Pediatric Pneumonia Detection using Deep Learning

This project presents an end-to-end deep learning pipeline for detecting pediatric pneumonia from chest X-ray images. The goal is to build a binary image classification model that distinguishes between NORMAL and PNEUMONIA cases using transfer learning.

The workflow demonstrates how medical imaging data can be prepared, processed, and used to train a convolutional neural network for automated disease classification. This implementation is designed for educational and applied machine learning purposes, showcasing practical steps involved in building an AI-based diagnostic support system.

Methodology and Workflow

The project follows a structured machine learning pipeline:

• Dataset Acquisition
Chest X-ray images are automatically downloaded and organized for supervised learning.

• Data Preparation
The dataset is divided into training, validation, and testing subsets. Images are resized, normalized, and augmented to improve model generalization.

• Model Development
A pretrained convolutional neural network is used as a feature extractor, and custom classification layers are added for pneumonia detection.

• Model Evaluation
The trained model is evaluated on unseen test data to assess its performance and generalization capability.

This workflow ensures systematic model development and fair performance assessment.

📂 Step 1 — Dataset Download using KaggleHub

In this step, we download the Pediatric Chest X-ray Pneumonia dataset directly from Kaggle using KaggleHub. This method avoids manual upload and automatically stores the dataset in the Colab environment for further processing.

```
!pip install kagglehub

import kagglehub

path = kagglehub.dataset_download(
    "yusufmurtaza01/pediatric-chest-xray-pneumonia-balanced-dataset"
)

print("Dataset Path:", path)
```

```
Requirement already satisfied: kagglehub in /usr/local/lib/python3.12/dist-packages (0.3.13)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from kagglehub) (26.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.12/dist-packages (from kagglehub) (6.0.3)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kagglehub) (2.32.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from kagglehub) (4.67.3)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->kagglehub) (3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->kagglehub) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->kagglehub) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->kagglehub) (2026.1.
Warning: Looks like you're using an outdated `kagglehub` version (installed: 0.3.13), please consider upgrading to the latest ve
Downloading from https://www.kaggle.com/api/v1/datasets/download/yusufmurtaza01/pediatric-chest-xray-pneumonia-balanced-dataset?
100%|██████████| 1.66G/1.66G [00:11<00:00, 154MB/s]Extracting files...

Dataset Path: /root/.cache/kagglehub/datasets/yusufmurtaza01/pediatric-chest-xray-pneumonia-balanced-dataset/versions/1
```

📂 Step 2 — Understanding Dataset Structure

Here, we explore the dataset directory to verify its structure. We check whether it contains Train, Validation, and Test folders along with their respective class subfolders (NORMAL and PNEUMONIA).

```
import os

os.listdir(path)
```

```
os.listdir(path)
```

```
['NORMAL', 'PNEUMONIA']
```

📂 Step 2.1 — Exploring Complete Dataset Directory

In this step, we explore the full directory tree of the downloaded dataset to understand its internal folder hierarchy. This helps us identify whether the dataset is already split into training, validation, and testing sets or if we need to perform the split manually.

```
for root, dirs, files in os.walk(path):
    print("Root:", root)
    print("Folders:", dirs)
    print("Files count:", len(files))
    print("-"*40)
```

```
Root: /root/.cache/kagglehub/datasets/yusufmurtaza01/pediatric-chest-xray-pneumonia-balanced-dataset/versions/1
Folders: ['NORMAL', 'PNEUMONIA']
Files count: 0
----------------------------------------
Root: /root/.cache/kagglehub/datasets/yusufmurtaza01/pediatric-chest-xray-pneumonia-balanced-dataset/versions/1/NORMAL
Folders: []
Files count: 4265
----------------------------------------
Root: /root/.cache/kagglehub/datasets/yusufmurtaza01/pediatric-chest-xray-pneumonia-balanced-dataset/versions/1/PNEUMONIA
Folders: []
Files count: 4265
----------------------------------------
```

📂 Step 3 — Dataset Splitting (Train / Validation / Test)

In this step, we split the dataset into three subsets: Training, Validation, and Testing.

Since the dataset is not pre-split, we manually divide the images into:

- 70% Training set
- 15% Validation set
- 15% Testing set

This ensures that the model is trained on one portion of the data and evaluated on unseen data for fair performance assessment.

```
import os
import shutil
import random
```

📂 Creating Directory Structure for Split Data

Here, we create a new directory structure to store the split datasets. Each subset (Train, Validation, Test) will contain two class folders: NORMAL and PNEUMONIA.

```
base_dir = "/content/chest_xray_split"

folders = [
    "train/NORMAL",
    "train/PNEUMONIA",
    "val/NORMAL",
    "val/PNEUMONIA",
    "test/NORMAL",
    "test/PNEUMONIA"
]

for folder in folders:
    os.makedirs(os.path.join(base_dir, folder), exist_ok=True)
```

🔀 Splitting Images into Train, Validation, and Test Sets

In this step, we randomly shuffle the images and divide them into training, validation, and testing subsets according to the defined ratios.

```
def split_data(source, train, val, test, split_ratio=(0.7, 0.15, 0.15)):

    images = os.listdir(source)
    random.shuffle(images)

    total = len(images)
    train_end = int(split_ratio[0] * total)
    val_end = int((split_ratio[0] + split_ratio[1]) * total)

    train_imgs = images[:train_end]
    val_imgs = images[train_end:val_end]
    test_imgs = images[val_end:]

    for img in train_imgs:
        shutil.copy(os.path.join(source, img), train)

    for img in val_imgs:
        shutil.copy(os.path.join(source, img), val)

    for img in test_imgs:
        shutil.copy(os.path.join(source, img), test)
```

🚀 Applying Dataset Splitting

Here, we apply the splitting function to both classes (Normal and Pneumonia) to generate the final dataset structure for model training and evaluation.

```
original_path = path

split_data(
    original_path + "/NORMAL",
    base_dir + "/train/NORMAL",
    base_dir + "/val/NORMAL",
    base_dir + "/test/NORMAL"
)

split_data(
    original_path + "/PNEUMONIA",
    base_dir + "/train/PNEUMONIA",
    base_dir + "/val/PNEUMONIA",
    base_dir + "/test/PNEUMONIA"
)

print("Dataset splitting completed ✅")
```

```
Dataset splitting completed ✅
```

📊 Verifying Dataset Split

In this step, we verify the number of images in each subset to ensure the dataset has been split correctly.

```
for root, dirs, files in os.walk(base_dir):
    print(root, "→", len(files))
```

```
/content/chest_xray_split → 0
/content/chest_xray_split/train → 0
/content/chest_xray_split/train/NORMAL → 2985
/content/chest_xray_split/train/PNEUMONIA → 2985
/content/chest_xray_split/val → 0
/content/chest_xray_split/val/NORMAL → 640
/content/chest_xray_split/val/PNEUMONIA → 640
/content/chest_xray_split/test → 0
/content/chest_xray_split/test/NORMAL → 640
/content/chest_xray_split/test/PNEUMONIA → 640
```

🖼 Step 4 — Visualizing Sample Chest X-ray Images

In this step, we display sample images from both classes (Normal and Pneumonia) to visually understand the dataset.

This helps in identifying visible differences between healthy and infected lungs and ensures that images are loaded correctly before training the model.

```
import matplotlib.pyplot as plt
import cv2
import os
import random
```

🔍 Displaying Random Samples from Each Class

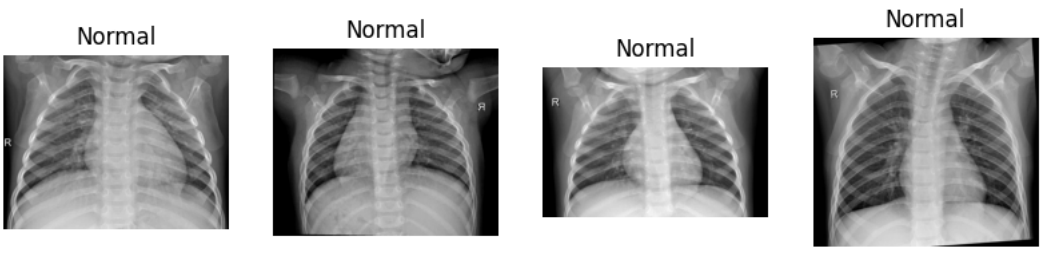
Here, we randomly select images from both categories and display them for visual comparison.

```
def show_samples(class_path, title):

    images = os.listdir(class_path)
    sample_images = random.sample(images, 4)

    plt.figure(figsize=(10,5))

    for i, img_name in enumerate(sample_images):
        img_path = os.path.join(class_path, img_name)
        img = cv2.imread(img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        plt.subplot(1,4,i+1)
        plt.imshow(img)
        plt.title(title)
        plt.axis("off")

    plt.show()
```

🫁 Normal Chest X-ray Samples

These images represent healthy lungs without pneumonia infection.

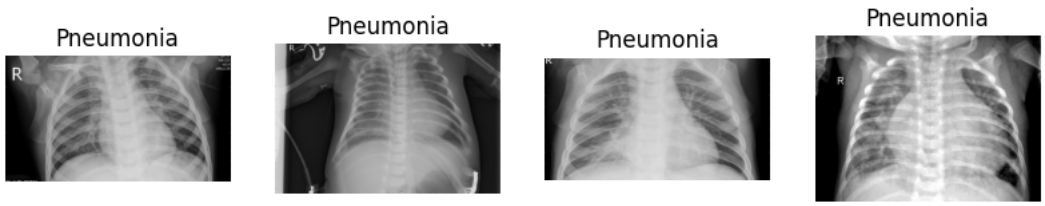
```
show_samples(
    base_dir + "/train/NORMAL",
    "Normal"
)
```



🦠 Pneumonia Chest X-ray Samples

These images represent lungs affected by pneumonia infection, showing visible opacities or abnormalities.

```
show_samples(
    base_dir + "/train/PNEUMONIA",
    "Pneumonia"
)
```



🧹 Step 5 — Image Preprocessing and Data Augmentation

In this step, we prepare the dataset for model training by applying preprocessing techniques.

The preprocessing includes:

- Rescaling pixel values (normalization)
- Resizing images to a fixed shape
- Applying augmentation techniques to the training set

Data augmentation helps improve model generalization and reduces overfitting by creating slightly modified variations of existing images.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

🔄 Training Data Generator with Augmentation

Here, we apply augmentation techniques such as rotation, zoom, and horizontal flipping to artificially expand the training dataset and improve model robustness.

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    zoom_range=0.1,
    horizontal_flip=True
)
```

📏 Validation and Test Data Preprocessing

For validation and testing datasets, we only apply rescaling.

No augmentation is used here to ensure fair evaluation on real, unmodified images.

```
val_test_datagen = ImageDataGenerator(
    rescale=1./255
)
```

📥 Step 6 — Loading Images Using Data Generators

In this step, we load images from the split dataset directories using Keras data generators.

The images are resized to 224×224 pixels and converted into batches for efficient model training.

```
img_size = (224, 224)
batch_size = 32
```

```
train_data = train_datagen.flow_from_directory(
    base_dir + "/train",
    target_size=img_size,
    batch_size=batch_size,
    class_mode='binary'
)
```

```
Found 5970 images belonging to 2 classes.
```

```
val_data = val_test_datagen.flow_from_directory(
    base_dir + "/val",
    target_size=img_size,
    batch_size=batch_size,
    class_mode='binary'
)
```

```
Found 1280 images belonging to 2 classes.
```

```
test_data = val_test_datagen.flow_from_directory(
    base_dir + "/test",
    target_size=img_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False
)
```

```
Found 1280 images belonging to 2 classes.
```

🔎 Checking Class Indices

This step verifies how class labels are encoded by the generator.

Typically:

- Normal → 0
- Pneumonia → 1

```
train_data.class_indices
```

```
{'NORMAL': 0, 'PNEUMONIA': 1}
```

🧠 Step 7 — Building Transfer Learning Model (DenseNet121)

In this step, we build a deep learning model using the DenseNet121 architecture with pretrained ImageNet weights.

Transfer learning allows us to leverage previously learned features from large datasets, improving performance on medical imaging tasks such as pneumonia detection.

```
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras import layers, models
```

📦 Loading Pretrained DenseNet121

Here, we load DenseNet121 without its top classification layer and use ImageNet pretrained weights for feature extraction.

```
base_model = DenseNet121(
    weights='imagenet',
    include_top=False,
    input_shape=(224,224,3)
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/densenet/densenet121_weights_tf_dim_ordering_
29084464/29084464 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

❄️ Freezing Base Model Layers

We freeze the convolutional base to prevent pretrained weights from updating during initial training. This helps retain learned low-level image features.

```
base_model.trainable = False
```

🏗️ Adding Custom Classification Layers

We add new fully connected layers on top of the pretrained base model to adapt it for binary pneumonia classification.

```
x = layers.GlobalAveragePooling2D()(base_model.output)
x = layers.Dense(128, activation='relu')(x)
x = layers.Dropout(0.5)(x)
output = layers.Dense(1, activation='sigmoid')(x)

model = models.Model(
    inputs=base_model.input,
    outputs=output
)
```

⚙️ Compiling the Model

Here, we configure the model for training using:

- Adam optimizer
- Binary crossentropy loss
- Accuracy as evaluation metric

```
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)
```

```
model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	-
zero_padding2d (ZeroPadding2D)	(None, 230, 230, 3)	0	input_layer[0][0]
conv1_conv (Conv2D)	(None, 112, 112, 64)	9,408	zero_padding2d[0…
conv1_bn (BatchNormalizatio…	(None, 112, 112, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None, 112, 112, 64)	0	conv1_bn[0][0]
zero_padding2d_1 (ZeroPadding2D)	(None, 114, 114, 64)	0	conv1_relu[0][0]
pool1 (MaxPooling2D)	(None, 56, 56, 64)	0	zero_padding2d_1…
conv2_block1_0_bn (BatchNormalizatio…	(None, 56, 56, 64)	256	pool1[0][0]
conv2_block1_0_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1_0_b…
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 128)	8,192	conv2_block1_0_r…
conv2_block1_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block1_1_c…
conv2_block1_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block1_1_b…
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block1_1_r…
conv2_block1_concat (Concatenate)	(None, 56, 56, 96)	0	pool1[0][0], conv2_block1_2_c…
conv2_block2_0_bn (BatchNormalizatio…	(None, 56, 56, 96)	384	conv2_block1_con…
conv2_block2_0_relu (Activation)	(None, 56, 56, 96)	0	conv2_block2_0_b…
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 128)	12,288	conv2_block2_0_r…
conv2_block2_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block2_1_c…
conv2_block2_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block2_1_b…
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block2_1_r…
conv2_block2_concat (Concatenate)	(None, 56, 56, 128)	0	conv2_block1_con… conv2_block2_2_c…
conv2_block3_0_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block2_con…
conv2_block3_0_relu (Activation)	(None, 56, 56, 128)	0	conv2_block3_0_b…
conv2_block3_1_conv (Conv2D)	(None, 56, 56, 128)	16,384	conv2_block3_0_r…
conv2_block3_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block3_1_c…
conv2_block3_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block3_1_b…
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block3_1_r…
conv2_block3_concat	(None, 56, 56,	0	conv2_block2_con…

(Concatenate)	160)		conv2_block3_2_c…
conv2_block4_0_bn (BatchNormalizatio…	(None, 56, 56, 160)	640	conv2_block3_con…
conv2_block4_0_relu (Activation)	(None, 56, 56, 160)	0	conv2_block4_0_b…
conv2_block4_1_conv (Conv2D)	(None, 56, 56, 128)	20,480	conv2_block4_0_r…
conv2_block4_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block4_1_c…
conv2_block4_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block4_1_b…
conv2_block4_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block4_1_r…
conv2_block4_concat (Concatenate)	(None, 56, 56, 192)	0	conv2_block3_con… conv2_block4_2_c…
conv2_block5_0_bn (BatchNormalizatio…	(None, 56, 56, 192)	768	conv2_block4_con…
conv2_block5_0_relu (Activation)	(None, 56, 56, 192)	0	conv2_block5_0_b…
conv2_block5_1_conv (Conv2D)	(None, 56, 56, 128)	24,576	conv2_block5_0_r…
conv2_block5_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block5_1_c…
conv2_block5_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block5_1_b…
conv2_block5_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block5_1_r…
conv2_block5_concat (Concatenate)	(None, 56, 56, 224)	0	conv2_block4_con… conv2_block5_2_c…
conv2_block6_0_bn (BatchNormalizatio…	(None, 56, 56, 224)	896	conv2_block5_con…
conv2_block6_0_relu (Activation)	(None, 56, 56, 224)	0	conv2_block6_0_b…
conv2_block6_1_conv (Conv2D)	(None, 56, 56, 128)	28,672	conv2_block6_0_r…
conv2_block6_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block6_1_c…
conv2_block6_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block6_1_b…
conv2_block6_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block6_1_r…
conv2_block6_concat (Concatenate)	(None, 56, 56, 256)	0	conv2_block5_con… conv2_block6_2_c…
pool2_bn (BatchNormalizatio…	(None, 56, 56, 256)	1,024	conv2_block6_con…
pool2_relu (Activation)	(None, 56, 56, 256)	0	pool2_bn[0][0]
pool2_conv (Conv2D)	(None, 56, 56, 128)	32,768	pool2_relu[0][0]
pool2_pool (AveragePooling2D)	(None, 28, 28, 128)	0	pool2_conv[0][0]
conv3_block1_0_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	pool2_pool[0][0]
conv3_block1_0_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_0_b…
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128)	16,384	conv3_block1_0_r…
conv3_block1_1_bn	(None, 28, 28	512	conv3_block1_1_c

(BatchNormalizatio…	128)		
conv3_block1_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_1_b…
conv3_block1_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block1_1_r…
conv3_block1_concat (Concatenate)	(None, 28, 28, 160)	0	pool2_pool[0][0], conv3_block1_2_c…
conv3_block2_0_bn (BatchNormalizatio…	(None, 28, 28, 160)	640	conv3_block1_con…
conv3_block2_0_relu (Activation)	(None, 28, 28, 160)	0	conv3_block2_0_b…
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	20,480	conv3_block2_0_r…
conv3_block2_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block2_1_c…
conv3_block2_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_1_b…
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block2_1_r…
conv3_block2_concat (Concatenate)	(None, 28, 28, 192)	0	conv3_block1_con… conv3_block2_2_c…
conv3_block3_0_bn (BatchNormalizatio…	(None, 28, 28, 192)	768	conv3_block2_con…
conv3_block3_0_relu (Activation)	(None, 28, 28, 192)	0	conv3_block3_0_b…
conv3_block3_1_conv (Conv2D)	(None, 28, 28, 128)	24,576	conv3_block3_0_r…
conv3_block3_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block3_1_c…
conv3_block3_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block3_1_b…
conv3_block3_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block3_1_r…
conv3_block3_concat (Concatenate)	(None, 28, 28, 224)	0	conv3_block2_con… conv3_block3_2_c…
conv3_block4_0_bn (BatchNormalizatio…	(None, 28, 28, 224)	896	conv3_block3_con…
conv3_block4_0_relu (Activation)	(None, 28, 28, 224)	0	conv3_block4_0_b…
conv3_block4_1_conv (Conv2D)	(None, 28, 28, 128)	28,672	conv3_block4_0_r…
conv3_block4_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block4_1_c…
conv3_block4_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block4_1_b…
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block4_1_r…
conv3_block4_concat (Concatenate)	(None, 28, 28, 256)	0	conv3_block3_con… conv3_block4_2_c…
conv3_block5_0_bn (BatchNormalizatio…	(None, 28, 28, 256)	1,024	conv3_block4_con…
conv3_block5_0_relu (Activation)	(None, 28, 28, 256)	0	conv3_block5_0_b…
conv3_block5_1_conv (Conv2D)	(None, 28, 28, 128)	32,768	conv3_block5_0_r…
conv3_block5_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block5_1_c…

conv3_block5_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block5_1_b…
conv3_block5_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block5_1_r…
conv3_block5_concat (Concatenate)	(None, 28, 28, 288)	0	conv3_block4_con… conv3_block5_2_c…
conv3_block6_0_bn (BatchNormalizatio…	(None, 28, 28, 288)	1,152	conv3_block5_con…
conv3_block6_0_relu (Activation)	(None, 28, 28, 288)	0	conv3_block6_0_b…
conv3_block6_1_conv (Conv2D)	(None, 28, 28, 128)	36,864	conv3_block6_0_r…
conv3_block6_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block6_1_c…
conv3_block6_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block6_1_b…
conv3_block6_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block6_1_r…
conv3_block6_concat (Concatenate)	(None, 28, 28, 320)	0	conv3_block5_con… conv3_block6_2_c…
conv3_block7_0_bn (BatchNormalizatio…	(None, 28, 28, 320)	1,280	conv3_block6_con…
conv3_block7_0_relu (Activation)	(None, 28, 28, 320)	0	conv3_block7_0_b…
conv3_block7_1_conv (Conv2D)	(None, 28, 28, 128)	40,960	conv3_block7_0_r…
conv3_block7_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block7_1_c…
conv3_block7_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block7_1_b…
conv3_block7_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block7_1_r…
conv3_block7_concat (Concatenate)	(None, 28, 28, 352)	0	conv3_block6_con… conv3_block7_2_c…
conv3_block8_0_bn (BatchNormalizatio…	(None, 28, 28, 352)	1,408	conv3_block7_con…
conv3_block8_0_relu (Activation)	(None, 28, 28, 352)	0	conv3_block8_0_b…
conv3_block8_1_conv (Conv2D)	(None, 28, 28, 128)	45,056	conv3_block8_0_r…
conv3_block8_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block8_1_c…
conv3_block8_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block8_1_b…
conv3_block8_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block8_1_r…
conv3_block8_concat (Concatenate)	(None, 28, 28, 384)	0	conv3_block7_con… conv3_block8_2_c…
conv3_block9_0_bn (BatchNormalizatio…	(None, 28, 28, 384)	1,536	conv3_block8_con…
conv3_block9_0_relu (Activation)	(None, 28, 28, 384)	0	conv3_block9_0_b…
conv3_block9_1_conv (Conv2D)	(None, 28, 28, 128)	49,152	conv3_block9_0_r…
conv3_block9_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block9_1_c…
conv3_block9_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block9_1_b…

conv3_block9_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block9_1_r…
conv3_block9_concat (Concatenate)	(None, 28, 28, 416)	0	conv3_block8_con… conv3_block9_2_c…
conv3_block10_0_bn (BatchNormalizatio…	(None, 28, 28, 416)	1,664	conv3_block9_con…
conv3_block10_0_re… (Activation)	(None, 28, 28, 416)	0	conv3_block10_0_…
conv3_block10_1_co… (Conv2D)	(None, 28, 28, 128)	53,248	conv3_block10_0_…
conv3_block10_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block10_1_…
conv3_block10_1_re… (Activation)	(None, 28, 28, 128)	0	conv3_block10_1_…
conv3_block10_2_co… (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block10_1_…
conv3_block10_conc… (Concatenate)	(None, 28, 28, 448)	0	conv3_block9_con… conv3_block10_2_…
conv3_block11_0_bn (BatchNormalizatio…	(None, 28, 28, 448)	1,792	conv3_block10_co…
conv3_block11_0_re… (Activation)	(None, 28, 28, 448)	0	conv3_block11_0_…
conv3_block11_1_co… (Conv2D)	(None, 28, 28, 128)	57,344	conv3_block11_0_…
conv3_block11_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block11_1_…
conv3_block11_1_re… (Activation)	(None, 28, 28, 128)	0	conv3_block11_1_…
conv3_block11_2_co… (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block11_1_…
conv3_block11_conc… (Concatenate)	(None, 28, 28, 480)	0	conv3_block10_co… conv3_block11_2_…
conv3_block12_0_bn (BatchNormalizatio…	(None, 28, 28, 480)	1,920	conv3_block11_co…
conv3_block12_0_re… (Activation)	(None, 28, 28, 480)	0	conv3_block12_0_…
conv3_block12_1_co… (Conv2D)	(None, 28, 28, 128)	61,440	conv3_block12_0_…
conv3_block12_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block12_1_…
conv3_block12_1_re… (Activation)	(None, 28, 28, 128)	0	conv3_block12_1_…
conv3_block12_2_co… (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block12_1_…
conv3_block12_conc… (Concatenate)	(None, 28, 28, 512)	0	conv3_block11_co… conv3_block12_2_…
pool3_bn (BatchNormalizatio…	(None, 28, 28, 512)	2,048	conv3_block12_co…
pool3_relu (Activation)	(None, 28, 28, 512)	0	pool3_bn[0][0]
pool3_conv (Conv2D)	(None, 28, 28, 256)	131,072	pool3_relu[0][0]
pool3_pool (AveragePooling2D)	(None, 14, 14, 256)	0	pool3_conv[0][0]
conv4_block1_0_bn (BatchNormalizatio…	(None, 14, 14, 256)	1,024	pool3_pool[0][0]
conv4_block1_0_relu (Activation)	(None, 14, 14, 256)	0	conv4_block1_0_b…

conv4_block1_1_conv (Conv2D)	(None, 14, 14, 128)	32,768	conv4_block1_0_r…
conv4_block1_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block1_1_c…
conv4_block1_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block1_1_b…
conv4_block1_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block1_1_r…
conv4_block1_concat (Concatenate)	(None, 14, 14, 288)	0	pool3_pool[0][0], conv4_block1_2_c…
conv4_block2_0_bn (BatchNormalizatio…	(None, 14, 14, 288)	1,152	conv4_block1_con…
conv4_block2_0_relu (Activation)	(None, 14, 14, 288)	0	conv4_block2_0_b…
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 128)	36,864	conv4_block2_0_r…
conv4_block2_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block2_1_c…
conv4_block2_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block2_1_b…
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block2_1_r…
conv4_block2_concat (Concatenate)	(None, 14, 14, 320)	0	conv4_block1_con… conv4_block2_2_c…
conv4_block3_0_bn (BatchNormalizatio…	(None, 14, 14, 320)	1,280	conv4_block2_con…
conv4_block3_0_relu (Activation)	(None, 14, 14, 320)	0	conv4_block3_0_b…
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 128)	40,960	conv4_block3_0_r…
conv4_block3_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block3_1_c…
conv4_block3_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block3_1_b…
conv4_block3_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block3_1_r…
conv4_block3_concat (Concatenate)	(None, 14, 14, 352)	0	conv4_block2_con… conv4_block3_2_c…
conv4_block4_0_bn (BatchNormalizatio…	(None, 14, 14, 352)	1,408	conv4_block3_con…
conv4_block4_0_relu (Activation)	(None, 14, 14, 352)	0	conv4_block4_0_b…
conv4_block4_1_conv (Conv2D)	(None, 14, 14, 128)	45,056	conv4_block4_0_r…
conv4_block4_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block4_1_c…
conv4_block4_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block4_1_b…
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block4_1_r…
conv4_block4_concat (Concatenate)	(None, 14, 14, 384)	0	conv4_block3_con… conv4_block4_2_c…
conv4_block5_0_bn (BatchNormalizatio…	(None, 14, 14, 384)	1,536	conv4_block4_con…
conv4_block5_0_relu (Activation)	(None, 14, 14, 384)	0	conv4_block5_0_b…
conv4_block5_1_conv (Conv2D)	(None, 14, 14, 128)	49,152	conv4_block5_0_r…

conv4_block5_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block5_1_c…
conv4_block5_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block5_1_b…
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block5_1_r…
conv4_block5_concat (Concatenate)	(None, 14, 14, 416)	0	conv4_block4_con… conv4_block5_2_c…
conv4_block6_0_bn (BatchNormalizatio…	(None, 14, 14, 416)	1,664	conv4_block5_con…
conv4_block6_0_relu (Activation)	(None, 14, 14, 416)	0	conv4_block6_0_b…
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 128)	53,248	conv4_block6_0_r…
conv4_block6_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block6_1_c…
conv4_block6_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block6_1_b…
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block6_1_r…
conv4_block6_concat (Concatenate)	(None, 14, 14, 448)	0	conv4_block5_con… conv4_block6_2_c…
conv4_block7_0_bn (BatchNormalizatio…	(None, 14, 14, 448)	1,792	conv4_block6_con…
conv4_block7_0_relu (Activation)	(None, 14, 14, 448)	0	conv4_block7_0_b…
conv4_block7_1_conv (Conv2D)	(None, 14, 14, 128)	57,344	conv4_block7_0_r…
conv4_block7_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block7_1_c…
conv4_block7_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block7_1_b…
conv4_block7_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block7_1_r…
conv4_block7_concat (Concatenate)	(None, 14, 14, 480)	0	conv4_block6_con… conv4_block7_2_c…
conv4_block8_0_bn (BatchNormalizatio…	(None, 14, 14, 480)	1,920	conv4_block7_con…
conv4_block8_0_relu (Activation)	(None, 14, 14, 480)	0	conv4_block8_0_b…
conv4_block8_1_conv (Conv2D)	(None, 14, 14, 128)	61,440	conv4_block8_0_r…
conv4_block8_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block8_1_c…
conv4_block8_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block8_1_b…
conv4_block8_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block8_1_r…
conv4_block8_concat (Concatenate)	(None, 14, 14, 512)	0	conv4_block7_con… conv4_block8_2_c…
conv4_block9_0_bn (BatchNormalizatio…	(None, 14, 14, 512)	2,048	conv4_block8_con…
conv4_block9_0_relu (Activation)	(None, 14, 14, 512)	0	conv4_block9_0_b…
conv4_block9_1_conv (Conv2D)	(None, 14, 14, 128)	65,536	conv4_block9_0_r…
conv4_block9_1_bn	(None, 14, 14,	512	conv4_block9_1_c…

(BatchNormalizatio…	128)		
conv4_block9_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block9_1_b…
conv4_block9_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block9_1_r…
conv4_block9_concat (Concatenate)	(None, 14, 14, 544)	0	conv4_block8_con… conv4_block9_2_c…
conv4_block10_0_bn (BatchNormalizatio…	(None, 14, 14, 544)	2,176	conv4_block9_con…
conv4_block10_0_re… (Activation)	(None, 14, 14, 544)	0	conv4_block10_0_…
conv4_block10_1_co… (Conv2D)	(None, 14, 14, 128)	69,632	conv4_block10_0_…
conv4_block10_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block10_1_…
conv4_block10_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block10_1_…
conv4_block10_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block10_1_…
conv4_block10_conc… (Concatenate)	(None, 14, 14, 576)	0	conv4_block9_con… conv4_block10_2_…
conv4_block11_0_bn (BatchNormalizatio…	(None, 14, 14, 576)	2,304	conv4_block10_co…
conv4_block11_0_re… (Activation)	(None, 14, 14, 576)	0	conv4_block11_0_…
conv4_block11_1_co… (Conv2D)	(None, 14, 14, 128)	73,728	conv4_block11_0_…
conv4_block11_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block11_1_…
conv4_block11_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block11_1_…
conv4_block11_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block11_1_…
conv4_block11_conc… (Concatenate)	(None, 14, 14, 608)	0	conv4_block10_co… conv4_block11_2_…
conv4_block12_0_bn (BatchNormalizatio…	(None, 14, 14, 608)	2,432	conv4_block11_co…
conv4_block12_0_re… (Activation)	(None, 14, 14, 608)	0	conv4_block12_0_…
conv4_block12_1_co… (Conv2D)	(None, 14, 14, 128)	77,824	conv4_block12_0_…
conv4_block12_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block12_1_…
conv4_block12_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block12_1_…
conv4_block12_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block12_1_…
conv4_block12_conc… (Concatenate)	(None, 14, 14, 640)	0	conv4_block11_co… conv4_block12_2_…
conv4_block13_0_bn (BatchNormalizatio…	(None, 14, 14, 640)	2,560	conv4_block12_co…
conv4_block13_0_re… (Activation)	(None, 14, 14, 640)	0	conv4_block13_0_…
conv4_block13_1_co… (Conv2D)	(None, 14, 14, 128)	81,920	conv4_block13_0_…
conv4_block13_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block13_1_…
conv4_block13_1_re…	(None, 14, 14,	0	conv4_block13_1_…

(Activation)	128)		
conv4_block13_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block13_1_…
conv4_block13_conc… (Concatenate)	(None, 14, 14, 672)	0	conv4_block12_co… conv4_block13_2_…
conv4_block14_0_bn (BatchNormalizatio…	(None, 14, 14, 672)	2,688	conv4_block13_co…
conv4_block14_0_re… (Activation)	(None, 14, 14, 672)	0	conv4_block14_0_…
conv4_block14_1_co… (Conv2D)	(None, 14, 14, 128)	86,016	conv4_block14_0_…
conv4_block14_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block14_1_…
conv4_block14_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block14_1_…
conv4_block14_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block14_1_…
conv4_block14_conc… (Concatenate)	(None, 14, 14, 704)	0	conv4_block13_co… conv4_block14_2_…
conv4_block15_0_bn (BatchNormalizatio…	(None, 14, 14, 704)	2,816	conv4_block14_co…
conv4_block15_0_re… (Activation)	(None, 14, 14, 704)	0	conv4_block15_0_…
conv4_block15_1_co… (Conv2D)	(None, 14, 14, 128)	90,112	conv4_block15_0_…
conv4_block15_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block15_1_…
conv4_block15_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block15_1_…
conv4_block15_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block15_1_…
conv4_block15_conc… (Concatenate)	(None, 14, 14, 736)	0	conv4_block14_co… conv4_block15_2_…
conv4_block16_0_bn (BatchNormalizatio…	(None, 14, 14, 736)	2,944	conv4_block15_co…
conv4_block16_0_re… (Activation)	(None, 14, 14, 736)	0	conv4_block16_0_…
conv4_block16_1_co… (Conv2D)	(None, 14, 14, 128)	94,208	conv4_block16_0_…
conv4_block16_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block16_1_…
conv4_block16_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block16_1_…
conv4_block16_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block16_1_…
conv4_block16_conc… (Concatenate)	(None, 14, 14, 768)	0	conv4_block15_co… conv4_block16_2_…
conv4_block17_0_bn (BatchNormalizatio…	(None, 14, 14, 768)	3,072	conv4_block16_co…
conv4_block17_0_re… (Activation)	(None, 14, 14, 768)	0	conv4_block17_0_…
conv4_block17_1_co… (Conv2D)	(None, 14, 14, 128)	98,304	conv4_block17_0_…
conv4_block17_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block17_1_…
conv4_block17_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block17_1_…

conv4_block17_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block17_1_…
conv4_block17_conc… (Concatenate)	(None, 14, 14, 800)	0	conv4_block16_co… conv4_block17_2_…
conv4_block18_0_bn (BatchNormalizatio…	(None, 14, 14, 800)	3,200	conv4_block17_co…
conv4_block18_0_re… (Activation)	(None, 14, 14, 800)	0	conv4_block18_0_…
conv4_block18_1_co… (Conv2D)	(None, 14, 14, 128)	102,400	conv4_block18_0_…
conv4_block18_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block18_1_…
conv4_block18_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block18_1_…
conv4_block18_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block18_1_…
conv4_block18_conc… (Concatenate)	(None, 14, 14, 832)	0	conv4_block17_co… conv4_block18_2_…
conv4_block19_0_bn (BatchNormalizatio…	(None, 14, 14, 832)	3,328	conv4_block18_co…
conv4_block19_0_re… (Activation)	(None, 14, 14, 832)	0	conv4_block19_0_…
conv4_block19_1_co… (Conv2D)	(None, 14, 14, 128)	106,496	conv4_block19_0_…
conv4_block19_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block19_1_…
conv4_block19_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block19_1_…
conv4_block19_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block19_1_…
conv4_block19_conc… (Concatenate)	(None, 14, 14, 864)	0	conv4_block18_co… conv4_block19_2_…
conv4_block20_0_bn (BatchNormalizatio…	(None, 14, 14, 864)	3,456	conv4_block19_co…
conv4_block20_0_re… (Activation)	(None, 14, 14, 864)	0	conv4_block20_0_…
conv4_block20_1_co… (Conv2D)	(None, 14, 14, 128)	110,592	conv4_block20_0_…
conv4_block20_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block20_1_…
conv4_block20_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block20_1_…
conv4_block20_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block20_1_…
conv4_block20_conc… (Concatenate)	(None, 14, 14, 896)	0	conv4_block19_co… conv4_block20_2_…
conv4_block21_0_bn (BatchNormalizatio…	(None, 14, 14, 896)	3,584	conv4_block20_co…
conv4_block21_0_re… (Activation)	(None, 14, 14, 896)	0	conv4_block21_0_…
conv4_block21_1_co… (Conv2D)	(None, 14, 14, 128)	114,688	conv4_block21_0_…
conv4_block21_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block21_1_…
conv4_block21_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block21_1_…
conv4_block21_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block21_1_…

conv4_block21_conc… (Concatenate)	(None, 14, 14, 928)	0	conv4_block20_co… conv4_block21_2_…
conv4_block22_0_bn (BatchNormalizatio…	(None, 14, 14, 928)	3,712	conv4_block21_co…
conv4_block22_0_re… (Activation)	(None, 14, 14, 928)	0	conv4_block22_0_…
conv4_block22_1_co… (Conv2D)	(None, 14, 14, 128)	118,784	conv4_block22_0_…
conv4_block22_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block22_1_…
conv4_block22_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block22_1_…
conv4_block22_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block22_1_…
conv4_block22_conc… (Concatenate)	(None, 14, 14, 960)	0	conv4_block21_co… conv4_block22_2_…
conv4_block23_0_bn (BatchNormalizatio…	(None, 14, 14, 960)	3,840	conv4_block22_co…
conv4_block23_0_re… (Activation)	(None, 14, 14, 960)	0	conv4_block23_0_…
conv4_block23_1_co… (Conv2D)	(None, 14, 14, 128)	122,880	conv4_block23_0_…
conv4_block23_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block23_1_…
conv4_block23_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block23_1_…
conv4_block23_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block23_1_…
conv4_block23_conc… (Concatenate)	(None, 14, 14, 992)	0	conv4_block22_co… conv4_block23_2_…
conv4_block24_0_bn (BatchNormalizatio…	(None, 14, 14, 992)	3,968	conv4_block23_co…
conv4_block24_0_re… (Activation)	(None, 14, 14, 992)	0	conv4_block24_0_…
conv4_block24_1_co… (Conv2D)	(None, 14, 14, 128)	126,976	conv4_block24_0_…
conv4_block24_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block24_1_…
conv4_block24_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block24_1_…
conv4_block24_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block24_1_…
conv4_block24_conc… (Concatenate)	(None, 14, 14, 1024)	0	conv4_block23_co… conv4_block24_2_…
pool4_bn (BatchNormalizatio…	(None, 14, 14, 1024)	4,096	conv4_block24_co…
pool4_relu (Activation)	(None, 14, 14, 1024)	0	pool4_bn[0][0]
pool4_conv (Conv2D)	(None, 14, 14, 512)	524,288	pool4_relu[0][0]
pool4_pool (AveragePooling2D)	(None, 7, 7, 512)	0	pool4_conv[0][0]
conv5_block1_0_bn (BatchNormalizatio…	(None, 7, 7, 512)	2,048	pool4_pool[0][0]
conv5_block1_0_relu (Activation)	(None, 7, 7, 512)	0	conv5_block1_0_b…
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 128)	65,536	conv5_block1_0_r…

conv5_block1_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block1_1_c…
conv5_block1_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block1_1_b…
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block1_1_r…
conv5_block1_concat (Concatenate)	(None, 7, 7, 544)	0	pool4_pool[0][0], conv5_block1_2_c…
conv5_block2_0_bn (BatchNormalizatio…	(None, 7, 7, 544)	2,176	conv5_block1_con…
conv5_block2_0_relu (Activation)	(None, 7, 7, 544)	0	conv5_block2_0_b…
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 128)	69,632	conv5_block2_0_r…
conv5_block2_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block2_1_c…
conv5_block2_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block2_1_b…
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block2_1_r…
conv5_block2_concat (Concatenate)	(None, 7, 7, 576)	0	conv5_block1_con… conv5_block2_2_c…
conv5_block3_0_bn (BatchNormalizatio…	(None, 7, 7, 576)	2,304	conv5_block2_con…
conv5_block3_0_relu (Activation)	(None, 7, 7, 576)	0	conv5_block3_0_b…
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 128)	73,728	conv5_block3_0_r…
conv5_block3_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block3_1_c…
conv5_block3_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block3_1_b…
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block3_1_r…
conv5_block3_concat (Concatenate)	(None, 7, 7, 608)	0	conv5_block2_con… conv5_block3_2_c…
conv5_block4_0_bn (BatchNormalizatio…	(None, 7, 7, 608)	2,432	conv5_block3_con…
conv5_block4_0_relu (Activation)	(None, 7, 7, 608)	0	conv5_block4_0_b…
conv5_block4_1_conv (Conv2D)	(None, 7, 7, 128)	77,824	conv5_block4_0_r…
conv5_block4_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block4_1_c…
conv5_block4_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block4_1_b…
conv5_block4_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block4_1_r…
conv5_block4_concat (Concatenate)	(None, 7, 7, 640)	0	conv5_block3_con… conv5_block4_2_c…
conv5_block5_0_bn (BatchNormalizatio…	(None, 7, 7, 640)	2,560	conv5_block4_con…
conv5_block5_0_relu (Activation)	(None, 7, 7, 640)	0	conv5_block5_0_b…
conv5_block5_1_conv (Conv2D)	(None, 7, 7, 128)	81,920	conv5_block5_0_r…
conv5_block5_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block5_1_c…

conv5_block5_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block5_1_b…
conv5_block5_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block5_1_r…
conv5_block5_concat (Concatenate)	(None, 7, 7, 672)	0	conv5_block4_con… conv5_block5_2_c…
conv5_block6_0_bn (BatchNormalizatio…	(None, 7, 7, 672)	2,688	conv5_block5_con…
conv5_block6_0_relu (Activation)	(None, 7, 7, 672)	0	conv5_block6_0_b…
conv5_block6_1_conv (Conv2D)	(None, 7, 7, 128)	86,016	conv5_block6_0_r…
conv5_block6_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block6_1_c…
conv5_block6_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block6_1_b…
conv5_block6_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block6_1_r…
conv5_block6_concat (Concatenate)	(None, 7, 7, 704)	0	conv5_block5_con… conv5_block6_2_c…
conv5_block7_0_bn (BatchNormalizatio…	(None, 7, 7, 704)	2,816	conv5_block6_con…
conv5_block7_0_relu (Activation)	(None, 7, 7, 704)	0	conv5_block7_0_b…
conv5_block7_1_conv (Conv2D)	(None, 7, 7, 128)	90,112	conv5_block7_0_r…
conv5_block7_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block7_1_c…
conv5_block7_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block7_1_b…
conv5_block7_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block7_1_r…
conv5_block7_concat (Concatenate)	(None, 7, 7, 736)	0	conv5_block6_con… conv5_block7_2_c…
conv5_block8_0_bn (BatchNormalizatio…	(None, 7, 7, 736)	2,944	conv5_block7_con…
conv5_block8_0_relu (Activation)	(None, 7, 7, 736)	0	conv5_block8_0_b…
conv5_block8_1_conv (Conv2D)	(None, 7, 7, 128)	94,208	conv5_block8_0_r…
conv5_block8_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block8_1_c…
conv5_block8_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block8_1_b…
conv5_block8_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block8_1_r…
conv5_block8_concat (Concatenate)	(None, 7, 7, 768)	0	conv5_block7_con… conv5_block8_2_c…
conv5_block9_0_bn (BatchNormalizatio…	(None, 7, 7, 768)	3,072	conv5_block8_con…
conv5_block9_0_relu (Activation)	(None, 7, 7, 768)	0	conv5_block9_0_b…
conv5_block9_1_conv (Conv2D)	(None, 7, 7, 128)	98,304	conv5_block9_0_r…
conv5_block9_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block9_1_c…
conv5_block9_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block9_1_b…

(Activation)			
conv5_block9_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block9_1_r…
conv5_block9_concat (Concatenate)	(None, 7, 7, 800)	0	conv5_block8_con… conv5_block9_2_c…
conv5_block10_0_bn (BatchNormalizatio…	(None, 7, 7, 800)	3,200	conv5_block9_con…
conv5_block10_0_re… (Activation)	(None, 7, 7, 800)	0	conv5_block10_0_…
conv5_block10_1_co… (Conv2D)	(None, 7, 7, 128)	102,400	conv5_block10_0_…
conv5_block10_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block10_1_…
conv5_block10_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block10_1_…
conv5_block10_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block10_1_…
conv5_block10_conc… (Concatenate)	(None, 7, 7, 832)	0	conv5_block9_con… conv5_block10_2_…
conv5_block11_0_bn (BatchNormalizatio…	(None, 7, 7, 832)	3,328	conv5_block10_co…
conv5_block11_0_re… (Activation)	(None, 7, 7, 832)	0	conv5_block11_0_…
conv5_block11_1_co… (Conv2D)	(None, 7, 7, 128)	106,496	conv5_block11_0_…
conv5_block11_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block11_1_…
conv5_block11_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block11_1_…
conv5_block11_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block11_1_…
conv5_block11_conc… (Concatenate)	(None, 7, 7, 864)	0	conv5_block10_co… conv5_block11_2_…
conv5_block12_0_bn (BatchNormalizatio…	(None, 7, 7, 864)	3,456	conv5_block11_co…
conv5_block12_0_re… (Activation)	(None, 7, 7, 864)	0	conv5_block12_0_…
conv5_block12_1_co… (Conv2D)	(None, 7, 7, 128)	110,592	conv5_block12_0_…
conv5_block12_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block12_1_…
conv5_block12_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block12_1_…
conv5_block12_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block12_1_…
conv5_block12_conc… (Concatenate)	(None, 7, 7, 896)	0	conv5_block11_co… conv5_block12_2_…
conv5_block13_0_bn (BatchNormalizatio…	(None, 7, 7, 896)	3,584	conv5_block12_co…
conv5_block13_0_re… (Activation)	(None, 7, 7, 896)	0	conv5_block13_0_…
conv5_block13_1_co… (Conv2D)	(None, 7, 7, 128)	114,688	conv5_block13_0_…
conv5_block13_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block13_1_…
conv5_block13_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block13_1_…
conv5_block13_2_co…	(None, 7, 7, 32)	36,864	conv5_block13_1_…

```
| (Conv2D)                        |                        |                |                        |
| conv5_block13_conc… (Concatenate) | (None, 7, 7, 928)    |              0 | conv5_block12_co… conv5_block13_2_… |
| conv5_block14_0_bn (BatchNormal… | (None, 7, 7, 928)     |          3,712 | conv5_block13_co… |
```

Step 8 — Training the Transfer Learning Model

In this step, we train the DenseNet121-based transfer learning model using the prepared training and validation datasets.
The model learns to classify chest X-ray images as Normal or Pneumonia by optimizing its weights over multiple epochs.

```
epochs = 15
```

■ Using Callbacks for Efficient Training

Callbacks help improve training efficiency:

- **EarlyStopping** stops training when validation performance stops improving.
- **ModelCheckpoint** saves the best-performing model weights.

This prevents overfitting and preserves the optimal model.

```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
early_stop = EarlyStopping(
    monitor='val_loss',
    patience=3,
    restore_best_weights=True
)

checkpoint = ModelCheckpoint(
    "best_model.h5",
    monitor='val_loss',
    save_best_only=True
)
```

🚀 Training the Model

Here, we train the model using the training dataset and validate it on the validation dataset to monitor performance.

```
history = model.fit(
    train_data,
    validation_data=val_data,
    epochs=epochs,
    callbacks=[early_stop, checkpoint]
)
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset
  self._warn_if_super_not_called()
Epoch 1/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 0s 709ms/step - accuracy: 0.7934 - loss: 0.4605WARNING:absl:You are saving your model as an HDF5 fi
187/187 ━━━━━━━━━━━━━━━━━━━━ 187s 850ms/step - accuracy: 0.7939 - loss: 0.4596 - val_accuracy: 0.9492 - val_loss: 0.1404
Epoch 2/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 0s 646ms/step - accuracy: 0.9421 - loss: 0.1671WARNING:absl:You are saving your model as an HDF5 fi
187/187 ━━━━━━━━━━━━━━━━━━━━ 134s 719ms/step - accuracy: 0.9421 - loss: 0.1671 - val_accuracy: 0.9516 - val_loss: 0.1256
Epoch 3/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 0s 636ms/step - accuracy: 0.9426 - loss: 0.1555WARNING:absl:You are saving your model as an HDF5 fi
187/187 ━━━━━━━━━━━━━━━━━━━━ 133s 709ms/step - accuracy: 0.9426 - loss: 0.1555 - val_accuracy: 0.9547 - val_loss: 0.1138
Epoch 4/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 130s 698ms/step - accuracy: 0.9391 - loss: 0.1579 - val_accuracy: 0.9523 - val_loss: 0.1152
Epoch 5/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 0s 669ms/step - accuracy: 0.9539 - loss: 0.1284WARNING:absl:You are saving your model as an HDF5 fi
187/187 ━━━━━━━━━━━━━━━━━━━━ 140s 745ms/step - accuracy: 0.9539 - loss: 0.1284 - val_accuracy: 0.9664 - val_loss: 0.1009
Epoch 6/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 133s 711ms/step - accuracy: 0.9604 - loss: 0.1156 - val_accuracy: 0.9594 - val_loss: 0.1044
Epoch 7/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 141s 708ms/step - accuracy: 0.9541 - loss: 0.1261 - val_accuracy: 0.9508 - val_loss: 0.1162
Epoch 8/15
187/187 ━━━━━━━━━━━━━━━━━━━━ 133s 711ms/step - accuracy: 0.9559 - loss: 0.1145 - val_accuracy: 0.9375 - val_loss: 0.1553
```

🧪 Step 10 — Evaluating Model on Test Dataset

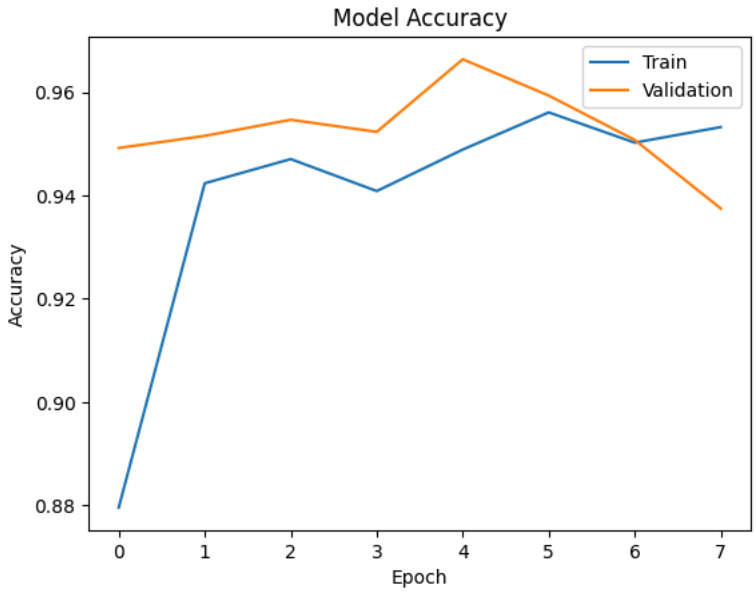
In this step, we evaluate the trained model on the unseen test dataset to measure its real-world performance.

This ensures that the model generalizes well beyond the training data.

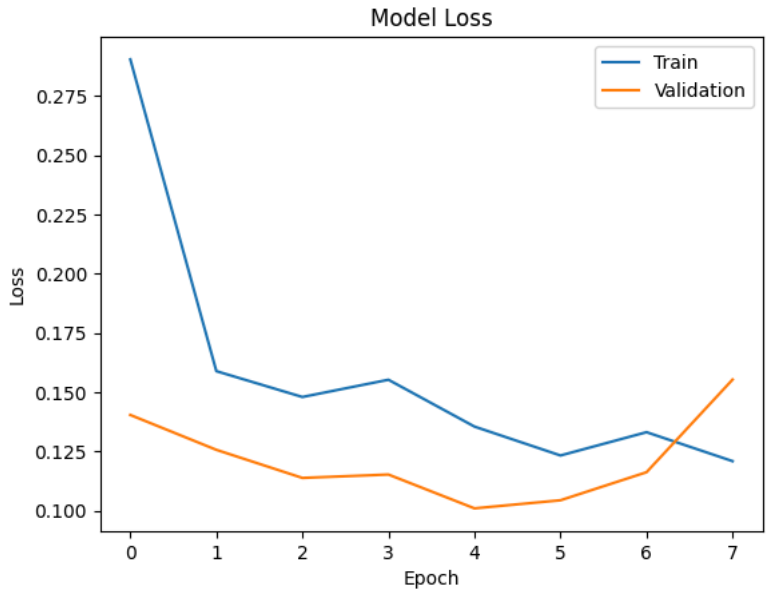
```
test_loss, test_accuracy = model.evaluate(test_data)

print("Test Accuracy:", test_accuracy)
print("Test Loss:", test_loss)
```

```
40/40 ━━━━━━━━━━━━━━━━━━━━ 13s 324ms/step - accuracy: 0.9787 - loss: 0.0729
Test Accuracy: 0.9624999761581421
Test Loss: 0.10545533895492554
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title("Model Accuracy")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend(["Train", "Validation"])
plt.show()
```




```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title("Model Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend(["Train", "Validation"])
plt.show()
```



📊 Step 11 — Generating Classification Report

Here, we compute precision, recall, and F1-score to better understand model performance, especially for medical diagnosis tasks.

```
import numpy as np
from sklearn.metrics import classification_report
```

```
predictions = model.predict(test_data)
pred_classes = (predictions > 0.5).astype("int32")

true_classes = test_data.classes
class_labels = list(test_data.class_indices.keys())

print(classification_report(true_classes, pred_classes, target_names=class_labels))
```

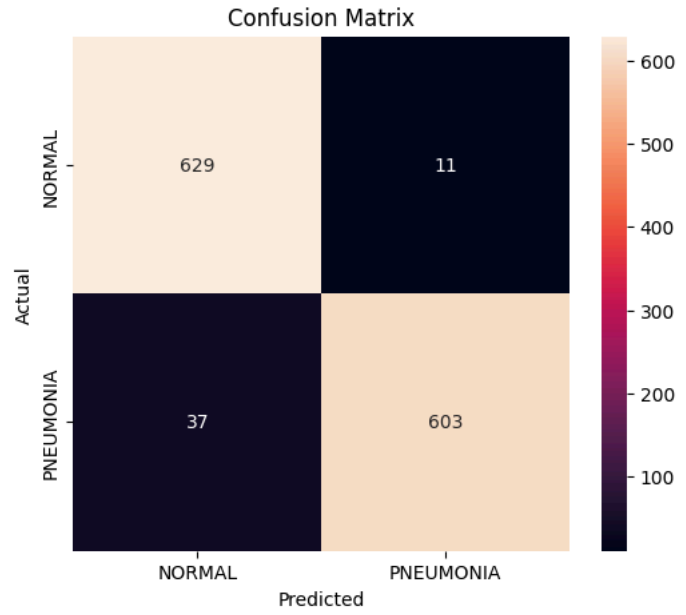
```
40/40 ━━━━━━━━━━━━━━━━━━━━ 24s 317ms/step
              precision    recall  f1-score   support

      NORMAL       0.94      0.98      0.96       640
   PNEUMONIA       0.98      0.94      0.96       640

    accuracy                           0.96      1280
   macro avg       0.96      0.96      0.96      1280
weighted avg       0.96      0.96      0.96      1280
```

🔲 Step 12 — Confusion Matrix Visualization

This step visualizes correct and incorrect predictions made by the model, helping identify false positives and false negatives.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

```
cm = confusion_matrix(true_classes, pred_classes)

plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt="d",
            xticklabels=class_labels,
            yticklabels=class_labels)

plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```




```
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np

# Predictions
y_pred = model.predict(test_data)

# Convert probabilities → classes
y_pred_classes = (y_pred > 0.5).astype("int32").reshape(-1)

# Confusion Matrix
cm = confusion_matrix(test_data.classes, y_pred_classes)
print(cm)

# Classification Report
print(classification_report(
    test_data.classes,
    y_pred_classes,
    target_names=['NORMAL','PNEUMONIA']
))
```

```
40/40 ━━━━━━━━━━━━━━━━━━━━ 13s 316ms/step
[[629  11]
 [ 37 603]]
              precision    recall  f1-score   support

      NORMAL       0.94      0.98      0.96       640
   PNEUMONIA       0.98      0.94      0.96       640

    accuracy                           0.96      1280
   macro avg       0.96      0.96      0.96      1280
weighted avg       0.96      0.96      0.96      1280
```

💾 Step — Saving Trained Model

Is step me hum apna trained CNN model save karte hain taake future me bina dobara training ke use load karke prediction kar saken.

Model ko `.keras` format me save karna recommended hai (HDF5 legacy ho chuka hai).

```
# Save model in Keras format
model.save("pneumonia_cnn_model.keras")
```

⬇️ Step — Downloading Model File

Is step me hum saved model file ko Google Colab se apne computer me download karte hain.

```
from google.colab import files

files.download("pneumonia_cnn_model.keras")
```

Double-click (or enter) to edit

🔮 Step — Model Prediction on New Image

In this step, we test our trained Pneumonia Detection model on a new unseen chest X-ray image.

First, we load the saved trained model. Then, we upload a new image and apply the same preprocessing steps used during training, including resizing the image to 224×224 pixels and rescaling pixel values.

Finally, the model predicts whether the uploaded X-ray image belongs to the **NORMAL** class or **PNEUMONIA** class.

This step demonstrates the real-world usability of our model for automatic pneumonia detection.

```
uploaded = files.upload()
```

Choose Files N X_ray.jfif

N X_ray.jfif(image/jpeg) - 5234 bytes, last modified: 2/16/2026 - 100% done

Saving N X_ray.jfif to N X_ray.jfif

```
# 1️⃣ Load Trained Model
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt

model = load_model("pneumonia_cnn_model.keras")
print("✅ Model Loaded")

# 2️⃣ Image Path (Already Uploaded)
img_path = "N X_ray.jfif"   # ← apni uploaded image ka exact naam likho

# 3️⃣ Load & Preprocess Image
img = image.load_img(img_path, target_size=(224, 224))
img_array = image.img_to_array(img)

img_array = img_array / 255.0
img_array = np.expand_dims(img_array, axis=0)

# 4️⃣ Prediction
prediction = model.predict(img_array)

if prediction[0][0] > 0.5:
    result = "PNEUMONIA"
```