Unzip the dataset

```
!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
```

Image augmentation

```
# Import required lib

from tensorflow.keras.preprocessing.image import ImageDataGenerator


# Creating augumentation on training variable

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


# Creating augumentation on testing variable

test_datagen = ImageDataGenerator(rescale=1./255)


# Passing training data to train variable

xtrain = train_datagen.flow_from_directory(r'/content/training_set', target_size=(64, 64), cl

     Found 4317 images belonging to 5 classes.


# Passing testing data to test variable

xtest = test_datagen.flow_from_directory(r'/content/testing_set', target_size=(64, 64), class

     Found 4307 images belonging to 5 classes.
```

Create Model and Add Layers

```
# Importing required lib

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

```
# Creating CNN block

model = Sequential()

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

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

# Flatten layer
model.add(Flatten())

# Fully Connected layers(ANN)
model.add(Dense(300,activation='relu'))# Hidden layer1
model.add(Dense(150,activation='relu'))# Hidden layer2
model.add(Dense(5,activation='softmax'))# Output layer
```

Compile The Model

```
# Compile the Model

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


# train the model

model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=10,
                    validation_data=xtest,
                    validation_steps=len(xtest))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: UserWarning: `Model.fit_
  import sys
Epoch 1/10
44/44 [==============================] - 51s 1s/step - loss: 1.3874 - accuracy: 0.4341 -
Epoch 2/10
44/44 [==============================] - 44s 995ms/step - loss: 1.0770 - accuracy: 0.567
Epoch 3/10
44/44 [==============================] - 44s 991ms/step - loss: 0.9909 - accuracy: 0.612
Epoch 4/10
44/44 [==============================] - 44s 990ms/step - loss: 0.9366 - accuracy: 0.634
Epoch 5/10
44/44 [==============================] - 44s 995ms/step - loss: 0.8704 - accuracy: 0.663
Epoch 6/10
44/44 [==============================] - 44s 990ms/step - loss: 0.8013 - accuracy: 0.694
Epoch 7/10
44/44 [==============================] - 44s 995ms/step - loss: 0.7667 - accuracy: 0.709
Epoch 8/10
```

```
44/44 [==============================] - 43s 983ms/step - loss: 0.7666 - accuracy: 0.707
Epoch 9/10
44/44 [==============================] - 43s 979ms/step - loss: 0.7461 - accuracy: 0.725
Epoch 10/10
44/44 [==============================] - 43s 985ms/step - loss: 0.7126 - accuracy: 0.731
<keras.callbacks.History at 0x7f610e02ead0>
```

```
# saving the model

model.save("Flower.h5")
```

Testing Model

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


img = image.load_img(r'/content/testing_set/daisy/10555815624_dc211569b0.jpg',target_size=(64


img
```

```
# Converting image to array

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

```
array([[[ 10.,  15.,   8.],
        [ 12.,  17.,  10.],
        [ 15.,  20.,  13.],
        ...,
        [ 68.,  56.,  16.],
        [ 60.,  49.,  21.],
        [ 60.,  47.,  15.]],

       [[ 10.,  11.,   5.],
        [ 12.,  15.,   8.],
        [ 12.,  19.,  11.],
        ...,
        [ 70.,  61.,  22.],
        [ 59.,  53.,  19.],
        [ 66.,  51.,  18.]],

       [[  8.,  11.,   4.],
        [ 10.,  13.,   6.],
        [ 13.,  18.,  11.],
```

```
        ...,
        [ 53.,  64.,  22.],
        [ 54.,  54.,  20.],
        [ 56.,  51.,  21.]],

       ...,

       [[225., 212., 229.],
        [223., 212., 228.],
        [203., 195., 210.],
        ...,
        [ 60.,  81.,  25.],
        [ 82., 105.,  33.],
        [ 63.,  79.,  30.]],

       [[220., 207., 227.],
        [226., 217., 212.],
        [171., 158., 167.],
        ...,
        [ 21.,  32.,  15.],
        [ 52.,  73.,  30.],
        [ 38.,  58.,  23.]],

       [[212., 201., 217.],
        [224., 220., 208.],
        [160., 169., 112.],
        ...,
        [ 23.,  32.,  13.],
        [ 30.,  44.,  18.],
        [ 24.,  36.,  14.]]], dtype=float32)
```

```
# Expanding dimensions

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

```
array([[[[ 10.,  15.,   8.],
         [ 12.,  17.,  10.],
         [ 15.,  20.,  13.],
         ...,
         [ 68.,  56.,  16.],
         [ 60.,  49.,  21.],
         [ 60.,  47.,  15.]],

        [[ 10.,  11.,   5.],
         [ 12.,  15.,   8.],
         [ 12.,  19.,  11.],
         ...,
         [ 70.,  61.,  22.],
         [ 59.,  53.,  19.],
         [ 66.,  51.,  18.]],

        [[  8.,  11.,   4.],
         [ 10.,  13.,   6.],
         [ 13.,  18.,  11.],
```

```
       ...,
       [ 53.,  64.,  22.],
       [ 54.,  54.,  20.],
       [ 56.,  51.,  21.]],

      ...,

      [[225., 212., 229.],
       [223., 212., 228.],
       [203., 195., 210.],
       ...,
       [ 60.,  81.,  25.],
       [ 82., 105.,  33.],
       [ 63.,  79.,  30.]],

      [[220., 207., 227.],
       [226., 217., 212.],
       [171., 158., 167.],
       ...,
       [ 21.,  32.,  15.],
       [ 52.,  73.,  30.],
       [ 38.,  58.,  23.]],

      [[212., 201., 217.],
       [224., 220., 208.],
       [160., 169., 112.],
       ...,
       [ 23.,  32.,  13.],
       [ 30.,  44.,  18.],
       [ 24.,  36.,  14.]]]], dtype=float32)
```

```
# Predicting Flower

model.predict(x)
```

```
array([[1., 0., 0., 0., 0.]], dtype=float32)
```

```
# For visualizing class index

xtrain.class_indices
```

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

```
# predicting and Index Matching

op = ['daisy','dandelion','rose','sunflower','tulip']
pred = np.argmax(model.predict(x))
op[pred]
```

```
'daisy'
```

Colab paid products - Cancel contracts here