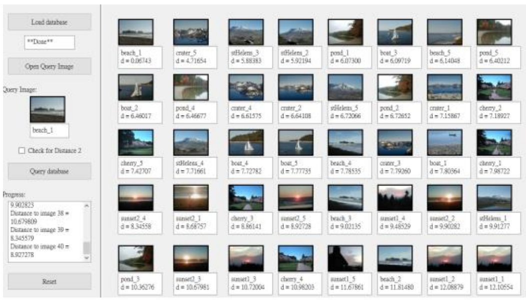
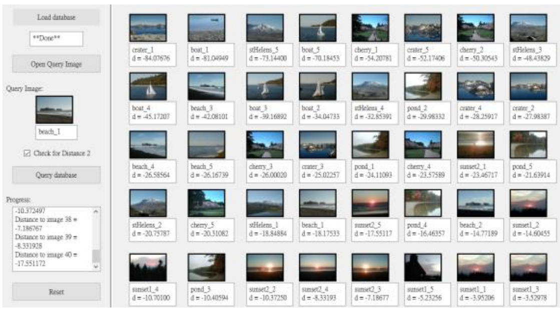
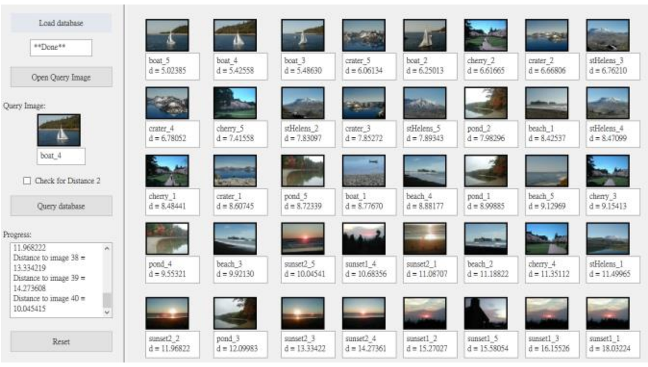
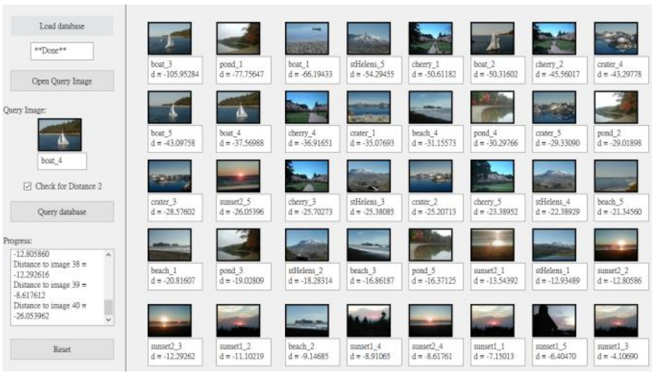
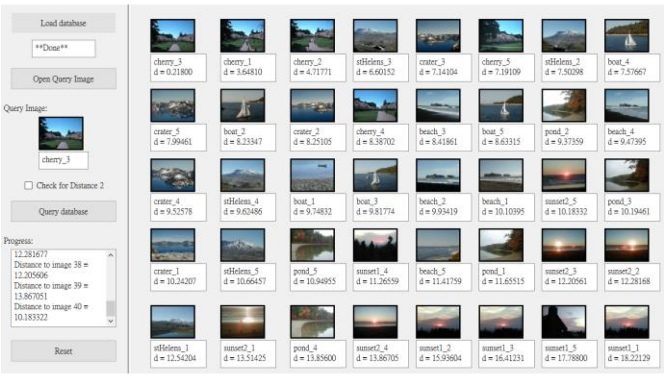
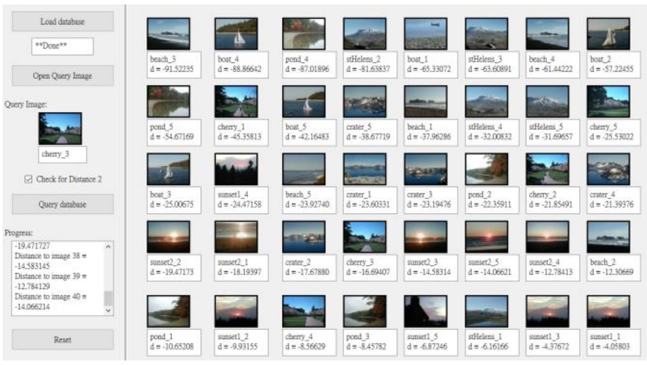
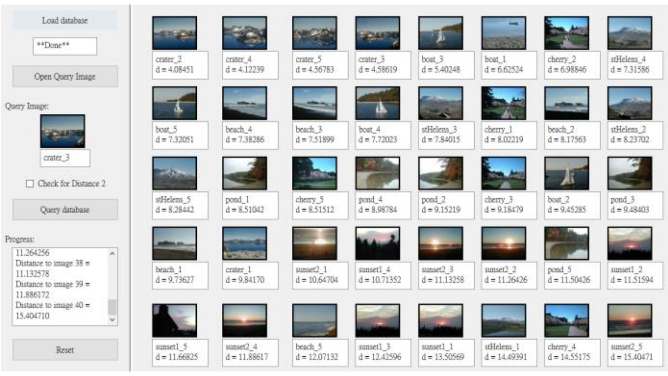
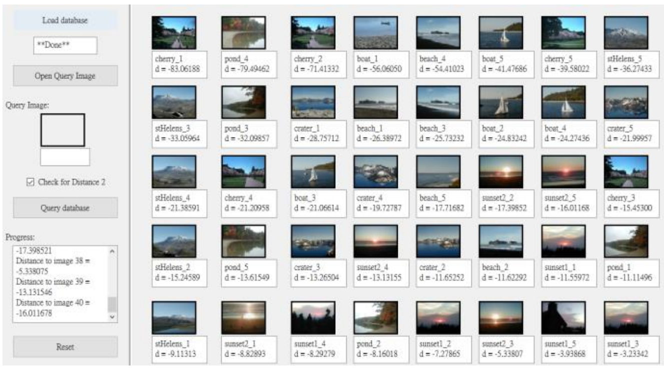
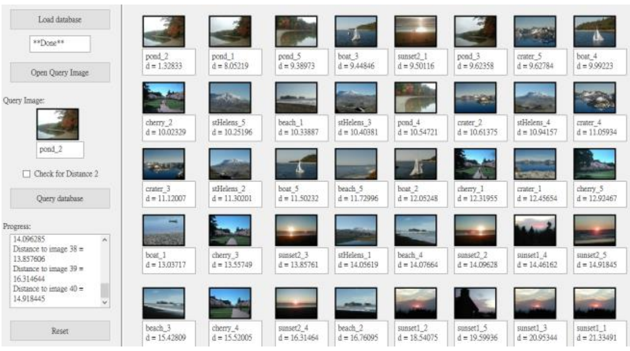
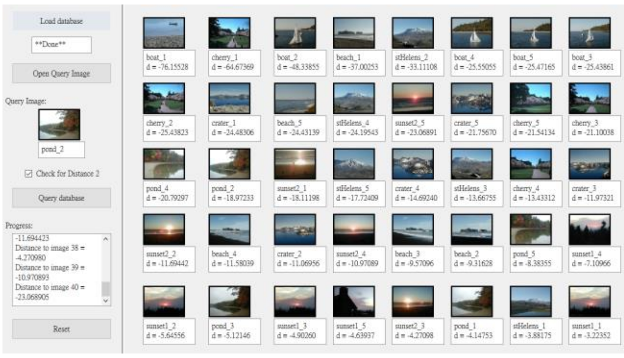
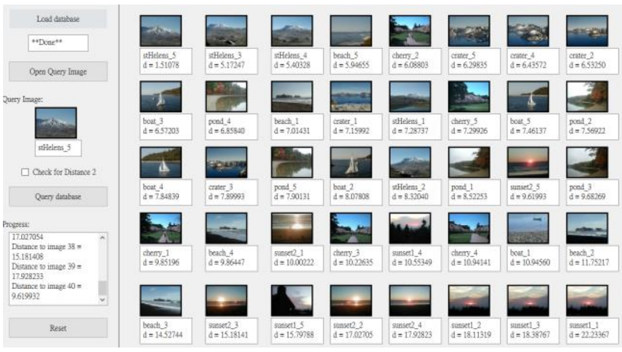
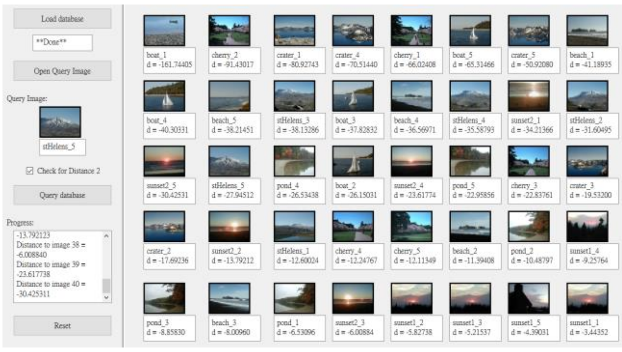
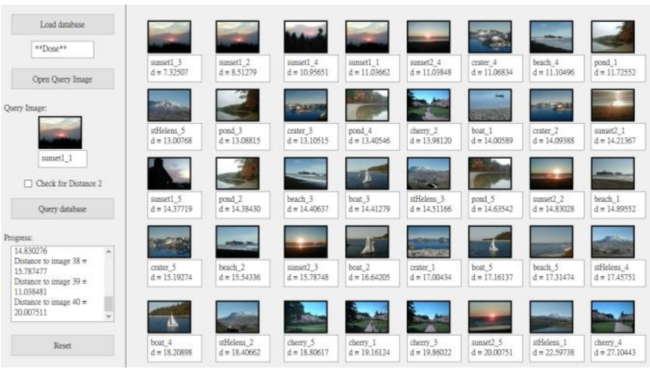
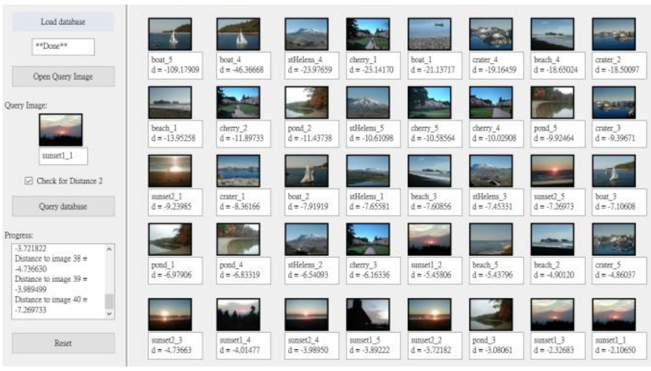
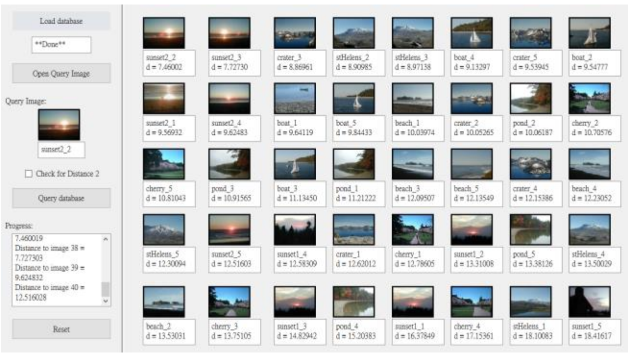
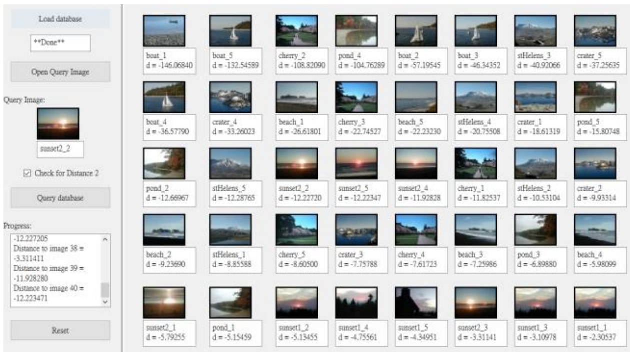
CSE576_HW3_Yao-Chung Liang

L1 norm (distance 2)	Reference distance weighted (distance 1)
Beach1	
	
Boat4	
	
Cherry3	
	
Crater_3	
	

L1	Reference distance weighted
Pond_2	
	
stHelens_5	
	
Sunset1_1	
	
Sunset2_2	
	

Discussion:

0.

I implemented 8*8 and 256*256 GLCM but they are not much difference in retrieval and in running time. The major problem for me in this assignment is how to tune parameters of weighting.

1.

From the experiment results above, you can see that both measurements yield the same top 1 result. And I found that "beach_1" is difficult to yield 3 images retrieve from the same class on the top 5. For "beach_1", I think it's because the light is much different from the other images in the same class. For "sunset1_1", I got 5 out of top 5 result which is impressive. Also, I found that every time when clustering the image pixels, the cluster may be different which also influence the result a little bit.

2. Attributes: I implemented energy, entropy, contrast, boundary, centroid, 8*8 GLCM and 256*256 GLCM

Formula :

Gray-Level Co-occurrence Matrices (GLCMs)

Consider the image (below left). If we use the position operator "1 pixel to the right and 1 pixel down" then we get the gray-level co-occurrence matrix (below right)

$$\begin{matrix} 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{matrix} \qquad C = \frac{1}{16}\begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

where an entry c_{ij} is a count of the number of times that $F(x, y) = i$ and $F(x+1, y+1) = j$. For example, the first entry comes from the fact that 4 times a 0 appears below and to the right of another 0. The factor 1/16 is because there are 16 pairs entering into this matrix, so this normalizes the matrix entries to be estimates of the co-occurrence probabilities.

For statistical confidence in the estimation of the joint probability distribution, the matrix must contain a reasonably large average occupancy level. Achieved either by (a) restricting the number of amplitude quantization levels (causes loss of accuracy for low-amplitude texture), or (b) using large measurement window. (causes errors if texture changes over the large window). Typical compromise: 16 gray levels and window size of 30 or 50 pixels on each side. Now we can analyze C:

- maximum probability entry
- element difference moment of order k: $\sum_i \sum_j (i-j)^k c_{ij}$

  This descriptor has relatively low values when the high values of C are near the main diagonal. For this position operator, high values near the main diagonal would indicate that bands of constant intensity running "1 pixel to the right and 1 down" are likely. When $k = 2$, it is called the contrast:
- Contrast $= \sum_i \sum_j (i-j)^2 c_{ij}$
- Entropy $= -\sum_i \sum_j c_{ij} \log c_{ij}$

For entropy term, I found that a smoother image will yield high entropy. But for image retrieval, it's not that important. For contrast term, I found that a image with more difference will result in higher contrast. But for image retrieval, it's not that important. For boundary term, it's greatly influenced by shifting, thus this term isn't suitable for looking for similar images in the same class. For centroid term, this is ok if there is not much rotation in camera coordinate.

Among all the features, I found that energy is the most important one which quite make sense since it’s a important feature when you take a picture at about the same position and orientation, the relative position of the landscape is about the same which will result in similar parts of GLCM. But if you only focus on this term, when there are two totally different picture but with the same color distribution, then it hard to tell the difference. Thus, in stead of only focus on energy, we also need to consider the correlation between pixels. So I also implemented contrast, correlation, entropy. But I think I should not only focus on local features, thus I also tried to add some global feature like GIST feature but I failed, so I just use randomSelect to select some global pixels’ values but it turns out to be a noise term. Thus I deleted it. I also think boundary and centroid are important terms which also separate the similar image from the dissimilar ones.

3. Distance measures: I implemented L1, reference distance weighted (RDW), cosine and Bray-Curtis.

L1 norm is

```
Sum += abs(vector1[i] - vector2[i]);
return Sum;
```

RDW is

```
Sum+=abs((vector1[i]-vector2[i])/vector1[i]);
Sum /= vector length;
Return Sum;
```

Bray-Curtis is

```
sum+= (vector1[i]-vector2[i])/(vector1[i]+vector2[i]);
return sum;
```

L1 works quite well while RDW doesn’t. RDW is used to measure the dissimilarity between two vectors and I thought if I use it when RDW is small which means you are similar but in the end it doesn’t work. It’s mainly because the denominator would be too small so the value will be close to infinity so they cannot be compared. And also some features’ values are dominating the measurements in denominator terms. Thus I also considered these edge cases. I also tried cosine of two vectors but it doesn’t work well for it’s a higher dimension vector. Bray-Curtis is not suitable for this case though.

4. I also study a paper[1] which includes the methods developments of CBIR after 2003. From this paper I learned whole ideas of how they deal with computer vision problems in CBIR. I just want to mention for it’s a good paper.

Reference:

[1] W. Zhou, H. Li, Q. Tian : Recent Advance in Content-based Image Retrieval: A Literature Survey, 2017.