Importing Dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

Image Augmentation

```
# Importing Library
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# expanding training and testing variable
train_d=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True)
test_d=ImageDataGenerator(rescale=1./255)
```

```
#Data augmentation on testing data
vtrain = train_d.flow_from_directory('/content/drive/MyDrive/flowers/Testing',target_size=
```

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

```
#Data augmentation on training data
vtest = test_d.flow_from_directory('/content/drive/MyDrive/flowers/Training',target_size=(
```

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

Creating CNN Model

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

```
#Building a CNN block
model = Sequential()
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(76,76,3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(500,activation='relu'))
model.add(Dense(250,activation='relu'))
model.add(Dense(5,activation='softmax'))
```

```
#Compiling the model

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


#Fittting the model

model.fit_generator(vtrain,steps_per_epoch=len(vtrain),epochs=15,validation_data=vtest,val
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: `Model.f
  This is separate from the ipykernel package so we can avoid doing imports until
Epoch 1/15
22/22 [==============================] - 96s 4s/step - loss: 1.2093 - accuracy: 0.503
Epoch 2/15
22/22 [==============================] - 69s 3s/step - loss: 1.0804 - accuracy: 0.581
Epoch 3/15
22/22 [==============================] - 69s 3s/step - loss: 1.0100 - accuracy: 0.608
Epoch 4/15
22/22 [==============================] - 69s 3s/step - loss: 0.9307 - accuracy: 0.646
Epoch 5/15
22/22 [==============================] - 70s 3s/step - loss: 0.8715 - accuracy: 0.662
Epoch 6/15
22/22 [==============================] - 70s 3s/step - loss: 0.8210 - accuracy: 0.696
Epoch 7/15
22/22 [==============================] - 70s 3s/step - loss: 0.7659 - accuracy: 0.716
Epoch 8/15
22/22 [==============================] - 72s 3s/step - loss: 0.7307 - accuracy: 0.724
Epoch 9/15
22/22 [==============================] - 70s 3s/step - loss: 0.7181 - accuracy: 0.735
Epoch 10/15
22/22 [==============================] - 71s 3s/step - loss: 0.6603 - accuracy: 0.752
Epoch 11/15
22/22 [==============================] - 70s 3s/step - loss: 0.6256 - accuracy: 0.775
Epoch 12/15
22/22 [==============================] - 71s 3s/step - loss: 0.5942 - accuracy: 0.777
Epoch 13/15
22/22 [==============================] - 70s 3s/step - loss: 0.5693 - accuracy: 0.795
Epoch 14/15
22/22 [==============================] - 70s 3s/step - loss: 0.5345 - accuracy: 0.805
Epoch 15/15
22/22 [==============================] - 70s 3s/step - loss: 0.5097 - accuracy: 0.813
<keras.callbacks.History at 0x7fb99243cd90>
```

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

Testing model

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


# Testing 1.1(daisy)
```

```
img = image.load_img('/content/drive/MyDrive/flowers/Testing/daisy/10993818044_4c19b86c82.
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'
```

```
# Testing 1.2(daisy)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/daisy/525780443_bba812c26a_m.
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'
```

```
# Testing 2.1(dandelion)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/dandelion/1195255751_d58b3d30
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'
```

```
# Testing 2.2(dandelion)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/dandelion/1297972485_33266a18
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'
```

```
# Testing 3.1(rose)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/rose/7456887736_54e4ebac03_n.
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'
```

```
# Testing 3.2(rose)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/rose/33411423082_8150d9254e_n
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'
```

```
# Testing 4.1(sunflower)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/sunflower/7012364067_5ffc7654
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'
```

```
# Testing 4.2(sunflower)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/sunflower/2720698862_486d3ec0
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'
```

```
# Testing 5.1(tulip)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/tulip/8892851067_79242a7362_n
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'
```

```
# Testing 5.2(tulip)

img = image.load_img('/content/drive/MyDrive/flowers/Testing/tulip/5546723510_39a5a10d3a_n
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'
```