ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC:

NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

Đề tài:

XÂY DỰNG HỆ THỐNG TÌM KIẾM ẢNH DÙNG PHƯƠNG PHÁP BAG OF VISUAL WORDS

GVHD: TS. Lê Đình Duy

TS. Nguyễn Tấn Trần Minh Khang

HỌC VIÊN: Nguyễn Anh Tú – CH1601020

Lớp: Cao học khóa 11

TP. HỒ CHÍ MINH - NĂM 2017

MỤC LỤC

LỜI NÓI ĐẦU

Trong lĩnh vực nhận dạng thị giác và ứng dụng thì tìm kiếm thông tin bằng hình ảnh là một trong những bài toán được ứng dụng nhiều của khoa học máy tính. Việc tra cứu bằng cách đưa vào một tấm ảnh, sau đó hệ thống sẽ phân tích những đặc trưng và truy tìm trong cơ sở dữ liệu hình ảnh để đưa ra danh sách các hình ảnh có những điểm tương đồng là một bài toán mà đã có nhiều giải pháp thực hiện, sau đây em xin trình bày một hệ thống truy tìm hình ảnh mà em đã tìm hiểu và thực hiện.

Em xin chân thành cảm ơn TS. Lê Đình Duy, TS. Nguyễn Tấn Trần Minh Khang, những tiết giảng về lý thuyết cũng như thực hành quý báu của quý Thầy đã cung cấp cho em những kiến thức nền tảng về các bước trong lĩnh vực nhận dạng thông tin thị giác.

Dù có nhiều cố gắng nhưng do những điều kiện khách quan cũng như năng lực xây dựng đồ án này còn chưa được hoàn hảo, mong được quý Thầy góp ý thêm cho em có đi hướng tốt hơn.

Một lần nữa em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, ngày 31 tháng 12 năm 2017

Học viên thực hiện

Nguyễn Anh Tú

CH1601020 – KHMT K11

CHƯƠNG 1: MỤC TIÊU, PHẠM VI VÀ QUY TRÌNH THỰC HIỆN

1.1. Mục tiêu và phạm vi đề tài

- Xây dựng hệ thống truy vấn tìm các ảnh có nội dung tương tự trong dataset Oxford bao gồm 5063 ảnh (Oxford Building (5K)).
- Input đầu vào là một ảnh truy vấn (ảnh có nội dung thuộc tập Groundtruth Queries của dataset Oxford), output là danh sách ảnh được sắp xếp từ thấp đến cao theo mức độ tương đồng với ảnh truy vấn.

1.2. Quy trình thực hiện

1.2.1. Chuẩn bị môi trường

- Cài đặt Matlab 2017a.
- Cài đặt Visual Studio 2008.
- Download kho dữ liệu hình ảnh từ địa chỉ https://drive.google.com/a/gm.uit.edu.vn/file/d/1SE9WTKXznrNy1zxtHpx0GWGYsGI0gXAV/view?usp=sharing

  (nguồn ảnh: http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/)

1.2.2. Xây dựng hệ thống truy vấn

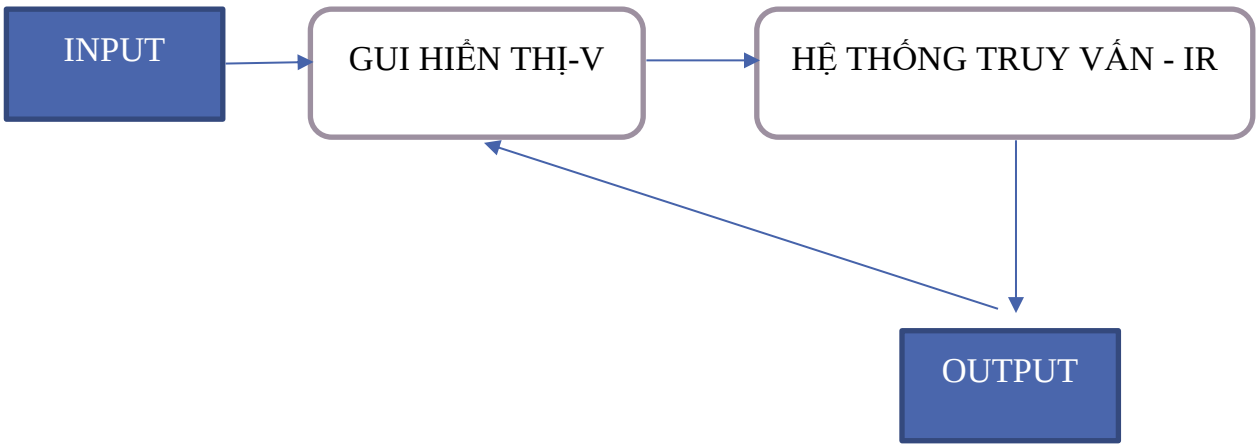
- Thực hiện việc training cho dataset gồm 5063 ảnh: rút trích đặc trưng của dataset này thành tập tin feature.bin, ngoài ra ta cũng xây dựng Visual Words (dictionary), tính WordID, Inverted file (file chỉ mục ngược) và lưu thành các tập tin để sử dụng trong bước truy vấn.
- Thực hiện truy vấn ảnh: input ảnh query đầu vào, rút trích đặc trưng, tính Word-ID và xây dựng ranked list và trả ra kết quả các ảnh tương tự cho ảnh truy vấn.

1.2.3. Xây dựng hệ thống nhận và trả kết quả

- Xây dựng thư mục ảnh làm ảnh query (ảnh tra cứu).
- Xây dựng giao diện tra cứu hình ảnh.
- Tra cứu bằng một tấm ảnh.

CHƯƠNG 2: CHI TIẾT CÁC BƯỚC XÂY DỰNG HỆ THỐNG

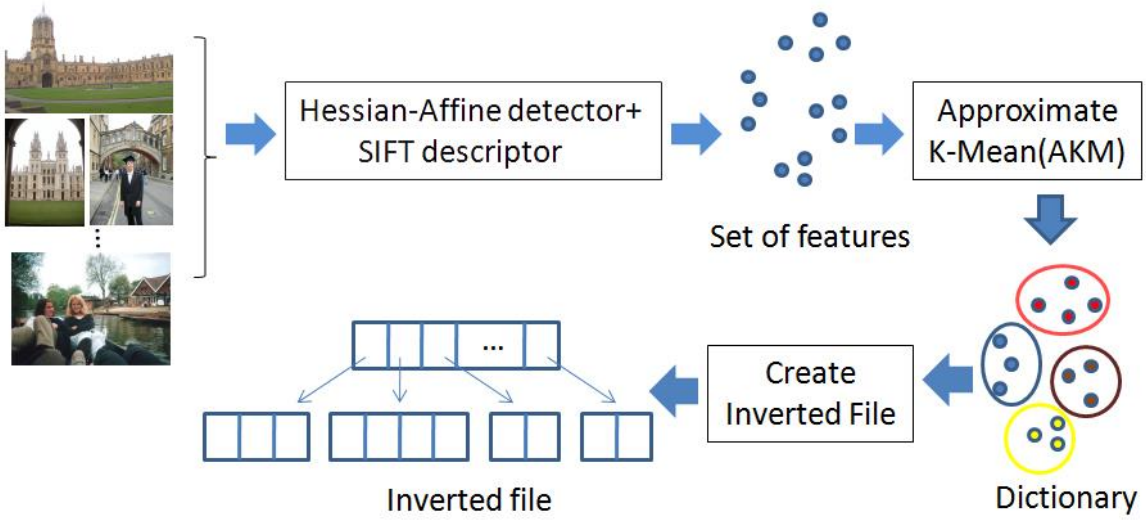
2.1. Sơ đồ thực hiện tổng quan



Mô tả: Input vào hệ thống website V, hệ thống V sẽ gửi thông tin và hệ thống IR, IR sẽ rút trích đặc trưng sau đó so khớp với các đặc trưng đã được lưu trữ ở bước training. Hệ thống truy vấn IR sẽ trả kết quả các ảnh có độ tương đồng được sắp xếp từ lớn đến nhỏ và hiển thị lên giao diện V.

2.2. Sơ đồ chi tiết các bước của hệ thống truy vấn IR

2.2.1. Training dataset



Bước 1: Rút trích đặc trưng

- Sử dụng Hessian-Affine region detector để rút trích các keypoint.
- Tính đặc trưng SIFT trên các keypoint.
- Số chiều đặc trưng: 128.

Bước 2: Xây dựng Visual Words (dictionary)

- Sử dụng thuật toán gom cụm Approximate K-Mean (AKM).
- Số lượng cluster: 1.000.000.
- Số lượng k-d tree: 8.
- Số lần lặp: 5.

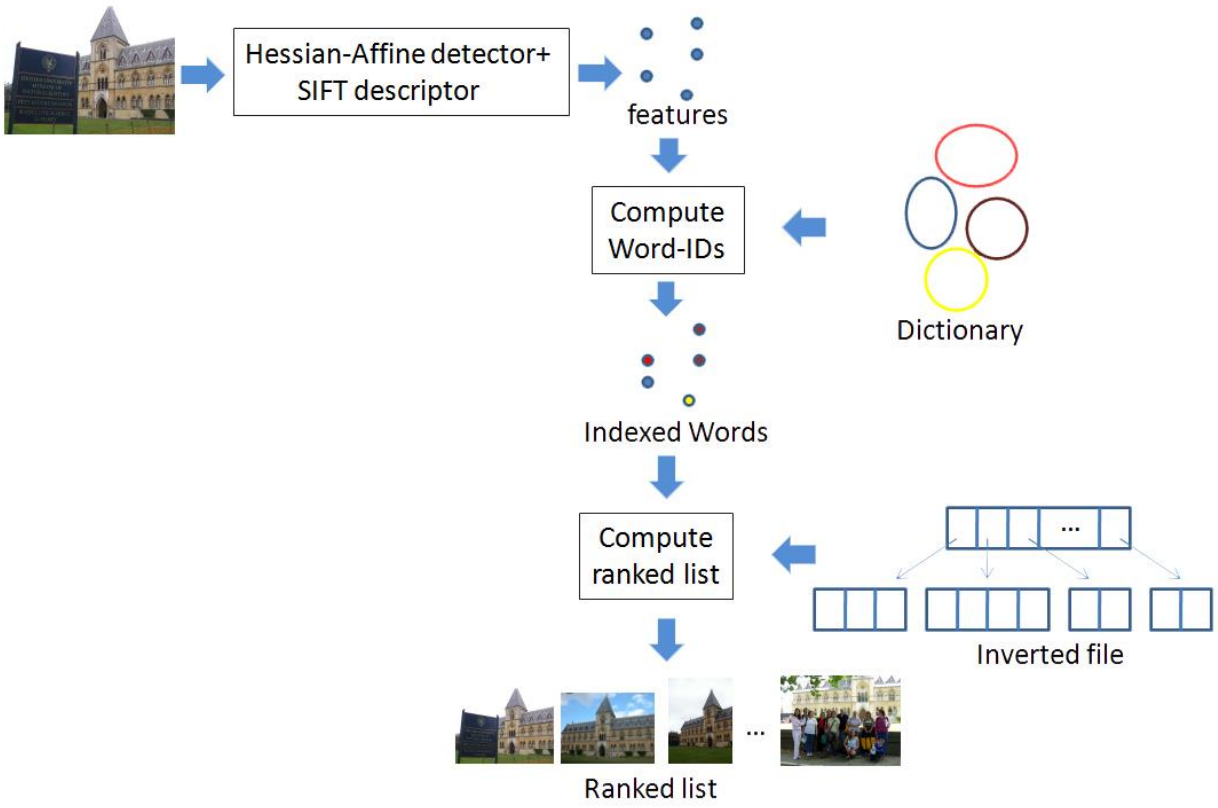
Bước 3: Tính Word-ID cho từng ảnh trong dataset

- Ở bước này ta sẽ chuyển các khái niệm của xử lý ảnh sang bài toán xử lý văn bản hay ngôn ngữ tự nhiên:
  + visual words →dictionary
  + feature →word
  + index của feature →word ID
  + ảnh → documents
- Với mỗi word trong từng document, ta tìm Word-ID của word dựa trên dictionary đã xây dựng ở bước 2.

Bước 4: Xây dựng Inverted file

- Theo thứ tự bình thường, với mỗi document, ta sẽ biết được trong document này có các word nào.
- Inverted file: với mỗi word, ta sẽ lưu danh sách những document có chứa nó, trọng số ‘tf-idf’: các visual word xuất hiện ở nhiều lớp của document thì càng ít có vai trò để phân loại một document nên được đánh trọng số thấp hơn. Các visual word xuất hiện ở càng ít ở các lớp của document thì có trọng số cao hơn.

2.2.2. Truy vấn ảnh



Bước 5: phát hiện và rút trích đặc trưng SIFT sử dụng Hessian-Affine region detector (tương tự bước 1).

Bước 6: tính Word-ID cho từng feature trong ảnh query (tương tự bước 3).

Bước 7: Tính ranked list

- Sử dụng inverted file để so sánh query document với tất cả các document trong trong inverted file →list score distance.
- Sắp xếp list score theo tự giảm dần.

2.3. Xây dựng hệ thống

2.3.1. Xây dựng trên Matlab

- **Training dataset:**

Bước 1: Rút trích đặc trưng và lưu thành file feature.bin

```
%% Load dac trung SIFT
        if ~exist('oxford\feat\5k\feature.bin', 'file')
        features = zeros(128, 2000000);
        nfeat = 0;
        files = dir(fullfile(datasetDir, '*.jpg'));
        nfiles = length(files);
        features_per_image = zeros(1,nfiles);
        for i=1:nfiles
            fprintf('Extracting features %d/%d\n', i, nfiles);
            imgPath = strcat(datasetDir, files(i).name);
            I = im2single(rgb2gray(imread(imgPath)));
            I = imresize(I, 0.6);
            [frame, sift] = vl_covdet(I, 'method', 'Hessian',
'estimateAffineShape', true);
            if nfeat+size(sift,2) > size(features,2)
                features = [features zeros(128,1000000)];
            end
            features(:,nfeat+1:nfeat+size(sift,2)) = sift;
            nfeat = nfeat+size(sift,2);
            features_per_image(i) = size(sift, 2);
        end
        features = features(:,1:nfeat);
        fid = fopen('oxford\feat\5k\feature.bin', 'w');
        fwrite(fid, features, 'float');
        fclose(fid);
        save('oxford\feat\5k\feat_info.mat', 'features_per_image',
'files');
        end
        fprintf('Loading SIFT features:\n');
        file = dir('oxford\feat\5k\feature.bin');
        fid = fopen('oxford\feat\5k\feature.bin', 'r');
        fclose(fid);
        load('oxford\feat\5k\feat_info.mat');
```

Bước 2: Sử dụng thuật toán gom cụm Approximate K-Mean (AKM) để xây dựng dictionary và lưu thành file dict.mat

```
%% Chay AKM de xay dung tu dien
    num_images = length(files);
    dict_params =  {num_iterations, 'kdt', num_trees};
    % build the dictionary
    if exist('oxford\feat\5k\dict.mat', 'file')
        load('oxford\feat\5k\dict.mat');
```

```
    else
        randIndex = randperm(size(features,2));
        dict_words = ccvBowGetDict(features(:,randIndex(1:100000)),
[], [], num_words, 'flat', 'akmeans', ...
            [], dict_params);
        save('oxford\feat\5k\dict.mat', 'dict_words');
    end
```

Bước 3: Tính Word-ID cho từng ảnh trong dataset và lưu thành file words.mat

```
    dict = ccvBowGetWordsInit(dict_words, 'flat', 'akmeans', [],
dict_params);
    if exist('oxford\feat\5k\words.mat', 'file')
        load('oxford\feat\5k\words.mat');
    else
        words = cell(1, num_images);
        for i=1:num_images
            if i==1
                bIndex = 1;
            else
                bIndex = sum(features_per_image(1:i-1))+1;
            end
            eIndex = bIndex + features_per_image(i)-1;
            words{i} = ccvBowGetWords(dict_words, features(:,
bIndex:eIndex), [], dict);
        end;
        save('oxford\feat\5k\words.mat', 'words');
    end
```

Bước 4: Xây dựng Inverted file

```
% Tao inverted file for the images
fprintf('Creating and searching an inverted file\n');
inv_file = ccvInvFileInsert([], words, num_words);
ccvInvFileCompStats(inv_file, if_weight, if_norm);
```

- **Truy vấn ảnh:**

Bước 5: truyền ảnh truy vấn (ảnh truy vấn là 1 trong 55 file trong folder **queryImg** được xây dựng dựa trên thông tin tập groundtruth của The Oxford Buildings Dataset), phát hiện và rút trích đặc trưng SIFT sử dụng Hessian-Affine region detector, tham số k được truyền vào là thứ tự của ảnh query trong tập groundtruth.

```
%% Query images
    k = str2num(strtok(filename,'.jpg'));
```

```
    q_files = dir(fullfile('oxford\groundtruth', '*query.txt'));
    ntop = 0;
    % load anh query
    fid = fopen(strcat('oxford\groundtruth\', q_files(k).name), 'r');
    str = fgetl(fid);
    [image_name, remain] = strtok(str, ' ');
    fclose(fid);
    numbers = str2num(remain);
    x1 = numbers(1);
    y1 = numbers(2);
    x2 = numbers(3);
    y2 = numbers(4);
    file = strcat('oxford\images\', image_name(6:end), '.jpg');
    I = im2single(rgb2gray(imread(file)));
    % Tinh dac trung SIFT cho anh query
    [frame, sift] = vl_covdet(I, 'method', 'Hessian',
'estimateAffineShape', true);
    sift = sift(:,(frame(1,:)<=x2) &  (frame(1,:) >= x1) & (frame(2,:)
<= y2) & (frame(2,:) >= y1));
```

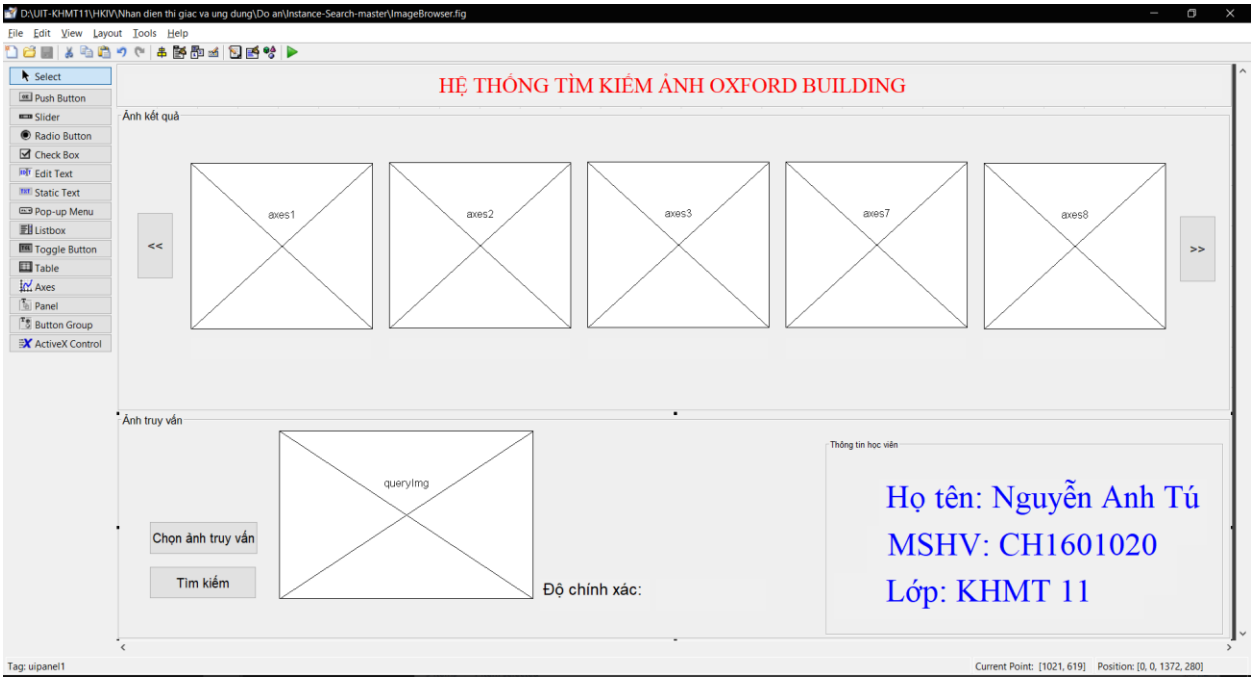
Bước 6: tính Word-ID cho feature trong ảnh query

```
% Tinh Word-ID
q_words = cell(1,1);
    q_words{1} = ccvBowGetWords(dict_words, double(sift), [], dict);
    [ids dists] = ccvInvFileSearch(inv_file, q_words(1), if_weight,
if_norm, if_dist, ntop);
```

Bước 7: Tính ranked list (các file *.txt trong thư mục result sẽ là độ chính xác của các ảnh truy vấn) và trả về danh sách (`listResultImg`) các ảnh có độ tương tự từ cao đến thấp với ảnh truy vấn.

```
script = ['oxford\groundtruth\Test.exe oxford\groundtruth\', ...
        q_files(k).name(1:end-10), ...
        ' oxford\groundtruth\rank_list.txt',...
        ' >oxford\result\', image_name(6:end), '_result.txt'];
    system(script);
    r_files = dir(fullfile('oxford\result\', '*.txt'));
    acc = [];
    for i=1:length(r_files)
        file = ['oxford\result\' r_files(i).name];
        fid = fopen(file, 'r');
        acc = [acc fscanf(fid, '%f')];
        fclose(fid);
    end
    set(handles.txtAcc,'String',num2str(mean(acc)*100,'%g%%'));%Hien
thi do chinh xac
```

2.3.2. Xây dựng giao diện trên Matlab



- Xử lý nút mở ảnh truy vấn:

```
% Xu ly nut load anh
function btnImgBrowser_Callback(hObject, eventdata, handles)
% hObject    handle to btnImgBrowser (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global filename;
[filename pathname] = uigetfile({'*.jpg';'*.bmp';'*.png'},'File
Selector');
image = strcat(pathname, filename);
axes(handles.queryImg);
imshow(image);
```

- Xử lý nút tìm kiếm ảnh:

```
% Xu ly nut tim kiem
function btnSearch_Callback(hObject, eventdata, handles)
    %% Khai bao bien
   global files;
   global ids;
   global img;
    addpath('AKM');
    run('vlfeat\toolbox\vl_setup.m');
    datasetDir = 'oxford\images\';
    num_words = 10000000;
    num_iterations = 5;
```

```
    num_trees = 8;
    dim = 128;
    if_weight = 'tfidf';    if_norm = 'l1';
    if_dist = 'l1';
    global filename;
    %% Load dac trung SIFT
        if ~exist('oxford\feat\5k\feature.bin', 'file')
        features = zeros(128, 2000000);
        nfeat = 0;
        files = dir(fullfile(datasetDir, '*.jpg'));
        nfiles = length(files);
        features_per_image = zeros(1,nfiles);
        for i=1:nfiles
            fprintf('Extracting features %d/%d\n', i, nfiles);
            imgPath = strcat(datasetDir, files(i).name);
            I = im2single(rgb2gray(imread(imgPath)));
            I = imresize(I, 0.6);
            [frame, sift] = vl_covdet(I, 'method', 'Hessian',
'estimateAffineShape', true);
            if nfeat+size(sift,2) > size(features,2)
                features = [features zeros(128,1000000)];
            end
            features(:,nfeat+1:nfeat+size(sift,2)) = sift;
            nfeat = nfeat+size(sift,2);
            features_per_image(i) = size(sift, 2);
        end
        features = features(:,1:nfeat);
        fid = fopen('oxford\feat\5k\feature.bin', 'w');
        fwrite(fid, features, 'float');
        fclose(fid);
        save('oxford\feat\5k\feat_info.mat', 'features_per_image',
'files');
        end
        fprintf('Loading SIFT features:\n');
        file = dir('oxford\feat\5k\feature.bin');
        fid = fopen('oxford\feat\5k\feature.bin', 'r');
        fclose(fid);
        load('oxford\feat\5k\feat_info.mat');
    %% Chay AKM de xay dung tu dien
    num_images = length(files);
    dict_params =  {num_iterations, 'kdt', num_trees};
    % build the dictionary
    if exist('oxford\feat\5k\dict.mat', 'file')
        load('oxford\feat\5k\dict.mat');
    else
        randIndex = randperm(size(features,2));
        dict_words = ccvBowGetDict(features(:,randIndex(1:100000)),
[], [], num_words, 'flat', 'akmeans', ...
            [], dict_params);
        save('oxford\feat\5k\dict.mat', 'dict_words');
    end
    %% Tinh toan sparse frequency vector
```

```
    dict = ccvBowGetWordsInit(dict_words, 'flat', 'akmeans', [],
dict_params);
    if exist('oxford\feat\5k\words.mat', 'file')
        load('oxford\feat\5k\words.mat');
    else
        words = cell(1, num_images);
        for i=1:num_images
            if i==1
                bIndex = 1;
            else
                bIndex = sum(features_per_image(1:i-1))+1;
            end
            eIndex = bIndex + features_per_image(i)-1;
            words{i} = ccvBowGetWords(dict_words, features(:,
bIndex:eIndex), [], dict);
        end;
        save('oxford\feat\5k\words.mat', 'words');
    end

    %% Tao inverted file for the images
    inv_file = ccvInvFileInsert([], words, num_words);
    ccvInvFileCompStats(inv_file, if_weight, if_norm);
    %% Query images
    k = str2num(strtok(filename,'.jpg'));
    q_files = dir(fullfile('oxford\groundtruth', '*query.txt'));
    ntop = 0;
    % load anh query
    fid = fopen(strcat('oxford\groundtruth\', q_files(k).name), 'r');
    str = fgetl(fid);
    [image_name, remain] = strtok(str, ' ');
    fclose(fid);
    numbers = str2num(remain);
    x1 = numbers(1);
    y1 = numbers(2);
    x2 = numbers(3);
    y2 = numbers(4);
    file = strcat('oxford\images\', image_name(6:end), '.jpg');
    I = im2single(rgb2gray(imread(file)));
    % Tinh dac trung SIFT cho anh query
    [frame, sift] = vl_covdet(I, 'method', 'Hessian',
'estimateAffineShape', true);
    sift = sift(:,(frame(1,:)<=x2) &  (frame(1,:) >= x1) & (frame(2,:)
<= y2) & (frame(2,:) >= y1));

    % Tinh Word-ID
    q_words = cell(1,1);
    q_words{1} = ccvBowGetWords(dict_words, double(sift), [], dict);
    [ids dists] = ccvInvFileSearch(inv_file, q_words(1), if_weight,
if_norm, if_dist, ntop);
    fid = fopen('oxford\groundtruth\rank_list.txt', 'w');
    img = 1;
              axes(handles.axes1);
```

```
                imshow(imread(fullfile('oxford\images\',
files(ids{1}(img)).name)));

set(handles.txtImg1,'String',num2str(files(ids{1}(img)).name));
                axes(handles.axes2);
                imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+1)).name)));
                set(handles.txtImg2,'String',files(ids{1}(img+1)).name);
                axes(handles.axes3);
                imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+2)).name)));
                set(handles.txtImg3,'String',files(ids{1}(img+2)).name);
                axes(handles.axes7);
                imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+3)).name)));
                set(handles.txtImg4,'String',files(ids{1}(img+3)).name);
                axes(handles.axes8);
                imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+4)).name)));
                set(handles.txtImg5,'String',files(ids{1}(img+4)).name);
                if(size(ids{1},2)<=5)
                   set(handles.btnNext,'Enable','off') ;
                end
                   set(handles.btnBack,'Enable','off') ;
    for i=1:size(ids{1},2)
        fprintf(fid, '%s\n', files(ids{1}(i)).name(1:end-4));
    end
    fclose(fid);
    script = ['oxford\groundtruth\Test.exe oxford\groundtruth\', ...
        q_files(k).name(1:end-10), ...
        ' oxford\groundtruth\rank_list.txt',...
        ' >oxford\result\', image_name(6:end), '_result.txt'];
    system(script);
    r_files = dir(fullfile('oxford\result\', '*.txt'));
    acc = [];
    for i=1:length(r_files)
        file = ['oxford\result\' r_files(i).name];
        fid = fopen(file, 'r');
        acc = [acc fscanf(fid, '%f')];
        fclose(fid);
    end
    set(handles.txtAcc,'String',num2str(mean(acc)*100,'%g%%'));%Hien
thi do chinh xac
% clear inv file
ccvInvFileClean(inv_file);


% Xu ly nut lui
function btnBack_Callback(hObject, eventdata, handles)
% hObject    handle to btnBack (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
   global files;
   global ids;
   global img;

set(handles.btnNext,'Enable','on') ;
img=img-5;
axes(handles.axes1);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img)).name)));
set(handles.txtImg1,'String',files(ids{1}(img)).name);

axes(handles.axes2);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+1)).name)));
set(handles.txtImg2,'String',files(ids{1}(img+1)).name);

axes(handles.axes3);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+2)).name)));
set(handles.txtImg3,'String',files(ids{1}(img+2)).name);

axes(handles.axes7);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+3)).name)));
set(handles.txtImg4,'String',files(ids{1}(img+3)).name);

axes(handles.axes8);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+4)).name)));
set(handles.txtImg5,'String',files(ids{1}(img+4)).name);

if(img==1)
    set(handles.btnBack,'Enable','off') ;
end
```

- Xử lý nút Next (mở tiếp 5 ảnh kết quả)

```
function btnNext_Callback(hObject, eventdata, handles)
% hObject    handle to btnNext (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
   global files;
   global ids;
   global img;

set(handles.btnBack,'Enable','on') ;

img=img+5;
axes(handles.axes1);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img)).name)));
set(handles.txtImg1,'String',files(ids{1}(img)).name);

if(img+1<size(ids{1},2))
    axes(handles.axes2);
    imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+1)).name)));
```

```
    set(handles.txtImg2,'String',files(ids{1}(img+1)).name);
else
    cla(handles.axes2);
    set(handles.btnNext,'Enable','off') ;
end

if(img+2<size(ids{1},2))
    axes(handles.axes3);
    imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+2)).name)));
    set(handles.txtImg3,'String',files(ids{1}(img+2)).name);
else
    cla(handles.axes3);
    set(handles.btnNext,'Enable','off') ;
end

if(img+3<size(ids{1},2))
    axes(handles.axes7);
    imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+3)).name)));
    set(handles.txtImg4,'String',files(ids{1}(img+3)).name);
else
    cla(handles.axes7);
    set(handles.btnNext,'Enable','off') ;
end

if(img+4<size(ids{1},2))
    axes(handles.axes8);
    imshow(imread(fullfile('oxford\images\',
files(ids{1}(img+4)).name)));
    set(handles.txtImg5,'String',files(ids{1}(img+5)).name);
else
    cla(handles.axes8);
    set(handles.btnNext,'Enable','off') ;
end
```

- Xử lý nút Back (lùi lại 5 ảnh kết quả)

```
% Xu ly nut lui
function btnBack_Callback(hObject, eventdata, handles)
% hObject    handle to btnBack (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
   global files;
   global ids;
   global img;

set(handles.btnNext,'Enable','on') ;
img=img-5;
axes(handles.axes1);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img)).name)));
```

```
set(handles.txtImg1,'String',files(ids{1}(img)).name);

axes(handles.axes2);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+1)).name)));
set(handles.txtImg2,'String',files(ids{1}(img+1)).name);

axes(handles.axes3);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+2)).name)));
set(handles.txtImg3,'String',files(ids{1}(img+2)).name);

axes(handles.axes7);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+3)).name)));
set(handles.txtImg4,'String',files(ids{1}(img+3)).name);

axes(handles.axes8);
imshow(imread(fullfile('oxford\images\', files(ids{1}(img+4)).name)));
set(handles.txtImg5,'String',files(ids{1}(img+4)).name);

if(img==1)
    set(handles.btnBack,'Enable','off') ;
end
```

CHƯƠNG 3: CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM

3.1. Hướng dẫn cài đặt

- Tải project theo đường dẫn https://github.com/pleasecall113/VRA.NguyenAnhTu/tree/master/VRA.Final.NguyenAnhTu/ImageRetrieval
- Truy cập đường dẫn https://drive.google.com/drive/u/1/folders/17ZTTyshUIrBTnZwZ73CPzqXtNDkQfLjS và tải, cài đặt các file theo hướng dẫn:

  - Copy file feature.bin vào …\ImageRetrieval\oxford\feat\5k (nếu không tải có thể chạy code để build nhưng mất thời gian khá lâu).

  - Giải nén file images.rar vào …\ImageRetrieval\oxford\images.

- Demo https://youtu.be/Ziw5u6D56js

3.2. Kết quả thực nghiệm

- Độ chính xác 55 ảnh truy vấn

STT	Ảnh truy vấn	Độ chính xác (%)
1	all_souls_000013	13.2353
2	all_souls_000026	24.1878
3	all_souls_000051	70.2383
4	ashmolean_000000	65.5927
5	ashmolean_000007	35.8484
6	ashmolean_000058	13.4377
7	ashmolean_000269	38.4492
8	ashmolean_000305	41.355
9	balliol_000051	55.1061
10	balliol_000167	9.25049
11	balliol_000187	27.5947

12	balliol_000194	52.5821
13	bodleian_000107	6.48962
14	bodleian_000108	45.3897
15	bodleian_000132	75.1373
16	bodleian_000163	30.9107
17	bodleian_000407	1.95648
18	christ_church_000179	65.0525
19	christ_church_000999	42.1939
20	christ_church_001020	57.6323
21	cornmarket_000019	58.6491
22	cornmarket_000047	67.7209
23	cornmarket_000105	34.1662
24	cornmarket_000131	41.7364
25	hertford_000015	40.3121
26	hertford_000027	47.7073
27	hertford_000063	60.6072
28	keble_000028	29.9738
29	keble_000055	41.6912
30	keble_000214	83.6021
31	keble_000227	73.1399
32	keble_000245	83.5473
33	magdalen_000058	7.83035
34	magdalen_000078	2.88389

35	magdalen_000560	9.28085
36	oxford_000317	11.4865
37	oxford_000545	30.599
38	oxford_001115	2.04692
39	oxford_001752	51.1139
40	oxford_001753	46.2445
41	oxford_002416	53.4233
42	oxford_002562	8.37262
43	oxford_002734	39.434
44	oxford_002904	44.6879
45	oxford_002985	29.1903
46	oxford_003335	8.57126
47	oxford_003410	28.7112
48	pitt_rivers_000033	17.3579
49	pitt_rivers_000058	67.0156
50	pitt_rivers_000087	54.0919
51	pitt_rivers_000119	69.5235
52	pitt_rivers_000153	24.1262
53	radcliffe_camera_000095	43.8671
54	radcliffe_camera_000519	53.7545
55	radcliffe_camera_000523	72.5166
Trung bình		**40.19316**

-

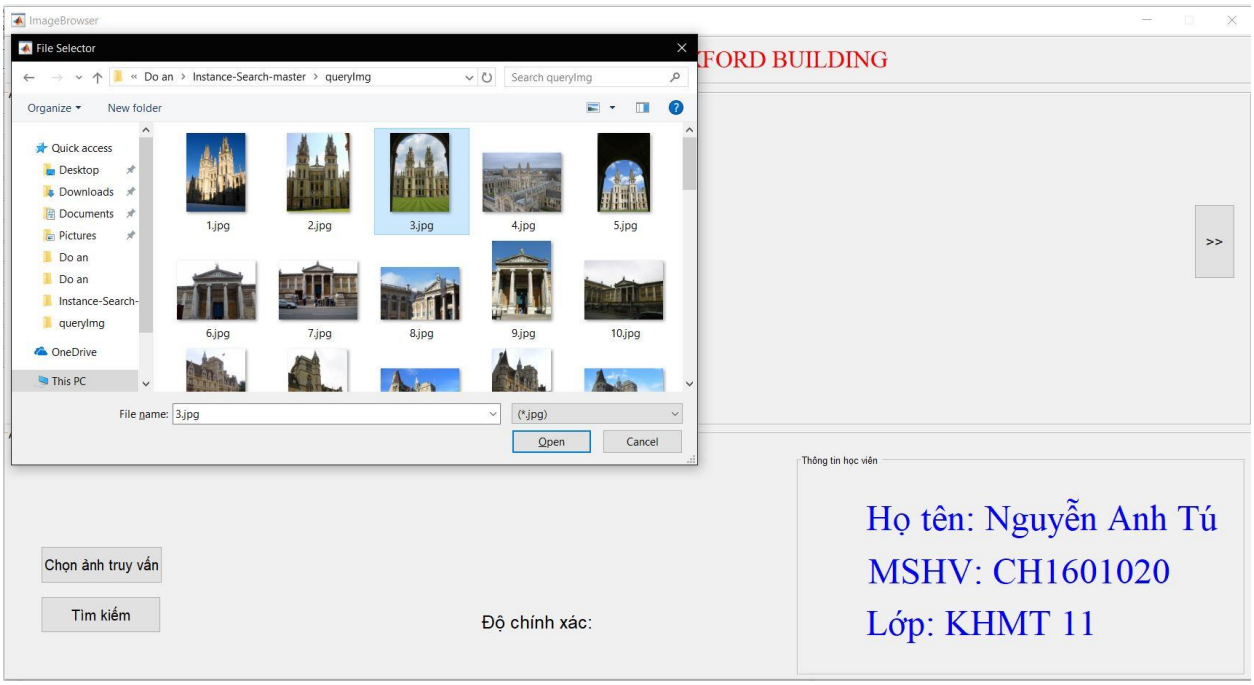
- 10 ảnh truy vấn có độ chính xác cao nhất:

STT	Tên ảnh	Độ chính xác (%)
1	keble_000214	83.6021
2	keble_000245	83.5473
3	bodleian_000132	75.1373
4	keble_000227	73.1399
5	radcliffe_camera_000523	72.5166
6	all_souls_000051	70.2383
7	pitt_rivers_000119	69.5235
8	cornmarket_000047	67.7209
9	pitt_rivers_000058	67.0156
10	ashmolean_000000	65.5927

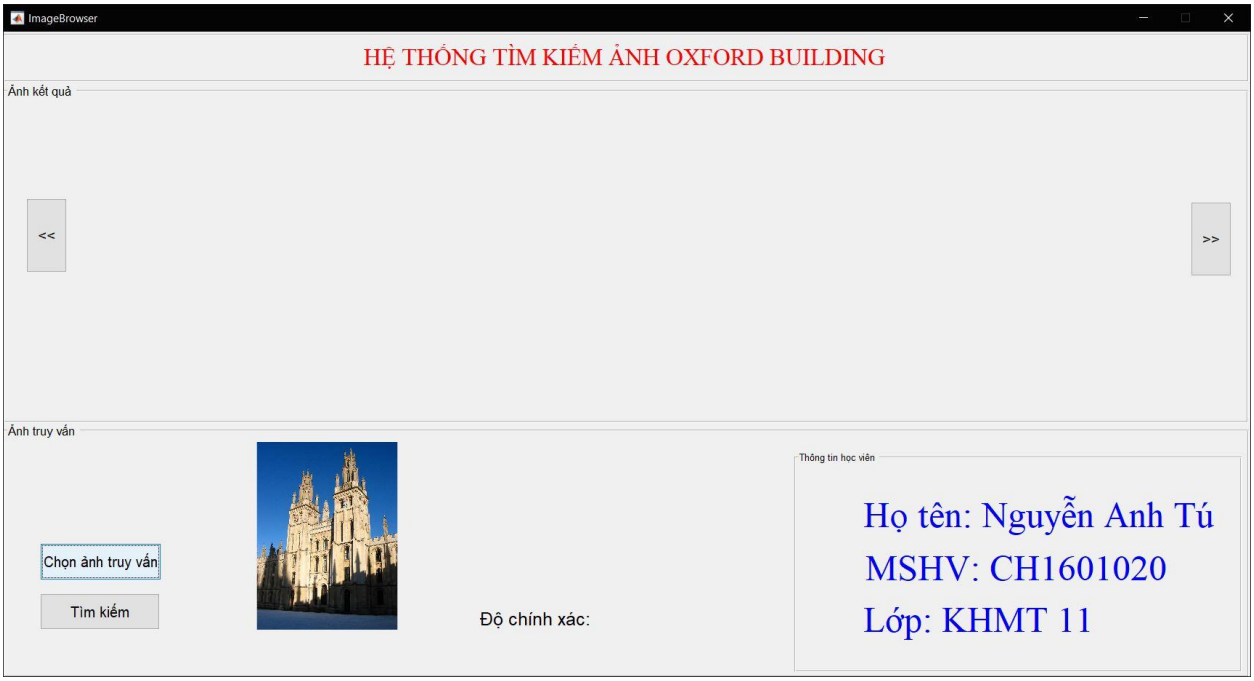
- 10 ảnh truy vấn có độ chính xác thấp nhất:

STT	Tên ảnh	Độ chính xác (%)
1	oxford_000317	11.4865
2	magdalen_000560	9.28085
3	balliol_000167	9.25049
4	oxford_003335	8.57126
5	oxford_002562	8.37262
6	magdalen_000058	7.83035
7	bodleian_000107	6.48962
8	magdalen_000078	2.88389
9	oxford_001115	2.04692
10	bodleian_000407	1.95648

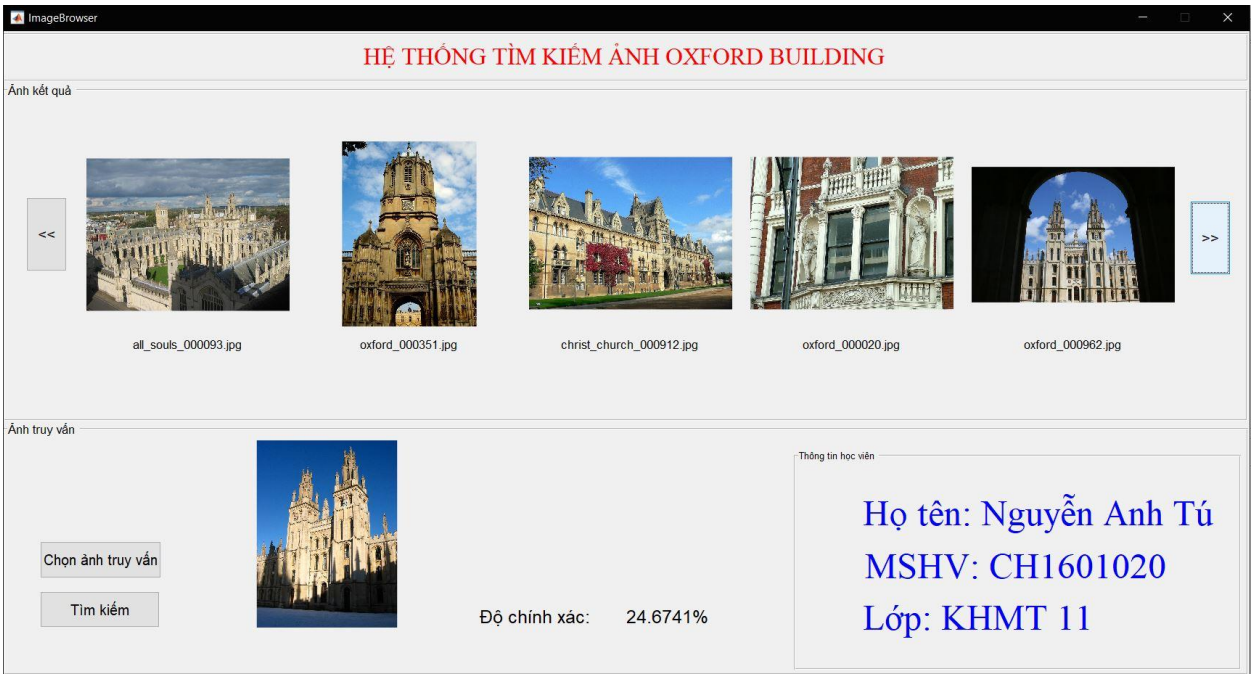
- Một số hình ảnh giao diện:



Chọn ảnh từ thư mục queryImg



Load ảnh lên



Kết quả truy vấn ảnh

CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

- Xây dựng, phân tích được hệ thống truy vấn hình ảnh dựa trên tập dataset Oxford Building 5k.
- Trả về kết quả các ảnh có độ tương tự từ cao đến thấp của một ảnh truy vấn.
- Xây dựng giao hiển thị hình ảnh.
- Tính độ chính xác của ảnh truy vấn.

4.2. Hướng phát triển

- Tìm kiếm ảnh bằng cách chọn một vùng trên ảnh truy vấn.
- Sử dụng đặc trưng rootSIFT để tăng độ chính xác so với đặc trưng SIFT.
- Cần cập nhật hình ảnh vào để hệ thống IR có thể học được.
- Cải tiến các thuật toán để thực hiện nhanh hơn.
- Xây dựng hệ thống Web để hiển thị.
- Đánh giá kết quả với các câu query được cung cấp trong bộ dữ liệu Oxford Building, so sánh với kết quả của các phương pháp khác đã công bố.

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] Bài giảng môn Nhận dạng thị giác và ứng dụng, TS. Lê Đình Duy, TS. Nguyễn Tấn Trần Minh Khang.

Tiếng Anh

[2] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/

Tài liệu Website

[3] https://github.com/nvtiep/Instance-Search

[4] http://www.vlfeat.org/benchmarks/overview/retrieval.html

[5] http://viet.wordnet.vn/wnms