

```
In [1]: # !pip3 install opencv-python
```

```
In [2]: # !pip install tensorflow-gpu
```

```
In [3]: # !pip3 install tensorflow
```

```
In [2]: # Import the libraries
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import pathlib
import random
import os
import tensorflow as tf
import pandas as pd
import seaborn as sns
```

```
In [3]: from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import LearningRateScheduler, EarlyStopping
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
```

Data Aquisition

```
In [4]: # Create a function to get an overview of contents inside the main directory
```

```
def dir_overview(directory_path):
    """
    Gives an overview of all the inside directories and files within
    it(directory_path) and gives the total number of overall images.

    Args:
        directory_path (str): Main(target) directory

    Returns:
        A print out of:
            number of subdirectories in the dir_path
            number of images in each subdir
            name of each subdir
            total number of overall images
    """
    image_count = 0
    for dirpath, dirnames, filenames in os.walk(directory_path):
        print(f"There are {len(dirnames)} directories and {len(filenames)} images in '{dirpath}'")
        image_count += len(filenames)
    print(f"\nTotal number of images in the {directory_path}: {image_count}")
```

```
In [5]: # Overview of dataset
dataset = "../Dataset/Reduced_Dataset/"
dir_overview(dataset)
```

```
There are 4 directories and 0 images in '../Dataset/Reduced_Dataset/'.
There are 0 directories and 1 images in '../Dataset/Reduced_Dataset/prediction_images'.
There are 1 directories and 0 images in '../Dataset/Reduced_Dataset/testing_set'.
There are 0 directories and 134 images in '../Dataset/Reduced_Dataset/testing_set\Prediction'.
There are 50 directories and 0 images in '../Dataset/Reduced_Dataset/training_set'.
There are 0 directories and 128 images in '../Dataset/Reduced_Dataset/training_set\ALBERT'.
```

S TOWHEE'.
There are 0 directories and 126 images in './Dataset/Reduced_Dataset/training_set\AMERICAN COOT'.
There are 0 directories and 120 images in './Dataset/Reduced_Dataset/training_set\ANTBIRD'.
There are 0 directories and 122 images in './Dataset/Reduced_Dataset/training_set\APOSTLEBIRD'.
There are 0 directories and 149 images in './Dataset/Reduced_Dataset/training_set\AUCLAND SHAQ'.
There are 0 directories and 132 images in './Dataset/Reduced_Dataset/training_set\AZURE TIT'.
There are 0 directories and 120 images in './Dataset/Reduced_Dataset/training_set\BAIKAL TEAL'.
There are 0 directories and 113 images in './Dataset/Reduced_Dataset/training_set\BEARDED BELLBIRD'.
There are 0 directories and 116 images in './Dataset/Reduced_Dataset/training_set\BLACK THROATED HUET'.
There are 0 directories and 109 images in './Dataset/Reduced_Dataset/training_set\BLACK-NECKED GREBE'.
There are 0 directories and 148 images in './Dataset/Reduced_Dataset/training_set\BLUE MALLOHA'.
There are 0 directories and 146 images in './Dataset/Reduced_Dataset/training_set\BREWER'S BLACKBIRD'.
There are 0 directories and 129 images in './Dataset/Reduced_Dataset/training_set\CINNAMON FLYCATCHER'.
There are 0 directories and 128 images in './Dataset/Reduced_Dataset/training_set\COMMON POORWILL'.
There are 0 directories and 121 images in './Dataset/Reduced_Dataset/training_set\CRESTED COUA'.
There are 0 directories and 115 images in './Dataset/Reduced_Dataset/training_set\EVENING GROSBEAK'.
There are 0 directories and 122 images in './Dataset/Reduced_Dataset/training_set\GAMBEL'S QUAIL'.
There are 0 directories and 130 images in './Dataset/Reduced_Dataset/training_set\GREAT JACAMAR'.
There are 0 directories and 127 images in './Dataset/Reduced_Dataset/training_set\GREY CUCKOOSHRIKE'.
There are 0 directories and 129 images in './Dataset/Reduced_Dataset/training_set\GUINEA TURACO'.
There are 0 directories and 120 images in './Dataset/Reduced_Dataset/training_set\HIMALAYAN BLUETAIL'.
There are 0 directories and 120 images in './Dataset/Reduced_Dataset/training_set\INDIAN VULTURE'.
There are 0 directories and 128 images in './Dataset/Reduced_Dataset/training_set\INLAND DOTTEREL'.
There are 0 directories and 124 images in './Dataset/Reduced_Dataset/training_set\IWI'.
There are 0 directories and 126 images in './Dataset/Reduced_Dataset/training_set\JAPANESE ROBIN'.
There are 0 directories and 140 images in './Dataset/Reduced_Dataset/training_set\LITTLE AUK'.
There are 0 directories and 109 images in './Dataset/Reduced_Dataset/training_set\MANGROVE CUCKOO'.
There are 0 directories and 122 images in './Dataset/Reduced_Dataset/training_set\MOURNING DOVE'.
There are 0 directories and 112 images in './Dataset/Reduced_Dataset/training_set\NORTHERN MOCKINGBIRD'.
There are 0 directories and 166 images in './Dataset/Reduced_Dataset/training_set\ORNATE HAWK EAGLE'.
There are 0 directories and 124 images in './Dataset/Reduced_Dataset/training_set\PEREGRINE FALCON'.
There are 0 directories and 160 images in './Dataset/Reduced_Dataset/training_set\PLUSH CRESTED JAY'.
There are 0 directories and 123 images in './Dataset/Reduced_Dataset/training_set\PURPLE SWAMPHEN'.
There are 0 directories and 106 images in './Dataset/Reduced_Dataset/training_set\RED HEADED WOODPECKER'.

There are 0 directories and 135 images in './Dataset/Reduced_Dataset/training_set\RED LE
GGED HONEYCREEPER'.

There are 0 directories and 137 images in './Dataset/Reduced_Dataset/training_set\RED SH
OULDERED HAWK'.

There are 0 directories and 135 images in './Dataset/Reduced_Dataset/training_set\SATYR
TRAGOPAN'.

There are 0 directories and 128 images in './Dataset/Reduced_Dataset/training_set\SNOWY
OWL'.

There are 0 directories and 120 images in './Dataset/Reduced_Dataset/training_set\SPOTTE
D CATBIRD'.

There are 0 directories and 147 images in './Dataset/Reduced_Dataset/training_set\SPOTTE
D WHISTLING DUCK'.

There are 0 directories and 128 images in './Dataset/Reduced_Dataset/training_set\STRIPP
ED SWALLOW'.

There are 0 directories and 148 images in './Dataset/Reduced_Dataset/training_set\SURF S
COTER'.

There are 0 directories and 108 images in './Dataset/Reduced_Dataset/training_set\TAIWAN
MAGPIE'.

There are 0 directories and 130 images in './Dataset/Reduced_Dataset/training_set\TURKEY
VULTURE'.

There are 0 directories and 154 images in './Dataset/Reduced_Dataset/training_set\VARIED
THRUSH'.

There are 0 directories and 121 images in './Dataset/Reduced_Dataset/training_set\VENEZU
ELIAN TROUPIAL'.

There are 0 directories and 140 images in './Dataset/Reduced_Dataset/training_set\VIOLET
CUCKOO'.

There are 0 directories and 130 images in './Dataset/Reduced_Dataset/training_set\VISAYA
N HORNBILL'.

There are 0 directories and 140 images in './Dataset/Reduced_Dataset/training_set\WHITE
EARED HUMMINGBIRD'.

There are 0 directories and 124 images in './Dataset/Reduced_Dataset/training_set\YELLOW
CACIQUE'.

There are 50 directories and 0 images in './Dataset/Reduced_Dataset/validation_set'.

There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\ALBER
TS TOWHEE'.

There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\AMERI
CAN COOT'.

There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\ANTBI
RD'.

There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\APOST
LEBIRD'.

There are 0 directories and 19 images in './Dataset/Reduced_Dataset/validation_set\AUCKL
AND SHAQ'.

There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\AZURE
TIT'.

There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\BAIKA
L TEAL'.

There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\BEARD
ED BELLBIRD'.

There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\BLACK
THROATED HUET'.

There are 0 directories and 14 images in './Dataset/Reduced_Dataset/validation_set\BLACK
-NECKED GREBE'.

There are 0 directories and 19 images in './Dataset/Reduced_Dataset/validation_set\BLUE
MALKOHA'.

There are 0 directories and 19 images in './Dataset/Reduced_Dataset/validation_set\BREWE
RS BLACKBIRD'.

There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\CINNA
MON FLYCATCHER'.

There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\COMMO
N POORWILL'.

There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\CREST
ED COUA'.

There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\EVENI
NG GROSBEAK'.

There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\GAMBE

LS QUAIL'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\GREAT JACAMAR'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\GREY CUCKOOSHRIKE'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\GUINEA TURACO'.
There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\HIMALAYAN BLUETAIL'.
There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\INDIAN VULTURE'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\INLAND DOTTEREL'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\IWI'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\JAPANESE ROBIN'.
There are 0 directories and 18 images in './Dataset/Reduced_Dataset/validation_set\LITTLE AUK'.
There are 0 directories and 14 images in './Dataset/Reduced_Dataset/validation_set\MANGROVE CUCKOO'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\MOURNING DOVE'.
There are 0 directories and 14 images in './Dataset/Reduced_Dataset/validation_set\NORTHERN MOCKINGBIRD'.
There are 0 directories and 21 images in './Dataset/Reduced_Dataset/validation_set\ORNATE HAWK EAGLE'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\PEREGRINE FALCON'.
There are 0 directories and 20 images in './Dataset/Reduced_Dataset/validation_set\PLUSH CRESTED JAY'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\PURPLE SWAMPHEN'.
There are 0 directories and 14 images in './Dataset/Reduced_Dataset/validation_set\RED HEADED WOODPECKER'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\RED LEGGED HONEYCREEPER'.
There are 0 directories and 18 images in './Dataset/Reduced_Dataset/validation_set\RED SHOULDERED HAWK'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\SATYR TRAGOPAN'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\SNOWY OWL'.
There are 0 directories and 15 images in './Dataset/Reduced_Dataset/validation_set\SPOTTED CATBIRD'.
There are 0 directories and 19 images in './Dataset/Reduced_Dataset/validation_set\SPOTTED WHISTLING DUCK'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\STRIPPED SWALLOW'.
There are 0 directories and 19 images in './Dataset/Reduced_Dataset/validation_set\SURF SCOTER'.
There are 0 directories and 14 images in './Dataset/Reduced_Dataset/validation_set\TAIWAN MAGPIE'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\TURKEY VULTURE'.
There are 0 directories and 20 images in './Dataset/Reduced_Dataset/validation_set\VARIED THRUSH'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\VENEZUELIAN TROUPIAL'.
There are 0 directories and 18 images in './Dataset/Reduced_Dataset/validation_set\VIOLET CUCKOO'.
There are 0 directories and 17 images in './Dataset/Reduced_Dataset/validation_set\VISAYAN HORNBILL'.
There are 0 directories and 18 images in './Dataset/Reduced_Dataset/validation_set\WHITE EARED HUMMINGBIRD'.
There are 0 directories and 16 images in './Dataset/Reduced_Dataset/validation_set\YELLOW CACIQUE'.

Total number of images in the ./Dataset/Reduced_Dataset/: 7400

```
In [6]: # Get the class names
main_directory = pathlib.Path("Dataset/Reduced_Dataset/training_set/")
class_names = np.array(sorted([item.name for item in main_directory.glob("*")]))
```

```
In [7]: class_names
```

```
Out[7]: array(['ALBERTS TOWHEE', 'AMERICAN COOT', 'ANTBIRD', 'APOSTLEBIRD',
              'AUCKLAND SHAQ', 'AZURE TIT', 'BAIKAL TEAL', 'BEARDED BELLBIRD',
              'BLACK THROATED HUET', 'BLACK-NECKED GREBE', 'BLUE MALKOHA',
              'BREWERS BLACKBIRD', 'CINNAMON FLYCATCHER', 'COMMON POORWILL',
              'CRESTED COUA', 'EVENING GROSBEAK', 'GAMBELS QUAIL',
              'GREAT JACAMAR', 'GREY CUCKOOSHRIKE', 'GUINEA TURACO',
              'HIMALAYAN BLUETAIL', 'INDIAN VULTURE', 'INLAND DOTTEREL', 'IWI',
              'JAPANESE ROBIN', 'LITTLE AUK', 'MANGROVE CUCKOO', 'MOURNING DOVE',
              'NORTHERN MOCKINGBIRD', 'ORNATE HAWK EAGLE', 'PEREGRINE FALCON',
              'PLUSH CRESTED JAY', 'PURPLE SWAMPHEN', 'RED HEADED WOODPECKER',
              'RED LEGGED HONEYCREEPER', 'RED SHOULDERED HAWK', 'SATYR TRAGOPAN',
              'SNOWY OWL', 'SPOTTED CATBIRD', 'SPOTTED WHISTLING DUCK',
              'STRIPPED SWALLOW', 'SURF SCOTER', 'TAIWAN MAGPIE',
              'TURKEY VULTURE', 'VARIED THRUSH', 'VENEZUELIAN TROUPIAL',
              'VIOLET CUCKOO', 'VISAYAN HORNBILL', 'WHITE EARED HUMMINGBIRD',
              'YELLOW CACIQUE'], dtype='<U23')
```

```
In [8]: len(class_names)
```

```
Out[8]: 50
```

```
In [9]: # Visualize image
def visualize_random_img(target_directory, target_class):
    """
    View's the random image from the given target directory and given target
    class.

    Args:
        target_directory (str): target directory
        target_class (str): Name of the subdirectory

    Returns:
        prints the name of the file
        prints the image shape
        Reutrn's a random image from the directory
    """
    # Setup the target directory
    target_folder = target_directory + target_class

    # Get a random image path
    random_image = random.sample(os.listdir(target_folder), 1)
    print(random_image)

    # Read and plot the image
    image = mpimg.imread(target_folder + "/" + random_image[0])
    plt.imshow(image)
    plt.title(target_class)
    plt.axis("off");

    print(f"Image shape: {image.shape}")

    return image
```

```
In [10]: # Visualize a random image from the dataset
image = visualize_random_img("./Dataset/Reduced_Dataset/training_set/", "ALBERTS TOWHEE")
```

['005.jpg']
Image shape: (224, 224, 3)

ALBERTS TOWHEE



```
In [62]: # Create a function to visualize random images(15) inside the directory
def random_images(dataset_dir):
    """
    Plots 15 random images for the given directory(dataset_dir).

    Args:
        dataset_dir: Main Directory that contains subdirectories and
                     image files inside subdirectories.

    Returns:
        Plots a random images from 15 random classes
    """
    # Path to your dataset directory containing bird images
    # dataset_dir = "./Dataset/train/"

    # Get a list of all labes in the dataset directory
    image_labels = [f for f in os.listdir(dataset_dir)]

    # Choose 15 random labels
    random_labels = random.choices(image_labels, k=min(15, len(image_labels)))

    # Plot the images
    fig, axes = plt.subplots(3, 5, figsize=(15, 9))

    for i, ax in enumerate(axes.flatten()):
        # Choose random image from the random labels
        random_image_path = dataset_dir + random_labels[i]
        random_image = dataset_dir + random_labels[i] + "/" + random.choice(os.listdir(r
        img = mpimg.imread(random_image)
        ax.imshow(img)
        ax.axis('off')
        ax.set_title(random_labels[i])

    plt.show()
```

```
In [63]: dataset_dir = "./Dataset/Reduced_Dataset/training_set/"
```

```
random_images(dataset_dir)
```



```
In [64]: # Create a function to visualize the number of images in random subdirectories
def plot_graph(main_dir):

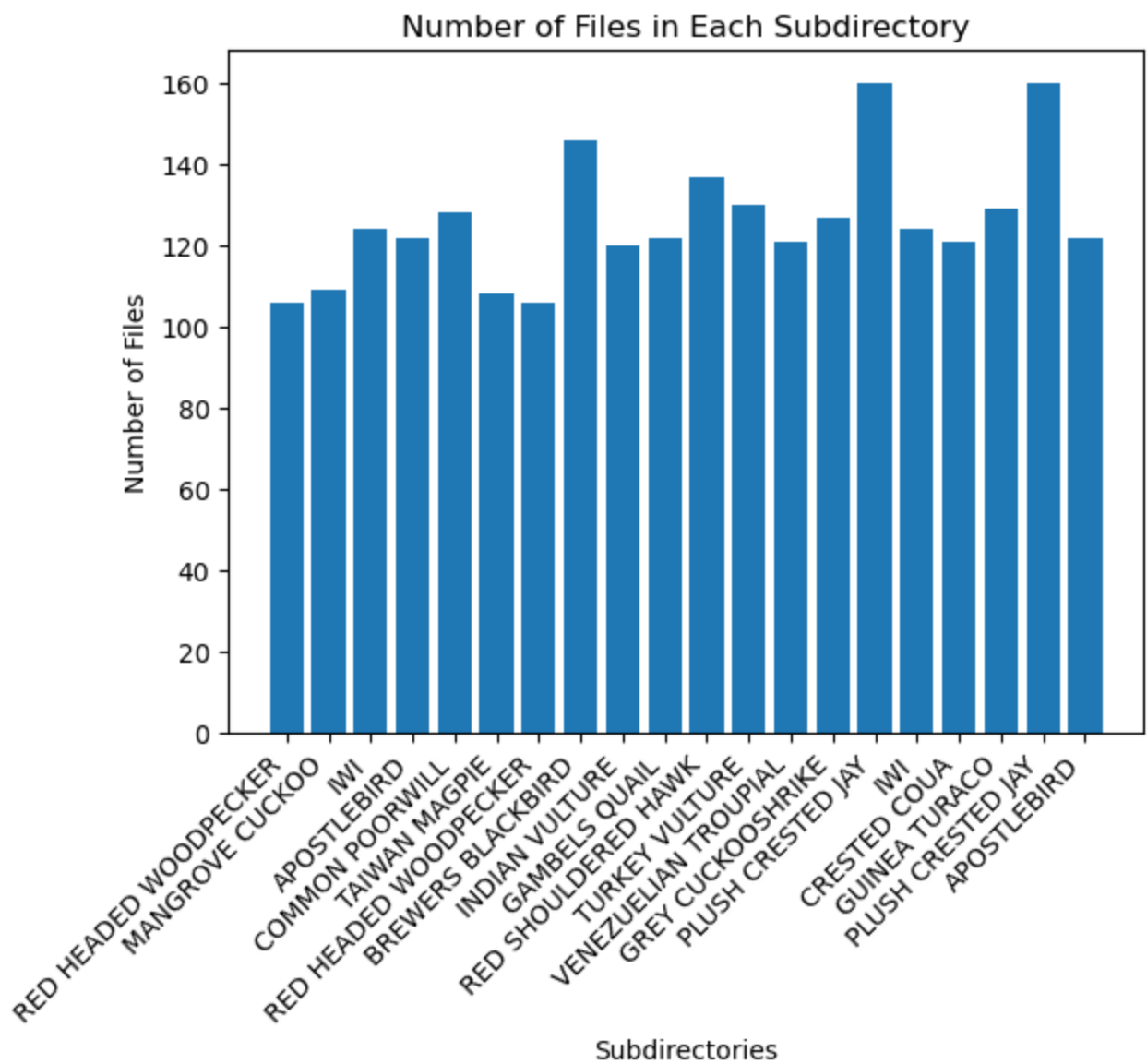
    # Get a list of subdirectories
    sub_dirs = [subdir for subdir in pathlib.Path(main_dir).iterdir() if subdir.is_dir()]

    # Get random 20 subdirectories
    random_subdirs = random.choices(sub_dirs, k=20)

    # Numbers of files in each random sub directory
    num_files = [len(list(random_subdir.glob("*"))) for random_subdir in random_subdirs]

    # Plot a bar chart
    plt.bar(range(len(random_subdirs)), num_files, tick_label=[subdir.name for subdir in
    plt.xlabel('Subdirectories')
    plt.ylabel('Number of Files')
    plt.title('Number of Files in Each Subdirectory')
    plt.xticks(rotation=45, ha='right') # Rotate x-axis labels for better visibility

In [65]: main_directory = "./Dataset/Reduced_Dataset/training_set/"
plot_graph(main_directory)
```

Building the model using CNN

```
In [66]: img_path = "../Dataset/Reduced_Dataset/training_set/ALBERTS TOWHEE/001.jpg"
img = mpimg.imread(img_path)
```

```
In [67]: img
```

```
Out[67]: array([[188, 175, 120],
 [184, 171, 116],
 [180, 169, 115],
 ...,
 [144, 127, 73],
 [145, 126, 68],
 [148, 129, 71]],

 [[188, 175, 120],
 [183, 170, 117],
 [179, 168, 114],
 ...,
 [145, 126, 70],
 [145, 126, 68],
 [149, 130, 71]],

 [[187, 174, 121],
 [181, 170, 116],
 [177, 166, 112],
```



```

.../
[143, 124, 68],
[145, 124, 67],
[150, 130, 71]],

...,

[[164, 133, 86],
 [166, 138, 91],
 [163, 136, 93],
 ...,
 [ 78,  69, 60],
 [ 65,  56, 47],
 [ 65,  56, 47]],

[[149, 115, 70],
 [154, 121, 78],
 [151, 119, 78],
 ...,
 [ 86,  78, 67],
 [ 60,  52, 41],
 [ 59,  51, 40]],

[[158, 120, 81],
 [167, 132, 92],
 [165, 131, 93],
 ...,
 [ 46,  38, 27],
 [ 39,  31, 20],
 [ 55,  47, 36]]], dtype=uint8)

```

```

In [73]: # Rescale the images using ImageDataGenerator
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Set the seed
tf.random.set_seed(42)

# Setup paths to our data directories
train_dir_path = "./Dataset/Reduced_Dataset/training_set/"
test_dir_path = "./Dataset/Reduced_Dataset/testing_set/"

# Preprocess data (get all the pixel values between 0 and 1(sclaing/Normalizing))

# Initialize the ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1 / 255.0,
    rotation_range=20,
    zoom_range=0.05,
    width_shift_range=0.05,
    height_shift_range=0.05,
    shear_range=0.05,
    horizontal_flip=True,
    fill_mode="nearest",
    validation_split=0.20)

test_datagen = ImageDataGenerator(rescale=1 / 255.0)

# Create a generator for reading images and scaling them
batch_size = 8
train_generator = train_datagen.flow_from_directory(
    directory=train_dir_path,
    target_size=(224, 224),
    color_mode="rgb",
    batch_size=batch_size,
    class_mode="categorical",

```

```

        subset='training',
        shuffle=True,
        seed=42
    )
    valid_generator = train_datagen.flow_from_directory(
        directory=train_dir_path,
        target_size=(224, 224),
        color_mode="rgb",
        batch_size=batch_size,
        class_mode="categorical",
        subset='validation',
        shuffle=True,
        seed=42
    )
    test_generator = test_datagen.flow_from_directory(
        directory=test_dir_path,
        target_size=(224, 224),
        color_mode="rgb",
        batch_size=1,
        class_mode=None,
        shuffle=False,
        seed=42
    )

```

Found 5166 images belonging to 50 classes.
 Found 1269 images belonging to 50 classes.
 Found 134 images belonging to 1 classes.

In []:

In [80]:

```

# Build a CNN model
model_1 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(50, activation='softmax')
])

# Compile our CNN
model_1.compile(loss="categorical_crossentropy",
                optimizer=tf.keras.optimizers.Adam(),
                metrics=["accuracy"])

# Fit the model
history_1 = model_1.fit(train_generator,
                        epochs=100,
                        steps_per_epoch=len(train_generator),
                        validation_data=valid_generator,
                        validation_steps=len(valid_generator))

```

```

Epoch 1/100
646/646 [=====] - 337s 520ms/step - loss: 4.5612 - accuracy: 0.0312 - val_loss: 3.7792 - val_accuracy: 0.0433
Epoch 2/100
646/646 [=====] - 335s 519ms/step - loss: 3.6612 - accuracy: 0.0925 - val_loss: 3.5144 - val_accuracy: 0.1182
Epoch 3/100
646/646 [=====] - 335s 518ms/step - loss: 3.3310 - accuracy: 0.1601 - val_loss: 3.2254 - val_accuracy: 0.1883
Epoch 4/100
646/646 [=====] - 333s 515ms/step - loss: 2.9799 - accuracy: 0.2313 - val_loss: 2.8458 - val_accuracy: 0.2656
Epoch 5/100
646/646 [=====] - 332s 514ms/step - loss: 2.6895 - accuracy: 0.3117 - val_loss: 2.6952 - val_accuracy: 0.3160

```

Epoch 6/100
646/646 [=====] - 333s 516ms/step - loss: 2.4690 - accuracy: 0.3469 - val_loss: 2.6347 - val_accuracy: 0.3483
Epoch 7/100
646/646 [=====] - 347s 538ms/step - loss: 2.3603 - accuracy: 0.3748 - val_loss: 2.4763 - val_accuracy: 0.3696
Epoch 8/100
646/646 [=====] - 393s 608ms/step - loss: 2.2754 - accuracy: 0.4100 - val_loss: 2.7034 - val_accuracy: 0.3325
Epoch 9/100
646/646 [=====] - 401s 621ms/step - loss: 2.1810 - accuracy: 0.4183 - val_loss: 2.5066 - val_accuracy: 0.3672
Epoch 10/100
646/646 [=====] - 339s 524ms/step - loss: 2.0968 - accuracy: 0.4373 - val_loss: 2.4558 - val_accuracy: 0.3759
Epoch 11/100
646/646 [=====] - 338s 524ms/step - loss: 2.0275 - accuracy: 0.4574 - val_loss: 2.3709 - val_accuracy: 0.4129
Epoch 12/100
646/646 [=====] - 337s 522ms/step - loss: 1.9876 - accuracy: 0.4617 - val_loss: 2.3579 - val_accuracy: 0.4137
Epoch 13/100
646/646 [=====] - 334s 516ms/step - loss: 1.9353 - accuracy: 0.4789 - val_loss: 2.4112 - val_accuracy: 0.3901
Epoch 14/100
646/646 [=====] - 336s 520ms/step - loss: 1.9079 - accuracy: 0.4830 - val_loss: 2.1863 - val_accuracy: 0.4358
Epoch 15/100
646/646 [=====] - 337s 521ms/step - loss: 1.8548 - accuracy: 0.4899 - val_loss: 2.4413 - val_accuracy: 0.4019
Epoch 16/100
646/646 [=====] - 337s 522ms/step - loss: 1.8324 - accuracy: 0.4934 - val_loss: 2.3042 - val_accuracy: 0.4452
Epoch 17/100
646/646 [=====] - 348s 539ms/step - loss: 1.8190 - accuracy: 0.5002 - val_loss: 2.3159 - val_accuracy: 0.4232
Epoch 18/100
646/646 [=====] - 357s 553ms/step - loss: 1.7772 - accuracy: 0.5141 - val_loss: 2.4207 - val_accuracy: 0.4161
Epoch 19/100
646/646 [=====] - 343s 531ms/step - loss: 1.7874 - accuracy: 0.5122 - val_loss: 2.3004 - val_accuracy: 0.4374
Epoch 20/100
646/646 [=====] - 338s 524ms/step - loss: 1.7354 - accuracy: 0.5354 - val_loss: 2.4158 - val_accuracy: 0.4137
Epoch 21/100
646/646 [=====] - 338s 523ms/step - loss: 1.7018 - accuracy: 0.5341 - val_loss: 2.2382 - val_accuracy: 0.4413
Epoch 22/100
646/646 [=====] - 341s 528ms/step - loss: 1.7168 - accuracy: 0.5346 - val_loss: 2.3354 - val_accuracy: 0.4476
Epoch 23/100
646/646 [=====] - 366s 567ms/step - loss: 1.6533 - accuracy: 0.5412 - val_loss: 2.3798 - val_accuracy: 0.4208
Epoch 24/100
646/646 [=====] - 360s 557ms/step - loss: 1.6663 - accuracy: 0.5443 - val_loss: 2.4808 - val_accuracy: 0.4358
Epoch 25/100
646/646 [=====] - 360s 557ms/step - loss: 1.6619 - accuracy: 0.5449 - val_loss: 2.2410 - val_accuracy: 0.4799
Epoch 26/100
646/646 [=====] - 352s 545ms/step - loss: 1.5728 - accuracy: 0.5681 - val_loss: 2.3349 - val_accuracy: 0.4618
Epoch 27/100
646/646 [=====] - 357s 553ms/step - loss: 1.6099 - accuracy: 0.5523 - val_loss: 2.3523 - val_accuracy: 0.4381

Epoch 28/100
646/646 [=====] - 352s 544ms/step - loss: 1.5843 - accuracy: 0.5623 - val_loss: 2.4175 - val_accuracy: 0.4594
Epoch 29/100
646/646 [=====] - 354s 547ms/step - loss: 1.5586 - accuracy: 0.5654 - val_loss: 2.2442 - val_accuracy: 0.4728
Epoch 30/100
646/646 [=====] - 346s 536ms/step - loss: 1.5597 - accuracy: 0.5697 - val_loss: 2.2992 - val_accuracy: 0.4641
Epoch 31/100
646/646 [=====] - 373s 577ms/step - loss: 1.5546 - accuracy: 0.5681 - val_loss: 2.2616 - val_accuracy: 0.4657
Epoch 32/100
646/646 [=====] - 375s 580ms/step - loss: 1.5240 - accuracy: 0.5730 - val_loss: 2.4391 - val_accuracy: 0.4720
Epoch 33/100
646/646 [=====] - 330s 511ms/step - loss: 1.5269 - accuracy: 0.5823 - val_loss: 2.3660 - val_accuracy: 0.4626
Epoch 34/100
646/646 [=====] - 338s 523ms/step - loss: 1.5142 - accuracy: 0.5778 - val_loss: 2.2688 - val_accuracy: 0.4728
Epoch 35/100
646/646 [=====] - 381s 590ms/step - loss: 1.5024 - accuracy: 0.5863 - val_loss: 2.4370 - val_accuracy: 0.4571
Epoch 36/100
646/646 [=====] - 331s 512ms/step - loss: 1.4749 - accuracy: 0.5881 - val_loss: 2.3506 - val_accuracy: 0.5051
Epoch 37/100
646/646 [=====] - 337s 522ms/step - loss: 1.4649 - accuracy: 0.5902 - val_loss: 2.5008 - val_accuracy: 0.4563
Epoch 38/100
646/646 [=====] - 396s 612ms/step - loss: 1.4479 - accuracy: 0.6003 - val_loss: 2.4923 - val_accuracy: 0.4807
Epoch 39/100
646/646 [=====] - 353s 546ms/step - loss: 1.4844 - accuracy: 0.5838 - val_loss: 2.3033 - val_accuracy: 0.4807
Epoch 40/100
646/646 [=====] - 395s 611ms/step - loss: 1.4396 - accuracy: 0.6009 - val_loss: 2.4143 - val_accuracy: 0.4657
Epoch 41/100
646/646 [=====] - 345s 534ms/step - loss: 1.3956 - accuracy: 0.6117 - val_loss: 2.5166 - val_accuracy: 0.4610
Epoch 42/100
646/646 [=====] - 380s 588ms/step - loss: 1.4375 - accuracy: 0.5985 - val_loss: 2.6033 - val_accuracy: 0.4468
Epoch 43/100
646/646 [=====] - 356s 550ms/step - loss: 1.4058 - accuracy: 0.6067 - val_loss: 2.2988 - val_accuracy: 0.4862
Epoch 44/100
646/646 [=====] - 373s 577ms/step - loss: 1.3988 - accuracy: 0.6078 - val_loss: 2.4235 - val_accuracy: 0.4933
Epoch 45/100
646/646 [=====] - 351s 542ms/step - loss: 1.4163 - accuracy: 0.6113 - val_loss: 2.4425 - val_accuracy: 0.4484
Epoch 46/100
646/646 [=====] - 344s 533ms/step - loss: 1.4013 - accuracy: 0.6057 - val_loss: 2.3553 - val_accuracy: 0.4878
Epoch 47/100
646/646 [=====] - 343s 530ms/step - loss: 1.3758 - accuracy: 0.6136 - val_loss: 2.4766 - val_accuracy: 0.4602
Epoch 48/100
646/646 [=====] - 381s 590ms/step - loss: 1.4194 - accuracy: 0.6094 - val_loss: 2.5981 - val_accuracy: 0.4523
Epoch 49/100
646/646 [=====] - 360s 557ms/step - loss: 1.3663 - accuracy: 0.6214 - val_loss: 2.5174 - val_accuracy: 0.4492

Epoch 50/100
646/646 [=====] - 384s 595ms/step - loss: 1.3417 - accuracy: 0.6171 - val_loss: 2.6112 - val_accuracy: 0.4799
Epoch 51/100
646/646 [=====] - 361s 558ms/step - loss: 1.3808 - accuracy: 0.6181 - val_loss: 2.4544 - val_accuracy: 0.4886
Epoch 52/100
646/646 [=====] - 351s 544ms/step - loss: 1.3827 - accuracy: 0.6243 - val_loss: 2.4339 - val_accuracy: 0.4736
Epoch 53/100
646/646 [=====] - 458s 709ms/step - loss: 1.3303 - accuracy: 0.6293 - val_loss: 2.4637 - val_accuracy: 0.4752
Epoch 54/100
646/646 [=====] - 370s 572ms/step - loss: 1.3038 - accuracy: 0.6359 - val_loss: 2.5323 - val_accuracy: 0.4799
Epoch 55/100
646/646 [=====] - 354s 547ms/step - loss: 1.3130 - accuracy: 0.6324 - val_loss: 2.5917 - val_accuracy: 0.4578
Epoch 56/100
646/646 [=====] - 351s 544ms/step - loss: 1.3451 - accuracy: 0.6245 - val_loss: 2.4303 - val_accuracy: 0.4894
Epoch 57/100
646/646 [=====] - 346s 536ms/step - loss: 1.3499 - accuracy: 0.6314 - val_loss: 2.5640 - val_accuracy: 0.4736
Epoch 58/100
646/646 [=====] - 354s 548ms/step - loss: 1.3173 - accuracy: 0.6295 - val_loss: 2.4408 - val_accuracy: 0.4894
Epoch 59/100
646/646 [=====] - 359s 555ms/step - loss: 1.3140 - accuracy: 0.6291 - val_loss: 2.5156 - val_accuracy: 0.4862
Epoch 60/100
646/646 [=====] - 351s 544ms/step - loss: 1.2776 - accuracy: 0.6367 - val_loss: 2.6819 - val_accuracy: 0.4760
Epoch 61/100
646/646 [=====] - 346s 535ms/step - loss: 1.3275 - accuracy: 0.6295 - val_loss: 2.4878 - val_accuracy: 0.4831
Epoch 62/100
646/646 [=====] - 353s 546ms/step - loss: 1.2961 - accuracy: 0.6432 - val_loss: 2.5856 - val_accuracy: 0.4862
Epoch 63/100
646/646 [=====] - 348s 539ms/step - loss: 1.2376 - accuracy: 0.6510 - val_loss: 2.5156 - val_accuracy: 0.4965
Epoch 64/100
646/646 [=====] - 357s 552ms/step - loss: 1.2759 - accuracy: 0.6409 - val_loss: 2.7285 - val_accuracy: 0.4736
Epoch 65/100
646/646 [=====] - 352s 545ms/step - loss: 1.2672 - accuracy: 0.6473 - val_loss: 2.8806 - val_accuracy: 0.4350
Epoch 66/100
646/646 [=====] - 369s 571ms/step - loss: 1.3059 - accuracy: 0.6332 - val_loss: 2.6697 - val_accuracy: 0.4894
Epoch 67/100
646/646 [=====] - 348s 539ms/step - loss: 1.2771 - accuracy: 0.6388 - val_loss: 2.4424 - val_accuracy: 0.4957
Epoch 68/100
646/646 [=====] - 370s 573ms/step - loss: 1.2648 - accuracy: 0.6469 - val_loss: 2.5804 - val_accuracy: 0.4744
Epoch 69/100
646/646 [=====] - 356s 550ms/step - loss: 1.2347 - accuracy: 0.6554 - val_loss: 2.9102 - val_accuracy: 0.4712
Epoch 70/100
646/646 [=====] - 334s 517ms/step - loss: 1.2791 - accuracy: 0.6400 - val_loss: 2.5681 - val_accuracy: 0.4768
Epoch 71/100
646/646 [=====] - 433s 670ms/step - loss: 1.2214 - accuracy: 0.6537 - val_loss: 2.7194 - val_accuracy: 0.4720

Epoch 72/100
646/646 [=====] - 370s 572ms/step - loss: 1.2226 - accuracy: 0.
6616 - val_loss: 2.7020 - val_accuracy: 0.4854
Epoch 73/100
646/646 [=====] - 409s 634ms/step - loss: 1.2277 - accuracy: 0.
6570 - val_loss: 2.8447 - val_accuracy: 0.4807
Epoch 74/100
646/646 [=====] - 422s 654ms/step - loss: 1.2188 - accuracy: 0.
6549 - val_loss: 2.5851 - val_accuracy: 0.4925
Epoch 75/100
646/646 [=====] - 390s 604ms/step - loss: 1.2054 - accuracy: 0.
6611 - val_loss: 2.9134 - val_accuracy: 0.4870
Epoch 76/100
646/646 [=====] - 349s 540ms/step - loss: 1.2213 - accuracy: 0.
6581 - val_loss: 2.8533 - val_accuracy: 0.4815
Epoch 77/100
646/646 [=====] - 360s 558ms/step - loss: 1.2373 - accuracy: 0.
6514 - val_loss: 3.0286 - val_accuracy: 0.4673
Epoch 78/100
646/646 [=====] - 356s 552ms/step - loss: 1.2031 - accuracy: 0.
6657 - val_loss: 2.8186 - val_accuracy: 0.4752
Epoch 79/100
646/646 [=====] - 367s 569ms/step - loss: 1.1996 - accuracy: 0.
6558 - val_loss: 2.9501 - val_accuracy: 0.4768
Epoch 80/100
646/646 [=====] - 359s 555ms/step - loss: 1.1724 - accuracy: 0.
6676 - val_loss: 2.8772 - val_accuracy: 0.4807
Epoch 81/100
646/646 [=====] - 355s 549ms/step - loss: 1.2139 - accuracy: 0.
6659 - val_loss: 2.9673 - val_accuracy: 0.4768
Epoch 82/100
646/646 [=====] - 329s 510ms/step - loss: 1.1882 - accuracy: 0.
6674 - val_loss: 2.6927 - val_accuracy: 0.5051
Epoch 83/100
646/646 [=====] - 328s 507ms/step - loss: 1.2197 - accuracy: 0.
6626 - val_loss: 2.8116 - val_accuracy: 0.4807
Epoch 84/100
646/646 [=====] - 325s 502ms/step - loss: 1.1916 - accuracy: 0.
6638 - val_loss: 2.9544 - val_accuracy: 0.4815
Epoch 85/100
646/646 [=====] - 324s 501ms/step - loss: 1.1405 - accuracy: 0.
6779 - val_loss: 2.9600 - val_accuracy: 0.4846
Epoch 86/100
646/646 [=====] - 333s 515ms/step - loss: 1.1750 - accuracy: 0.
6711 - val_loss: 2.7865 - val_accuracy: 0.5083
Epoch 87/100
646/646 [=====] - 350s 541ms/step - loss: 1.1826 - accuracy: 0.
6758 - val_loss: 2.8091 - val_accuracy: 0.4901
Epoch 88/100
646/646 [=====] - 356s 551ms/step - loss: 1.2056 - accuracy: 0.
6678 - val_loss: 3.0862 - val_accuracy: 0.4531
Epoch 89/100
646/646 [=====] - 347s 537ms/step - loss: 1.2149 - accuracy: 0.
6678 - val_loss: 2.7688 - val_accuracy: 0.4988
Epoch 90/100
646/646 [=====] - 364s 563ms/step - loss: 1.1789 - accuracy: 0.
6713 - val_loss: 2.8321 - val_accuracy: 0.4775
Epoch 91/100
646/646 [=====] - 362s 560ms/step - loss: 1.1152 - accuracy: 0.
6831 - val_loss: 3.0160 - val_accuracy: 0.5028
Epoch 92/100
646/646 [=====] - 358s 553ms/step - loss: 1.1772 - accuracy: 0.
6628 - val_loss: 2.9254 - val_accuracy: 0.4925
Epoch 93/100
646/646 [=====] - 331s 512ms/step - loss: 1.1502 - accuracy: 0.
6721 - val_loss: 2.7959 - val_accuracy: 0.4831

```

Epoch 94/100
646/646 [=====] - 321s 497ms/step - loss: 1.1313 - accuracy: 0.6756 - val_loss: 2.7201 - val_accuracy: 0.4980
Epoch 95/100
646/646 [=====] - 325s 503ms/step - loss: 1.1376 - accuracy: 0.6775 - val_loss: 3.0831 - val_accuracy: 0.4752
Epoch 96/100
646/646 [=====] - 328s 508ms/step - loss: 1.2144 - accuracy: 0.6651 - val_loss: 2.6498 - val_accuracy: 0.5035
Epoch 97/100
646/646 [=====] - 339s 525ms/step - loss: 1.1699 - accuracy: 0.6700 - val_loss: 3.1417 - val_accuracy: 0.4831
Epoch 98/100
646/646 [=====] - 349s 541ms/step - loss: 1.1833 - accuracy: 0.6682 - val_loss: 2.7653 - val_accuracy: 0.4823
Epoch 99/100
646/646 [=====] - 335s 519ms/step - loss: 1.1577 - accuracy: 0.6791 - val_loss: 2.9359 - val_accuracy: 0.4941
Epoch 100/100
646/646 [=====] - 366s 566ms/step - loss: 1.0809 - accuracy: 0.7011 - val_loss: 2.8779 - val_accuracy: 0.5067

```

```
In [81]: model_1.summary()
```

```
Model: "sequential_6"
```

Layer (type)	Output Shape	Param #
=====		
conv2d_6 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 111, 111, 32)	0
flatten_6 (Flatten)	(None, 394272)	0
dense_7 (Dense)	(None, 128)	50466944
dense_8 (Dense)	(None, 50)	6450
=====		
Total params: 50,474,290		
Trainable params: 50,474,290		
Non-trainable params: 0		

Evaluating our model

```
In [84]: score = model_1.evaluate_generator(valid_generator)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```

C:\Users\Sam\AppData\Local\Temp\ipykernel_20500\488157943.py:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.
  score = model_1.evaluate_generator(valid_generator)
Test loss: 2.8711252212524414
Test accuracy: 0.49566587805747986

```

```
In [86]: predict=model_1.predict_generator(test_generator)
# predict the class label
y_classes = predict.argmax(axis=-1)
```

```

C:\Users\Sam\AppData\Local\Temp\ipykernel_20500\3170131605.py:1: UserWarning: `Model.predict_generator` is deprecated and will be removed in a future version. Please use `Model.predict`

```



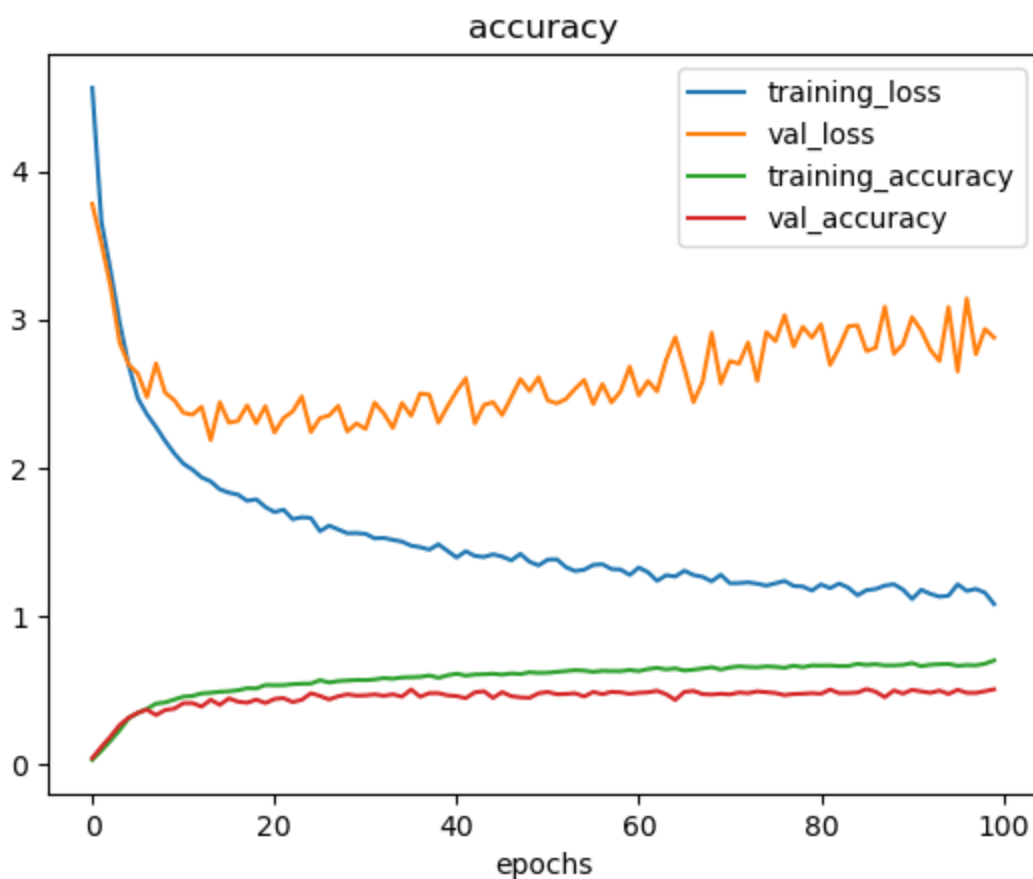
```
l.predict`, which supports generators.  
predict=model_1.predict_generator(test_generator)
```

```
In [87]: y_classes
```

```
Out[87]: array([42, 47, 19, 33, 16, 31, 49, 46, 38, 28, 48, 12, 18, 49, 26, 31, 40,  
         14, 45, 37, 35, 46, 45, 42, 24, 38,  1, 43, 48, 39, 48, 42, 10, 41,  
         45, 44, 25, 32, 15, 44, 29, 45, 33,  6, 23, 31, 26, 48, 49, 42, 31,  
         45, 41, 44, 19, 44, 16, 26, 30,  5, 47, 40, 42, 38,  7, 46, 45, 48,  
         18, 49, 44, 19, 42, 35, 41, 29, 49, 46, 46,  7, 49, 42, 19, 24, 12,  
         47, 49, 43, 48, 35, 48, 49, 44, 39, 14,  1, 49, 47, 47,  4, 14,  4,  
         12, 44, 18, 36, 44, 28, 26,  5, 29, 41, 29, 41,  2, 48, 22, 41, 44,  
         24, 11, 39, 41, 18, 31, 29, 21, 47, 29,  2,  4,  4, 47, 47],  
        dtype=int64)
```

```
In [142... # Plot the Validation and training curves  
def plot_loss_curves(history):  
    """  
    Returns separate loss curves for training and validation metrics.  
    """  
    loss = history.history["loss"]  
    val_loss = history.history["val_loss"]  
  
    accuracy = history.history["accuracy"]  
    val_accuracy = history.history["val_accuracy"]  
  
    epochs = range(len(history.history["loss"]))  
  
    # Plot loss  
    plt.plot(epochs, loss, label="training_loss")  
    plt.plot(epochs, val_loss, label="val_loss")  
    plt.title("loss")  
    plt.xlabel("epochs")  
    plt.legend()  
  
    # Plot accuracy  
    plt.plot(epochs, accuracy, label="training_accuracy")  
    plt.plot(epochs, val_accuracy, label="val_accuracy")  
    plt.title("accuracy")  
    plt.xlabel("epochs")  
    plt.legend()
```

```
In [143... plot_loss_curves(history_1)
```



Improving our model

```
In [94]: # Augment the data for improving model performance

# Create an ImageDataGenerator for training data with data augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

# Create ImageDataGenerators for validation and testing data without data augmentation
validation_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# Define the paths to your datasets
train_dir = './Dataset/Reduced_Dataset/training_set/'
validation_dir = './Dataset/Reduced_Dataset/validation_set/'
test_dir = './Dataset/Reduced_Dataset/testing_set/'

# Create generators for training, validation, and testing datasets
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

validation_generator = validation_datagen.flow_from_directory(
```

```

validation_dir,
target_size=(224, 224),
batch_size=32,
class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode=None
)

```

Found 6435 images belonging to 50 classes.
Found 830 images belonging to 50 classes.
Found 134 images belonging to 1 classes.

```

In [98]: # Build an improved CNN model
model_2 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

    tf.keras.layers.Conv2D(256, (3, 3), activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu', kernel_regularizer=l2(0.01)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(0.5),

    tf.keras.layers.Dense(50, activation='softmax', kernel_regularizer=l2(0.01))
])

# Compile the model
model_2.compile(optimizer=tf.keras.optimizers.Adam(),
                loss='categorical_crossentropy',
                metrics=['accuracy'])

```

```
In [99]: model_2.summary()
```

Model: "sequential_9"

Layer (type)	Output Shape	Param #
=====		
conv2d_16 (Conv2D)	(None, 222, 222, 64)	1792
batch_normalization_9 (Batch Normalization)	(None, 222, 222, 64)	256
max_pooling2d_11 (MaxPooling2D)	(None, 111, 111, 64)	0
conv2d_17 (Conv2D)	(None, 109, 109, 128)	73856
batch_normalization_10 (Batch Normalization)	(None, 109, 109, 128)	512
max_pooling2d_12 (MaxPooling2D)	(None, 54, 54, 128)	0

conv2d_18 (Conv2D)	(None, 52, 52, 256)	295168
batch_normalization_11 (Batch Normalization)	(None, 52, 52, 256)	1024
max_pooling2d_13 (MaxPooling2D)	(None, 26, 26, 256)	0
flatten_10 (Flatten)	(None, 173056)	0
dense_13 (Dense)	(None, 512)	88605184
batch_normalization_12 (Batch Normalization)	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_14 (Dense)	(None, 50)	25650

=====

Total params: 89,005,490
Trainable params: 89,003,570
Non-trainable params: 1,920

```
In [104... # Define a learning rate scheduler
def lr_scheduler(epoch, lr):
    if epoch % 10 == 0:
        return lr * 0.9
    return lr

lr_schedule = LearningRateScheduler(lr_scheduler)

# Fit the improved model
history_2 = model_2.fit(
    train_generator,
    epochs=100,
    steps_per_epoch=len(train_generator),
    validation_data=validation_generator,
    validation_steps=len(validation_generator),
    callbacks=[lr_schedule] # Add the learning rate scheduler callback
)
```

```
Epoch 1/100
202/202 [=====] - 926s 5s/step - loss: 15.1466 - accuracy: 0.11
16 - val_loss: 9.1216 - val_accuracy: 0.0253 - lr: 8.1000e-04
Epoch 2/100
202/202 [=====] - 928s 5s/step - loss: 7.7699 - accuracy: 0.165
5 - val_loss: 7.9982 - val_accuracy: 0.0904 - lr: 8.1000e-04
Epoch 3/100
202/202 [=====] - 919s 5s/step - loss: 7.7320 - accuracy: 0.184
8 - val_loss: 7.1432 - val_accuracy: 0.2349 - lr: 8.1000e-04
Epoch 4/100
202/202 [=====] - 913s 5s/step - loss: 6.9982 - accuracy: 0.198
4 - val_loss: 6.6952 - val_accuracy: 0.2627 - lr: 8.1000e-04
Epoch 5/100
202/202 [=====] - 900s 4s/step - loss: 6.7225 - accuracy: 0.235
1 - val_loss: 7.7873 - val_accuracy: 0.2916 - lr: 8.1000e-04
Epoch 6/100
202/202 [=====] - 874s 4s/step - loss: 7.5444 - accuracy: 0.245
8 - val_loss: 7.3201 - val_accuracy: 0.2373 - lr: 8.1000e-04
Epoch 7/100
202/202 [=====] - 854s 4s/step - loss: 7.1579 - accuracy: 0.265
0 - val_loss: 6.8052 - val_accuracy: 0.3627 - lr: 8.1000e-04
Epoch 8/100
202/202 [=====] - 844s 4s/step - loss: 7.6971 - accuracy: 0.279
```

```
3 - val_loss: 7.4544 - val_accuracy: 0.2807 - lr: 8.1000e-04
Epoch 9/100
202/202 [=====] - 844s 4s/step - loss: 7.6983 - accuracy: 0.310
8 - val_loss: 11.2128 - val_accuracy: 0.1566 - lr: 8.1000e-04
Epoch 10/100
202/202 [=====] - 845s 4s/step - loss: 8.4845 - accuracy: 0.313
9 - val_loss: 8.6889 - val_accuracy: 0.3373 - lr: 8.1000e-04
Epoch 11/100
202/202 [=====] - 847s 4s/step - loss: 8.4705 - accuracy: 0.340
0 - val_loss: 8.7992 - val_accuracy: 0.3506 - lr: 7.2900e-04
Epoch 12/100
202/202 [=====] - 856s 4s/step - loss: 8.8176 - accuracy: 0.359
6 - val_loss: 9.7615 - val_accuracy: 0.2614 - lr: 7.2900e-04
Epoch 13/100
202/202 [=====] - 841s 4s/step - loss: 9.6433 - accuracy: 0.370
9 - val_loss: 9.9425 - val_accuracy: 0.3904 - lr: 7.2900e-04
Epoch 14/100
202/202 [=====] - 842s 4s/step - loss: 9.9977 - accuracy: 0.384
3 - val_loss: 10.3320 - val_accuracy: 0.4386 - lr: 7.2900e-04
Epoch 15/100
202/202 [=====] - 839s 4s/step - loss: 10.3308 - accuracy: 0.39
86 - val_loss: 12.2980 - val_accuracy: 0.3952 - lr: 7.2900e-04
Epoch 16/100
202/202 [=====] - 840s 4s/step - loss: 11.2024 - accuracy: 0.41
63 - val_loss: 10.9870 - val_accuracy: 0.5289 - lr: 7.2900e-04
Epoch 17/100
202/202 [=====] - 848s 4s/step - loss: 11.9446 - accuracy: 0.41
40 - val_loss: 13.3228 - val_accuracy: 0.4663 - lr: 7.2900e-04
Epoch 18/100
202/202 [=====] - 849s 4s/step - loss: 11.7448 - accuracy: 0.43
28 - val_loss: 13.1383 - val_accuracy: 0.3410 - lr: 7.2900e-04
Epoch 19/100
202/202 [=====] - 847s 4s/step - loss: 12.2617 - accuracy: 0.44
02 - val_loss: 12.0649 - val_accuracy: 0.4012 - lr: 7.2900e-04
Epoch 20/100
202/202 [=====] - 847s 4s/step - loss: 12.0105 - accuracy: 0.45
16 - val_loss: 11.9166 - val_accuracy: 0.5120 - lr: 7.2900e-04
Epoch 21/100
202/202 [=====] - 850s 4s/step - loss: 11.7918 - accuracy: 0.46
12 - val_loss: 11.2953 - val_accuracy: 0.5036 - lr: 6.5610e-04
Epoch 22/100
202/202 [=====] - 846s 4s/step - loss: 11.9582 - accuracy: 0.48
33 - val_loss: 12.1409 - val_accuracy: 0.5193 - lr: 6.5610e-04
Epoch 23/100
202/202 [=====] - 860s 4s/step - loss: 12.3491 - accuracy: 0.49
56 - val_loss: 12.8979 - val_accuracy: 0.4880 - lr: 6.5610e-04
Epoch 24/100
202/202 [=====] - 862s 4s/step - loss: 12.5745 - accuracy: 0.48
89 - val_loss: 11.5999 - val_accuracy: 0.5325 - lr: 6.5610e-04
Epoch 25/100
202/202 [=====] - 874s 4s/step - loss: 11.7406 - accuracy: 0.51
44 - val_loss: 13.1831 - val_accuracy: 0.5386 - lr: 6.5610e-04
Epoch 26/100
202/202 [=====] - 876s 4s/step - loss: 12.3814 - accuracy: 0.51
53 - val_loss: 12.0948 - val_accuracy: 0.4916 - lr: 6.5610e-04
Epoch 27/100
202/202 [=====] - 880s 4s/step - loss: 11.7924 - accuracy: 0.52
90 - val_loss: 15.5876 - val_accuracy: 0.3602 - lr: 6.5610e-04
Epoch 28/100
202/202 [=====] - 869s 4s/step - loss: 12.3238 - accuracy: 0.52
94 - val_loss: 12.4116 - val_accuracy: 0.5807 - lr: 6.5610e-04
Epoch 29/100
202/202 [=====] - 863s 4s/step - loss: 12.6930 - accuracy: 0.53
91 - val_loss: 12.1047 - val_accuracy: 0.5663 - lr: 6.5610e-04
Epoch 30/100
202/202 [=====] - 861s 4s/step - loss: 13.1657 - accuracy: 0.50
```

```
64 - val_loss: 12.3816 - val_accuracy: 0.5663 - lr: 6.5610e-04
Epoch 31/100
202/202 [=====] - 867s 4s/step - loss: 11.7935 - accuracy: 0.54
33 - val_loss: 11.1162 - val_accuracy: 0.6530 - lr: 5.9049e-04
Epoch 32/100
202/202 [=====] - 859s 4s/step - loss: 11.5715 - accuracy: 0.55
23 - val_loss: 11.5581 - val_accuracy: 0.5976 - lr: 5.9049e-04
Epoch 33/100
202/202 [=====] - 860s 4s/step - loss: 11.6264 - accuracy: 0.57
03 - val_loss: 11.3319 - val_accuracy: 0.5663 - lr: 5.9049e-04
Epoch 34/100
202/202 [=====] - 852s 4s/step - loss: 11.2001 - accuracy: 0.57
58 - val_loss: 11.8346 - val_accuracy: 0.6036 - lr: 5.9049e-04
Epoch 35/100
202/202 [=====] - 858s 4s/step - loss: 11.3047 - accuracy: 0.58
63 - val_loss: 11.9482 - val_accuracy: 0.5566 - lr: 5.9049e-04
Epoch 36/100
202/202 [=====] - 853s 4s/step - loss: 11.3323 - accuracy: 0.58
31 - val_loss: 10.8125 - val_accuracy: 0.5988 - lr: 5.9049e-04
Epoch 37/100
202/202 [=====] - 852s 4s/step - loss: 11.1347 - accuracy: 0.58
49 - val_loss: 11.2118 - val_accuracy: 0.5819 - lr: 5.9049e-04
Epoch 38/100
202/202 [=====] - 854s 4s/step - loss: 11.0377 - accuracy: 0.59
69 - val_loss: 11.2792 - val_accuracy: 0.6434 - lr: 5.9049e-04
Epoch 39/100
202/202 [=====] - 920s 5s/step - loss: 10.9717 - accuracy: 0.59
44 - val_loss: 12.3719 - val_accuracy: 0.5904 - lr: 5.9049e-04
Epoch 40/100
202/202 [=====] - 997s 5s/step - loss: 11.5436 - accuracy: 0.59
98 - val_loss: 12.0319 - val_accuracy: 0.5506 - lr: 5.9049e-04
Epoch 41/100
202/202 [=====] - 953s 5s/step - loss: 10.8142 - accuracy: 0.60
85 - val_loss: 10.2861 - val_accuracy: 0.6735 - lr: 5.3144e-04
Epoch 42/100
202/202 [=====] - 958s 5s/step - loss: 10.4603 - accuracy: 0.61
72 - val_loss: 10.3043 - val_accuracy: 0.6675 - lr: 5.3144e-04
Epoch 43/100
202/202 [=====] - 953s 5s/step - loss: 10.4800 - accuracy: 0.61
91 - val_loss: 10.4024 - val_accuracy: 0.6434 - lr: 5.3144e-04
Epoch 44/100
202/202 [=====] - 942s 5s/step - loss: 10.4182 - accuracy: 0.63
28 - val_loss: 11.9118 - val_accuracy: 0.4880 - lr: 5.3144e-04
Epoch 45/100
202/202 [=====] - 1030s 5s/step - loss: 10.9350 - accuracy: 0.6
106 - val_loss: 12.7253 - val_accuracy: 0.4506 - lr: 5.3144e-04
Epoch 46/100
202/202 [=====] - 1041s 5s/step - loss: 10.7529 - accuracy: 0.6
193 - val_loss: 10.4502 - val_accuracy: 0.6735 - lr: 5.3144e-04
Epoch 47/100
202/202 [=====] - 1069s 5s/step - loss: 10.5345 - accuracy: 0.6
320 - val_loss: 10.3748 - val_accuracy: 0.6482 - lr: 5.3144e-04
Epoch 48/100
202/202 [=====] - 1069s 5s/step - loss: 10.4515 - accuracy: 0.6
311 - val_loss: 10.9082 - val_accuracy: 0.5325 - lr: 5.3144e-04
Epoch 49/100
202/202 [=====] - 1048s 5s/step - loss: 10.3644 - accuracy: 0.6
413 - val_loss: 10.2501 - val_accuracy: 0.6928 - lr: 5.3144e-04
Epoch 50/100
202/202 [=====] - 1070s 5s/step - loss: 10.4034 - accuracy: 0.6
398 - val_loss: 10.6996 - val_accuracy: 0.5783 - lr: 5.3144e-04
Epoch 51/100
182/202 [=====>...] - ETA: 1:40 - loss: 10.2010 - accuracy: 0.6457
```

KeyboardInterrupt

Traceback (most recent call last)

```

~\AppData\Local\Temp\ipykernel_20500\1825073648.py in <module>
      8
      9 # Fit the improved model
----> 10 history_2 = model_2.fit(
      11     train_generator,
      12     epochs=100,

C:\ProgramData\Anaconda3\lib\site-packages\keras\utils\traceback_utils.py in error_handler(*args, **kwargs)
      63     filtered_tb = None
      64     try:
----> 65         return fn(*args, **kwargs)
      66     except Exception as e:
      67         filtered_tb = _process_traceback_frames(e.__traceback__)

C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\training.py in fit(self, x, y, batch_size, epochs, verbose, callbacks, validation_split, validation_data, shuffle, class_weight, sample_weight, initial_epoch, steps_per_epoch, validation_steps, validation_batch_size, validation_freq, max_queue_size, workers, use_multiprocessing)
     1648         ):
     1649             callbacks.on_train_batch_begin(step)
-> 1650             tmp_logs = self.train_function(iterator)
     1651             if data_handler.should_sync:
     1652                 context.async_wait()

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\util\traceback_utils.py in error_handler(*args, **kwargs)
     148     filtered_tb = None
     149     try:
--> 150         return fn(*args, **kwargs)
     151     except Exception as e:
     152         filtered_tb = _process_traceback_frames(e.__traceback__)

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\eager\polymorphic_function\polymorphic_function.py in __call__(self, *args, **kwargs)
     878
     879     with OptionalXlaContext(self._jit_compile):
--> 880         result = self._call(*args, **kwargs)
     881
     882         new_tracing_count = self.experimental_get_tracing_count()

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\eager\polymorphic_function\polymorphic_function.py in _call(self, *args, **kwargs)
     910     # In this case we have created variables on the first call, so we run the
     911     # defunned version which is guaranteed to never create variables.
--> 912     return self._no_variable_creation_fn(*args, **kwargs) # pylint: disable=not-
-callable
     913     elif self._variable_creation_fn is not None:
     914         # Release the lock early so that multiple threads can perform the call

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\eager\polymorphic_function\tracing_compiler.py in __call__(self, *args, **kwargs)
     132     (concrete_function,
     133      filtered_flat_args) = self._maybe_define_function(args, kwargs)
--> 134     return concrete_function._call_flat(
     135         filtered_flat_args, captured_inputs=concrete_function.captured_inputs)
     # pylint: disable=protected-access
     136

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\eager\polymorphic_function\monomorphic_function.py in _call_flat(self, args, captured_inputs, cancellation_manager)
     1743         and executing_eagerly):
     1744             # No tape is watching; skip to running the function.
-> 1745             return self._build_call_outputs(self._inference_function.call(
     1746                 ctx, args, cancellation_manager=cancellation_manager))

```



```

1747         forward_backward = self._select_forward_and_backward_functions(

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\eager\polymorphic_function\monomorphic_function.py in call(self, ctx, args, cancellation_manager)
    376         with _InterpolateFunctionError(self):
    377             if cancellation_manager is None:
--> 378                 outputs = execute.execute(
    379                     str(self.signature.name),
    380                     num_outputs=self._num_outputs,

C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\eager\execute.py in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
    50     try:
    51         ctx.ensure_initialized()
---> 52         tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
    53                                             inputs, attrs, num_outputs)
    54     except core._NotOkStatusException as e:

KeyboardInterrupt:

```

In [106.. *# I stopped the training since above model was not performing well*

Using a pretrained Model

```

In [118.. # Define a learning rate schedule function
def lr_schedule_2(epoch):
    initial_lr = 0.001 # Initial learning rate
    decay_factor = 0.9 # Decay factor
    lr = initial_lr * (decay_factor ** epoch)
    return lr

# Create a learning rate scheduler callback
lr_scheduler_2 = LearningRateScheduler(lr_schedule_2)

# Load the InceptionV3 model pre-trained on ImageNet data
base_model = InceptionV3(weights='imagenet',
                          include_top=False,
                          input_shape=(224, 224, 3))

# Classification layers
x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dropout(0.5)(x)
x = tf.keras.layers.Dense(512, activation='relu')(x)
x = tf.keras.layers.Dropout(0.5)(x)
predictions = tf.keras.layers.Dense(50, activation='softmax')(x)

# Create the new model
model_3 = tf.keras.models.Model(inputs=base_model.input, outputs=predictions)

# Freeze the layers of the pre-trained model
for layer in base_model.layers:
    layer.trainable = False

# Compile the model
model_3.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
               loss='categorical_crossentropy',
               metrics=['accuracy'])

```

```

In [127.. # Early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

# Train the model with the learning rate scheduler and early stopping

```

```

history_3 = model_3.fit(
    train_generator,
    epochs=30,
    validation_data=validation_generator,
    steps_per_epoch=len(train_generator),
    validation_steps=len(validation_generator),
    callbacks=[lr_scheduler_2, early_stopping]
)

```

Epoch 1/30

202/202 [=====] - 357s 2s/step - loss: 2.4891 - accuracy: 0.3649 - val_loss: 0.9620 - val_accuracy: 0.7518 - lr: 0.0010

Epoch 2/30

202/202 [=====] - 341s 2s/step - loss: 1.4087 - accuracy: 0.5991 - val_loss: 0.6961 - val_accuracy: 0.7928 - lr: 9.0000e-04

Epoch 3/30

202/202 [=====] - 365s 2s/step - loss: 1.1765 - accuracy: 0.6577 - val_loss: 0.6049 - val_accuracy: 0.8289 - lr: 8.1000e-04

Epoch 4/30

202/202 [=====] - 378s 2s/step - loss: 1.0676 - accuracy: 0.6867 - val_loss: 0.5533 - val_accuracy: 0.8434 - lr: 7.2900e-04

Epoch 5/30

202/202 [=====] - 357s 2s/step - loss: 0.9568 - accuracy: 0.7207 - val_loss: 0.4823 - val_accuracy: 0.8554 - lr: 6.5610e-04

Epoch 6/30

202/202 [=====] - 358s 2s/step - loss: 0.8907 - accuracy: 0.7378 - val_loss: 0.4639 - val_accuracy: 0.8723 - lr: 5.9049e-04

Epoch 7/30

202/202 [=====] - 365s 2s/step - loss: 0.8647 - accuracy: 0.7450 - val_loss: 0.4364 - val_accuracy: 0.8699 - lr: 5.3144e-04

Epoch 8/30

202/202 [=====] - 358s 2s/step - loss: 0.8230 - accuracy: 0.7596 - val_loss: 0.4180 - val_accuracy: 0.8771 - lr: 4.7830e-04

Epoch 9/30

202/202 [=====] - 377s 2s/step - loss: 0.7831 - accuracy: 0.7671 - val_loss: 0.4134 - val_accuracy: 0.8747 - lr: 4.3047e-04

Epoch 10/30

202/202 [=====] - 381s 2s/step - loss: 0.7543 - accuracy: 0.7708 - val_loss: 0.4056 - val_accuracy: 0.8843 - lr: 3.8742e-04

Epoch 11/30

202/202 [=====] - 410s 2s/step - loss: 0.7451 - accuracy: 0.7683 - val_loss: 0.3772 - val_accuracy: 0.8795 - lr: 3.4868e-04

Epoch 12/30

202/202 [=====] - 378s 2s/step - loss: 0.7082 - accuracy: 0.7869 - val_loss: 0.3782 - val_accuracy: 0.8880 - lr: 3.1381e-04

Epoch 13/30

202/202 [=====] - 362s 2s/step - loss: 0.6955 - accuracy: 0.7916 - val_loss: 0.3649 - val_accuracy: 0.8867 - lr: 2.8243e-04

Epoch 14/30

202/202 [=====] - 363s 2s/step - loss: 0.6761 - accuracy: 0.7930 - val_loss: 0.3609 - val_accuracy: 0.8904 - lr: 2.5419e-04

Epoch 15/30

202/202 [=====] - 396s 2s/step - loss: 0.6615 - accuracy: 0.8023 - val_loss: 0.3602 - val_accuracy: 0.8880 - lr: 2.2877e-04

Epoch 16/30

202/202 [=====] - 429s 2s/step - loss: 0.6694 - accuracy: 0.7915 - val_loss: 0.3424 - val_accuracy: 0.8952 - lr: 2.0589e-04

Epoch 17/30

202/202 [=====] - 649s 3s/step - loss: 0.6361 - accuracy: 0.8075 - val_loss: 0.3498 - val_accuracy: 0.8952 - lr: 1.8530e-04

Epoch 18/30

202/202 [=====] - 352s 2s/step - loss: 0.6271 - accuracy: 0.8145 - val_loss: 0.3333 - val_accuracy: 0.8928 - lr: 1.6677e-04

Epoch 19/30

202/202 [=====] - 376s 2s/step - loss: 0.6232 - accuracy: 0.8120 - val_loss: 0.3336 - val_accuracy: 0.8964 - lr: 1.5009e-04

```

Epoch 20/30
202/202 [=====] - 384s 2s/step - loss: 0.6006 - accuracy: 0.819
6 - val_loss: 0.3350 - val_accuracy: 0.8904 - lr: 1.3509e-04
Epoch 21/30
202/202 [=====] - 368s 2s/step - loss: 0.5880 - accuracy: 0.817
6 - val_loss: 0.3209 - val_accuracy: 0.8976 - lr: 1.2158e-04
Epoch 22/30
202/202 [=====] - 380s 2s/step - loss: 0.5759 - accuracy: 0.823
9 - val_loss: 0.3214 - val_accuracy: 0.8952 - lr: 1.0942e-04
Epoch 23/30
202/202 [=====] - 387s 2s/step - loss: 0.5873 - accuracy: 0.824
4 - val_loss: 0.3178 - val_accuracy: 0.8976 - lr: 9.8477e-05
Epoch 24/30
202/202 [=====] - 402s 2s/step - loss: 0.5744 - accuracy: 0.824
1 - val_loss: 0.3123 - val_accuracy: 0.8976 - lr: 8.8629e-05
Epoch 25/30
202/202 [=====] - 331s 2s/step - loss: 0.5805 - accuracy: 0.825
6 - val_loss: 0.3223 - val_accuracy: 0.8916 - lr: 7.9766e-05
Epoch 26/30
202/202 [=====] - 335s 2s/step - loss: 0.5700 - accuracy: 0.827
7 - val_loss: 0.3178 - val_accuracy: 0.9012 - lr: 7.1790e-05
Epoch 27/30
202/202 [=====] - 339s 2s/step - loss: 0.5793 - accuracy: 0.823
3 - val_loss: 0.3050 - val_accuracy: 0.9012 - lr: 6.4611e-05
Epoch 28/30
202/202 [=====] - 346s 2s/step - loss: 0.5814 - accuracy: 0.823
9 - val_loss: 0.3066 - val_accuracy: 0.8964 - lr: 5.8150e-05
Epoch 29/30
202/202 [=====] - 328s 2s/step - loss: 0.5449 - accuracy: 0.831
7 - val_loss: 0.3075 - val_accuracy: 0.9024 - lr: 5.2335e-05
Epoch 30/30
202/202 [=====] - 329s 2s/step - loss: 0.5426 - accuracy: 0.833
4 - val_loss: 0.3086 - val_accuracy: 0.9024 - lr: 4.7101e-05

```

In [129.. `model_3.summary()`

Model: "model_5"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_12 (InputLayer)	[(None, 224, 224, 3 0)]		[]
conv2d_1053 (Conv2D)	(None, 111, 111, 32 864)		['input_12[0][0]']
batch_normalization_1047 (Batch Normalization)	(None, 111, 111, 32 96)		['conv2d_1053[0][0]']
activation_1034 (Activation)	(None, 111, 111, 32 0)		['batch_normalization_1047[0][0]']

```

conv2d_1054 (Conv2D)          (None, 109, 109, 32) 9216      ['activation_1034[0]
[0]']
                                )

batch_normalization_1048 (Batch Normalization) (None, 109, 109, 32) 96      ['conv2d_1054[0][0]']

activation_1035 (Activation)    (None, 109, 109, 32) 0        ['batch_normalization_1
048[0][0]']
                                )

conv2d_1055 (Conv2D)          (None, 109, 109, 64) 18432    ['activation_1035[0]
[0]']
                                )

batch_normalization_1049 (Batch Normalization) (None, 109, 109, 64) 192    ['conv2d_1055[0][0]']

activation_1036 (Activation)    (None, 109, 109, 64) 0        ['batch_normalization_1
049[0][0]']
                                )

max_pooling2d_58 (MaxPooling2D) (None, 54, 54, 64) 0        ['activation_1036[0]
[0]']
                                )

conv2d_1056 (Conv2D)          (None, 54, 54, 80) 5120      ['max_pooling2d_58[0]
[0]']

batch_normalization_1050 (Batch Normalization) (None, 54, 54, 80) 240      ['conv2d_1056[0][0]']

activation_1037 (Activation)    (None, 54, 54, 80) 0        ['batch_normalization_1
050[0][0]']
                                ]

conv2d_1057 (Conv2D)          (None, 52, 52, 192) 138240    ['activation_1037[0]
[0]']

batch_normalization_1051 (Batch Normalization) (None, 52, 52, 192) 576      ['conv2d_1057[0][0]']
hNormalization)

```

activation_1038 (Activation)	(None, 52, 52, 192)	0	['batch_normalization_1051[0][0]']
max_pooling2d_59 (MaxPooling2D)	(None, 25, 25, 192)	0	['activation_1038[0][0]']
conv2d_1061 (Conv2D)	(None, 25, 25, 64)	12288	['max_pooling2d_59[0][0]']
batch_normalization_1055 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1061[0][0]']
activation_1042 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1055[0][0]']
conv2d_1059 (Conv2D)	(None, 25, 25, 48)	9216	['max_pooling2d_59[0][0]']
conv2d_1062 (Conv2D)	(None, 25, 25, 96)	55296	['activation_1042[0][0]']
batch_normalization_1053 (Batch Normalization)	(None, 25, 25, 48)	144	['conv2d_1059[0][0]']
batch_normalization_1056 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_1062[0][0]']
activation_1040 (Activation)	(None, 25, 25, 48)	0	['batch_normalization_1053[0][0]']
activation_1043 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_1056[0][0]']
average_pooling2d_99 (Average Pooling2D)	(None, 25, 25, 192)	0	['max_pooling2d_59[0][0]']

conv2d_1058 (Conv2D)	(None, 25, 25, 64)	12288	['max_pooling2d_59[0][0]']
conv2d_1060 (Conv2D)	(None, 25, 25, 64)	76800	['activation_1040[0][0]']
conv2d_1063 (Conv2D)	(None, 25, 25, 96)	82944	['activation_1043[0][0]']
conv2d_1064 (Conv2D)	(None, 25, 25, 32)	6144	['average_pooling2d_99[0][0]']
batch_normalization_1052 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1058[0][0]']
batch_normalization_1054 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1060[0][0]']
batch_normalization_1057 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_1063[0][0]']
batch_normalization_1058 (Batch Normalization)	(None, 25, 25, 32)	96	['conv2d_1064[0][0]']
activation_1039 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1052[0][0]']
activation_1041 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1054[0][0]']
activation_1044 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_1057[0][0]']
activation_1045 (Activation)	(None, 25, 25, 32)	0	['batch_normalization_1058[0][0]']

mixed0 (Concatenate)	(None, 25, 25, 256)	0	['activation_1039[0] [0]', [0]', [0]', [0]']
conv2d_1068 (Conv2D)	(None, 25, 25, 64)	16384	['mixed0[0][0]']
batch_normalization_1062 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1068[0][0]']
activation_1049 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1062[0][0]']
conv2d_1066 (Conv2D)	(None, 25, 25, 48)	12288	['mixed0[0][0]']
conv2d_1069 (Conv2D)	(None, 25, 25, 96)	55296	['activation_1049[0][0]']
batch_normalization_1060 (Batch Normalization)	(None, 25, 25, 48)	144	['conv2d_1066[0][0]']
batch_normalization_1063 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_1069[0][0]']
activation_1047 (Activation)	(None, 25, 25, 48)	0	['batch_normalization_1060[0][0]']
activation_1050 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_1063[0][0]']
average_pooling2d_100 (Average Pooling2D)	(None, 25, 25, 256)	0	['mixed0[0][0]']
conv2d_1065 (Conv2D)	(None, 25, 25, 64)	16384	['mixed0[0][0]']

conv2d_1067 (Conv2D)	(None, 25, 25, 64)	76800	['activation_1047[0][0]']
conv2d_1070 (Conv2D)	(None, 25, 25, 96)	82944	['activation_1050[0][0]']
conv2d_1071 (Conv2D)	(None, 25, 25, 64)	16384	['average_pooling2d_100[0][0]']
batch_normalization_1059 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1065[0][0]']
batch_normalization_1061 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1067[0][0]']
batch_normalization_1064 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_1070[0][0]']
batch_normalization_1065 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1071[0][0]']
activation_1046 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1059[0][0]']
activation_1048 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1061[0][0]']
activation_1051 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_1064[0][0]']
activation_1052 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1065[0][0]']
mixed1 (Concatenate)	(None, 25, 25, 288)	0	['activation_1046[0][0]', 'activation_1048[0][0]']

[0]',				'activation_1051[0]
[0]',				'activation_1052[0]
[0]']				
conv2d_1075 (Conv2D)	(None, 25, 25, 64)	18432		['mixed1[0][0]']
batch_normalization_1069 (Batch Normalization)	(None, 25, 25, 64)	192		['conv2d_1075[0][0]']
activation_1056 (Activation)	(None, 25, 25, 64)	0		['batch_normalization_1069[0][0]']
conv2d_1073 (Conv2D)	(None, 25, 25, 48)	13824		['mixed1[0][0]']
conv2d_1076 (Conv2D)	(None, 25, 25, 96)	55296		['activation_1056[0][0]']
batch_normalization_1067 (Batch Normalization)	(None, 25, 25, 48)	144		['conv2d_1073[0][0]']
batch_normalization_1070 (Batch Normalization)	(None, 25, 25, 96)	288		['conv2d_1076[0][0]']
activation_1054 (Activation)	(None, 25, 25, 48)	0		['batch_normalization_1067[0][0]']
activation_1057 (Activation)	(None, 25, 25, 96)	0		['batch_normalization_1070[0][0]']
average_pooling2d_101 (Average Pooling2D)	(None, 25, 25, 288)	0		['mixed1[0][0]']
conv2d_1072 (Conv2D)	(None, 25, 25, 64)	18432		['mixed1[0][0]']
conv2d_1074 (Conv2D)	(None, 25, 25, 64)	76800		['activation_1054[0][0]']

[0]']

conv2d_1077 (Conv2D) (None, 25, 25, 96) 82944 ['activation_1057[0]
[0]']

conv2d_1078 (Conv2D) (None, 25, 25, 64) 18432 ['average_pooling2d_101
[0][0]']

batch_normalization_1066 (Batch Normalization) (None, 25, 25, 64) 192 ['conv2d_1072[0][0]']

batch_normalization_1068 (Batch Normalization) (None, 25, 25, 64) 192 ['conv2d_1074[0][0]']

batch_normalization_1071 (Batch Normalization) (None, 25, 25, 96) 288 ['conv2d_1077[0][0]']

batch_normalization_1072 (Batch Normalization) (None, 25, 25, 64) 192 ['conv2d_1078[0][0]']

activation_1053 (Activation) (None, 25, 25, 64) 0 ['batch_normalization_1
066[0][0]']
]

activation_1055 (Activation) (None, 25, 25, 64) 0 ['batch_normalization_1
068[0][0]']
]

activation_1058 (Activation) (None, 25, 25, 96) 0 ['batch_normalization_1
071[0][0]']
]

activation_1059 (Activation) (None, 25, 25, 64) 0 ['batch_normalization_1
072[0][0]']
]

mixed2 (Concatenate) (None, 25, 25, 288) 0 ['activation_1053[0]
[0]',
['activation_1055[0]
[0]',
['activation_1058[0]
[0]',
['activation_1059[0]

[0]']

conv2d_1080 (Conv2D)	(None, 25, 25, 64)	18432	['mixed2[0][0]']
batch_normalization_1074 (Batch Normalization)	(None, 25, 25, 64)	192	['conv2d_1080[0][0]']
activation_1061 (Activation)	(None, 25, 25, 64)	0	['batch_normalization_1074[0][0]']
conv2d_1081 (Conv2D)	(None, 25, 25, 96)	55296	['activation_1061[0][0]']
batch_normalization_1075 (Batch Normalization)	(None, 25, 25, 96)	288	['conv2d_1081[0][0]']
activation_1062 (Activation)	(None, 25, 25, 96)	0	['batch_normalization_1075[0][0]']
conv2d_1079 (Conv2D)	(None, 12, 12, 384)	995328	['mixed2[0][0]']
conv2d_1082 (Conv2D)	(None, 12, 12, 96)	82944	['activation_1062[0][0]']
batch_normalization_1073 (Batch Normalization)	(None, 12, 12, 384)	1152	['conv2d_1079[0][0]']
batch_normalization_1076 (Batch Normalization)	(None, 12, 12, 96)	288	['conv2d_1082[0][0]']
activation_1060 (Activation)	(None, 12, 12, 384)	0	['batch_normalization_1073[0][0]']
activation_1063 (Activation)	(None, 12, 12, 96)	0	['batch_normalization_1076[0][0]']

max_pooling2d_60 (MaxPooling2D	(None, 12, 12, 288)	0	['mixed2[0][0]']
)			
mixed3 (Concatenate)	(None, 12, 12, 768)	0	['activation_1060[0]
[0]',			'activation_1063[0]
[0]',			'max_pooling2d_60[0]
[0]']			
conv2d_1087 (Conv2D)	(None, 12, 12, 128)	98304	['mixed3[0][0]']
batch_normalization_1081 (Batc	(None, 12, 12, 128)	384	['conv2d_1087[0][0]']
hNormalization)			
activation_1068 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_1
081[0][0]']
conv2d_1088 (Conv2D)	(None, 12, 12, 128)	114688	['activation_1068[0]
[0]']			
batch_normalization_1082 (Batc	(None, 12, 12, 128)	384	['conv2d_1088[0][0]']
hNormalization)			
activation_1069 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_1
082[0][0]']
conv2d_1084 (Conv2D)	(None, 12, 12, 128)	98304	['mixed3[0][0]']
conv2d_1089 (Conv2D)	(None, 12, 12, 128)	114688	['activation_1069[0]
[0]']			
batch_normalization_1078 (Batc	(None, 12, 12, 128)	384	['conv2d_1084[0][0]']
hNormalization)			
batch_normalization_1083 (Batc	(None, 12, 12, 128)	384	['conv2d_1089[0][0]']
hNormalization)			

activation_1065 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_1078[0][0]']
activation_1070 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_1083[0][0]']
conv2d_1085 (Conv2D)	(None, 12, 12, 128)	114688	['activation_1065[0][0]']
conv2d_1090 (Conv2D)	(None, 12, 12, 128)	114688	['activation_1070[0][0]']
batch_normalization_1079 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_1085[0][0]']
batch_normalization_1084 (Batch Normalization)	(None, 12, 12, 128)	384	['conv2d_1090[0][0]']
activation_1066 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_1079[0][0]']
activation_1071 (Activation)	(None, 12, 12, 128)	0	['batch_normalization_1084[0][0]']
average_pooling2d_102 (Average Pooling2D)	(None, 12, 12, 768)	0	['mixed3[0][0]']
conv2d_1083 (Conv2D)	(None, 12, 12, 192)	147456	['mixed3[0][0]']
conv2d_1086 (Conv2D)	(None, 12, 12, 192)	172032	['activation_1066[0][0]']
conv2d_1091 (Conv2D)	(None, 12, 12, 192)	172032	['activation_1071[0][0]']
conv2d_1092 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_102[0][0]']

batch_normalization_1077 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1083[0][0]']
batch_normalization_1080 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1086[0][0]']
batch_normalization_1085 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1091[0][0]']
batch_normalization_1086 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1092[0][0]']
activation_1064 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1077[0][0]']
activation_1067 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1080[0][0]']
activation_1072 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1085[0][0]']
activation_1073 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1086[0][0]']
mixed4 (Concatenate)	(None, 12, 12, 768)	0	['activation_1064[0][0]', 'activation_1067[0][0]', 'activation_1072[0][0]', 'activation_1073[0][0]']
conv2d_1097 (Conv2D)	(None, 12, 12, 160)	122880	['mixed4[0][0]']
batch_normalization_1091 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1097[0][0]']

activation_1078 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1091[0][0]']
------------------------------	---------------------	---	------------------------------------

conv2d_1098 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1078[0][0]']
----------------------	---------------------	--------	---------------------------

batch_normalization_1092 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1098[0][0]']
--	---------------------	-----	-----------------------

activation_1079 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1092[0][0]']
------------------------------	---------------------	---	------------------------------------

conv2d_1094 (Conv2D)	(None, 12, 12, 160)	122880	['mixed4[0][0]']
----------------------	---------------------	--------	------------------

conv2d_1099 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1079[0][0]']
----------------------	---------------------	--------	---------------------------

batch_normalization_1088 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1094[0][0]']
--	---------------------	-----	-----------------------

batch_normalization_1093 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1099[0][0]']
--	---------------------	-----	-----------------------

activation_1075 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1088[0][0]']
------------------------------	---------------------	---	------------------------------------

activation_1080 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1093[0][0]']
------------------------------	---------------------	---	------------------------------------

conv2d_1095 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1075[0][0]']
----------------------	---------------------	--------	---------------------------

conv2d_1100 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1080[0][0]']
----------------------	---------------------	--------	---------------------------

batch_normalization_1089 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1095[0][0]']
--	---------------------	-----	-----------------------

hNormalization)

batch_normalization_1094 (Batch Normalization) (None, 12, 12, 160) 480 ['conv2d_1100[0][0]']

hNormalization)

activation_1076 (Activation) (None, 12, 12, 160) 0 ['batch_normalization_1089[0][0]']

activation_1081 (Activation) (None, 12, 12, 160) 0 ['batch_normalization_1094[0][0]']

average_pooling2d_103 (Average Pooling2D) (None, 12, 12, 768) 0 ['mixed4[0][0]']

conv2d_1093 (Conv2D) (None, 12, 12, 192) 147456 ['mixed4[0][0]']

conv2d_1096 (Conv2D) (None, 12, 12, 192) 215040 ['activation_1076[0][0]']

conv2d_1101 (Conv2D) (None, 12, 12, 192) 215040 ['activation_1081[0][0]']

conv2d_1102 (Conv2D) (None, 12, 12, 192) 147456 ['average_pooling2d_103[0][0]']

batch_normalization_1087 (Batch Normalization) (None, 12, 12, 192) 576 ['conv2d_1093[0][0]']

batch_normalization_1090 (Batch Normalization) (None, 12, 12, 192) 576 ['conv2d_1096[0][0]']

batch_normalization_1095 (Batch Normalization) (None, 12, 12, 192) 576 ['conv2d_1101[0][0]']

batch_normalization_1096 (Batch Normalization) (None, 12, 12, 192) 576 ['conv2d_1102[0][0]']

activation_1074 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1087[0][0]']
activation_1077 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1090[0][0]']
activation_1082 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1095[0][0]']
activation_1083 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1096[0][0]']
mixed5 (Concatenate)	(None, 12, 12, 768)	0	['activation_1074[0][0]', 'activation_1077[0][0]', 'activation_1082[0][0]', 'activation_1083[0][0]']
conv2d_1107 (Conv2D)	(None, 12, 12, 160)	122880	['mixed5[0][0]']
batch_normalization_1101 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1107[0][0]']
activation_1088 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1101[0][0]']
conv2d_1108 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1088[0][0]']
batch_normalization_1102 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1108[0][0]']
activation_1089 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1102[0][0]']

conv2d_1104 (Conv2D)	(None, 12, 12, 160)	122880	['mixed5[0][0]']
conv2d_1109 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1089[0][0]']
batch_normalization_1098 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1104[0][0]']
batch_normalization_1103 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1109[0][0]']
activation_1085 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1098[0][0]']
activation_1090 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1103[0][0]']
conv2d_1105 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1085[0][0]']
conv2d_1110 (Conv2D)	(None, 12, 12, 160)	179200	['activation_1090[0][0]']
batch_normalization_1099 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1105[0][0]']
batch_normalization_1104 (Batch Normalization)	(None, 12, 12, 160)	480	['conv2d_1110[0][0]']
activation_1086 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1099[0][0]']
activation_1091 (Activation)	(None, 12, 12, 160)	0	['batch_normalization_1104[0][0]']

average_pooling2d_104 (Average Pooling2D)	(None, 12, 12, 768)	0	['mixed5[0][0]']
conv2d_1103 (Conv2D)	(None, 12, 12, 192)	147456	['mixed5[0][0]']
conv2d_1106 (Conv2D)	(None, 12, 12, 192)	215040	['activation_1086[0][0]']
conv2d_1111 (Conv2D)	(None, 12, 12, 192)	215040	['activation_1091[0][0]']
conv2d_1112 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_104[0][0]']
batch_normalization_1097 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1103[0][0]']
batch_normalization_1100 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1106[0][0]']
batch_normalization_1105 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1111[0][0]']
batch_normalization_1106 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1112[0][0]']
activation_1084 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1097[0][0]']
activation_1087 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1100[0][0]']
activation_1092 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1105[0][0]']
activation_1093 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1106[0][0]']

```

106[0][0]'
]

mixed6 (Concatenate)      (None, 12, 12, 768)  0      ['activation_1084[0]
[0]',
                                'activation_1087[0]
[0]',
                                'activation_1092[0]
[0]',
                                'activation_1093[0]
[0]']

conv2d_1117 (Conv2D)      (None, 12, 12, 192)  147456  ['mixed6[0][0]']

batch_normalization_1111 (Batch Normalization) (None, 12, 12, 192)  576      ['conv2d_1117[0][0]']

activation_1098 (Activation) (None, 12, 12, 192)  0      ['batch_normalization_1
111[0][0]']

conv2d_1118 (Conv2D)      (None, 12, 12, 192)  258048  ['activation_1098[0]
[0]']

batch_normalization_1112 (Batch Normalization) (None, 12, 12, 192)  576      ['conv2d_1118[0][0]']

activation_1099 (Activation) (None, 12, 12, 192)  0      ['batch_normalization_1
112[0][0]']

conv2d_1114 (Conv2D)      (None, 12, 12, 192)  147456  ['mixed6[0][0]']

conv2d_1119 (Conv2D)      (None, 12, 12, 192)  258048  ['activation_1099[0]
[0]']

batch_normalization_1108 (Batch Normalization) (None, 12, 12, 192)  576      ['conv2d_1114[0][0]']

batch_normalization_1113 (Batch Normalization) (None, 12, 12, 192)  576      ['conv2d_1119[0][0]']

```

activation_1095 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1 108[0][0]']
activation_1100 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1 113[0][0]']
conv2d_1115 (Conv2D)	(None, 12, 12, 192)	258048	['activation_1095[0] [0]']
conv2d_1120 (Conv2D)	(None, 12, 12, 192)	258048	['activation_1100[0] [0]']
batch_normalization_1109 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1115[0][0]']
batch_normalization_1114 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1120[0][0]']
activation_1096 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1 109[0][0]']
activation_1101 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1 114[0][0]']
average_pooling2d_105 (Average Pooling2D)	(None, 12, 12, 768)	0	['mixed6[0][0]']
conv2d_1113 (Conv2D)	(None, 12, 12, 192)	147456	['mixed6[0][0]']
conv2d_1116 (Conv2D)	(None, 12, 12, 192)	258048	['activation_1096[0] [0]']
conv2d_1121 (Conv2D)	(None, 12, 12, 192)	258048	['activation_1101[0] [0]']
conv2d_1122 (Conv2D)	(None, 12, 12, 192)	147456	['average_pooling2d_105 [0][0]']

batch_normalization_1107 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1113[0][0]']
batch_normalization_1110 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1116[0][0]']
batch_normalization_1115 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1121[0][0]']
batch_normalization_1116 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1122[0][0]']
activation_1094 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1107[0][0]']
activation_1097 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1110[0][0]']
activation_1102 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1115[0][0]']
activation_1103 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_1116[0][0]']
mixed7 (Concatenate)	(None, 12, 12, 768)	0	['activation_1094[0][0]', 'activation_1097[0][0]', 'activation_1102[0][0]', 'activation_1103[0][0]']
conv2d_1125 (Conv2D)	(None, 12, 12, 192)	147456	['mixed7[0][0]']
batch_normalization_1119 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1125[0][0]']

activation_1106 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_119[0][0]']
conv2d_1126 (Conv2D)	(None, 12, 12, 192)	258048	['activation_1106[0][0]']
batch_normalization_1120 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1126[0][0]']
activation_1107 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_120[0][0]']
conv2d_1123 (Conv2D)	(None, 12, 12, 192)	147456	['mixed7[0][0]']
conv2d_1127 (Conv2D)	(None, 12, 12, 192)	258048	['activation_1107[0][0]']
batch_normalization_1117 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1123[0][0]']
batch_normalization_1121 (Batch Normalization)	(None, 12, 12, 192)	576	['conv2d_1127[0][0]']
activation_1104 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_117[0][0]']
activation_1108 (Activation)	(None, 12, 12, 192)	0	['batch_normalization_121[0][0]']
conv2d_1124 (Conv2D)	(None, 5, 5, 320)	552960	['activation_1104[0][0]']
conv2d_1128 (Conv2D)	(None, 5, 5, 192)	331776	['activation_1108[0][0]']
batch_normalization_1118 (Batch Normalization)	(None, 5, 5, 320)	960	['conv2d_1124[0][0]']

hNormalization)				
batch_normalization_1122 (Batch Normalization)	(None, 5, 5, 192)	576	['conv2d_1128[0][0]']	
hNormalization)				
activation_1105 (Activation)	(None, 5, 5, 320)	0	['batch_normalization_1118[0][0]']	
activation_1109 (Activation)	(None, 5, 5, 192)	0	['batch_normalization_1122[0][0]']	
max_pooling2d_61 (MaxPooling2D)	(None, 5, 5, 768)	0	['mixed7[0][0]']	
mixed8 (Concatenate)	(None, 5, 5, 1280)	0	['activation_1105[0][0]', 'activation_1109[0][0]', 'max_pooling2d_61[0][0]']	
conv2d_1133 (Conv2D)	(None, 5, 5, 448)	573440	['mixed8[0][0]']	
batch_normalization_1127 (Batch Normalization)	(None, 5, 5, 448)	1344	['conv2d_1133[0][0]']	
activation_1114 (Activation)	(None, 5, 5, 448)	0	['batch_normalization_1127[0][0]']	
conv2d_1130 (Conv2D)	(None, 5, 5, 384)	491520	['mixed8[0][0]']	
conv2d_1134 (Conv2D)	(None, 5, 5, 384)	1548288	['activation_1114[0][0]']	
batch_normalization_1124 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1130[0][0]']	

batch_normalization_1128 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1134[0][0]']
activation_1111 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_1124[0][0]']
activation_1115 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_1128[0][0]']
conv2d_1131 (Conv2D)	(None, 5, 5, 384)	442368	['activation_1111[0][0]']
conv2d_1132 (Conv2D)	(None, 5, 5, 384)	442368	['activation_1111[0][0]']
conv2d_1135 (Conv2D)	(None, 5, 5, 384)	442368	['activation_1115[0][0]']
conv2d_1136 (Conv2D)	(None, 5, 5, 384)	442368	['activation_1115[0][0]']
average_pooling2d_106 (Average Pooling2D)	(None, 5, 5, 1280)	0	['mixed8[0][0]']
conv2d_1129 (Conv2D)	(None, 5, 5, 320)	409600	['mixed8[0][0]']
batch_normalization_1125 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1131[0][0]']
batch_normalization_1126 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1132[0][0]']
batch_normalization_1129 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1135[0][0]']
batch_normalization_1130 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1136[0][0]']

conv2d_1137 (Conv2D)	(None, 5, 5, 192)	245760	['average_pooling2d_106[0][0]']
batch_normalization_1123 (Batch Normalization)	(None, 5, 5, 320)	960	['conv2d_1129[0][0]']
activation_1112 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_125[0][0]']
activation_1113 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_126[0][0]']
activation_1116 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_129[0][0]']
activation_1117 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_130[0][0]']
batch_normalization_1131 (Batch Normalization)	(None, 5, 5, 192)	576	['conv2d_1137[0][0]']
activation_1110 (Activation)	(None, 5, 5, 320)	0	['batch_normalization_123[0][0]']
mixed9_0 (Concatenate)	(None, 5, 5, 768)	0	['activation_1112[0][0]', 'activation_1113[0][0]']
concatenate_22 (Concatenate)	(None, 5, 5, 768)	0	['activation_1116[0][0]', 'activation_1117[0][0]']
activation_1118 (Activation)	(None, 5, 5, 192)	0	['batch_normalization_131[0][0]']

mixed9 (Concatenate) [0]',	(None, 5, 5, 2048)	0	['activation_1110[0]
			'mixed9_0[0][0]',
			'concatenate_22[0]
[0]',			'activation_1118[0]
[0]']			
conv2d_1142 (Conv2D)	(None, 5, 5, 448)	917504	['mixed9[0][0]']
batch_normalization_1136 (Batch Normalization)	(None, 5, 5, 448)	1344	['conv2d_1142[0][0]']
activation_1123 (Activation) 136[0][0]'	(None, 5, 5, 448)	0	['batch_normalization_1]
conv2d_1139 (Conv2D)	(None, 5, 5, 384)	786432	['mixed9[0][0]']
conv2d_1143 (Conv2D) [0]']	(None, 5, 5, 384)	1548288	['activation_1123[0]]
batch_normalization_1133 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1139[0][0]']
batch_normalization_1137 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1143[0][0]']
activation_1120 (Activation) 133[0][0]'	(None, 5, 5, 384)	0	['batch_normalization_1]
activation_1124 (Activation) 137[0][0]'	(None, 5, 5, 384)	0	['batch_normalization_1]
conv2d_1140 (Conv2D) [0]']	(None, 5, 5, 384)	442368	['activation_1120[0]]
conv2d_1141 (Conv2D) [0]']	(None, 5, 5, 384)	442368	['activation_1120[0]]

conv2d_1144 (Conv2D)	(None, 5, 5, 384)	442368	['activation_1124[0][0]']
conv2d_1145 (Conv2D)	(None, 5, 5, 384)	442368	['activation_1124[0][0]']
average_pooling2d_107 (Average Pooling2D)	(None, 5, 5, 2048)	0	['mixed9[0][0]']
conv2d_1138 (Conv2D)	(None, 5, 5, 320)	655360	['mixed9[0][0]']
batch_normalization_1134 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1140[0][0]']
batch_normalization_1135 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1141[0][0]']
batch_normalization_1138 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1144[0][0]']
batch_normalization_1139 (Batch Normalization)	(None, 5, 5, 384)	1152	['conv2d_1145[0][0]']
conv2d_1146 (Conv2D)	(None, 5, 5, 192)	393216	['average_pooling2d_107[0][0]']
batch_normalization_1132 (Batch Normalization)	(None, 5, 5, 320)	960	['conv2d_1138[0][0]']
activation_1121 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_1134[0][0]']
activation_1122 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_1135[0][0]']
activation_1125 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_1134[0][0]']

138[0][0]']
activation_1126 (Activation)	(None, 5, 5, 384)	0	['batch_normalization_1	
139[0][0]']	
batch_normalization_1140 (Batch Normalization)	(None, 5, 5, 192)	576	['conv2d_1146[0][0]']	
activation_1119 (Activation)	(None, 5, 5, 320)	0	['batch_normalization_1	
132[0][0]']	
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0	['activation_1121[0]	
[0]',			'activation_1122[0]	
[0]']				
concatenate_23 (Concatenate)	(None, 5, 5, 768)	0	['activation_1125[0]	
[0]',			'activation_1126[0]	
[0]']				
activation_1127 (Activation)	(None, 5, 5, 192)	0	['batch_normalization_1	
140[0][0]']	
mixed10 (Concatenate)	(None, 5, 5, 2048)	0	['activation_1119[0]	
[0]',			'mixed9_1[0][0]',	
[0]',			'concatenate_23[0]	
[0]']			'activation_1127[0]	
global_average_pooling2d_8 (Global Average Pooling2D)	(None, 2048)	0	['mixed10[0][0]']	
dropout_13 (Dropout)	(None, 2048)	0	['global_average_poolin	
g2d_8[0][0]			']']	
dense_28 (Dense)	(None, 512)	1049088	['dropout_13[0][0]']	

dropout_14 (Dropout)	(None, 512)	0	['dense_28[0][0]']
dense_29 (Dense)	(None, 50)	25650	['dropout_14[0][0]']

```
=====
Total params: 22,877,522
Trainable params: 1,074,738
Non-trainable params: 21,802,784
=====
```

```
In [133... # Evaluating model
score_3 = model_3.evaluate_generator(validation_generator)
```

```
C:\Users\Sam\AppData\Local\Temp\ipykernel_20500\2728850985.py:2: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.
score_3 = model_3.evaluate_generator(validation_generator)
```

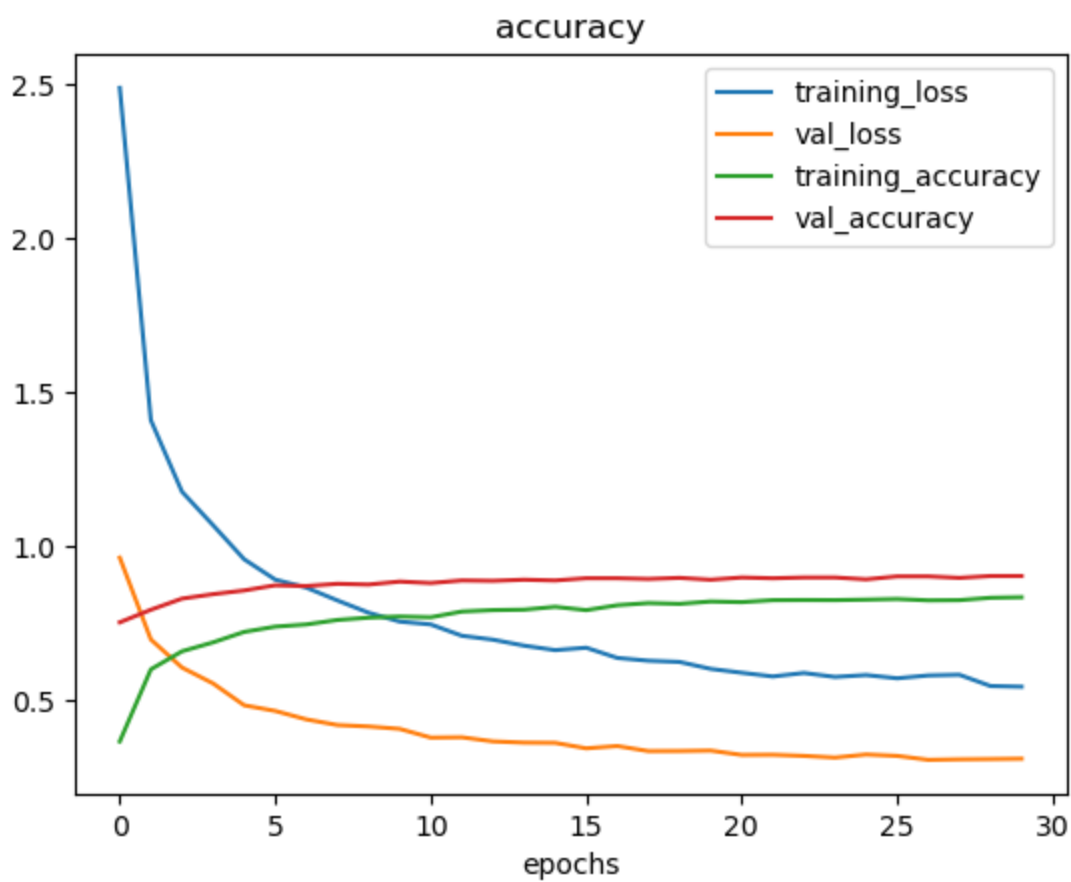
```
In [134... print('Test loss (Model 3):', score_3[0])
print('Test accuracy: (Model 3)', score_3[1])
```

```
Test loss (Model 3): 0.3085521161556244
Test accuracy: (Model 3) 0.9024096131324768
```

```
In [135... predict_3 = model_3.predict_generator(test_generator)
# predict the class label
y_classes_3 = predict_3.argmax(axis=-1)
```

```
C:\Users\Sam\AppData\Local\Temp\ipykernel_20500\4206833547.py:1: UserWarning: `Model.predict_generator` is deprecated and will be removed in a future version. Please use `Model.predict`, which supports generators.
predict_3 = model_3.predict_generator(test_generator)
```

```
In [144... plot_loss_curves(history_3)
```

```
In [148... model_3.save("./Models/best_model.h5")
```

```
In [ ]:
```

Prediction on some images

```
In [152... # Load the saved model
loaded_model = tf.keras.models.load_model('./Models/best_model.h5')
```

```
In [156... # Load and preprocess a single image
img_path = './Dataset/Reduced_Dataset/prediction_images/01.jpeg'
img = image.load_img(img_path, target_size=(224, 224))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array = preprocess_input(img_array)

# Make a prediction
predictions = loaded_model.predict(img_array)

# Decode and print the predicted class
predicted_class = np.argmax(predictions)
print(f"Predicted Class Index: {predicted_class}")
print(f"The image is of: {class_names[predicted_class]}")

1/1 [=====] - 0s 88ms/step
Predicted Class Index: 0
The image is of: ALBERTS TOWHEE
```

```
In [29]:
```

```
In [53]:
```

```
In [68]:
```

