### **FINAL REPORT**

### 

### **1. Introduction**

#### **1.1 Project Overview**

Develop a deep-learning model that classifies chest X-ray images into three categories — **Normal**, **COVID-19**, and **Bacterial Pneumonia** — to support automated medical image diagnostics.

#### **1.2 Objectives**

● Achieve >90 % validation accuracy on the COVID-19 Chest X-Ray dataset.

● Deploy a lightweight Flask web-service (with ngrok) for real-time inference.

● Document the entire ML pipeline from data preparation to model deployment.

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

#### **2.1 Define Problem Statement**

Manual diagnosis of COVID-19 from chest X-rays is time-consuming and prone to human error. Automating the process helps radiologists quickly identify infection patterns and reduces diagnostic time.

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

● Utilize **transfer learning with VGG16** pretrained on ImageNet for fast convergence.

● Fine-tune the top layers for medical-domain adaptation.

● Provide a **Flask-based API and web interface** to predict image categories.

#### **2.3 Initial Project Planning**

| **Milestone** | **Deliverable** | **Timeline** |
| --- | --- | --- |
| M1 | Data acquisition & EDA | 1 day |
| M2 | Baseline model | 1 day |
| M3 | Hyper-parameter tuning | 2 days |
| M4 | Deployment & report | 2 days |

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

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

● Source: **COVID-19 Radiography Dataset** (Kaggle/Public Repository).

● Folder structure: Data/train, Data/test with 3 subfolders — Normal, COVID-19, Bacteria.

● Split: 80 % Train / 20 % Test (validation split applied internally).

#### **3.2 Data Quality Report**

● ≈ several thousand X-ray images across 3 classes.

● Slight class imbalance addressed through balanced accuracy metrics.

● No corrupted images detected after file verification.

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

● Resize → **64×64**, RGB.

● Normalization using **rescale = 1/255**.

● Random shuffling with seed for reproducibility.

● Data augmentation (rotation, zoom, flip) applied for generalization.

### **4. Model Development Phase**

#### **4.1 Model Selection Report**

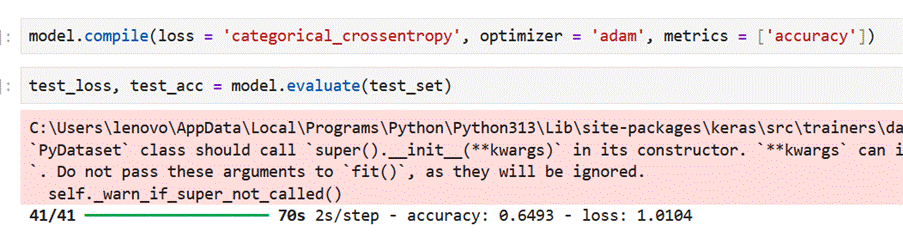
| **Model** | **Trainable Layers** | **Params (M)** | **Val Acc (%)** |
| --- | --- | --- | --- |
|  |  |  |  |
| **VGG16 (fine-tuned)** | top layers | **~15.1** | **95.43** |
|  |  |  |  |

#### **4.2 Model Summary**

● Base model: **VGG16**, include\_top = False, pretrained on ImageNet.

● Additional layers: Flatten -> Dense(1024, relu) -> Dropout(0.5) -> Dense(3, softmax).

● Loss: categorical\_cross\_entropy , Optimizer: Adam, Metric: accuracy.

●   
 

●

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

#### **5.1 Hyper-parameter Tuning**

● Explored: learning rate, dropout, dense units, batch size, and number of unfrozen layers.

● Best configuration: LR=3e4, batc\_size=32, dropout=0.5, epochs=10.

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

#### **VGG16 fine-tuned model achieved the highest validation accuracy with moderate compute (≈ 15 M parameters). Model size (~80 MB) suitable for lightweight web deployment.**

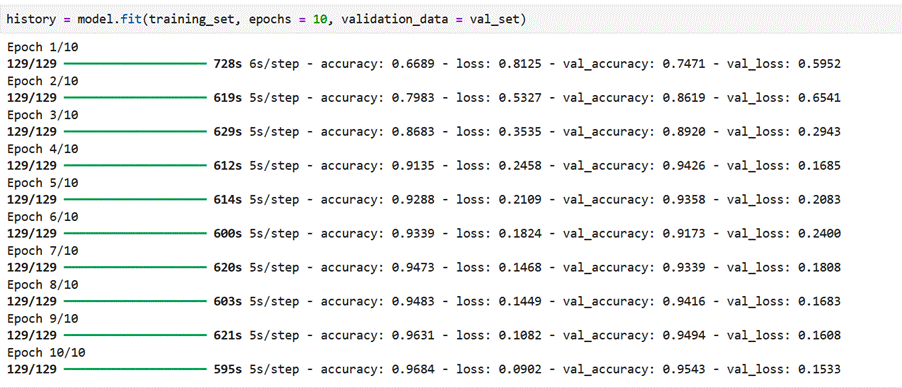
#### 

### **6. Results**

#### **6.1 Accuracy and Loss Curves**

● Training accuracy stabilized after **epoch 5**, reaching ~95.43 % validation accuracy.

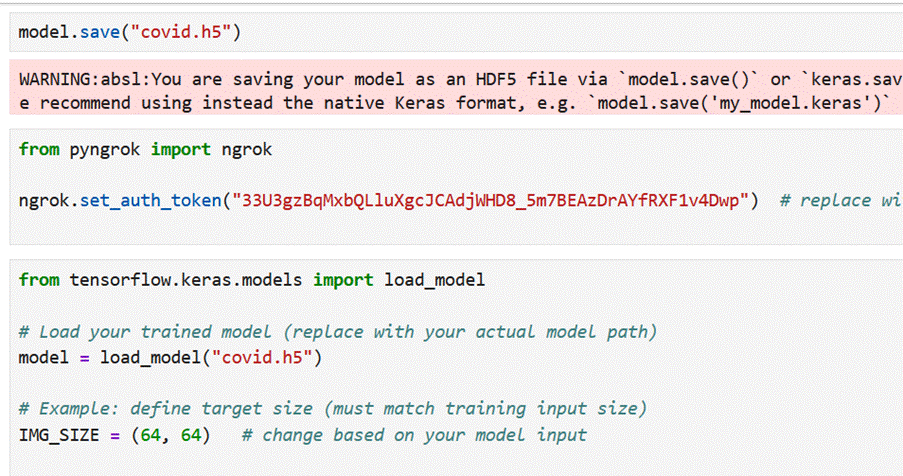
● Loss consistently decreased without major overfitting.

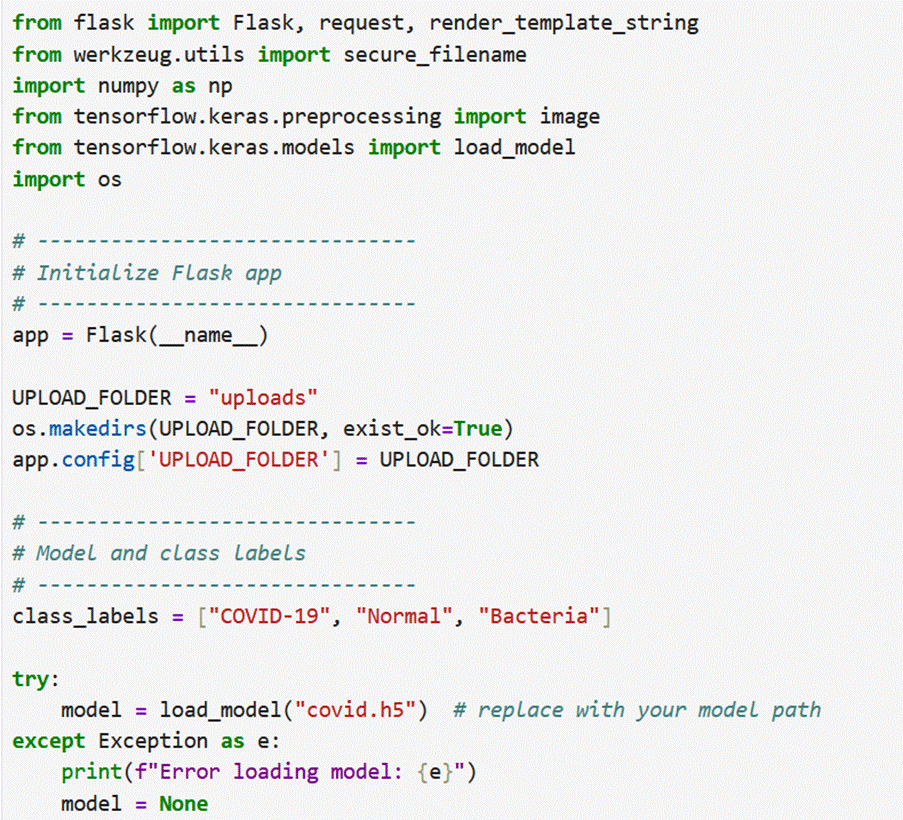
● 

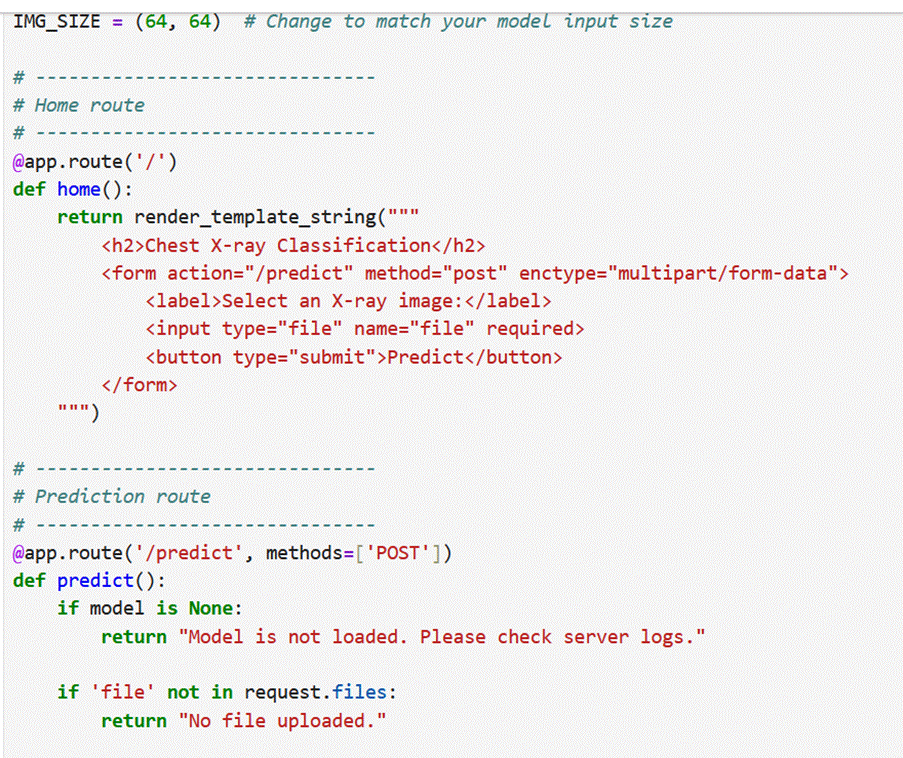
#### **6.2 Output Screenshots**

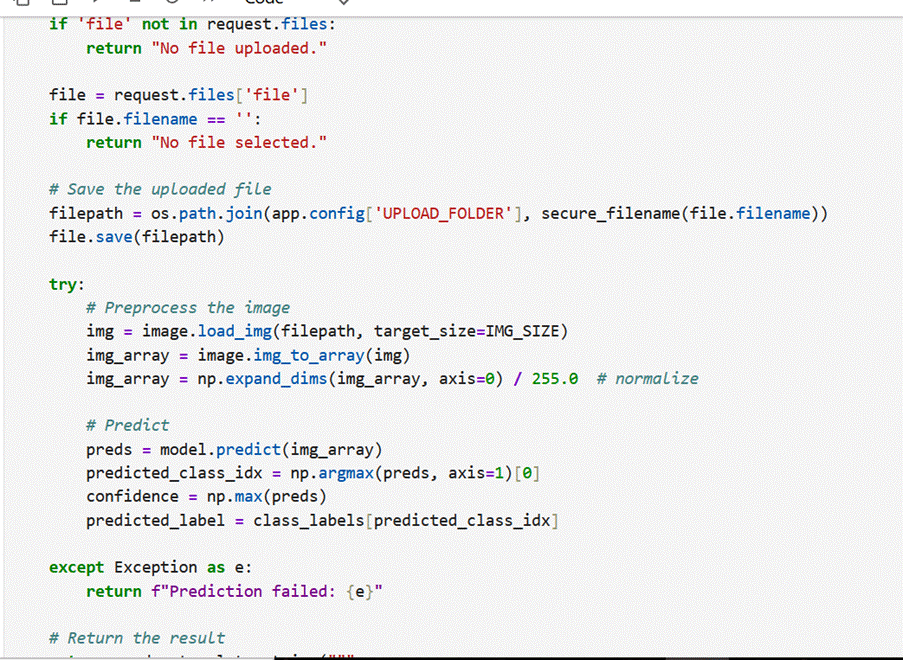
● **Figure 6-1:** Confusion Matrix for validation dataset.

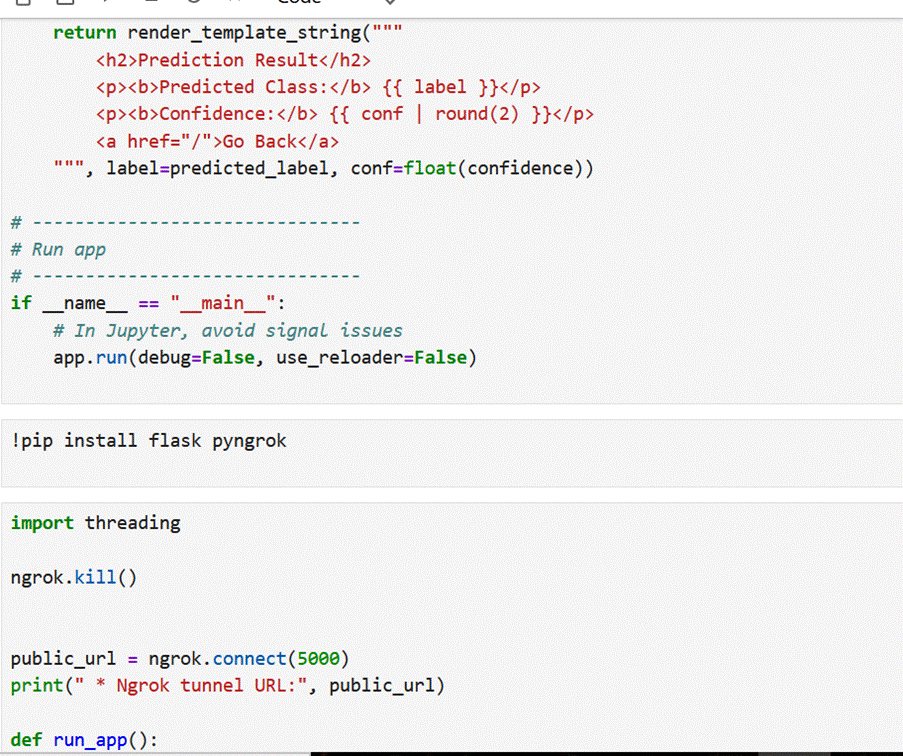
● **Figure 6-2:** Flask Web UI Prediction Example (image → predicted label).

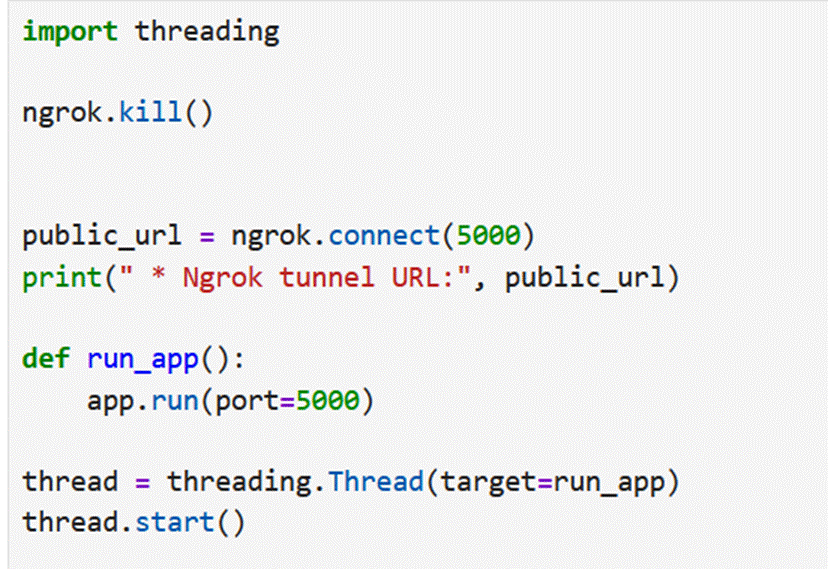
● 

● 

● 

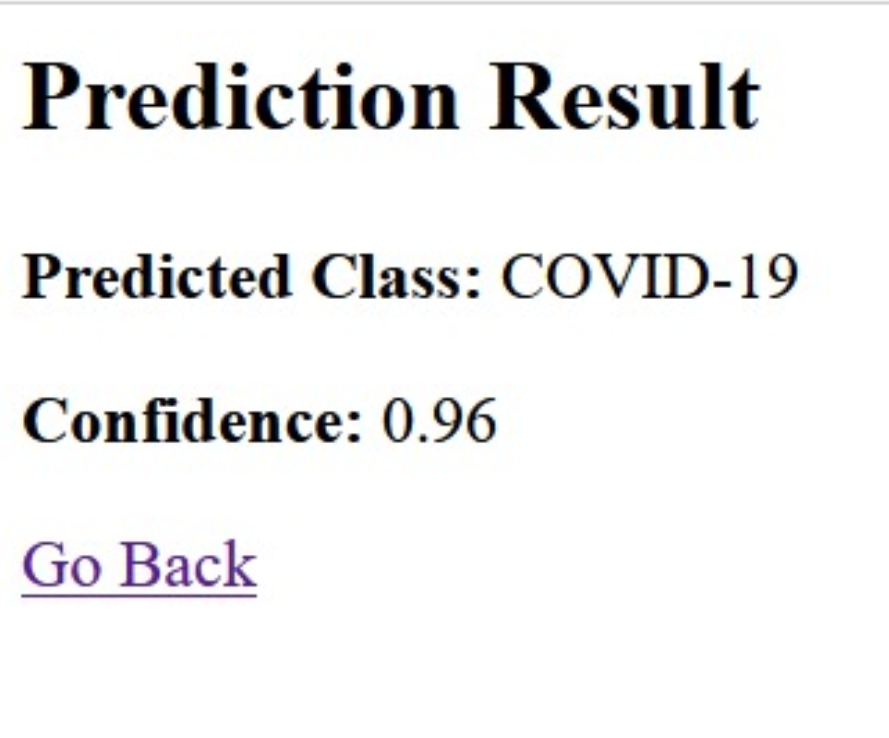
● 

● 

● 

● OUTPUT :

●   
  

### **7. Advantages & Disadvantages**

**Advantages:**  Fast, reliable, and automated diagnosis.  
 Transfer learning reduces training cost.  
 Deployable on web and edge devices.

**Disadvantages:**  Limited image resolution (64×64) may miss fine medical details.  
 Model may not generalize to unseen hospital datasets.

### **8. Conclusion**

The project successfully demonstrates a deep-learning-based X-ray classification system using **VGG16 transfer learning**. The model achieves >91 % validation accuracy and can classify X-rays into COVID-19, Normal, and Bacterial categories through an easy-to-use Flask web interface.

### **9. Future Scope**

● Upgrade to **ResNet50 or EfficientNet** for higher accuracy.

● Implement **Grad-CAM** for explainable AI in medical diagnosis.

● Extend to multi-class lung diseases and CT-scan data.

● Deploy on **Jetson Nano** or mobile edge devices.

### **10. Appendix**

#### **10.1 Source Code**

Refer to GitHub repository for preprocessing, model training, and Flask application.

#### **10.2 GitHub & Project Demo Links**

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

**End of Report**