

Image retrieval by example query image

Experiment with one query image

In [170... `%reset`

Once deleted, variables cannot be recovered. Proceed (y/[n])? y

In [171... `from PIL import Image`
`import torch`
`import matplotlib.pyplot as plt`
`import numpy as np`
`import os`
`import sys`
`from time import time`
`import pickle`

`experimentPath = r'D:\1_Code\F0E\ECE3086\lab1_cbir_student'`
`os.chdir(experimentPath)`
Lab related module
`from ai_pytorch_module import *`
`from cbir_module import * # LabelDic defined here`
`import cbir_module`
`from cbir_module import *`
`import importlib`
`importlib.reload(cbir_module)`

Out[171... `<module 'cbir_module' from 'D:\\1_Code\\F0E\\ECE3086\\lab1_cbir_student\\cbir_module.py'>`

In [172... *### Set Path*

`imgpath = r'\\.images'`
`sys.path.append(os.getcwd())`
`sys.path.append(imgpath)`

In [173... *### Load database*
Need this since pickle store a List of Database objects
Pickle need to refer to this class
`class Database :`
 `def __init__(self) :`
 `self.imageName = None`
 `self.featsCNN = None`

`with open("CBIR_database.pickle","rb") as f:`
 `dataDict = pickle.load(f)`
`database = dataDict['database']`

In [174... `print(database[1].featColorHist.shape)`
`print(database[1].featCNN.shape)`

(768,)
(1, 4096)

Question 6

Implement the following functions. Test your function with 1 image from each label category

1. retrievedID = doRetrieval(featsQuery , k, database, imgpath, showImage=True)
2. Precision_K = getPrecisionRank_K(k, queryLabel, retrievedID, database)

In [175...

```
# student code for function definition
# def showImageInfoFromDB(id, imgpath, database):
# def doRetrieval(featsQuery , k, database, imgpath, showImage=True):
# hint use np.argsort()
def showImageInfoFromDB(id, imgpath, database):
    # your code
    plt.imshow(im) , plt.axis('off')
    titleStr = " Image {}.jpg label = {} Label name = {}".format(str(id), label, Lab
    plt.title(titleStr)
```

In [176...

```
# hint use argsort()
def doRetrieval(featsQuery , k, database, imgpath, showImage=True):
    numImages = len(database)
    # your code
    return idx_k
```

In [177...

```
%% Test your code with the script in this cell for CNN feature

# Do retrieval by nearest neighbour search
# Use query by example

k=10 # select the top K image to be retrieved
queryID=101 # Select query image ID
featsQuery = database[queryID].featsCNN
print("Display Query Image id = ", queryID)
showImageInfoFromDB(queryID, imgpath, database)

featsQueryCNN = database[queryID].featsCNN
retrievedID = doRetrieval(featsQueryCNN , k, database, imgpath, showImage=True)
```

Display Query Image id = 101
 Image name = 101.jpg
 Label ID = 2
 Label Name = Beach
 Image 101.jpg label = 2 Label name = Beach



```
In [178... # student code for function definition
# def getPrecisionRank_K(k, queryLabel, retrievedID, database):
def getPrecisionRank_K(k, queryLabel, retrievedID, database):
    # your code
    return precision_k
#endfunc()
```

```
In [179... # Report the precision result
print("\n Experiment on CBIR with CNN feature as image feature")
queryLabel = database[queryID].classLabel
Precision_K = getPrecisionRank_K(k, queryLabel, retrievedID, database)
print(" Query image label :", queryLabel)
print("\n Precision when retrieving {} images for query image {} = {:.2f}".format(
```

Experiment on CBIR with CNN feature as image feature

Class label of retrieved img

2 2 2 2 2 2 2 2 2 2 Query image label : 2

Precision when retrieving 10 images for query image 101 = 1.000

```
In [180... ### Repeat the experiment above for colour histogram feature
# Your code
```

```
In [181... database[1].featColorHist.shape
```

Out[181... (768,)

```
In [182... def doRetrieval2(featQuery , k, database, imgpath, showImage=True):
    # your code
    return idx_k
```

```
In [183... print("\n Experiment on CBIR with color histogram as image feature")
k=10 # select the top K image to be retrieved
queryID=101 # Select query image ID
featQuery = database[queryID].featColorHist
retrievedID = doRetrieval2(featQuery, k, database, imgpath, showImage=True)
# Report the precision result
queryLabel = database[queryID].classLabel
Precision_K = getPrecisionRank_K(k, queryLabel, retrievedID, database)
print("Query image label :", queryLabel)
print("\n Precision when retrieving {} images for query image {} = {:.2f}".format(
```

Experiment on CBIR with color histogram as image feature

Class label of retrieved img

9 9 9 10 9 6 2 6 9 6 Query image label : 2

Precision when retrieving 10 images for query image 101 = 0.100

In []: