**Deep Learning for Chest X-Ray Image Classification: Detecting Pneumonia**

**Sameen Yaser**

**Introduction**

The code will train a deep learning model on the Chest X-Ray Images (Normal vs. Pneumonia) dataset. It will preprocess the images, split the data into training and testing sets, and use a convolutional neural network (CNN) to learn patterns and features for classifying chest X-ray images as normal or pneumonia. The model's performance will be evaluated on the testing set to assess its accuracy in detecting pneumonia.

**Importing the Required Libraries**

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import os

import zipfile

from google.colab import files

**Upload the zip file containing test and train images**

uploaded = files.upload()

**Set the path to the uploaded zip file**

zip\_filename = list(uploaded.keys())[0]

extracted\_folder = 'extracted\_dataset\_folder/New folder'

**Extract the zip file**

with zipfile.ZipFile(zip\_filename, 'r') as zf:

zf.extractall(extracted\_folder)

**Check if the extracted train and test directories exist**

train\_dir = os.path.join(extracted\_folder, 'train')

test\_dir = os.path.join(extracted\_folder, 'test')

if not os.path.exists(train\_dir) or not os.path.exists(test\_dir):

print("Directory structure in 'extracted\_dataset\_folder':")

for root, dirs, files in os.walk(extracted\_folder):

level = root.replace(extracted\_folder, '').count(os.sep)

indent = ' ' \* 4 \* (level)

print('{}{}/'.format(indent, os.path.basename(root)))

subindent = ' ' \* 4 \* (level + 1)

for f in files:

print('{}{}'.format(subindent, f))

raise FileNotFoundError("Train or test directory not found. Make sure the extraction process was successful.")

**Define the image dimensions and batch size**

image\_size = (224, 224)

batch\_size = 32

**Data augmentation and preprocessing**

train\_datagen = ImageDataGenerator(

rescale=1.0 / 255,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True

)

test\_datagen = ImageDataGenerator(rescale=1.0 / 255)

**Load and prepare the training data**

train\_data = train\_datagen.flow\_from\_directory(

train\_dir,

target\_size=image\_size,

batch\_size=batch\_size,

class\_mode='binary'

)

**Load and prepare the testing data**

test\_data = test\_datagen.flow\_from\_directory(

test\_dir,

target\_size=image\_size,

batch\_size=batch\_size,

class\_mode='binary'

)

**Define the CNN model architecture**

model = tf.keras.models.Sequential([

tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(224, 224, 3)),

tf.keras.layers.MaxPooling2D(2, 2),

tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),

tf.keras.layers.MaxPooling2D(2, 2),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.Dense(1, activation='sigmoid')

])

C**ompile the model**

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

**Train the model**

history = model.fit(

train\_data,

epochs=10,

validation\_data=test\_data

)

**Evaluate the model on the test set**

test\_loss, test\_acc = model.evaluate(test\_data)

print("Test Accuracy:", test\_acc)

**CONCLUSION**

The model achieved a test accuracy of 83.14%, demonstrating its effectiveness in automating pneumonia diagnosis from chest X-ray images.s