16-720B Computer Vision: Homework 1 Spatial Pyramid Matching for Scene Classification

Apoorv Singh

September 27, 2018

1 Representing the World with Visual Words

1.1 Extracting Filter Responses

Q 1.1.1 Properties of the filer:

1) Gaussian Filter:

This feature removes out the noise in the images. It kinds of smooth-ens out the image, making it free from unnecessary details in the image. For example, if you are taking a High-Definition picture of a beach, you don't want clear representation of every sand particle. Gaussian filter takes care of such unnecessary information. It is exactly opposite of what we do while sharpening the image.

2) Laplacian of Gaussian:

This filter detects the blobs in the image. It is a derivative filter - Used to find areas of rapid changes in the image. This filter is very noise-sensitive. So, it is common to pass gaussian filter before applying Laplacian filteG

3) Derivative of Gaussian in the x direction:

This filter picks out the rapid-changes (Edges) in the horizontal direction. Hence all the vertical lines (from the image) will be picked up after applying this filter.

4) Derivative of Gaussian in the y direction:

This filter picks out the rapid-changes (Edges) in the vertical direction. Hence all the horizontal lines (from the image) will be picked up after applying this filter.

Multiple scales of features are used to to pick things of different sizes (based on pixel they take up). For examples things that occupy less number of pixels will be picked up by smaller scale filter and things that occupy more number of pixels can be picked up by filter of larger scales.

1.2 Extracting Filter Responses

Values:

Alpha = 75

K = 100

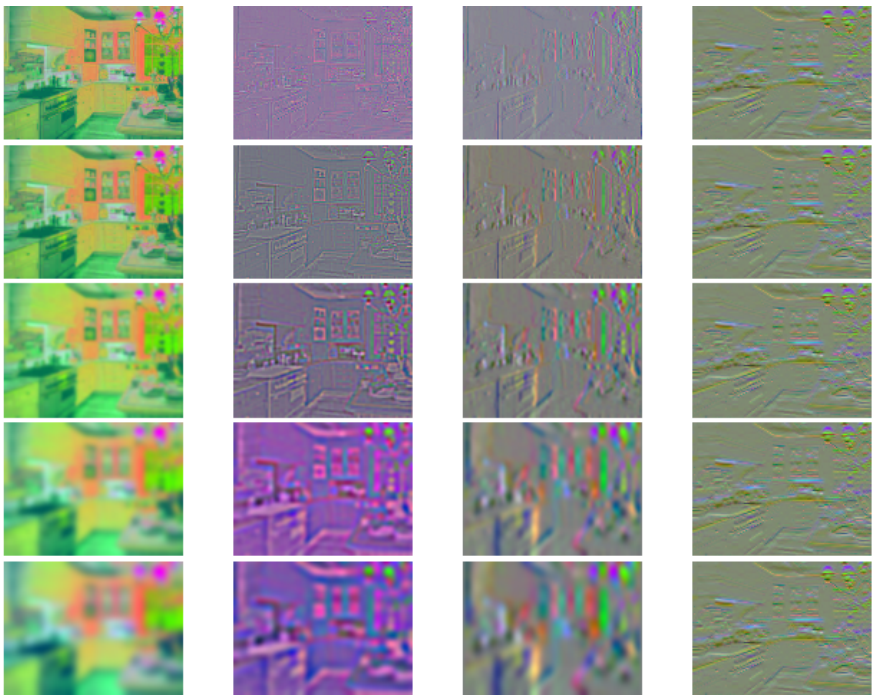

Figure 1: 5 * 4 images. 5 Scales, each of 4 filters.

Figure 2: Image one - Baseball$_{f}ield1$

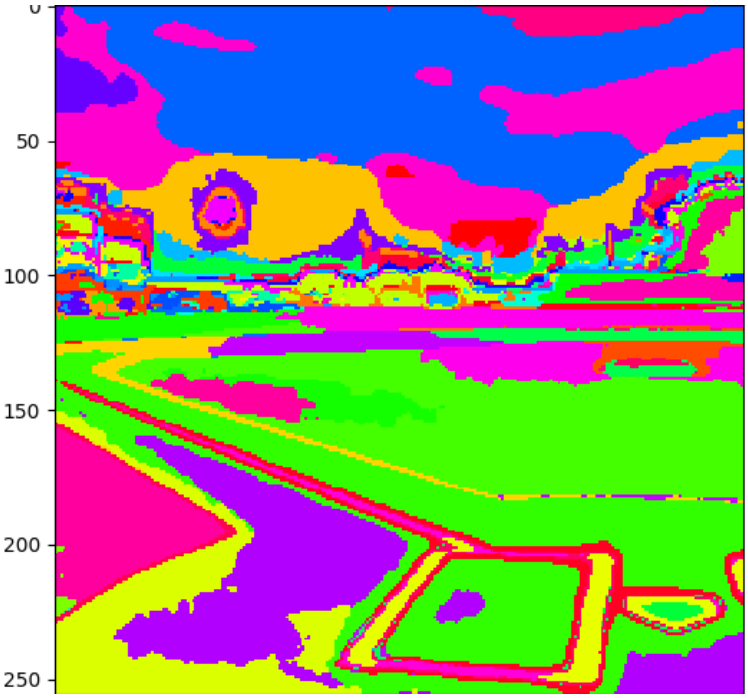

Figure 3: Image one - Baseball$_field1 - Wordmap$

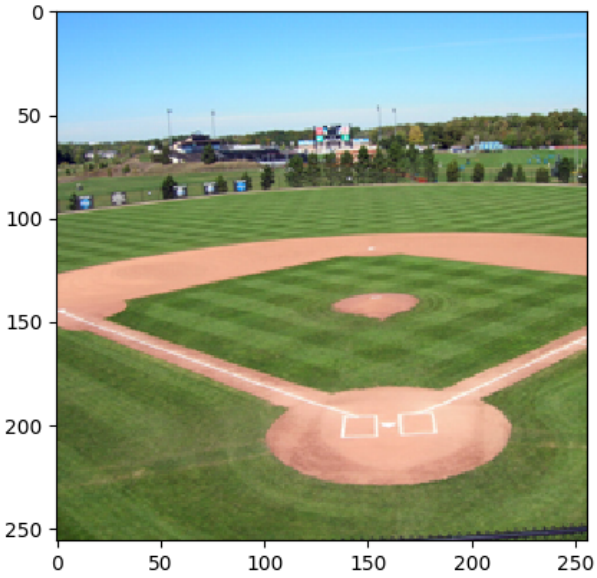

Figure 4: Image two - Baseball$_field2$

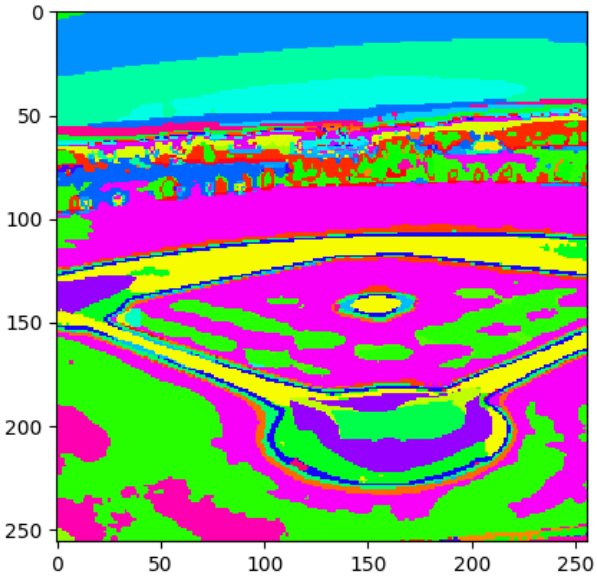

Figure 5: Image two - Baseball$_field2 - Wordmap$

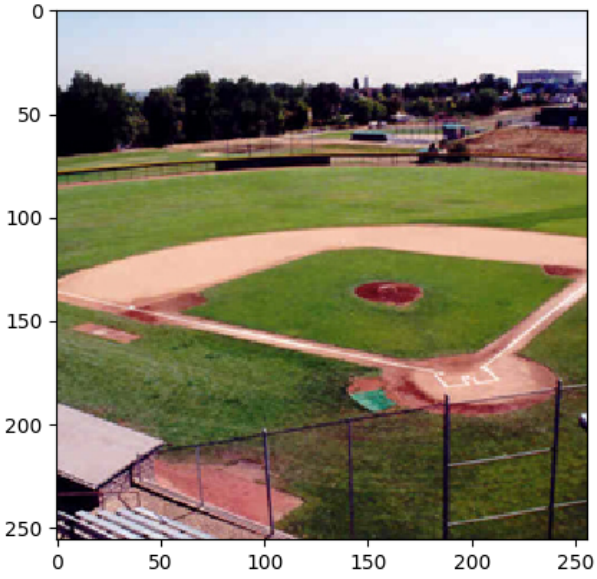

Figure 6: Image three - Baseball$_field3$

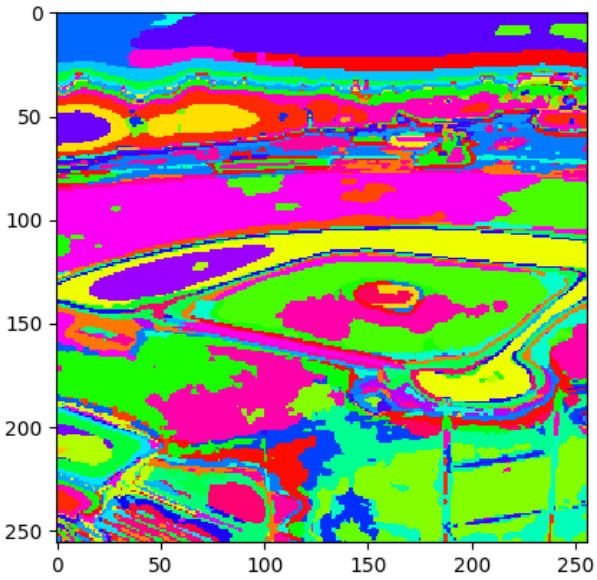

Figure 7: Image three - Baseball$_field3 - Wordmap$

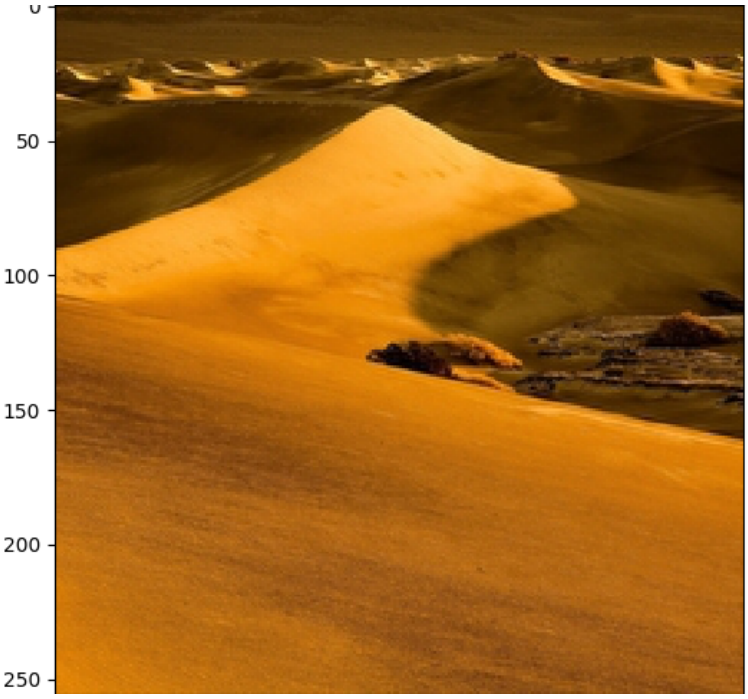

Figure 8: Image 2: Desert

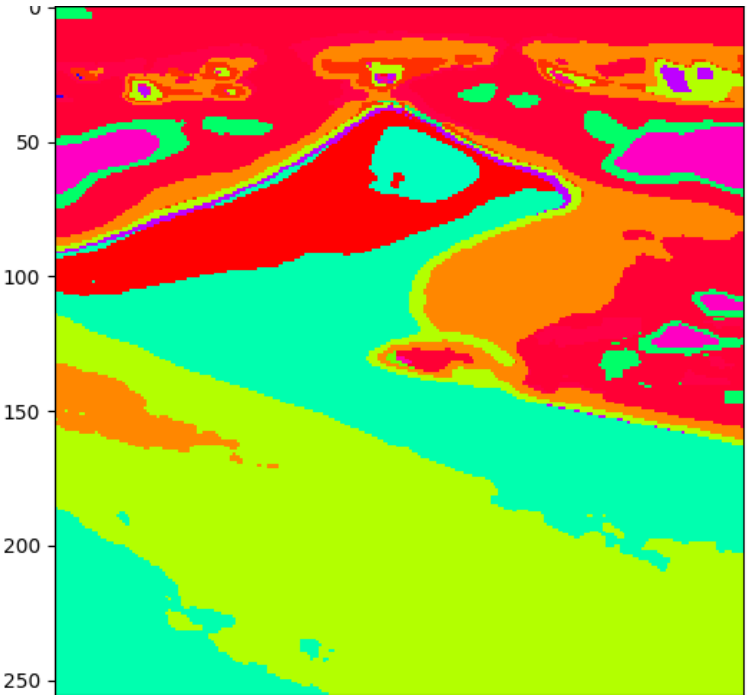

Figure 9: Image 2: Desert - Wordmap

Figure 10: Image 3: Laundromat

Figure 11: Image 3: Laundromat- Wordmap

1.3 Computing Visual Words

Similar objects are being represented by same color. For example if you look at the first three images. Square Tracks in the baseball field (On which the batsmen runs) is represented by similar color i.e. "yellow" in all the wordmap. So, this track has been stored as bag of word in this representation.

2 Build a recognition system

According to my confusion matrix following classes are having bad deficiencies: 1) Desert, 2) Windmill, 3) Kitchen

Reason for desert and Kitchen might be that there are lot of blank spaces in the image, hence it might be hard for the visual words to assign such a big values. Reason for Kitchen being bad at it, may be the intricacies of the objects in the kitchen. There are so many tiny objects and each of them are carrying different items. Bag of words probably is not able to pick up those tine objects.

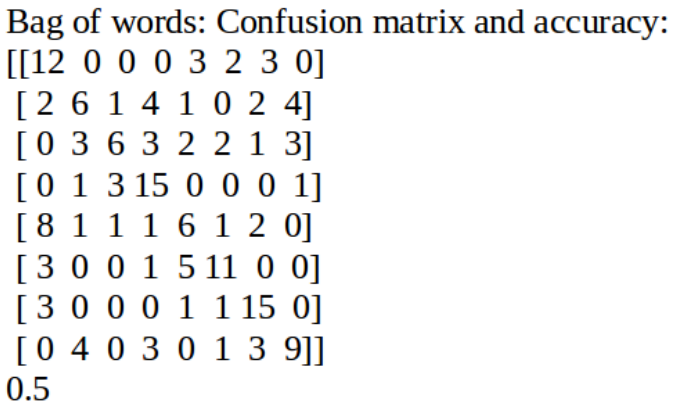

Figure 12: Confusion matrix and accuracy results for Bag of words method

3 Deep Learning Features - An Alternative to "Bag of Words"

Deep learning is a better approach because:

1) In deep learning lot of spatial features are considered are taken into account. But, in BOW we have taken only 3 layers and lot of information is lost due to that. In deep learning Maxpooling function is used to reduce the size, but it does in very efferent manner.

2) In deep learning we have used more number of filters compared to BOW method.

3) In BOW approach we have used random alpha pixels only. But in deep learning we have taken into account the whole image.

```
Deep learning: Confusion matrix and accuracy
[[20  0  0  0  0  0  0  0]
 [ 1 17  1  0  0  0  1  0]
 [ 0  0 19  1  0  0  0  0]
 [ 0  0  0 20  0  0  0  0]
 [ 0  0  0  0 19  1  0  0]
 [ 0  0  0  0  2 18  0  0]
 [ 0  0  1  0  0  0 19  0]
 [ 0  0  0  2  0  0  0 18]]
0.9375
```

Figure 13: Confusion matrix and accuracy results for Deep Learning