Image Search Engine: Looking for a Needle in a Haystack

A dissertation submitted in partial fulfilment of

the requirements for the degree of

BACHELOR OF *ENGINEERING* in Computer Science

in

The Queen's University of Belfast

by

Adam Cregan

17/04/2023

SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE

CSC3002 – COMPUTER SCIENCE PROJECT

Dissertation Cover Sheet

A signed and completed cover sheet must accompany the submission of the Software Engineering dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name: | Adam Cregan | Student Number: | 40261335 |
| Project Title: | Image Search Engine: Looking for a Needle in a Haystack | | |
| Supervisor: | Dr Paul Miller | | |

Declaration of Academic Integrity

Before submitting your dissertation please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

**By submitting your dissertation you declare that you have completed the tutorial on plagiarism at** [**http://www.qub.ac.uk/cite2write/introduction5.html**](http://www.qub.ac.uk/cite2write/introduction5.html) **and are aware that it is an academic offence to plagiarise. You declare that the submission is your own original work. No part of it has been submitted for any other assignment and you have acknowledged all written and electronic sources used.**

1. If selected as an exemplar, I agree to allow my dissertation to be used as a sample for future students. (Please delete this if you do not agree.)

|  |  |  |  |
| --- | --- | --- | --- |
| *Student’s signature* | *Adam Cregan* | *17/04/2023* |  |

Acknowledgements

To those who have helped the author during the project and the preparation of the dissertation and to anybody who has given financial support.

Abstract

*A summary (100 words) which provides an outline of the subject matter and the results, findings and/or conclusions of the dissertation.*

This project entails creating an image search engine in which the user enters a query image, and the image search engine searches its image database for images that are most similar to the query image in terms of content. To represent an image, the search engine extracts feature from it (colour, shape, and texture). The query image feature representation is then compared to those in the database to find those that provide the best match based on distance measurements. The image database is divided into four categories: sporting crowds, Formula One cars, horses, and landscapes. The final model will have the best performance, prioritising precision then speed of retrieval. The image features should be stored externally to increase speed of retrieval and scalability.

Contents

A complete list of chapters, sections, appendices etc. with page numbers.

1. Introduction and Problem Area

Content-based image retrieval (CBIR) is a technique for retrieving images from a database based on their content, such as colour, texture, shape, and other visual characteristics. In the past, CBIR has been used in various scientific and technological applications. For example, in the field of medicine, CBIR has been used for the classification of different types of tumours in MRI images [1]. In video surveillance, CBIR has been utilized for the recognition of human activities and facial recognition [2]. In the field of remote sensing, CBIR has been used for the classification of land cover [3]. In addition, CBIR has been used in digital libraries for content-based image retrieval and browsing [4]. CBIR technology is currently in a promising state, and researchers are constantly working to boost its efficiency and precision. Advances in computer hardware, machine learning, and deep learning techniques have all played a significant role in improving the performance of CBIR systems. Researchers are also exploring novel methods, such as hybrid approaches combining multiple visual features and machine learning algorithms, to improve the accuracy and efficiency of CBIR systems [5].

CBIR systems often suffer from poor precision and slow response times, which are major issues that limit their effectiveness. In CBIR systems, poor precision refers to the inability to accurately retrieve images that match the user's intended query. This issue arises due to a number of factors, including low-level feature extraction methods, query ambiguity, and a semantic gap between low-level features and high-level semantic concepts [6]. Slow response times in CBIR systems refer to the time it takes to retrieve relevant images from a large image database, which can be caused by factors such as inefficient indexing and retrieval algorithms [7]. Storing image data externally can positively impact the speed and scalability of CBIR systems. The reduced latency and faster access times are two of the most significant benefits of storing image data externally. External storage solutions are intended to provide fast data access and can use advanced caching techniques to reduce data access time [8]. As a result, image retrieval times are reduced, and system performance is improved. Scalability is another critical advantage of storing image data externally. Scalability is another critical advantage of storing image data externally. CBIR systems must handle a large amount of data, and internal storage systems may not be able to handle the scale of these databases. These systems can also provide fault-tolerant and highly available data storage, which is critical for CBIR systems that require continuous access to image data.

1. System Requirements and Specification
   1. Solution Description

This application is developed in Eclipse IDE for Java Developers - 2022-12(4.26.0), using Java.

The project developed is a content-based image retrieval (CBIR) system that allows users to select a query image and retrieve a set of similar images from a gallery using effective distance measurement techniques for colour, texture, and shape.

The application will be designed with a Graphical User Interface (GUI) that will pop up when executed. The user will interact with the GUI to select a query image, which will be used to perform a search.

A functional requirement is that the image database will be split into two categories: gallery and test, with an 80/20 split ratio [9]. All the of the tests taken will use the test images, and the image database will use the images in the gallery. For this application there are 431 total images that will be used. Therefore, 80% = 334.8 and 20% = 86.2. This is rounded to 345 gallery images and 86 test images. The 4 image categories are Crowds, F1-Cars, Horses, and Landscapes, so the images will be split accordingly. 86 test images/4 categories = 21.5 test images. Therefore, the test folder will include 22 Crowds images, 21 F1-Cars images, 22 Horses images, and 21 Landscapes images.

User Characteristics

This application has been developed for a user with minimal technical skills, as this only requires control of the cursor to navigate the GUI. Moreover, the system is robust enough so that if the user does make mistakes that the system should catch it and allow the user to try again. This is covered in section 2.2 UI Requirements.

* 1. UI Requirements

The UI should be designed in a user-friendly way so that users can easily navigate and interact with the application. The UI should allow the user to select the query image, display the results of the search, and provide information about any errors or issues that may arise during the search process.

The UI should also include error handling mechanisms to inform the user if they click the "Search" button when no query image has been selected, that they must select a query image. Moreover, a pop-up message should appear if the application cannot access the image database when the user is trying to save/read data and inform the user of the situation. Furthermore, there are not enough images in the database to populate the results section when the user tries to search, a pop-up message should inform the user that there are not enough images in the database. Also, if the user selects an invalid file type then a pop-up message should appear telling the user to select a valid file type, and the search function is cancelled.

* 1. Functional Requirements

*Colour space and pixel representation:*

The images in the database will be stored in an appropriate colour space. HSV has been chosen due to its superior performance in similar applications against the RGB colour space [10, 11]. This CBIR system will also use colour histograms to represent the pixels as it is better other descriptors like colour moments and colour coherent vectors [12]. TEST!!!!

*Query image selection and resizing:*

The system shall allow the user to select a query image from either inside or outside the image database. If the query image is not of size 512x512 pixels, the system shall resize it to the required size. Test: The system shall correctly resize the query image to 512x512 pixels and display it in the GUI.

*Image search and comparison:*

The system shall allow the user to initiate a search for similar images in the gallery by clicking the search button in the GUI. The system will have multiple, manually implemented distance measurement techniques, including Correlation, Chi-Square, Intersection, and Bhattacharyya measures [?], to compare the colour histograms of the query image with those in the gallery. The system will iterate through all images in gallery and measure the distance to the query image. Then the system shall return the top 15 images from the gallery that are closest to the query image, based on the distance measures. Test: The system will correctly display the top 15 closest images in the GUI after a search is performed.

*Graphical user interface (GUI):*

The system shall have a GUI that displays buttons for image selection, search, update database, and clear data. The GUI will allow the user to select a query image using a file chooser dialog. The GUI will display the query image and closest images after a search is performed. Test: The GUI correctly displays all buttons and images and allows the user to select a query image and initiate a search.

*Scalability:*

The system shall improve scalability by calculating features offline. Test: The system shall correctly calculate features and store them externally so they can be accessed at any time.

*Sorting:*

The system shall implement a sorting method that uses a HashMap[?] to associate and rearrange elements from the list of unsorted images into the order of the sorted distance scores. Test: The system shall correctly sort the images in the gallery based on distance measures using the implemented sorting method.

* 1. Non-Functional Requirements

*Timers for Performance Analysis:*

The CBIR system should be designed to analyse the performance using timers. The system should display all image results in less to the end-user in less than 5 seconds. Timers will be used to calculate the time taken by the system to display the results. Test: The timers are correctly implemented in the system and that the system provides results to the developer on the console.

*Optimize Histogram Bins:*

The system will have 4 different distance measuring techniques (Correlation, Chi-Square, Intersection, and Bhattacharyya) where the number of bins in the histograms need to be optimised for precision and time for each measurement. Test: A bins number has been set for each measurement that is optimal based on tests using precision and time and selecting the bins number which performed best.

*Timer:*

Timers will be used to analyse the performance of the CBIR system. The best performing model should display all results in less than 5 seconds. Test: The system shall correctly use the implemented timers for analysis, so that model performance can be observed.

*Pop-up Message for Image Database Update:*

The application should generate a pop-up message to inform the user when the image database has been updated. Test: The pop-up message is displayed when the image database is updated, after the user chose to do so.

*Determine Best Model from Distance Measures:*

The performance of the system should be analysed using generated Precision-Recall curves. Test: Precision-Recall curves are used in selecting the model that performs the best.

*Determine Best Data Storage Technique:*

Various storage techniques, including using a database manager (SQL Server Management Studio Manager 19) and other simple file formats (.ser, .txt, .bin, .csv), will be tested when finding the best storage technique.The best method will be based on the time taken to update the database. Test: Storage methods mentioned above have been implemented and time tested, thus the best performing method will be deduced.

1. Design

This section should describe the design of your proposed system. Normally this several parts, depending on your project:

1. Architectural Description of the system – textual and/or diagrammatic. This could be a simple diagram showing the components and how they relate or it could describe the choice of architectural style or pattern used.
2. User Interface Design (if applicable). Show sketches of the design or screenshots with explanations of choices made, if necessary.
3. Software System Design. The role of each component and the interfaces between components should be described. There should be a clear correlation between your design and your specification.
4. Where applicable give a critical discussion of key design decisions/styles/patterns used; data model; UI design, external Interfaces, other important issues e.g. concurrency, event handling, data persistence, error and exception handling, fault tolerance, security, distribution of components.

The design should be linked to requirements and, where applicable give a critical discussion of key design decisions/styles/patterns used. There might be a data model, a UI design, details of external interfaces, and of other important issues e.g. concurrency, event handling, error and exception handling, security, data persistence. No particular notation or tool is mandated. A satisfactory design will show a grasp of the main design issues. For top marks aim for outstanding design documentation approaching that of the best professionals. Prove that you have a very strong grasp of the design issues and aim for documentation that could be passed on to a developer without the need for further explanation

Application sets maximum amount of memory that can be allocated to the Java Virtual Machine (JVM) when running Eclipse. This is set to 9GB

1. Implementation

You should describe any languages, packages, and libraries etc. that are used in the development of your system. There is no need to describe your code in detail. You may highlight data types and implementation techniques that are of special interest. If appropriate, you may provide:

1. Choice of implementation language(s)/ development environment(s)
2. Use of software libraries;
3. Key implementation decisions
4. A description of how some important functions and algorithms were implemented.
5. A description of how each component is implemented.

Program code can be accessed by the assessors via the git repository **so there is no need to include code listings**. It is recommended that you comment code appropriately (not excessively). Programs should be written in a clear style with good program structure and well-defined data structures. The program code should reflect its design and show an understanding of relevant implementation issues.

1. Testing

This section will be judged in tandem with other evidence including evidence of unit tests and/or test documentation on the Repo. There should be a discussion of Test Approach e.g. unit testing, system testing, regression testing etc; Test cases should be described and justified; Include Testing tools used and provide evidence that testing coverage was complete. Provide proof that testing was completed, either showing sample test history and/or describing automated tests.

1. System Evaluation and Experimental Results

Provide a summary evaluation of the success of the project with respect to criteria identified in the introduction. Different projects will have a different emphasis. In all cases you are expected to provide empirical results and to draw conclusions from those results. You may use your software to generate experimental results. Be sure to describe the methodology of your evaluation or experimentation. An experiment is typically described in terms of its goals, the hypotheses being tested, the subject of the experiment, what is being measured and what is controlled, the results obtained and the analysis and interpretation of those results. A discussion of the significance of your experimental results may be appropriate or why the new system you have developed improves on what was already there. Do your results agree with other previous work or ideas? How does your system compare with similar ones?

Alternatively (or additionally), you can assess the product in terms of how it compares with other similar products and/or in terms of user feedback (e.g. via a survey or interviews) or some measurable quality aspect such performance efficiency or reliability.

Draw conclusions on the *process* used in the project as well. What went well? What did not go well? What are the strengths of your solution or conclusions? What are the weaknesses? Suggestions for further work should also be discussed. You can be critical and draw a negative conclusion. Not all projects will be successful. A well-explained failure is as an acceptable an outcome as a spectacular success. Assessors are looking for excellence in a critical appraisal of the work and a convincing argument for the significance of contribution in the context of wider work. This section should be objective, fair and comprehensive.

In all cases, societal implications and commercial and economic aspects should be evaluated. Has your project an outcome that potentially could improve some community or group of people? Perhaps your project can impact on the lives of others for example In education, employment, health, public policy or services, security, the environment, general wellbeing etc. There may be commercial opportunities arising from your product or findings. Describe these and include how the project could eventually brought to deployment and to deliver value. Discuss the feasibility of doing that. It may be that your project could make some process more efficient. Try to quantify the savings or improvements, generally or in one or more scenarios. You should be realistic though and include the risks and any negative impacts of your work and the potential impact as well.

Your supervisor can guide you on what is appropriate, but typically the very best projects have shown results derived using scientific method, that could be publishable with little or no work or show an exemplary empirically based evaluation of a software product. Those projects will also fairly and honestly assess the potential impact of the work socially or economically.

Any publication of results of the student's work is left to the discretion of the supervisor, but you can expect appropriate credit to be given to your work.

1. Appendices

Appendices will not be marked but may be referred to by the assessor to aid their understanding. They are useful if there is something that helps in understanding earlier parts of the dissertation, but if included inline might break the flow or readability of the document. For example, there may be large tables of data, design documents, evidence of testing etc etc.

These should include as appropriate:

(a) A User manual giving details on how to use the software, including details of input data, output formats and error messages.

(b) Test results, if appropriate.

(c) Other information which is not convenient or appropriate to include in the main body of the dissertation.

References

A list of references to documents (books, papers, web pages etc.) which are referred to in the main body of the text. Use the IEEE citation style as detailed here <https://www.ieee.org/documents/ieeecitationref.pdf>. There is some guidance on referencing at <http://www.qub.ac.uk/cite2write/home.html>.

**The first citation should be the URL to the software code repository which should contain the code and any other resource required to run the software.**

[1] Arakeri, M.P., Ram Mohana Reddy, G. An intelligent content-based image retrieval system for clinical decision support in brain tumor diagnosis. Int J Multimed Info Retr 2, 175–188 (2013). https://doi.org/10.1007/s13735-013-0037-5

[2] Sultana, M., Gavrilova, M. (2013). A Content Based Feature Combination Method for Face Recognition. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds) Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013. Advances in Intelligent Systems and Computing, vol 226. Springer, Heidelberg. https://doi.org/10.1007/978-3-319-00969-8\_19.

[3] T. F. Stepinski, P. Netzel and J. Jasiewicz, "LandEx—A GeoWeb Tool for Query and Retrieval of Spatial Patterns in Land Cover Datasets," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 1, pp. 257-266, Jan. 2014, doi: 10.1109/JSTARS.2013.2260727. https://ieeexplore.ieee.org/abstract/document/6517548

[4] D. Edmundon, G. Schaefer and M. E. Celebi, "Similarity-Based Browsing of Image Search Results," 2013 IEEE International Symposium on Multimedia, Anaheim, CA, USA, 2013, pp. 502-503, doi: 10.1109/ISM.2013.97. https://ieeexplore.ieee.org/abstract/document/6746849

[5] Miguel Arevalillo-Herráez, Francesc J. Ferri, Salvador Moreno-Picot, “A hybrid multi-objective optimization algorithm for content based image retrieval”, Applied Soft Computing, Volume 13, Issue 11, 2013, Pages 4358-4369, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2013.06.016.

[6] R. Datta, D. Joshi, J. Li, and J. Wang, "Image retrieval: Ideas, influences, and trends of the new age," ACM Computing Surveys, vol. 40, no. 2, pp. 1-60, 2008. [Online]. Available: <https://doi.org/10.1145/1348246.1348248>

[7] S. Gandhani, R. Bhujade and A. Sinhal, "An improved and efficient implementation of CBIR system based on combined features," Third International Conference on Computational Intelligence and Information Technology (CIIT 2013), Mumbai, 2013, pp. 353-359, doi: 10.1049/cp.2013.2613. <https://ieeexplore.ieee.org/abstract/document/6950897>

[8] T. G. Bhat, H. R. Kulkarni and N. D. Gundi, "A novel approach for Content Based Image Retrieval from huge database sets," Third International Conference on Computational Intelligence and Information Technology (CIIT 2013), Mumbai, 2013, pp. 345-352, doi: 10.1049/cp.2013.2612. <https://ieeexplore.ieee.org/abstract/document/6950896>

[9] A. Urmaliya and J. Singhai, "Sequential minimal optimization for support vector machine with feature selection in breast cancer diagnosis," 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), Shimla, India, 2013, pp. 481-486, doi: 10.1109/ICIIP.2013.6707638. <https://ieeexplore.ieee.org/abstract/document/6707638>

[10] I. S. P. James, “Face Image Retrieval with HSV Color Space using Clustering Techniques”, The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), vol. 1, no. 1, Mar-Apr., pp. 18-19, 2013

<https://pdfs.semanticscholar.org/b83f/b49bcc91a41345ae839bfe6a95c96b93d7d9.pdf>

[11] G. Jaswal, A. Kaul, and R. Parmar, “Content based Image Retrieval using Color Space Approaches”, International Journal of Engineering and Advanced Technology (IJEAT), vol. 2, no. 1, Oct., pp.7, 2012

https://www.sciencedirect.com/science/article/pii/S0031320312002713

[12] P. Anantharatnasamy, K. Sriskandaraja, V. Nandakumar and S. Deegalla, "Fusion of colour, shape and texture features for content based image retrieval," 2013 8th International Conference on Computer Science & Education, Colombo, Sri Lanka, 2013, pp. 422-427, doi: 10.1109/ICCSE.2013.6553949.

https://ieeexplore.ieee.org/abstract/document/6553949