Build CNN Model for Classication Of Flowers

Download the dataset here.

```
# Unzip data
!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/104290366_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
```

1. Image Augmentation

```
#import lib.

from tensorflow.keras.preprocessing.image import ImageDataGenerator


#augmentation on flowers

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


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


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

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

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

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

2. Create Model

```
#import lib.

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

3. Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers),Output)

```
# Add a layers

model = Sequential() # Initializing sequential model
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3))) # convolution l
model.add(MaxPooling2D(pool_size=(2, 2))) # Max pooling layer
model.add(Flatten()) # Flatten layer
model.add(Dense(300,activation='relu')) # Hidden layer 1
model.add(Dense(150,activation='relu')) # Hidden layer 2
model.add(Dense(5,activation='softmax')) # Output layer
```

4. Compile The Model

```
# Compiling the model

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

5. Fit The Model

- List item

```
model.fit_generator(xrose,
                    steps_per_epoch=len(xrose),
                    epochs=10,
                    validation_data=xtulip,
                    validation_steps=len(xtulip))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: `Model.f
  """
Epoch 1/10
```

```
44/44 [==============================] - 38s 853ms/step - loss: 2.0894 - accuracy: 0
Epoch 2/10
44/44 [==============================] - 36s 821ms/step - loss: 1.1907 - accuracy: 0
Epoch 3/10
44/44 [==============================] - 35s 809ms/step - loss: 1.1105 - accuracy: 0
Epoch 4/10
44/44 [==============================] - 42s 953ms/step - loss: 1.0239 - accuracy: 0
Epoch 5/10
44/44 [==============================] - 35s 805ms/step - loss: 0.9284 - accuracy: 0
Epoch 6/10
44/44 [==============================] - 37s 841ms/step - loss: 0.8882 - accuracy: 0
Epoch 7/10
44/44 [==============================] - 35s 804ms/step - loss: 0.8362 - accuracy: 0
Epoch 8/10
44/44 [==============================] - 35s 804ms/step - loss: 0.8018 - accuracy: 0
Epoch 9/10
44/44 [==============================] - 36s 822ms/step - loss: 0.7742 - accuracy: 0
Epoch 10/10
44/44 [==============================] - 35s 798ms/step - loss: 0.7536 - accuracy: 0
<keras.callbacks.History at 0x7fd857224cd0>
```

6. Save The Model

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

7. Test The Model

```
from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt


#testing 1
img = image.load_img('/content/flowers/sunflower/12471443383_b71e7a7480_m.jpg',target_size
x = image.img_to_array(img) # Converting image into array
x = np.expand_dims(x,axis=0) # expanding Dimensions
pred = np.argmax(model.predict(x)) # Predicting the higher probablity index
op = ['daisy','dandelion','rose','sunflower','tulip'] # Creating list
op[pred] # List indexing with output
```

```
'sunflower'
```

```
img = image.load_img('/content/flowers/sunflower/12471443383_b71e7a7480_m.jpg',target_size
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7fd85279e450>
```


```
img = image.load_img('/content/flowers/rose/14145188939_b4de638bd3_n.jpg',target_size=(102
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7fd8522e9310>
```