6.3

July 13, 2021

1 Assignment 6.3

Load the ResNet50 model. Perform image classification on five to ten images of your choice. They can be personal images or publically available images. Include the images in dsc650/assignments/assignment06/images/. Save the predictions dsc650/assignments/assignment06/results/predictions/resnet50 directory. If you are using JupyterHub, you can include those plots in your Jupyter notebook.

[7]:

```
#pip install opencv-python
```

[8]:

```
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input,␣
 ↪decode_predictions
import numpy as np
import os, cv2
```

[9]:

```
model = ResNet50(weights = 'imagenet')

img_path = 'images'

images = os.listdir(img_path)

for i,name in enumerate(images):
    print(name)
```

```
.ipynb_checkpoints
scooter.jpg
basketball.jpg
deer.jpg
black_lab.jpg
flower.jpg
bear.jpg
2021_hyundai_kona.jpg
cheese.jpg
tree.jpg
```

[10]:

```
for i, name in enumerate(images):
    if name != '.ipynb_checkpoints':
        img = cv2.imread(img_path + '/' + name)
        img = cv2.resize(img, (224, 224))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis = 0)
        x = preprocess_input(x)

        preds = model.predict(x)

        decpr = name, decode_predictions(preds, top = 3)[0]

        print(decpr)

        with open('results/6_3_Predictions.txt', 'w') as f:
            f.write(decpr[0])
    else:
        pass
```

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/dat
a/imagenet_class_index.json
40960/35363 [==================================] - 0s 1us/step
('scooter.jpg', [('n03532672', 'hook', 0.1937823), ('n04336792', 'stretcher',
0.16099794), ('n02790996', 'barbell', 0.11712359)])
('basketball.jpg', [('n02802426', 'basketball', 0.83630806), ('n04118538',
'rugby_ball', 0.14065029), ('n04540053', 'volleyball', 0.015264203)])
('deer.jpg', [('n02423022', 'gazelle', 0.94556403), ('n02422699', 'impala',
0.0301733), ('n02415577', 'bighorn', 0.008969743)])
('black_lab.jpg', [('n02099712', 'Labrador_retriever', 0.93532854),
('n02099267', 'flat-coated_retriever', 0.03833932), ('n02100236', 'German_short-
haired_pointer', 0.00900713)])
('flower.jpg', [('n11939491', 'daisy', 0.9257678), ('n07730033', 'cardoon',
0.025526844), ('n02206856', 'bee', 0.006743652)])
('bear.jpg', [('n02132136', 'brown_bear', 0.8424757), ('n02133161',
'American_black_bear', 0.14036416), ('n02134418', 'sloth_bear', 0.0066179605)])
('2021_hyundai_kona.jpg', [('n03459775', 'grille', 0.37407413), ('n03930630',
'pickup', 0.3106482), ('n02814533', 'beach_wagon', 0.10853493)])
('cheese.jpg', [('n03291819', 'envelope', 0.28879088), ('n03944341', 'pinwheel',
0.10782532), ('n03938244', 'pillow', 0.09603379)])
('tree.jpg', [('n09332890', 'lakeside', 0.25150883), ('n09428293', 'seashore',
0.18795349), ('n09256479', 'coral_reef', 0.05809415)])
```

[ ]:

```
```