

NAME : VISHNUTHEERTH E P

CLASS : S2 M.TECH DSC

ROLL NO : CB.SC.P2DSC23017

21DS706 -- DEEP LEARNING FOR VISUAL RECOGNITION

ASSIGNMENT-2

Select any of the open-source signal/image dataset. (Same dataset must not be used by 2 persons).

Selected celeb_a dataset from tensorflow datasets. It is an image dataset.

It had 2lakh+ images. Since, i cannot process such a big dataset(i tried, shown kernel death, GPU limit exceeded and memory issues). So, i have cutdown maximum number of images to 10,000 and did 80:20 train:test split and did the experiments.

Apply the transfer learning approach using the standard architectures: Inception and Exception. (10 marks)

Inception & Xception

Inception

```
In [1]: import os
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.preprocessing.image import load_img, img_to_array
from keras.utils import to_categorical
from keras.applications import InceptionV3,Xception
from keras.models import Sequential
from keras.layers import Dense, GlobalAveragePooling2D, Dropout, Flatten
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import L2
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# paths of train and test datasets
train_dataset = r"C:\Users\vishn\Downloads\celeb_a_dataset_10000\train"
test_dataset = r"C:\Users\vishn\Downloads\celeb_a_dataset_10000\test"

def load_data(data_dir):
    images = []
    labels = []
    for label in os.listdir(data_dir):
        label_dir = os.path.join(data_dir, label)
        for image_file in os.listdir(label_dir):
            image_path = os.path.join(label_dir, image_file)
            image = load_img(image_path, target_size=(299,299))
            image = img_to_array(image)
            image = image / 255.0 # Normalize pixel values
            images.append(image)
            labels.append(label)
    return np.array(images), np.array(labels)

# Load train and test data
X_train, y_train = load_data(train_dataset)
X_test, y_test = load_data(test_dataset)

# Encode categorical labels to numerical labels
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

# Convert numerical labels to one-hot encoding
y_train = tf.keras.utils.to_categorical(y_train_encoded, num_classes=2)
y_test = tf.keras.utils.to_categorical(y_test_encoded, num_classes=2)

# Load InceptionV3 base model
incept_conv = InceptionV3(weights='imagenet', include_top=False, input_shape=(299, 299, 3))

# Freeze all the layers
for layer in incept_conv.layers:
    layer.trainable = False
```

```
# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
# Reduce learning rate on plateau to prevent getting stuck
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=2)

# Creating the model
gap_layer = GlobalAveragePooling2D()(incept_conv.output)
dense_layer_1 = Dense(1024, activation='relu',kernel_regularizer=L2(0.001))(gap_layer)
dropout_layer = Dropout(0.2)(dense_layer_1)
output_layer = Dense(2, activation='softmax')(dropout_layer)

incept_model = Model(inputs=incept_conv.input, outputs=output_layer)
incept_model.summary()
# Compile the model
incept_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
history_incept = incept_model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test,y_test), callbacks=[early_stopping, reduce_lr])
```

Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 299, 299, 3)	0	-
conv2d (Conv2D)	(None, 149, 149, 32)	864	input_layer[0][0]
batch_normalization (BatchNormalization)	(None, 149, 149, 32)	96	conv2d[0][0]
activation (Activation)	(None, 149, 149, 32)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 147, 147, 32)	9,216	activation[0][0]
batch_normalization_1 (BatchNormalization)	(None, 147, 147, 32)	96	conv2d_1[0][0]
activation_1 (Activation)	(None, 147, 147, 32)	0	batch_normalization_1[0][...]
conv2d_2 (Conv2D)	(None, 147, 147, 64)	18,432	activation_1[0][0]
batch_normalization_2 (BatchNormalization)	(None, 147, 147, 64)	192	conv2d_2[0][0]
activation_2 (Activation)	(None, 147, 147, 64)	0	batch_normalization_2[0][...]
max_pooling2d (MaxPooling2D)	(None, 73, 73, 64)	0	activation_2[0][0]
conv2d_3 (Conv2D)	(None, 73, 73, 80)	5,120	max_pooling2d[0][0]
batch_normalization_3 (BatchNormalization)	(None, 73, 73, 80)	240	conv2d_3[0][0]
activation_3 (Activation)	(None, 73, 73, 80)	0	batch_normalization_3[0][...]
conv2d_4 (Conv2D)	(None, 71, 71, 192)	138,240	activation_3[0][0]
batch_normalization_4 (BatchNormalization)	(None, 71, 71, 192)	576	conv2d_4[0][0]
activation_4 (Activation)	(None, 71, 71, 192)	0	batch_normalization_4[0][...]
max_pooling2d_1 (MaxPooling2D)	(None, 35, 35, 192)	0	activation_4[0][0]
conv2d_8 (Conv2D)	(None, 35, 35, 64)	12,288	max_pooling2d_1[0][0]
batch_normalization_8 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_8[0][0]
activation_8 (Activation)	(None, 35, 35, 64)	0	batch_normalization_8[0][...]
conv2d_6 (Conv2D)	(None, 35, 35, 48)	9,216	max_pooling2d_1[0][0]
conv2d_9 (Conv2D)	(None, 35, 35, 96)	55,296	activation_8[0][0]
batch_normalization_6 (BatchNormalization)	(None, 35, 35, 48)	144	conv2d_6[0][0]
batch_normalization_9 (BatchNormalization)	(None, 35, 35, 96)	288	conv2d_9[0][0]
activation_6 (Activation)	(None, 35, 35, 48)	0	batch_normalization_6[0][...]
activation_9 (Activation)	(None, 35, 35, 96)	0	batch_normalization_9[0][...]
average_pooling2d (AveragePooling2D)	(None, 35, 35, 192)	0	max_pooling2d_1[0][0]
conv2d_5 (Conv2D)	(None, 35, 35, 64)	12,288	max_pooling2d_1[0][0]
conv2d_7 (Conv2D)	(None, 35, 35, 64)	76,800	activation_6[0][0]
conv2d_10 (Conv2D)	(None, 35, 35, 96)	82,944	activation_9[0][0]
conv2d_11 (Conv2D)	(None, 35, 35, 32)	6,144	average_pooling2d[0][0]
batch_normalization_5 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_5[0][0]
batch_normalization_7 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_7[0][0]
batch_normalization_10 (BatchNormalization)	(None, 35, 35, 96)	288	conv2d_10[0][0]
batch_normalization_11 (BatchNormalization)	(None, 35, 35, 32)	96	conv2d_11[0][0]
activation_5 (Activation)	(None, 35, 35, 64)	0	batch_normalization_5[0][...]
activation_7 (Activation)	(None, 35, 35, 64)	0	batch_normalization_7[0][...]

activation_10 (Activation)	(None, 35, 35, 96)	0	batch_normalization_10[0]...
activation_11 (Activation)	(None, 35, 35, 32)	0	batch_normalization_11[0]...
mixed0 (Concatenate)	(None, 35, 35, 256)	0	activation_5[0][0], activation_7[0][0], activation_10[0][0], activation_11[0][0]
conv2d_15 (Conv2D)	(None, 35, 35, 64)	16,384	mixed0[0][0]
batch_normalization_15 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_15[0][0]
activation_15 (Activation)	(None, 35, 35, 64)	0	batch_normalization_15[0]...
conv2d_13 (Conv2D)	(None, 35, 35, 48)	12,288	mixed0[0][0]
conv2d_16 (Conv2D)	(None, 35, 35, 96)	55,296	activation_15[0][0]
batch_normalization_13 (BatchNormalization)	(None, 35, 35, 48)	144	conv2d_13[0][0]
batch_normalization_16 (BatchNormalization)	(None, 35, 35, 96)	288	conv2d_16[0][0]
activation_13 (Activation)	(None, 35, 35, 48)	0	batch_normalization_13[0]...
activation_16 (Activation)	(None, 35, 35, 96)	0	batch_normalization_16[0]...
average_pooling2d_1 (AveragePooling2D)	(None, 35, 35, 256)	0	mixed0[0][0]
conv2d_12 (Conv2D)	(None, 35, 35, 64)	16,384	mixed0[0][0]
conv2d_14 (Conv2D)	(None, 35, 35, 64)	76,800	activation_13[0][0]
conv2d_17 (Conv2D)	(None, 35, 35, 96)	82,944	activation_16[0][0]
conv2d_18 (Conv2D)	(None, 35, 35, 64)	16,384	average_pooling2d_1[0][0]
batch_normalization_12 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_12[0][0]
batch_normalization_14 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_14[0][0]
batch_normalization_17 (BatchNormalization)	(None, 35, 35, 96)	288	conv2d_17[0][0]
batch_normalization_18 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_18[0][0]
activation_12 (Activation)	(None, 35, 35, 64)	0	batch_normalization_12[0]...
activation_14 (Activation)	(None, 35, 35, 64)	0	batch_normalization_14[0]...
activation_17 (Activation)	(None, 35, 35, 96)	0	batch_normalization_17[0]...
activation_18 (Activation)	(None, 35, 35, 64)	0	batch_normalization_18[0]...
mixed1 (Concatenate)	(None, 35, 35, 288)	0	activation_12[0][0], activation_14[0][0], activation_17[0][0], activation_18[0][0]
conv2d_22 (Conv2D)	(None, 35, 35, 64)	18,432	mixed1[0][0]
batch_normalization_22 (BatchNormalization)	(None, 35, 35, 64)	192	conv2d_22[0][0]
activation_22 (Activation)	(None, 35, 35, 64)	0	batch_normalization_22[0]...
conv2d_20 (Conv2D)	(None, 35, 35, 48)	13,824	mixed1[0][0]
conv2d_23 (Conv2D)	(None, 35, 35, 96)	55,296	activation_22[0][0]
batch_normalization_20 (BatchNormalization)	(None, 35, 35, 48)	144	conv2d_20[0][0]
batch_normalization_23 (BatchNormalization)	(None, 35, 35, 96)	288	conv2d_23[0][0]
activation_20 (Activation)	(None, 35, 35, 48)	0	batch_normalization_20[0]...
activation_23 (Activation)	(None, 35, 35, 96)	0	batch_normalization_23[0]...
average_pooling2d_2 (AveragePooling2D)	(None, 35, 35, 288)	0	mixed1[0][0]
conv2d_19 (Conv2D)	(None, 35, 35, 64)	18,432	mixed1[0][0]
conv2d_21 (Conv2D)	(None, 35, 35, 64)	76,800	activation_20[0][0]

conv2d_24 (Conv2D)	(None , 35 , 35 , 96)	82,944	activation_23[0][0]
conv2d_25 (Conv2D)	(None , 35 , 35 , 64)	18,432	average_pooling2d_2[0][0]
batch_normalization_19 (BatchNormalization)	(None , 35 , 35 , 64)	192	conv2d_19[0][0]
batch_normalization_21 (BatchNormalization)	(None , 35 , 35 , 64)	192	conv2d_21[0][0]
batch_normalization_24 (BatchNormalization)	(None , 35 , 35 , 96)	288	conv2d_24[0][0]
batch_normalization_25 (BatchNormalization)	(None , 35 , 35 , 64)	192	conv2d_25[0][0]
activation_19 (Activation)	(None , 35 , 35 , 64)	0	batch_normalization_19[0]...
activation_21 (Activation)	(None , 35 , 35 , 64)	0	batch_normalization_21[0]...
activation_24 (Activation)	(None , 35 , 35 , 96)	0	batch_normalization_24[0]...
activation_25 (Activation)	(None , 35 , 35 , 64)	0	batch_normalization_25[0]...
mixed2 (Concatenate)	(None , 35 , 35 , 288)	0	activation_19[0][0], activation_21[0][0], activation_24[0][0], activation_25[0][0]
conv2d_27 (Conv2D)	(None , 35 , 35 , 64)	18,432	mixed2[0][0]
batch_normalization_27 (BatchNormalization)	(None , 35 , 35 , 64)	192	conv2d_27[0][0]
activation_27 (Activation)	(None , 35 , 35 , 64)	0	batch_normalization_27[0]...
conv2d_28 (Conv2D)	(None , 35 , 35 , 96)	55,296	activation_27[0][0]
batch_normalization_28 (BatchNormalization)	(None , 35 , 35 , 96)	288	conv2d_28[0][0]
activation_28 (Activation)	(None , 35 , 35 , 96)	0	batch_normalization_28[0]...
conv2d_26 (Conv2D)	(None , 17 , 17 , 384)	995,328	mixed2[0][0]
conv2d_29 (Conv2D)	(None , 17 , 17 , 96)	82,944	activation_28[0][0]
batch_normalization_26 (BatchNormalization)	(None , 17 , 17 , 384)	1,152	conv2d_26[0][0]
batch_normalization_29 (BatchNormalization)	(None , 17 , 17 , 96)	288	conv2d_29[0][0]
activation_26 (Activation)	(None , 17 , 17 , 384)	0	batch_normalization_26[0]...
activation_29 (Activation)	(None , 17 , 17 , 96)	0	batch_normalization_29[0]...
max_pooling2d_2 (MaxPooling2D)	(None , 17 , 17 , 288)	0	mixed2[0][0]
mixed3 (Concatenate)	(None , 17 , 17 , 768)	0	activation_26[0][0], activation_29[0][0], max_pooling2d_2[0][0]
conv2d_34 (Conv2D)	(None , 17 , 17 , 128)	98,304	mixed3[0][0]
batch_normalization_34 (BatchNormalization)	(None , 17 , 17 , 128)	384	conv2d_34[0][0]
activation_34 (Activation)	(None , 17 , 17 , 128)	0	batch_normalization_34[0]...
conv2d_35 (Conv2D)	(None , 17 , 17 , 128)	114,688	activation_34[0][0]
batch_normalization_35 (BatchNormalization)	(None , 17 , 17 , 128)	384	conv2d_35[0][0]
activation_35 (Activation)	(None , 17 , 17 , 128)	0	batch_normalization_35[0]...
conv2d_31 (Conv2D)	(None , 17 , 17 , 128)	98,304	mixed3[0][0]
conv2d_36 (Conv2D)	(None , 17 , 17 , 128)	114,688	activation_35[0][0]
batch_normalization_31 (BatchNormalization)	(None , 17 , 17 , 128)	384	conv2d_31[0][0]
batch_normalization_36 (BatchNormalization)	(None , 17 , 17 , 128)	384	conv2d_36[0][0]
activation_31 (Activation)	(None , 17 , 17 , 128)	0	batch_normalization_31[0]...
activation_36 (Activation)	(None , 17 , 17 , 128)	0	batch_normalization_36[0]...

conv2d_32 (Conv2D)	(None, 17, 17, 128)	114,688	activation_31[0][0]
conv2d_37 (Conv2D)	(None, 17, 17, 128)	114,688	activation_36[0][0]
batch_normalization_32 (BatchNormalization)	(None, 17, 17, 128)	384	conv2d_32[0][0]
batch_normalization_37 (BatchNormalization)	(None, 17, 17, 128)	384	conv2d_37[0][0]
activation_32 (Activation)	(None, 17, 17, 128)	0	batch_normalization_32[0]...
activation_37 (Activation)	(None, 17, 17, 128)	0	batch_normalization_37[0]...
average_pooling2d_3 (AveragePooling2D)	(None, 17, 17, 768)	0	mixed3[0][0]
conv2d_30 (Conv2D)	(None, 17, 17, 192)	147,456	mixed3[0][0]
conv2d_33 (Conv2D)	(None, 17, 17, 192)	172,032	activation_32[0][0]
conv2d_38 (Conv2D)	(None, 17, 17, 192)	172,032	activation_37[0][0]
conv2d_39 (Conv2D)	(None, 17, 17, 192)	147,456	average_pooling2d_3[0][0]
batch_normalization_30 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_30[0][0]
batch_normalization_33 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_33[0][0]
batch_normalization_38 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_38[0][0]
batch_normalization_39 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_39[0][0]
activation_30 (Activation)	(None, 17, 17, 192)	0	batch_normalization_30[0]...
activation_33 (Activation)	(None, 17, 17, 192)	0	batch_normalization_33[0]...
activation_38 (Activation)	(None, 17, 17, 192)	0	batch_normalization_38[0]...
activation_39 (Activation)	(None, 17, 17, 192)	0	batch_normalization_39[0]...
mixed4 (Concatenate)	(None, 17, 17, 768)	0	activation_30[0][0], activation_33[0][0], activation_38[0][0], activation_39[0][0]
conv2d_44 (Conv2D)	(None, 17, 17, 160)	122,880	mixed4[0][0]
batch_normalization_44 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_44[0][0]
activation_44 (Activation)	(None, 17, 17, 160)	0	batch_normalization_44[0]...
conv2d_45 (Conv2D)	(None, 17, 17, 160)	179,200	activation_44[0][0]
batch_normalization_45 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_45[0][0]
activation_45 (Activation)	(None, 17, 17, 160)	0	batch_normalization_45[0]...
conv2d_41 (Conv2D)	(None, 17, 17, 160)	122,880	mixed4[0][0]
conv2d_46 (Conv2D)	(None, 17, 17, 160)	179,200	activation_45[0][0]
batch_normalization_41 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_41[0][0]
batch_normalization_46 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_46[0][0]
activation_41 (Activation)	(None, 17, 17, 160)	0	batch_normalization_41[0]...
activation_46 (Activation)	(None, 17, 17, 160)	0	batch_normalization_46[0]...
conv2d_42 (Conv2D)	(None, 17, 17, 160)	179,200	activation_41[0][0]
conv2d_47 (Conv2D)	(None, 17, 17, 160)	179,200	activation_46[0][0]
batch_normalization_42 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_42[0][0]
batch_normalization_47 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_47[0][0]
activation_42 (Activation)	(None, 17, 17, 160)	0	batch_normalization_42[0]...
activation_47 (Activation)	(None, 17, 17, 160)	0	batch_normalization_47[0]...
average_pooling2d_4	(None, 17, 17, 768)	0	mixed4[0][0]

(AveragePooling2D)			
conv2d_40 (Conv2D)	(None, 17, 17, 192)	147,456	mixed4[0][0]
conv2d_43 (Conv2D)	(None, 17, 17, 192)	215,040	activation_42[0][0]
conv2d_48 (Conv2D)	(None, 17, 17, 192)	215,040	activation_47[0][0]
conv2d_49 (Conv2D)	(None, 17, 17, 192)	147,456	average_pooling2d_4[0][0]
batch_normalization_40 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_40[0][0]
batch_normalization_43 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_43[0][0]
batch_normalization_48 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_48[0][0]
batch_normalization_49 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_49[0][0]
activation_40 (Activation)	(None, 17, 17, 192)	0	batch_normalization_40[0]...
activation_43 (Activation)	(None, 17, 17, 192)	0	batch_normalization_43[0]...
activation_48 (Activation)	(None, 17, 17, 192)	0	batch_normalization_48[0]...
activation_49 (Activation)	(None, 17, 17, 192)	0	batch_normalization_49[0]...
mixed5 (Concatenate)	(None, 17, 17, 768)	0	activation_40[0][0], activation_43[0][0], activation_48[0][0], activation_49[0][0]
conv2d_54 (Conv2D)	(None, 17, 17, 160)	122,880	mixed5[0][0]
batch_normalization_54 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_54[0][0]
activation_54 (Activation)	(None, 17, 17, 160)	0	batch_normalization_54[0]...
conv2d_55 (Conv2D)	(None, 17, 17, 160)	179,200	activation_54[0][0]
batch_normalization_55 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_55[0][0]
activation_55 (Activation)	(None, 17, 17, 160)	0	batch_normalization_55[0]...
conv2d_51 (Conv2D)	(None, 17, 17, 160)	122,880	mixed5[0][0]
conv2d_56 (Conv2D)	(None, 17, 17, 160)	179,200	activation_55[0][0]
batch_normalization_51 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_51[0][0]
batch_normalization_56 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_56[0][0]
activation_51 (Activation)	(None, 17, 17, 160)	0	batch_normalization_51[0]...
activation_56 (Activation)	(None, 17, 17, 160)	0	batch_normalization_56[0]...
conv2d_52 (Conv2D)	(None, 17, 17, 160)	179,200	activation_51[0][0]
conv2d_57 (Conv2D)	(None, 17, 17, 160)	179,200	activation_56[0][0]
batch_normalization_52 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_52[0][0]
batch_normalization_57 (BatchNormalization)	(None, 17, 17, 160)	480	conv2d_57[0][0]
activation_52 (Activation)	(None, 17, 17, 160)	0	batch_normalization_52[0]...
activation_57 (Activation)	(None, 17, 17, 160)	0	batch_normalization_57[0]...
average_pooling2d_5 (AveragePooling2D)	(None, 17, 17, 768)	0	mixed5[0][0]
conv2d_50 (Conv2D)	(None, 17, 17, 192)	147,456	mixed5[0][0]
conv2d_53 (Conv2D)	(None, 17, 17, 192)	215,040	activation_52[0][0]
conv2d_58 (Conv2D)	(None, 17, 17, 192)	215,040	activation_57[0][0]
conv2d_59 (Conv2D)	(None, 17, 17, 192)	147,456	average_pooling2d_5[0][0]
batch_normalization_50 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_50[0][0]
batch_normalization_53 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_53[0][0]

batch_normalization_58 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_58[0][0]
batch_normalization_59 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_59[0][0]
activation_50 (Activation)	(None, 17, 17, 192)	0	batch_normalization_50[0]...
activation_53 (Activation)	(None, 17, 17, 192)	0	batch_normalization_53[0]...
activation_58 (Activation)	(None, 17, 17, 192)	0	batch_normalization_58[0]...
activation_59 (Activation)	(None, 17, 17, 192)	0	batch_normalization_59[0]...
mixed6 (Concatenate)	(None, 17, 17, 768)	0	activation_50[0][0], activation_53[0][0], activation_58[0][0], activation_59[0][0]
conv2d_64 (Conv2D)	(None, 17, 17, 192)	147,456	mixed6[0][0]
batch_normalization_64 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_64[0][0]
activation_64 (Activation)	(None, 17, 17, 192)	0	batch_normalization_64[0]...
conv2d_65 (Conv2D)	(None, 17, 17, 192)	258,048	activation_64[0][0]
batch_normalization_65 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_65[0][0]
activation_65 (Activation)	(None, 17, 17, 192)	0	batch_normalization_65[0]...
conv2d_61 (Conv2D)	(None, 17, 17, 192)	147,456	mixed6[0][0]
conv2d_66 (Conv2D)	(None, 17, 17, 192)	258,048	activation_65[0][0]
batch_normalization_61 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_61[0][0]
batch_normalization_66 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_66[0][0]
activation_61 (Activation)	(None, 17, 17, 192)	0	batch_normalization_61[0]...
activation_66 (Activation)	(None, 17, 17, 192)	0	batch_normalization_66[0]...
conv2d_62 (Conv2D)	(None, 17, 17, 192)	258,048	activation_61[0][0]
conv2d_67 (Conv2D)	(None, 17, 17, 192)	258,048	activation_66[0][0]
batch_normalization_62 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_62[0][0]
batch_normalization_67 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_67[0][0]
activation_62 (Activation)	(None, 17, 17, 192)	0	batch_normalization_62[0]...
activation_67 (Activation)	(None, 17, 17, 192)	0	batch_normalization_67[0]...
average_pooling2d_6 (AveragePooling2D)	(None, 17, 17, 768)	0	mixed6[0][0]
conv2d_60 (Conv2D)	(None, 17, 17, 192)	147,456	mixed6[0][0]
conv2d_63 (Conv2D)	(None, 17, 17, 192)	258,048	activation_62[0][0]
conv2d_68 (Conv2D)	(None, 17, 17, 192)	258,048	activation_67[0][0]
conv2d_69 (Conv2D)	(None, 17, 17, 192)	147,456	average_pooling2d_6[0][0]
batch_normalization_60 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_60[0][0]
batch_normalization_63 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_63[0][0]
batch_normalization_68 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_68[0][0]
batch_normalization_69 (BatchNormalization)	(None, 17, 17, 192)	576	conv2d_69[0][0]
activation_60 (Activation)	(None, 17, 17, 192)	0	batch_normalization_60[0]...
activation_63 (Activation)	(None, 17, 17, 192)	0	batch_normalization_63[0]...
activation_68 (Activation)	(None, 17, 17, 192)	0	batch_normalization_68[0]...
activation_69 (Activation)	(None, 17, 17, 192)	0	batch_normalization_69[0]...

mixed7 (Concatenate)	(None , 17 , 17 , 768)	0	activation_60[0][0], activation_63[0][0], activation_68[0][0], activation_69[0][0]
conv2d_72 (Conv2D)	(None , 17 , 17 , 192)	147,456	mixed7[0][0]
batch_normalization_72 (BatchNormalization)	(None , 17 , 17 , 192)	576	conv2d_72[0][0]
activation_72 (Activation)	(None , 17 , 17 , 192)	0	batch_normalization_72[0]...
conv2d_73 (Conv2D)	(None , 17 , 17 , 192)	258,048	activation_72[0][0]
batch_normalization_73 (BatchNormalization)	(None , 17 , 17 , 192)	576	conv2d_73[0][0]
activation_73 (Activation)	(None , 17 , 17 , 192)	0	batch_normalization_73[0]...
conv2d_70 (Conv2D)	(None , 17 , 17 , 192)	147,456	mixed7[0][0]
conv2d_74 (Conv2D)	(None , 17 , 17 , 192)	258,048	activation_73[0][0]
batch_normalization_70 (BatchNormalization)	(None , 17 , 17 , 192)	576	conv2d_70[0][0]
batch_normalization_74 (BatchNormalization)	(None , 17 , 17 , 192)	576	conv2d_74[0][0]
activation_70 (Activation)	(None , 17 , 17 , 192)	0	batch_normalization_70[0]...
activation_74 (Activation)	(None , 17 , 17 , 192)	0	batch_normalization_74[0]...
conv2d_71 (Conv2D)	(None , 8 , 8 , 320)	552,960	activation_70[0][0]
conv2d_75 (Conv2D)	(None , 8 , 8 , 192)	331,776	activation_74[0][0]
batch_normalization_71 (BatchNormalization)	(None , 8 , 8 , 320)	960	conv2d_71[0][0]
batch_normalization_75 (BatchNormalization)	(None , 8 , 8 , 192)	576	conv2d_75[0][0]
activation_71 (Activation)	(None , 8 , 8 , 320)	0	batch_normalization_71[0]...
activation_75 (Activation)	(None , 8 , 8 , 192)	0	batch_normalization_75[0]...
max_pooling2d_3 (MaxPooling2D)	(None , 8 , 8 , 768)	0	mixed7[0][0]
mixed8 (Concatenate)	(None , 8 , 8 , 1280)	0	activation_71[0][0], activation_75[0][0], max_pooling2d_3[0][0]
conv2d_80 (Conv2D)	(None , 8 , 8 , 448)	573,440	mixed8[0][0]
batch_normalization_80 (BatchNormalization)	(None , 8 , 8 , 448)	1,344	conv2d_80[0][0]
activation_80 (Activation)	(None , 8 , 8 , 448)	0	batch_normalization_80[0]...
conv2d_77 (Conv2D)	(None , 8 , 8 , 384)	491,520	mixed8[0][0]
conv2d_81 (Conv2D)	(None , 8 , 8 , 384)	1,548,288	activation_80[0][0]
batch_normalization_77 (BatchNormalization)	(None , 8 , 8 , 384)	1,152	conv2d_77[0][0]
batch_normalization_81 (BatchNormalization)	(None , 8 , 8 , 384)	1,152	conv2d_81[0][0]
activation_77 (Activation)	(None , 8 , 8 , 384)	0	batch_normalization_77[0]...
activation_81 (Activation)	(None , 8 , 8 , 384)	0	batch_normalization_81[0]...
conv2d_78 (Conv2D)	(None , 8 , 8 , 384)	442,368	activation_77[0][0]
conv2d_79 (Conv2D)	(None , 8 , 8 , 384)	442,368	activation_77[0][0]
conv2d_82 (Conv2D)	(None , 8 , 8 , 384)	442,368	activation_81[0][0]
conv2d_83 (Conv2D)	(None , 8 , 8 , 384)	442,368	activation_81[0][0]
average_pooling2d_7 (AveragePooling2D)	(None , 8 , 8 , 1280)	0	mixed8[0][0]
conv2d_76 (Conv2D)	(None , 8 , 8 , 320)	409,600	mixed8[0][0]
batch_normalization_78 (BatchNormalization)	(None , 8 , 8 , 384)	1,152	conv2d_78[0][0]
batch_normalization_79 (BatchNormalization)	(None , 8 , 8 , 384)	1,152	conv2d_79[0][0]

batch_normalization_82 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_82[0][0]
batch_normalization_83 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_83[0][0]
conv2d_84 (Conv2D)	(None, 8, 8, 192)	245,760	average_pooling2d_7[0][0]
batch_normalization_76 (BatchNormalization)	(None, 8, 8, 320)	960	conv2d_76[0][0]
activation_78 (Activation)	(None, 8, 8, 384)	0	batch_normalization_78[0]...
activation_79 (Activation)	(None, 8, 8, 384)	0	batch_normalization_79[0]...
activation_82 (Activation)	(None, 8, 8, 384)	0	batch_normalization_82[0]...
activation_83 (Activation)	(None, 8, 8, 384)	0	batch_normalization_83[0]...
batch_normalization_84 (BatchNormalization)	(None, 8, 8, 192)	576	conv2d_84[0][0]
activation_76 (Activation)	(None, 8, 8, 320)	0	batch_normalization_76[0]...
mixed9_0 (Concatenate)	(None, 8, 8, 768)	0	activation_78[0][0], activation_79[0][0]
concatenate (Concatenate)	(None, 8, 8, 768)	0	activation_82[0][0], activation_83[0][0]
activation_84 (Activation)	(None, 8, 8, 192)	0	batch_normalization_84[0]...
mixed9 (Concatenate)	(None, 8, 8, 2048)	0	activation_76[0][0], mixed9_0[0][0], concatenate[0][0], activation_84[0][0]
conv2d_89 (Conv2D)	(None, 8, 8, 448)	917,504	mixed9[0][0]
batch_normalization_89 (BatchNormalization)	(None, 8, 8, 448)	1,344	conv2d_89[0][0]
activation_89 (Activation)	(None, 8, 8, 448)	0	batch_normalization_89[0]...
conv2d_86 (Conv2D)	(None, 8, 8, 384)	786,432	mixed9[0][0]
conv2d_90 (Conv2D)	(None, 8, 8, 384)	1,548,288	activation_89[0][0]
batch_normalization_86 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_86[0][0]
batch_normalization_90 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_90[0][0]
activation_86 (Activation)	(None, 8, 8, 384)	0	batch_normalization_86[0]...
activation_90 (Activation)	(None, 8, 8, 384)	0	batch_normalization_90[0]...
conv2d_87 (Conv2D)	(None, 8, 8, 384)	442,368	activation_86[0][0]
conv2d_88 (Conv2D)	(None, 8, 8, 384)	442,368	activation_86[0][0]
conv2d_91 (Conv2D)	(None, 8, 8, 384)	442,368	activation_90[0][0]
conv2d_92 (Conv2D)	(None, 8, 8, 384)	442,368	activation_90[0][0]
average_pooling2d_8 (AveragePooling2D)	(None, 8, 8, 2048)	0	mixed9[0][0]
conv2d_85 (Conv2D)	(None, 8, 8, 320)	655,360	mixed9[0][0]
batch_normalization_87 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_87[0][0]
batch_normalization_88 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_88[0][0]
batch_normalization_91 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_91[0][0]
batch_normalization_92 (BatchNormalization)	(None, 8, 8, 384)	1,152	conv2d_92[0][0]
conv2d_93 (Conv2D)	(None, 8, 8, 192)	393,216	average_pooling2d_8[0][0]
batch_normalization_85 (BatchNormalization)	(None, 8, 8, 320)	960	conv2d_85[0][0]
activation_87 (Activation)	(None, 8, 8, 384)	0	batch_normalization_87[0]...
activation_88 (Activation)	(None, 8, 8, 384)	0	batch_normalization_88[0]...

activation_91 (Activation)	(None, 8, 8, 384)	0	batch_normalization_91[0]...
activation_92 (Activation)	(None, 8, 8, 384)	0	batch_normalization_92[0]...
batch_normalization_93 (BatchNormalization)	(None, 8, 8, 192)	576	conv2d_93[0][0]
activation_85 (Activation)	(None, 8, 8, 320)	0	batch_normalization_85[0]...
mixed9_1 (Concatenate)	(None, 8, 8, 768)	0	activation_87[0][0], activation_88[0][0]
concatenate_1 (Concatenate)	(None, 8, 8, 768)	0	activation_91[0][0], activation_92[0][0]
activation_93 (Activation)	(None, 8, 8, 192)	0	batch_normalization_93[0]...
mixed10 (Concatenate)	(None, 8, 8, 2048)	0	activation_85[0][0], mixed9_1[0][0], concatenate_1[0][0], activation_93[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	mixed10[0][0]
dense (Dense)	(None, 1024)	2,098,176	global_average_pooling2d[...
dropout (Dropout)	(None, 1024)	0	dense[0][0]
dense_1 (Dense)	(None, 2)	2,050	dropout[0][0]

Total params: 23,903,010 (91.18 MB)

Trainable params: 2,100,226 (8.01 MB)

Non-trainable params: 21,802,784 (83.17 MB)

Epoch 1/5
250/250 560s 2s/step - accuracy: 0.8766 - loss: 1.2579 - val_accuracy: 0.9030 - val_loss: 0.5233 - learning_rate: 0.0010
Epoch 2/5
250/250 559s 2s/step - accuracy: 0.9226 - loss: 0.4190 - val_accuracy: 0.9300 - val_loss: 0.2961 - learning_rate: 0.0010
Epoch 3/5
250/250 551s 2s/step - accuracy: 0.9362 - loss: 0.2671 - val_accuracy: 0.9390 - val_loss: 0.2383 - learning_rate: 0.0010
Epoch 4/5
250/250 545s 2s/step - accuracy: 0.9318 - loss: 0.2403 - val_accuracy: 0.9320 - val_loss: 0.2233 - learning_rate: 0.0010
Epoch 5/5
250/250 544s 2s/step - accuracy: 0.9354 - loss: 0.2088 - val_accuracy: 0.9350 - val_loss: 0.2184 - learning_rate: 0.0010

Xception

```
In [3]: xcept_conv = Xception(weights='imagenet', include_top=False, input_shape=(299, 299, 3))

# Freeze all the layers
for layer in xcept_conv.layers:
    layer.trainable = False

# Creating the model
gap_layer = GlobalAveragePooling2D()(xcept_conv.output)
dense_layer_1 = Dense(1024, activation='relu')(gap_layer)
dropout_layer = Dropout(0.2)(dense_layer_1)
output_layer = Dense(2, activation='softmax')(dropout_layer)

xcept_model = Model(inputs=xcept_conv.input, outputs=output_layer)
xcept_model.summary()
# Compile the model
xcept_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
history_xcept = xcept_model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test,y_test), callbacks=[early_stopping, reduce_lr])
```

Model: "functional_5"

Layer (type)	Output Shape	Param #	Connected to
input_layer_2 (InputLayer)	(None, 299, 299, 3)	0	-
block1_conv1 (Conv2D)	(None, 149, 149, 32)	864	input_layer_2[0][0]
block1_conv1_bn (BatchNormalization)	(None, 149, 149, 32)	128	block1_conv1[0][0]
block1_conv1_act (Activation)	(None, 149, 149, 32)	0	block1_conv1_bn[0][0]
block1_conv2 (Conv2D)	(None, 147, 147, 64)	18,432	block1_conv1_act[0][0]
block1_conv2_bn (BatchNormalization)	(None, 147, 147, 64)	256	block1_conv2[0][0]
block1_conv2_act (Activation)	(None, 147, 147, 64)	0	block1_conv2_bn[0][0]
block2_sepconv1 (SeparableConv2D)	(None, 147, 147, 128)	8,768	block1_conv2_act[0][0]
block2_sepconv1_bn (BatchNormalization)	(None, 147, 147, 128)	512	block2_sepconv1[0][0]
block2_sepconv2_act (Activation)	(None, 147, 147, 128)	0	block2_sepconv1_bn[0][0]
block2_sepconv2 (SeparableConv2D)	(None, 147, 147, 128)	17,536	block2_sepconv2_act[0][0]
block2_sepconv2_bn (BatchNormalization)	(None, 147, 147, 128)	512	block2_sepconv2[0][0]
conv2d_98 (Conv2D)	(None, 74, 74, 128)	8,192	block1_conv2_act[0][0]
block2_pool (MaxPooling2D)	(None, 74, 74, 128)	0	block2_sepconv2_bn[0][0]
batch_normalization_98 (BatchNormalization)	(None, 74, 74, 128)	512	conv2d_98[0][0]
add_12 (Add)	(None, 74, 74, 128)	0	block2_pool[0][0], batch_normalization_98[0]...
block3_sepconv1_act (Activation)	(None, 74, 74, 128)	0	add_12[0][0]
block3_sepconv1 (SeparableConv2D)	(None, 74, 74, 256)	33,920	block3_sepconv1_act[0][0]
block3_sepconv1_bn (BatchNormalization)	(None, 74, 74, 256)	1,024	block3_sepconv1[0][0]
block3_sepconv2_act (Activation)	(None, 74, 74, 256)	0	block3_sepconv1_bn[0][0]
block3_sepconv2 (SeparableConv2D)	(None, 74, 74, 256)	67,840	block3_sepconv2_act[0][0]
block3_sepconv2_bn (BatchNormalization)	(None, 74, 74, 256)	1,024	block3_sepconv2[0][0]
conv2d_99 (Conv2D)	(None, 37, 37, 256)	32,768	add_12[0][0]
block3_pool (MaxPooling2D)	(None, 37, 37, 256)	0	block3_sepconv2_bn[0][0]
batch_normalization_99 (BatchNormalization)	(None, 37, 37, 256)	1,024	conv2d_99[0][0]
add_13 (Add)	(None, 37, 37, 256)	0	block3_pool[0][0], batch_normalization_99[0]...
block4_sepconv1_act (Activation)	(None, 37, 37, 256)	0	add_13[0][0]
block4_sepconv1 (SeparableConv2D)	(None, 37, 37, 728)	188,672	block4_sepconv1_act[0][0]
block4_sepconv1_bn (BatchNormalization)	(None, 37, 37, 728)	2,912	block4_sepconv1[0][0]
block4_sepconv2_act (Activation)	(None, 37, 37, 728)	0	block4_sepconv1_bn[0][0]
block4_sepconv2 (SeparableConv2D)	(None, 37, 37, 728)	536,536	block4_sepconv2_act[0][0]
block4_sepconv2_bn (BatchNormalization)	(None, 37, 37, 728)	2,912	block4_sepconv2[0][0]
conv2d_100 (Conv2D)	(None, 19, 19, 728)	186,368	add_13[0][0]
block4_pool (MaxPooling2D)	(None, 19, 19, 728)	0	block4_sepconv2_bn[0][0]

batch_normalization_100 (BatchNormalization)	(None, 19, 19, 728)	2,912	conv2d_100[0][0]
add_14 (Add)	(None, 19, 19, 728)	0	block4_pool[0][0], batch_normalization_100[0...
block5_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_14[0][0]
block5_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block5_sepconv1_act[0][0]
block5_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block5_sepconv1[0][0]
block5_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block5_sepconv1_bn[0][0]
block5_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block5_sepconv2_act[0][0]
block5_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block5_sepconv2[0][0]
block5_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block5_sepconv2_bn[0][0]
block5_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block5_sepconv3_act[0][0]
block5_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block5_sepconv3[0][0]
add_15 (Add)	(None, 19, 19, 728)	0	block5_sepconv3_bn[0][0], add_14[0][0]
block6_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_15[0][0]
block6_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block6_sepconv1_act[0][0]
block6_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block6_sepconv1[0][0]
block6_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block6_sepconv1_bn[0][0]
block6_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block6_sepconv2_act[0][0]
block6_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block6_sepconv2[0][0]
block6_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block6_sepconv2_bn[0][0]
block6_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block6_sepconv3_act[0][0]
block6_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block6_sepconv3[0][0]
add_16 (Add)	(None, 19, 19, 728)	0	block6_sepconv3_bn[0][0], add_15[0][0]
block7_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_16[0][0]
block7_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block7_sepconv1_act[0][0]
block7_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block7_sepconv1[0][0]
block7_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block7_sepconv1_bn[0][0]
block7_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block7_sepconv2_act[0][0]
block7_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block7_sepconv2[0][0]
block7_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block7_sepconv2_bn[0][0]
block7_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block7_sepconv3_act[0][0]
block7_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block7_sepconv3[0][0]

add_17 (Add)	(None, 19, 19, 728)	0	block7_sepconv3_bn[0][0], add_16[0][0]
block8_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_17[0][0]
block8_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block8_sepconv1_act[0][0]
block8_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block8_sepconv1[0][0]
block8_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block8_sepconv1_bn[0][0]
block8_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block8_sepconv2_act[0][0]
block8_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block8_sepconv2[0][0]
block8_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block8_sepconv2_bn[0][0]
block8_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block8_sepconv3_act[0][0]
block8_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block8_sepconv3[0][0]
add_18 (Add)	(None, 19, 19, 728)	0	block8_sepconv3_bn[0][0], add_17[0][0]
block9_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_18[0][0]
block9_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block9_sepconv1_act[0][0]
block9_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block9_sepconv1[0][0]
block9_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block9_sepconv1_bn[0][0]
block9_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block9_sepconv2_act[0][0]
block9_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block9_sepconv2[0][0]
block9_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block9_sepconv2_bn[0][0]
block9_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block9_sepconv3_act[0][0]
block9_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block9_sepconv3[0][0]
add_19 (Add)	(None, 19, 19, 728)	0	block9_sepconv3_bn[0][0], add_18[0][0]
block10_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_19[0][0]
block10_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block10_sepconv1_act[0][0]
block10_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block10_sepconv1[0][0]
block10_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block10_sepconv1_bn[0][0]
block10_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block10_sepconv2_act[0][0]
block10_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block10_sepconv2[0][0]
block10_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block10_sepconv2_bn[0][0]
block10_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block10_sepconv3_act[0][0]
block10_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block10_sepconv3[0][0]
add_20 (Add)	(None, 19, 19, 728)	0	block10_sepconv3_bn[0][0], add_19[0][0]

block11_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_20[0][0]
block11_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block11_sepconv1_act[0][0]
block11_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block11_sepconv1[0][0]
block11_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block11_sepconv1_bn[0][0]
block11_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block11_sepconv2_act[0][0]
block11_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block11_sepconv2[0][0]
block11_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block11_sepconv2_bn[0][0]
block11_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block11_sepconv3_act[0][0]
block11_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block11_sepconv3[0][0]
add_21 (Add)	(None, 19, 19, 728)	0	block11_sepconv3_bn[0][0], add_20[0][0]
block12_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_21[0][0]
block12_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block12_sepconv1_act[0][0]
block12_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block12_sepconv1[0][0]
block12_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block12_sepconv1_bn[0][0]
block12_sepconv2 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block12_sepconv2_act[0][0]
block12_sepconv2_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block12_sepconv2[0][0]
block12_sepconv3_act (Activation)	(None, 19, 19, 728)	0	block12_sepconv2_bn[0][0]
block12_sepconv3 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block12_sepconv3_act[0][0]
block12_sepconv3_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block12_sepconv3[0][0]
add_22 (Add)	(None, 19, 19, 728)	0	block12_sepconv3_bn[0][0], add_21[0][0]
block13_sepconv1_act (Activation)	(None, 19, 19, 728)	0	add_22[0][0]
block13_sepconv1 (SeparableConv2D)	(None, 19, 19, 728)	536,536	block13_sepconv1_act[0][0]
block13_sepconv1_bn (BatchNormalization)	(None, 19, 19, 728)	2,912	block13_sepconv1[0][0]
block13_sepconv2_act (Activation)	(None, 19, 19, 728)	0	block13_sepconv1_bn[0][0]
block13_sepconv2 (SeparableConv2D)	(None, 19, 19, 1024)	752,024	block13_sepconv2_act[0][0]
block13_sepconv2_bn (BatchNormalization)	(None, 19, 19, 1024)	4,096	block13_sepconv2[0][0]
conv2d_101 (Conv2D)	(None, 10, 10, 1024)	745,472	add_22[0][0]
block13_pool (MaxPooling2D)	(None, 10, 10, 1024)	0	block13_sepconv2_bn[0][0]
batch_normalization_101 (BatchNormalization)	(None, 10, 10, 1024)	4,096	conv2d_101[0][0]
add_23 (Add)	(None, 10, 10, 1024)	0	block13_pool[0][0], batch_normalization_101[0...
block14_sepconv1 (SeparableConv2D)	(None, 10, 10, 1536)	1,582,080	add_23[0][0]
block14_sepconv1_bn	(None, 10, 10, 1536)	6,144	block14_sepconv1[0][0]

(BatchNormalization)			
block14_sepconv1_act (Activation)	(None, 10, 10, 1536)	0	block14_sepconv1_bn[0][0]
block14_sepconv2 (SeparableConv2D)	(None, 10, 10, 2048)	3,159,552	block14_sepconv1_act[0][0]
block14_sepconv2_bn (BatchNormalization)	(None, 10, 10, 2048)	8,192	block14_sepconv2[0][0]
block14_sepconv2_act (Activation)	(None, 10, 10, 2048)	0	block14_sepconv2_bn[0][0]
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 2048)	0	block14_sepconv2_act[0][0]
dense_4 (Dense)	(None, 1024)	2,098,176	global_average_pooling2d_...
dropout_2 (Dropout)	(None, 1024)	0	dense_4[0][0]
dense_5 (Dense)	(None, 2)	2,050	dropout_2[0][0]

Total params: 22,961,706 (87.59 MB)

Trainable params: 2,100,226 (8.01 MB)

Non-trainable params: 20,861,480 (79.58 MB)

Epoch 1/5
250/250 982s 4s/step - accuracy: 0.8985 - loss: 0.2363 - val_accuracy: 0.9370 - val_loss: 0.1766 - learning_rate: 0.0010
Epoch 2/5
250/250 984s 4s/step - accuracy: 0.9321 - loss: 0.1663 - val_accuracy: 0.9390 - val_loss: 0.1704 - learning_rate: 0.0010
Epoch 3/5
250/250 973s 4s/step - accuracy: 0.9351 - loss: 0.1547 - val_accuracy: 0.9250 - val_loss: 0.1850 - learning_rate: 0.0010
Epoch 4/5
250/250 984s 4s/step - accuracy: 0.9460 - loss: 0.1385 - val_accuracy: 0.9290 - val_loss: 0.2024 - learning_rate: 0.0010
Epoch 5/5
250/250 987s 4s/step - accuracy: 0.9566 - loss: 0.1097 - val_accuracy: 0.9430 - val_loss: 0.1637 - learning_rate: 1.0000e-04

Since, i have done one hot encoding for the numerical labels instead of just putting integers, this is the reason why i went for loss='categorical_crossentropy' instead of loss='binary_crossentropy'

Use all the standard performance measures to evaluate the performance of the applied transfer learning approach. Compare the performance measures for the two architectures used. Present your inference with justifications. (10 marks)

In [4]: # Evaluate the model on the test set
test_loss, test_accuracy = incept_model.evaluate(X_test, y_test)
print("Test Accuracy:", test_accuracy)

32/32 61s 2s/step - accuracy: 0.9130 - loss: 0.2782
Test Accuracy: 0.9350000023841858

In [5]: test_loss, test_accuracy = xcept_model.evaluate(X_test, y_test)
print("Test Accuracy:", test_accuracy)

32/32 107s 3s/step - accuracy: 0.9426 - loss: 0.1699
Test Accuracy: 0.9430000185966492

I'm getting good test accuracies for both the models(where Xception model outperforms InceptionV3 model), which means it can effectively generalize its predictions on new or unseen data.

Xception model outperforms because it has depthwise convolution which allows better feature representation compared to InceptionV3 model which uses pointwise convolution.

In [6]: incept_y_pred = incept_model.predict(X_test)
xcept_y_pred = xcept_model.predict(X_test)

32/32 65s 2s/step
32/32 115s 4s/step

In [7]: incept_labels = np.argmax(incept_y_pred, axis=1)
xcept_labels = np.argmax(xcept_y_pred, axis=1)
y_true_labels = np.argmax(y_test, axis=1)

Now plotting training accuracy vs validation accuracy & training loss vs validation loss for both the models

In [9]: fig, axes = plt.subplots(2, 2, figsize=(14, 10))

Plot training accuracy for Inception and Xception
axes[0, 0].plot(history_incept.history['accuracy'], label='Inception Training Accuracy')
axes[0, 0].plot(history_xcept.history['accuracy'], label='Xception Training Accuracy')
axes[0, 0].set_title('Training Accuracy')
axes[0, 0].set_xlabel('Epochs')
axes[0, 0].set_ylabel('Accuracy')
axes[0, 0].legend()

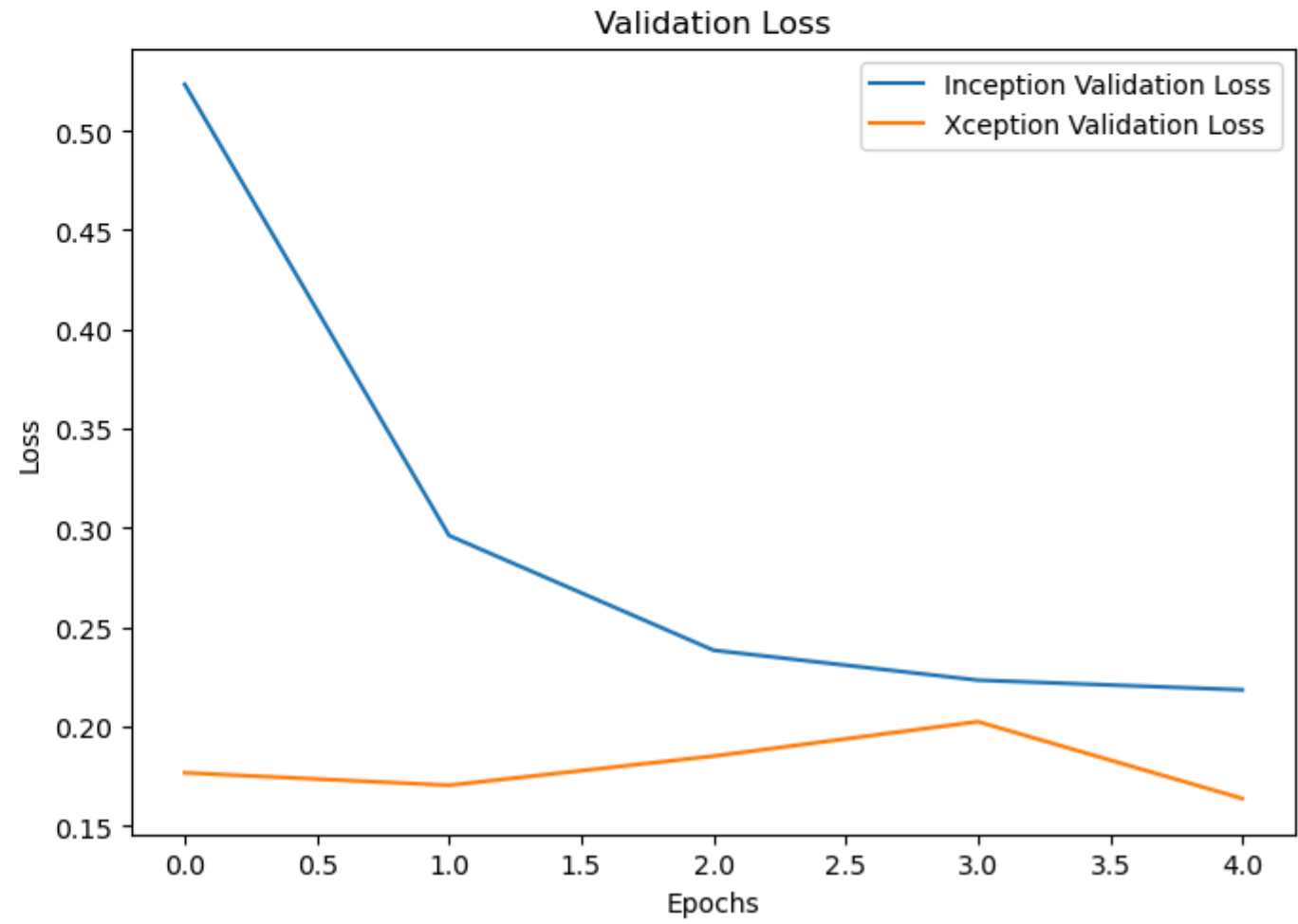
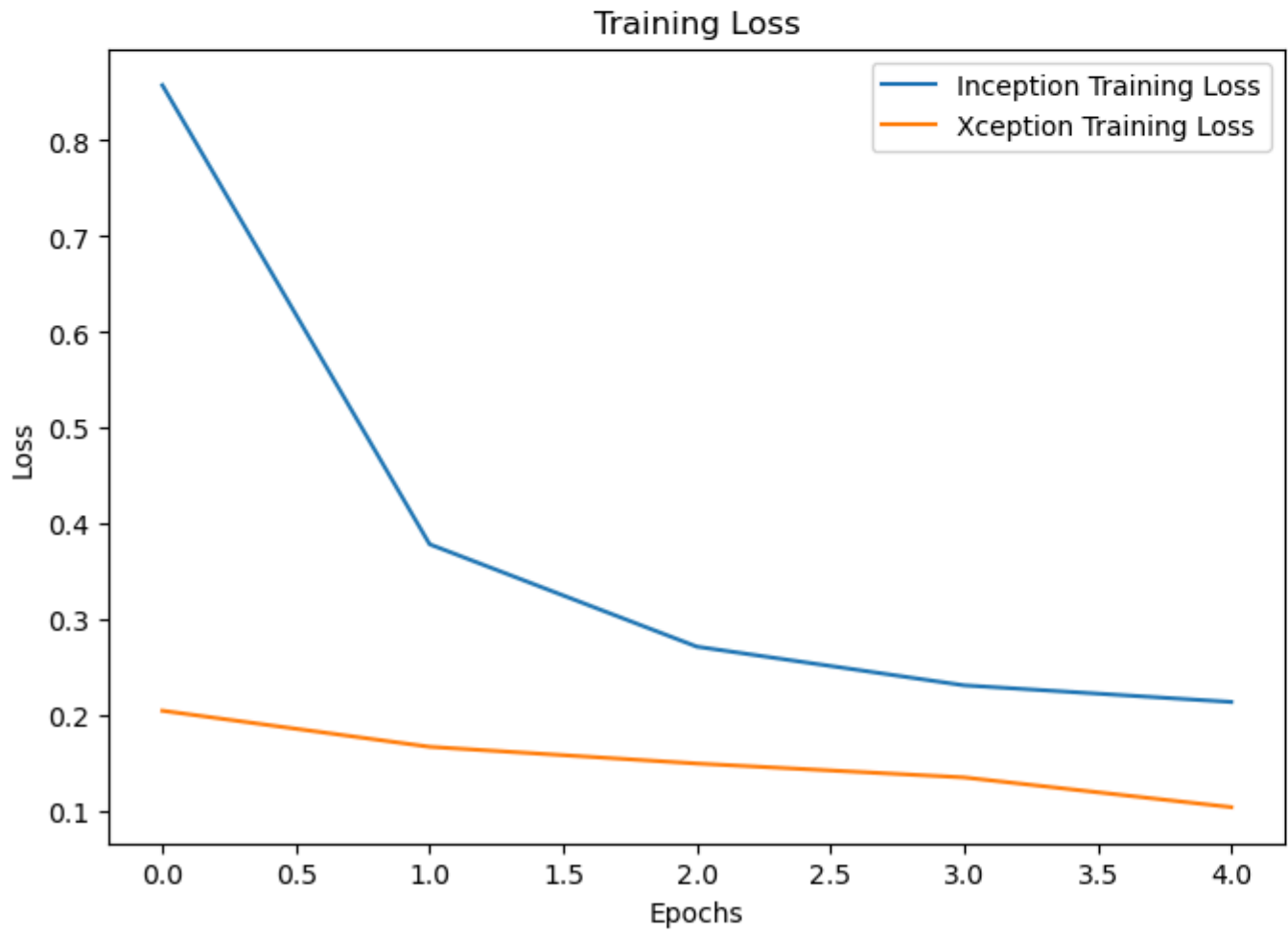
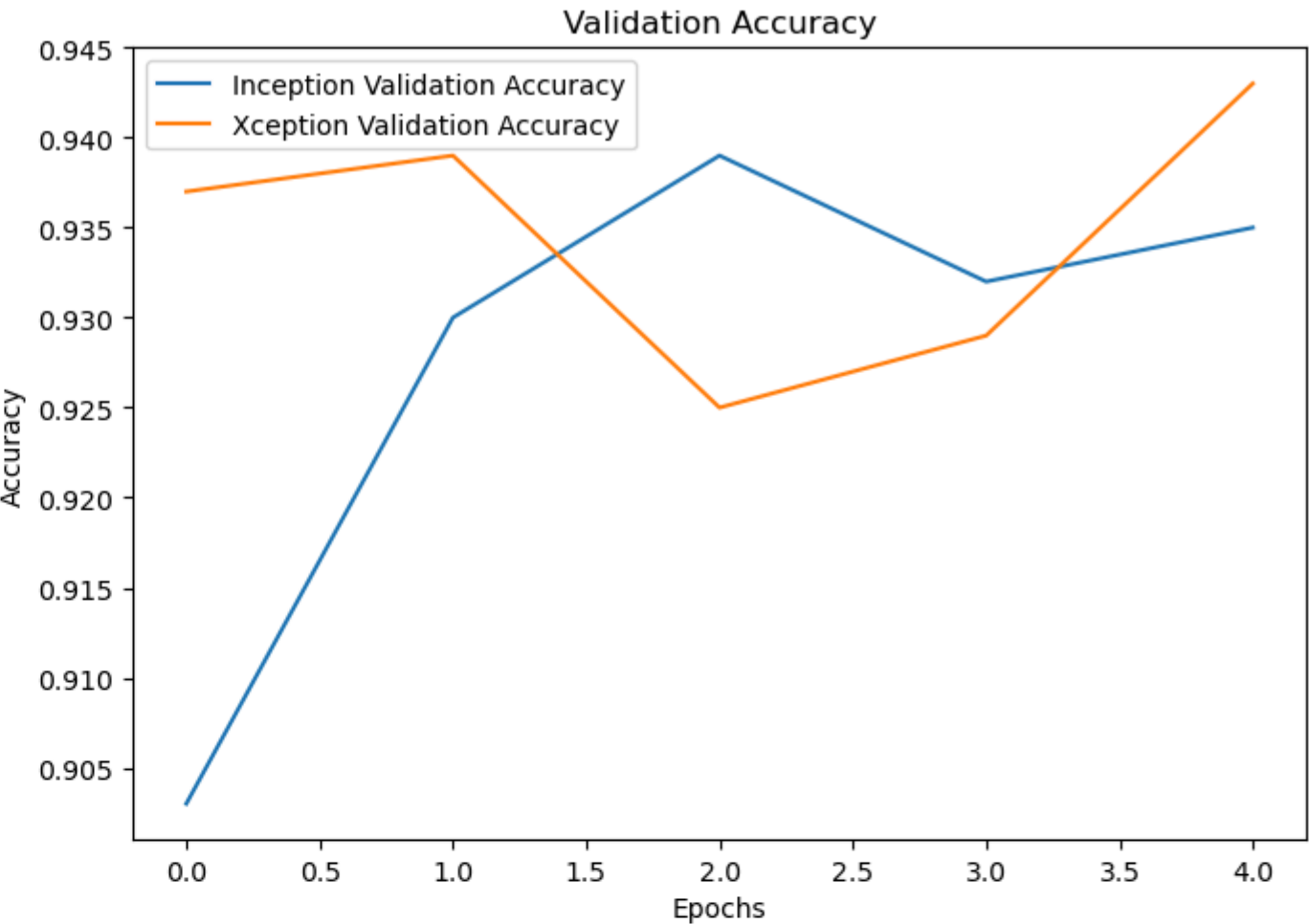
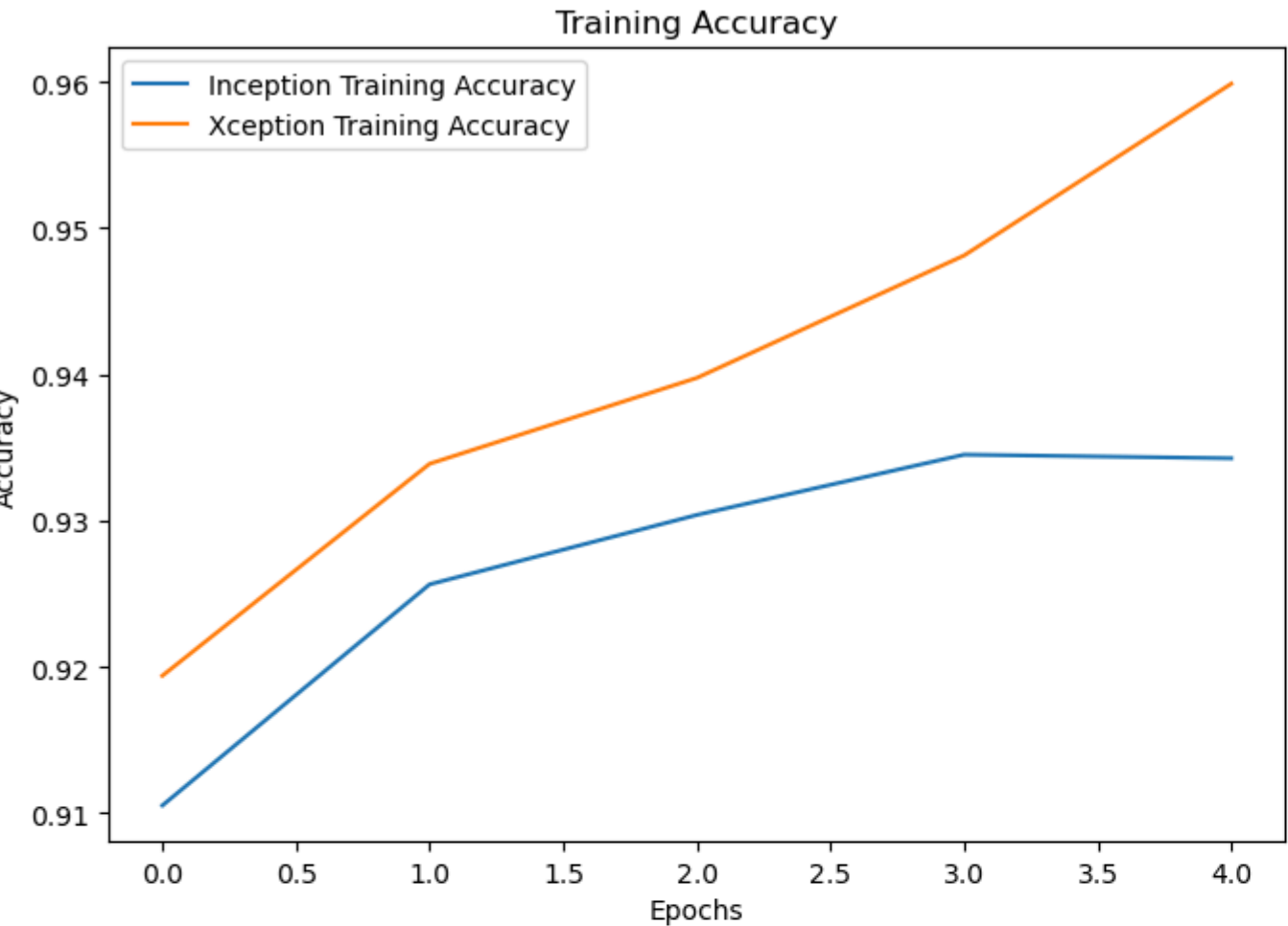
Plot validation accuracy for Inception and Xception
axes[0, 1].plot(history_incept.history['val_accuracy'], label='Inception Validation Accuracy')
axes[0, 1].plot(history_xcept.history['val_accuracy'], label='Xception Validation Accuracy')
axes[0, 1].set_title('Validation Accuracy')
axes[0, 1].set_xlabel('Epochs')
axes[0, 1].set_ylabel('Accuracy')


```
axes[0, 1].legend()

# Plot training loss for Inception and Xception
axes[1, 0].plot(history_incept.history['loss'], label='Inception Training Loss')
axes[1, 0].plot(history_xcept.history['loss'], label='Xception Training Loss')
axes[1, 0].set_title('Training Loss')
axes[1, 0].set_xlabel('Epochs')
axes[1, 0].set_ylabel('Loss')
axes[1, 0].legend()

# Plot validation loss for Inception and Xception
axes[1, 1].plot(history_incept.history['val_loss'], label='Inception Validation Loss')
axes[1, 1].plot(history_xcept.history['val_loss'], label='Xception Validation Loss')
axes[1, 1].set_title('Validation Loss')
axes[1, 1].set_xlabel('Epochs')
axes[1, 1].set_ylabel('Loss')
axes[1, 1].legend()

plt.tight_layout()
plt.show()
```



Inference:

Here, we can clearly see that Xception model outperforms InceptionV3 model in terms of:

- 1) Training Accuracy (Xception has deeper architecture which can capture more intricate patterns in the training data, leading to higher accuracy)
- 2) Validation Accuracy (Xception has efficient convolutional design which improves generalization, resulting in better performance on unseen validation data)
- 3) Training Loss (Deeper layers and optimized architecture enables Xception model to minimize training loss more effectively)
- 4) Validation Loss (Xception has superior generalization ability translates to lower loss on validation data compared to InceptionV3 model).

```
In [10]: incept_accuracy = accuracy_score(y_true_labels, incept_labels)
xcept_accuracy = accuracy_score(y_true_labels, xcept_labels)

incept_precision = precision_score(y_true_labels, incept_labels, average = 'macro')
xcept_precision = precision_score(y_true_labels, xcept_labels, average = 'macro')

incept_recall = recall_score(y_true_labels, incept_labels, average = 'macro')
xcept_recall = recall_score(y_true_labels, xcept_labels, average = 'macro')

incept_f1 = f1_score(y_true_labels, incept_labels, average = 'macro')
xcept_f1 = f1_score(y_true_labels, xcept_labels, average = 'macro')
```

```
In [11]: print("Inception Model:")
print(f"Accuracy: {incept_accuracy}")
print(f"Precision: {incept_precision}")
print(f"Recall: {incept_recall}")
print(f"F1-Score: {incept_f1}")

print("\nXception Model:")
print(f"Accuracy: {xcept_accuracy}")
print(f"Precision: {xcept_precision}")
print(f"Recall: {xcept_recall}")
print(f"F1-Score: {xcept_f1}")
```

Inception Model:
Accuracy: 0.935
Precision: 0.9376627401108343
Recall: 0.935
F1-Score: 0.9349009843972682

Xception Model:
Accuracy: 0.943
Precision: 0.9432996705773102
Recall: 0.9430000000000001
F1-Score: 0.9429903653717479

Inference:

It is clear that both the Inception and Xception models perform well, with the Xception model showing slight dominance across accuracy, precision, recall, and F1-score. This means that the Xception model may offer slightly better overall performance in terms of both classifying positive instances correctly and minimizing false positives, compared to our InceptionV3 model.

Extract the deep features and use at least 4 ML algorithms for classification. (5 marks)

```
In [13]: # Extract deep features from the InceptionV3 model
inception_features_model = Model(inputs=incept_model.input, outputs=incept_model.layers[-3].output)
X_train_inception_features = inception_features_model.predict(X_train)
X_test_inception_features = inception_features_model.predict(X_test)
```

250/250 465s 2s/step
32/32 62s 2s/step

```
In [14]: # Extract deep features from the Xception model
xception_features_model = Model(inputs=xcept_model.input, outputs=xcept_model.layers[-3].output)
X_train_xception_features = xception_features_model.predict(X_train)
X_test_xception_features = xception_features_model.predict(X_test)
```

250/250 880s 3s/step
32/32 110s 3s/step

```
In [15]: # Concatenating the deep features
X_train_deep_features = np.concatenate((X_train_inception_features, X_train_xception_features), axis=1)
X_test_deep_features = np.concatenate((X_test_inception_features, X_test_xception_features), axis=1)
```

1. SVM

```
In [25]: svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train_deep_features, np.argmax(y_train, axis=1))
svm_predictions = svm_classifier.predict(X_test_deep_features)
```

2. KNN

```
In [26]: knn_classifier = KNeighborsClassifier(n_neighbors=7)
knn_classifier.fit(X_train_deep_features, np.argmax(y_train, axis=1))
knn_predictions = knn_classifier.predict(X_test_deep_features)
```

3. Decision Tree

```
In [27]: dt_classifier = DecisionTreeClassifier(random_state=32)
dt_classifier.fit(X_train_deep_features, np.argmax(y_train, axis=1))
dt_predictions = dt_classifier.predict(X_test_deep_features)
```

4. Random Forest

```
In [28]: rf_classifier = RandomForestClassifier(n_estimators=100, random_state=32)
rf_classifier.fit(X_train_deep_features, np.argmax(y_train, axis=1))
```

```
rf_predictions = rf_classifier.predict(X_test_deep_features)
```

Use all the standard performance measures to evaluate the performance of the applied ML algorithms. Compare the performance measures and present your inference with justifications. (5 marks)

```
In [32]: def performance_measures(y_true, y_pred, ml_classifier):
accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred, average = 'macro')
recall = recall_score(y_true, y_pred, average = 'macro')
f1 = f1_score(y_true, y_pred, average = 'macro')
cm = confusion_matrix(y_true,y_pred)
print("Performance measures for", ml_classifier)
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1)
print("Confusion matrix:\n",cm)
print("\n")
performance_measures(np.argmax(y_test, axis=1), svm_predictions, "SVM Classifier")
performance_measures(np.argmax(y_test, axis=1), rf_predictions, "Random Forest Classifier")
performance_measures(np.argmax(y_test, axis=1), dt_predictions, "Decision Tree Classifier")
performance_measures(np.argmax(y_test, axis=1), knn_predictions, "K-Nearest Neighbors (KNN) Classifier")
```

```
Performance measures for SVM Classifier
Accuracy: 0.934
Precision: 0.9341111324499072
Recall: 0.9339999999999999
F1-score: 0.9339957757296466
Confusion matrix:
[[463  37]
 [ 29 471]]
```

```
Performance measures for Random Forest Classifier
Accuracy: 0.949
Precision: 0.9492174212318762
Recall: 0.949
F1-score: 0.9489938282532187
Confusion matrix:
[[469  31]
 [ 20 480]]
```

```
Performance measures for Decision Tree Classifier
Accuracy: 0.932
Precision: 0.932338953739732
Recall: 0.9319999999999999
F1-score: 0.9319866693871999
Confusion matrix:
[[459  41]
 [ 27 473]]
```

```
Performance measures for K-Nearest Neighbors (KNN) Classifier
Accuracy: 0.944
Precision: 0.9442558913934427
Recall: 0.944
F1-score: 0.9439919348386168
Confusion matrix:
[[466  34]
 [ 22 478]]
```

Inference:

Random Forest Classifier is the best performing ML classifier in overall terms, considering its higher accuracy, F1-score, and fewer misclassifications.

KNN Classifier also performs well and may be a good alternative if we are not bothered about computational efficiency.

Decision Tree and SVM Classifiers perform slightly lower compared to Random Forest and KNN.