

CSE573
HW1 Write-Up
Enze Qian
50091378

Q1.0

There are four types of filter:

Gaussian filters:

They are low-pass filters, which means they keep low frequency signals and remove high frequency signals. They can smooth and blur images.

LoG filters:

They take second derivative of images to capture changes and corners in images. It will give a positive result when changes occur in the images.

Derivative of Gaussian in x direction:

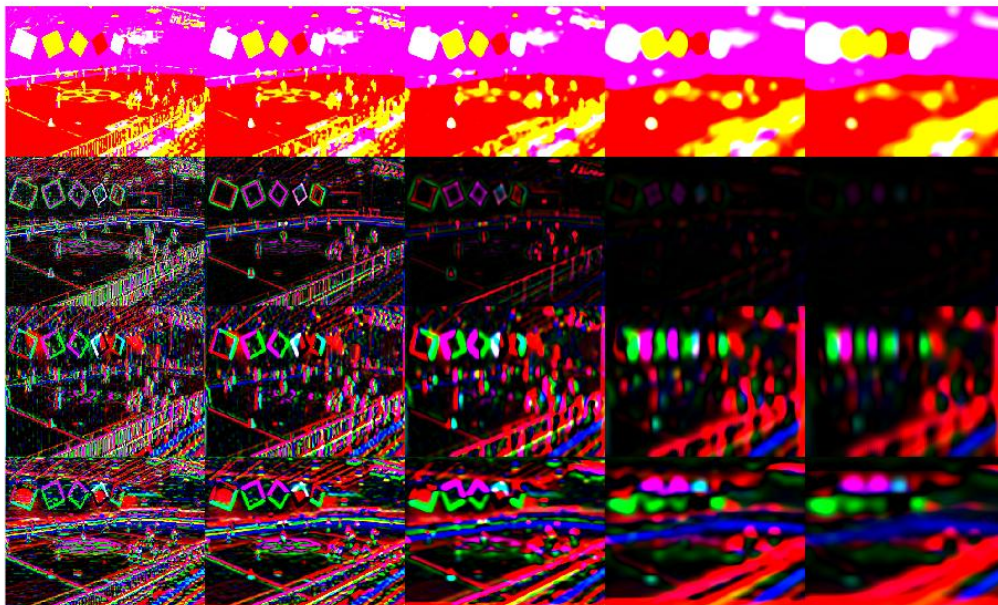
They can capture changes along x direction and it is used to detect edges.

Derivative of Gaussian in y direction:

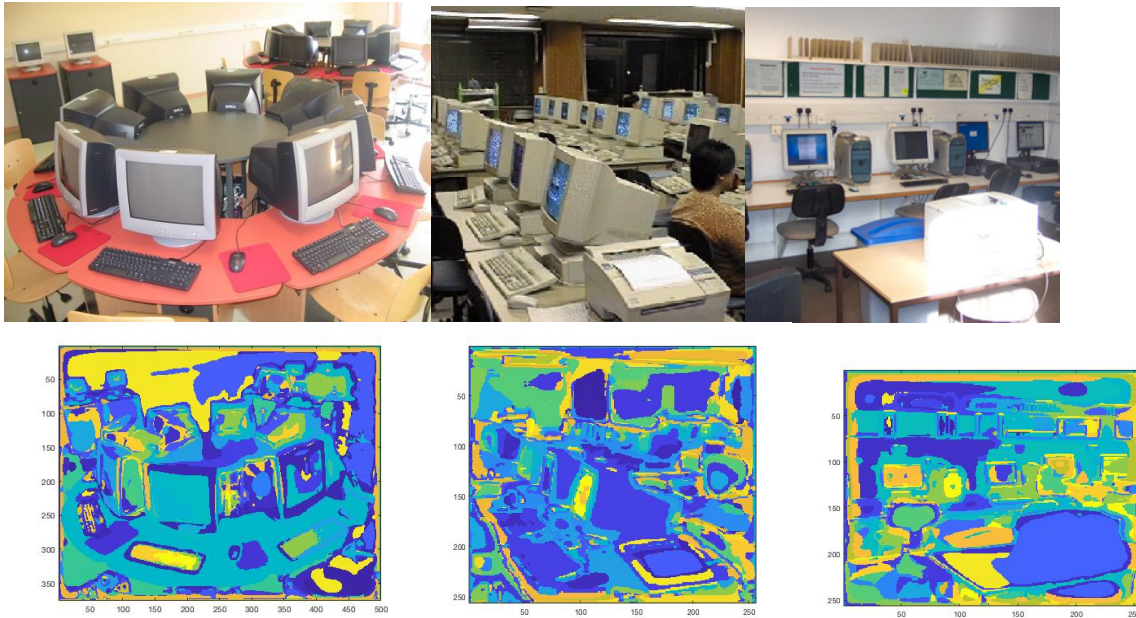
Similar to Derivative of Gaussian in x direction, they can capture changes along y direction and it is used to detect edges.

Q1.1

I use the ice skating image which is the same as the image on homework description, the code to generate collage of 20 images is in the file Collage.m



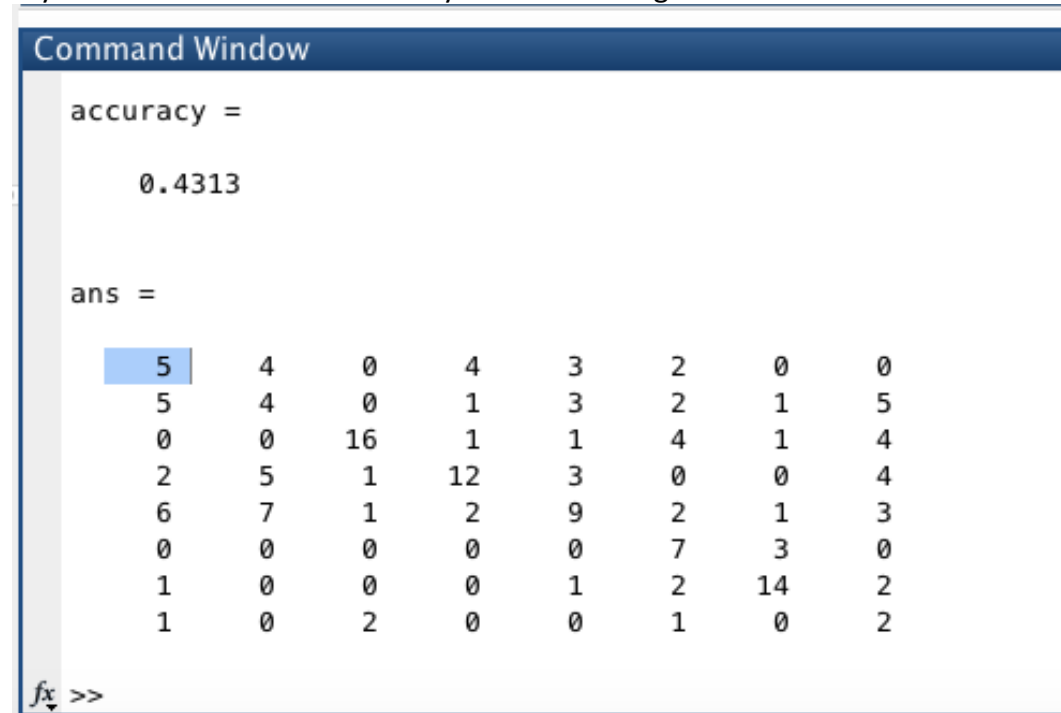
Q1.3



I choose $K=100$ and $\alpha=50$ for creating dictionary. For these three images from computer room, I can clearly see the shape of monitor and keyboard. The code for wordmap visualization can be found in VisualizeQ13.m

Q2.5

My confusion matrix and accuracy are as following:



The column of matrix is the label of test image and the row is the result after classification.

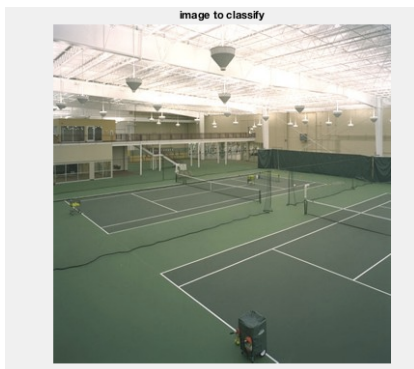
Q2.6

As we can see, only 2 of 20 tennis court images are classified correctly. So I want to check the failed cases in tennis court.

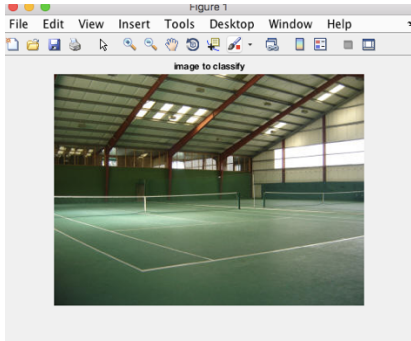
First, I find tennis court in test images, which are:

'tennis_court/sun_dvhqzdoqxfgo.jpg'
'tennis_court/sun_dvovayactfgzlfwn.jpg'
'tennis_court/sun_dwctpqdcbjazzrj.jpg'
'tennis_court/sun_dwjqsmijgddxzjiy.jpg'
'tennis_court/sun_dwnadypvmdjxjcyf.jpg'
'tennis_court/sun_dwwzytlfbauewiw.jpg'
'tennis_court/sun_dwxmzlnkuptffnge.jpg'
'tennis_court/sun_dxcoumniqgienzne.jpg'
'tennis_court/sun_dxuzgfcwbrxidwiw.jpg'
'tennis_court/sun_dycyjcqkttvutfs.jpg'
'tennis_court/sun_dyftunonjhgtwinw.jpg'
'tennis_court/sun_dygifieuigsylfeo.jpg'
'tennis_court/sun_dyhsjpneryxxruad.jpg'
'tennis_court/sun_dynpnqbmjncnlxbq.jpg'
'tennis_court/sun_dyrqkxisgvljeyz.jpg'
'tennis_court/sun_dyvxlyphqcqkfbiu.jpg'
'tennis_court/sun_dyxmdestrwscgjtq.jpg'
'tennis_court/sun_dyzaolvwzgqrrjzp.jpg'
'tennis_court/sun_dzbpvxokbbfqmqsc.jpg'
'tennis_court/sun_dzlhnmruidpszex.jpg'

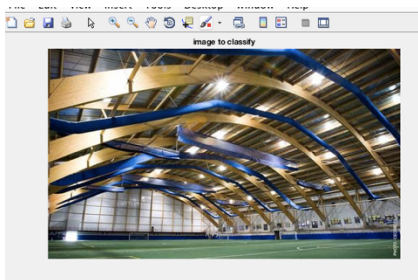
Let's check some failed cases:



My program classifies it into Ocean. I think the reason is the structure of this image is similar to ocean. (i.e. wide and flat).



My program classifies it into computer room. There are many rectangles in this image, which is similar to the shape of computer monitor.



My program classifies it into library. Because it contains a wooden roof, which refers to library.

In conclusion, I think the reason tennis court images have the most failed cases is that tennis court image doesn't have too many unique properties which are distinct from other cases.

Q2.7

At the first time I implemented function `getFilterBankAndDictionary`, I chose $K=100$ and $\alpha=50$. I think the accuracy could be improved by increasing the value of K and α . So I set $K=300$ and $\alpha=2$ and the results are:

```
Command Window

accuracy =

    0.4938

ans =

    6     1     1     1     3     1     1     3
    1     8     0     5     3     0     1     5
    0     1    19     1     1     5     0     5
    2     4     0    12     3     1     0     3
    9     6     0     1    10     3     1     1
    0     0     0     0     0     6     2     0
    2     0     0     0     0     2    15     0
    0     0     0     0     0     2     0     3
```

The accuracy has been increased by 6%. I think the reason is that a larger dictionary can give more accurate result when the image is being compared to the dictionary.