Exercise 1

Machine Learning in Graphics & Vision

Driton Goxhufi xxxxxxx driton.goxhufi@student.uni-tuebingen.de
Damir Ravlija 5503184 damir.ravlija@student.uni-tuebingen.de

1 Task 1

(a) The complexity of the method in subtask (a) is $O(d * n^2)$ where d is the dimension of feature vector and n is the number of examples. Plot of the results in figure 1a).

Code in `problem_1_1_a_b.py`.

(b) A single vector has to be compared with
30 FPS $*120$ s $= 3,600$ frames $= 3,600 * 20,000 = 7.2 * 10^7$(or $3,600N$) vectors
in one video, and each one of these has to be compared with all of the vectors from another video (i.e. $7.2 * 10^7$(or $3,600N$) vectors as well).
There are therefore $128 * 7.2^2 * 10^{14}$(or $128 * (3,600N)^2$ comparisons. Assuming that the machine can compute $3 * 10^9$ comparisons in a second, it would take $221,184,000$ s ≈ 7.0137 years or ($0.55296N^2$ seconds) to find all matchings of the vectors between two different 2 minute long videos (30 FPS) using `exhaustive_search`.

Code in `problem_1_1_a_b.py`. Plot of the results in figure 1b).

(c) Query times in both, exhaustive and KDTree search grow linearly as the number of dimensions increases. However, KDTree is us for datasets with more dimensional vectors up more than 20 times faster. This implies that another variable, the number of vectors in the dataset, is the source of this difference.

Code in `problem_1_1_c.py`. Plot of the results in 1c

2 Task 2

(a) Code in `problem_1_2_a.py`. Top-K accuracy for K between 1 and 10 as expected grows with K and is in the range between 0.8 and 0.97 (Figure 2).

(b) Code in `problem_1_2_b.py`.

Obtained results:

```
Precision (with "Pullover" (2) as positive):  0.8333333333333334
Precision (with "Shirt" (6) as positive):  0.780952380952381
```

```
Recall (with "Pullover" (2) as positive):  0.7653061224489796
Recall (with "Shirt" (6) as positive):  0.845360824742268
```

Driton Goxhufi xxxxxxx
Damir Ravlija 5503184

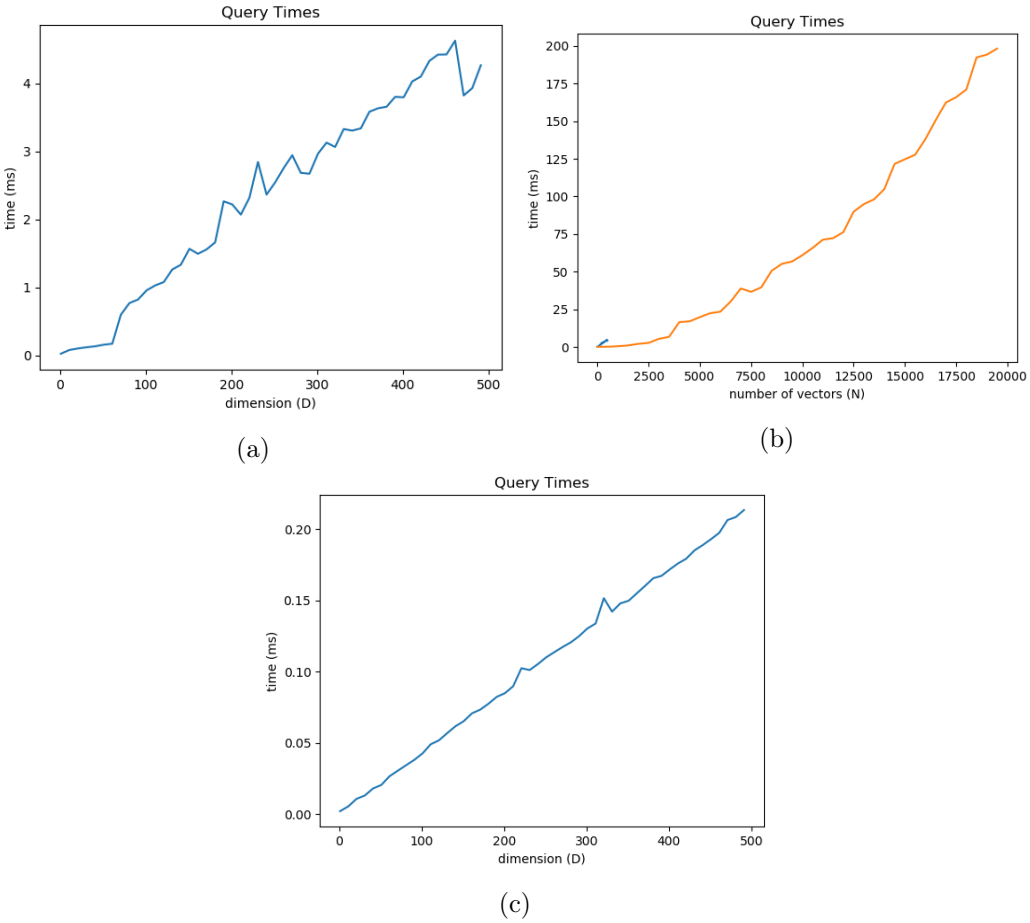

Figure 1: Plot of results from task 1

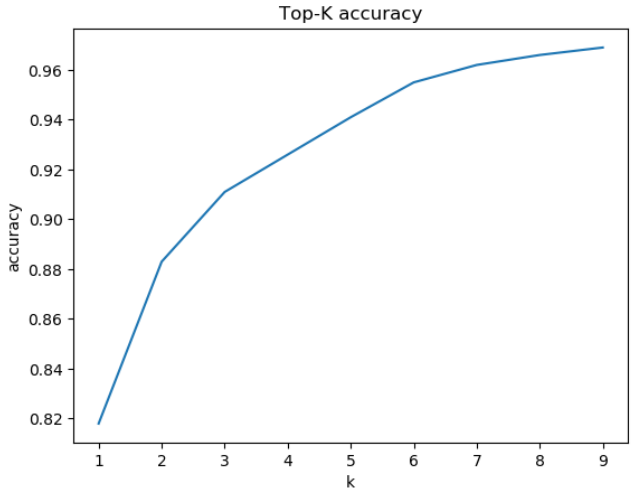

Figure 2: Top-K accuracy for $K = 1, 2, \ldots, 10$ using the KD-tree