

```
In [1]: pip install split_folders
```

Requirement already satisfied: split_folders in c:\programdata\anaconda3\lib\site-packages (0.4.2)
Note: you may need to restart the kernel to use updated packages.

```
In [2]: import splitfolders
```

```
In [3]: input_folders="F:/models/flower_dataset/input_dataset"  
output="F:/models/flower_dataset/output_dataset"  
splitfolders.ratio(input_folders, output, seed=42, ratio=(.6, .2, .2))
```

Copying files: 4326 files [02:06, 34.11 files/s]

In [4]: `import tensorflow`

```
C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood a
s (type, (1,)) / '(1,)type'.
_np_qint8 = np.dtype(["qint8", np.int8, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood a
s (type, (1,)) / '(1,)type'.
_np_quint8 = np.dtype(["quint8", np.uint8, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood a
s (type, (1,)) / '(1,)type'.
_np_qint16 = np.dtype(["qint16", np.int16, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood a
s (type, (1,)) / '(1,)type'.
_np_quint16 = np.dtype(["quint16", np.uint16, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood a
s (type, (1,)) / '(1,)type'.
_np_qint32 = np.dtype(["qint32", np.int32, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:525: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood a
s (type, (1,)) / '(1,)type'.
_np_resource = np.dtype(["resource", np.ubyte, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541: FutureWarning: Pa
ssing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be unders
tood as (type, (1,)) / '(1,)type'.
_np_qint8 = np.dtype(["qint8", np.int8, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542: FutureWarning: Pa
ssing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be unders
tood as (type, (1,)) / '(1,)type'.
_np_quint8 = np.dtype(["quint8", np.uint8, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Pa
ssing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be unders
tood as (type, (1,)) / '(1,)type'.
_np_qint16 = np.dtype(["qint16", np.int16, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544: FutureWarning: Pa
ssing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be unders
tood as (type, (1,)) / '(1,)type'.
_np_quint16 = np.dtype(["quint16", np.uint16, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Pa
```

```

ssing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be unders
tood as (type, (1,)) / '(1,)type'.
_np_qint32 = np.dtype(["qint32", np.int32, 1])
C:\ProgramData\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550: FutureWarning: Pa
ssing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be unders
tood as (type, (1,)) / '(1,)type'.
    np_resource = np.dtype(["resource", np.ubyte, 1])

```

```

In [5]: import tensorflow
from tensorflow.keras.layers import Conv2D, Flatten, Dense, MaxPool2D, BatchNormalization, GlobalAveragePooling2D
from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import Model
import matplotlib.pyplot as plt
import numpy as np

```

```

In [6]: img_height, img_width=(224, 224)
batch_size=32
train_data_dir=r"F:\models\flower_dataset\output_dataset\train"
test_data_dir=r"F:\models\flower_dataset\output_dataset\test"
valid_data_dir=r"F:\models\flower_dataset\output_dataset\val"

```

```
In [12]: train_datagen=ImageDataGenerator(preprocessing_function=preprocess_input,
                                          shear_range=.2,
                                          zoom_range=.2,
                                          horizontal_flip=True,
                                          validation_split=.4)
train_generator=train_datagen.flow_from_directory(train_data_dir,
                                                  target_size=(img_height, img_width),
                                                  batch_size=batch_size,
                                                  class_mode='categorical',
                                                  subset='training')
valid_generator=train_datagen.flow_from_directory(valid_data_dir,
                                                  target_size=(img_height, img_width),
                                                  batch_size=batch_size,
                                                  class_mode='categorical',
                                                  subset='validation')
test_generator=train_datagen.flow_from_directory(test_data_dir,
                                                  target_size=(img_height, img_width),
                                                  batch_size=1,
                                                  class_mode='categorical',
                                                  subset='validation')
```

Found 1557 images belonging to 5 classes.
Found 342 images belonging to 5 classes.
Found 347 images belonging to 5 classes.

```
In [13]: x,y=test_generator.next()
x.shape
```

```
Out[13]: (1, 224, 224, 3)
```

```
In [25]: base_model=ResNet50(include_top=False, weights='imagenet')
x=base_model.output
x=GlobalAveragePooling2D()(x)
x=Dense(1024, activation='relu')(x)
predictions=Dense(train_generator.num_classes,activation='softmax')(x)
model=Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable=False

In [26]: model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(train_generator, epochs=10)
```

```
Epoch 1/10
49/49 [=====] - 1246s 25s/step - loss: 1.1088 - acc: 0.7020
Epoch 2/10
49/49 [=====] - 1297s 26s/step - loss: 0.3367 - acc: 0.8831
Epoch 3/10
49/49 [=====] - 1099s 22s/step - loss: 0.2124 - acc: 0.9242
Epoch 4/10
49/49 [=====] - 1559s 32s/step - loss: 0.1912 - acc: 0.9377
Epoch 5/10
49/49 [=====] - 1101s 22s/step - loss: 0.1741 - acc: 0.9383
Epoch 6/10
49/49 [=====] - 1150s 23s/step - loss: 0.1177 - acc: 0.9589
Epoch 7/10
49/49 [=====] - 1104s 23s/step - loss: 0.1676 - acc: 0.9486
Epoch 8/10
49/49 [=====] - 1345s 27s/step - loss: 0.1063 - acc: 0.9666
Epoch 9/10
49/49 [=====] - 1373s 28s/step - loss: 0.0877 - acc: 0.9692
Epoch 10/10
49/49 [=====] - 1227s 25s/step - loss: 0.0693 - acc: 0.9788
```

```
Out[26]: <tensorflow.python.keras.callbacks.History at 0x20c9517c608>
```

```
In [30]: test_loss, test_acc = model.evaluate(test_generator, verbose=2)
print('\n test accuracy:', test_acc)
```

347/347 - 328s - loss: 0.5831 - acc: 0.8588

test accuracy: 0.8587896

```
In [ ]:
```