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

Archive: /content/Flowers-Dataset.zip
  inflating: flowers/daisy/100080576_f52e8ee070_n.jpg
  inflating: flowers/daisy/10140303196_b88d3d6cec.jpg
  inflating: flowers/daisy/10172379554_b296050f82_n.jpg
  inflating: flowers/daisy/10172567486_2748826a8b.jpg
  inflating: flowers/daisy/10172636503_21bededa75_n.jpg
  inflating: flowers/daisy/102841525_bd6628ae3c.jpg
  inflating: flowers/daisy/10300722094_28fa978807_n.jpg
  inflating: flowers/daisy/1031799732_e7f4008c03.jpg
  inflating: flowers/daisy/10391248763_1d16681106_n.jpg
  inflating: flowers/daisy/10437754174_22ec990b77_m.jpg
  inflating: flowers/daisy/10437770546_8bb6f7bdd3_m.jpg
  inflating: flowers/daisy/10437929963_bc13eebe0c.jpg
  inflating: flowers/daisy/10466290366_cc72e33532.jpg
  inflating: flowers/daisy/10466558316_a7198b87e2.jpg
  inflating: flowers/daisy/10555749515_13a12a026e.jpg
  inflating: flowers/daisy/10555815624_dc211569b0.jpg
  inflating: flowers/daisy/10555826524_423eb8bf71_n.jpg
  inflating: flowers/daisy/10559679065_50d2b16f6d.jpg
  inflating: flowers/daisy/105806915_a9c13e2106_n.jpg
  inflating: flowers/daisy/10712722853_5632165b04.jpg
  inflating: flowers/daisy/107592979_aaa9cdfe78_m.jpg
  inflating: flowers/daisy/10770585085_4742b9dac3_n.jpg
  inflating: flowers/daisy/10841136265_af473efc60.jpg
  inflating: flowers/daisy/10993710036_2033222c91.jpg
  inflating: flowers/daisy/10993818044_4c19b86c82.jpg
  inflating: flowers/daisy/10994032453_ac7f8d9e2e.jpg
  inflating: flowers/daisy/11023214096_b5b39fab08.jpg
  inflating: flowers/daisy/11023272144_fce94401f2_m.jpg
  inflating: flowers/daisy/11023277956_8980d53169_m.jpg
  inflating: flowers/daisy/11124324295_503f3a0804.jpg
  inflating: flowers/daisy/1140299375_3aa7024466.jpg
  inflating: flowers/daisy/11439894966_dca877f0cd.jpg
  inflating: flowers/daisy/1150395827_6f94a5c6e4_n.jpg
  inflating: flowers/daisy/11642632_1e7627a2cc.jpg
  inflating: flowers/daisy/11834945233_a53b7a92ac_m.jpg
  inflating: flowers/daisy/11870378973_2ec1919f12.jpg
  inflating: flowers/daisy/11891885265_ccefec7284_n.jpg
  inflating: flowers/daisy/12193032636_b50ae7db35_n.jpg
  inflating: flowers/daisy/12348343085_d4c396e5b5_m.jpg
  inflating: flowers/daisy/12585131704_0f64b17059_m.jpg
  inflating: flowers/daisy/12601254324_3cb62c254a_m.jpg
  inflating: flowers/daisy/1265350143_6e2b276ec9.jpg
  inflating: flowers/daisy/12701063955_4840594ea6_n.jpg
  inflating: flowers/daisy/1285423653_18926dc2c8_n.jpg
  inflating: flowers/daisy/1286274236_1d7ac84efb_n.jpg
  inflating: flowers/daisy/12891819633_e4c82b51e8.jpg
  inflating: flowers/daisy/1299501272_59d9da5510_n.jpg
  inflating: flowers/daisy/1306119996_ab8ae14d72_n.jpg
  inflating: flowers/daisy/1314069875_da8dc023c6_m.jpg
  inflating: flowers/daisy/1342002397_9503c97b49.jpg
  inflating: flowers/daisy/134409839_71069a95d1_m.jpg
  inflating: flowers/daisy/1344985627_c3115e2d71_n.jpg
  inflating: flowers/daisy/13491959645_2cd9df44d6_n.jpg
  inflating: flowers/daisy/1354396826_2868631432_m.jpg
  inflating: flowers/daisy/1355787476_32e9f2a30b.jpg
  inflating: flowers/daisy/13583238844_573df2de8e_m.jpg
  inflating: flowers/daisy/1374193928_a52320eafa.jpg

Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
F_dgen=ImageDataGenerator(rescale=1./255,
                          zoom_range=0.2,
                          horizontal_flip=True)
```

```
Flowers= F_dgen.flow_from_directory('/content/flowers',
                                    target_size=(64,64),
                                    class_mode='categorical',
                                    batch_size=100)
```

Found 4317 images belonging to 5 classes.

Create a CNN model and add layers

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

```python
#Initializing the model
model = Sequential()

#convolution layer
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))

#MaxPooling layer
model.add(MaxPooling2D(pool_size=(2, 2)))

#Flatten layer
model.add(Flatten())

#Hidden layers
model.add(Dense(350,activation='relu'))
model.add(Dense(150,activation='relu'))

#output layer
model.add(Dense(5,activation='softmax'))
```

Compile the model

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

Fit the model

```python
model.fit_generator(Flowers,
                    steps_per_epoch=len(Flowers),
                    epochs=20,
                    validation_data=Flowers,
                    validation_steps=len(Flowers))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: `Model.fit_generator` is deprecated and will be remov
  """
Epoch 1/20
44/44 [==============================] - 116s 3s/step - loss: 1.7629 - accuracy: 0.3572 - val_loss: 1.1586 - val_accuracy: 0.5224
Epoch 2/20
44/44 [==============================] - 41s 941ms/step - loss: 1.1060 - accuracy: 0.5513 - val_loss: 0.9977 - val_accuracy: 0.607
Epoch 3/20
44/44 [==============================] - 43s 976ms/step - loss: 0.9431 - accuracy: 0.6361 - val_loss: 0.8046 - val_accuracy: 0.732
Epoch 4/20
44/44 [==============================] - 41s 933ms/step - loss: 0.7820 - accuracy: 0.7160 - val_loss: 0.6750 - val_accuracy: 0.766
Epoch 5/20
44/44 [==============================] - 42s 954ms/step - loss: 0.6301 - accuracy: 0.7753 - val_loss: 0.5807 - val_accuracy: 0.790
Epoch 6/20
44/44 [==============================] - 42s 960ms/step - loss: 0.4723 - accuracy: 0.8515 - val_loss: 0.3839 - val_accuracy: 0.874
Epoch 7/20
44/44 [==============================] - 41s 935ms/step - loss: 0.3292 - accuracy: 0.9011 - val_loss: 0.2238 - val_accuracy: 0.942
Epoch 8/20
44/44 [==============================] - 41s 929ms/step - loss: 0.2217 - accuracy: 0.9432 - val_loss: 0.1386 - val_accuracy: 0.979
Epoch 9/20
44/44 [==============================] - 41s 935ms/step - loss: 0.1554 - accuracy: 0.9648 - val_loss: 0.1079 - val_accuracy: 0.981
Epoch 10/20
44/44 [==============================] - 41s 943ms/step - loss: 0.0960 - accuracy: 0.9822 - val_loss: 0.0626 - val_accuracy: 0.991
Epoch 11/20
44/44 [==============================] - 42s 964ms/step - loss: 0.0540 - accuracy: 0.9917 - val_loss: 0.0341 - val_accuracy: 0.997
Epoch 12/20
44/44 [==============================] - 41s 945ms/step - loss: 0.0405 - accuracy: 0.9947 - val_loss: 0.0258 - val_accuracy: 0.997
Epoch 13/20
44/44 [==============================] - 41s 933ms/step - loss: 0.0275 - accuracy: 0.9968 - val_loss: 0.0224 - val_accuracy: 0.997
Epoch 14/20
44/44 [==============================] - 43s 974ms/step - loss: 0.0195 - accuracy: 0.9979 - val_loss: 0.0303 - val_accuracy: 0.996
Epoch 15/20
44/44 [==============================] - 41s 938ms/step - loss: 0.0181 - accuracy: 0.9986 - val_loss: 0.0102 - val_accuracy: 0.998
Epoch 16/20
44/44 [==============================] - 42s 953ms/step - loss: 0.0117 - accuracy: 0.9986 - val_loss: 0.0097 - val_accuracy: 0.999
Epoch 17/20
44/44 [==============================] - 41s 934ms/step - loss: 0.0110 - accuracy: 0.9988 - val_loss: 0.0076 - val_accuracy: 0.999
Epoch 18/20
44/44 [==============================] - 41s 934ms/step - loss: 0.0105 - accuracy: 0.9988 - val_loss: 0.0097 - val_accuracy: 0.999
Epoch 19/20
44/44 [==============================] - 41s 936ms/step - loss: 0.0109 - accuracy: 0.9986 - val_loss: 0.0057 - val_accuracy: 0.999
Epoch 20/20
44/44 [==============================] - 42s 951ms/step - loss: 0.0087 - accuracy: 0.9981 - val_loss: 0.0054 - val_accuracy: 0.999
<keras.callbacks.History at 0x7f1039e95290>
```

Save the model

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

Testing the model

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

```
img = image.load_img('/content/flowers/rose/11233672494_d8bf0a3dbf_n.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
      'rose'
```

```
img = image.load_img('/content/flowers/daisy/1286274236_1d7ac84efb_n.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
      'daisy'
```

```
img = image.load_img('/content/tulip image.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
      'tulip'
```

```
img = image.load_img('/content/sunflower  image.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
      'sunflower'
```

```
img = image.load_img('/content/dandelion images.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
      'dandelion'
```

```
img = image.load_img('/content/daisy image.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
prediction = np.argmax(model.predict(x))
op = ['daisy','dandelion','rose','sunflower','tulip']
op[prediction]
```

```
      'rose'
```