```
from keras.models import Sequential
from keras.layers import Dense, Conv2D,  MaxPooling2D, Flatten


model = Sequential()


model.add(Conv2D(32, (3,3), activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D(pool_size=(2,2,)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(4, activation='softmax'))


model.summary()
```

```
Model: "sequential_1"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ conv2d_3 (Conv2D)                    │ (None, 62, 62, 32)          │             896 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_1 (MaxPooling2D)       │ (None, 31, 31, 32)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ flatten_1 (Flatten)                  │ (None, 30752)               │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_2 (Dense)                      │ (None, 128)                 │       3,936,384 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_3 (Dense)                      │ (None, 4)                   │             516 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘
 Total params: 3,937,796 (15.02 MB)
 Trainable params: 3,937,796 (15.02 MB)
 Non-trainable params: 0 (0.00 B)
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   rotation_range = 0.2,
                                   width_shift_range = 0.2,
                                   height_shift_range = 0.2,
                                   fill_mode = 'nearest',
                                   vertical_flip = True,
                                   horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)


train_path =  '/content/drive/MyDrive/image/train'
test_path =  '/content/drive/MyDrive/image/test'
train_generator = train_datagen.flow_from_directory(train_path,
                                                    target_size=(64,64),
                                                    batch_size=8,
                                                    class_mode='categorical')
test_generator = test_datagen.flow_from_directory(test_path,
                                                  target_size=(64,64),
                                                  batch_size=8,
                                                  class_mode='categorical')
```

```
Found 49 images belonging to 4 classes.
Found 16 images belonging to 4 classes.
```

```
train_generator.class_indices
```

```
{'butter fly': 0, 'peacock': 1, 'rabbit': 2, 'sunflower': 3}
```

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

```
model.fit(train_generator, epochs=100, validation_data=test_generator)
```

```
Epoch 91/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 273ms/step - accuracy: 0.9266 - loss: 0.1994 - val_accurac
Epoch 92/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 237ms/step - accuracy: 0.8816 - loss: 0.2446 - val_accurac
Epoch 93/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 211ms/step - accuracy: 0.9423 - loss: 0.1744 - val_accurac
Epoch 94/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 224ms/step - accuracy: 0.9403 - loss: 0.1507 - val_accurac
Epoch 95/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 204ms/step - accuracy: 0.9276 - loss: 0.1469 - val_accurac
Epoch 96/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 213ms/step - accuracy: 0.8925 - loss: 0.1682 - val_accurac
Epoch 97/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 282ms/step - accuracy: 0.9680 - loss: 0.1394 - val_accurac
Epoch 98/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 282ms/step - accuracy: 0.9830 - loss: 0.1067 - val_accurac
Epoch 99/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 191ms/step - accuracy: 0.9372 - loss: 0.1740 - val_accurac
Epoch 100/100
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 219ms/step - accuracy: 0.9267 - loss: 0.1577 - val_accurac
<keras.src.callbacks.history.History at 0x7e43d10ed410>
```

```python
model.save('peacock-sunflower-classifier.h5')
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.savi
```

Start coding or generate with AI.