from google.colab import drive

drive.mount('/content/drive')

ls

cd /content/drive/MyDrive/CNN

pwd

##unzipping the zip file

!unzip 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,vertical\_flip=True)

test\_datagen=ImageDataGenerator(rescale=1./255)

x\_train=train\_datagen.flow\_from\_directory(r"/content/drive/MyDrive/CNN/flowers",target\_size=(64,64),class\_mode="categorical",batch\_size=24)

x\_test=test\_datagen.flow\_from\_directory(r"/content/drive/MyDrive/CNN/flowers",target\_size=(64,64),class\_mode="categorical",batch\_size=24)

x\_train.class\_indices

## Creating The Model

from tensorflow.keras.models import Sequential

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

model=Sequential()

## Adding The Layers

##Adding Convolution2D Layer

model.add(Convolution2D(32,(3,3),activation="relu",strides=(1,1),input\_shape=(64,64,3)))

##Adding MaxPooling2D Layer

model.add(MaxPooling2D(pool\_size=(2,2)))

##Adding Flatten Layer

model.add(Flatten())

##Adding Dense-(Hidden Layers)

model.add(Dense(300,activation="relu"))

model.add(Dense(300,activation="relu"))

##Adding Output Layer

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

##To see the added layers

model.summary()

## Compiling The Model

model.compile(loss="categorical\_crossentropy",optimizer="adam",metrics=['accuracy'])

## Fitting The Model

len(x\_train)

model.fit(x\_train,epochs=10,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validation\_steps=len(x\_test))

## Saving The Model

model.save('flowers.h5')

## Testing The Model

import numpy as np

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

model=load\_model('flowers.h5')

img=image.load\_img(r"/content/drive/MyDrive/CNN/flowers/rose/537207677\_f96a0507bb.jpg")

img

img=image.load\_img(r"/content/drive/MyDrive/CNN/flowers/rose/537207677\_f96a0507bb.jpg",target\_size=(64,64))

img

x=image.img\_to\_array(img)

x

x=np.expand\_dims(x,axis=0)

x

pred=model.predict(x)

pred

x\_test.class\_indices

index=['daisy','dandelion','rose','sunflower','tulip']

index[np.argmax(pred)]