**ImageMatchX Write-Up**

**Task 1**

**1.1**This task was completed by filling in the codes in extractFilterResponses.py file, with the help of the given createFilterBank.py file.

Below is an image from the dataset and any 3 (out of 20) of its filter responses.

*desert\sun\_afferxhafrjnpuri*

3 Filter Responses (in LAB colour space)

|  |  |  |
| --- | --- | --- |
| Gaussian (Scale=2) | Laplacian (Scale=4) | X Gradient (Scale=1) |
| img | img | img |

**1.2**  
This task was done by implementing the files getHarrisPoints.py and getRandomPoints.py.

Below are the results of both the methods on three images from the given dataset.

Harris Points

|  |  |  |
| --- | --- | --- |
| *landscape\* *sun\_abvvlqznpdszhjnh* | *bedroom/sun\_aacyfyrluprisdrx* | *airport/sun\_afbxsdfksjhcunpb* |
| img | img | img |

Random Points

|  |  |  |
| --- | --- | --- |
| *landscape\* *sun\_abvvlqznpdszhjnh* | *bedroom/sun\_aacyfyrluprisdrx* | *airport/sun\_afbxsdfksjhcunpb* |
| img | img | img |

**1.3**  
For this task, the files getDictionary.py and computeDictionary.py were used. My dictionaries have 50 visual words each. The files dictionaryHarris.pkl and dictionaryRandom.pkl were generated.

**Task 2**

**2.1**  
The python file getVisualWords.py was implemented for this task. Then, the file batchToVisualWords.py was run to generate word-maps for all images, using both, Harris points and Random points.

Below are top-left 12x12 sections of the word-maps of 3 images from desert class and 3 from rainforest class, using both dictionaries, Harris points and Random points.

Desert

|  |  |  |
| --- | --- | --- |
| Image Name | Word-map using Harris Dictionary | Word-map using Random Dictionary |
| *sun\_aafqfjpechscyidz* | [[40 30 30 30 30 30 30 30 30 30 30 30]  [ 8 40 30 30 30 30 30 30 30 30 30 30]  [ 8 49 49 49 49 49 33 20 37 37 37 37]  [ 8 49 49 49 49 33 37 37 37 37 37 37]  [ 8 49 49 49 20 37 37 37 37 37 37 37]  [ 8 49 49 49 37 37 37 37 37 37 37 37]  [ 8 49 49 45 37 37 37 37 37 37 37 37]  [ 8 9 45 45 37 37 37 37 37 37 37 37]  [ 8 9 45 37 37 37 37 37 37 37 37 37]  [ 8 9 45 37 37 37 37 37 37 37 37 37]  [ 8 9 45 37 37 37 37 37 37 37 37 37]  [ 8 9 45 37 37 37 37 37 37 37 37 37]] | [[41 41 41 41 41 41 41 41 41 41 41 41]  [43 41 41 41 41 41 41 41 24 24 24 24]  [43 43 24 24 24 24 24 24 24 24 24 24]  [43 43 21 21 21 21 45 45 45 45 45 45]  [43 43 21 21 21 21 45 45 45 45 45 45]  [43 43 21 21 21 21 45 45 45 45 45 8]  [43 43 21 21 21 21 45 45 45 45 45 8]  [43 43 21 21 21 45 45 45 45 45 8 8]  [43 43 21 21 21 45 45 45 45 8 8 8]  [43 43 21 21 45 45 45 45 45 8 8 8]  [43 43 21 21 45 45 45 45 5 8 8 8]  [43 43 21 21 45 45 45 5 8 8 8 8]] |
| *sun\_acqlitnnratfsrsk* | [[40 30 30 30 30 30 30 30 30 30 30 30]  [ 8 40 49 49 30 30 49 49 49 49 49 49]  [ 8 49 49 49 49 37 37 37 37 37 37 37]  [ 8 49 49 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]  [ 8 49 37 37 37 37 37 37 37 37 37 37]] | [[41 41 41 41 41 41 41 41 41 41 41 41]  [43 41 41 41 41 24 24 24 24 24 24 24]  [43 43 24 24 24 24 24 24 24 24 24 24]  [43 43 21 45 45 45 45 45 45 45 8 8]  [43 43 21 45 45 45 45 45 8 8 8 8]  [43 43 45 45 45 45 45 8 8 8 8 8]  [43 43 45 45 45 45 8 8 8 8 8 8]  [43 43 45 45 45 8 8 8 8 8 8 8]  [43 43 45 45 8 8 8 8 8 8 8 8]  [43 43 45 8 8 8 8 8 8 8 8 8]  [43 43 8 8 8 8 8 8 8 8 8 8]  [43 43 8 8 8 8 8 8 8 8 8 8]] |
| *sun\_acrqldhmwdraspza* | [[40 30 30 30 30 30 30 30 30 30 30 30]  [ 8 40 49 49 49 49 49 49 49 49 30 30]  [ 8 49 49 49 49 49 49 49 49 49 20 20]  [ 8 9 49 49 49 20 20 20 20 20 20 20]  [ 8 9 45 45 45 45 45 45 15 15 20 20]  [ 8 49 45 45 45 37 37 15 15 15 15 15]  [ 8 49 45 37 37 37 37 15 15 15 15 15]  [ 8 49 45 37 37 37 37 15 15 15 15 15]  [ 8 49 20 20 37 37 37 37 15 15 15 15]  [ 8 49 20 20 20 20 37 15 15 15 15 15]  [ 8 49 20 20 20 20 20 20 20 15 15 15]  [ 8 49 49 20 20 20 20 15 15 15 15 15]] | [[41 41 41 41 41 41 41 41 41 41 41 41]  [43 41 24 24 24 24 24 24 24 24 24 24]  [43 43 24 24 24 24 24 24 24 24 24 24]  [43 43 21 21 21 21 21 21 21 45 45 45]  [43 43 21 21 21 21 21 45 45 45 45 45]  [43 43 21 21 21 45 45 45 45 45 45 45]  [43 43 21 45 45 45 45 45 45 45 45 45]  [43 43 21 45 45 45 45 45 45 45 45 45]  [43 43 45 45 45 45 45 45 45 45 45 45]  [43 43 45 45 45 45 45 45 45 45 45 45]  [43 43 21 45 45 45 45 45 45 45 45 45]  [43 43 21 45 45 45 45 45 45 45 45 45]] |

Rainforest

|  |  |  |
| --- | --- | --- |
| Image Name | Word-map using Harris Dictionary | Word-map using Random Dictionary |
| *sun\_aacykkmwtykjfldf* | [[40 40 30 30 30 30 30 30 30 30 30 30]  [ 8 8 23 23 12 12 12 12 49 30 30 30]  [ 8 8 9 9 29 29 29 15 49 49 49 12]  [ 8 18 9 9 29 29 41 15 49 49 49 29]  [ 8 8 49 9 29 29 33 15 45 9 29 29]  [ 8 49 49 49 49 29 33 15 45 45 45 29]  [ 8 49 9 9 9 29 29 41 15 45 45 41]  [ 8 9 45 45 45 29 29 41 15 15 45 41]  [ 8 9 45 45 45 29 29 41 15 15 45 41]  [ 8 49 45 45 45 41 41 41 15 15 41 41]  [ 8 49 20 34 34 41 41 15 15 15 15 15]  [ 8 9 20 20 33 33 19 15 15 15 15 19]] | [[43 41 41 41 41 41 41 41 41 41 41 41]  [ 9 43 43 24 24 24 24 24 24 41 41 41]  [ 9 43 43 33 33 49 21 45 24 24 24 24]  [ 9 43 43 33 33 21 21 45 21 33 33 21]  [ 9 43 43 33 33 21 21 45 21 33 21 21]  [ 9 43 33 33 33 33 21 45 21 21 21 21]  [ 9 43 33 33 33 33 21 21 45 21 21 21]  [ 9 43 33 21 33 21 21 21 45 45 21 21]  [ 9 43 21 21 21 21 21 21 45 45 21 21]  [43 43 28 21 21 21 21 21 45 45 45 42]  [43 43 28 21 21 21 45 45 45 45 42 42]  [ 9 43 28 28 28 45 45 42 42 42 42 42]] |
| *sun\_aainjqctyswzidnk* | [[40 30 30 30 30 30 30 30 30 30 30 30]  [ 8 40 49 49 49 49 49 49 49 49 49 49]  [ 8 49 49 49 49 49 49 49 49 49 49 49]  [ 8 49 49 49 49 49 20 20 20 20 20 20]  [ 8 49 49 49 20 20 20 20 20 20 20 20]  [ 8 49 49 45 20 20 20 20 20 20 20 20]  [ 8 49 49 45 20 20 20 20 20 15 15 15]  [ 8 49 49 45 20 20 20 15 15 15 15 15]  [ 8 49 45 45 20 20 15 15 15 15 15 15]  [ 8 49 45 45 20 20 15 15 15 15 15 15]  [ 8 9 45 45 20 15 15 15 15 15 15 15]  [ 8 9 45 45 20 15 15 15 15 15 15 15]] | [[41 41 41 41 41 41 41 41 41 41 41 41]  [43 41 24 24 24 24 24 24 24 24 24 24]  [43 43 24 24 24 24 24 24 24 24 24 24]  [ 9 43 33 33 21 21 21 21 21 21 21 21]  [ 9 43 33 21 21 21 21 21 21 21 21 21]  [ 9 43 33 21 21 21 21 21 21 21 21 21]  [ 9 43 33 21 21 21 21 21 21 21 21 21]  [ 9 43 33 21 21 21 21 21 21 21 45 45]  [ 9 43 33 21 21 21 21 21 21 45 45 45]  [ 9 43 33 21 21 21 21 21 21 21 21 21]  [ 9 43 33 21 21 21 21 21 21 21 21 21]  [ 9 43 33 21 21 21 21 21 45 45 21 21]] |
| *sun\_aaispeummyknccnf* | [[40 40 30 30 30 30 30 30 30 30 30 30]  [ 8 8 23 49 49 12 12 49 49 49 49 49]  [ 8 8 9 9 9 49 49 49 49 49 49 49]  [ 8 31 9 9 9 9 9 9 49 29 29 29]  [ 8 9 9 27 45 9 9 9 9 29 29 29]  [ 8 9 45 45 45 45 9 9 9 29 29 45]  [ 8 9 45 45 45 45 45 9 9 39 39 39]  [ 8 9 9 45 45 15 14 39 39 39 39 39]  [18 38 45 9 45 34 15 14 39 39 14 14]  [18 38 45 45 39 15 22 15 15 15 15 15]  [ 8 17 34 14 14 20 20 22 22 34 34 22]  [ 8 49 20 22 17 49 7 34 34 34 34 15]] | [[41 41 41 41 41 41 41 41 41 41 41 41]  [43 43 24 24 24 24 24 24 24 24 24 24]  [ 9 43 33 33 33 33 33 33 33 33 33 33]  [ 9 43 33 33 33 33 33 33 33 33 21 21]  [ 9 43 21 48 48 21 33 33 33 21 21 21]  [ 9 43 21 21 21 21 21 33 49 49 21 21]  [43 43 33 21 21 21 48 48 48 48 48 48]  [ 9 43 33 33 21 21 48 48 48 48 48 48]  [ 9 43 43 33 49 21 45 48 48 48 42 42]  [ 9 43 21 48 48 45 45 32 47 32 47 47]  [ 9 43 32 32 32 32 28 32 32 32 32 32]  [ 9 43 28 32 32 43 33 32 32 32 32 32]] |

**2.2**  
getImageFeatures.py was implemented to complete this task.

Below are histograms of the same 6 images (3 from desert and 3 from rainforest) whose partial word-maps have been shown in the write-up of task 2.2 above.

Desert

|  |  |  |
| --- | --- | --- |
| Image Name | Histogram using Harris Dictionary | Histogram using Random Dictionary |
| *sun\_aafqfjpechscyidz* |  |  |
| *sun\_acqlitnnratfsrsk* |  |  |
| *sun\_acrqldhmwdraspza* |  |  |

(In the above table, it is interesting to note that the first and the third images are quite similar when seen in the data set and their histograms are also quite similar (for both, Harris and Random). The peaks can be observed at the same visual word number.)

Rainforest

|  |  |  |
| --- | --- | --- |
| Image Name | Histogram using Harris Dictionary | Histogram using Random Dictionary |
| *sun\_aacykkmwtykjfldf* |  |  |
| *sun\_aainjqctyswzidnk* |  |  |
| *sun\_aaispeummyknccnf* |  |  |

**2.3**  
visionHarris.pkl and visionRandom.pkl files were generated by creating and running getVisionPkls.py file.

**Task 3**

**3.1**  
The python file getSimilarImage.py was created. It has a function get\_similar\_image which takes inputs as one image (say A) and a set of images (say B) and also a method (Harris or Random). This function returns the most similar image to A from set B, along with the similarity score. This uses the histograms of the images to find similarity. Based on the similarity scores, this function returns the most similar image.

To find the similarity, I created three algorithms. Those are:

1. Cosine similarity: Based on the cosine of the angle between the histogram vectors).
2. Euclidean similarity: Based on the ratio of original and maximum values of the distance between the two histogram vectors.
3. Intersection over Union (IoU)

All the three algorithms were giving nearly the same accuracy of about 50% (with IoU being slightly inferior). Then, I tested 2 new algorithms to find the similarity scores.

1. max(cosine similarity, euclidean similarity, iou similarity)
2. min(cosine similarity, euclidean similarity, iou similarity)

I found the first one (max of all three) to be the worst indicator but the second method (min of all three) was found to be better than three individual algorithms.

Finally, I realised that the range of values returned by ‘cosine’, ‘euclidean’ and ‘iou’ are different. Thus, I found a better similarity score algorithm which was *min(0.9\*(cosine similarity), euclidean similarity, 1.3\*(iou similarity))* because the values of 0.9\*cosine, euclidean and 1.3\*iou lied approximately in the same range. Finally, I fine tuned these parameters by running the model multiple times (500+ times) to finally come up with the following formula for the similarity score:

*min(0.873\*(cosine similarity), 0.983\*(euclidean similarity), 1.278\*(iou similarity))*

It was kept in mind that two images might look very similar even if they belong to two different classes (airport, bedroom, etc.).

The following are the results when the image (A) is selected randomly from the test set and the image set B is a set of unique images from 200 randomly selected images from the train set.

Using Harris Points

|  |  |  |
| --- | --- | --- |
| Test Image | Most Similar Image Found | Similarity Score |
| *rainforest/sun\_ablovgvaufrdjgvt.jpg* | *rainforest/sun\_allebkcnakigqrhk.jpg* | 85.77% |
| *auditorium/sun\_adrraoxyfqegzvou.jpg* | *auditorium/sun\_aloyjcjigqudtjmr.jpg* | 77.61% |
| *airport/sun\_aesyuxjawitlduic.jpg* | *auditorium/sun\_auvnalsffqkramht.jpg* | 86.37% |
| *bedroom/sun\_abpmyykpuijkvxbq.jpg* | *bedroom/sun\_aohayxcvefirmdaw.jpg* | 85.13% |
| *airport/sun\_aeziduzkrhfvkqvy.jpg* | *campus/sun\_ayhpvcbprplqkoyt.jpg* | 86.19% |
| *desert/sun\_adpbjcrpyetqykvt.jpg* | *desert/sun\_bqquzkzrdmcddrmm.jpg* | 65.17% |

Using Random Points

|  |  |  |
| --- | --- | --- |
| Test Image | Most Similar Image Found | Similarity Score |
| *desert/sun\_acxdijcmkawhkhob.jpg* | *desert/sun\_bshburneneixsrvy.jpg* | 61.06% |
| *bedroom/sun\_aazccosvjonbhrff.jpg* | *auditorium/sun\_aonudvgihuzzujwz.jpg* | 61.92% |
| *football\_stadium/sun\_atwkqosabmxbzymv.jpg* | *football\_stadium/sun\_byvvwensmjdeeifn.jpg* | 73.70% |
| *campus/sun\_agruqaefvrvumwod.jpg* | *campus/sun\_bugunyckbqwnedtt.jpg* | 68.66% |
| *auditorium/sun\_abbbtjsssstmdjja.jpg* | *auditorium/sun\_akchvyhqfvdcbaly.jpg* | 69.92% |
| *landscape/sun\_aduxybjjxgfhwggu.jpg* | *landscape/sun\_awybospjclnunhnq.jpg* | 60.90% |