Import and Unzip the Dataset

↳ *2 cells hidden*

Image Augmentation

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


#Data augmentation on training variable
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)


#Data augmentation on testing variable
test_datagen = ImageDataGenerator(rescale=1./255)


#Data augmentation on training data
xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64,64),
                                           class_mode='categorical',
                                           batch_size=100)
```

```
Found 3384 images belonging to 5 classes.
```

Import Layers

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

Add CNN Layers

```
model = Sequential()
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(5,activation='softmax'))
```

Compile the Model

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

```
model.fit(xtrain,steps_per_epoch=len(xtrain),epochs=10)
```

```
Epoch 1/10
34/34 [==============================] - 23s 646ms/step - loss: 1.7948 - accuracy: 0
Epoch 2/10
34/34 [==============================] - 22s 644ms/step - loss: 1.1695 - accuracy: 0
Epoch 3/10
34/34 [==============================] - 22s 641ms/step - loss: 1.0633 - accuracy: 0
Epoch 4/10
34/34 [==============================] - 26s 759ms/step - loss: 1.0108 - accuracy: 0
Epoch 5/10
34/34 [==============================] - 22s 637ms/step - loss: 0.9663 - accuracy: 0
Epoch 6/10
34/34 [==============================] - 22s 642ms/step - loss: 0.8937 - accuracy: 0
Epoch 7/10
34/34 [==============================] - 22s 637ms/step - loss: 0.8554 - accuracy: 0
Epoch 8/10
34/34 [==============================] - 22s 644ms/step - loss: 0.8235 - accuracy: 0
Epoch 9/10
34/34 [==============================] - 22s 643ms/step - loss: 0.7987 - accuracy: 0
Epoch 10/10
34/34 [==============================] - 22s 639ms/step - loss: 0.7775 - accuracy: 0
<keras.callbacks.History at 0x7fdc3ba7b090>
```

Save Model

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

Testing Model

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

Testdata:Daisy

```
img = image.load_img('/content/flowers/rose/10090824183_d02c613f10_m.jpg',target_size=(64,
img
```

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

```
array([[[14., 22.,  7.],
        [11., 22.,  6.],
        [ 8., 19.,  3.],
        ...,
        [32., 47., 24.],
        [30., 48., 22.],
        [33., 49., 23.]],

       [[13., 20., 12.],
        [11., 21., 10.],
        [11., 22.,  8.],
        ...,
        [37., 51., 26.],
        [35., 49., 26.],
        [25., 45., 20.]],

       [[19., 30., 16.],
        [19., 31., 17.],
        [16., 29., 12.],
        ...,
        [31., 47., 20.],
        [28., 49., 18.],
        [27., 43., 17.]],

       ...,

       [[15., 17.,  6.],
        [ 2.,  9.,  2.],
        [ 2.,  9.,  1.],
        ...,
        [ 8., 21., 11.],
        [ 2., 12.,  3.],
        [ 9., 16.,  9.]],

       [[12., 20.,  9.],
        [ 1.,  8.,  1.],
        [ 5., 10.,  3.],
        ...,
        [ 3.,  8.,  2.],
        [ 6., 16.,  5.],
        [ 5.,  7.,  4.]],

       [[24., 27., 18.],
        [11., 21., 13.],
        [ 8., 13.,  6.],
        ...,
        [ 1.,  6.,  0.],
        [ 2.,  9.,  1.],
        [ 2.,  9.,  1.]]], dtype=float32)
```

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

```
x
model.predict(x)
```

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

```
xtrain.class_indices
```

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

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

```
'daisy'
```

Test data:Rose

```
img = image.load_img('/content/flowers/rose/102501987_3cdb8e5394_n.jpg',target_size=(64,64
img
```

```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
outp[pred]
```

```
'rose'
```

Test data:Sunflower

```
img = image.load_img('/content/flowers/sunflower/1022552036_67d33d5bd8_n.jpg',target_size=
img
```

```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
outp[pred]
```

```
'sunflower'
```

Test data:Tulip

```
img = image.load_img('/content/flowers/tulip/10128546863_8de70c610d.jpg',target_size=(64,6
img
```

```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
outp[pred]
```

```
'tulip'
```

Test data:Dandelion

```
img = image.load_img('/content/flowers/dandelion/11405573_24a8a838cc_n.jpg',target_size=(6
img
```

```
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
outp[pred]
```

```
'dandelion'
```

Colab paid products - Cancel contracts here

✓ 0s completed at 10:46 PM