Computer Vision Final Assignment

# Checkpoint 1: Extracting Filter Responses

Below are 3 example filter responses produced by our code for a sample image. In order to view the last two as non-dark images, a value of 0.5 was added to each pixel

|  |  |
| --- | --- |
| A picture containing text, nature, shore  Description automatically generated |  |
|  | A picture containing text  Description automatically generated |

## Checkpoint 2: Mapping pixels to visual words

Here we take 3 random images and display the overlayed word map that resulted from using Random-chosen points and Harris algorithm-chosen points. In order to get these images, the labels with the image were converted into RGB, and then it was converted to BGR.

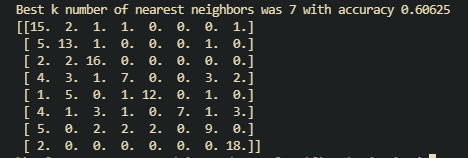
|  |  |  |
| --- | --- | --- |
| A picture containing text, indoor, building, ceiling  Description automatically generated |  |  |
| A picture containing conference room  Description automatically generated |  |  |
| A picture containing indoor, bed, room, bedroom  Description automatically generated |  |  |

Checkpoint 3: Confusion Matrices

Below are the resulting confusion matrices that show the accuracy of our overall program (which images were correctly labeled). We ran the system with *𝛼* = 200 and *K* = 500. Based on this data, our system performed well with chi-squared distance function (both with random points and top Harris points).

|  |  |
| --- | --- |
|  |  |

Using Harris + Chi Squared for kNN, our best accuracy was 60.625% with k = 7.



**Program requirements:**

* CV2
* Numpy
* Scipy
* Skimage

**Execution instructions:**

If you wish to run one script that will execute the whole system, you may run “main.py”.

Our values right now are hard coded as *𝛼* = 200 and *K* = 500 in “computeDictionary.py”.

**Known issues:**

None that we know of.

**Final comments:**

We will include some sample saved pickle files in the submission.