MULTIMEDIA UNIVERSITY®

FACULTY OF ENGINEERING

LAB REPORT SUBMISSION COVER PAGE

ECE3086 MULTIMEDIA TECHNOLOGY AND APPLICATION

TRIMESTER 1 SESSION 2021/2022

Student Name: WAN HAO ZHE

Student ID: 1171100294

Degree Major: EE / LE / CE / TE / ME / OPE / MCE / NANO / BMM

Declaration of originality:
I declare that all sentences, results and data mentioned in this report are from my own work. All work derived from other authors have been listed in the references. I understand that failure to do this is considered plagiarism and will be penalized.
Note that collaboration and discussions in conducting the experiments are allowed but copying and any act of cheating in the report, results and data are strictly prohibited

Student signature: ……………………………………….

Experiment title: MTA1 – Image Processing for Image Retrieval

Experiment Date: 14th September 2021

Date Submitted: 29th September 2021

Introduction

To investigate on the semantic gap problem in Content-based image retrieval (CBIR) system. CBIR system also known as Query by Image Content (QBIC), the technology allow user to search image for digital images in large database. The technique of CBIR retrieve the most visually similar images given to a query image from database. By using the feature vector used by image search engine allow to automatically extract feature vector from image as a searchable index. Feature vector describes the attributes of the obect that consist of its height and weight. For example, the feature vector v = [height weight] is a vector value. In order to retrieve similar images, the retrieval algotihm needed the query image compare with the dissimilarity of all the images in the database. One of the most common dissimilarity distance metric function is Euclidian distance. By calculation the Euclidian distance between two feature vector of query image and the one of the dissimilarity images in the database, the result would able to tell the similarity of images. The larger the Euclidian distance, the more dissimilar the images comparing to query image.

Objective

In this experiments, we will implement basic image processing methods using Python and to perform image feature extraction and compare the feature effectiveness in image retrieval task by using the CBIR system of image feature pre-trained convolutional neutral network (CNN) and colour histogram that uses red, green and blue (RGB) channels.

Experiment Questions

Question 1 :
Write a short Python code to count the number of jpeg images in the \images folder

```
fileList=[]

for file in os.listdir(imgpath):
        if file.endswith(".jpg"):
            fileList.append(os.path.join(imgpath, file))

print("\n The number of images in the folder used for image database = " ,
len(fileList))
```

```
The number of images in the folder used for image database =  1000
```

Question 2 :

Compare the difference between retrieval in traditional database with content based image retrieval (CBIR). What is the benefit of CBIR?

```
Traditional database query techniques is based on attaching textual metadata
to each image and retrieve them by keywords. In CBIR, image procesing
algorithm are used to extract feature vectors that represent the image
properties. The benefit of CBIR is it required less time to find related
images and feature extraction methods are easy, effective and less expensive.
Also, it is possible to retrieve a very similar image to the one choose by
the user.
```

Question 3:

(a) Compute dissimilarity distance (euclidian) with feature from colour histogram
Extract the feature from image '001.jpg' , '022.jpg' and '400.jpg'. Compute the euclidian distance between the two feature vector. Compute the euclidian distance between the two feature vector from '001.jpg' , '022.jpg'. Then compute the euclidian distance between the two feature vector from '001.jpg' , '400.jpg'.

```
h1 = feat_hists[0] #001
h2 = feat_hists[1] #022
h3 = feat_hists[2] #400

h_dist1_vs_22 = np.linalg.norm(h1-h2)
print("With color hist, image 1 vs image 22 , distance = ", h_dist1_vs_22)

h_dist1_vs_400 = np.linalg.norm(h1-h3)
print("With color hist, image 1 vs image 400 , distance = ", h_dist1_vs_400)
```

```
With color hist, image 1 vs image 22 , distance =  0.0345468839243469
With color hist, image 1 vs image 400 , distance =  0.7658398822987645
```

(b) Compute dissimilarity distance (euclidian) with feature from pre-trained convolutional neural network CNN

Extract the feature from image '001.jpg' , '022.jpg' and '400.jpg'. Compute the euclidian distance between the two feature vector. Compute the euclidian distance between the two feature vector from '001.jpg' , '022.jpg'. Then Compute the euclidian distance between the two feature vector from '001.jpg' , '400.jpg'.

```
c1 = feat_cnns[0]
c2 = feat_cnns[1]
c3 = feat_cnns[2]

cnns_dist1_vs_22 = np.linalg.norm(c1-c2)
print("With cnn feature, image 1 vs image 22 , distance = ",
cnns_dist1_vs_22)
```

```
cnns_dist1_vs_400 = np.linalg.norm(c1-c3)
print("With cnn feature, image 1 vs image 400 , distance = ",
cnns_dist1_vs_400)
```

```
With cnn feature, image 1 vs image 22 , distance =  100.03564
With cnn feature, image 1 vs image 400 , distance =  135.05408
```

Question 4:
Explain your conclusion on the pairwise similarity between the three images '001.jpg' , '022.jpg' and '400.jpg' . Comment on the semantic similarity between the three images and the euclidian distance among their feature vectors. Compare colour histogram and CNN feature.

The smaller the euclidean distance, the more similarity between images. By comparing the colour histogram and CNN feature, the euclidean distance among their feature vector between '001.jpg' and '022.jpg' are smaller, however, between '001.jpg' and '400.jpg' are bigger. Hence, the similarity between images '001.jpg' and '022.jpg' are more similar, meanwhile, '001.jpg' and '400jpg' are less similar.

Question 5:
Implement the following function to display the selected image and information related to the image.
showImageInfoFromDB(id, imgpath, database)

```
    label = database[id].classLabel
    feat1 = database[id].featCNN
    feat2 = database[id].featColorHist

    print("Image name = " , database[id].imageName)
    print("Label ID = " , label)
    print("Label Name = " , LabelDic[label])
    print("Feature dimension CNN = " , feat1.shape)
    print("Feature dimension Colour Histogram = " , feat2.shape)

    imFile = database[id].imageName
    imFile = os.path.join(imgpath, imFile)
    im = Image.open(imFile)
    plt.figure(figsize=(8,6))

    plt.imshow(im) , plt.axis('off')
    titleStr = " Image {}.jpg label = {} Label name = {}".format(str(id),
label, LabelDic[label])
    plt.title(titleStr)
```

```
Image name =  900.jpg
Label ID =  10
Label Name =  Food
Feature dimension CNN =  (1, 4096)
Feature dimension Colour Histogram =  (768,)
```

Question 6

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

retrievedID = doRetrieval(featQuery , k, database, imgpath, showImage=True)
Precision_K = getPrecisionRank_K(k, queryLabel, retrievedID, database)

```
def doRetrieval(featQuery , k, database, imgpath, showImage=True):

    numImages = len(database)
    dist_cnn = []
    idx_k = []

    for f in range (0,numImages) :
        dist = np.linalg.norm(featQuery - database[f].featCNN)
        dist_cnn.append(dist)

    idx_k = np.argsort(dist_cnn)

    return idx_k[1:k+1]


def getPrecisionRank_K(k, queryLabel, retrievedID, database):

    rel_img = 0

    for f in retrievedID:
        label = database[f].classLabel
        print(label, end=' ')
        if queryLabel == label:
                rel_img += 1

    precision_k = rel_img/k

    return precision_k
```

```
Experiment on CBIR with CNN feature as image feature

Class labl of retrieve 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
```

Question 7

Write and implement the functions and complete the scripts to compute the average precision for **ten** queries. Use the following image index for class 1 (Africa) as queries, 0,1,2,3,4,5,6,7,8,9 . Use CNN feature.

```
k =10
i=0
Precision_K = np.zeros(k)
queryID_list = [0,1,2,3,4,5,6,7,8,9]


for queryID in queryID_list :

    featQuery = database[queryID].featCNN
    retrievedID = doRetrieval(featQuery, k, database, imgpath,
showImage=True)
    queryLabel = database[queryID].classLabel
    print(" Class label of retrieved img")
    Precision_K[i] = getPrecisionRank_K(k, queryLabel, retrievedID, database)

    print("Query image label :" , queryLabel)
    print("Precision for query ID {} when retrieving {} images =
{:02.3f}\n".format(queryID,k, Precision_K[i]))
    i=i+1

average_precision = np.mean(Precision_K)

print("\n\n Average precision = {:02.3f} ".format( average_precision ) )
```

```
 Class label of retrieved img
1 1 1 9 1 1 1 3 3 1 Query image label : 1
Precision for query ID 0 when retrieving 10 images = 0.700

 Class label of retrieved img
1 1 1 1 1 1 1 1 1 1 Query image label : 1
Precision for query ID 1 when retrieving 10 images = 1.000

 Class label of retrieved img
1 1 1 9 3 1 1 1 1 1 Query image label : 1
Precision for query ID 2 when retrieving 10 images = 0.800

 Class label of retrieved img
1 1 1 1 2 1 2 1 1 1 Query image label : 1
Precision for query ID 3 when retrieving 10 images = 0.800

 Class label of retrieved img
1 2 1 2 2 2 1 2 3 2 Query image label : 1
Precision for query ID 4 when retrieving 10 images = 0.300

 Class label of retrieved img
1 1 1 1 1 1 2 1 1 1 Query image label : 1
Precision for query ID 5 when retrieving 10 images = 0.900

 Class label of retrieved img
1 1 1 1 1 1 1 1 2 1 Query image label : 1
Precision for query ID 6 when retrieving 10 images = 0.900

 Class label of retrieved img
1 1 1 1 1 1 1 1 1 1 Query image label : 1
Precision for query ID 7 when retrieving 10 images = 1.000

 Class label of retrieved img
1 1 1 1 1 1 1 1 1 1 Query image label : 1
Precision for query ID 8 when retrieving 10 images = 1.000

 Class label of retrieved img
1 1 1 1 1 1 1 1 1 1 Query image label : 1
Precision for query ID 9 when retrieving 10 images = 1.000


 Average precision = 0.840
```

Question 8

Repeat the task in question 7 by completing the short script in the cell to compute the average precision for 100 queries. Use the first 10 images for each class as query image. All the 10 classes must be included. Use **CNN feature**.

```
print("\n Performing CBIR experiment with 100 queries with CNN feature\n\n")

k=10 # select the top K image to be retrieved

precisionArr = []
Precision_K = []

for queryID in qlist:
```

```
    featQueryCNN = database[queryID].featCNN
    retrievedID = doRetrieval(featQueryCNN , k, database, imgpath,
showImage=True)
    queryLabel = database[queryID].classLabel
    print(" Class label of retrieved img")
    prec = getPrecisionRank_K(k, queryLabel, retrievedID, database)

    print("Query image label :" , queryLabel)
    print(" Precision for query ID {} when retrieving {} images =
{:02.3f}\n".format(queryID,k, prec))
    precisionArr.append(prec)

Precision_K = precisionArr
average_precision = np.mean(Precision_K)

print("\n\n Average precision for all 10 classes = {:02.3f} ".format(
average_precision ) )
```

```
Average precision for all 10 classes = 0.904
```

Question 9

Repeat the experiment in question 8 with the use of **colour histogram** image feature.

```
print("\n Performing CBIR experiment with 100 queries with Color histogram
feature")

precisionArr = []
Precision_K = []

for queryID in qlist:

    featQuery = database[queryID].featColorHist
    retrievedID = doRetrieval_hist(featQuery, k, database, imgpath,
showImage=True)
    queryLabel = database[queryID].classLabel
    print(" Class label of retrieved img")
    prec = getPrecisionRank_K(k, queryLabel, retrievedID, database)

    print("Query image label :" , queryLabel)
    print(" Precision for query ID {} when retrieving {} images =
{:02.3f}\n".format(queryID,k, prec))
    precisionArr.append(prec)


Precision_K = precisionArr
average_precision = np.mean(Precision_K)
```

```
print("\n\n Average precision for all 10 classes = {:02.3f} ".format(
average_precision ) )
```

```
Average precision for all 10 classes = 0.601
```

Question 10
Write a conclusion based on the result of the image retrieval experiment. Which feature gives better accuracy for image retrieval?

```
To perform CBIR experiment with 100 queries with CNN Feature and Color
Histogram Feature, CNN Feature has an average precision for all 10 classes is
0.904, however, Color Histogram Feature has an average precision for all 10
classes is 0.601. Hence, CNN Feature gives better accuracy for image
retreval.
```

Conclusion

In this experiment, I have learned the CBIR technique to perform searching similarity image retrieval by using feature extraction from query image and feature vectors from image database, then by calculate the distance of Euclidian distance to know the similarity of the image.

By performing CBIR experiment with 100 queries, CNN feature provides 90.4% accuracy and color histogram feature provides 60.1% accuracy. Hence, we can conclude CNN feature is more suitable for image retrieval since CNN gives better accuracy than color histogram

CNN Feature

```
Average precision for all 10 classes = 0.904
```

Color Histogram Feature

```
Average precision for all 10 classes = 0.601
```