

```

# -*- coding: utf-8 -*-
"""Pon muthu aravind -Assignment3.ipynb

Automatically generated by Colaboratory.

Original file is located at
https://colab.research.google.com/drive/1SVxTBy6ApNgJhbhvIcYuWs-ud2TyIoKD
"""

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

"""#Image Augmentation"""

from tensorflow.keras.preprocessing.image import ImageDataGenerator

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

test_datagen = ImageDataGenerator(rescale=1./255)

xtrain = train_datagen.flow_from_directory('/content/flowers',
                                           target_size=(64, 64),
                                           class_mode='categorical',
                                           batch_size=100)

xtest = test_datagen.flow_from_directory('/content/flowers',
                                          target_size=(64, 64),
                                          class_mode='categorical',
                                          batch_size=100)

"""#Create Model"""

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

model = Sequential()

"""#Convolution Layer"""

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

"""#MaxPooling"""

model.add(MaxPooling2D(pool_size=(2, 2)))

"""#Flatten"""

model.add(Flatten())

"""#Dense Layer"""

model.add(Dense(300, activation='relu')) #hiddenlayer 1
model.add(Dense(150, activation='relu')) #hiddenlayer 2

"""#Output"""

model.add(Dense(5, activation='softmax'))

"""#Compile the model"""

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

"""#Train"""

from keras.callbacks import EarlyStopping, ReduceLROnPlateau

early_stopping = EarlyStopping(monitor='val_accuracy',
                               patience=5)
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy',
                              patience=5,
                              factor=0.5, min_lr=0.00001)

callback = [reduce_lr, early_stopping]

```

```

model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=100,
                    callbacks=callback,
                    validation_data=xtest,
                    validation_steps=len(xtest))

"""#Saving"""

model.save('Flowers.h5')

"""#Testing the model"""

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

img = image.load_img('/content/flowers/sunflower/10386503264_e05387e1f7_m.jpg',target_size=(64,64))

img

"""#Array Conversion"""

x = image.img_to_array(img)
x

"""Expanding dimensions"""

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

"""#Prediction"""

model.predict(x)

xtrain.class_indices

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

img = image.load_img('/content/flowers/daisy/10466558316_a7198b87e2.jpg',target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x))
op[pred]

```