CSC1121
2024-2025

Assignment 2

Image Search Engine

Mark **50**% of the module mark

Groups You will be working independently

Submission deadline **Friday 12 April 2025 – 23:59**
If you do not adhere to the prescribed submission guidelines below, the lecturer assumes no responsibility if your assignment is overlooked and consequently is not corrected.

How to submit? Upload your assignment on Loop in PDF format only and as one single file. Be sure to clearly identify your name on the title page of your assignment. There will be a follow-on interview scheduled during which you will be asked to demonstrate your information retrieval system operating.

The name of the PDF file should **respect the following naming template**:

SurnameOfStudent_NumberOfStudent1 _CSC1121_Assignment2.pdf
Example:
Gurrin_1768573_CSC1121_Assignment2.pdf

Objectives You will use the search engine developed in your previous assignment to implement a prototype image search engine and report on the interface and quality of the search result.

Description Assignment 2 is designed to follow-on from assignment 1. In assignment 2, you will implement a basic image search engine with a basic goal of replicating Google Image search as it was working when it was first released (80% of the marks). A stretch goal is to enhance the quality of the search engine using the output of an open-source computer vision toolkit (20%).

The main steps are:

1. To gather a collection of 1,000 (at least) images from web pages using a web crawler, which can be of your own development (e.g. python or java), or using an open source tool.
2. Prepare a textual surrogate of each image using available sources from the HTML content (at least) and (if possible as a stretch goal) include richer annotations from computer vision-based tools.
3. Index the image surrogates using the search engine that you have previously developed for Assignment 1[1].
4. Provide a web-based interface to the image search engine and submit the url

[1] In the event that your search engine in Assignment 1 was not competed or working properly, it is ok to use an open-source search tool (e.g. Lucene, MG4J, Solr, ElasticSearch, Lemur, Terrier, etc) for this assignment.

with a list of sample queries, along with a short paper describing how your search engine works, in the submitted report on loop. You may use any appropriate HTTP server for this.

NOTE: Indexing and retrieval are separate processes and do not need to be synchronised. The project can be seen as two distinct parts: 1, Gather the images and annotation, and then, 2. Provide retrieval facilities. A python or java web crawler is not as challenging to write as you may think, and can easily be developed to download and parse HTML pages. This is much easier to achieve if developed for a restricted domain (e.g. wikipedia).

Data

The data or the assignment is up to you to gather. At least 1,000 images should be gathered for this assignment. The data should be gathered and held until the assignment is competed and assessed. The data should not be published in any way to avoid any potential copyright issues. There is no restriction on where to crawl the images, but restricting your crawl to a domain such as wikipedia or other publicly-accessible image archive is suggested.

Programming Language & Code

You can use any language you want, but it is suggested that you use either Python or Java for the crawler and indexer for this project. The choice of web interface is up to you. For this assignment, you do not need to provide the code, just the url of where to access the search engine[2].

Report

Along with the URL to your search engine[2], you should submit a report documenting your activity. As with assignment 1, the report should be written in ACM sigconf template (latex or word) using the suggested templates.

The report length should be between 4 and 8 pages (exceeding pages will not be considered and the corresponding grades will be lost). The report should contain a URL of the image search engine (see footnote 2).

The report documents your project. You should provide a general architecture of your system and provide motivations behind specific choices that you had to take when implementing the different components of your system. When these choices were supported by scientific publications, you should provide references and include them in the Bibliography section.

Your report should contain at least the following sections:

- **Abstract**. A quick overview of the content of the report.
- **Introduction**. An introduction to the problem and an overview of the architecture of your system, including the crawler developed/deployed and the data source used.
- **Annotation**. A section outlining how each image is annotated.
- **Indexing**. A section describing the process of indexing the collection.
- **Retrieval**. A section describing the search and ranking component of your system. Here you can provide details about how you implemented the retrieval and ranking of image documents, as well as describing your user interface choices.

[2] If you cannot get access to a HTTP Server, I will organise a live zoom demonstration after the submission date.

- **Evaluation**. In this section, you should provide some basic evaluation of the quality of your search engine, highlighting 3 queries that you feel work well.
- **Conclusions**. This section provides an overview of the main findings of your project: what worked well and what did not, what you would change and how this work can be extended in the future.

Grading

Grading will be based on the actual search engine and the report content:

- 20% Functional Search Engine at a shared URL (see footnote 2), along with sample queries and a functional user interface.
- 40% Indexing and Retrieval description (html for standard goal and computer vision-based annotation for stretch goal).
- 20% Crawler description
- 20% Interface design and evaluation description

Submission

You should submit your report via Loop by midnight on the due date. This is an individual project, and students are supposed to work individually. Your report will be checked for plagiarism. We expect all of our students to conform to DCU Academic Integrity and Plagiarism Policy

Resources:

There are many available resources open-source resources to assist with this project:

- Crawlers: https://www.octoparse.com/blog/10-best-open-source-web-scraper#1
- OpenCV Image Annotation: https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/
- Open Source Search Engines: linuxlinks.com/searchengines/