School of Computing Sciences

Module: CMP-6035B / CMP-7026B Computer Vision
Assignment: Image categorisation I.

Set by: Michal Mackiewicz M.Mackiewicz@uea.ac.uk
Date set: 25th February 2022
Value: 50%

Date due: 23rd March 2022 3pm week 8
Returned by: 27th April 2022
Submission: Blackboard

Learning outcomes

To reinforce material presented in the lectures and to give you practical experience programming computer vision algorithms in Matlab. An in-depth understanding of the related computer vision algorithms. An understanding of the algorithm evaluation process. Improved Matlab programming skills with a focus on creating efficient solutions.

Specification

Overview

Image classification is one of the major research areas in computer vision. The main task in coursework 1 will be to perform a literature review of the image classification/scene categorisation research and implement simple benchmark image classification algorithms. You will continue your work in this area in coursework 2 where we will ask you to use what you will have learned during your literature review in coursework 1 to design more advanced algorithms.

You will work on this coursework in pairs.

You should submit the names of students in your pair and their email addresses by **Tue, 1st March, 5pm** by filling in the online form at this address `https://tinyurl.com/t2fdmm3r`. Anyone who misses this deadline will be paired by me and informed about their partner on Tue 2nd Mar. You can pair only with a student with the same module code i.e. CMP-6035B students cannot pair with CMP-7026B students.

Description

The dataset you will work with was created as part of the SUN database [1]. The dataset comprises 3000 images split into 1500 training and 1500 test images. These images belong to the following scene categories: *Kitchen*, *Store*, *Bedroom*, *LivingRoom*, *House*, *Industrial*, *Stadium*, *Underwater*, *TallBuilding*, *Street*, *Highway*, *Field*, *Coast*, *Mountain*, *Forest*. The idea of image classification is that given the image of an unknown category you wish to extract features from the image (e.g. the colour histogram) and then compare the extracted features with those found in labelled (i.e. classified) images in the database.

[1] See `http://vision.princeton.edu/projects/2010/SUN/`

We will provide you with some basic code, which you will need to edit to complete this coursework exercise. The Matlab file `coursework_starter.m` file contains a simple algorithm that you should use as the skeleton solution for both coursework 1 and 2.

There are many ways one could approach image classification problem. Or, put another way, there are many features we might extract and then use as an aid to classifying an image (to one of Kitchen, Store, etc). We might, for example, test the hypothesis that colour is an important cue in describing the scenes in images. In coursework 1, you will implement a benchmark algorithm that is based on this idea.

In particular you will use the 'Color Indexing' method method described by Swain and Ballard (S&B) [1] which is available on the Portal. In broad sweep, the S&B approach summarizes image content by the image's colour histogram. If an image has a colour histogram that is similar to images in a specific category then one might conclude that the image is also a scene in that category. However, like all algorithms the S&B method has many algorithm parameters and these can be varied to try and arrive at the best overall classification performance. So, you will need to perform a series of experiments tweaking various parameters of the S&B method such as: the choosing the best colour space (RGB or something else?), quantisation (the number of bins in the histogram) as well as test the settings of a simple k-nearest neighbour classifier.

Your coursework submission should include an in depth presentation of experimental results. You should complete lab 5 (part 2) before you begin implementing feature extractor and lab 4 before you begin implementing kNN classifier.

In addition to colour histograms, you should also investigate classification performance using the 'tiny image' feature. The idea here is that an image can be down sampled to a 16×16 thumbnail. And then, the 256 colors in this image can be stretched out as long vector (which is 256x3 = 768 numbers long). This 'feature vector' can be compared against other tiny image feature vectors calculated for the database (again using the k nearest neighbour algorithm). As with the colour histogram approach, there are certain parameters you can tune to investigate classification performance. For example each vector could be normalized so it has 0 mean and unit length. Another option is to try a grayscale 'tiny image' instead of RGB.

The expectation is that you will find that the tiny image and colour histogram representations do not deliver great classification results. You need to discuss why you think the implemented methods work (or don't work) in detail in your report.

The `coursework_starter.m` is divided into step 1 and step 2. In step 1, you will need to implement two aforementioned feature extraction methods. In step 2, you will need to implement a k-nearest neighbour classifier. The `coursework_starter.m` script can be run as is since the functions *get_tiny_images* and *nearest_neighbour_classify* have been implemented and compiled into the *pcode*(see MATLAB help). P-code can be executed, but it does not give you access to the source code. You will need to re-implement these functions yourself as well as implement the missing *get_colour_histograms* function. The reason why we provided you some functions as the P-code is to allow you to run the code for one set of input parameters immediately and have a feel of the classification results. Notice, that the above starter code creates the visualisation of the classification results including a confusion matrix as well as an html page containing image thumbnails of correct classifications, false positives and false negatives for each category.

To complete the assignment you should:

1. Re-implement *get_tiny_images* and *nearest_neighbour_classify* functions. Test the algorithm for various parameters of both functions and discuss the results.

2. Write Matlab code that implements the S&B algorithm [1] and place it in *get_colour_histograms* function. Test the algorithm for various parameters and discuss the results.

3. Perform a literature review into image classification in general and scene recognition in

particular with a view of improving on the two simple benchmark algorithms in coursework 2. An excellent starting point would be the following reference [2] which is available on the Portal. Hint: you will have access to a larger selection of journal and conference articles if you are logged on at the university.

Installation of the dataset and the starter code.

The *code.zip* file contains the starter code that you will use as a skeleton for your coursework 1 and 2 submissions. The *data.zip* contains the dataset of 3000 images, split into the training and test sets of the same size. Both training and test sets contain 100 images for each of 15 image categories.

Although not necessary for coursework 1, you may want to install a VLFeat MATLAB toolbox [2]. Download the toolbox and run `vl_setup`. The toolbox contains a large selection of computer vision related functions. This toolbox will be really required for coursework 2, but you may find some useful functions for coursework 1 as well. You should not bundle this toolbox with your submission. You can assume that on my machine the toolbox file path is set in MATLAB.

Relationship to formative assessment

You will receive formative feedback during all subsequent Fri lab sessions. You should complete lab 4 and lab 5 part 2 before attempting this coursework.

Deliverables

Marks will be awarded for the electronically submitted code and the report. This is a group coursework that should be done in pairs. You should submit the names of students in your pair and their email addresses by **Tue, 1st March, 5pm** by filling in the online form at this address `https://tinyurl.com/t2fdmm3r`. Anyone who misses this deadline will be paired by me and informed about their partner on Wed 2nd Mar.

Your written report should be no longer than 10 pages in length and can be formatted as a double column text. Your report should **not** include any Matlab code (algorithms can be described as mathematical equations and/or pseudo code). All pages should have reasonable margins and a font size of 11pt should be used. A part of a report that is over the 10 allowed pages will not be marked. Make sure you include any relevant citations in your report. For reference you are pointed to the university plagiarism regulations. Make sure that both parts of your report contain informative clearly labelled figures describing your results and the literature review algorithms.

1. Report and code hand-in electronically, **Wed 3pm, Week 8** through Blackboard. Both members of the pair must submit the same report and code. You must submit two files - a PDF and a ZIP. The Matlab source code must be zipped into a file with the name containing Student IDs of both members of the group. The report should have the same file name and the PDF extension. **Make sure you test your solutions on the lab machine BEFORE you submit**.

2. The functions implementing your algorithm have to be called according to the specified naming scheme and placed in a folder named with your pair registration numbers. Any additional functions must be placed in the same folder.

3. You should submit only one version of the 'get_colour_histogram' and 'get_tiny_images' methods i.e. for a certain fixed set of parameters - I would imagine the best performing

[2] `http://www.vlfeat.org/install-matlab.html`

version. You should state in your report what is the final set of parameters that were submitted with your code.

4. After copying your coursework folder into my machine, I should be able to run your code on my machine as is. The only edit to the code you may expect me to make is changing the `data_path` variable in `coursework_starter.m`. Therefore, you must not include the image dataset in your submission.

Late submissions need to follow the appropriate late hand-in procedure. You can find out this information from the Hub.

If you have medical or other problems you can seek extensions to coursework deadlines. However, it is essential you obtain proper documentation in such cases (i.e. a medical certificate).

If one of the pair members is granted an extension, the other member of the pair also receives an extension.

Both members of the pair are equally responsible for the joint work. In case of any breakdown of co-operation, both students have to complete their work individually and notify me by email that this is what they are doing.

Resources

You should do the previous labsheets which are available on BB. Lab 4 and lab 5 part 2 will contain exercises which will be particularly relevant from the point of view of this coursework. You should also consult the lecture notes, week 4 in particular.

You should attend all the lab sessions and discuss your progress with the teaching staff who will be able to provide you formative feedback.

Marking Scheme

This coursework carries 50% of the module weight, therefore you may receive up to 50 marks for this coursework.

The marks will be awarded according to the following scheme:

- Implementation, testing and reporting of the algorithms:
    1. Description of the experiments and results. (**16 marks**).
    2. Conclusions drawn from the results. (**4 marks**).
    3. Code quality. (**4 marks**).
- Literature review into image classification and scene recognition.
    1. Literature review into feature detection. (**13 marks**).
    2. Literature review into classifiers. (**13 marks**).

You should add a brief commentary describing the individual contributions of both members of the pair at the end of your report.

Hints:

Matlab exercises and reporting results.

You should try to analyse several parameters of colour histogram features in a similar way as Swain and Ballard have done it e.g. colour spaces, distance metrics etc. When investigating the latter you may come across distance metrics such as: 'city-block', 'Euclidean', 'L1', 'L2',

'Minkovski' and not technically a distance - 'histogram intersection'. What is the default parameter for the `pdist` and `pdist2` function in Matlab? You should try to understand the relation between the above distance measures. Read the relevant part of Swain and Ballard paper.

Try to depict your data using plots and figures rather than giving them in tables. Figures are more readable and can save you some space as well. Do not repeat the same data in figures and tables.

Remember to discuss the confusion matrix/matrices and the individual category results.

Literature review

Textbooks are good starting points for any literature review as they give you a broad view of a particular research area.

I am looking for clear descriptions with figures. If you give equations, always remember to explain what each variable means. This said, having too many equations in a wide literature review and a limited space is probably not the best idea.

You should produce bibliography that looks professional. No matter which referencing style you use, journal and conference references should at least contain authors, title, conference/journal name, year, volume and pages. Avoid web references e.g. wikipedia. Most or even better all references should be academic i.e. journals, conferences and books. You must use LaTeXfor your report. You will loose 5 marks if your submission has not been produced using LaTeX.

References

[1] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[2] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.68