Unzip the dataset

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

```
inflating: flowers/daisy/15853110333_229c439e7f.jpg
inflating: flowers/daisy/158869618_f1a6704236_n.jpg
inflating: flowers/daisy/16020253176_60f2a6a5ca_n.jpg
inflating: flowers/daisy/16025261368_911703a536_n.jpg
inflating: flowers/daisy/16056178001_bebc2153fe_n.jpg
inflating: flowers/daisy/16121105382_b96251e506_m.jpg
inflating: flowers/daisy/16161045294_70c76ce846_n.jpg
inflating: flowers/daisy/162362896_99c7d851c8_n.jpg
inflating: flowers/daisy/162362897_1d21b70621_m.jpg
inflating: flowers/daisy/16291797949_a1b1b7c2bd_n.jpg
inflating: flowers/daisy/16323838000_3818bce5c6_n.jpg
inflating: flowers/daisy/16360180712_b72695928c_n.jpg
inflating: flowers/daisy/163978992_8128b49d3e_n.jpg
inflating: flowers/daisy/16401288243_36112bd52f_m.jpg
inflating: flowers/daisy/16482676953_5296227d40_n.jpg
inflating: flowers/daisy/16492248512_61a57dfec1_m.jpg
inflating: flowers/daisy/16527403771_2391f137c4_n.jpg
inflating: flowers/daisy/16577886423_9b23622f1d_n.jpg
inflating: flowers/daisy/16737503507_431768a927.jpg
inflating: flowers/daisy/16819071290_471d99e166_m.jpg
inflating: flowers/daisy/16833748795_b681b2839f_n.jpg
inflating: flowers/daisy/169371301_d9b91a2a42.jpg
inflating: flowers/daisy/17027891179_3edc08f4f6.jpg
inflating: flowers/daisy/17101762155_2577a28395.jpg
inflating: flowers/daisy/171972704_389cf7a953.jpg
inflating: flowers/daisy/17249393016_093e915012_n.jpg
inflating: flowers/daisy/172882635_4cc7b86731_m.jpg
inflating: flowers/daisy/17357636476_1953c07aa4_n.jpg
inflating: flowers/daisy/174131220_c853df1287.jpg
inflating: flowers/daisy/175106495_53ebdef092_n.jpg
inflating: flowers/daisy/176375506_201859bb92_m.jpg
inflating: flowers/daisy/17821980772_35164ae1e8_n.jpg
inflating: flowers/daisy/18023717391_e2c9089e10.jpg
inflating: flowers/daisy/181007802_7cab5ee78e_n.jpg
inflating: flowers/daisy/18195689904_46619b7e16_n.jpg
inflating: flowers/daisy/18203367608_07a04e98a4_n.jpg
inflating: flowers/daisy/18354545086_693ea7bc2a.jpg
inflating: flowers/daisy/18400014056_2e4c601ed5.jpg
inflating: flowers/daisy/18442919723_d1251d3e14_n.jpg
inflating: flowers/daisy/18474740346_ffdaa18032.jpg
inflating: flowers/daisy/18582579815_4c6637e9ff_m.jpg
inflating: flowers/daisy/18622672908_eab6dc9140_n.jpg
inflating: flowers/daisy/18635898912_eb8e058ef0.jpg
inflating: flowers/daisy/18679421522_3be9879e32.jpg
inflating: flowers/daisy/18684594849_7dd3634f5e_n.jpg
inflating: flowers/daisy/18711159980_11d3bd5042.jpg
inflating: flowers/daisy/1879567877_8ed2a5faa7_n.jpg
inflating: flowers/daisy/18901817451_43e2b45f6c.jpg
```

```
inflating: flowers/daisy/19019544592_b64469bf84_n.jpg
inflating: flowers/daisy/19177263840_6a316ea639.jpg
inflating: flowers/daisy/19178753159_a471bf4b6b.jpg
inflating: flowers/daisy/19280272025_57de24e940_m.jpg
inflating: flowers/daisy/19527362416_8bdcbefb8b_n.jpg
inflating: flowers/daisy/19544831049_0d738d4872_m.jpg
inflating: flowers/daisy/1955336401_fbb206d6ef_n.jpg
inflating: flowers/daisy/19653086178_28156b7ce4_m.jpg
inflating: flowers/daisy/19813618946_93818db7aa_m.jpg
inflating: flowers/daisy/19834392829_7d697871f6.jpg
inflating: flowers/daisy/19865728236_a62f8f445b_n.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'
```