

# CS-E4850 Computer Vision

## Exercise Round 7

The following instructions are for the Matlab version. The instructions for the Python version are in Github. Matlab is available on Aalto computers and also for students' own computers via <https://download.aalto.fi>.

The problems should be solved before the exercise session and solutions returned via the MyCourses page. Upload two files: (1) a PDF file containing your written answers to all problems, and (2) a matlab m-file which contains your solution to the problem 1.

**Exercise 1.** Comparing bags-of-words with tf-idf weighting. (Matlab exercise)

Assume that we have an indexed collection of documents containing the five terms of the following table, where the second row indicates the percentage of documents in which each term appears.

term	cat	dog	mammals	mouse	pet
% of documents	5	20	2	10	60

Now, given the query  $Q = \{\text{mouse, cat, pet, mammals}\}$ , compute the similarity between  $Q$  and the following example documents  $D1, D2, D3$ , by using the cosine similarity measure with tf-idf weights (i.e. term frequency - inverse document frequency) for the bag-of-words histogram representations of the documents and the query.

- $D1 = \{\text{Cat is a pet, dog is a pet, and mouse may be a pet too.}\}$
- $D2 = \{\text{Cat, dog and mouse are all mammals.}\}$
- $D3 = \{\text{Cat and dog get along well, but cat may eat a mouse.}\}$

Ignore other words except the five terms, which are listed in the table above. You may proceed with the following steps:

- Compute and report the inverse document frequency (idf) for each of the five terms. Use the logarithm with base 2. (idf is the logarithm term on slide 69 of Lecture 6 where values  $n_i/N$  are given in the table above.)
- Compute the term frequencies for the query and each document.
- Form the tf-idf weighted word occurrence histograms for the query and documents.
- Evaluate the cosine similarity between the query and each document (i.e. normalized scalar product between the weighted occurrence histograms as shown on slide 45 of Lecture 6).
- Report the relative ranking of the documents. (You should get similarities 0.95, 0.64, and 0.63, but you need to determine which corresponds to which document.)

**Exercise 2.** Precision and recall. (pen & paper problem)

There is a database of 10000 images and a user, who is only interested in images which contain a car. It is known that there are 500 such images in the database. An automatic image retrieval system retrieves 300 car images and 50 other images from the database. Determine and report the precision and recall of the retrieval system in this particular case.

(Hint: Precision and recall are explained on slide 67 of Lecture 6 and there is a good explanation also in Wikipedia.)

**Exercise 3.** VGG Object instance recognition practical. (Matlab exercise)

Do the *Object instance recognition practical* provided by the Oxford Visual Geometry Group at <https://www.robots.ox.ac.uk/~vgg/practicals/overview/>.

Read the installation instructions from the above page and do the *fast track* as instructed at <https://www.robots.ox.ac.uk/~vgg/practicals/instance-recognition/>. That is, do the *Tasks* and return short written answers to the corresponding questions. You do not need to program any new functions or return any source codes. The estimated duration of the fast track is 1.5 hours. (Note that the link to Stage III.B on fast track is broken but you can get there by just scrolling down the page.)

The basic steps are as follows:

- Download the code and data, and then run `exercise1.m -exercise3.m` as instructed.

Answering questions in part 1 (`exercise1.m`) corresponds to one bonus point and parts 2 and 3 together correspond to one additional point. Hence, in total this third task is worth of 2 points, whereas the previous tasks (Exercise 1 and Exercise 2) are both worth of 1 point so that in total this homework round is worth of 4 points.