Assignment 7

Computer Vision
Zeno Sambugaro 785367

October 31, 2019

Exercise 1

a. Inverse document frequency of the five terms:
cat = 4.321928094 dog = 2.321928094 mammals = 5.6438561809 mouse = 3.321928094
pet = 0.736965594

b. Term frequencies:
Query
'cat': 0.25, 'mouse': 0.25, 'mammals' : 0.25, 'pet' : 0.25
Document 1
'dog': 0.067, 'pet': 0.2, 'mouse': 0.067, 'cat': 0.067
Document 2
'dog': 0.143, 'cat': 0.143, 'mouse': 0.143, 'mammals': 0.143
Document 3
'dog': 0.083, 'cat': 0.083, 'mouse': 0.143, 'cat': 0.167

c. Histograms for tf-idf weighted word occurrence are in appendix.

d. Cosine similarity evaluation:

$$sim(d_j, q) = \frac{d_j \cdot q}{||d_j||||q||} = \frac{\sum_{i=1}^{V} d_j(i) \times q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2}\sqrt{\sum_{i=1}^{V} q(i)^2}}$$

e. similarity(Query,Document 1): 0.6291036970635482
similarity(Query,Document 2): 0.9546948111493485
similarity(Query,Document 3): 0.6430077230767424

f. Relative ranking of the documents:
Document 2: 0.95, Document 3: 0.64
Document 1: 0.63
Document 2 is most similar and Document 1 is the least similar.

Exercise 2

Precision
precision = # relevant / # returned
precision = 300 / 350 = 0.857

Recall
recall = # relevant / # total relevant
recall = 300 / 500 = 0.600

Exercise 3

a. **Question:** Note the change in density of detections across the image. Why does it change? Will it be a problem for matching? How could it be avoided?
The change in density of detections across the image is caused by the variation of the contrast. The less is the contrast the more difficult is to find detections. It is a problem since it cause a loss of information in low contrast area of the images. To obtain resistance against this type of problem we can use an adaptive threshold for the number of feature found in the image. This threshold value has to increase if the quality of the image is high and the image is not saturated or decrease otherwise.

Question: Occasionally, a feature is detected multiple times, with different orientations. This may happen when the orientation assignment is ambiguous. Which kind of image structure would result in ambiguous orientation assignment?
An image structure that would result in ambiguous orientation assignment is corner, in fact SIFT works by computing the gradient of orientation at each pixel within a region around the keypoint. Then a histogram is formed by all of the orientation gradients, the highest peak in this histogram and up to three further peaks id larger than 80% of the highest peak are taken. In the case of the corner we can have gradients similar in module but with different orientation.

b. **Question:** Note the descriptors are computed over a much larger region (shown in blue) than the detection (shown in green). Why?
Because the assignment algorithms are based on a local neighborhood R around the keypoint. Like most local feature descriptors, they assume that this neighborhood is approximately rigid and planar. It is intuitively clear that the robustness to small keypoint shifts and image noise increases with an increasing size of R. So what they tried to do is to find a trade-off between the robustness and the loss of information caused by a too big neighborhood.

Question: Notice that there are many mismatches. Examine some of the mismatches to understand why the mistakes are being made. For example, is the change in lighting a problem? What additional constraints can be applied to remove the mismatches?
The mismatches are caused by many factors, among them there is the problem of

the lighting, in fact some features are difficult to be recognized since the following saturation. The key factor that affects these mismatches is the presence of descriptors similar to the right match but also similar to other ones in the image. To solve this problem we can implement Lowe's second nearest neighbour identifying distinctive matches by a threshold on the ratio of first to second Nearest Neighbour distances.

c. **Question:** Examine some of the remaining mismatches to understand why they have occurred. How could they be removed?
In order to remove the remaining mismatches the algorithm can be improved using the spatial consistency of the descriptors.In particular we build a similarity function using the correspondences (obtaining translation parameters, angle of rotation and scaling), then using RANSAC we can decide if they are inliers or not and considered them into the final solution accordingly. In this way we avoid to consider pairs of descriptors linked to totally different similarity transformation, which in most of the cases are false matches.

PARTE 2

Question: The transformation between the images induced by the plane is a planar homography. The detections are only affine co-variant (not as general as a planar homography). So how can descriptors computed on these detections possibly match?
The descriptors compute on the affine detections found some match because affine transformation approximates viewpoint changes for roughly planar objects and roughly orthographic cameras. It is a first order approximation of the projective transformation.

PARTE 3

a. **Question:** The size of the vocabulary (the number of clusters) is an important parameter in visual word algorithms. How does the size affect the number of inliers and the difficulty of computing the transformation?
Increasing the size of the vocabulary there will be more inliners so the accuracy will be improved. The difficulty of computing the transformation will increase as the complexity will. In addition as the size of the vocabulary increases, so does the vector representation of documents. Thus if the vocabulary is too wide the image may contains very few words in the vocabulary resulting in sparse vectors. Sparse vectors require more memory and computational resources resulting in an increase of the complexity.

Question: In the above procedure the time required to convert the descriptors into visual words was not accounted for. Why?
The time required to convert the descriptors in not taken into account because we are only comparing the time spent by the algorithms to match the two image. Thus in the case of SIFT compare the descriptors of the two images and in the case of "Bag of word" compare the visual words.

Question: What is the speedup in searching a large, fixed database of 10, 100, 1000 images?

The speed up lays in the fact that the increment is linear, in fact the time spent by the two algorithm match operation for two images is:

- ANN algorithm: 0.0399 s
- BoW algorithm: 0.0081 s

for 10 images:

- ANN algorithm: 0.3990 s
- BoW algorithm: 0.0810 s

for 100 images:

- ANN algorithm: 3.990 s
- BoW algorithm: 0.810 s

for 1000 images:

- ANN algorithm: 39.9 s
- BoW algorithm: 8.1 s

b. **Question:** Why does the top image have a score of 1?
It has a score of one because we are measuring the similarity between two unit vector, thus using the cosine similarity the best result will be one.

c. **Question:** Why is the top score much larger than 1 now? Are the retrieval results improved after geometric verification?
The top score is much larger than one because it counts the number of similar inliners between the images, since an image has around 1000 SIFT descriptors the top score will be near this quantity. The retrieval has been improved, in the results from the first we can count some erroneously matched images, while in the results of the improved algorithm no. This is because we try to match also the position in the image of the descriptors instead of only compare the number of words in the images.

Appendix

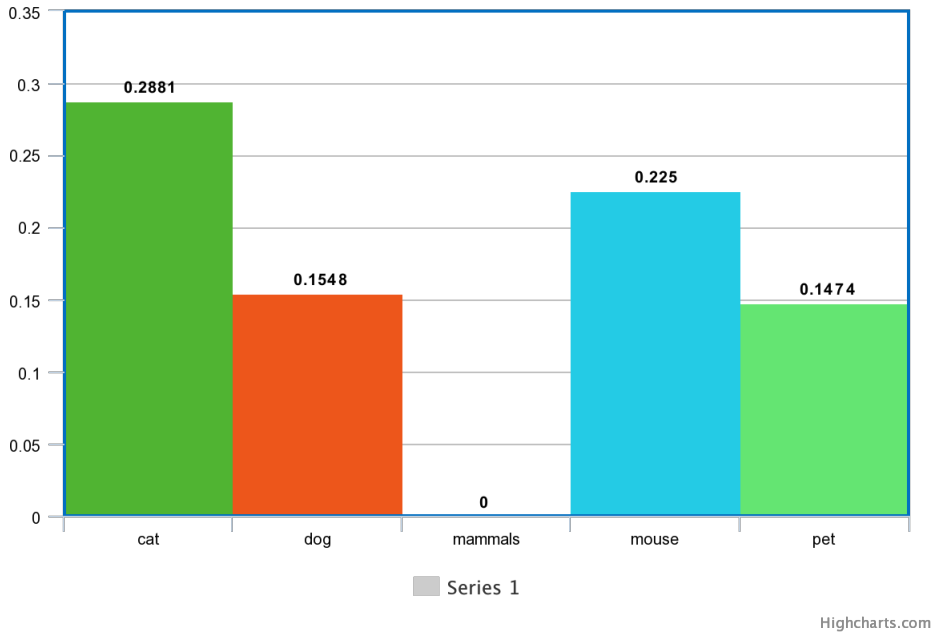

Figure 1: Document 1

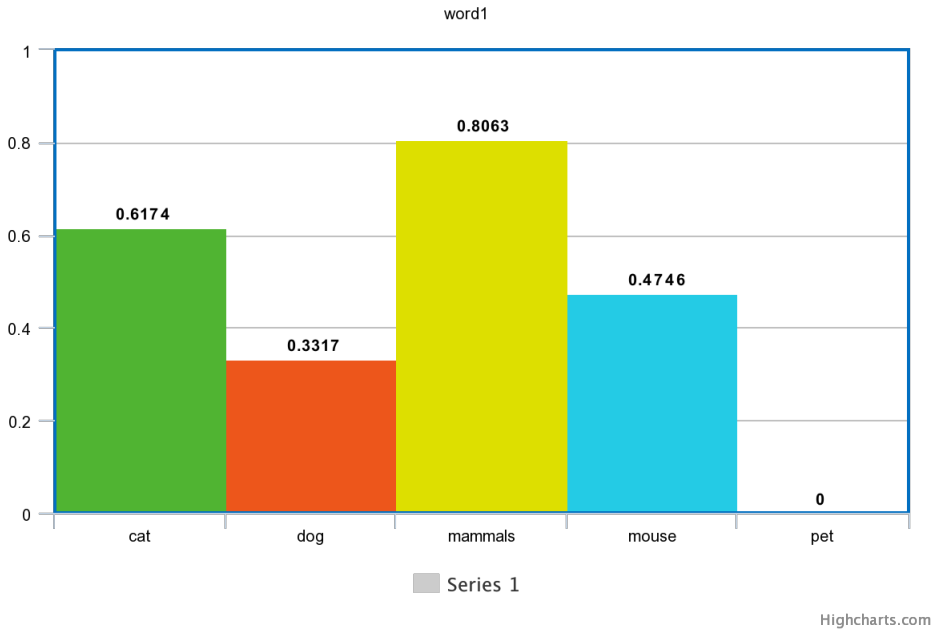

Figure 2: Document 2

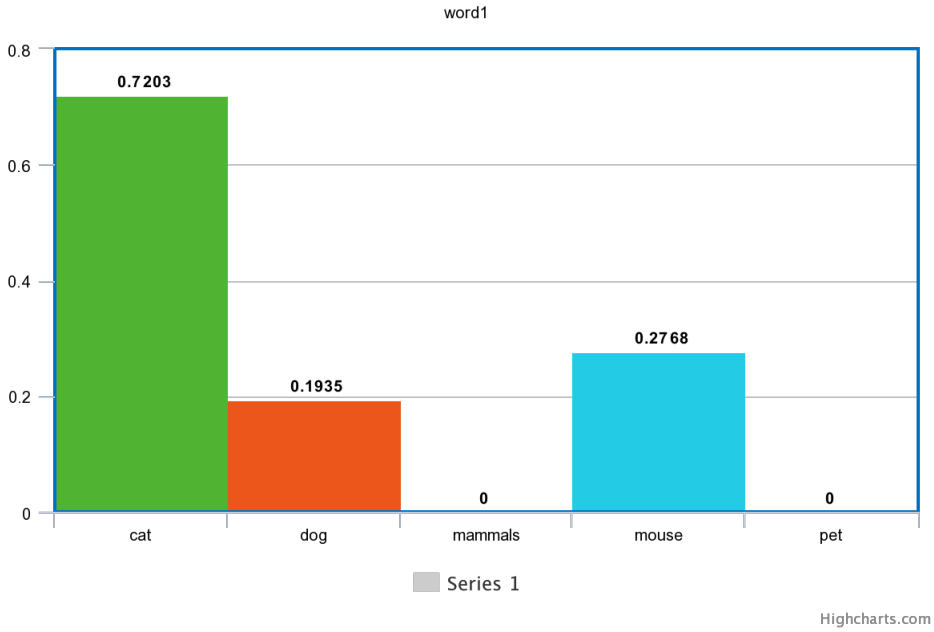

Figure 3: Document 3

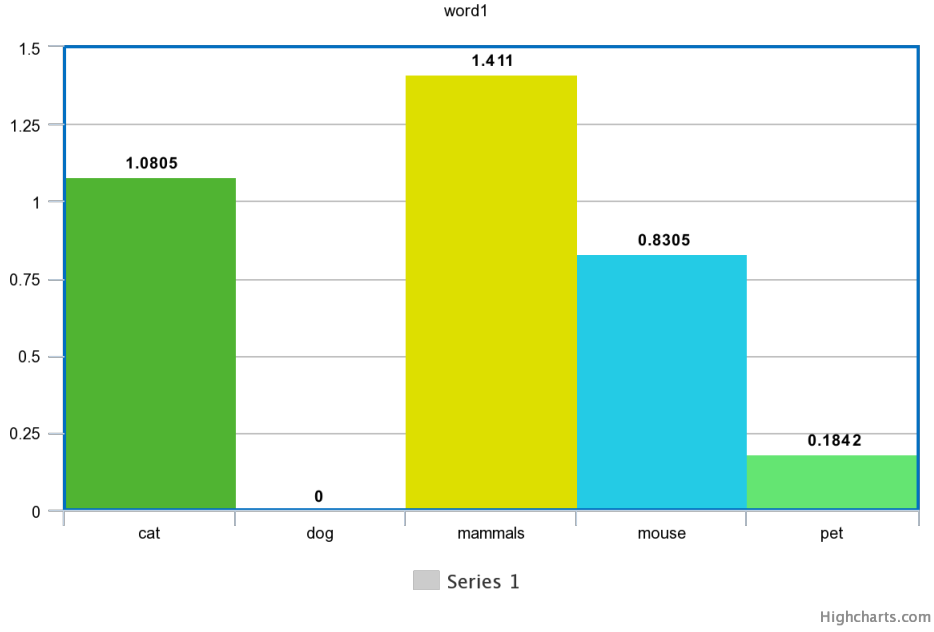

Figure 4: Query