Q1.1.1

What properties do each of the filter functions pick up? (See Figure 3) Try to group the filters into broad categories (e.g. all the Gaussians). Why do we need multiple scales of filter responses?

Answer: There are three types of filters that we are using in this part

- Gaussian - The Gaussian filter smooths the image, and removes noise in the image (usually higher frequencies) while retaining lower frequency components i.e., large scale changes over a larger area are preserved in the image
- Derivative of Gaussian - The Derivative of Gaussian picks up the higher frequency components or the edges in the image, in the orientation that the filter is applied.
- Laplacian of Gaussian - This is essentially the second order derivative of Gaussian filter, and is used to pick up rapid changes/edges in the image. On taking the second derivative of an edge, the edge would display as a zero crossing in the image. Laplacians are used to pick up blob features in the image

We need to apply the filter at different scales, so as to pick up edges of different frequencies (for derivative of gaussian) and pick up blobs in a scale invariant fashion (for Laplacian) and smooth the image at different rates (for gaussian filtering) in the image (Nyquist theorem). Smoothing is also required as the laplacian and derivative are highly sensitive to noise.

Q1.1.2

Apply all 20 filters on `aquarium/sun_aztvjgubyrgvirup.jpg`, and visualize the responses as an image collage. Submit the collage of 20 images in your write-up.

Answer:

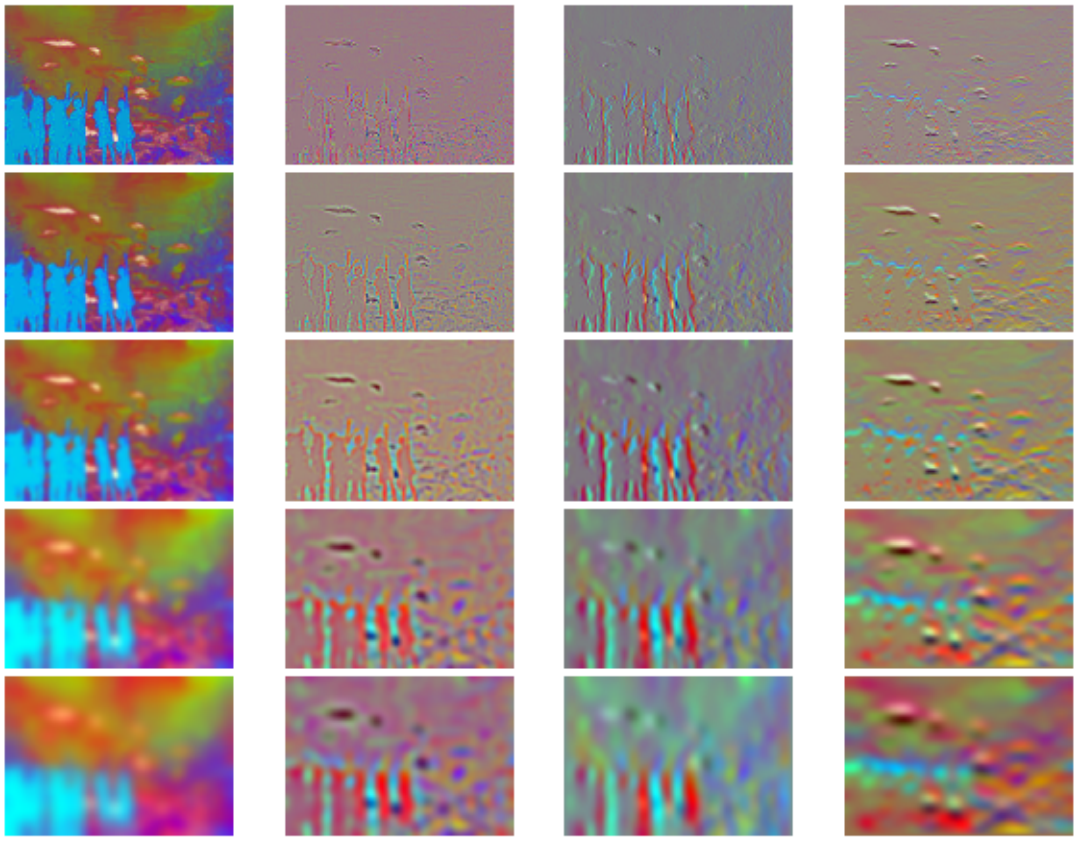
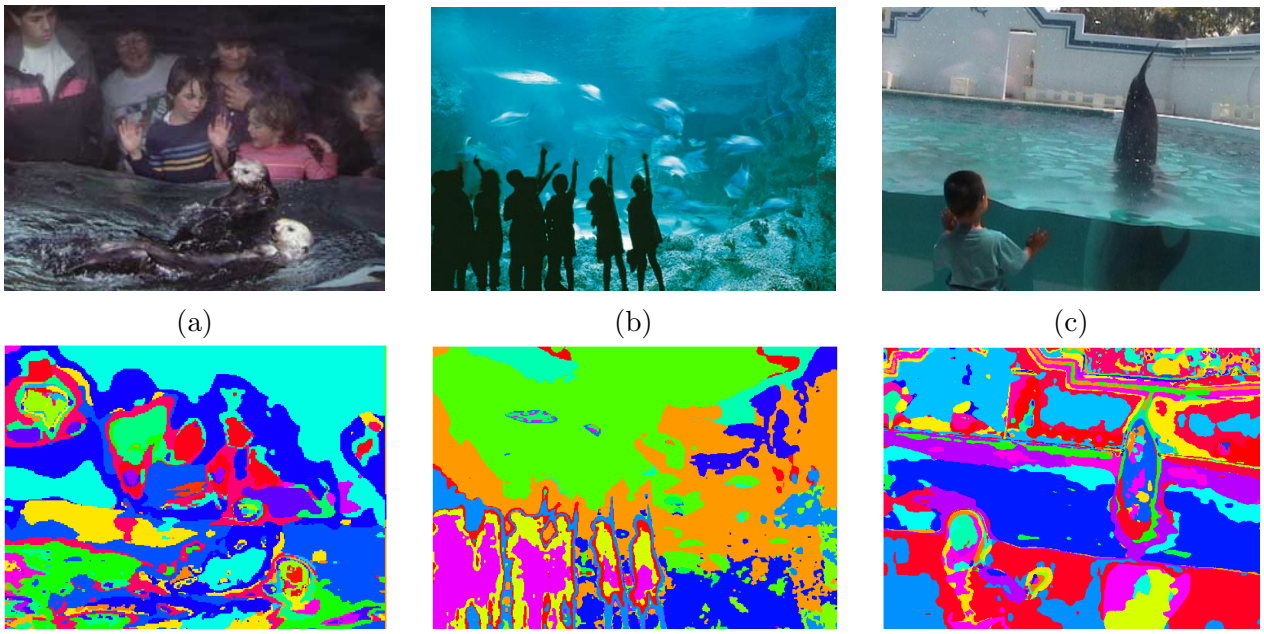


Figure 1: Filter responses for the required image: `sun_aztvjgubyrgvirup.jpg`

Q1.3

Visualize 3 wordmaps for each of three images from any one 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?

Answer:



(a) (b) (c)

(d) (e) (f)

Figure 2: Wordmap outputs for each of the images in the aquarium category

Comments: Each image is divided into 200 code-words (clusters), and every word-map is a representation of the image with each pixel classified as one of the learnt code-words. These word boundaries make sense, because we are clustering similar filter responses of image patches (within a threshold) to generate a representation of image as a combination or "wordmap" of these clusters/features. From images 2(c) and 2(f), we can notice that water regions with different textures are mapped to different features. Similarly, in images 2(b) and 2(e) the aquarium roof is mapped to a single cluster. Therefore, similar textures in the images are mapped to the same cluster (represented as a color in visualization).

Q2.5

Include the confusion matrix and your overall accuracy in your write-up.

Answer:

The confusion matrix obtained is as below:

$$\begin{bmatrix} 13 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13 & 0 & 1 & 0 & 1 & 3 & 0 \\ 1 & 0 & 13 & 3 & 5 & 2 & 0 & 1 \\ 2 & 2 & 1 & 12 & 0 & 2 & 3 & 4 \\ 1 & 1 & 0 & 0 & 10 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 & 6 & 13 & 0 & 1 \\ 1 & 3 & 0 & 0 & 2 & 1 & 12 & 2 \\ 2 & 3 & 2 & 3 & 0 & 1 & 1 & 7 \end{bmatrix}$$

Overall accuracy of the system with SPM matching = 58.125%

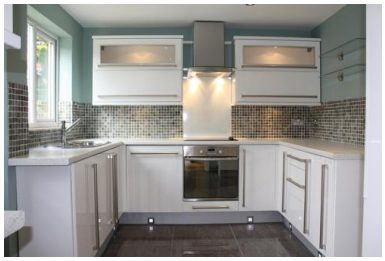
Q2.6

In your writeup, list some of these hard classes/samples, and discuss why they are more difficult than the rest.

Answer:

- There are 6 images wherein image category 5 (laundromat) has been mis-classified as category 4 (kitchen). This is because there are many similar patterns in between the two classes. Note the tiles, and the similarities between ovens and washing machines. Both of the classes are also indoor environments with structured surroundings.

(a) Mis-classified image of laundromat class



(b) Representative image of park class

- There are 3 images wherein image category 6 (waterfall) has been mis-classified as category 1 (park). These two classes are difficult to classify because of similar textures in the images of both the images i.e., there are many waterfall images, which have significant parts filled with grass, or trees, which would correspond to textures seen in the park training images. One such mis-classification is shown below in Fig 4a

(a) Mis-classified image of waterfall class

(b) Representative image of park class

Finally there are some other mistakes between category 2 (desert) and category 4 (kitchen), these are probably due to similarity in textures and color between wood panels (in kitchen images) and sand (in desert images).

Q3.2

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?

Answer: The confusion matrix result after running modified VGG16 CNN on the test set is as given below:

$$\begin{bmatrix} 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 17 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 26 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 23 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 19 \end{bmatrix}$$

Overall accuracy with VGG16 CNN is 97.5%

Comments: In my opinion, VGGnet performs better than classical BoW approach because:

Classical BoW approach represents the images as a *linear combination* of the clusters/features learnt. In this case, the linear combination is just 1-hot vector, fully discretizing each feature to the closest cluster. On the other hand, VGG is able to learn much more complex features, due to *non-linear combinations* of previous layer features. Further, In VGG the features are learnt in a hierarchical way, i.e., we cascade multiple convolutions with non-linear operations (layers further inside the VGG train with feature representations of image from previous layers). This provides us with highly complex features of the image, allowing us to classify images more robustly.