6.3 Assignment

October 11, 2020

0.0.1 Week6: Computer Vision

0.0.2 6.3 Assignment

Load the ResNet50 model and classify the images found in the data/raw/images directory. Save the predictions dsc650/assignments/assignment06/results/predictions/resnet50 directory.

1 Usage examples for image classification models

1.0.1 Classify ImageNet classes with ResNet50

https://keras.io/api/applications/#classify-imagenet-classes-with-resnet50

[19]:
```
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

model = ResNet50(weights='imagenet')

img1_path = 'images/elephant.jpg'
img2_path = 'images/dog.jpeg'
img3_path = 'images/monkey.jpg'
img4_path = 'images/classroom.jpg'
img5_path = 'images/cow.jpeg'
img6_path = 'images/house.jpeg'

img1 = image.load_img(img1_path, target_size=(224, 224))
img2 = image.load_img(img2_path, target_size=(224, 224))
img3 = image.load_img(img3_path, target_size=(224, 224))
img4 = image.load_img(img4_path, target_size=(224, 224))
img5 = image.load_img(img5_path, target_size=(224, 224))
img6 = image.load_img(img6_path, target_size=(224, 224))

images = [ img1, img2, img3, img4, img5, img6]

results = []
```

```
for img in images:

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

    preds = model.predict(x, verbose=0)
    print('Predicted:', decode_predictions(preds, top=3)[0])
    results.append(f'{decode_predictions(preds,␣
 ↪top=3)[0][0][1]}-{decode_predictions(preds, top=3)[0][0][2]}')

# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
#print('Predicted:', decode_predictions(preds, top=3)[0])
print(results)
np.savetxt("results/predictions/resnet50/score.csv",  results, delimiter=",",␣
 ↪fmt="%s")
# Predicted: [(u'n02504013', u'Indian_elephant', 0.82658225), (u'n01871265',␣
 ↪u'tusker', 0.1122357), (u'n02504458', u'African_elephant', 0.061040461)]
```

```
Predicted: [('n02504458', 'African_elephant', 0.7787162), ('n01871265',
'tusker', 0.12713987), ('n02504013', 'Indian_elephant', 0.09208985)]
Predicted: [('n02113799', 'standard_poodle', 0.7399097), ('n02093647',
'Bedlington_terrier', 0.1224053), ('n02088094', 'Afghan_hound', 0.06499452)]
Predicted: [('n02487347', 'macaque', 0.90523654), ('n02486261', 'patas',
0.054982003), ('n02484975', 'guenon', 0.014403992)]
Predicted: [('n04081281', 'restaurant', 0.50078684), ('n03661043', 'library',
0.40669793), ('n02871525', 'bookshop', 0.014488094)]
Predicted: [('n02403003', 'ox', 0.7208774), ('n03868242', 'oxcart', 0.22500475),
('n04604644', 'worm_fence', 0.009863633)]
Predicted: [('n02859443', 'boathouse', 0.3012838), ('n09332890', 'lakeside',
0.26672164), ('n03930313', 'picket_fence', 0.1399544)]
['African_elephant-0.7787162065505981', 'standard_poodle-0.7399097084999084',
'macaque-0.905236542224884', 'restaurant-0.5007868409156799',
'ox-0.7208774089813232', 'boathouse-0.3012838065624237']
```

```
[ ]:
```