EEE3032 – Computer Vision and Pattern Recognition Coursework Assignment

Visual Search of an Image Collection

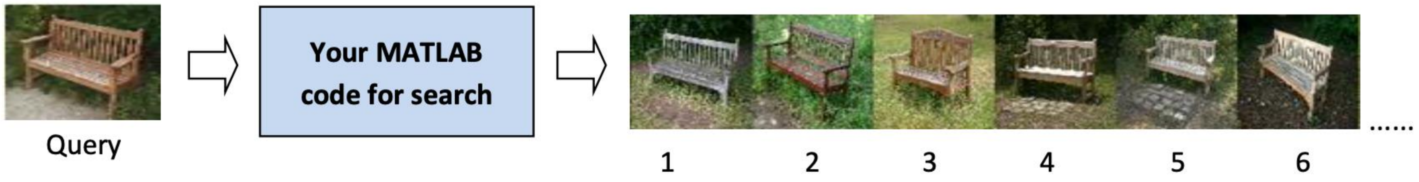
Prof Miroslaw Bober

Autumn 2024 (R1)

1. Introduction

Digital image collections are traditionally searched used textual queries (e.g. Google). Text is ideal for describing the objects present in an image, but is poor at describing the *visual appearance* of an image. Yet it is often desirable to search based on visual appearance, e.g. when shopping online.

Your assignment is to write a program to *visually search* an image collection. Your program will accept an image as a search query, and return a list of images in the collection ranked according to their similarity. You may choose to display only the top 20 images. You must demonstrate your system working using several different query images.



You must use the Microsoft Research (MSVC-v2) dataset of 591 images (20 classes) which can be downloaded from SurreyLearn or direct from MSR here: http://download.microsoft.com/download/3/3/9/339D8A24-47D7-412F-A1E8-1A415BC48A15/msrc_objcategimagedatabase_v2.zip

Please note the above download is 110 Mb, zipped. You may use Python instead of MATLAB

2. Deliverables

You must write software in MATLAB* to perform this task. The requirements specification of this software is given later; the more challenging the requirements implemented (successfully), the higher the mark you will receive.

You must document your software describing the techniques you have chosen and how you implemented them (including how you overcame any difficulties). **Most importantly, your document must include evidence of testing.**

50% of marks will be awarded for software (i.e. your code), and 50% for your documentation and testing. A more detailed marking scheme is included in Section 6. This coursework is worth 20% of your overall mark for EEE3032.

This year we have moved to **electronic submission of the coursework on SurreyLearn**. It is important that you **submit both your written documentation and your code**. You should use a zip file to package up your code and include a short text file called "README.TXT" that explains in a paragraph or two how to run it.

The deadline for this coursework is 4PM on Tuesday in Week 8

*Note that this year you can submit your assigment in Python or MATLAB

3. Resources

You are supplied with two skeleton MATLAB programs (on SurreyLearn) enabling you to start coding quickly.

1) **cvpr_computedescriptors**

This program will iterate through every image in the dataset and generate an image descriptor from it by calling the supplied function "`extractRandom`". As supplied, that function generates a set of random numbers instead of an image descriptor. You should replace "`extractRandom`" with your own code for extracting your chosen type of image descriptor. The descriptors are all saved into a separate folder. You can configure folder names in the code.

2) **cvpr_visualsearch**

This program will load in all the image descriptors computed by "`cvpr_computedescriptors`". It will then pick an image at random to use as a query. It will repeatedly call "`cvpr_compare`" with the query descriptor and the descriptor from each image. Currently this function returns a random number. You should edit it to return the distance between the features being compared. The program "`cvpr_visualsearch`" will visualise your top 10 results in a basic way. Depending on the level of sophistication of your coursework solution, you might improve this.

The purpose of the skeleton code is to get you started quickly. You should be able to create a working basic visual search system by replacing "`extractRandom`" with a global colour histogram (slide 4, lecture 7) and "`cvpr_compare`" to output Euclidean distance between the two features it is passed. This will meet Requirement 1 (see section 6. You might then go on to enhance the "`cvpr_visualsearch`" code to produce **precision-recall** statistics, and then try other descriptors or distance metrics to improve performance (you can measure the improvement using your aforementioned **precision-recall** code).

4. Learning Outcomes

After completing this coursework, and attending the relevant lectures, you should be able to:

1. **Select** and **apply** appropriate computer vision techniques to extract descriptors from images.
2. **Select** and **apply** appropriate pattern recognition techniques to match images using their descriptors.
3. **Evaluate** a visual search system using standard information retrieval metrics.
4. **Conduct** a scientific investigation into the relative performance of various approaches to visual search.

Please note that MATLAB is a vehicle for trying out Computer Vision ideas – we will mark you on your achievements and experimentation in visual search. We will not be assessing neatness of code or programming style (but your code must be legible for marking).

5. Plagiarism

You must complete this coursework individually. If you rip code or text from the web, or another student, and include it in your project without clear attribution, then you have committed plagiarism. Undetected plagiarism degrades the quality of your degree, as it interferes with our ability to assess you and prevents you learning through properly attempting the coursework. Consequently if we suspect plagiarism you will referred to an Academic Misconduct Panel, which for a first offence, may result in you receiving zero marks for the coursework and your mark for the entire module being capped at the pass mark (40%).

Please try not copy figures and text from lecture notes or internet sources for your documentation, even if you cite its source. Whilst not plagiarism this is still poor practice – if you wish to explain a concept please draw your own figures and use your own words. Cite any material you might have found on the internet to back up your statements. This use of citation *will* gain you credit. Re-using some one else's material will not be worth credit.

6. Requirements Specification and Marking Scheme

You are provided with basic code for iterating through files in the Microsoft dataset. You must enhance this basic code to implement as many of the following functional requirements as possible.

Implement feature 1, then feature 2, and then choose any other features 3-5,8 to implement (in any order)

We do not expect you to implement all features in the time available. In particular features 6-7 require major additional work beyond expectations of this coursework; only 1 or 2 students each year successfully attempt this.

A mark budget is given for each requirement – for example, if you complete 1 perfectly (i.e. bug free, and your documentation is flawless) then you gain 30 marks (15 for the implementation, 15 for documentation). Completing a few features well is much preferred over attempting many features poorly.

You cannot earn more than 100% as the final mark for this coursework!

Req No.	**Requirement**	**Mark Budget up to…**
1	**Global colour histogram** Implement the global colour histogram from lecture 7 (slides 4,5) using a Euclidean distance metric. Experiment with different levels of RGB quantization.	30%
2	**Evaluation methodology (applies to results of all experiments)** Compute precision recall (PR) statistics for each of your experiments, e.g. PR for the top 10 results. Plot the PR curve. If you are defining similarity in terms of object categories, compute a confusion matrix. Discuss and analyse your results e.g. which experiments were most successful (which images worked well; why do you think this is given your descriptor choice).	25%
3	**Spatial Grid (Colour and Texture)** Implement gridding of the image, and concatenation of features from grid cells to form the image descriptor. Experiment with colour and/or texture features. In the latter case, experiment with different levels of angular quantization.	15%
4	**Use of PCA** Use PCA to project your image descriptors into a lower dimensional space and explore the use of Mahalanobis distance as an alternative distance metric. Does performance improve?	15%
5	**Different descriptors and distance measures** Experiment with different choices of distance measure e.g. L1 norm and note the effect they have on performance. Can you discover other distance measures or descriptors not covered on this module and try these out?	15%
6	**Bag of Visual Words retrieval (hard)** Implement a basic BoVW system using a sparse feature detector (e.g. Harris or the SIFT keypoint detector) and a descriptor (e.g. SIFT descriptor). Use k-Means to create the codebook. Compare the performance with other descriptors you have tried.	40%
7	**Object classification using SVM (hard)** Apply an SVM to try to decide the category of the image e.g. “bike” or “sheep” based on the descriptors you have extracted from each image. (*This is classification not strictly visual search)*	30%
8	**Extra Credit** If you have your own idea (entirely new or a twist on the above themes) then we will award up to a further 20% based on technical merit. Remember the innovation should be based on Computer Vision (which we are assessing) and not fancy coding (we are not assessing) e.g. a UI.	20%

7. Document Structure

Your written report must be written in 11pt Arial (or similar) with 2cm margins. It need not exceed 10 pages excluding title page, references, and the abstract/contents page. This is not a page 'limit' it is a guide to prevent you over-investing time beyond the requirements of this assignment. You may adopt any document structure you wish, however we recommend the following:

- Title page (1 page)
- Abstract and Table of contents (1 page)
- Description of Visual search techniques implemented (3 pages)
- Experimental results (5 pages)
- Conclusions drawn from experiments (1 page)
- Bibliography (1 page)

If you have additional test results to show, appendices may be added.

Do not forget to submit your **code as a zip ('compressed archive')** and your **report as PDF**. The latter is required for submission to JISC plagiarism detection software which under Departmental guidelines we are required to do.

8. Feedback

In line with University regulations you will receive feedback within 3 teaching weeks of the deadline for this coursework. This will comprise an overall mark and some written feedback against each of the criteria in section 6.

9. Extensions

No extensions will be granted to this coursework unless in exceptional circumstances. Even then, extension requests must be made to the Director of Undergraduate Studies (Dr Carey) via the Extenuating Circumstances (ECs) process as your lecturer is not permitted to grant adhoc extension requests whatever the reason. The MATLAB license server crashing just prior to submission does not count as an exceptional circumstance - please plan your time!