**Mini Project: COVID-19 Detection from Chest X-rays using CNN**

**Objective**

A healthcare startup aims to accelerate and improve COVID-19 diagnosis using deep learning technologies. The current testing procedures are time-consuming and rely heavily on manual radiological examination, which limits scalability in pandemic situations.

As a data scientist, your task is to build a Convolutional Neural Network (CNN) model that can automatically detect COVID-19 from chest X-ray images. This solution can help hospitals:

* Reduce diagnosis time.
* Minimize burden on radiologists.
* Scale testing across regions with limited radiological expertise.

This project allows you to apply deep learning concepts, understand how CNNs process image data, and build real-world applications that can assist the medical community.

**Dataset**

You will use the dataset provided in the following Kaggle project:[**https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset**](https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset)

**The dataset contains:**

* COVID-19 chest X-ray images
* Normal (no disease) chest X-ray images
* Viral Pneumonia chest X-ray images

**Data Dictionary**

Each image is labelled into one of the following classes:

* COVID-19 – Confirmed COVID infection.
* Viral Pneumonia – Non-COVID lung infection.
* Normal – No visible lung abnormality.

**Tasks**

**1. Data Loading and Exploration**

* Import necessary libraries (os, cv2, matplotlib, tensorflow, keras etc.)
* Import Covid19 dataset from Kaggle into colab using Kaggle API.
* Load images from different folders and label them
* Resize images to a fixed shape (e.g 128x128 or 224x224)
* Display a few sample images from each class
* Print dataset size per class

**2. Data Preprocessing**

* Normalize pixel values (scale from 0–255 to 0–1)
* Encode class labels using one-hot encoding or LabelEncoder
* Split the data into training, validation and test sets (e.g 80-20%)

**3. Exploratory Data Analysis (EDA)**

* Visualize class distribution using bar plots
* Plot sample images with their class names
* Observe patterns in data

**4. CNN Model Building**

Create and train multiple CNN architectures:

* Model 1: Basic CNN
  + Conv2D -> MaxPooling -> Flatten -> Dense
* Model 2: Transfer Learning
  + Use pre-trained models like VGG16, ResNet50 etc
  + Fine-tune last few layers on the COVID dataset
* Model 3: Transfer Learning + Data Augmentation
  + Do the data Augmentation using ImageDataGenerator.
  + The use the pretrained models to get the prediction.

**5. Model Evaluation**

Evaluate each model using:

* Accuracy on test set
* Confusion matrix
* Classification report (Precision, Recall, F1-score)
* ROC-AUC score (binary or per class if multi-class)

Also check for:

* Overfitting (training vs validation loss/accuracy plots)
* Number of trainable parameters

**6. Handle Class Imbalance**

* Analyze class imbalance using visual plots
* Apply techniques like:
  + Oversampling using data augmentation
  + Class weights in model.fit()

**7. Model Tuning**

Use techniques like:

* Early stopping
* Hyperparameter tuning (number of filters, dropout rate, optimizer etc.)

Use Keras Tuner for deeper optimization.

**8. Model Comparison Table**

| Model | Train Acc | Test Acc | F1 Score | Overfitting (Y/N) |
| --- | --- | --- | --- | --- |
| CNN Basic |  |  |  |  |
| Deep CNN |  |  |  |  |
| ResNet50 |  |  |  |  |
| VGG16 |  |  |  |  |
| Best Model |  |  |  |  |

**9. Build a web app using Streamlit that:**

* Takes a chest X-ray image as input
* Displays prediction