

Assignment -3
CONVOLUTIONAL NEURAL NETWORKS

Assignment Date	5 October 2022
Student Name	K.Redem Emima
Student Roll Number	9517201903120
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_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/flowers/Training", target_size=(128,128), batch_size=32, class_mode="categorical")
x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/flowers/Testing", target_size=(128,128), batch_size=32, class_mode="categorical")
x_train.class_indices
```

```
[4] #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')
Mounted at /content/drive

[5] #Image augmentation
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, vertical_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)

[6] 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/Testing", target_size=(128,128), batch_size=32, class_mode="categorical")
x_train.class_indices

Found 3023 images belonging to 5 classes.
Found 1326 images belonging to 5 classes.
{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}

[7] model = Sequential()
#Add layers
```

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()

```

#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()

```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
flatten (Flatten)	(None, 127008)	0
dense (Dense)	(None, 300)	38102700
dense_1 (Dense)	(None, 200)	60200
dense_2 (Dense)	(None, 5)	1005
Total params: 38,164,801		
Trainable params: 38,164,801		
Non-trainable params: 0		

Model: "sequential"

#compile the model

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

#Fit the model

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

```

Epoch 1/15
75/75 [=====] - ETA: 0s - loss: 0.6384 - accuracy: 0.7667WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least one more batch
75/75 [=====] - 50s 655ms/step - loss: 0.6384 - accuracy: 0.7667 - val_loss: 1.1798 - val_accuracy: 0.6169
Epoch 2/15
75/75 [=====] - 36s 484ms/step - loss: 0.6148 - accuracy: 0.7679
Epoch 3/15
75/75 [=====] - 36s 480ms/step - loss: 0.5826 - accuracy: 0.7793
Epoch 4/15
75/75 [=====] - 36s 480ms/step - loss: 0.5455 - accuracy: 0.7944
Epoch 5/15
75/75 [=====] - 35s 466ms/step - loss: 0.5434 - accuracy: 0.7986
Epoch 6/15
75/75 [=====] - 36s 473ms/step - loss: 0.5884 - accuracy: 0.7952
Epoch 7/15
75/75 [=====] - 35s 466ms/step - loss: 0.5410 - accuracy: 0.8015
Epoch 8/15
75/75 [=====] - 35s 469ms/step - loss: 0.5044 - accuracy: 0.8082
Epoch 9/15
75/75 [=====] - 36s 478ms/step - loss: 0.5257 - accuracy: 0.8032
Epoch 10/15
75/75 [=====] - 36s 473ms/step - loss: 0.4749 - accuracy: 0.8187
Epoch 11/15
75/75 [=====] - 36s 478ms/step - loss: 0.4996 - accuracy: 0.8163
Epoch 12/15
75/75 [=====] - 36s 481ms/step - loss: 0.4588 - accuracy: 0.8250
Epoch 13/15
75/75 [=====] - 36s 482ms/step - loss: 0.4483 - accuracy: 0.8410
Epoch 14/15
75/75 [=====] - 36s 478ms/step - loss: 0.4264 - accuracy: 0.8384
Epoch 15/15
75/75 [=====] - 39s 516ms/step - loss: 0.4099 - accuracy: 0.8548
<keras.callbacks.History at 0x7fa71c114fd0>

```

#Save the model

model.save("flower.h5")

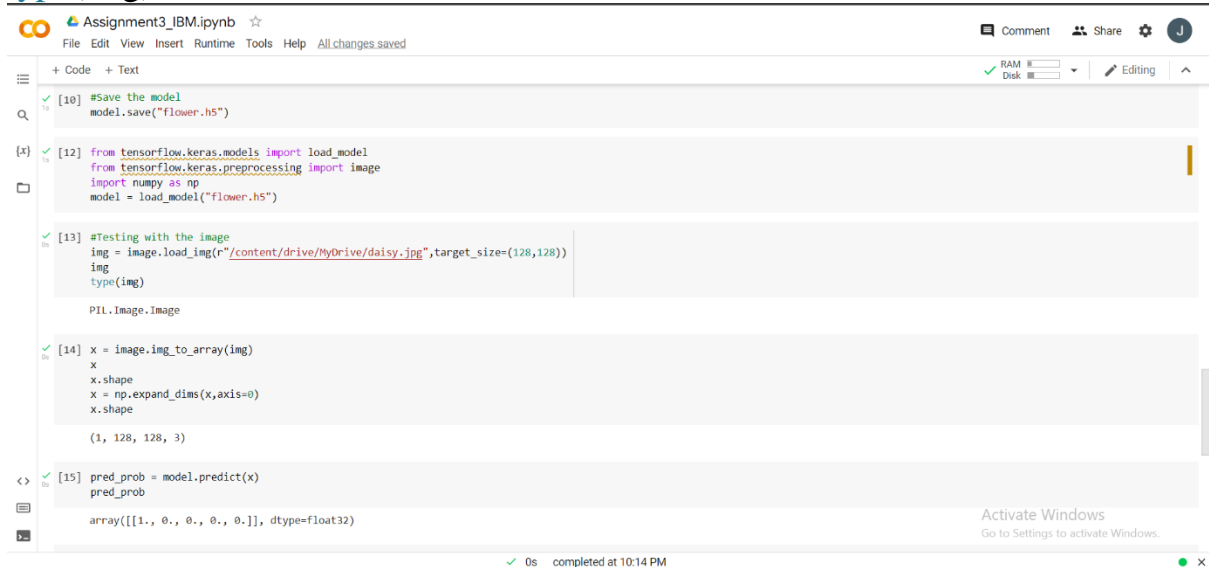
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
model = load_model("flower.h5")
```

```
#Testing with the image
```

```
img = image.load_img(r"/content/drive/MyDrive/daisy.jpg",target_size=(128,128))
```

```
img
```

```
type(img)
```



The screenshot shows a Jupyter Notebook titled "Assignment3_IBM.ipynb". The code is executed in five cells:

- Cell 10: `model.save("flower.h5")`
- Cell 12: `from tensorflow.keras.models import load_model`, `from tensorflow.keras.preprocessing import image`, `import numpy as np`, `model = load_model("flower.h5")`
- Cell 13: `#Testing with the image`, `img = image.load_img(r"/content/drive/MyDrive/daisy.jpg",target_size=(128,128))`, `img`, `type(img)`. The output is `PIL.Image.Image`.
- Cell 14: `x = image.img_to_array(img)`, `x`, `x.shape`, `x = np.expand_dims(x,axis=0)`, `x.shape`. The output is `(1, 128, 128, 3)`.
- Cell 15: `pred_prob = model.predict(x)`, `pred_prob`. The output is `array([[1., 0., 0., 0., 0.]], dtype=float32)`.

The bottom status bar indicates "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","sunflower","tulip"]
```

```
pred_id = pred_prob.argmax(axis=1)[0]
```

```
pred_id
```

```
print("Predicted flower is",str(class_name[pred_id]))
```

Assignment3_IBM.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM
Disk

Editing

[14]

✓

```
x = image.img_to_array(img)
x
x.shape
x = np.expand_dims(x,axis=0)
x.shape
```

(1, 128, 128, 3)

[15]

✓

```
pred_prob = model.predict(x)
pred_prob
```

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

[16]

✓

```
class_name = ["daisy","dandelion","rose","sunflower","tulip"]
```

↑ ↓ ↺ ⌂ ⚙ 🗑 ⋮

```
pred_id = pred_prob.argmax(axis=1)[0]
pred_id
print("Predicted flower is",str(class_name[pred_id]))
```

Predicted flower is daisy

[]

<>

Activate Windows
Go to Settings to activate Windows.

0s completed at 10:14 PM

● x