In [ ]:

Computer Vision

16720-B Fall 2022

16720 (B) Bag of Visual Words - Assignment 2

Instructor: Kris Kitani TAs: Sheng-Yu, Jinkun, Rawal, Arka, Rohan

Theory Questions

This section should include the visualizations and answers to specifically highlighted questions from P1 to P4. This section will be manually Graded

Q1.1.1 (5 Points WriteUp)

What visual properties do each of the filter functions (See Figure below) pick up? You should group the filters into categories by its purpose/functionality. Also, why do we need multiple scales of filter responses? **Answer in the writeup. Answer in your write-up.**

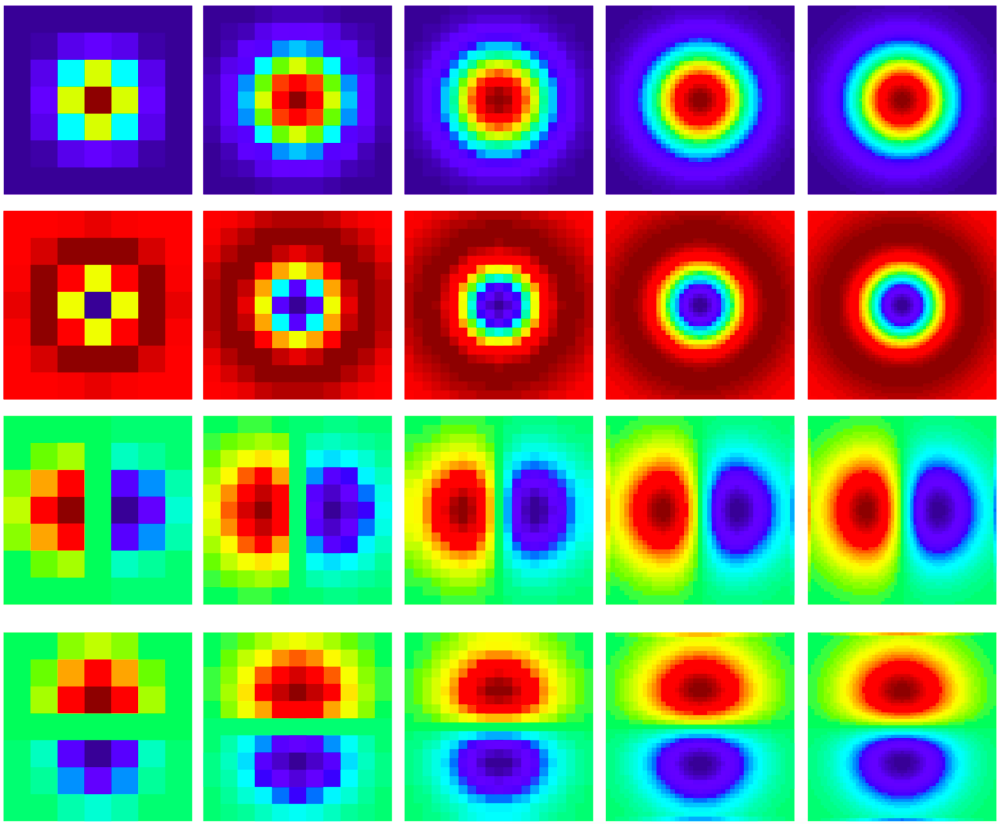


Figure1. The provided multi-scale filter bank

First horizontal layer:- Gaussian Blur Second horizontal layer:- Gaussian laplace Third horizontal layer:- Gaussian in x-direction Fourth horizontal layer:- Gaussian in y-direction

Gaussian blur smoothens the image (removes high frequency content) Gaussian laplace picks up (blobs) points which exhibit a particular high light intensity in a better way as compared to the other filters. Gaussian in x and Gaussian in y picks up edges in x and y direction respictively.

We can see that with the use of different types of filters we can extract different kinds of features. For ex, using Gaussian in x, y we can detect edges and corners. But using just one of them wont give us corners neither will we have any information about the other direction that we didnt use. Therefore, to extract features in a better way, it makes sense to use different types of filters.

Now, even in one filter we use different scales! This is to capture features of different sizes. We can say that using different scales makes our feature extraction scale invarient. This can be very well understood by the example of sunflower that we dicsussed in the class. We get the maximum response for one particular jscale for a particular feature. And while extracting the feature, that is the response we will use!!

Q1.3.1 (5 Points WriteUp)

Visualize three wordmaps of images from any one of the category. **Include these in your write-up, along with the original RGB images. Include some comments on these visualizations: do the "word" boundaries make sense to you?**. We have provided helper function to save and visualize the resulting wordmap in the util.py file. They should look similar to the ones in Figure 2.

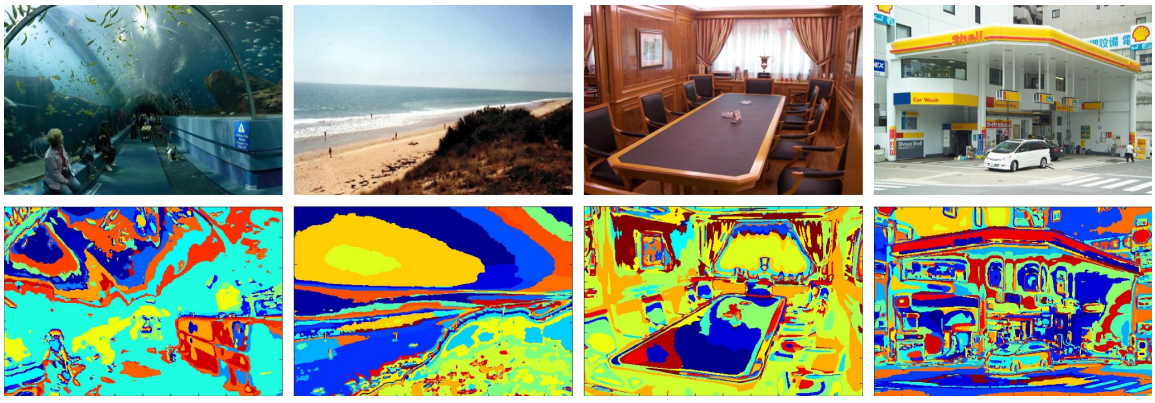
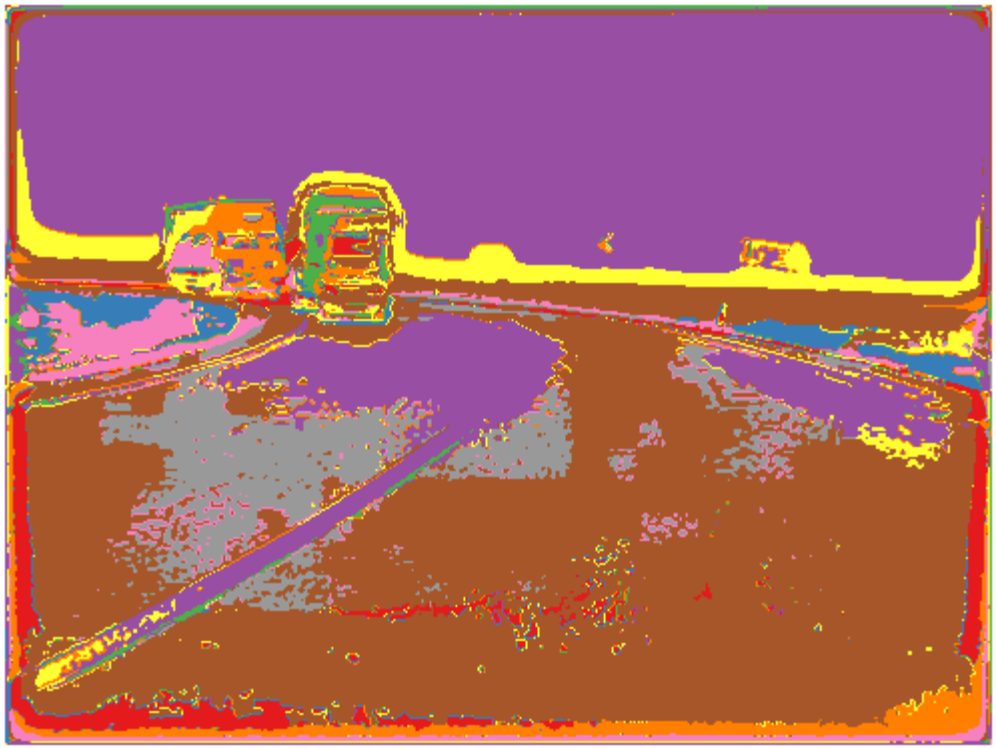
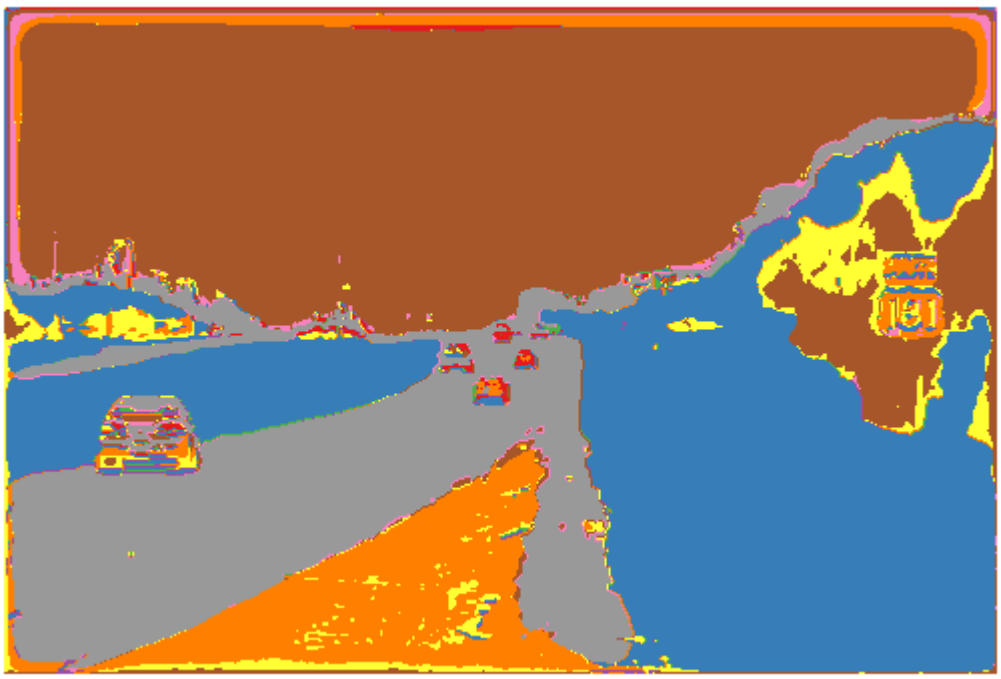


Figure 2. Visual words over images. You will use the spatially un-ordered distribution of visual words in a region (a bag of visual words) as a feature for scene classification, with some coarse information provided by spatial pyramid matching [2]

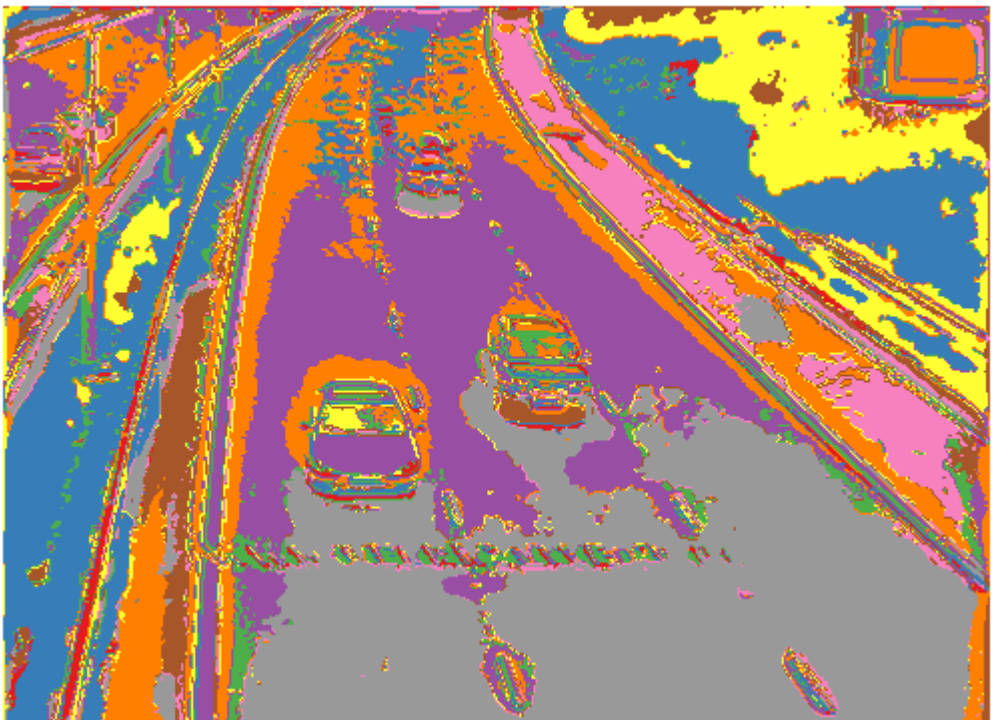
We can see from the following wordmaps when compared with the original image, that similar kind of features (for example car, sign boards, bridges, sky, etc) are expressed with similar colors (i.e., wordmaps). Yes, the "word" boundaries make sense when we compare the images. All the different features have different colors (words) and all the same ones have similar ones. And these boundaries can be seen on the wordmaps.

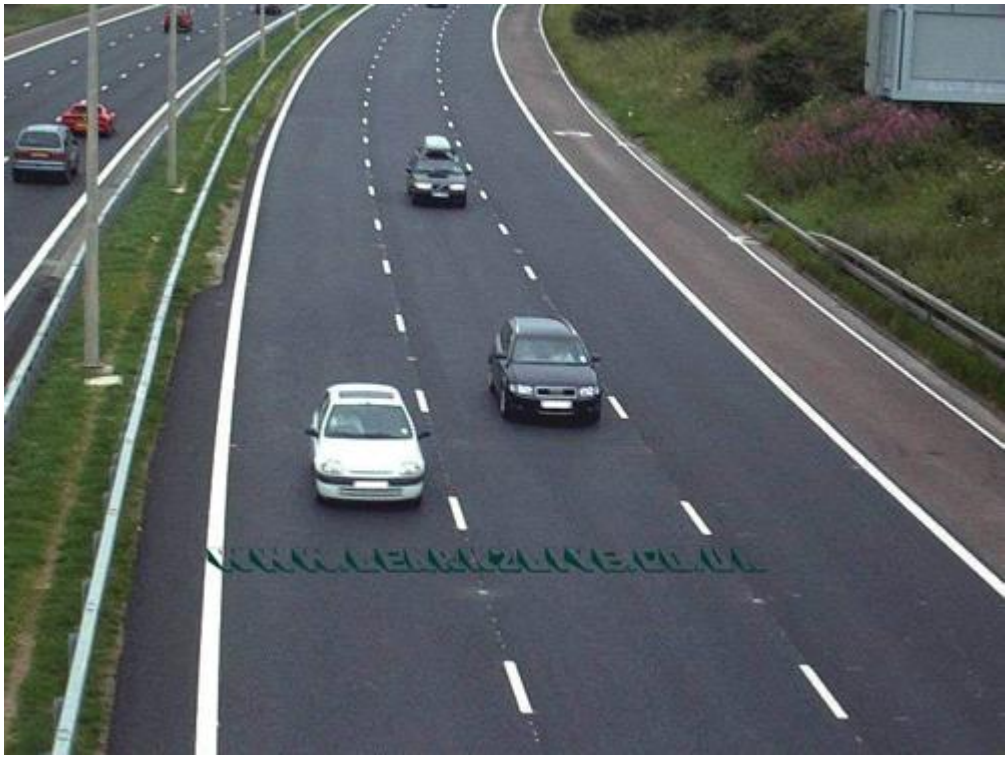





SOUTH
181

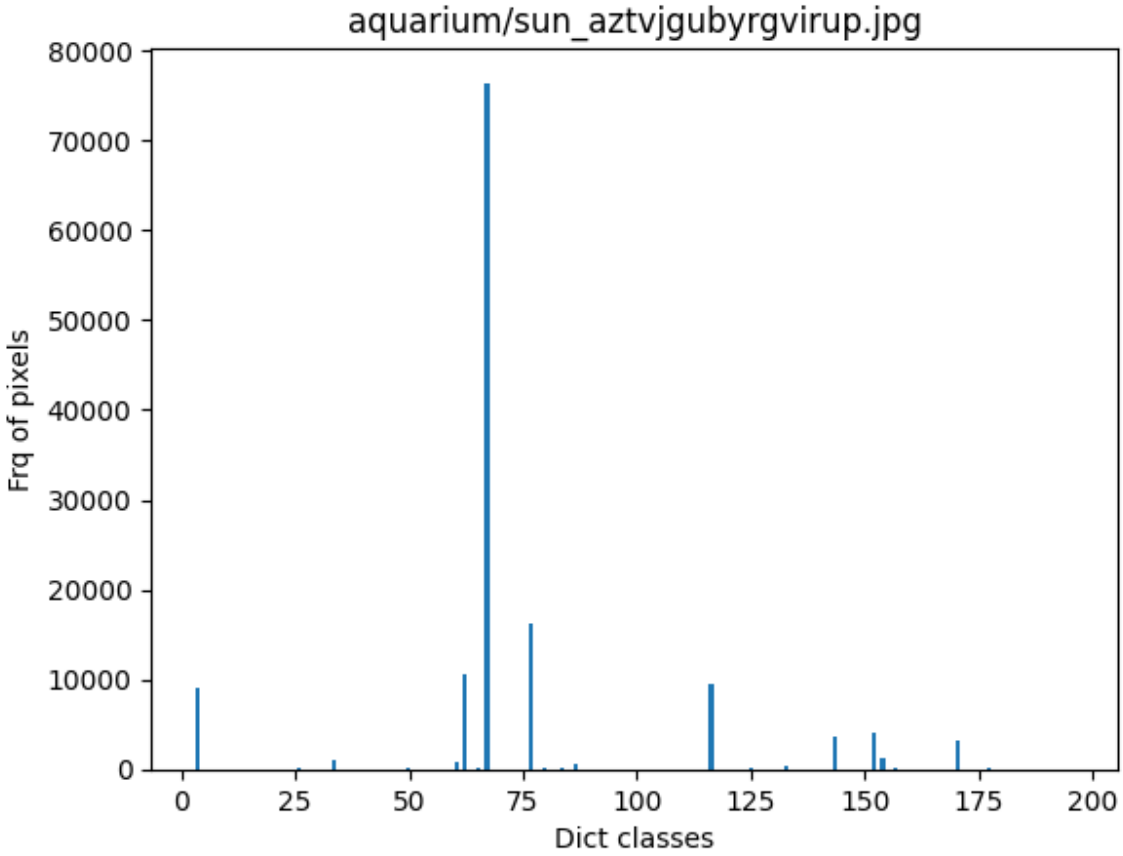

EXIT 1C
EXIT 1B
H3
EAST
Halawa Hts.
Halawa Valley
Stadium
H201
EAST
Honolulu
Kaneohe
½ MILE
EXIT ONLY

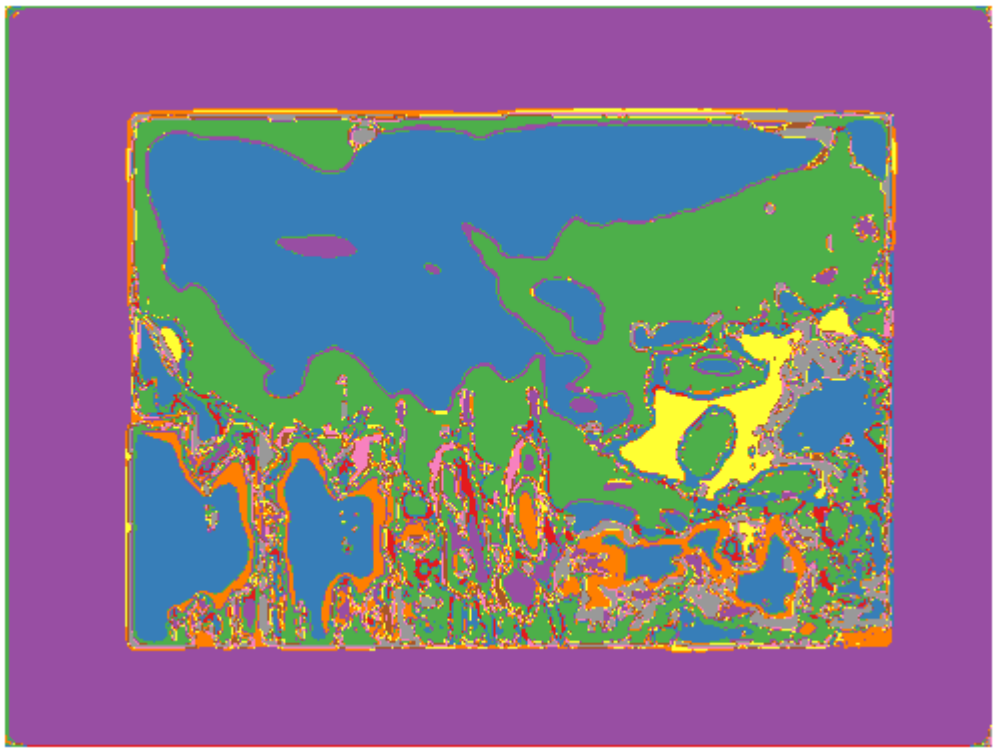


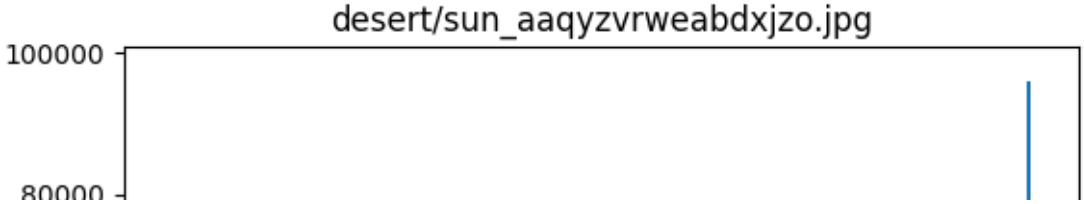


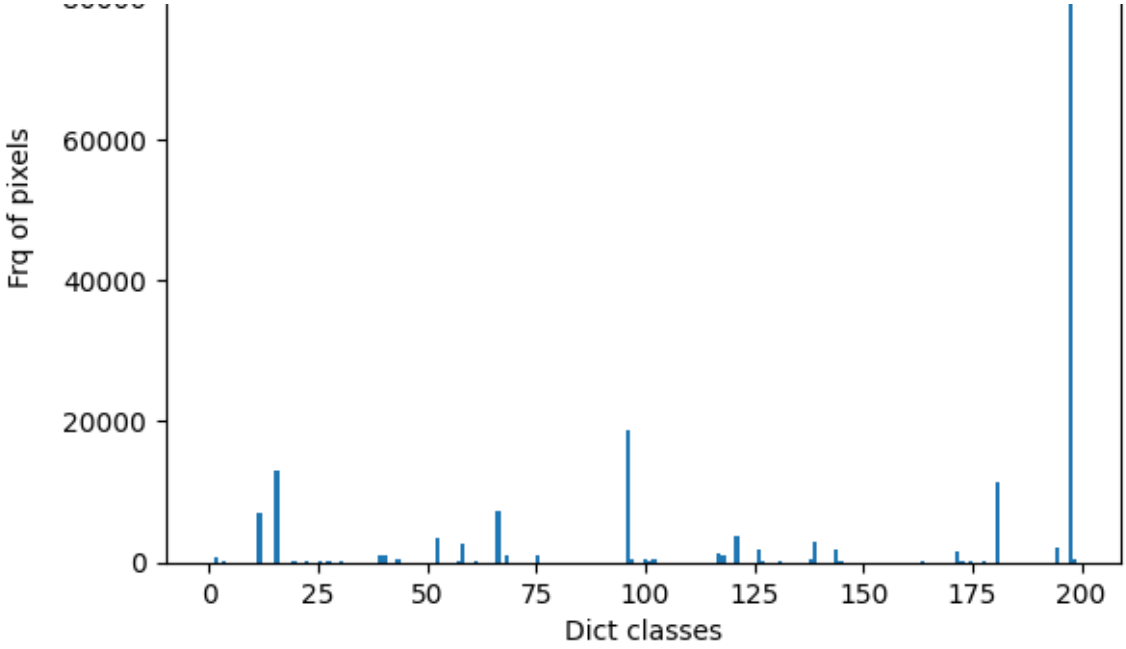
Q2.1

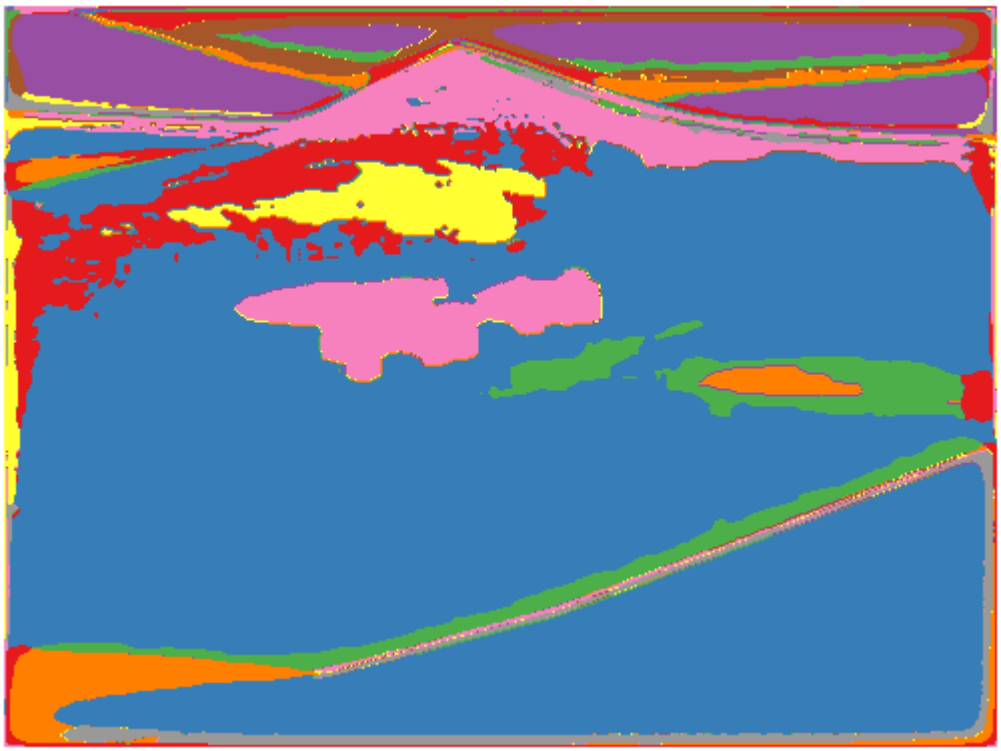
For 5 Images, include their visual word maps and histograms

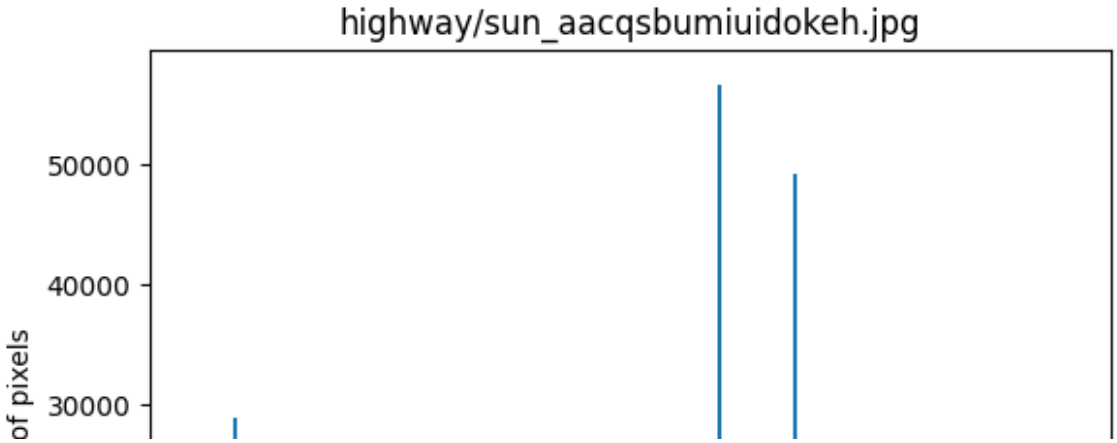

aquarium/sun_aztvjgubyrgvirup.jpg
Frq of pixels
Dict classes
0
10000
20000
30000
40000
50000
60000
70000
80000
0
25
50
75
100
125
150
175
200

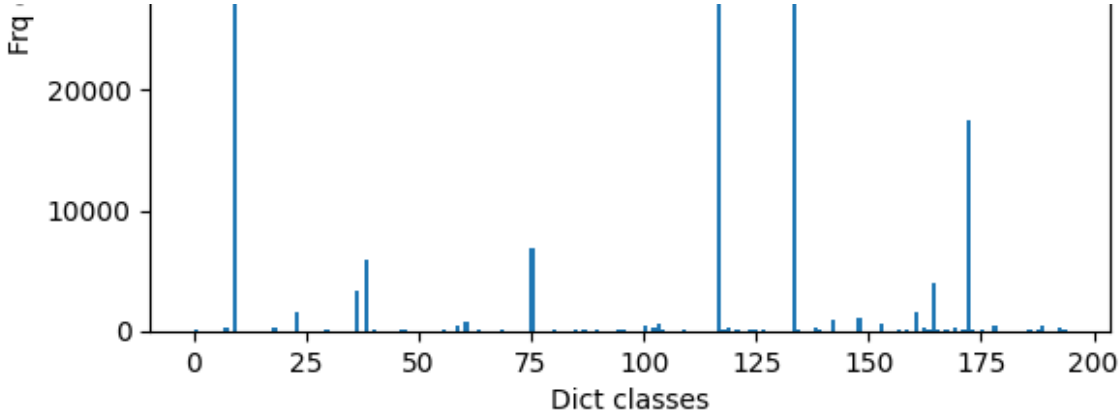


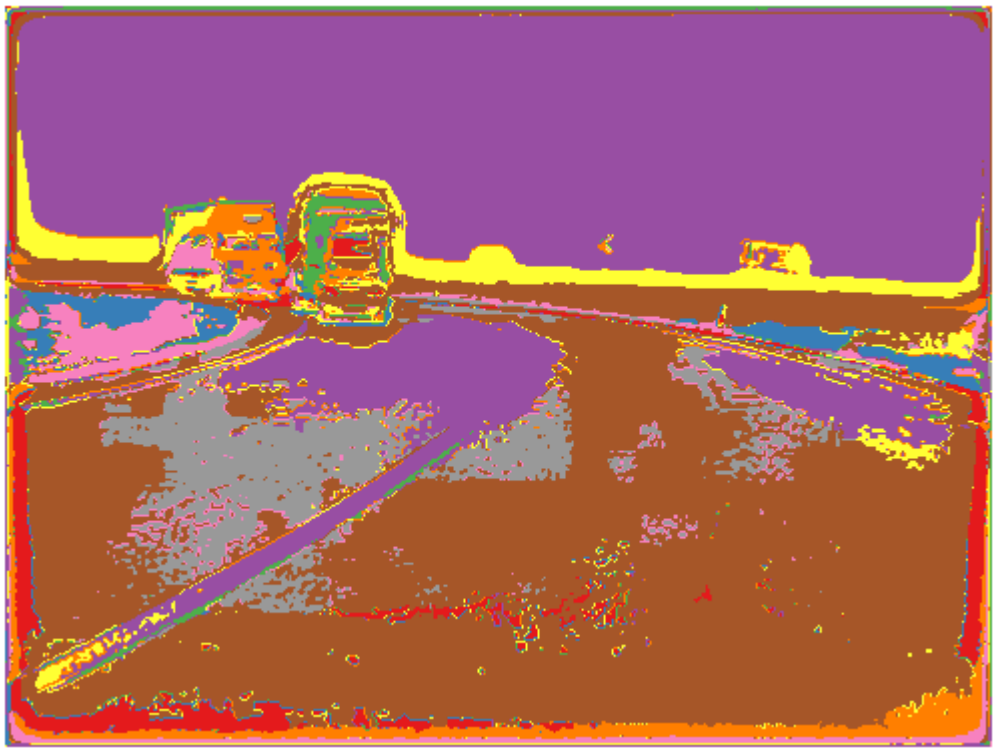

desert/sun_aaqyzvrweabdxjzo.jpg
100000
80000

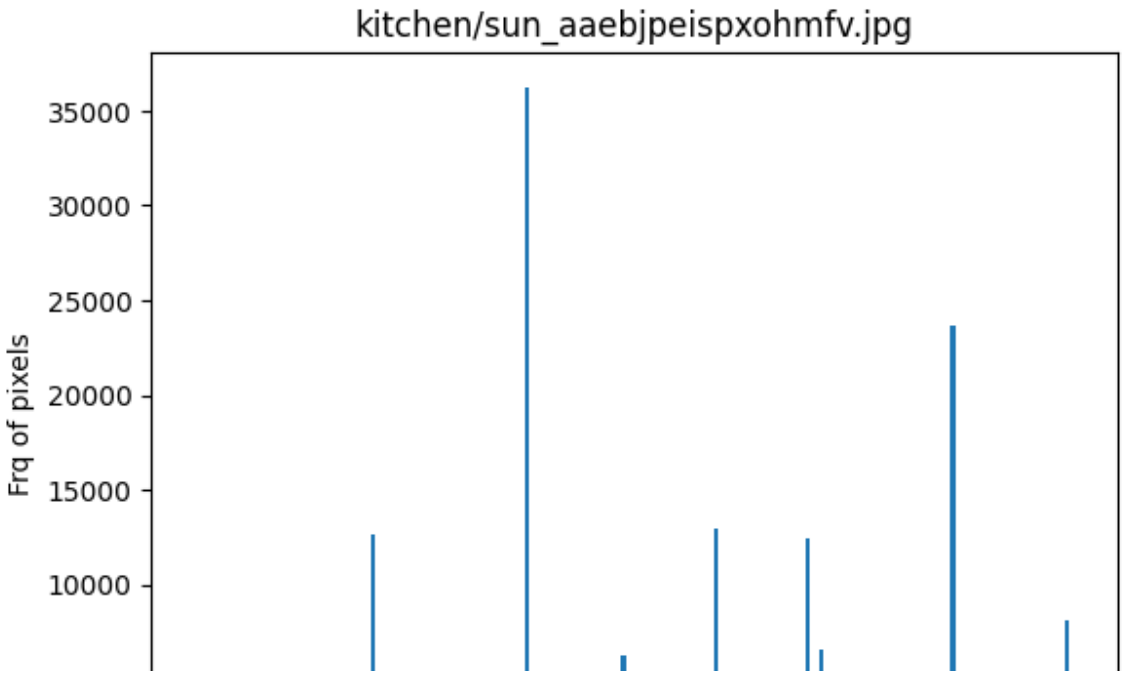

Frq of pixels
60000
40000
20000
0
0
25
50
75
100
125
150
175
200
Dict classes

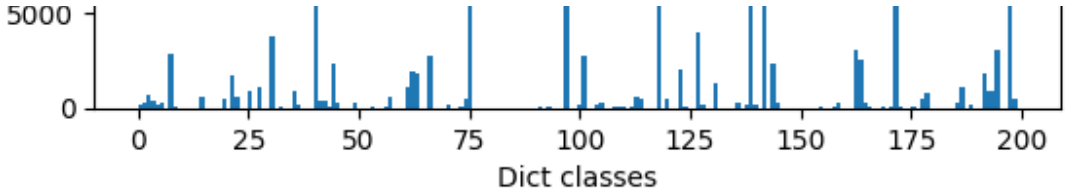


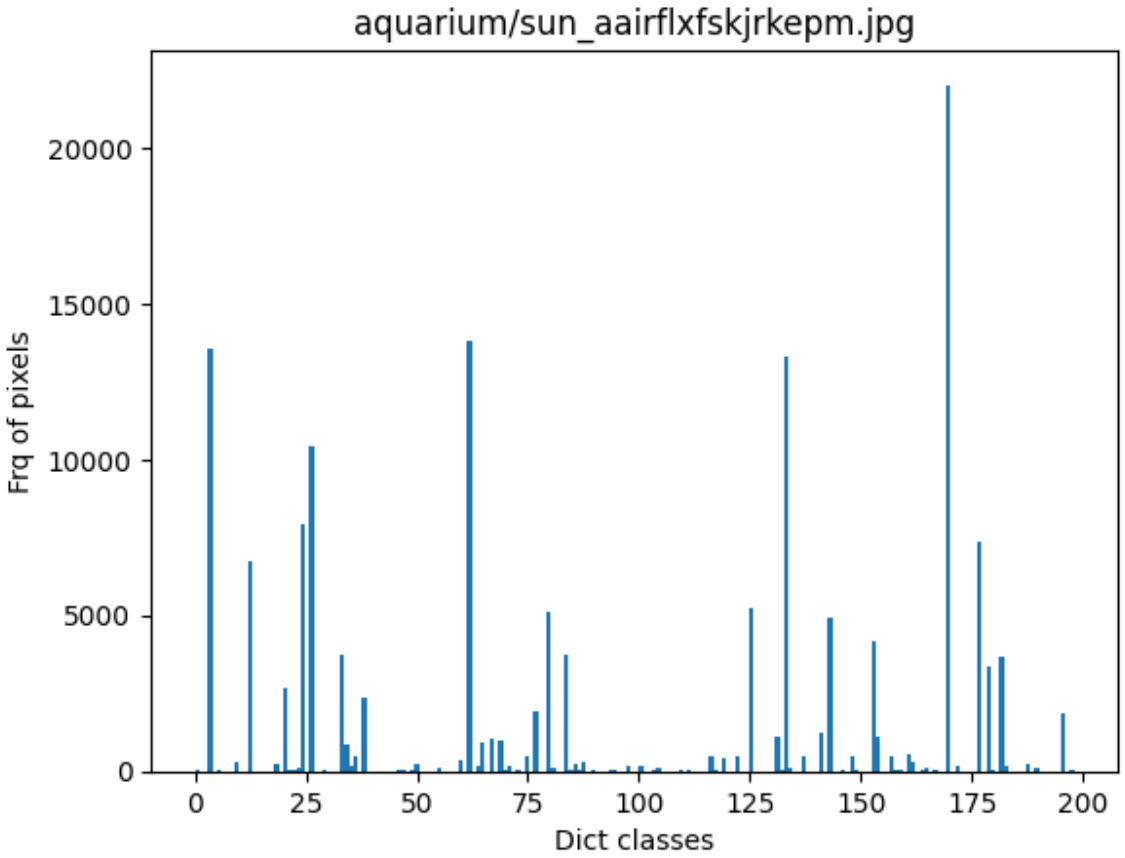

highway/sun_aacqsbumiuidokeh.jpg
50000
40000
30000
of pixels

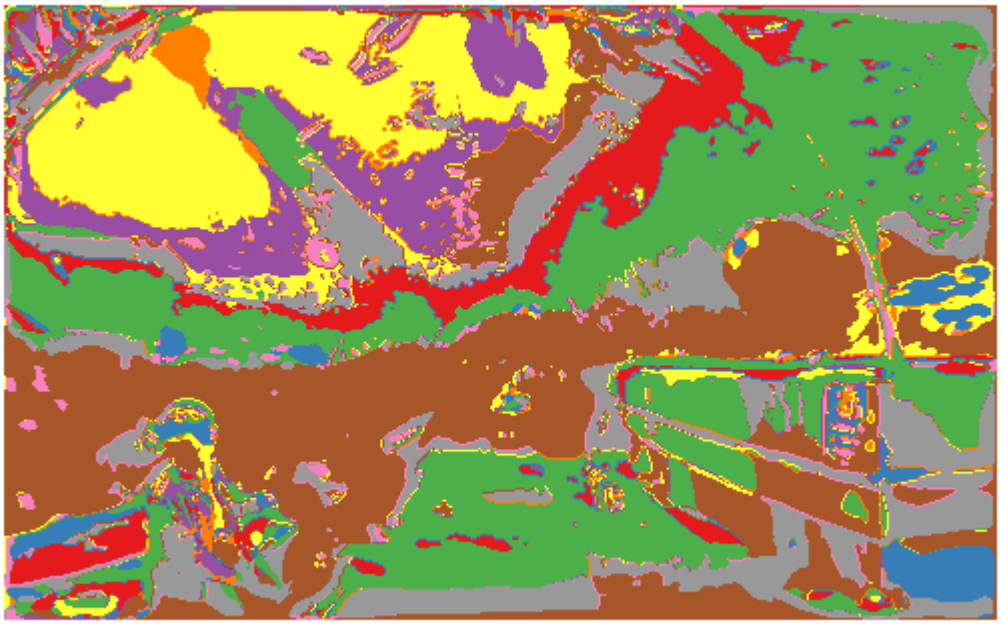

Frq
20000
10000
0
0
25
50
75
100
125
150
175
200
Dict classes




kitchen/sun_aaebjpeispxohmfv.jpg
35000
30000
25000
20000
15000
10000
Frq of pixels

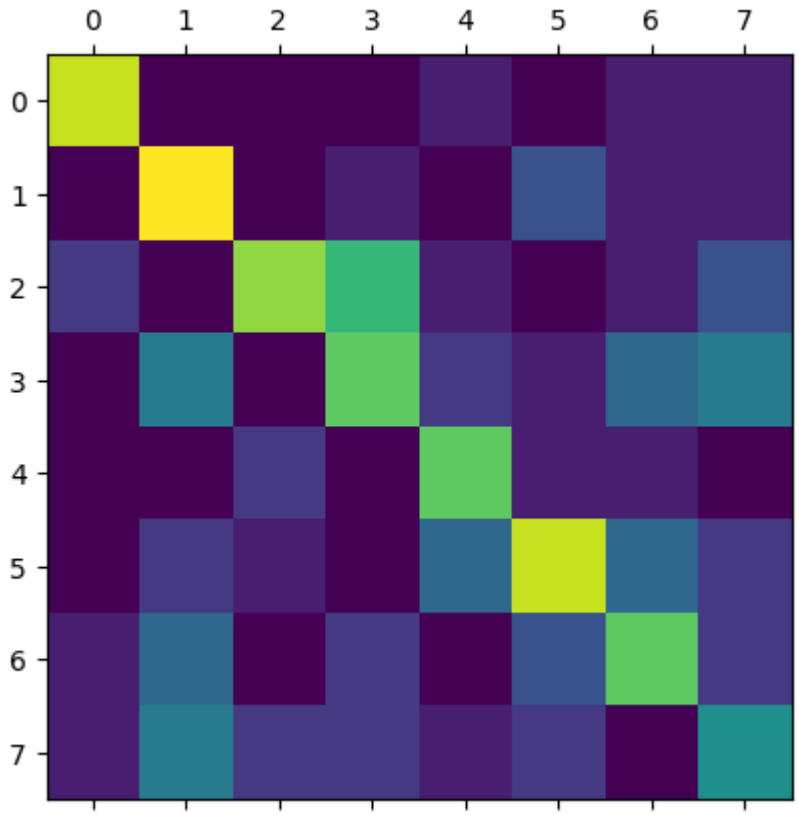

5000
0
0
25
50
75
100
125
150
175
200
Dict classes


aquarium/sun_aairflxfskjrkepm.jpg
20000
15000
10000
5000
0
Frq of pixels
0
25
50
75
100
125
150
175
200
Dict classes



Q3.1.1

Submit the visualization of Confusion Matrix and the Accuracy value



Accuracy = 0.48125

Q3.1.2 (5 points WriteUp):

As there are some classes/samples that are more difficult to classify than the rest using the bags-of-words approach, they are more easily classified incorrectly into other categories. **List some of these classes/samples and discuss why they are more difficult.**

For classes that have a lot of similar features its difficult to classify them seperately. For example, an aquarium that has water color as light blue and an image of a park where very less greenery but a lot of open sky (with a color similar to the one we dicsussed before), the classification will be easily messed up. Another example would be a kitchen and a laundromat. In both the cases there is a lot of structre to the enviornment. And therefore it will be difficult to classify them differently. In general, it becomes difficult to classify such similar classes/samples because there is similar structure of the individual features that the images consist of and hence while comparing the test image with the bag of words to create the histogram, similar features (that are not actually same and belong to some different class) get mapped incorrectly. And hence leading to incorrect classification.

Q3.1.3 [Extra Credit](10 points) Manually Graded:

Now that you have seen how well your recognition system can perform on a set of real images, you can experiment with different ways of improving this baseline system.

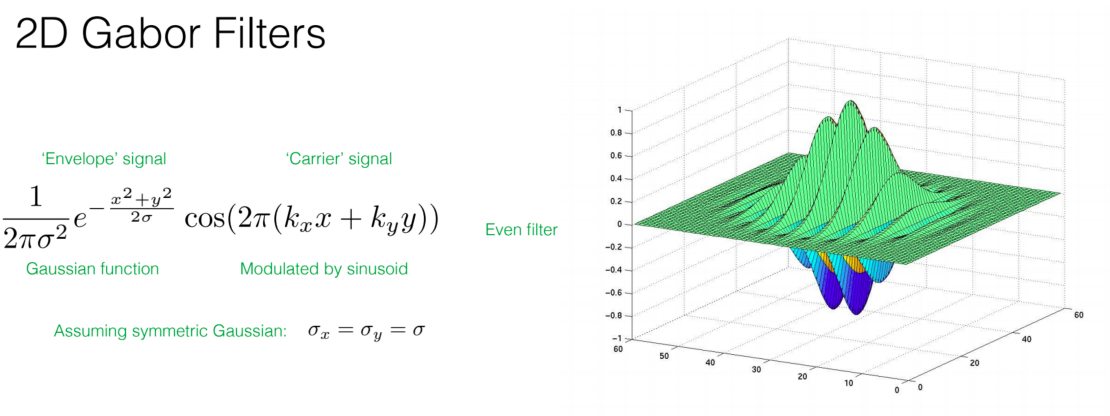
Include the changes, modification you made and the impact it had on accuracy.

Tune the system you build to reach around 65% accuracy on the provided test set (`data/test_data.npz`). **In your writeup, document what you did to achieve such performance: (1) what you did, (2) what you expected would happen, and (3) what actually happened.** Also, include a file called `custom.py/ipynb` for running your code.

YOUR ANSWER HERE

Q3.1.4 [Extra Credit] (5 points write up):

GIST feature descriptor: As introduced during the lecture, GIST feature descriptor is a feature extractor based on Gabor Filters. When we apply it to images, we have to implement the 2D Gabor Filters as described below



In your writeup: How does GIST descriptor affect the performance? Better or worse? Explain your reasoning?

YOUR ANSWER HERE

Q4.2.1 (2 points write up)

Report the confusion matrix and accuracy for your results in your write-up. Can you comment in your writeup on whether the results are better or worse than classical BoW - why do you think that is?

YOUR ANSWER HERE

Q4.3.2 [Extra Credit] (2 points write up)

Report the confusion matrix and accuracy for your ViT results in your write-up. Can you comment in your writeup on whether the results are better or worse than VGG - why do you think that is? A short answer is okay.

YOUR ANSWER HERE

References

[1] James Hays and Alexei A Efros. Scene completion using millions of photographs.ACM Transactions onGraphics (SIGGRAPH 2007), 26(3), 2007.

[2] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recogniz-ing natural scene categories. InComputer Vision and Pattern Recognition (CVPR), 2006 IEEE Conferenceon, volume 2, pages 2169–2178, 2006.

[3] Jian xiong Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recogni-tion from abbey to zoo.2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,pages 3485–3492, 2010.14

In [ ]: