```
!unzip "/content/Flowers-Dataset.zip"
```

```
  inflating: flowers/dandelion/7179487220_56e4725195_m.jpg
  inflating: flowers/dandelion/7184780734_3baab127c2_m.jpg
  inflating: flowers/dandelion/7188112181_571434b058_n.jpg
  inflating: flowers/dandelion/7193058132_36fd883048_m.jpg
  inflating: flowers/dandelion/7196409186_a59957ce0b_m.jpg
  inflating: flowers/dandelion/7196683612_6c4cf05b24.jpg
  inflating: flowers/dandelion/7197581386_8a51f1bb12_n.jpg
  inflating: flowers/dandelion/7218569994_de7045c0c0.jpg
  inflating: flowers/dandelion/7222962522_36952a67b6_n.jpg
  inflating: flowers/dandelion/7226987694_34552c3115_n.jpg
  inflating: flowers/dandelion/7232035352_84a39e99ba_n.jpg
  inflating: flowers/dandelion/7243174412_d3628e4cc4_m.jpg
  inflating: flowers/dandelion/7243478942_30bf542a2d_m.jpg
  inflating: flowers/dandelion/7247192002_39b79998f0_n.jpg
  inflating: flowers/dandelion/7249354462_21925f7d95_n.jpg
  inflating: flowers/dandelion/7262863194_682209e9fb.jpg
  inflating: flowers/dandelion/7267547016_c8903920bf.jpg
  inflating: flowers/dandelion/7270523166_b62fc9e5f1_m.jpg
  inflating: flowers/dandelion/7280217714_fb9ffccf2d_n.jpg
  inflating: flowers/dandelion/7280221020_98b473b20d_n.jpg
  inflating: flowers/dandelion/7280222348_a87725ca77.jpg
  inflating: flowers/dandelion/7280227122_7ea2bef7f4_n.jpg
  inflating: flowers/dandelion/7291185504_b740bbeba4_m.jpg
  inflating: flowers/dandelion/7295618968_c08a326cc1_m.jpg
  inflating: flowers/dandelion/7308600792_27cff2f73f.jpg
  inflating: flowers/dandelion/7315832212_b0ceeb8de8_n.jpg
  inflating: flowers/dandelion/7355522_b66e5d3078_m.jpg
  inflating: flowers/dandelion/7367491658_9eb4dc2384_m.jpg
  inflating: flowers/dandelion/7368435774_0045b9dc4e.jpg
  inflating: flowers/dandelion/7368449232_c99f49b2e6_n.jpg
  inflating: flowers/dandelion/7401173270_ebaf04c9b0_n.jpg
  inflating: flowers/dandelion/7425858848_d04dab08dd_n.jpg
  inflating: flowers/dandelion/7448453384_fb9caaa9af_n.jpg
  inflating: flowers/dandelion/7465850028_cdfaae235a_n.jpg
  inflating: flowers/dandelion/7469617666_0e1a014917.jpg
  inflating: flowers/dandelion/751941983_58e1ae3957_m.jpg
  inflating: flowers/dandelion/7719263062_3c8a307a5d.jpg
  inflating: flowers/dandelion/7808430998_31ba639031_n.jpg
  inflating: flowers/dandelion/7808545612_546cfca610_m.jpg
  inflating: flowers/dandelion/7843447416_847e6ba7f4_m.jpg
  inflating: flowers/dandelion/7884440256_91c033732d.jpg
  inflating: flowers/dandelion/7950892504_33142110c2.jpg
  inflating: flowers/dandelion/7950901292_2dea05f9a2_n.jpg
  inflating: flowers/dandelion/7998106328_c3953f70e9_n.jpg
  inflating: flowers/dandelion/8011324555_375b7b5b0a.jpg
  inflating: flowers/dandelion/8058286066_acdf082487_n.jpg
  inflating: flowers/dandelion/8079778274_f2a400f749_n.jpg
  inflating: flowers/dandelion/808239968_318722e4db.jpg
  inflating: flowers/dandelion/8083321316_f62ea76f72_n.jpg
  inflating: flowers/dandelion/80846315_d997645bea_n.jpg
  inflating: flowers/dandelion/8168031302_6e36f39d87.jpg
  inflating: flowers/dandelion/8181477_8cb77d2e0f_n.jpg
  inflating: flowers/dandelion/8194560480_bfc1fb5801.jpg
```

```
        inflating: flowers/dandelion/8209318399_ae72aefdb5.jpg
        inflating: flowers/dandelion/8220011556_28e0cab67f.jpg
        inflating: flowers/dandelion/8223949_2928d3f6f6_n.jpg
        inflating: flowers/dandelion/8223968_6b51555d2f_n.jpg
        inflating: flowers/dandelion/8267315764_129f2e1d77_m.jpg
```

```
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)
xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=24)
xtest = test_datagen.flow_from_directory('/content/flowers',
                                         target_size=(64,64),
                                         class_mode='categorical',
                                         batch_size=124)
```

```
     Found 4317 images belonging to 5 classes.
     Found 4317 images belonging to 5 classes.
```

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3))) # Convolution laye
```

```
model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling layer
```

```
model.add(Flatten()) # Flatten layer
```

```
model.add(Dense(300,activation='relu')) # Hidden layer 1
```

```
model.add(Dense(150,activation='relu')) # Hidden layer 2
```

```
model.add(Dense(4,activation='softmax')) # Output layer
```

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

```
len(xtrain)
```

```
180
```

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

```
model.fit(xtrain,
          steps_per_epoch=len(xtrain),
          epochs=5,
          validation_data=xtest,
          validation_steps=len(xtest))
```

```
Epoch 1/5
------------------------------------------------------------------------------
InvalidArgumentError                      Traceback (most recent call last)
<ipython-input-61-9a4ef8c8eb25> in <module>
      3           epochs=5,
      4           validation_data=xtest,
----> 5           validation_steps=len(xtest))

                                  1 frames
/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/execute.py in
quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     53     ctx.ensure_initialized()
     54     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
---> 55                                         inputs, attrs, num_outputs)
     56   except core._NotOkStatusException as e:
     57     if name is not None:

InvalidArgumentError: Graph execution error:

Detected at node 'categorical_crossentropy/softmax_cross_entropy_with_logits' defined
at (most recent call last):
    File "/usr/lib/python3.7/runpy.py", line 193, in _run_module_as_main
      "__main__", mod_spec)
    File "/usr/lib/python3.7/runpy.py", line 85, in _run_code
      exec(code, run_globals)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py", line 16, in
<module>
      app.launch_new_instance()
    File "/usr/local/lib/python3.7/dist-packages/traitlets/config/application.py",
line 846, in launch_instance
      app.start()
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/kernelapp.py", line 612,
in start
      self.io_loop.start()
    File "/usr/local/lib/python3.7/dist-packages/tornado/platform/asyncio.py", line
132, in start
      self.asyncio_loop.run_forever()
    File "/usr/lib/python3.7/asyncio/base_events.py", line 541, in run_forever
      self._run_once()
    File "/usr/lib/python3.7/asyncio/base_events.py", line 1786, in _run_once
      handle._run()
    File "/usr/lib/python3.7/asyncio/events.py", line 88, in _run
      self._context.run(self._callback, *self._args)
    File "/usr/local/lib/python3.7/dist-packages/tornado/ioloop.py", line 758, in
_run_callback
      ret = callback()
    File "/usr/local/lib/python3.7/dist-packages/tornado/stack_context.py", line 300,
in null_wrapper
      return fn(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 1233, in inner
      self.run()
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 1147, in run
      yielded = self.gen.send(value)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/kernelbase.py", line 365,
in process_one
      yield gen.maybe_future(dispatch(*args))
```

```
            File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 326, in
```

```
model.save('daisy.h5')
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
model=load_model("/content/daisy.h5")
```

```
img = image.load_img('/content/flowers/daisy/100080576_f52e8ee070_n.jpg',target_size=(64,64))
```

```
img
```

```
x = image.img_to_array(img)
```

```
x
```

```
array([[[141., 141., 139.],
        [149., 149., 149.],
        [152., 152., 154.],
        ...,
        [162., 161., 166.],
        [154., 154., 152.],
        [153., 153., 153.]],

       [[136., 135., 131.],
        [146., 145., 143.],
        [169., 168., 174.],
        ...,
        [159., 158., 163.],
        [155., 155., 153.],
        [149., 149., 149.]],

       [[125., 125., 117.],
        [138., 140., 137.],
        [152., 152., 152.],
```

```
       ...,
       [156., 156., 156.],
       [157., 157., 155.],
       [143., 142., 140.]],

      ...,

      [[ 41.,  44.,  23.],
       [ 43.,  46.,  25.],
       [ 49.,  51.,  37.],
       ...,
       [128., 124., 121.],
       [125., 121., 118.],
       [125., 122., 117.]],

      [[ 43.,  46.,  25.],
       [ 43.,  46.,  25.],
       [ 54.,  55.,  37.],
       ...,
       [130., 126., 125.],
       [129., 125., 124.],
       [127., 123., 122.]],

      [[ 44.,  47.,  26.],
       [ 45.,  48.,  27.],
       [ 53.,  55.,  34.],
       ...,
       [137., 133., 132.],
       [133., 129., 128.],
       [130., 126., 125.]]], dtype=float32)
```

```
x.ndim
```

```
3
```

```
x = np.expand_dims(x,axis=0)
```

```
x.ndim
```

```
4
```

```
pred = model.predict(x)
```

```
pred
```

```
array([[1.0000000e+00, 3.0940360e-19, 2.0464132e-20, 1.8283872e-10]],
      dtype=float32)
```

```
labels=["daisy","dandelion","rose","sunflower","tulip"]
```