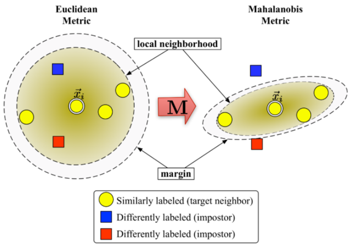
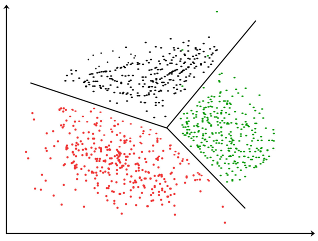
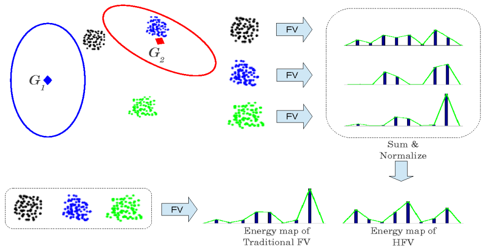
EE4-68 PATTERN RECOGNITION







Released on 12 November 2019.

Submission deadline: 13 December 2019

Submission instructions:

One joint report by each pair Page limit: 4 (four) A4 pages per report. List of references and appendix do not count for this page limit. Use report Latex template from Blackboard.

Use the IEEE standard double column paper format as in coursework 1:

http://www.pamitc.org/cvpr16/files/egpaper_for_review.pdf

http://www.pamitc.org/cvpr16/files/cvpr2016AuthorKit.zip

General principles for writing technical report are expected to be known and adhered to. Similarly for practices in conducting experiments, some are as listed below:

- Select relevant results that support the points you want to make rather than everything that matlab gives.
- The important results should be in the report, not just in the appendix.
- Use clear and tidy presentation style, consistent across the report e.g. figures, tables.
- The experiments should be described such that there is no ambiguity in the settings, protocol and metrics used.
- The main points are made clear, identifying the best and the worst case results or other important observations.
- Do not copy standard formulas from lecture notes, explain algorithms in detail, or copy figures from other sources. References to lecture slides or publications/webpages are enough in such cases, however short explanations of new terms or parameters referred to are needed.
- Find and demonstrate the parameters that lead to optimal performance and validate it by presenting supporting results.
- Include formulas only where appropriate, results presented in figures, and their discussions. Try to visualise any interesting observations to support your answers.

Submit the report in **pdf** through the Blackboard system. Do not submit the code but you may be asked to provide it later in a separate zip file. No paper copy is needed. Write your **full names, logins and CID numbers on the first page. Use both logins in the submitted filename e.g., login1_login2.pdf**. The latest submission before the deadline will be assessed.

If you have questions, please use BlackBoard discussion forum in Pattern Recognition course or contact Mikolaj Jankowski (mikolaj.jankowski17@imperial.ac.uk)

The objective of this coursework is to experiment with various metrics and data representations to observe their properties.

Dataset: Available in blackboard. It includes 520 face images of 52 people. There are 10 images per person thus 52 classes.

Retrieval: Perform k-nearest neighbour (kNN) face retrieval experiments according to standard practices in pattern recognition. Use retrieval accuracy (i.e. @rank1, @rank10) and mean average precision (mAP) as the performance metric to evaluate different methods. Thus each method will have three performance scores.

Report the performance for all experiments using tables or figures, then analyse the results and draw conclusions.

mAP and accuracy @rank-k:

In the coursework you will be asked to create ranklists, which are lists of nearest neighbours for each test image. When creating ranklist remember to exclude test image for which the ranklist is being created from the test set (because distance between two same images is zero, therefore test image will always be the nearest neighbour of itself).

Once you created all the ranklist for the test set, you can calculate **accuracy @rank-k**. To do so, for each test image consider k nearest neighbours and assign a number:

$$Acc_k = \begin{cases} 1, \text{if there is at least one image of the same person within } k \text{ nearest neighbours,} \\ 0 \text{ otherwise.} \end{cases}$$

Then average the sum of all Acc_k across the whole test set.

Once calculated accuracy @rank-k, follow the steps below to calculate the **mAP**:

1. For each ranklist and each k calculate corresponding precission @k given by:
$$P_k = \frac{TP_k}{k},$$
where TP_k is number of correctly retrieved images (images of the same person) up to k-th nearest neighbour.

2. Similarly, for each ranklist and each k calculate coresponding recall @k given by:
$$R_k = \frac{TP_k}{total\ number\ of\ matching\ faces},$$

where $total\ number\ of\ matching\ faces$ in considered dataset corresponds to 9 (10 total images minus 1 we consider a query and remove from the test set).

3. Consider following 11 recall levels:

$$r \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\},$$

and calculate interpolated precision $p_{interp}(r)$ as:

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}),$$

where $\tilde{r}$ belongs to final list of recalls obtained in point 2. and $p(\tilde{r})$ is precision (calculated in point 1.) corresponding to this recall.

4. The AP can be calculated as follows:

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \ldots 1.0\}} p_{interp}(r).$$

5. To calculate mAP reapeat the procedure for each image in the test set and calculate the mean of all APs.

More detailed explanation can be found in
`https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173`
and [2].

Follow the instructions below.

Q1. [25 marks] **Distance Metrics**

A. **Data.** Prepare 520 of 2576-dimensional vectors by:

   a) Using original unmodified images as feature vectors.
   b) Normalizing feature vectors to unit norm L2.

   Partition the data into training and testing set as follows:

   - First 320 images (classes 1-32) as the *training* split.
   - Remaining 200 images (classes 33-52) as the *test* split.

   Note: Report all results by providing performance scores @rank1, @rank10 and mAP for the *test* split.

B. [3 marks] **Baseline.** Perform retrieval with kNN on features from A.a and A.b, using standard non-learned distance metrics discussed in the lecture.

C. [4 marks] **Experiment 1.** Implement histogram of pixel intensities as the feature representations (consider different ways of choosing bins or quantities different than pixel intensities) and use standard non-learned distance metrics discussed in the lecture to evaluate the performance.

D. [5 marks] **Experiment 2.** Implement Mahalanobis distance metric by calculating covariance matrix on the *training* split of features in A and report the performance by applying it to the *test* split. Use it to reduce the dimensionality to $\{16, 32, 64, 128, 256\}$. More details can be found in [3]. In our experiments we did non observe that but use pseudo-inverse in case the matrix decomposition throws an exception.

E. [6 marks] **Experiment 3.** Perform dimensionality reduction with PCA and LDA on feature vectors obtained in point A and C. Report your scores.

F. [7 marks] **Experiment 4.** Perform Mahalanobis metric learning:

   a) Report results for at least one of the following distance metric learning techniques: RCA (Relevant Component Analysis), NCA (Neighborhood Components Analysis), LMNN (Large Margin Nearest Neighbors) and compare to D.

   b) Optional. Consider using different distance metric learning method, a good overview is provided in [1].

Q2. [25 marks] **Cluster based representations**

A. **Clustering.**

   a) Implement k-means clustering.

   b) Use or implement Agglomerative clustering.

B. [10 marks] **Experiment 5.** Perform clustering on the *training* data. Propose and implement a method to assign label to the cluster (e.g. Hungarian algorithm). Report the labelling performance (@rank1 accuracy) for clusters from A.a and A.b. Based on the results choose A.a or A.b for the experiments in C.

C. **Fisher vectors.** Obtain new representations of images:

   a) Use cluster centres as a codebook and then vectors of distances to the cluster centres as feature vectors.

   b) Consider using softmax probabilities of inverse distances to cluster centres as your vector representation.

   c) Obtain Fisher vectors from GMM as described in lecture notes. Note that you can obtain GMM representation based on clusters obtained in B.

   Note: Use the *training* split to obtain the codebook and GMM and calculate new representations for the *test* split. Consider different numbers of gaussians/clusters.

D. [15 marks] **Experiment 6.** Report results for representations from C.a, C.b and C.c.

References

[1] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.

[2] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Jun 2010.

[3] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 521–528. MIT Press, 2003.