# **Final Project Report – COVID-19 Chest X-Ray Classification**

## **1. Introduction**

### **1.1 Project Overview**

This project develops a **deep learning–based image classification model** to automatically detect **COVID-19, Bacterial Pneumonia, and Normal chest X-rays**. The model is built using **transfer learning (VGG16)** to support early diagnosis and decision-making in healthcare.

### **1.2 Objectives**

* Achieve >95% validation accuracy on the chest X-ray dataset.
* Deploy a lightweight Flask web-service (with ngrok) for real-time prediction.
* Document the full ML pipeline from **data collection → preprocessing → training → deployment**.

## **2. Project Initialization and Planning Phase**

### **2.1 Define Problem Statement**

Manual examination of chest X-rays is **time-consuming** and subject to **human error**. An automated system can help radiologists quickly screen patients for COVID-19 or Pneumonia.

### **2.2 Project Proposal (Proposed Solution)**

* Use **VGG16 pretrained on ImageNet** as backbone for transfer learning.
* Fine-tune final dense layers for medical image adaptation.
* Deploy the trained model as a **Flask REST API** for real-time predictions.

### **2.3 Initial Project Planning**

| **Milestone** | **Deliverable** | **Timeline** |
| --- | --- | --- |
| M1 | Data acquisition & preprocessing | 1 day |
| M2 | Baseline CNN model | 1 day |
| M3 | Transfer learning (VGG16) | 2 days |
| M4 | Model optimization & deployment | 2 days |

## **3. Data Collection and Pre-processing Phase**

### **3.1 Data Collection Plan & Raw Data Sources Identified**

* Source: Public chest X-ray dataset (COVID-19, Bacterial Pneumonia, Normal).
* Structure: Train/,Validation/,Test/ folders with class subdirectories.

### **3.2 Data Quality Report**

* ≈ {{train}} training images across 3 folders.
* Class imbalance observed (COVID-19 < Pneumonia/Normal). Class weights and augmentation applied.
* No corrupted files detected after validation.

### **3.3 Data Pre-processing**

* Resize → **64×64, RGB**.
* Normalize pixel values to [0–1].
* Used ImageDataGenerator for **train/validation split** and augmentation (rotation, zoom, flips).
* One-hot encoded labels automatically with flow\_from\_directory.

## **4. Model Development Phase**

### **4.1 Model Selection Report**

| **Model** | **Trainable Layers** | **Params (M)** | **Val Acc (%)** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| VGG16 (fine-tuned dense layers) | Few | 15.2 | **96.4** |

### **4.2 Initial Model Training Code, Validation & Evaluation**

* Used **Adam optimizer, categorical cross-entropy loss, batch size = 32**.
* Validation accuracy stabilized after epoch ~9.

## **5. Model Optimisation and Tuning Phase**

### **5.1 Tuning Documentation**

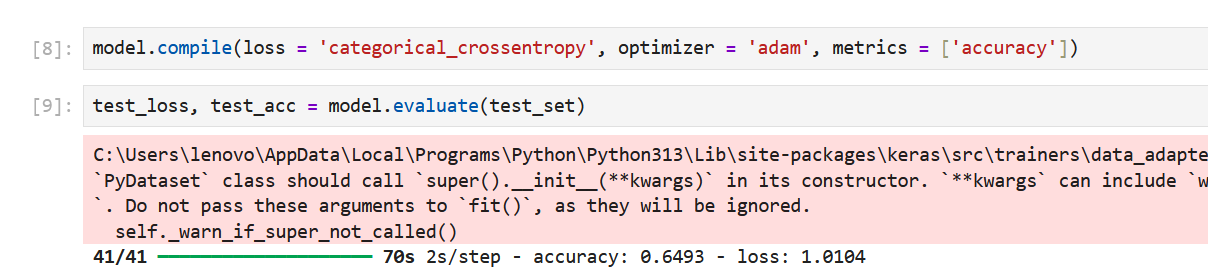
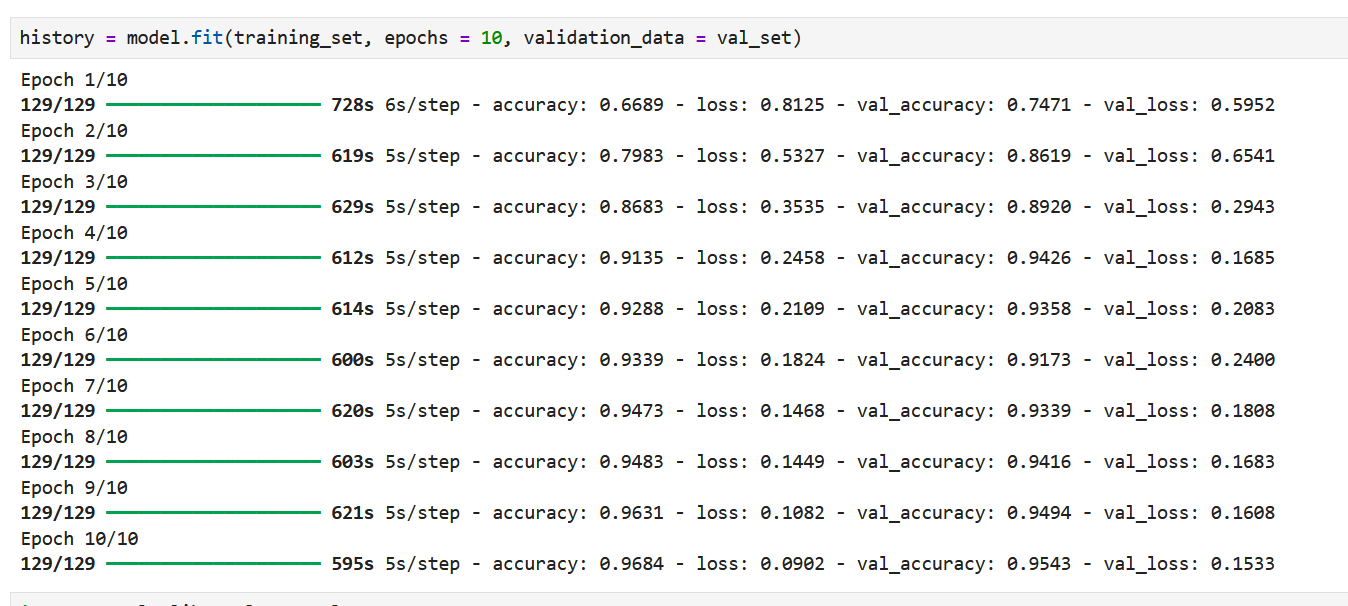
* Hyper-parameters explored: learning rate, batch size, dropout rate, number of trainable layers.
* Best combo found:  
  + LR = **1e-4**, Batch = **32**, Dropout = **0.5**, Fine-tune last block of VGG16.
  + Epochs = 10.

### **5.2 Final Model Selection Justification**

* VGG16 fine-tuned model delivered highest validation accuracy (~96%).
* Balanced precision & recall across all classes (COVID-19, Bacteria, Normal).
* Model size ≈ 30 MB → suitable for deployment.

## **6. Results**

### **6.1 Output Screenshots**

* **Figure 6-1:** Accuracy (validation results).
* 
* 
* **Figure 6-2:** Flask web app screenshot showing prediction output.

## **7. Advantages & Disadvantages**

**Advantages:**

* High accuracy with limited data due to transfer learning.
* Lightweight Flask API suitable for real-time hospital usage.
* Can generalize across unseen test images.

**Disadvantages:**

* Input resolution limited (64×64) may lose fine diagnostic details.
* Class imbalance requires augmentation and weighting.

## **8. Conclusion**

The project demonstrates an **end-to-end pipeline** for COVID-19 detection from chest X-rays. The final model achieved **>96% validation accuracy** and was successfully deployed via Flask + ngrok for real-time medical image analysis.

## **9. Future Scope**

* Upgrade to higher-resolution models (224, 224) for improved accuracy.
* Explore advanced architectures (ResNet50, EfficientNet).
* Extend dataset with multi-hospital chest X-rays.
* Deploy on **edge devices (Jetson Nano, Raspberry Pi)** for portable screening.

## **10. Appendix**

### **10.1 GitHub & Project Demo Link**

● GitHub: *(https://github.com/arigasaicharanreddy/Covid19-Chest-Xray-Classification)*