ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Báo cáo đề tài

VISUAL RECOGNITION AND APPLICATIONS

NHẬN DIỆN THỊ GIÁC VÀ ỨNG DỤNG

Môn học: Nhận diện thị giác và ứng dụng

GIẢNG VIÊN

TS. Lê Đình Duy – TS. Nguyễn Tấn Trần Minh Khang

HỌC VIÊN:

PHẠM QUANG ANH KHA – CH1501027

TP. HỒ CHÍ MINH, 2017

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc tới TS. Lê Đình Duy, TS. Nguyễn Tấn Trần Minh Khang, người đã trực tiếp tận tình hướng dẫn chỉ bảo chúng em trong quá trình học tập và nghiên cứu môn học Nhận diện thị giác và ứng dụng cũng như định hướng cho em hoàn thiện đề tài này.

TP Hồ Chí Minh, tháng 7 năm 2017

Học viên thực hiện

Phạm Quang Anh Kha

MỤC LỤC

1. Giới thiệu đề tài

Tra cứu, truy vấn thông tin trên các tài liệu, dữ liệu đã được lưu trữ trên máy tính là nhu cần cơ bản. Tương tự, với một kho dữ liệu ảnh đã có sẵn, nhu cầu truy vấn để lấy ra những hình ảnh theo một nhu cầu cụ thể nào đó là điều tất yếu. Bài toán truy vấn ảnh có thể phát biểu như sau:

- *Điều kiện : Cơ sở dữ liệu hình ảnh*
- *Input: Nhập vào từ khoá hoặc ảnh cần truy vấn*
- *Ouput: Tập các hình ảnh có liên quan đến từ khoá truy vấn hoặc gần giống nhất (có độ tương đồng cao – similarity) với ảnh truy vấn ở Input.*

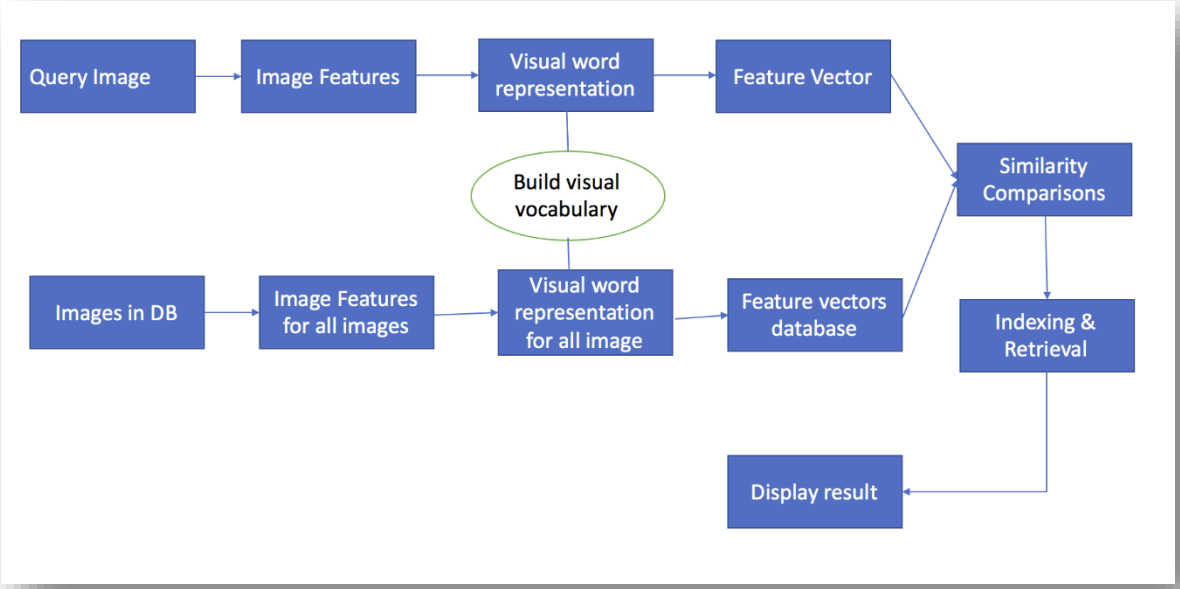
Content-based image retrieval (CBIR) là phương pháp được sử dụng trong các hệ thống truy vấn ảnh hiện đại. Phương pháp này ứng dụng computer vision vào việc truy vấn ảnh; dựa trên độ tương đồng (similarity) về nội dung (texture, màu sắc, hình dáng,…) của các ảnh trong cơ sở dữ liệu (CSDL) với ảnh mà người dùng cung cấp.

❖ Mục tiêu, phạm vi đề tài:

- Tìm hiểu về cách thức xây dựng một hệ thống truy vấn ảnh dựa trên Content-based image retrieval (CBIR).
- Xây dựng một ứng dụng đơn giản với giao diện người dùng cho phép chọn ảnh tìm kiếm, sau đó truy vấn và hiển thị danh sách ảnh kết quả (đi kèm với độ tương đồng của chúng với ảnh truy vấn). Sử dụng bộ dữ liệu Oxford Building (5K).

2. Content Based Image Retrieval (CBIR)

2.1 Mô hình tổng quát



Hình 1: Hệ thống truy vấn ảnh CBIR đã được sử dụng trong đề tài áp dụng Vector Space Model

2.2 Chi tiết các thành phần

Theo mô hình tổng quát phía trên, việc truy vấn ảnh sử dụng **Vector Space Model (VSM);** dựa trên việc biến ảnh truy vấn thành một vector đặc trưng đại diện cho ảnh đó. Công việc hoàn toàn tương tự được áp dụng cho tất cả các ảnh trong CSDL. Sau quá trình này, mỗi ảnh (ảnh truy vấn và ảnh trong CSDL) sẽ được biểu diễn bằng một Vector. Tính độ tương đồng (similarity) giữa vector của ảnh truy vấn với các vector của các ảnh trong CSDL, ta sẽ thu được danh sách các ảnh giống với ảnh truy vấn đi kèm với độ tương đồng.

2.1.1 Xây dựng vector đặc trưng cho ảnh truy vấn và ảnh trong CSDL

Ý tưởng: Coi mỗi ảnh như một tài liệu (document), ta có thể áp dụng cơ chế tìm kiếm dữ liệu dạng text trong việc tìm kiếm ảnh. Mỗi tài liệu được xây dựng từ tập các bộ từ vựng (vocabulary). Từ đó nảy ra ý tưởng xây dựng tập các bộ từ vựng cho ảnh gọi là visual vocabulary.

Xây dựng visual vocabulary:

- Với mỗi ảnh trong CSDL, ta tiến hành trích rút các đặc trưng ảnh (image feature), sau đó dùng đặc trưng này để tính ra được SIFT descriptors (là một vector 128 chiều, đại diện cho đặc trưng của ảnh).
- Từ tập tất cả các SIFT descriptors của tất cả các ảnh (là các vector trong không gian 128 chiều), ta tiến hành gom cụm để thu được k cụm (với k là số lượng từ trong visual vocabulary – do người cài đặt tự quy định). Tâm của mỗi cụm trong k cụm sẽ đại diện cho một visual word.

Biến đổi ảnh về dạng vector trong Vector space model:

- Bước đầu tiên là thể hiện một ảnh dưới dạng tập hợp các visual word (như tập hợp các từ vựng trong tài liệu truyền thống).
    - Tính SIFT descriptors của ảnh đó; với mỗi SIFT descriptor, tiến hành xác định xem SIFT descriptor đó thuộc cụm nào trong k cụm thu được ở bước xây dựng visual vocabulary → SIFT descriptor tương ứng với visual vocabulary của cụm đó.

- Sau khi thực hiện bước phía trên với tất cả các SIFT descriptor, ta thu được **bag of visual word** của ảnh; là một histogram thể hiện cho số lần xuất hiện của các visual word trong ảnh.

  Ví dụ giả sử xây dựng visual vocabulary với số từ vựng k = 5. Sau khi thực hiện các bước trên cho một ảnh X, ta thu được vector như sau:

VW1*	VW2	VW3	VW4	VW5
15	22	4	7	0

**viết tắt VW = Visual word*

  Vector V (15, 22, 4, 7, 0) sẽ đại diện cho hình ảnh X; có 15 VW1 xuất hiện trong ảnh X, 22 VW2 xuất hiện trong ảnh X, …

- Áp dụng phương pháp đánh trọng số TF-IDF weighting để cập nhật giá trị cho các feature vector đại diện cho các ảnh.
  - Sau bước trên mỗi ảnh sẽ là một vector $V(t_1, t_2, t_3, \dots, t_k)$ với số chiều k = số lượng từ vựng.
  - Giá trị t_i được cập nhật lại theo công thức TF-IDF weighting như sau:

    $$t_i = \frac{n_{id}}{n_d} \times \left(1 + \log\left(\frac{N}{n_i}\right)\right)$$

    trong đó t_i là giá trị tại chiều thứ i (tại vị trí visual word thứ i trong tập visual vocabulary) của feature vector đại diện cho ảnh.
    - ✓ *n_{id}: số lần xuất hiện của visual word thứ i trong ảnh*
    - ✓ *n_d: tổng số visual words trong ảnh*
    - ✓ *N: là tổng số ảnh trong CSDL*
    - ✓ *n_i: số ảnh trong CSDL có chứa visual word thứ i*

2.1.2 Truy vấn và trả về ảnh kết quả

Với mô hình Vector Space Model, việc tìm kiếm được ảnh tương tự với ảnh truy vấn được hiện thực bằng các tìm các vector đại diện cho các ảnh trong CSDL sao cho các vector này tương đồng với vector đặc trưng của ảnh truy vấn. Độ đo tương đồng giữa 2 vector được sử dụng trong đề tài là độ Cosin:

$$\cos(q, v) = q.v / |q|*|v|$$

trong đó q là vector đặc trưng của ảnh truy vấn; v là vector đặc trưng cho 1 ảnh trong CSDL. Khi giá trị cos (q, v) càng gần giá trị 1 thì độ tương đồng càng cao, nghĩa là ảnh có vector đặc trưng v càng giống với ảnh truy vấn.

Lần lượt tính độ tương đồng cosin của vector q với tất cả các vector đặc trưng của các ảnh trong CSDL, sau đó sắp xếp các kết quả giảm dần theo độ tương đồng cosin → ta thu được danh sách các ảnh giống với ảnh truy vấn (theo mức độ giống nhau giảm dần) → Hiển thị danh sách ảnh kết quả cho người dùng.

3. Cài đặt thực tế

3.1 Giới thiệu các công cụ, thư viện sử dụng

3.1.1 Thư viện VLFeat

VLFeat là một thư viện mã nguồn mở cài đặt các thuật toán liên quan đến xử lý ảnh như rút trích đặc trưng ảnh (local features extraction) và matching ảnh. Các thuật toán cơ bản như Fisher Vector, SIFT, k-means,… được cài đặt bằng ngôn ngữ

C với hiệu suất tính toán cao và phù hợp phát triển ứng dụng với nhiều nền tảng khác nhau (Windows, Mac OS X, Linux). Phiên bản mới nhất của VLFeat là 0.9.20. VLFeat hỗ trợ tích hợp và chạy được trên Matlab.

Báo cáo sử dụng hàm trong VLFeat để tính toán SIFT descriptor vector và chạy thuật toán gom cụm k-means khi xây dựng bộ visual vocabulary.

3.1.2 Bộ dữ liệu Oxford Building

Hệ thống xây dựng sử dụng bộ dữ liệu Oxford Building. Oxford Buildings Dataset là tập dữ liệu gồm 5062 ảnh về các địa danh nổi tiếng của Oxford được sưu tập từ trang lưu trữ ảnh Flick. Tập dữ liệu có thể được download miễn phí tại địa chỉ http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/index.html.

Đây là tập dữ liệu sử dụng trong bài báo "Object retrieval with large vocabularies and fast spatial matching" do đó mức độ tin cậy của dữ liệu được đảm bảo.

3.2 Cài đặt chi tiết hệ thống

Hệ truy vấn ảnh CBIR được cài đặt sử dụng Matlab, thư viện VLFeat, chạy trên dữ liệu Oxford Building.

3.2.1 Xây dựng lớp IRFinalProject

Trong Matlab, ta xây dựng lớp IRFinalProject (file matlab: IRFinalProject .m) bao gồm tất cả các phương thức cần thiết cho việc truy vấn và lấy về danh sách các ảnh kết quả.

Các thuộc tính:

- data_path: đường dẫn tới Oxford Building Dataset.
- vl_feat: đường dẫn tới file vl_setup.m của VLFeat library.

- vocab_size: số lượng từ trong bộ từ vựng (tương ứng với giá trị k cụm khi tiến hành gom cụm các SIFT descriptors của các ảnh trong CSDL)
- db_size: số lượng ảnh trong CSDL.

Các phương thức chính:

- function createTrainVocalbulary(image_paths, vocal_size)
- function createVisualWord(image_paths)
- function ranked_list = getCosineValue(imgListPath, query_vector, db_size);

3.2.2 Chi tiết cài đặt của các hàm trong IRFinalProject

❖ function createTrainVocalbulary: Hàm này giúp tạo ra bộ visual vocabulary.

Tập dữ liệu traning lớn nên việc dùng toàn bộ ảnh để tạo ra bộ visual vocabulary rất tốn thời gian, do đó ta chỉ sử dụng một tập con các ảnh ngẫu nhiên trên tổng số ảnh của dataset để thực hiện công việc này

```
function vocab = createTrainVocabulary( image_paths, vocab_size )
    images = cellfun(@imread, image_paths, 'UniformOutput', false);
    [~, all_SIFT_features] = vl_dsift(single(rgb2gray(images{1})),'fast', 'step', 50);
        for i = 2:(size(images,1))
            [~, SIFT_features] = vl_dsift(single(rgb2gray(images{i})),'fast', 'step', 50);
            all_SIFT_features = cat(2, all_SIFT_features, SIFT_features);
        end
    [vocab, ~] = vl_kmeans(single(all_SIFT_features), vocab_size);
end
```

Hình 2: Xây dựng visual vocabulary

Với mỗi ảnh đầu vào, tính được SIFT descriptors với hàm vl_dsift của thư viện VLFeat. Tiến hành gom tất cả những SIFT descriptors thành k cụm (*k = vocab_size = số lượng từ trọng visual vocabulary; trong code cài đặt tác giả sử dụng giá trị vocab_size = 500*) thu được sử dụng hàm vl_kmeans của thư viện VLFeat.

Biến vocab thu được là một matrix chứa thông tin về các vector, mỗi vector là tâm của 1 cụm (ở bước gom cụm phía trên) đại diện cho 1 visual word. Ta tiến hành lưu biến vocab này lại thành tập tin với hàm save của matlab để không phải xây dựng visual vocabulary sau này.

❖ function createVisualWord:

Hàm này giúp tạo ra CSDL mới gồm tập hợp các vector đặc trưng cho các ảnh trong CSDL hình ảnh ban đầu. Hàm nhận tập hợp tất cả các đường dẫn của các ảnh trong CSDL. Với mỗi ảnh trong CSDL sử dụng hàm vl_shift của thư viện VLFeat để tính ra các SIFT descriptors. Với mỗi SIFT descriptor thu được, tiến hành xác định xem nó thuộc về visual vocabulary nào (sử dụng hàm knnsearch - K nearest neighbors của Matlab) ➔ thu được bag of visual words → tiến hành cập nhật trọng số cho vector theo TF-IDF weighting.

```
parfor i=1:(size(images,1))
    [~, feats] = vl_dsift(single(rgb2gray(images{i})), 'fast', 'step', 5);
    D_matrix = zeros([1 vocab_size]);

    all_nearest = knnsearch(single(vocab_inv), single(feats'));
    for n=1:size(all_nearest, 1)
        nearest_vocab=all_nearest(n);
        D_matrix(nearest_vocab) = D_matrix(nearest_vocab)+1;
    end
    %out = 1/norm(D_matrix)*D_matrix;
    %image_feats(i, :) = out;
    image_feats(i, :) = D_matrix;
end
```

Hình 3: Tính Bag of visual words của ảnh

```
for i=1:size(image_feats, 1)
    for j = 1: size(image_feats, 2)
    %update tf-idf weighting
    total_words = number_of_words_in_each_doc(i,1);
    word_frequency = vocab_frequencies_in_DB(1,j);
    image_feats(i, j) = image_feats(i, j)/total_words * (1 + log(db_size/word_frequency));
    end
end
```

Hình 4: Cập nhật giá trị của vector đặc trưng cho ảnh sử dụng TF-IDF weighting

Biến imgvectors thu được là một matrix chứa thông tin về các vector đặc trưng đại diện cho các ảnh trong CSDL. Ta tiến hành lưu biến imgvectors này lại thành tập tin với hàm save của matlab để không phải tính lại sau này.

❖ function descending_ranked_list = get_ranked_result_for_query

```
load('imgvectors.mat');
query_vector = createQueryVector(imgQueryPath, db_size);

%get ranked list result
ranked_list =  getCosineValue(imgListPath, query_vector, db_size);
[Y,I]=sort([ranked_list{2,:}], 'descend');
descending_ranked_list = ranked_list(:,I);
```

Hình 2: Cài đặt hàm lấy về danh sách các ảnh phù hợp với ảnh truy vấn

Hàm này nhận vào tham số là đường dẫn của ảnh truy vấn, đầu tiên vector đặc trưng cho ảnh này được tính. Các thức tính hoàn toàn tương tự như việc tính vector đặc trưng của các ảnh trong CSDL như trong hàm createVisualWord. Sau đó, tiến hình tính độ tương đồng cosin của query vector và các vector đại diện của các ảnh trong CSDL cho về danh sách các ảnh phù hợp sau đó sắp xếp danh sách này theo độ tương đồng giảm dần.

```
for i = 1:nRow
    u = query_vector;
    v = imgvectors(i,:);
    a=imgListPath{i};
    ranked_list{1, i} = a;
    ranked_list{2, i} = dot(u,v)/(norm(u,2)*norm(v,2));
end
```

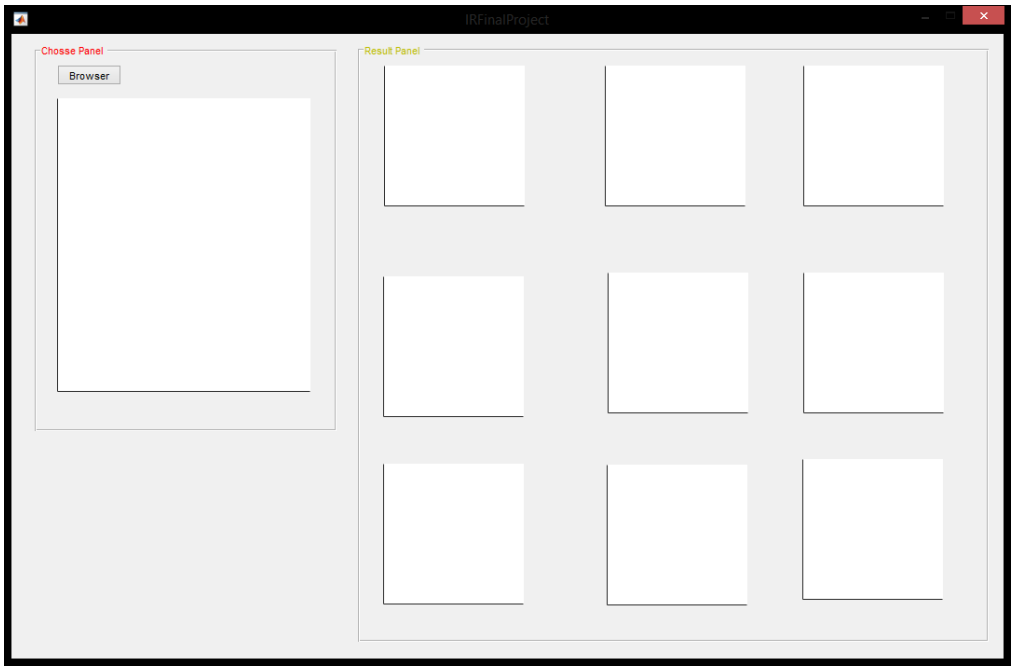
Tính vô hướng của 2 vector và độ dài của vector lần lượt tính bằng hàm dot() và hàm norm được cung cấp sẵn bởi Matlab.

3.2.3 Hiển thị ảnh kết quả

```
for n=1:9
    abc=rankValue{n};
    textLabel = sprintf('%d', abc);
    set(position(n), 'String', textLabel);
    imagePath = listImg{n};
    Selected_Image = imread(imagePath);
    imshow(Selected_Image, 'Parent', images(n));
end
```

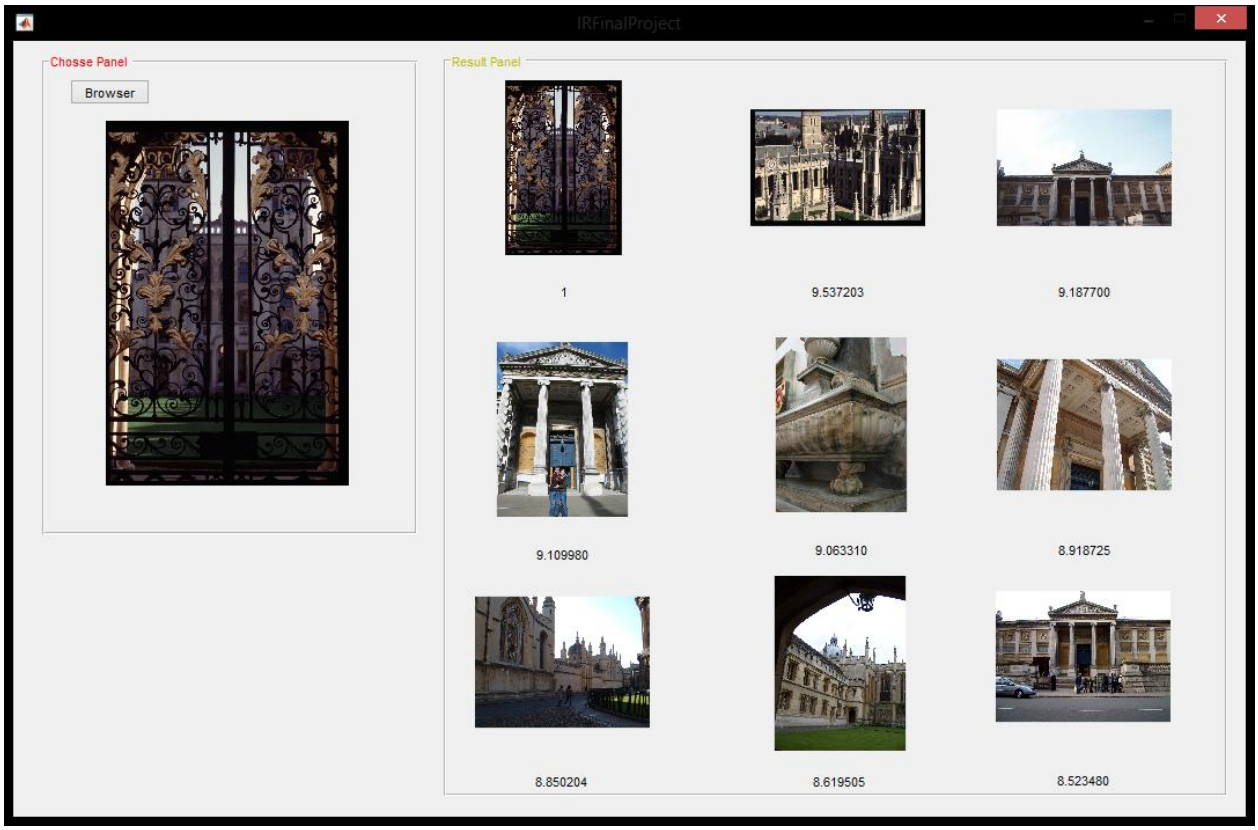
Hình 5: Hiển thị kết quả truy vấn

3.3 Kết quả thu được



Hình 6: Giao diện ứng dụng demo

Giao diện ứng dụng đơn giản, người dùng ấn nút “Browser” để lựa chọn ảnh cần truy vấn. Chương trình sẽ tính toán và hiển thị ra 10 ảnh cùng với độ tương đồng giống nhất với ảnh truy vấn.



Hình 7: Giao diện hiển thị kết quả

4. Kết luận

Sau khi thực hiện đề tài này, một số đánh giá tổng kết của tác giả được trình bày đưới đây:

Kết quả đạt được:

- Hiểu rõ cơ chế hoạt động của hệ thống Content Based Image Retrieval (CBIR); kỹ thuật TF-IDF weighting và mô hình Vector Space Model (VSM) trong truy vấn thông tin.
- Cài đặt được một ứng dụng truy vấn hình ảnh đơn giản (áp dụng CBIR, VSM và TF-IDF weighting) trả về danh sách ảnh kết quả với độ tương đồng so với ảnh truy vấn; cài đặt ứng dụng sử dụng Matlab và thư viện VLFeat.

Hạn chế và hướng phát triển:

- Ứng dụng demo muốn chạy phải cài đặt Matlab, VLFeat; chưa phải là một sản phẩm có thể chạy riêng biệt ➔ cần nghiên cứu việc export code Matlab ra file .dll hoặc .jar để xây dựng ứng dụng mang tính thực tế.

TÀI LIỆU THAM KHẢO

[1]. Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, *A "Object retrieval with large vocabularies and fast spatial matching"*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007) .

[2]. Ts. Lê Đình Duy, Ts. Nguyễn Tấn Trần Minh Khang, slide môn học Nhận diện thị giác và ứng dụng, *"Visual Recognition and Applications)"*, 2017.

[3]. Ts. Ngô Đức Thành, slide môn học Truy vấn thông tin thị giác, *"CS 2224 Visual Information Retrieval (Image Retrieval)"*, 2016.

[4]. Thư viện vlfeat, http://www.vlfeat.org/

[5]. Bộ dữ liệu Oxford Building (5K), http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/

[6]. Matlab, https://www.mathworks.com/products/matlab.html

[7]. Image Retrieval Using Customized Bag of Features,https://www.mathworks.com/help/vision/examples/image-retrieval-using-customized-bag-of-features.html