Assignment -3 CONVOLUTIONAL NEURAL NETWORKS

Assignment Date	5 October 2022
Student Name	N.Dharshinee
Student Roll Number	9517201906014
Maximum Marks	2 Marks

#Import necessary libraries

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Flatten

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

#Image augmentation

from tensorflow.keras.preprocessing.image import ImageDataGenerator train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_ran ge=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/flowers/
Training",target_size=(128,128),batch_size=32,class_mode="categorical")
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/flowers/T
esting",target_size=(128,128),batch_size=32,class_mode="categorical")
x_train.class_indices



```
model = Sequential()
#Add layers
#Convolution layer
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
#Maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
#flatten layer
model.add(Flatten())
#hidden layer
model.add(Dense(units=300,kernel_initializer="random_uniform",activation="relu"))
model.add(Dense(units=200,kernel_initializer="random_uniform",activation="relu"))
model.add(Dense(units=5,kernel_initializer="random_uniform",activation="softmax"))
model.summary()
```

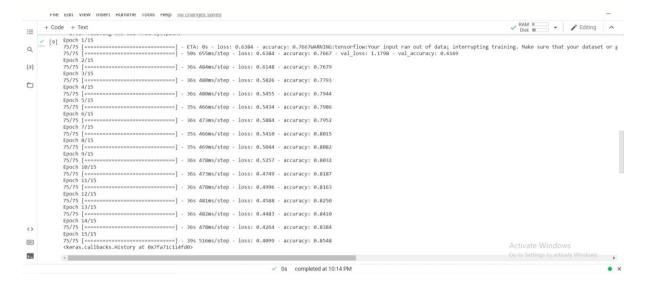


#compile the model

model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["a ccuracy"])

#Fit the model

model.fit_generator(x_train,steps_per_epoch=75,epochs=15,validation_data=x_test,validation_steps=80)



#Save the model model.save("flower.h5")

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
model = load_model("flower.h5")
#Testing with the image
img = image.load_img(r"/content/drive/MyDrive/daisy.jpg",target_size=(128,12
8))
img
type(img)
✓ RAM ■ ✓ ✓ Editing ^
[10] #Save the model
q model.save("flower.h5")
[x] \( \text{[12]} \) from tensorflow.keras.models import load_model from tensorflow.keras.preprocessing import imaging import numpy as np model = load_model("flower.h5")
   [13] #Testing with the image
img = image.load_img(r"/content/drive/MyOrive/daisy.jpg",target_size=(128,128))
img
type(img)
   [14] x = image.img_to_array(img)
       x x.shape x = np.expand_dims(x,axis=0) x.shape
       (1, 128, 128, 3)
array([[1., 0., 0., 0., 0.]], dtype=float32)

✓ 0s completed at 10:14 PM

x = image.img\_to\_array(img)
X
x.shape
x = np.expand\_dims(x,axis=0)
x.shape
pred_prob = model.predict(x)
pred_prob
class_name = ["daisy","dandelion","rose","sunfower","tulip"]
pred_id = pred_prob.argmax(axis=1)[0]
pred_id
print("Predicted flower is",str(class_name[pred_id]))
```

✓ RAM □ ✓ ✓ Editing ∧ (1, 128, 128, 3) [15] pred_prob = model.predict(x) pred_prob array([[1., 0., 0., 0., 0.]], dtype=float32) [16] class_name = ["daisy","dandelion","rose","sunfower","tulip"] ↑ ↓ © **□ ‡** 🖟 🖥 : pred_id = pred_prob.argmax(axis=1)[0]
pred_id
print("Predicted flower is",str(class_name[pred_id])) Predicted flower is daisy [] <> \equiv Activate Windows >_

✓ 0s completed at 10:14 PM

_ ^