MỤC LỤC

Trang

MỞ ĐẦU

Ngày nay nhu cầu về tìm kiếm thông tin về một đối tượng cụ thể là rất nhiều. Chúng ta có thể dùng "**text**" để mô tả nội dung tìm kiếm, nhưng kết quả đạt được cho độ chính xác không cao. Do đó việc tìm kiếm bằng "**hình ảnh**" trực tiếp sẽ giúp cho kết quả chính xác hơn.

Trong bài báo cáo này, em xin trình bày một ứng dụng được xây dựng để tìm kiếm một hình ảnh cụ thể. Qua đây em xin cám TS. Lê Đình Duy đã cung cấp những kiến thức quý báu giúp em hoàn thành tốt bài báo cáo này.

Chương 1 – PHƯƠNG PHÁP THỰC HIỆN

1.1. Thiết kế giao diện

- Sử dụng phần mềm Matlab phiên bản 2017a.

1.2. Các thư viện bên ngoài

- Thư viện vlfeat tại trang http://www.vlfeat.org.
- AKM: gồm các hàm được xây dựng sẵn phục vụ cho việc xây dựng tự điển.
- Mã nguồn tham khảo thêm: https://github.com/nvtiep/Instance-Search
- Bộ dữ liệu Oxford Building (5K): http://www.robots.ox.ac.uk/~vgg/data/oxbuildings

1.3. Các bước thực hiện

- Đầu tiên em sử dụng chức năng Matlab GUI để xây dựng giao diện cho chương trình chính, tham khảo nguồn SourceCode trên để hiểu được các chức năng từng hàm.

- Từ đó em đã viết lại SourceCode chia lại các module chức năng áp dụng đúng yêu cầu của đồ án, đồ án bao gồm :

Các thư mục chính:

- Thư mục **AKM**: chứa các hàm mở rộng phục vụ build từ điển cho đồ án
- Thư mục **oxford** gồm:

+ Thư mục **Feat**: phục vụ lưu trữ file được tạo qua các bước của chương trình : rút trích đặc trưng (feature.bin), số đặc trưng rút trích được của mỗi ảnh (feat_info.mat), lưu vector từ và từ điển từ được định nghĩa từ file feature.bin (words.mat, dict.mat).

+ Thư mục **Images**: lưu trữ bộ dataset ảnh Oxford Building (5K).

- Thư mục **vlfeat**: chứa thư viện mở rộng vlfeat được tải từ nguồn trên, thư viện hỗ trợ này đã implements sẵn nhiều thuật toán hỗ trợ trong lĩnh vực computer vision.

Các tập tin chính:

- **ExtractFeatures.m** : tập tin định nghĩa hàm rút trích đặc trưng ảnh.

- **BuildDictionary.m** : tập tin xây dựng bộ từ điển Term-Doc, được tính toán dựa trên file đặc trưng đã rút trích.

- **VectorWords.m** : tập tin tạo mảng lưu giá trị vector từ trên từng ảnh.

- **CreateInverted.m** : tập tin tạo mảng lưu giá trị Posting List của từng từ.

- **ProcessImage.m** : tập tin tổng hợp gọi từng bước thực hiện các chức năng các file trên, được xem như các bước trong qui trình xử lý.

- **QueryImage**: tập tin nhận ảnh hay 1 phần ảnh từ người dùng thực hiện rút trích đặc trưng trên ảnh, sau đó so sánh với đặc trưng đã học từ hệ thống và trả về kết quả phù hợp cho người dùng.

- **VIR.fig**: tập tin giao diện chính của chương trình.

- **VIR.m**: tập tin chứa SourceCode thực thi từ các chức năng trên giao diện người dùng (VIR.fig), nhận hình ảnh và truyền đối số vào QueryImage để nhận kết quả trả về và hiển thị.

Chương 2 – CÁC KỸ THUẬT SỬ DỤNG

- Đây là các kỹ thuật được em thao khảo và sử dụng.

+ **Bước rút trích đặc trưng:** Áp dụng thuật toán SIFT được xây dựng sẵn trong thư viện **vlfeat** bằng cách gọi hàm vl_covdet:

[frame, sift] = vl_covdet(I, 'method', 'Hessian', 'estimateAffineShape', true);

+ **Bước xây dựng từ điển:** Áp dụng thuật toán phân lớp với kmean được xây dựng sẵn trong AKM bằng cách gọi hàm ccvBowGetDict:

dict_words = ccvBowGetDict(features(:,randIndex(1:100)), [], [], num_words, 'flat', 'akmeans', ... []. dic_params);

+ **Bước xây dựng vector từ tần xuất xuất hiện của từ**: Áp dụng hàm ccvBowGetWords từ AKM.

words{i} = ccvBowGetWords(dict_words, features(:, bIndex:eIndex), [], dict);

+ **Bước tạo Posting List** : Sử dụng hàm ccvInvFileInsert từ AKM

inv_file = ccvInvFileInsert([], words, num_words).

Chương 3 – THIẾT KẾ CHƯƠNG TRÌNH VÀ HỆ THỐNG ĐÁNH GIÁ

3.1. Các bước chạy chương trình

- Trỏ đường dẫn matlab 2017a về thư mục gốc giải nén được.

- Đăng ký đường dẫn trỏ tới thư mục **vlfeat** trong thư mục vừa giải nén, trong comment nhập :

VLFEATROOT='\vlfeat'

- Tải tập tin **feature.bin** từ đường dẫn: https://drive.google.com/file/d/1Eyq2D-EyruydSAw1jIhL44mvBXuLIJw9/view?usp=sharing

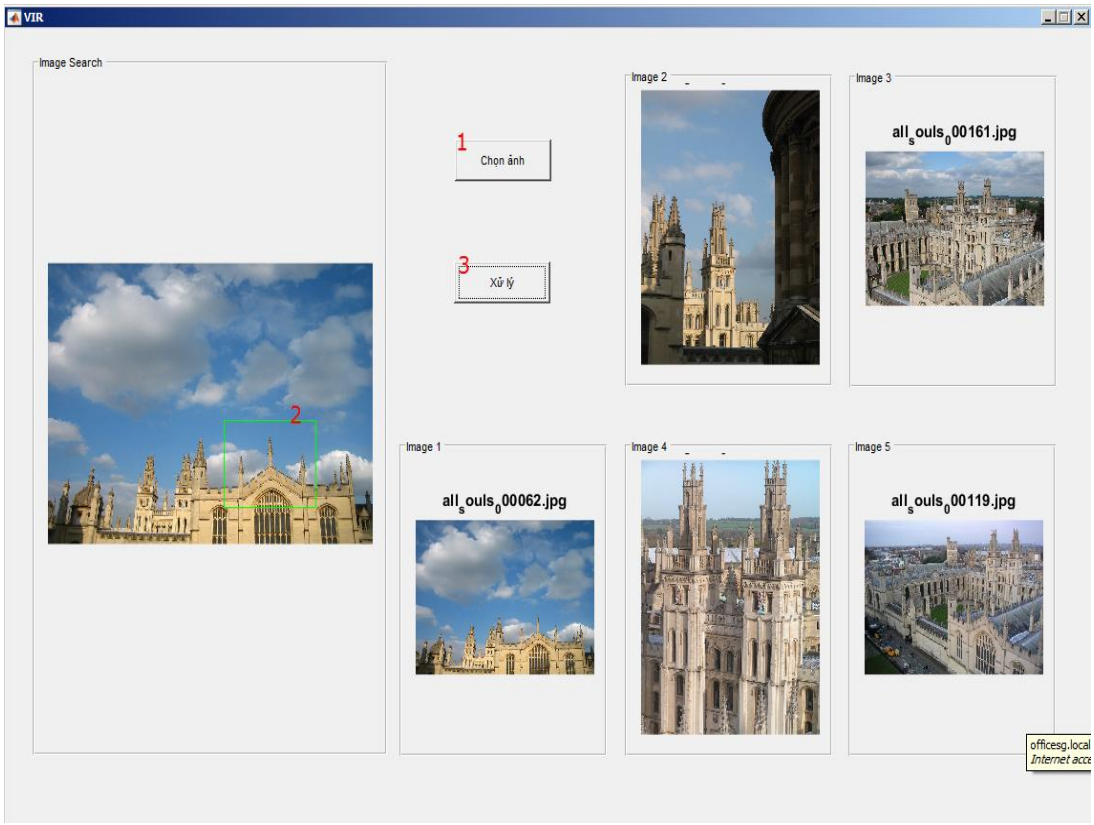
Về thư mục : **\oxford\feat\feature.bin**

- Download file **oxbuild_images.rar** từ đường dẫn : http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/oxbuild_images.tgz

Giải nén toàn bộ ảnh vào thư mục: **\oxford\images**

- Mở project chạy file code giao diện chương trình : VIR.m

3.2. Giao diện chính của chương trình



- Các thao tác thực hiện.

+ Nhấn nút chọn ảnh để chọn ảnh cần tìm từ ổ đĩa cứng, ảnh sẽ được show ở khung bên trái.

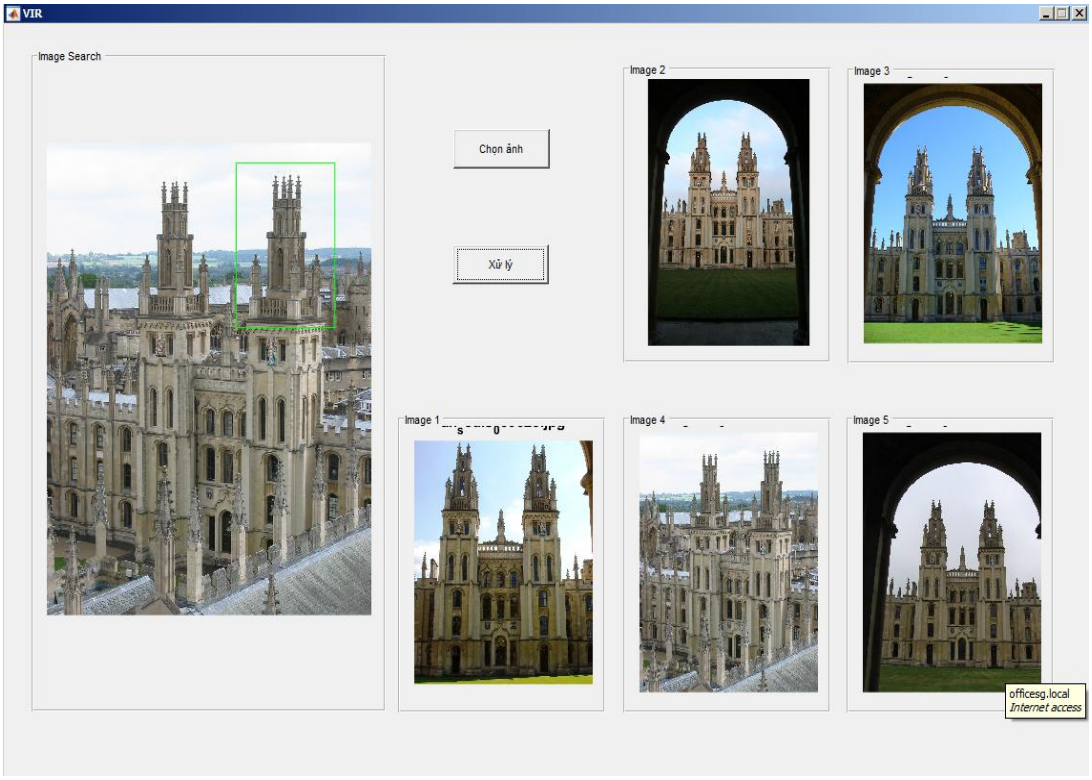
+ Nhấn chuột trên vùng ảnh xuất hiện biểu tượng dấu +, bắt đầu nhấn chuột quét chọn 1 vùng, vùng được chọn thể hiện trong khung màu xanh lá.

+ Nhấn nút xử lý hệ thống show ra 5 ảnh kết quả ở khung bên phải.

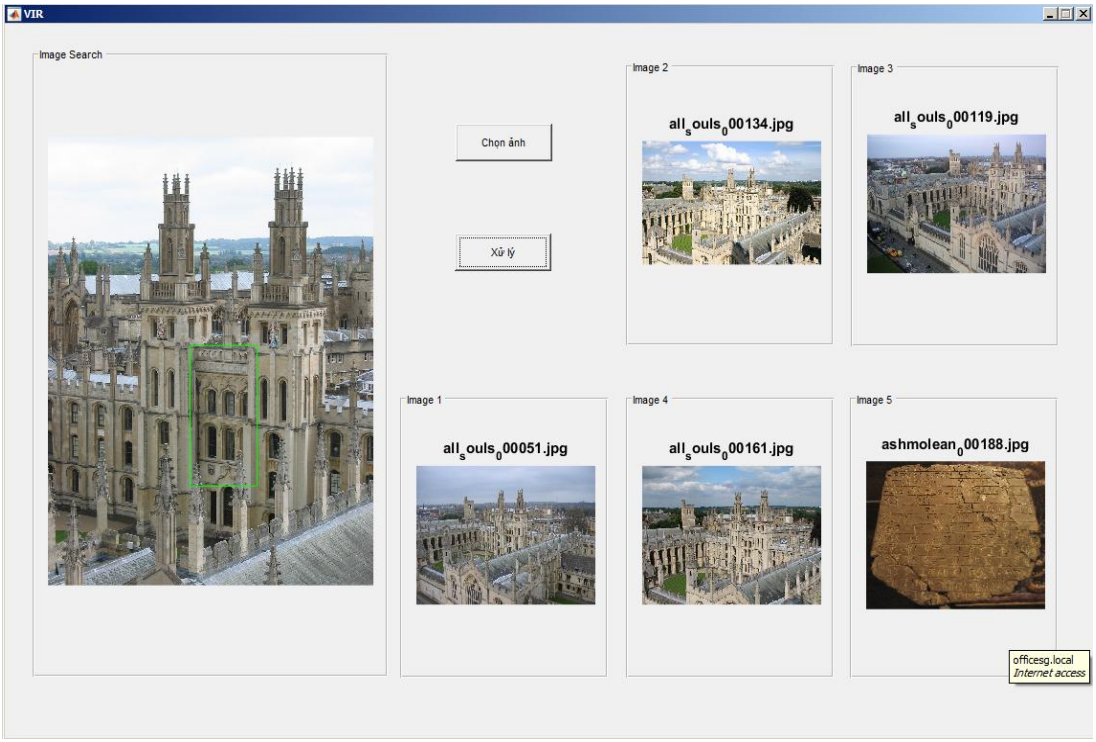
3.3. Đánh giá hệ thống

- Xét trên 2 câu truy vấn ngẫu nhiên về chủ đề building, dựa trên top 5 kết quả trả về:

+ Câu truy vấn 1



+ Câu truy vấn 2



- Kết quả

Relevent document cho câu truy vấn 1 = 5

Ranking #1						
Recall		0.2	0.4	0.6	0.8	1
Precission		1	1	1	1	1
Average Precission = (1 + 1 + 1 + 1 + 1) / 5 = 1						

Relevent document cho câu truy vấn 2 = 4

Ranking #2						
Recall		0.2	0.4	0.6	0.8	0.8
Precission		1	1	1	1	0.8
Average Precission = (1 + 1 + 1 + 1 + 0.8) / 5 = 0.96						

Mean Average Precission = (1 + 0.96) / 2 = 0.98

- Kết luận:

+ Câu truy vấn 1 cho kết quả tốt hơn câu truy vấn thứ 2, do vùng chọn có đặc trưng rõ ràng và riêng biệt hơn so với vùng chọn của câu truy vấn thứ 2.

+ Tổng quan lại, hệ thống hiệu quả và cho kết quả rất chính xác trên truy vấn với building do xét trên 2 câu truy vấn ngẫu nhiên đo được MAP = 0.98

Chương 4 – TÀI LIỆU THAM KHẢO

- C. Manning, P Raghavan, H. Schutze: **Introduction to Information Retrieval**, 2008.

- Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Addison-Wesley: **Modern Information Retrieval**, 2011.

- Stefan Buttcher, Charlie Clarke, Gordon Cormack, MIT Press: **Information Retrieval:Implementing and EvaluatingSearch Engines**, 2010.

- https://github.com/nvtiep/Instance-Search