

Assignment -3

Python Programming

Assignment Date	19 September 2022
Student Name	R.Mohamed Imran
Student Roll Number	210519205031
Maximum Marks	2 Marks

```
ls
[1] Python

... drive/ sample_data/

cd /content/drive/MyDrive/cnn
[6] Python

... /content/drive/MyDrive/cnn

pwd
[3] Python

... '/content/drive/MyDrive/cnn'

!unzip Animal_Dataset.zip
[7] Python

... Output exceeds the size limit. Open the full output data in a text editor
Archive: Animal_Dataset.zip
  creating: dataset/
  creating: dataset/Testing/
  creating: dataset/Testing/bears/
 inflating: dataset/Testing/bears/k4 (100).jpeg
 inflating: dataset/Testing/bears/k4 (100).jpg
 inflating: dataset/Testing/bears/k4 (101).jpeg
 inflating: dataset/Testing/bears/k4 (101).jpg
 inflating: dataset/Testing/bears/k4 (102).jpeg
 inflating: dataset/Testing/bears/k4 (102).jpg
 inflating: dataset/Testing/bears/k4 (103).jpeg
 inflating: dataset/Testing/bears/k4 (104).jpeg
 inflating: dataset/Testing/bears/k4 (105).jpeg
 inflating: dataset/Testing/bears/k4 (106).jpeg
 inflating: dataset/Testing/bears/k4 (107).jpeg
 inflating: dataset/Testing/bears/k4 (108).jpeg
 inflating: dataset/Testing/bears/k4 (109).jpeg
 inflating: dataset/Testing/bears/k4 (110).jpeg
 inflating: dataset/Testing/bears/k4 (71).jpg
 inflating: dataset/Testing/bears/k4 (74).jpeg
 inflating: dataset/Testing/bears/k4 (74).jpg
 ...
 inflating: dataset/Training/rats/Z (7).jpeg
 inflating: dataset/Training/rats/Z (8).jpeg
 inflating: dataset/Training/rats/Z (9).jpeg
 inflating: dataset/Training/rats/Z.jpeg

Image Augmentation

from tensorflow.keras.preprocessing.image import ImageDataGenerator
[8] Python

train_datagen = ImageDataGenerator(rescale = 1./255, zoom_range = 0.2, horizontal_flip=True, vertical_flip=True)
[9] Python
```

```

[10] test_datagen = ImageDataGenerator(rescale = 1./255)
Python

[11] x_train = train_datagen.flow_from_directory(r"/content/drive/MyDrive/cnn/dataset/Training",target_size=(64,64),class_mode = "categorical",batch_si:
Python
... Found 1238 images belonging to 4 classes.

[30] x_test = test_datagen.flow_from_directory(r"/content/drive/MyDrive/cnn/dataset/Testing",target_size=(64,64),class_mode = "categorical",batch_si:
Python
... Found 326 images belonging to 4 classes.

[13] x_train.class_indices
Python
... {'bears': 0, 'crows': 1, 'elephants': 2, 'rats': 3}

CNN

[18] from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
Python

[16] # initialize
model = Sequential()
Python

[17] model.add(Convolution2D(32,(3,3),activation="relu",strides=(1,1),input_shape = (64,64,3)))
Python

[19] model.add(MaxPooling2D(pool_size=(2,2)))
Python

model.add(Flatten())

[22] model.summary()
Python
... Model: "sequential"

Layer (type)                 Output Shape                 Param #
=====
conv2d (Conv2D)              (None, 62, 62, 32)          896

max_pooling2d (MaxPooling2D) (None, 31, 31, 32)          0

flatten (Flatten)            (None, 30752)                0

=====
Total params: 896
Trainable params: 896
Non-trainable params: 0
=====

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

```

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

[23] Python

```
model.add(Dense(4,activation = "softmax"))
```

[24] Python

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

[25] Python

```
len(x_train)
```

[31] Python

... 14

```
model.fit(x_train, epochs = 10,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test))
```

[32] Python

... Epoch 1/10
14/14 [=====] - 7s 462ms/step - loss: 2.5809 - accuracy: 0.2791 - val_loss: 1.3510 - val_accuracy: 0.2699
Epoch 2/10
14/14 [=====] - 6s 415ms/step - loss: 1.2150 - accuracy: 0.4509 - val_loss: 0.9841 - val_accuracy: 0.7301
Epoch 3/10
14/14 [=====] - 7s 511ms/step - loss: 0.8646 - accuracy: 0.7393 - val_loss: 0.6057 - val_accuracy: 0.8344
Epoch 4/10
14/14 [=====] - 6s 427ms/step - loss: 0.5112 - accuracy: 0.8528 - val_loss: 0.3082 - val_accuracy: 0.9417
Epoch 5/10
14/14 [=====] - 6s 421ms/step - loss: 0.3095 - accuracy: 0.9018 - val_loss: 0.3219 - val_accuracy: 0.8988
Epoch 6/10
14/14 [=====] - 6s 432ms/step - loss: 0.2028 - accuracy: 0.9479 - val_loss: 0.1639 - val_accuracy: 0.9601
Epoch 7/10
14/14 [=====] - 6s 418ms/step - loss: 0.0996 - accuracy: 0.9847 - val_loss: 0.0515 - val_accuracy: 1.0000
Epoch 8/10
14/14 [=====] - 6s 425ms/step - loss: 0.0511 - accuracy: 1.0000 - val_loss: 0.0320 - val_accuracy: 1.0000
Epoch 9/10
14/14 [=====] - 6s 419ms/step - loss: 0.0454 - accuracy: 0.9939 - val_loss: 0.0206 - val_accuracy: 1.0000
Epoch 10/10
14/14 [=====] - 6s 424ms/step - loss: 0.0270 - accuracy: 1.0000 - val_loss: 0.0195 - val_accuracy: 1.0000

<keras.callbacks.History at 0x7fe265d01dd0>

```
model.save('animal.h5')
```

[38] Python

Python

Testing the model

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

[34] Python

```
model = load_model('animal.h5')
```

[39] Python

```
img = image.load_img(r"/content/drive/MyDrive/cnn/dataset/Testing/elephants/Z (13).jpeg")
```

img

[41]

Python

...



```
img = image.load_img(r"/content/drive/MyDrive/cnn/dataset/Testing/elephants/Z (13).jpeg",target_size=(64,64))
```

[42]

Python

img

[43]

Python

...



```
x = image.img_to_array(img)
```

[44]

Python

x

[45]

Python

... Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
array([[139., 137., 148.],
       [142., 140., 151.],
       [146., 144., 155.],
       ...,
       [ 86.,  68.,  56.],
       [ 63.,  46.,  36.],
       [ 79.,  62.,  54.]],

       [[141., 139., 150.],
       [144., 142., 153.],
       [147., 145., 156.],
       ...,
       [ 74.,  56.,  46.],
       [ 66.,  49.,  41.],
       [ 55.,  38.,  30.]],

       [[142., 140., 151.],
       [144., 142., 153.],
       [148., 146., 157.],
       ...,
       [ 77.,  60.,  50.],
       [ 53.,  38.,  31.],
       [ 55.,  38.,  30.]],

       ...,

       [[172., 156., 157.],
       ...,
       [ 63.,  46.,  36.],
       [ 61.,  44.,  34.],
       [ 55.,  38.,  30.]]], dtype=float32)
```

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

[49]

Python

x

[50]

Python

```
''' Output exceeds the size limit. Open the full output data in a text editor
```

```
array([[[[139., 137., 148.],
        [142., 140., 151.],
        [146., 144., 155.],
        ...,
        [ 86., 68., 56.],
        [ 63., 46., 36.],
        [ 79., 62., 54.]],
       [[141., 139., 150.],
        [144., 142., 153.],
        [147., 145., 156.],
        ...,
        [ 74., 56., 46.],
        [ 66., 49., 41.],
        [ 55., 38., 30.]],
       [[142., 140., 151.],
        [144., 142., 153.],
        [148., 146., 157.],
        ...,
        [ 77., 60., 50.],
        [ 53., 38., 31.],
        [ 55., 38., 30.]],
       ...,
       [[172., 156., 157.],
        ...,
        [ 63., 46., 36.],
        [ 61., 44., 34.],
        [ 55., 38., 30.]]]], dtype=float32)
```

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

```
[52]
```

```
Python
```

```
''' 1/1 [=====] - 0s 28ms/step
```

```
pred
```

```
[53]
```

```
Python
```

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

```
x_test.class_indices
```

```
[54]
```

```
Python
```

```
''' {'bears': 0, 'crows': 1, 'elephants': 2, 'rats': 3}
```

```
index = ['bears', 'crows', 'elephants', 'rats']
```

```
[55]
```

```
Python
```

```
index[np.argmax(pred)]
```

```
[56]
```

```
Python
```

```
''' 'elephants'
```

```
Python
```