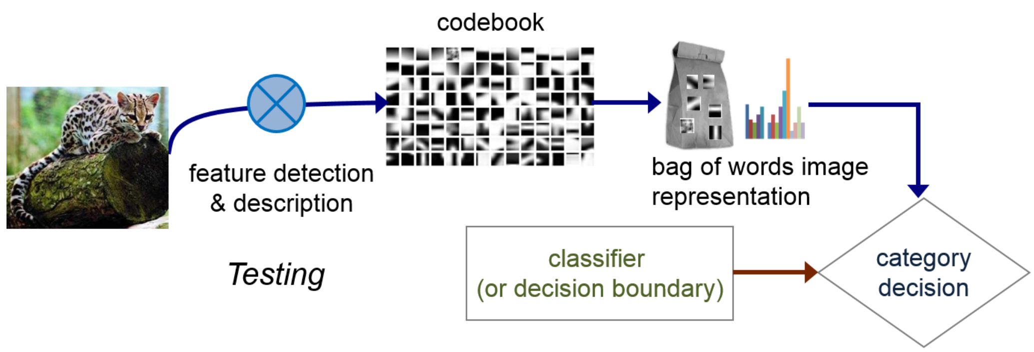
EE462, EE9SO25, EE9CS728

Machine Learning for Computer Vision

Coursework on visual categorisation by BoW (K-means) and SVM [30% mark]



Release on 8 Feb, the report due on 14 Mar (midnight)

The course work requires Matlab programming. Use the provided data for **Q2** and **Q3**, please see the questions. In all questions, you can use any existing toolbox/code for K-means and two-class (or binary-class) SVMs. Some suggestions are below.

Note: Your own lines of code for BoW and multi-class SVM for **Q3** should be provided in an appendix, which does not count for the page limit. Other source codes are not mandatory.

Submission instructions:

One joint report by each pair

Page limit: 4-6 A4 pages per report with 10 font size (use the IEEE standard double column paper format, either in MS word or latex).

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

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

Give insights, discussions, and reasons behind your answers, on the scope of lectures. Quality and completeness of discussions within the page limit will be marked.

Submit the report **in pdf** through the Blackboard system. No hard copy is needed. Write your full names and CID numbers on the first page.

If you have questions, please contact

Mr. Guillermo Garcia-Hernando (g.garcia-hernando@imperial.ac.uk)

Dr. Rigas Kouskouridas (r.kouskouridas@imperial.ac.uk)

Useful toolboxes:

You may use the statistical pattern recognition toolbox (STPR) or LibSVM, which is downloadable at

http://cmp.felk.cvut.cz/cmp/software/stprtool/

https://www.csie.ntu.edu.tw/~cjlin/libsvm/

If you encounter any trouble in using the STPR toolbox in college machines, please contact g.garcia-hernando@imperial.ac.uk.

Libsvm-3.18 is given as an external library in the provided codes (under the directory of 'external').

Q1. [25] Image quantisation (pixel clustering) by K-means

Take a color (RGB) image of your choice, apply the K-means (you can use any existing codes for K-means) to the image pixel data, and perform the image quantisation. Show and discuss the results with different K values (e.g. 3,10,20).

For a fixed K value and initialisations of cluster centers, show and discuss how many iterations are needed till a convergence, how the initial and updated cluster centres move every few iterations in a 3D plot, and how the objective measurement of K-means changes in the iterations.

For a fixed K value, try different (random) initialisations, and discuss the results and explain how to choose the best cluster result.

Q2. [30] Multi-class (M=3) SVM for the Toy Spiral data

Use the provided data. In order to access the data, start Matlab, go to the directory where the provided files are. Open the main_guideline.m and do the followings:

Do initialisation.

```
>> init;
```

We provide three toy data sets, each of which is a set of 2D points with their class labels, and the Caltech101 image categorisation data set. Load the Toy Spiral data set here.

```
>> [data_train, data_test] = getData('Toy_Spiral'); % {'Toy_Gaussian',
'Toy_Spiral', 'Toy_Circle', 'Caltech'}
```

The training and testing data are in the format of

```
data_train(:,1:2) : [num_data x dim] Training 2D vectors,
data_train(:,3) : [num_data x 1] Label of training data, {1,2,3}.

data_test(:,1:2) : [num_data x dim] Testing 2D vectors, 2D points in
the uniform dense grid within the range of [-1.5, 1.5],
data_test(:,3) : N/A.
```

Plot the training and testing data.

```
>> plot_toydata(data_train);
>> scatter(data_test(:,1),data_test(:,2),'.b');
```

Train and test multi-class SVM using the Toy Spiral data. You can use any existing toolbox for two-class (or binary-class) SVMs. Note, write your own lines of code for the multi-class extensions of SVM (*extensibility of the codes to more than 3 classes will be needed for Q3). Show, measure and discuss the results, including:

- extension of two-class SVMs to the multi-class problem (both one-versus-the-rest and one-versus-one),
- setting the SVM parameters, i.e. kernel type, kernel parameters, C (underfitting/overfitting),
- visualisation of the classification results of the training and testing data, using each of the two-class SVMs, and their combined in the multi-class extension,
- support vectors, margin, etc.,
- complexity.

Q3. [45] Bag-of-words, multi-class SVM for the Caltech101 dataset

Use the provided Caltech101 dataset for image categorisation. We use 10 classes, 15 images per class, randomly selected, for training, and 15 other images per class, for testing. Feature descriptors **d** are given. They are multi-scaled dense SIFT features, and their dimension is 128 (for details of the descriptor, see http://www.vlfeat.org/matlab/vl_phow.html). We randomly select 100k descriptors for K-means clustering for building the visual vocabulary (due to memory issue).

Open the main_guideline.m and select/load the dataset.

```
>> [data_train, data_test] = getData('Caltech'); % Select dataset
```

Set 'showImg = 0' in getData.m if you want to stop displaying training and testing images. Complete getData.m by writing your own lines of code to obtain the visual vocabulary and the bag-of-words histograms for both training and testing data. Provide you own lines of code in an appendix of your report. Correctness and efficiency of the code will be marked. You can use any existing codes for K-means (note different codes require different memory and computation time). Show, measure and discuss the followings:

- vocabulary size,
- bag-of-words histograms of example training/testing images,
- vector quantisation process

Train and test multi-class SVM using the bag-of-words histogram vectors **x** obtained above. Use your codes in **Q2** (they need to be extended to M>3 classes). Provide you own lines of code in an appendix of your report. Show, measure and discuss, including:

- one-versus-one and one-versus-rest
- kernel types and parameter values (including C)
- recognition accuracy and confusion matrix
- mean ranks
- time-efficiency of SVM training/testing
- impact of the vocabulary size on classification accuracy
- examples of support vectors and success/fail images

Give insights and reasons behind your answers.