PRACTICAL NO: 5

AIM: Image classification for medical imaging

---CODE---

```
# !kaggle datasets download -d paultimothymooney/chest-xray-pneumonia
# !unzip chest-xray-pneumonia.zip

import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing import image_dataset_from_directory
import matplotlib.pyplot as plt

train_dataset = image_dataset_from_directory(
    '/content/chest_xray/train',
    image_size=(224, 224),
    batch_size=32,
    label_mode='int',
)

val_dataset = image_dataset_from_directory(
    '/content/chest_xray/val',
    image_size=(224, 224),
    batch_size=32,
    label_mode='int',
)

base_model = tf.keras.applications.MobileNetV2(input_shape=(224, 224, 3), include_top=False,
weights='imagenet')

base_model.trainable = False

model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(128, activation='relu'),
    layers.Dense(2, activation='softmax')
])

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

history = model.fit(
    train_dataset,
    epochs=10,
    validation_data=val_dataset
)

model.save('medical_image_classifier.h5')

plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label = 'val_loss')
plt.xlabel('Epoch')
plt.ylabel('Accuracy/Loss')
plt.legend(loc='upper left')
plt.show()
```

---OUTPUT---

```
Found 5216 files belonging to 2 classes.
Found 16 files belonging to 2 classes.
Epoch 1/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 298s 2s/step - accuracy: 0.8471 - loss: 0.3711 - val_accuracy: 0.6875
- val_loss: 1.3905
Epoch 2/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 326s 2s/step - accuracy: 0.9310 - loss: 0.1663 - val_accuracy: 0.7500
```

```
- val_loss: 1.0755
Epoch 3/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 315s 2s/step - accuracy: 0.9512 - loss: 0.1262 - val_accuracy: 0.6875
- val_loss: 0.9275
Epoch 4/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 331s 2s/step - accuracy: 0.9519 - loss: 0.1193 - val_accuracy: 0.7500
- val_loss: 0.6073
Epoch 5/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 313s 2s/step - accuracy: 0.9589 - loss: 0.1023 - val_accuracy: 0.7500
- val_loss: 0.4536
Epoch 6/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 336s 2s/step - accuracy: 0.9646 - loss: 0.0933 - val_accuracy: 0.6875
- val_loss: 0.9344
Epoch 7/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 308s 2s/step - accuracy: 0.9690 - loss: 0.0848 - val_accuracy: 0.7500
- val_loss: 0.8086
Epoch 8/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 336s 2s/step - accuracy: 0.9576 - loss: 0.1086 - val_accuracy: 0.6875
- val_loss: 0.6075
Epoch 9/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 313s 2s/step - accuracy: 0.9698 - loss: 0.0782 - val_accuracy: 0.7500
- val_loss: 0.7221
Epoch 10/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 330s 2s/step - accuracy: 0.9714 - loss: 0.0746 - val_accuracy: 0.6875
- val_loss: 0.7743
```