**Assignment -3**

# Build CNN Model for classification of Flowers

|  |  |
| --- | --- |
| Assignment Date | 19 September 2022 |
| Student Name | Rubanchinnarathinam |
| Student Roll Number | 210519205042 |
| Maximum Marks | 2 Marks |

**Question 1:**

**Download Dataset**

from google.colab import drive

drive.mount('/content/drive')

Mounted at /content/drive

**Question 2:**

# 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)

## Load Data

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

Found 4317 images belonging to 5 classes.

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

Found 4317 images belonging to 5 classes.

x\_train.class\_indices

{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}

x\_test.class\_indices

{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}

**Question 3:**

# Create Model

from tensorflow.keras.models import Sequential

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

model=Sequential()

**Question 4:**

# Add Layers

a)Convolution Layer

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

b)MaxPooling Layer

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

c)Flatten

model.add(Flatten())

d)Dense(Hidden layer)

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

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

e)Output layer

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

**Question 5:**

**Compile the Model**

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

**Question 6:**

**Fit the Model**

model.fit(x\_train,epochs=5,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validation\_steps=len(x\_test))

Epoch 1/5

180/180 [==============================] - 613s 3s/step - loss: 1.1807 - accuracy: 0.5071 - val\_loss: 1.0645 - val\_accuracy: 0.5698

Epoch 2/5

180/180 [==============================] - 67s 375ms/step - loss: 1.0647 - accuracy: 0.5726 - val\_loss: 1.0555 - val\_accuracy: 0.5837

Epoch 3/5

180/180 [==============================] - 68s 377ms/step - loss: 1.0042 - accuracy: 0.6013 - val\_loss: 0.9352 - val\_accuracy: 0.6391

Epoch 4/5

180/180 [==============================] - 66s 368ms/step - loss: 0.9319 - accuracy: 0.6379 - val\_loss: 0.9133 - val\_accuracy: 0.6530

Epoch 5/5

180/180 [==============================] - 66s 370ms/step - loss: 0.8710 - accuracy: 0.6606 - val\_loss: 0.9661 - val\_accuracy: 0.6375

<keras.callbacks.History at 0x7fc655b9d350>

**Question 7:**

**Save the Model**

model.save("Flowers.h5")

**Question 8:**

**Test 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/Data/Flowers/flowers/sunflower/1008566138\_6927679c8a.jpg",target\_size=(64,64))

img

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

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

pred=model.predict(x)

pred

array([[0., 0., 0., 1., 0.]], dtype=float32)

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

index[np.argmax(pred)]

'sunflower'