Final Project Report Template

Team ID	SWTID1720440447
Project Name	CovidVision: Advanced COVID-19 Detection from Lung X-rays with Deep Learning

X Introduction

Project overviews

This project , "CovidVision: Advanced COVID-19 Detection from Lung X-rays with Deep Learning" utilizes deep learning algorithms to analyze lung X-ray images for signs of Covid-19 infection. By leveraging vast datasets and image recognition technology, this project aims to provide accurate and rapid diagnosis, aiding in early detection and containment of the virus.

The project also aims to address scenarios like integrating the AI system to expedite diagnosis, such that now the system swiftly analyzes X-rays, aiding medical staff in triage and treatment decisions, effectively managing patient influx during peak periods; automating analysis of lung X-rays enabling identification of cases, facilitating prompt isolation and treatment, thus assisting in Covid-19 screening and containing the spread; monitoring Covid-19 trends across regions by analyzing X-ray data from various healthcare facilities, identifying hotspots and allocating resources strategically, aiding in targeted interventions and control measures to curb transmission.

Objectives

- To obtain a dataset of lung X-ray images.
- Preprocessing the images.
- Create training and testing data to training and evaluation.1
- Apply Transfer learning algorithms on the dataset.
- Understand how deep neural networks detect the disease.
- Know how to find the accuracy of the model and find the accuracy of the model.
- Build a web applications using the Flask framework.

Project Initialization and Planning Phase

Define Problem Statement

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A radiologist in a community hospital.	Diagnose COVID-19 from lung X-rays swiftly and accurately.	The current diagnostic tools often produce inconsistent results, with frequent false positives and negatives.	These tools lack sophisticated analytical features and cannot always distinguish between COVID-19 and other lung conditions.	Anxious and dissatisfied with the diagnostic process, leading to potential delays in treatment and an increased burden of re-assessments.
PS-2	A hospital administrator tasked with streamlining clinical operations.	Implement a reliable COVID-19 diagnostic tool that seamlessly fits into our existing workflow and is user-friendly for our medical staff.	Many advanced diagnostic tools are complex, require extensive training, and are difficult to integrate into our current systems.	These tools often come with complicated interfaces and high computational demands.	Stressed about the practicality and cost-effectiveness of these tools, and concerned about the potential disruption to our workflow.
PS-3	A public health official focused on improving healthcare quality and access.	Ensure that COVID-19 diagnostic tools are effective for all populations and healthcare settings.	Existing tools frequently do not perform well across diverse groups, leading to disparities in healthcare outcomes.	These tools are often developed with non-diverse data and are not tailored to varied demographic and clinical scenarios.	Worried about the equity and inclusiveness of healthcare solutions, and driven to find technologies that deliver accurate results for all communities.

Project Proposal (Proposed Solution)

Project Proposal

Early Detection of COVID-19: The main challenge is to identify COVID-19 cases as early as possible. Early diagnosis allows for timely intervention, isolation, and appropriate medical care.
Differentiating COVID-19 from Other Lung Diseases: It is crucial to distinguish COVID-19 from other lung diseases (such as pneumonia or tuberculosis) based on chest X-rays. Accurate differentiation ensures proper treatment pathways.
Reducing False Negatives and False Positives: It is essential to avoid false negatives (missing actual COVID-19 cases) and false positives (misclassifying non-COVID-19 cases) for effective patient management.

Project Overview	
Objective	• Utilize deep learning to analyze lung X-ray images for COVID-19. • Obtain and preprocess a dataset, create training and testing data, apply transfer learning, determine model accuracy, and develop a web application using Flask.
Scope	Develop models to distinguish between normal and COVID-19 lung X-rays, aiding early diagnosis and patient management.
Problem Statement	
Description	• **Early Detection**: Identify COVID-19 cases early for timely intervention and care. • **Differentiating from Other Lung Diseases**: Distinguish COVID-19 from other lung diseases for accurate treatment.
Impact	• **Improved Outcomes**: Early detection reduces severity and mortality. • **Resource Optimization**: Accurate diagnosis allocates healthcare resources efficiently. • **Epidemiological Insights**: Accurate identification aids understanding of disease spread. • **Reduced Transmission**: Early detection prevents further community spread. • **Global Crisis Mitigation**: Addresses the ongoing COVID-19 health crisis.
Proposed Solution	
Approach	• Choose a pre-trained CNN model (e.g., ResNet, VGG, InceptionV3, Xception).

	• Adapt the model for COVID-19 detection by replacing final layers with task-specific layers.
Key Features	Utilize state-of-the-art pre-trained CNNs for high performance in image classification.

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	Python 3 Google Compute Engine backend (Google Colab)
Memory	RAM specifications	12.67GB
Storage	Disk space for data, models, and logs	107.72GB
Software		
Frameworks	Python frameworks	Flask
Libraries	Additional libraries	tensorflow
Development Environment	IDE, version control	Google colab, Git
Data		
Data	Source, size, format	Kaggle dataset, 10,000 images

Initial Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)

Sprint-1	Dataset Acquisition	USN-1	As a data scientist, I want to obtain a dataset of lung X-ray images for analysis.	2	High	Sahana Sree K, Shanmugha Priyan S K, S Ashwin	08/07/24	10/07/24
Sprint-2	Data Preprocessing	USN-2	As a machine learning engineer, I want to preprocess the images (resize, normalize pixel values, etc.).	1	High	Pramodh AVG, Sahana Sree K	11/07/24	14/07/24
Sprint-3	Model Selection	USN-3	As a machine learning engineer, I want to select a pre-trained deep learning model (e.g., VGG, ResNet) and modify it for classification.	2	Low	Shanmugha Priyan S K, S Ashwin	11/07/24	11/07/24
Sprint-4		USN-4	As a machine learning engineer, I want to train and evaluate the model on the preprocessed dataset.	2	Medium	Sahana Sree K, Shanmugha Priyan S K, S Ashwin	11/07/24	14/07/24
Sprint-5	Model Evaluation	USN-5	As a data analyst, I want to generate confusion matrices, classification reports, and visualize model attention maps to understand key features and identify	3	High	Sahana Sree K, Shanmugha Priyan S K, S Ashwin, Pramodh AVG	11/07/24	15/07/24

			areas for further improvement.					
Sprint-6	Web App Development	USN -6	As a web developer, I want to design the web app user interface and implement the backend using Flask to handle image uploads.	2	High	Sahana Sree K, Pramodh AVG	15/07/24	17/07/24
Sprint-7	Model Integration and Testing	USN-7	As a developer, I want to integrate the trained deep learning model into the web app, test it end-to-end, and fix any issues that come up.	1	High	Sahana Sree K, Shanmugha Priyan S K, S Ashwin	17/07/24	19/07/24

Data Collection and Preprocessing Phase

Data Collection Plan

Section	**Description**

Project Overview	The goal is to develop an advanced, precise, and real-time COVID-19 detection tool utilizing lung X-rays through transfer learning. This tool is intended to aid radiologists and healthcare professionals in swiftly and effectively diagnosing COVID-19, thereby enhancing patient outcomes and bolstering public health efforts..
Data Collection Plan	Data obtained from Kaggle is downloaded to Google Drive and accessed via Google Colab using the opendatasets Python library. Two directories are created: one for images labeled as 'Covid' and another for images labeled as 'Normal'.
Raw Data Sources Identified	The lung X-ray dataset is sourced from Kaggle.

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
Covid -19 Detection from Lung X-rays	The dataset contains data of chest X-ray images for Covid-19 positive cases along with Normal and Viral Pneumonia images.	https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database	Image	816 MB	Public

Data Quality Report

Data Quality Report Template Data Source	Data Quality Issue	Severity	Resolution Plan
Dataset (Normal and Covid) - Kaggle	The instances of Normal and Covid datasets were not the same.	Moderate	**Align Instances**: Ensure that each instance in the Covid dataset has a corresponding instance in the Normal dataset. **Modification of Instances**: Modify the Normal dataset to include only those instances present in the Covid dataset.

Data Preprocessing

Section	Description
Data Overview	The dataset contains data of chest X-ray images for Covid-19 positive cases along with Normal and Viral Pneumonia images
Resizing	Resize images to a target size of (256, 256)
Normalization	rescale=1/255: This parameter is used to normalize pixel values of images. In many image datasets, pixel values range from 0 to 255.
Data Augmentation	Applied augmentation techniques such as zooming range = 0.2 , shearing range = 0.2.

Data Preprocessing Code Screenshots	

Loading Data	`from google.colab import drive`
	`import zipfile`
	`import os`
	
	`# Mount Google Drive`
	`drive.mount('/content/drive')`
	
	`# Define the path to the zip file`
	`zip_file_path = '/content/drive/My Drive/archive (3).zip'`
	
	`Mounted at /content/drive`
	
	`%cd /content/drive/MyDrive`
	`/content/drive/MyDrive`
	
	Use /tmp to create temporary folder
	
	`[ ] !unzip '/content/drive/MyDrive/archive (3).zip' -d '/content/drive/MyDrive/dataset'`
	
	`Streaming output truncated to the last 5000 lines.`
	`inflating: /content/drive/MyDrive/dataset/COVID-19_Radiography_Dataset/Normal/masks/Normal-7921.png`
	`inflating: /content/drive/MyDrive/dataset/COVID-19_Radiography_Dataset/Normal/masks/Normal-7922.png`

Resizing	`IMAGE_SIZE = (256, 256)` `IMAGE_SHAPE = IMAGE_SIZE + (3,)`

Normalization	`from tensorflow.keras.preprocessing.image import ImageDataGenerator`
	`img_height, img_width= IMAGE_SIZE`
	`batch_size=16`
	`train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2,validation_split=0.3)`

Data Augmentation	`from tensorflow.keras.preprocessing.image import ImageDataGenerator`
	`img_height, img_width= IMAGE_SIZE`
	`batch_size=16`
	`train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2,validation_split=0.3)`

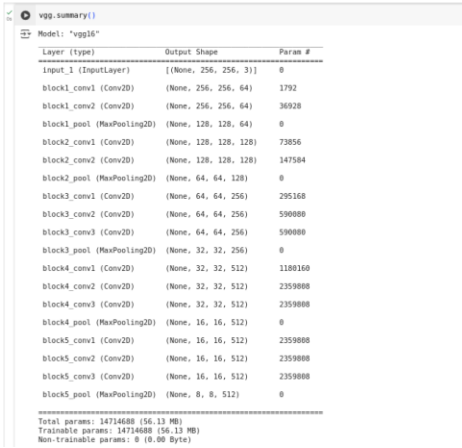
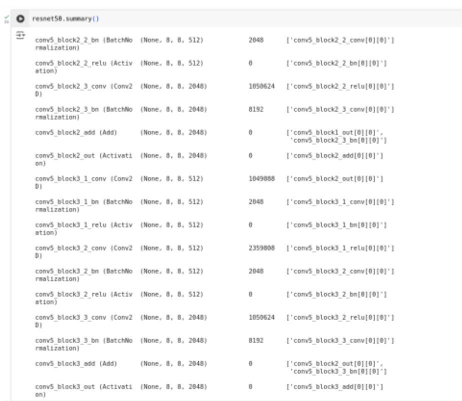
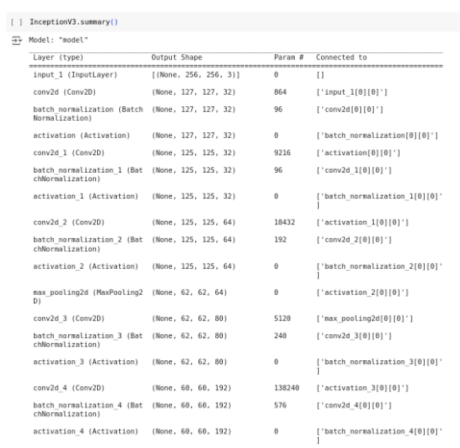
Model Development Phase

Model Selection Report

Model	**Description**

VGG16	VGG16 is a deep convolutional neural network model designed by the Visual Graphics Group at Oxford. It consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. The model is known for its simplicity and depth, making it effective for image classification tasks. Its architecture includes small 3x3 convolution filters and 2x2 max-pooling layers, which aid in extracting hierarchical features.
RESNET50	ResNet50 is a deep residual network that introduces residual learning to alleviate the vanishing gradient problem in very deep networks. It features 50 layers with residual blocks, each comprising a shortcut connection that bypasses one or more layers. This architecture allows gradients to flow through the network more effectively, facilitating the training of very deep networks with improved accuracy and reduced risk of overfitting. ResNet50 is ideal for complex image recognition tasks.
XCEPTION	Xception, short for Extreme Inception, extends the Inception model by applying depthwise separable convolutions. This approach enhances computational efficiency and reduces the number of parameters while maintaining high performance. The architecture uses depthwise separable convolutions that apply a single filter per input channel, resulting in state-of-the-art performance on several benchmarks.
INCEPTION	InceptionV3 is an advanced version of the Inception model, known for its novel architecture that includes multiple filter sizes in parallel at each convolutional layer. This design captures a wide range of spatial features with fewer parameters by using factorization and auxiliary classifiers to improve training speed and accuracy. InceptionV3 is highly effective for image recognition, object detection, and tasks requiring fine-grained feature extraction.

Initial Model Training Code, Model Validation and Evaluation Report

Model	Summary	Training and Validation Performance Metrics
VGG16		
ResNet50		
InceptionV3		

```
vgg.summary()

Model: "vgg16"
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]     0
 block1_conv1 (Conv2D)       (None, 256, 256, 64)      1792
 block1_conv2 (Conv2D)       (None, 256, 256, 64)      36928
 block1_pool (MaxPooling2D)  (None, 128, 128, 64)      0
 block2_conv1 (Conv2D)       (None, 128, 128, 128)     73856
 block2_conv2 (Conv2D)       (None, 128, 128, 128)     147584
 block2_pool (MaxPooling2D)  (None, 64, 64, 128)       0
 block3_conv1 (Conv2D)       (None, 64, 64, 256)       295168
 block3_conv2 (Conv2D)       (None, 64, 64, 256)       590080
 block3_conv3 (Conv2D)       (None, 64, 64, 256)       590080
 block3_pool (MaxPooling2D)  (None, 32, 32, 256)       0
 block4_conv1 (Conv2D)       (None, 32, 32, 512)       1180160
 block4_conv2 (Conv2D)       (None, 32, 32, 512)       2359808
 block4_conv3 (Conv2D)       (None, 32, 32, 512)       2359808
 block4_pool (MaxPooling2D)  (None, 16, 16, 512)       0
 block5_conv1 (Conv2D)       (None, 16, 16, 512)       2359808
 block5_conv2 (Conv2D)       (None, 16, 16, 512)       2359808
 block5_conv3 (Conv2D)       (None, 16, 16, 512)       2359808
 block5_pool (MaxPooling2D)  (None, 8, 8, 512)         0
=================================================================
Total params: 14714688 (56.13 MB)
Trainable params: 14714688 (56.13 MB)
Non-trainable params: 0 (0.00 Byte)
```

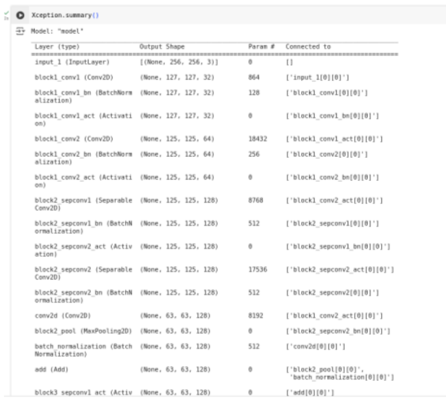
```
vgg16.fit(train_generator,validation_data=validation_generator,epochs=5)

Epoch 1/5
317/317 [==============================] - 1400s 4s/step - loss: 0.6234 - accuracy: 0.7345 - val_loss: 0.3295 - val_accuracy: 0.9546
Epoch 2/5
317/317 [==============================] - 138s 435ms/step - loss: 0.4876 - accuracy: 0.7921 - val_loss: 0.2908 - val_accuracy: 0.9442
Epoch 3/5
317/317 [==============================] - 137s 433ms/step - loss: 0.4310 - accuracy: 0.8182 - val_loss: 0.2754 - val_accuracy: 0.9347
Epoch 4/5
317/317 [==============================] - 137s 432ms/step - loss: 0.3895 - accuracy: 0.8370 - val_loss: 0.2637 - val_accuracy: 0.9546
Epoch 5/5
317/317 [==============================] - 136s 430ms/step - loss: 0.3529 - accuracy: 0.8515 - val_loss: 0.2543 - val_accuracy: 0.9549
<keras.src.callbacks.History at 0x7f701c2ef9a0>
<keras.src.callbacks.History at 0x7b9d8bf51870>
```

```
resnet50.summary()

 conv5_block2_2_bn (BatchNo  (None, 8, 8, 512)    2048     ['conv5_block2_2_conv[0][0]']
 rmalization)
 conv5_block2_2_relu (Activ  (None, 8, 8, 512)    0        ['conv5_block2_2_bn[0][0]']
 ation)
 conv5_block2_3_conv (Conv2  (None, 8, 8, 2048)   1050624  ['conv5_block2_2_relu[0][0]']
 D)
 conv5_block2_3_bn (BatchNo  (None, 8, 8, 2048)   8192     ['conv5_block2_3_conv[0][0]']
 rmalization)
 conv5_block2_add (Add)      (None, 8, 8, 2048)   0        ['conv5_block1_out[0][0]',
                                                            'conv5_block2_3_bn[0][0]']
 conv5_block2_out (Activati  (None, 8, 8, 2048)   0        ['conv5_block2_add[0][0]']
 on)
 conv5_block3_1_conv (Conv2  (None, 8, 8, 512)    1049088  ['conv5_block2_out[0][0]']
 D)
 conv5_block3_1_bn (BatchNo  (None, 8, 8, 512)    2048     ['conv5_block3_1_conv[0][0]']
 rmalization)
 conv5_block3_1_relu (Activ  (None, 8, 8, 512)    0        ['conv5_block3_1_bn[0][0]']
 ation)
 conv5_block3_2_conv (Conv2  (None, 8, 8, 512)    2359808  ['conv5_block3_1_relu[0][0]']
 D)
 conv5_block3_2_bn (BatchNo  (None, 8, 8, 512)    2048     ['conv5_block3_2_conv[0][0]']
 rmalization)
 conv5_block3_2_relu (Activ  (None, 8, 8, 512)    0        ['conv5_block3_2_bn[0][0]']
 ation)
 conv5_block3_3_conv (Conv2  (None, 8, 8, 2048)   1050624  ['conv5_block3_2_relu[0][0]']
 D)
 conv5_block3_3_bn (BatchNo  (None, 8, 8, 2048)   8192     ['conv5_block3_3_conv[0][0]']
 rmalization)
 conv5_block3_add (Add)      (None, 8, 8, 2048)   0        ['conv5_block2_out[0][0]',
                                                            'conv5_block3_3_bn[0][0]']
 conv5_block3_out (Activati  (None, 8, 8, 2048)   0        ['conv5_block3_add[0][0]']
 on)
```

```
resnet50.fit(train_generator,validation_data=validation_generator,epochs=5)

Epoch 1/5
317/317 [==============================] - 5727s 18s/step - loss: 0.8494 - accuracy: 0.6487 - val_loss: 0.4821 - val_accuracy: 0.7947
Epoch 2/5
317/317 [==============================] - 151s 476ms/step - loss: 0.6039 - accuracy: 0.7291 - val_loss: 0.4372 - val_accuracy: 0.8178
Epoch 3/5
317/317 [==============================] - 150s 475ms/step - loss: 0.7934 - accuracy: 0.7038 - val_loss: 0.9830 - val_accuracy: 0.5729
Epoch 4/5
317/317 [==============================] - 149s 471ms/step - loss: 0.7350 - accuracy: 0.7239 - val_loss: 0.6063 - val_accuracy: 0.7477
Epoch 5/5
317/317 [==============================] - 149s 470ms/step - loss: 0.6616 - accuracy: 0.7330 - val_loss: 0.4179 - val_accuracy: 0.8335
<keras.src.callbacks.History at 0x7f701c2ef9a0>
```

```
InceptionV3.summary()

Model: "model"
 Layer (type)                Output Shape           Param #  Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]  0        []
 conv2d (Conv2D)             (None, 127, 127, 32)   864      ['input_1[0][0]']
 batch_normalization (Batch  (None, 127, 127, 32)   96       ['conv2d[0][0]']
 Normalization)
 activation (Activation)     (None, 127, 127, 32)   0        ['batch_normalization[0][0]']
 conv2d_1 (Conv2D)           (None, 125, 125, 32)   9216     ['activation[0][0]']
 batch_normalization_1 (Bat  (None, 125, 125, 32)   96       ['conv2d_1[0][0]']
 chNormalization)
 activation_1 (Activation)   (None, 125, 125, 32)   0        ['batch_normalization_1[0][0]'
                                                              ]
 conv2d_2 (Conv2D)           (None, 125, 125, 64)   18432    ['activation_1[0][0]']
 batch_normalization_2 (Bat  (None, 125, 125, 64)   192      ['conv2d_2[0][0]']
 chNormalization)
 activation_2 (Activation)   (None, 125, 125, 64)   0        ['batch_normalization_2[0][0]'
                                                              ]
 max_pooling2d (MaxPooling2  (None, 62, 62, 64)     0        ['activation_2[0][0]']
 D)
 conv2d_3 (Conv2D)           (None, 62, 62, 80)     5120     ['max_pooling2d[0][0]']
 batch_normalization_3 (Bat  (None, 62, 62, 80)     240      ['conv2d_3[0][0]']
 chNormalization)
 activation_3 (Activation)   (None, 62, 62, 80)     0        ['batch_normalization_3[0][0]'
                                                              ]
 conv2d_4 (Conv2D)           (None, 60, 60, 192)    138240   ['activation_3[0][0]']
 batch_normalization_4 (Bat  (None, 60, 60, 192)    576      ['conv2d_4[0][0]']
 chNormalization)
 activation_4 (Activation)   (None, 60, 60, 192)    0        ['batch_normalization_4[0][0]'
                                                              ]
```

```
InceptionV3.fit(train_generator,validation_data=validation_generator,epochs=5)

Epoch 1/5
317/317 [==============================] - 4239s 13s/step - loss: 0.5572 - accuracy: 0.8221 - val_loss: 0.3898 - val_accuracy: 0.9045
Epoch 2/5
317/317 [==============================] - 3903s 12s/step - loss: 0.3359 - accuracy: 0.8841 - val_loss: 38.5079 - val_accuracy: 0.4451
Epoch 3/5
317/317 [==============================] - 3900s 12s/step - loss: 0.2376 - accuracy: 0.9139 - val_loss: 0.4145 - val_accuracy: 0.8459
Epoch 4/5
317/317 [==============================] - 3840s 12s/step - loss: 0.1801 - accuracy: 0.9327 - val_loss: 5.6839 - val_accuracy: 0.8252
Epoch 5/5
317/317 [==============================] - 3843s 12s/step - loss: 0.7318 - accuracy: 0.7561 - val_loss: 11.8876 - val_accuracy: 0.5798
<keras.src.callbacks.History at 0x7b9d8bf51870>
```

Xception		

Final Model Selection Justification

The model VGG 16 provides the best accuracy of 95.4%, hence it is the selected model.

Advantages and Disadvantages

Advantages

Chest X-ray (CXR) imaging has several advantages over other imaging and detection techniques, like the ability to perform them easily using portable X-ray machines providing faster, and accurate COVID-19 diagnosis. Utilizing pretrained CNNs enables an automatic learning process that identifies important features from raw input data through convolutional layers, reducing the need for manual feature engineering. Using pre-trained models like InceptionV3 saves significant time and computational resources compared to training a CNN from scratch, enabling quick deployment.

Disadvantages

Training a Convolutional Neural Network on large datasets can require significant computational resources, including powerful GPUs and large amounts of memory, which can be expensive and time consuming.

CXR imaging exposes patients to radiation although in low doses, repeated exposures can increase the cumulative risk of radiation-induced conditions. CXRs may not detect early-stage or subtle abnormalities. More traditional methods like RT-PCR is highly sensitive and specific for detecting the genetic material of pathogens, making it ideal for diagnosing infections like COVID-19, influenza, and other viral diseases, allowing detection at earlier stages, due to its ability to amplify small amounts of viral RNA.

Conclusion

In this project, we aimed to develop a machine learning model capable of predicting COVID-19 infection from chest X-ray (CXR) images. By leveraging the power of pre-trained TensorFlow models, we sought to achieve high accuracy and robust performance in this critical diagnostic task. The pre-trained models used in this study included InceptionV3, VGG-16, ResNet-50, and Xception.

The results of our experiments demonstrated that pre-trained models are highly effective for the task of COVID-19 detection from CXR images. Each model's performance was evaluated based on its accuracy and overall robustness. Our findings revealed notable differences in performance among the models, with VGG-16 providing the highest accuracy at 95.4%.

Overall, the pre-trained models significantly reduced the training time and resources needed, while achieving high accuracy even with limited labeled data. The transfer learning approach enabled the models to generalize well to the task of COVID-19 detection, highlighting the effectiveness of utilizing pre-trained networks in medical imaging applications.

In conclusion, our comparative analysis underscores the potential of using pre-trained CNN models for rapid and accurate COVID-19 diagnosis from CXR images. This project demonstrates the feasibility and promise of leveraging state-of-the-art deep learning techniques to aid in the timely diagnosis and management of COVID-19, ultimately contributing to better patient outcomes and healthcare resource optimization.

Future Scope

Future work could focus on further fine-tuning these models with larger and more diverse datasets, exploring the integration of additional clinical data and ensemble techniques, mutlimodal integration and bias mitigation to enhance predictive performance, and evaluating the models in real-world clinical settings.

Fine-Tuning the existing models and experimenting with with different architectures (such as attention-based networks) and hyperparameters hep to achieve even better performance. Ensemble techniques can enhance robustness and generalization.

Multi-modal approaches help in providing a holistic view of the disease.Collaborating with healthcare professionals to validate the model’s performance in clinical settings and conducting prospective studies help to assess its real-world impact.

Regularly auditing the model for biases related to demographics (race, gender, age) by implement techniques to mitigate bias and ensure fairness and safeguarding data during deployment are further future scopes for this project.

Appendix
Source Code

```
from google.colab import drive
drive.mount('/content/drive/')

%cd /content/drive/MyDrive


!pip install opendatasets
import opendatasets as od
import pandas as pd
od.download("https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database")

!unzip '/content/drive/MyDrive/covid19-radiography-database/covid19-radiography-database.zip' -d
'/content/drive/MyDrive/covid19-radiography-database'

!ls "/content/drive/MyDrive/covid19-radiography-database/COVID-19_Radiography_Dataset"

!mkdir /content/drive/MyDrive/Xray_train_data

!cp -R "/content/drive/MyDrive/covid19-radiography-database/COVID-19_Radiography_Dataset/COVID/images"
"/content/drive/MyDrive/Xray_train_data/COVID"

!ls -1 "/content/drive/MyDrive/Xray_train_data/COVID"| wc -l\

!mkdir "/content/drive/MyDrive/Xray_train_data/NORMAL"

import pandas as pd
PATH_TO_METADATA = "/content/drive/MyDrive/covid19-radiography-database/COVID-
19_Radiography_Dataset/Normal.metadata.xlsx"
df = pd.read_excel(PATH_TO_METADATA)
df.head()

import os
import shutil
cnt = 0
for (i, row) in df.iterrows():
  if (cnt < 3616):
    filename = row["FILE NAME"].lower().capitalize() + "." + row["FORMAT"].lower()
    image_path = os.path.join("/content/drive/MyDrive/covid19-radiography-database/COVID-
19_Radiography_Dataset/Normal/images", filename)
    image_copy_path = os.path.join("/content/drive/MyDrive/Xray_train_data/NORMAL", filename)
    shutil.copy2(image_path, image_copy_path)
    cnt += 1

print(cnt)

!ls "/content/drive/MyDrive/Xray_train_data"

IMAGE_SIZE = (256, 256)
IMAGE_SHAPE = IMAGE_SIZE + (3,)

train_data_dir= "/content/drive/MyDrive/Xray_train_data"

test_data_dir= "/content/drive/MyDrive/Xray_train_data"
```

The rest of the code is available in the GitHub respository.

GitHub Repository

https://github.com/sahanasree23/Advanced-COVID-19-Detection-from-Lung-X-rays-with-Deep-Learning

Project Demo Link

https://drive.google.com/file/d/1VddQWsfuv3MHd2wmcnfNAX6k5u4y5nya/view?usp=sharing