

```
In [2]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [8]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True, vertical_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
x_train=train_datagen.flow_from_directory("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\train",
                                         target_size=(64,64), class_mode='categorical', batch_size=24)
```

Found 4317 images belonging to 5 classes.

```
In [11]: y_test=test_datagen.flow_from_directory("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\test",
                                                target_size=(64,64), class_mode='categorical', batch_size=24)
```

Found 4317 images belonging to 5 classes.

```
In [26]: y_test=test_datagen.flow_from_directory("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\test",
                                                target_size=(64,64), class_mode='categorical', batch_size=24)
```

Found 4317 images belonging to 5 classes.

```
In [12]: x_train.class_indices
```

```
Out[12]: {'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

```
In [13]: from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D, Flatten, Dense
```

```
In [14]: model=Sequential()
```

```
In [15]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
```

```
In [16]: model.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [17]: model.add(Flatten())
```

```
In [19]: model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
```

```
In [20]: model.add(Dense(5,activation='softmax'))
```

```
In [21]: model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer='adam')
```

```
In [22]: len(x_train)
```

```
Out[22]: 180
```

```
In [29]: model.fit(x_train, epochs = 2, validation_data=y_test, steps_per_epoch=len(x_train), validation_steps=len(y_test))
```

Epoch 1/2

180/180 [=====] - 110s 595ms/step - loss: 1.2745 - accuracy: 0.4899 - val\_loss: 1.1505 - val\_accuracy: 0.5411

Epoch 2/2

180/180 [=====] - 65s 359ms/step - loss: 1.0020 - accuracy: 0.6074 - val\_loss: 0.9054 - val\_accuracy: 0.6542

```
Out[29]: <keras.callbacks.History at 0x1e719f05100>
```

```
In [35]: model.save('flowers.h5')
```

```
In [39]: ls C:\Users\ssdha\Downloads\Flowers-Dataset\flowers
```

Volume in drive C is dharma  
Volume Serial Number is 9E81-0CE5

Directory of C:\Users\ssdha\Downloads\Flowers-Dataset\flowers

31-10-2022	10.41 PM	<DIR>	.
31-10-2022	10.41 PM	<DIR>	..
31-10-2022	10.40 PM	<DIR>	daisy
31-10-2022	10.40 PM	<DIR>	dandelion
31-10-2022	10.41 PM	<DIR>	rose
31-10-2022	10.41 PM	<DIR>	sunflower
31-10-2022	10.42 PM	<DIR>	tulip
		0 File(s)	0 bytes
		7 Dir(s)	194,529,554,432 bytes free

```
In [40]: import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
In [41]: model=load_model('flowers.h5')
```

```
In [43]: img=image.load_img("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\flowers\\daisy\\2165")
```

```
In [44]: img
```

Out[44]:



```
In [45]: x=image.img_to_array(img)
```

```
In [46]: x
```

```
Out[46]: array([[1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.],
               ...,
               [1., 1., 1.],
               [1., 1., 1.],
               [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          ...,

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]], dtype=float32)
```

```
In [47]: x=np.expand_dims(x,axis=0)
```

```
In [48]: x
```

```

Out[48]: array([[[[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                ...,

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]]], dtype=float32)

```

```

In [57]: img=image.load_img("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\flowers\\daisy\\2165.jpg")
img
x=image.img_to_array(img)

```

```

In [58]: x

```

```

Out[58]: array([[1., 1., 1.],
                [1., 1., 1.],
                [1., 1., 1.],
                ...,
                [1., 1., 1.],
                [1., 1., 1.],
                [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          ...,

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]],

          [[1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.],
           ...,
           [1., 1., 1.],
           [1., 1., 1.],
           [1., 1., 1.]], dtype=float32)

```

```
In [59]: x=np.expand_dims(x,axis=0)
```

```
In [60]: x
```

```

Out[60]: array([[[[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                ...,

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]],

                [[1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.],
                  ...,
                  [1., 1., 1.],
                  [1., 1., 1.],
                  [1., 1., 1.]]], dtype=float32)

```

In [ ]:

```
In [61]: y=np.argmax(model.predict(x),axis=0)
```

1/1 [=====] - 0s 59ms/step

```
In [62]: y
```

```
Out[62]: array([0, 0, 0, 0, 0], dtype=int64)
```

```
In [63]: x_train.class_indices
```

```
Out[63]: {'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

```
In [64]: index=['daisy','dandelion','rose','sunflower']
```

```
In [65]: index[y[0]]
```

```
Out[65]: 'daisy'
```

```
In [68]: img=image.load_img("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\flowers\\dandelion\\  
x=image.img_to_array(img)  
x=np.expand_dims(x,axis=0)  
y=np.argmax(model.predict(x),axis=1)  
index=['daisy','dandelion','rose','sunflower']  
index[y[0]]
```

```
1/1 [=====] - 0s 23ms/step
```

```
Out[68]: 'sunflower'
```

```
In [69]: img=image.load_img("C:\\Users\\ssdha\\Downloads\\Flowers-Dataset\\flowers\\sunflower\\  
x=image.img_to_array(img)  
x=np.expand_dims(x,axis=0)  
y=np.argmax(model.predict(x),axis=0)  
index=['sunflower','daisy','dandelion','rose']  
index[y[0]]
```

```
1/1 [=====] - 0s 29ms/step
```

```
Out[69]: 'sunflower'
```

```
In [ ]:
```