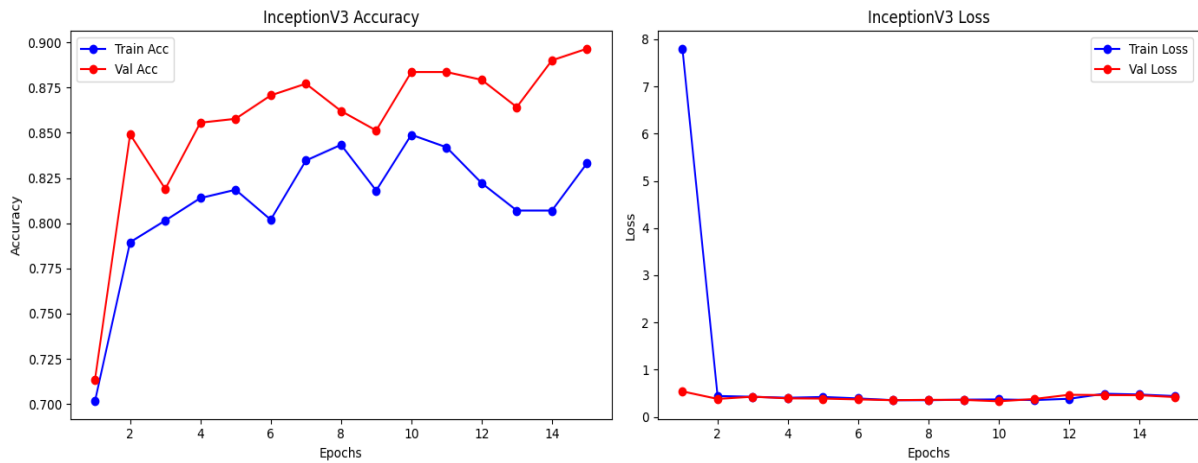


Figure 4.10 Train and Validation accuracy, loss curve InceptionV3

4.2.6 Xception

The Xception model performed the weakest, with 89% accuracy and the lowest F1-score

Confusion Matrix (Fig. 4.11): This model produced the highest number of false negatives, which is a major drawback in medical use.

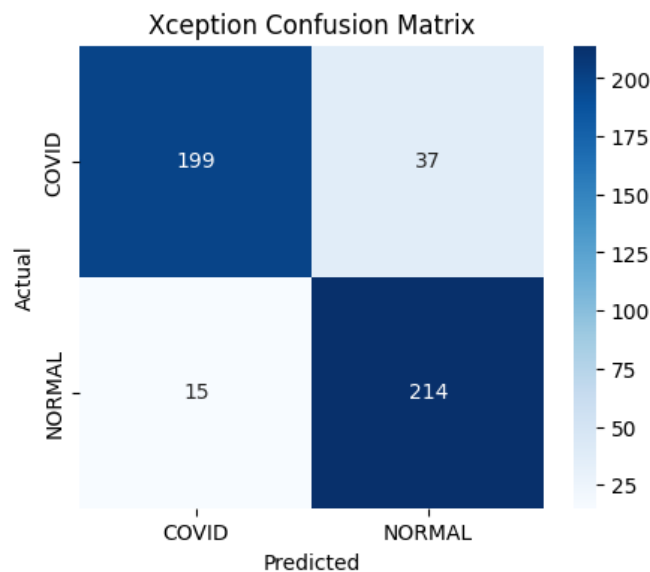


Figure 4.11 Confusion Matrix Xception

Training vs. Validation Accuracy (Fig. 4.12): Training accuracy increased steadily, but validation accuracy fluctuated irregularly and lagged behind.

Training vs. Validation Loss (Fig. 4.12): Overfitting was confirmed by the validation loss, which was still significantly higher than the training loss.

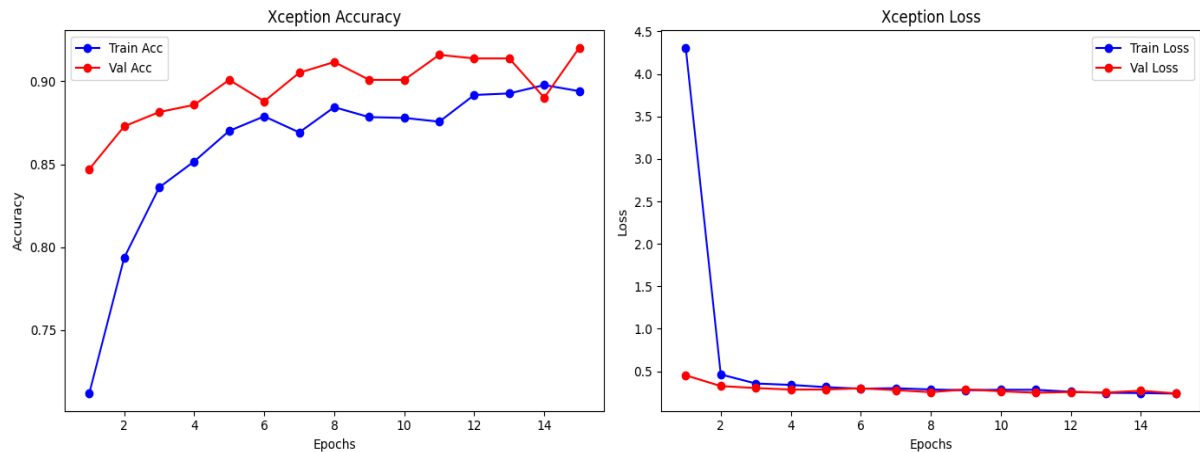


Figure 4.12 Train and Validation accuracy, loss curve Xception

Comparative Insights

From the above evaluation:

- VGG16 stood out as the most accurate and reliable model.
- ResNet50 and DenseNet121 also delivered strong performance, only slightly behind.
- The Custom CNN proved effective for limited-resource environments, balancing decent accuracy (95%) with efficiency.
- InceptionV3 and Xception underperformed, showing unstable training behavior and higher misclassification rates.

4.3 Results and Discussion

The purpose of the experiments conducted in this study was to assess the performance of pre-trained and custom-built deep learning models for the classification of normal chest X-ray images and COVID-19 images. A set of six models—Custom CNN, VGG16, ResNet50, DenseNet121, InceptionV3, and Xception—were trained and tested under the same experimental conditions to ensure a fair comparison. Accuracy, precision, recall, and F1-score—all common metrics in medical image analysis—were used to evaluate their performance.

Performance Comparison

The overall outcomes of the models are summarized in Table 4.1.

Model	Precision	Recall	F1-Score	Accuracy
Custom CNN	0.95	0.95	0.95	95%
VGG16	0.99	0.99	0.99	99%
ResNet50	0.99	0.99	0.98	98%
DenseNet121	0.97	0.97	0.97	97%
InceotionV3	0.90	0.90	0.90	90%
Xception	0.89	0.89	0.89	89%

The results offer a number of significant insights. First, in terms of raw accuracy and reliability, the pre-trained transfer learning models continuously beat the custom CNN. This was expected since these architectures rely on very deep networks that have been trained on massive datasets such as ImageNet, allowing them to accurately capture complex image features.

Here we see the graph below that show accuracy comparison .

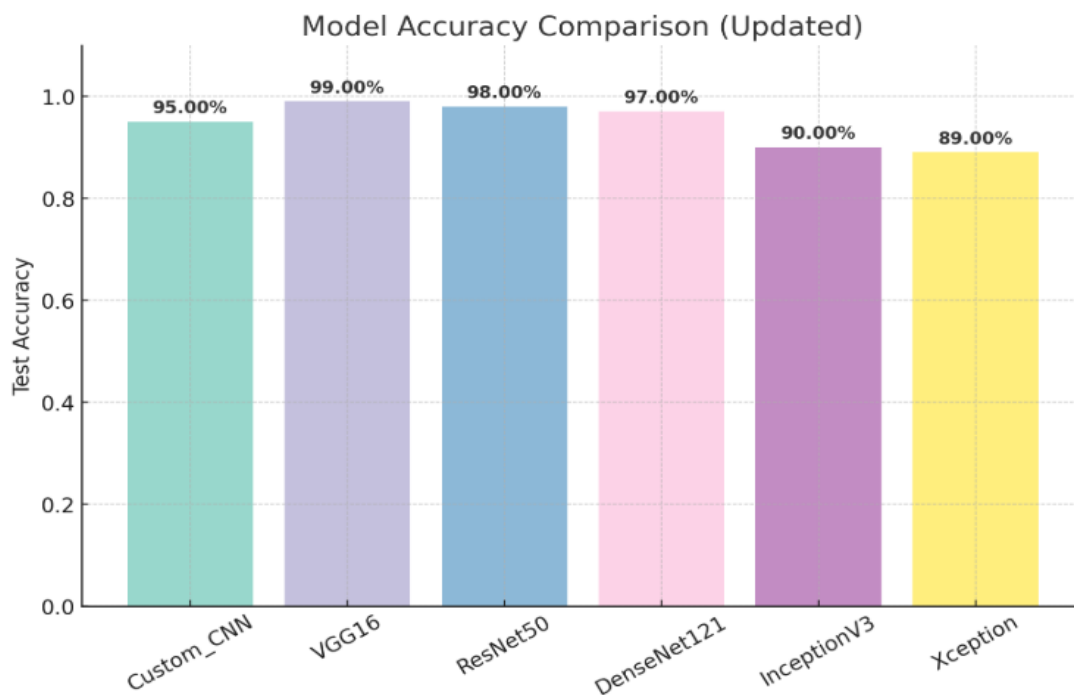


Figure 4.13 : Model accuracy comparison

Among all the models, VGG16 achieved the highest performance, touch 99% accuracy and F1-score. The ability of VGG16 to capture fine-grained differences between infected and healthy lungs highlights its suitability for COVID-19 diagnosis.

With a 98% accuracy rate, ResNet50 came in second. The introduction of residual connections, which avoids the issue of vanishing gradients in very deep networks, is its primary strength. This characteristic makes ResNet50 one of the most reliable architectures in contemporary deep learning applications by allowing it to learn more consistent and resilient patterns.

DenseNet121 also delivered strong performance with 97% accuracy. Its densely connected layers encourage feature reuse and strengthen gradient flow during training, which is

particularly useful in extracting subtle lung abnormalities from chest X-rays.

In contrast, InceptionV3 and Xception showed weaker results, with accuracies of 90% and 89%, respectively. These models are more complex, and although they perform exceptionally well in large and diverse image classification tasks, they seemed to overfit or underperform in this relatively limited dataset. The drop in their performance underlines that architectural sophistication alone does not guarantee success; dataset characteristics and model–task alignment are equally crucial.

A noteworthy aspect of this study is the performance of the custom CNN, which achieved 95% accuracy. The custom model was able to achieve competitive results even though it was far less complex than the pre-trained networks. This is a significant finding because it shows that networks that are lighter and smaller can still be very effective. Such effective models can be extremely helpful in real-world hospital settings, particularly in settings with limited resources and no access to powerful GPUs. They provide a balance between cost-effectiveness, speed, and diagnostic accuracy.

4.4 Summary

In this chapter we see how the project main work and also implementation and how it get results , the main focus was on showing how the project was carried out in practice and what results were achieved The setup of the working environment was described first so that the process of training and testing the models is clear. After that, the performance of different deep learning approaches was compared using the same dataset and evaluation measures.

From the experiments, it became evident that transfer learning models performed exceptionally well, with VGG16 standing out as the best among them. ResNet50 and DenseNet121 also gave highly reliable results, proving the strength of their advanced architectures. On the other hand, InceptionV3 and Xception did not perform as strongly, which underlined the fact that not every complex model is suitable for every dataset. The custom CNN, although much simpler, reached a strong accuracy level, which makes it very promising for hospitals that do not have advanced computing facilities.

Altogether, this section confirmed that deep learning can be a dependable tool for diagnosing COVID-19 from chest X-rays. It also showed that lightweight models should not be overlooked, since they can provide fast and practical solutions in real-world healthcare environments.

Chapter 5

Engineering Standards and Design Challenges

This chapter focuses on the main engineering standards that guided the development of the project and the key design challenges that were encountered along the way. It explains how recognized technical standards were followed to ensure reliability, safety, and compatibility, and also highlights the difficulties faced during data collection, preprocessing, and model implementation. By addressing both standards and challenges together, this chapter provides a practical view of the considerations involved in transforming an idea into a working system.

5.1 Compliance with the Standards

To guarantee that the COVID-19 detection system is dependable, scalable, and appropriate for medical use, it was imperative that pertinent standards be followed during development. The system design considered three categories of standards: software standards, hardware standards, and communication standards.

In terms of software standards, guidelines for coding practices and framework usage were applied to keep the implementation consistent and reproducible. On the hardware side, appropriate configurations were selected to handle deep learning workloads efficiently, while also keeping the solution practical for deployment in real hospital settings. In order to ensure that the system can seamlessly integrate with the current medical infrastructure, standards pertaining to secure data transfer and interoperability were examined.

Several alternative approaches were also studied. Each had its own strengths and weaknesses, but the final selections were made on the basis of performance, practicality, and alignment with healthcare requirements. By following these standards, the project ensures that the developed COVID-19 detection system is not only accurate but also dependable and fit for real-world application.

5.1.1 Software Standards

For this project, TensorFlow and Keras were selected because of their robustness, scalability, and strong adoption in medical AI. These frameworks work well with libraries like NumPy and OpenCV, support big datasets, and give users access to pre-trained models. Although they demand more computational resources, their rich documentation and community support make them highly reliable.

We mainly use Google Colab for this and we know google colab also provide Python3 and give GPU supports and enhance help my project fully .

Alternative: PyTorch was taken into consideration due to its ease of use and flexibility in research; however, it does not have the same deployment maturity or availability of medical AI models.

Rationale: Because they are more production-ready and more appropriate for scalable medical applications, TensorFlow and Keras were selected over PyTorch.

5.1.2 Hardware Standards

In this project, all experiments were carried out using Google Colab, which provided a cloud-based environment with 16 GB of RAM and an NVIDIA T4 GPU with 12 GB of memory. This setup was strong enough to train deep learning models on chest X-ray images within a reasonable time. The GPU support made it possible to handle image preprocessing, model training, and evaluation smoothly, while the large memory capacity ensured that the dataset could be managed without difficulty.

The decision to rely on Google Colab was made because it offered the best mix of accessibility, performance, and cost-effectiveness. It removed the need for expensive hardware while still delivering reliable results, making it the most practical solution for developing and testing the COVID-19 detection system.

5.1.3 Communication Standards

Good communication was one of the key factors behind the smooth execution of this project. Since the work was done in a collaborative environment, every stage included proper documentation and well-structured reporting. Google Colab notebooks were not only used for coding and experiments but also for organizing outputs, graphs, and notes. This made it easy to share results with teammates and supervisors so that progress could be reviewed and verified whenever needed.

GitHub and other version control systems were also taken into consideration for code management and code save and they help. This avoided repetitive work, decreased confusion, and made it easier to keep track of updates and changes. Emails, instant messaging, and planned meetings were used to keep internal communication going on a daily basis, which helped to keep the project on schedule.

The same consistency was applied while preparing this report. Citations were written in IEEE format, and a uniform template was followed throughout to keep the document professional and easy to read. Visual component such as charts, tables, and diagrams they are added where necessary to make the content clearer and more engaging.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The impact of this project on life is good meaningful. During the COVID-19 pandemic, many hospitals faced long delays in testing, and patients often had to wait for hours or even days to get their results. By using deep learning on chest X-rays, this system can give faster and more reliable support for diagnosis.

Another important impact is the cost. Compared to RT-PCR tests, which are not always accessible in remote areas, X-rays are more accessible and less costly. People in low-resource areas will have easier access to more affordable healthcare if an AI tool can interpret those X-rays and help physicians.

Beyond COVID-19, this project also shows a way to use similar systems for other diseases like pneumonia or tuberculosis. Early and correct diagnosis of such diseases can improve

survival rates and reduce pressure on hospitals.

Overall, the project technology highlights how this project can make human healthcare faster, cheaper and more widely available.

5.2.2 Impact on Society & Environment

When diseases like COVID-19 can be detected quickly with the help of AI, hospitals can manage patients more efficiently. This reduces the burden on healthcare workers, lowers waiting times, and makes medical services more organized. In a larger sense, society benefits because early detection helps to control the spread of infection, keeping communities safer.

Another impact is related to access. It is not always feasible to perform sophisticated tests like RT-PCR in rural or underdeveloped areas. But X-ray machines are more prevalent, and if AI can interpret them, people in those locations can also get timely assistance. This reduces inequality in healthcare and promotes social fairness.

From the environmental side, digital X-ray systems supported by AI are much better than repeated chemical-based tests. By reducing the use of disposable testing kits and laboratory chemicals, the system indirectly lowers medical waste.

In short, the project brings a positive influence not just for patients, but for society as a whole and the environment. Faster diagnosis, fair access, and more eco-friendly practices together show how technology can support both public health and sustainable development.

5.2.3 Ethical Aspects

When developing an AI system for healthcare, ethical issues must always be considered. One of the most important points is patient privacy. All medical images used in this project were taken from open-access datasets or collected with proper consent. Patients' personal information was never used, and the data was managed to preserve the patients' identities.

Another ethical aspect is fairness and bias. In some cases, if the training data is not balanced, AI systems may perform better for one group of people than another. To ensure that the models functioned fairly for both COVID and normal cases, we made an effort to maintain a balanced and diverse dataset for this project. This guarantees that a variety of patients can benefit from the tool and lowers the possibility of discrimination.

Transparency is also very important. Doctors and healthcare workers should not rely on a “black box” system without understanding how it makes decisions. For this reason, methods like heatmaps and performance reports were used to make the results explainable. This helps make trust between the AI system and the medical professionals who use it.

Finally, responsibility must be clear. AI is not intended to replace doctors; rather, it is meant to assist them. Qualified professionals should always make the final medical decision. This ensures that the technology is used safely and ethically in real clinical practice.

In short, ethical aspects such as privacy, fairness, transparency, and responsibility were considered carefully in this project to make sure the system can truly benefit patients and society in a safe way.

5.2.4 Sustainability Plan

For this project will remain useful in the long run, a proper sustainability plan is needed. First, updating the system should be simple. As new medical data becomes available, the AI models can be retrained on fresh datasets so that the results stay accurate and reliable. Using Google Colab and cloud storage makes this process simple, since new versions of the model can be trained and shared without heavy local hardware.

Another important point is cost. Because Colab provides free or low-cost GPU access, the system can be maintained without large expenses. Hospitals or research groups with limited resources can still use the tool, which supports long-term sustainability.

The plan also includes making the project open for collaboration. If doctors, researchers, or developers from various regions contribute data and improvements, the system will grow powerfull over time. This not only expands the life of the project but also ensures fairness across different populations.

From an environmental perspective, digital AI-based analysis reduces the need for repeated chemical-based tests, which cuts down medical waste. Cloud resources also avoid the need for setting up large local servers in every clinic, lowering energy use.

5.3 Project Management and Financial Analysis

Understanding the financial requirements is essential to project management. The overall cost is fairly reasonable because the work was done in Google Colab using primarily open-source tools. The comprehensive budget plan, up to BDT 60,000, is shown below.

Table 5.1 Budget Plan (Within BDT 60,000)

Category	Details	Estimated Cost (BDT)
Cloud Computing	Google Colab Pro/Pro+ (GPU access)	10,000 / year
Data Storage & Backup	Google Drive	6,000 / year
Software & Tools	Open-source frameworks	Free
Internet & Utilities	Reliable internet for cloud training	12,000
Documentation & Reports	Printing, diagrams, formatting, professional reports	5,000
Miscellaneous Expenses	Extra costs (unexpected needs)	7,000
Total Estimated Cost		~40,000 – 45,000

Safety Margin	For upgrades, more storage, unforeseen expenses	~15,000
----------------------	---	---------

Table 5.2 Revenue model

Model Type	Description
Institutional Subscription	Hospitals/diagnostic centers pay yearly fees.
Pay-per-Scan	Clinics/doctors pay per uploaded/analyzed X-ray.
Freemium Model	Free basic version with limited scans; premium for advanced features.

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

This section presents a mapping of the complex problem-solving categories to the specific engineering challenges addressed in this project. Each category is analyzed and mapped accordingly, with a rationale provided for each selection. The mapping is shown in **Table 5.1**, followed by detailed explanation for EP1 to EP7.

Table 5.1: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stakeholder Involvement	EP7 Interdependence
✓	✓	✓	✓	x	✓	✓

Mapping with Knowledge Profile

In alignment with **EP1**, which emphasizes the depth of knowledge necessary to solve complex engineering problems, this project integrates multiple domains of the knowledge profile.

- It applies **K3 (Engineering Fundamentals)** through the use of core principles in image preprocessing, convolutional neural networks, and evaluation metrics such as accuracy, precision, recall, and F1-score.
- It demonstrates **K4 (Specialist Knowledge)** by applying transfer learning with architectures like VGG16, ResNet50, and DenseNet121, and customizing them for COVID-19 classification tasks.
- It reflects **K5 (Engineering Design)** through the design, implementation, and optimization of both a custom CNN and multiple pre-trained models,

- ensuring a balance between accuracy and computational efficiency.
- It highlights **K6 (Engineering Practice)** in the use of practical tools such as Python, TensorFlow, Keras, and Google Colab for developing and testing the system.
 - It incorporates **K8 (Research Literature)** by grounding the project on prior peer-reviewed studies and medical imaging research, ensuring that the design decisions are evidence-based and scientifically sound.

Altogether, the project covers the necessary depth and breadth of engineering knowledge required under EP1.

Table 5.2: Mapping with knowledge Profile.

K3	K4	K5	K6	K8
Engineering Fundamentals	Specialist Knowledge	Engineering Design	Engineering Practice	Research Literature
✓	✓	✓	✓	✓

5.4.2 Justification for Mapping with Complex Problem Solving

- **EP1 – Depth of Knowledge:** Required advanced understanding of deep learning, medical imaging, and performance evaluation.
- **EP2 – Range of Conflicting Requirements:** The system needed to achieve very high accuracy while still being lightweight enough for low-resource hospital environments.
- **EP3 – Depth of Analysis:** Extensive preprocessing, model fine-tuning, and performance comparison were required to ensure reliability.
- **EP4 – Familiarity of Issues:** AI in medical imaging is not new, but COVID-19 detection posed a novel and urgent challenge.
- **EP5 – Applicable Codes:** No universal AI codes exist, but the project followed medical ethics, IEEE reporting standards, and data privacy practices.
- **EP6 – Stakeholder Involvement:** Multiple stakeholders were involved, including radiologists, hospitals, patients, and researchers.
- **EP7 – Interdependence:** Success required a combination of knowledge from computer science, healthcare, engineering design, and ethical practices.

5.4.3 Engineering Activities

This section provides the mapping of the project to complex engineering activities. Each activity (EA1–EA5) has been mapped in **Table 5.3**, followed by explanations and rationale.

Mapping with Complex Engineering Activities

Table 5.3: Mapping with Complex Engineering Activities.

EA1 Range of re- sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	x	✓	✓	✓

5.4.4 Justification for Mapping with Complex Engineering Activities

EA1: Range of Resources

Numerous cutting-edge resources were used in the project. Google Colab, which offered free GPU access, was used for all experiments. Model development was done using deep learning frameworks like TensorFlow and Keras, and preprocessing was done using libraries like NumPy, Pandas, and OpenCV. A carefully curated dataset of COVID-19 and Normal chest X-rays was collected from both hospitals and open-access repositories. Ethical guidelines were followed during data usage, and transfer learning with CNN architectures allowed for accurate classification. Altogether, the project required a broad range of computing, software, data, and ethical resources.

EA2: Level of Interaction

This project was primarily research-oriented, and most work was carried out in a digital environment. Direct patient interaction was not included, since the dataset was anonymized and pre-collected. For this reason, EA2 was not mapped. However, future deployment in hospitals may involve greater interaction between doctors, patients, and the system.

EA3: Innovation

In order to achieve high accuracy and computational efficiency, the project combined transfer learning with a custom CNN, introducing innovation. Even though it was lightweight, the custom CNN performed well, even though VGG16 and ResNet50 produced results that were almost flawless. This demonstrates that effective AI models can be developed for real-world hospitals with constrained budgets.

EA4: Consequences for Society and Environment

Faster and more accurate COVID-19 detection is made possible by the system, which directly improves public health. This allows physicians to treat patients more quickly and reduces the need for slower and more costly RT-PCR tests. Better illness control and more effective healthcare for society are the results of this. Utilizing AI-analyzed digital chest X-rays lowers medical waste and conserves resources by eliminating the need for chemical-based lab tests and disposable kits.

EA5: Familiarity

While AI in medical imaging is an active research field, applying it specifically for COVID-19 detection was a relatively new challenge

5.5 Summary

The engineering standards and design difficulties that influenced the project were covered in this chapter. To demonstrate how the system was constructed in an organized and expert manner, several elements were discussed, including software, communication, ethics, and sustainability. The budget and financial planning were also presented, including a cost-effective model within BDT 60,000 and an alternate higher-cost setup for larger institutions. A possible revenue model was outlined to ensure long-term sustainability.

The project was then mapped to the Complex Engineering Problems and Knowledge Profiles framework, demonstrating how it satisfies the requirements of interdependence, deep knowledge, conflicting requirements, and stakeholder involvement. The mapping showed that deep learning to solve COVID-19 detection calls for not only technical know-how but also knowledge of research literature, engineering practice, and ethics.

Finally, the Engineering Activities (EA1–EA5) were mapped, highlighting the wide range of resources used, the innovative design, the consequences for society and the environment, and the balance between familiarity and novelty.

Chapter 6

Conclusion

This chapter brings together the key points of the work and reflects on what has been achieved so far. It summarizes the overall process, the findings, and the importance of the project in the context of healthcare.

6.1 Summary

Throughout this project, our main objective was to design and evaluate a deep learning-based system for detecting COVID-19 from chest X-ray images. The work began with a detailed review of related studies, which helped us understand the limitations of existing methods and identify where our approach could make a contribution.

We then collected and prepared a dataset of chest X-rays, applying preprocessing steps such as grayscale conversion, resizing, normalization, and augmentation to ensure that the data was clean and consistent. Both a custom lightweight CNN and several pre-trained models (VGG16, ResNet50, DenseNet121, InceptionV3, and Xception) were tested under the same conditions. The models were trained, validated, and evaluated using standard performance metrics like accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices.

The results showed that while pre-trained models, particularly VGG16, achieved the highest accuracy, our custom CNN also performed strongly despite its simpler structure. This highlighted the balance between efficiency and accuracy—making it clear that lightweight models can still provide real value in low-resource hospital environments.

Finally, the system was integrated into a simple web application where users can upload chest X-rays and view predictions instantly. Beyond COVID-19, the pipeline was designed to be flexible so that it can be extended to other respiratory diseases such as pneumonia or tuberculosis in the future.

In short, this project successfully demonstrated that artificial intelligence can support medical diagnosis by providing fast, accurate, and practical solutions.

6.1 Limitation

This endeavor, like any other scientific initiative, has some limits. The dataset was well-prepared, but it was still modest compared to the huge image collections that are usually needed for deep learning. The model could be even better with a bigger dataset that has more varied examples. Another problem was that the pictures were taken from a small number of places, which may not properly show the range of X-rays observed in hospitals around the world.

In terms of technology, the models were trained and evaluated in Google Colab with some hardware limits. This often made it hard to make the models deeper or more complicated because training bigger models would need more computing resources. Lastly, even though the system was put into place on the web, it was never completely tested in a real hospital, therefore real-world problems like data protection, workflow integration, and scalability were not entirely solved in this project.

6.2 Future Work

Even with these problems, there are many ways that this project can flourish. First, gathering a much bigger and more varied dataset from several hospitals would make the models considerably more reliable and useful in general. The system might also be expanded to include more respiratory ailments, such pneumonia, tuberculosis, or lung cancer, which would make it a more complete diagnosis tool.

In the future, researchers could try using more advanced deep learning methods like ensemble learning or attention-based models to see if they can boost accuracy any further. Putting the system into real hospitals is also a crucial next step since it would let clinicians test it in real-world situations and give valuable input.

Finally, making the system more useful for real clinical usage by incorporating better data privacy and security features, such DICOM compatibility and compliance with healthcare rules, would be a good idea. These changes could make the project a dependable AI-based assistant for radiologists and other healthcare workers.

References

- [1] J. D. Arias-Londoño, J. A. Gómez-García, L. Moro-Velázquez and J. I. Godino-Llorente, "Artificial Intelligence Applied to Chest X-Ray Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach," in IEEE Access, vol. 8, pp. 226811-226827, 2020, doi: 10.1109/ACCESS.2020.3044858.
<https://ieeexplore.ieee.org/document/9293268?signout=success>
- [2] P. Bhowal, S. Sen, J. H. Yoon, Z. W. Geem and R. Sarkar, "Choquet Integral and Coalition Game-Based Ensemble of Deep Learning Models for COVID-19 Screening From Chest X-Ray Images," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 12, pp. 4328-4339, Dec. 2021, doi: 10.1109/JBHI.2021.3111415.
<https://ieeexplore.ieee.org/document/9534669>
- [3] P. M. Shah et al., "Deep GRU-CNN Model for COVID-19 Detection From Chest X-Rays Data," in IEEE Access, vol. 10, pp. 35094-35105, 2022, doi: 10.1109/ACCESS.2021.3077592.
<https://ieeexplore.ieee.org/document/9423965>
- [4] S. Hussain, G. A. Amran, A. Alabrah, L. Alkhalil and A. A. Al-Bakhrani, "C19-MLE: A Multi-Layer Ensemble Deep Learning Approach for COVID-19 Detection Using Cough Sounds and X-Ray Imaging," in IEEE Access, vol. 12, pp. 197151-197167, 2024, doi: 10.1109/ACCESS.2024.3521938.
<https://ieeexplore.ieee.org/document/10813173>
- [5] T. Kausar, Y. Lu, A. Kausar, M. Ali and A. Yousaf, "SD-GAN: A Style Distribution Transfer Generative Adversarial Network for Covid-19 Detection Through X-Ray Images," in IEEE Access, vol. 11, pp. 24545-24560, 2023, doi: 10.1109/ACCESS.2023.3253282.
<https://ieeexplore.ieee.org/document/10061161>
- [6] S. Calderon-Ramirez et al., "Improving Uncertainty Estimation With Semi-Supervised Deep Learning for COVID-19 Detection Using Chest X-Ray Images," in IEEE Access, vol. 9, pp. 85442-85454, 2021, doi: 10.1109/ACCESS.2021.3085418.
<https://ieeexplore.ieee.org/document/9445026>
- [7] Panetta K, Sanghavi F, Agaian S, Madan N. Automated Detection of COVID-19 Cases on Radiographs using Shape-Dependent Fibonacci-p Patterns. IEEE J Biomed Health Inform. 2021 Jun;25(6):1852-1863. doi: 10.1109/JBHI.2021.3069798. Epub 2021 Jun 3. PMID: 33788696; PMCID: PMC8768975.
<https://pubmed.ncbi.nlm.nih.gov/33788696/>
- [8] J. De Moura et al., "Deep Convolutional Approaches for the Analysis of COVID-19 Using Chest X-Ray Images From Portable Devices," in IEEE Access, vol. 8, pp. 195594-195607, 2020, doi: 10.1109/ACCESS.2020.3033762.
<https://ieeexplore.ieee.org/document/9239272>
- [9] C. Zhou, J. Song, S. Zhou, Z. Zhang and J. Xing, "COVID-19 Detection Based on Image Regrouping and Resnet-SVM Using Chest X-Ray Images," in IEEE Access, vol. 9, pp. 81902-81912, 2021, doi: 10.1109/ACCESS.2021.3086229.
<https://ieeexplore.ieee.org/document/9446895>
- [10] O. N. Oyelade, A. E. -S. Ezugwu and H. Chiroma, "CovFrameNet: An Enhanced Deep Learning Framework for COVID-19 Detection," in IEEE Access, vol. 9, pp. 77905-77919, 2021, doi: 10.1109/ACCESS.2021.3083516.
<https://ieeexplore.ieee.org/document/9440437>
- [11] E. F. Ohata et al., "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 1, pp. 239-248, January 2021, doi: 10.1109/JAS.2020.1003393.
<https://ieeexplore.ieee.org/document/9205687>
- [12] M. E. H. Chowdhury et al., "Can AI Help in Screening Viral and COVID-19 Pneumonia?," in IEEE Access, vol. 8, pp. 132665-132676, 2020, doi: 10.1109/ACCESS.2020.3010287.
<https://ieeexplore.ieee.org/document/9144185>
- [13] M. Gazda, J. Plavka, J. Gazda and P. Drotár, "Self-Supervised Deep Convolutional Neural Network for Chest X-Ray Classification," in IEEE Access, vol. 9, pp. 151972-151982, 2021, doi: 10.1109/ACCESS.2021.3125324.

- <https://ieeexplore.ieee.org/document/9600845>
- [14] Rajaraman S, Siegelman J, Alderson PO, Folio LS, Folio LR, Antani SK. Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-rays. IEEE Access. 2020;8:115041-115050. doi: 10.1109/access.2020.3003810. Epub 2020 Jun 19. PMID: 32742893; PMCID: PMC7394290.
<https://pubmed.ncbi.nlm.nih.gov/32742893/>
- [15] Sakib S, Tazrin T, Fouda MM, Fadlullah ZM, Guizani M. DL-CRC: Deep Learning-Based Chest Radiograph Classification for COVID-19 Detection: A Novel Approach. IEEE Access. 2020 Sep 18;8:171575-171589. doi: 10.1109/ACCESS.2020.3025010. PMID: 34976555; PMCID: PMC8675549.
<https://pmc.ncbi.nlm.nih.gov/articles/PMC8675549/>
- [16] Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR. CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. IEEE Access. 2020 May 14;8:91916-91923. doi: 10.1109/ACCESS.2020.2994762. PMID: 34192100; PMCID: PMC8043420.
<https://pmc.ncbi.nlm.nih.gov/articles/PMC8043420/>
- [17] M. Yamaç, M. Ahishali, A. Degerli, S. Kiranyaz, M. E. H. Chowdhury and M. Gabbouj, "Convolutional Sparse Support Estimator-Based COVID-19 Recognition From X-Ray Images," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 5, pp. 1810-1820, May 2021, doi: 10.1109/TNNLS.2021.3070467.
<https://ieeexplore.ieee.org/document/9408240>
- [18] M. M. Islam, F. Karray, R. Alhajj and J. Zeng, "A Review on Deep Learning Techniques for the Diagnosis of Novel Coronavirus (COVID-19)," in IEEE Access, vol. 9, pp. 30551-30572, 2021, doi: 10.1109/ACCESS.2021.3058537.
<https://ieeexplore.ieee.org/document/9351923>
- [19] El-Kenawy EM, Mirjalili S, Ibrahim A, Alrahmawy M, El-Said M, Zaki RM, Eid MM. Advanced Meta-Heuristics, Convolutional Neural Networks, and Feature Selectors for Efficient COVID-19 X-Ray Chest Image Classification. IEEE Access. 2021 Feb 22;9:36019-36037. doi: 10.1109/ACCESS.2021.3061058. PMID: 34812381; PMCID: PMC8545230.
<https://pubmed.ncbi.nlm.nih.gov/34812381/>
- [20] M. Abdel-Basset, R. Mohamed, M. Elhoseny, R. K. Chakraborty and M. Ryan, "A Hybrid COVID-19 Detection Model Using an Improved Marine Predators Algorithm and a Ranking-Based Diversity Reduction Strategy," in IEEE Access, vol. 8, pp. 79521-79540, 2020, doi: 10.1109/ACCESS.2020.2990893.
<https://ieeexplore.ieee.org/document/9079859>
- [21] Nguyen DC, Ding M, Pathirana PN, Seneviratne A. Blockchain and AI-Based Solutions to Combat Coronavirus (COVID-19)-Like Epidemics: A Survey. IEEE Access. 2021 Jun 30;9:95730-95753. doi: 10.1109/ACCESS.2021.3093633. PMID: 34812398; PMCID: PMC8545197.
<https://pmc.ncbi.nlm.nih.gov/articles/PMC8545197/>
- [22] L. Zhou et al., "A Rapid, Accurate and Machine-Agnostic Segmentation and Quantification Method for CT-Based COVID-19 Diagnosis," in IEEE Transactions on Medical Imaging, vol. 39, no. 8, pp. 2638-2652, Aug. 2020, doi: 10.1109/TMI.2020.3001810.
<https://ieeexplore.ieee.org/document/9115057>
- [23] Giuste F, Shi W, Zhu Y, Naren T, Isgut M, Sha Y, Tong L, Gupte M, Wang MD. Explainable Artificial Intelligence Methods in Combating Pandemics: A Systematic Review. IEEE Rev Biomed Eng. 2023;16:5-21. doi: 10.1109/RBME.2022.3185953. Epub 2023 Jan 5. PMID: 35737637.
<https://pubmed.ncbi.nlm.nih.gov/35737637/>
- [24] M. Nahiduzzaman et al., "A Novel Method for Multivariant Pneumonia Classification Based on Hybrid CNN-PCA Based Feature Extraction Using Extreme Learning Machine With CXR Images," in IEEE Access, vol. 9, pp. 147512-147526, 2021, doi: 10.1109/ACCESS.2021.3123782.
<https://ieeexplore.ieee.org/document/9591581>
- [25] Singh G, Yow KC. An Interpretable Deep Learning Model for Covid-19 Detection With Chest X-Ray Images. IEEE Access. 2021 Jun 8;9:85198-85208. doi: 10.1109/ACCESS.2021.3087583. PMID: 35256923; PMCID: PMC8864958.
<https://pubmed.ncbi.nlm.nih.gov/35256923/>
- [26] Bridge J, Meng Y, Zhao Y, Du Y, Zhao M, Sun R, Zheng Y. Introducing the GEV Activation Function for Highly Unbalanced Data to Develop COVID-19 Diagnostic Models. IEEE J Biomed Health Inform. 2020 Oct;24(10):2776-2786. doi: 10.1109/JBHI.2020.3012383. Epub 2020 Jul 28. PMID: 32750973; PMCID: PMC8545159.
<https://pubmed.ncbi.nlm.nih.gov/32750973/>
- [27] Benaliouche, Houda & Hafı, Houda & Bendjenna, Hakim & Alshaikh, Zeyad. (2025). Toward AI-Driven Cough Sound Analysis for Respiratory Disease Diagnosis. IEEE Access. PP. 1-1. 10.1109/ACCESS.2025.3572202.
https://www.researchgate.net/publication/391967464_Towards_AI-Driven_Cough_Sound_Analysis_for_Respiratory_Disease_Diagnosis/citation/download
- [28] M. K. Gourisaria, V. Singh, R. Chatterjee, S. K. Panda, M. R. Pradhan and B. Acharya, "PneuNetV1: A Deep Neural Network for Classification of Pneumothorax Using CXR Images," in IEEE Access, vol. 11, pp. 65028-65042, 2023, doi: 10.1109/ACCESS.2023.3289842.
<https://ieeexplore.ieee.org/document/10164097>
- [29] Md. Zahirul Islam, Md. Milon Islam, Amanullah Asraf,
A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images,
Informatics in Medicine Unlocked,
Volume 20,2020,100412,ISSN 2352-9148,
<https://www.sciencedirect.com/science/article/pii/S2352914820305621>

213-15-4470

ORIGINALITY REPORT

20% SIMILARITY INDEX	13% INTERNET SOURCES	13% PUBLICATIONS	10% STUDENT PAPERS
--------------------------------	--------------------------------	----------------------------	------------------------------

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	ela.kpi.ua Internet Source	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	Anooja Idupulapaati, M Sreedevi. "COVID-19 Detection from Chest Radiograph Images of Patients using Neural Networks", 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022 Publication	1%
5	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dharendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	1%
6	export.arxiv.org Internet Source	1%
7	www.wjgnet.com Internet Source	<1%
8	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	<1%

