

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
NGÀNH KHOA HỌC MÁY TÍNH



Báo cáo đồ án môn học

NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

Đề tài:

HỆ THỐNG TÌM KIẾM ẢNH SỬ DỤNG PHƯƠNG PHÁP BAG OF WORDS

Thực hiện:

Nguyễn Mạnh Phúc – CH1601033

GVHD:

TS. Lê Đình Duy

TS. Nguyễn Tấn Trần Minh Khang

Tháng 12/2017

Mục lục

1	Mục tiêu và nội dung đồ án.....	2
1.1	Mục tiêu	2
1.2	Nội dung đồ án	2
2	Harris affine region detector và SIFT descriptor	3
2.1	Harris affine region detector	3
2.1.1	Phép biến đổi Affine	3
2.1.2	Harris-Laplace Detector	4
2.1.3	Harris affine region detector	5
2.2	SIFT descriptor.....	5
2.2.1	SIFT Descriptor (Scale Invariant Feature Transform)	5
2.2.2	rootSIFT descriptor.....	7
3	Cách thực hiện chương trình	7
3.1	Giai đoạn huấn luyện	7
3.2	Giai đoạn Tìm kiếm ảnh	8
3.3	Tìm kiếm ảnh từ giao diện	9
3.4	Tạo file dll từ Matlab và gọi hàm từ C# (ASP.Net).....	9
4	Kết quả đánh giá	10
4.1	Một số hình ảnh kết quả Truy vấn:.....	10
4.2	Kết quả thực hiện.....	14
4.2.1	Kết quả sử dụng Harris affine region detector kết hợp với rootSIFT descriptor	14
4.2.2	Kết quả sử dụng Harris affine region detector kết hợp với SIFT descriptor	15
4.2.3	Kết quả sử dụng Hessian affine detector kết hợp với rootSIFT descriptor	17
4.2.4	Kết quả sử dụng Hessian affine detector kết hợp với SIFT descriptor	18
4.2.5	Kết quả sử dụng SIFT detector kết hợp với rootSIFT descriptor.....	19
4.2.6	Kết quả sử dụng SURF features	21
5	Lưu ý khi chạy chương trình.	22
5.1	Runtime.....	22
5.2	Các tập tin cần thiết	23
6	Tài liệu tham khảo	23

1 Mục tiêu và nội dung đồ án

1.1 Mục tiêu

Bài toán tìm kiếm ảnh dựa trên là một trong những bài quan trọng và được nghiên cứu rộng rãi.

Mục tiêu của đồ án này là:

- Vận dụng, ôn tập kiến thức môn học Nhận dạng thị giác và ứng dụng
- Vận dụng các công cụ, thư viện để xây dựng một ứng dụng cho phép tìm kiếm ảnh dựa trên nội dung của ảnh sử dụng phương pháp bag of words.
- Rèn luyện khả năng xây dựng một ứng dụng sử dụng Matlab
- Tìm hiểu chi tiết về Harris affine region detector và cách sử dụng detector để phát hiện ra các keypoints và biểu diễn các keypoints này sử dụng SIFT descriptor

1.2 Nội dung đồ án

Nội dung đồ án này là xây dựng một ứng dụng tìm kiếm ảnh dựa trên các phương pháp bag of words sử dụng Matlab, thư viện vlfeat, thư viện AKM và ASP.Net qua hai giai đoạn và training (trên Matlab) và tìm kiếm ảnh (sử dụng ASP.Net). Ngoài ra, trong đồ án này, em còn chạy thử training và tìm kiếm ảnh sử dụng sử dụng đặc trưng SURF được cung cấp bởi Matlab. Code phần training trên Matlab có tham khảo từ [2] và được thay đổi lại cho phù hợp với nhu cầu. Ứng dụng được giới thiệu tại: <https://youtu.be/QXQVEITOV1Y>

Đồ án sử dụng bộ dữ liệu oxford building (<http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>) để chạy thử và đánh giá hệ thống thống qua 55 truy vấn được cung cấp sẵn và so sánh hệ thống khi sử dụng các loại detector khác nhau cũng như sử dụng 2 loại descriptor là SIFT và rootSIFT.

Đồ án này còn trình bày nội dung về phương pháp phát hiện các key points Harris affine region detector và SIFT descriptor cũng như phiên bản rootSIFT descriptor

2 Harris affine region detector và SIFT descriptor

2.1 Harris affine region detector

2.1.1 Phép biến đổi Affine

Phép biến đổi Affine là phép biến đổi tọa độ điểm đặc trưng của đối tượng thành tập tương ứng các điểm mới để tạo ra các hiệu ứng cho toàn đối tượng

Ví dụ: phép biến đổi tọa độ với chỉ 2 điểm đầu cuối của đoạn thẳng tạo thành 2 điểm mới mà khi nối chúng với nhau tạo thành đoạn thẳng mới Một số phép biến đổi affine cơ bản là:

- Phép bất biến
- Phép biến đổi tỉ lệ - Scaling
- Phép biến dạng
- Phép xoay

$$\text{Phép tịnh tiến : } M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{Phép quay : } M = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$$

$$\text{Phép biến đổi tỉ lệ : } M = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

$$\text{Phép biến dạng : } M = \begin{pmatrix} 1 & g \\ h & 1 \end{pmatrix}$$

Hình 1. Các phép biến đổi affine transformation

Phép biến đổi affine có một số tính chất sau:

- Bảo tồn đường thẳng
- Bảo tồn sự song song
- Bảo tồn tỉ lệ khoảng cách.

Các phép biến đổi affine có thể kết hợp lại với nhau.

2.1.2 Harris-Laplace Detector

Harris corner detector

Ý tưởng của harris corner detector là di chuyển một khung trong hình để tìm được những điểm có sự thay đổi lớn về cường độ ánh sáng (độ xám)

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$
$$E \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

Trong đó:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Trong đó I_x là đạo hàm của ảnh theo x , I_y là đạo hàm của ảnh theo y , w là khung trong số.

Khi đó, ta tìm các ảnh có $R = \det(M) - \alpha \times \text{trace}^2(M)$ lớn hơn một mức nào đó, điểm đó sẽ được coi là key point

Gaussian scale space

Biểu diễn Gaussian scale space của một ảnh là tập hợp các ảnh là tích chập của ảnh gốc với nhiều Gaussian kernel ở nhiều tỉ lệ scale khác nhau.

$$L(\mathbf{x}, s) = G(s) \otimes I(\mathbf{x})$$

Trong đó L là ảnh đã được làm trơn (smooth) với tỉ lệ s . $G(s)$ là một isotropic, circular Gaussian kernel

Harris-Laplace detector

Harris-Laplace detector gồm 2 giai đoạn:

1. Áp dụng Harris corner detector ở nhiều mức tỉ lệ khác nhau
2. Chọn một tỉ lệ phù hợp

Cụ thể, tương tự ma trận M trong Harris corner detector, ma trận M trong Harris-Laplace detector được tính như sau:

$$M = \mu(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) \otimes \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}$$

Trong đó là tỉ lệ scale của ảnh, $g(\sigma_I)$ là kernel tương ứng với tỉ lệ scale σ_I , x là điểm ảnh, $L(x)$ là điểm ảnh đã được làm trơn, $L_x(x, \sigma_D)$, $L_y(x, \sigma_D)$ là đạo hàm của ảnh tại x với ảnh đã làm trơn khi sử dụng Gaussian kernel σ_D . σ_I là tỉ lệ scale hiện tại mà key point đang được phát hiện (detect).

2.1.3 Harris affine region detector

Harris affine region detector hoạt động như sau:

1. Xác định các vùng ban đầu bằng Harris-Laplace Detector.
2. Với mỗi vùng ban đầu, áp dụng thuật toán affine shape adaptation
3. Tìm kiếm các key points, tỉ lệ integration và tỉ lệ differentiation
4. Cập nhật vùng affine region sử dụng các tỉ lệ và các key points
5. Lặp lại bước 3 cho đến khi điều kiện dừng thỏa mãn.

2.2 SIFT descriptor

2.2.1 SIFT Descriptor (Scale Invariant Feature Transform)

SIFT descriptor là một trong những descriptor được sử dụng phổ biến để thể hiện các key points. Ý tưởng chính của SIFT descriptor như sau. SIFT là:

- Lấy 16 x 16 điểm ảnh xung quanh key points và chia thành 4 x 4 block. Có tổng cộng 16 block.
- Mỗi block được mô tả bởi histogram theo 8 hướng (image gradients) theo chiều kim đồng hồ. Bắt đầu là hướng nằm ngang
- Sau đó, các histogram của mỗi block sẽ được nối lại với nhau theo hàng rồi theo cột để được 128 chiều.

2.2.2 *rootSIFT descriptor*

rootSIFT là một phiên bản cải tiến của SIFT descriptor. rootSIFT được tính bằng cách:

- Normalize SIFT feature bằng phương pháp L1: lấy mỗi phần tử trong vector chia cho tổng các phần tử
- Lấy căn bậc 2 mỗi phần tử sau khi được normalize
- Normalize kết quả bằng phương pháp L2.

3 Cách thực hiện chương trình

3.1 Giai đoạn huấn luyện

Quy trình thực hiện là:

Bước 1: Rút trích đặc trưng

- Sử dụng Harris affine region detector để rút trích các keypoint sử dụng thư viện vlfeat (hàm vl_codet)
- Tính đặc trưng rootSIFT trên các keypoint
- Số chiều đặc trưng: 128
- Số ảnh: 5063
- Do số lượng features khi để ảnh gốc lớn nên trong quá trình rút trích đặc trưng thì đã giảm thử rút trích trên ảnh có được thu nhỏ với tỉ lệ kích thước là 60% mỗi chiều (`imresize(I, 0.6)`).

Bước 2: Xây dựng Visual Words (dictionary)

- Sử dụng thuật toán gom cụm Approximate K-Mean (AKM) trong bộ thư viện vlfeat.
- Sử dụng các đặc trưng đã rút trích ở trên.
- Số lượng cluster (words): 50000
- Số lượng k-d tree: 8
- Số lần lặp: 5
- Mức độ tương tự của các words được tính bằng phương pháp l2 – khoảng cách euclidean

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

- Với mỗi đặc trưng trong từng ảnh, ta tìm Word-ID của word dựa trên dictionary (tập Visual Words) đã xây dựng ở bước 2

Bước 4: Xây dựng Inverted file

- Inverted file: với mỗi visual word, ta xác định các ảnh có chứa visual word đó theo một thứ tự nhất định.
- Các visual word có trọng số được tính theo phương pháp tf- idf

3.2 Giai đoạn Tìm kiếm ảnh

Bước 1. Rút trích đặc trưng rootSIFT của ảnh cần tìm kiếm sử dụng HarrisLaplace - Affine region detector

Bước 2. Tính word-id cho các đặc trưng đã tính ở trên

Bước 3. Tính ranked list

- Tạo bảng Word Count để thống kê các visual word và tần số xuất hiện tương ứng trong ảnh tìm kiếm sử dụng hàm `ccvBowGetWords`
- Sử dụng `InvertedFile` để tạo ra bảng ranked list (hàm `ccvInvFileSearch`).
- Mức độ tương tự của các words được tính bằng phương pháp l2 – khoảng cách euclidean

Bước 4: Tạo ra kết quả.

- Dựa trên ranked list, ta tìm được top các ảnh phù hợp với ảnh truy vấn.

3.3 Tìm kiếm ảnh từ giao diện

Bước 1: Chọn ảnh để tìm kiếm

Bước 2: Chọn một vùng trong ảnh để tìm kiếm, vùng không được chọn sẽ được làm tối đi

Bước 3: Hệ thống lấy ảnh được tải lên từ client và lưu lại ảnh với 1 tên duy nhất

Bước 4: Hệ thống sử dụng ảnh ở bước 3 để tìm kiếm

Bước 5. Hiện thị các ảnh kết quả (tối đa 1000 ảnh) có sử dụng phân trang (20 ảnh/trang)

3.4 Tạo file dll từ Matlab và gọi hàm từ C# (ASP.Net)

- Sử dụng chức năng library compiler của Matlab để tạo ra các DLL để có thể gọi hàm Matlab từ C#. Trong quá trình build thì xây dựng
- DLL được build từ project `Search1.1prj` và có thể tìm thấy trong thư mục `Search 1.1`, file này nằm tại thư mục “`Search1.1\for_redistribution_files_only`”

4 Kết quả đánh giá

4.1 Một số hình ảnh kết quả Truy vấn:



Các ảnh chụp bằng

Kết quả:



Truy vấn:



Kết quả:



Truy vấn:



Kết quả:



4.2 Kết quả thực hiện

Các kết quả này được thực hiện dựa trên các bộ query mẫu trong thư mục groundtruth. Kết quả này được so sánh với các phương pháp khác (dùng SIFT descriptor, Hessian-affine detector).

4.2.1 Kết quả sử dụng Harris affine region detector kết hợp với rootSIFT descriptor

STT	Truy vấn	Độ chính xác
1.	all_souls_000013	0.143988
2.	all_souls_000026	0.15261
3.	all_souls_000051	0.800409
4.	ashmolean_000000	0.788215
5.	ashmolean_000007	0.320357
6.	ashmolean_000058	0.0588467
7.	ashmolean_000269	0.107383
8.	ashmolean_000305	0.45956
9.	balliol_000051	0.695223
10.	balliol_000167	0.100426
11.	balliol_000187	0.287816
12.	balliol_000194	0.339162
13.	bodleian_000107	0.0839179
14.	bodleian_000108	0.308126
15.	bodleian_000132	0.807885
16.	bodleian_000163	0.176462
17.	bodleian_000407	0.0286483
18.	christ_church_000179	0.52488
19.	christ_church_000999	0.426322
20.	christ_church_001020	0.390618
21.	cornmarket_000019	0.621912
22.	cornmarket_000047	0.677707
23.	cornmarket_000105	0.174183
24.	cornmarket_000131	0.585086
25.	hertford_000015	0.650082
26.	hertford_000027	0.670043
27.	hertford_000063	0.777921
28.	keble_000028	0.842063
29.	keble_000055	0.469601
30.	keble_000214	0.464671
31.	keble_000227	0.602623
32.	keble_000245	0.488037
33.	magdalen_000058	0.0827481

34.	magdalen_000078	0.0430635
35.	magdalen_000560	0.0643641
36.	oxford_000317	0.277183
37.	oxford_000545	0.409813
38.	oxford_001115	0.0546525
39.	oxford_001752	0.649196
40.	oxford_001753	0.643672
41.	oxford_002416	0.373663
42.	oxford_002562	0.132747
43.	oxford_002734	0.480675
44.	oxford_002904	0.655721
45.	oxford_002985	0.308979
46.	oxford_003335	0.0687936
47.	oxford_003410	0.316208
48.	pitt_rivers_000033	0.217667
49.	pitt_rivers_000058	0.59976
50.	pitt_rivers_000087	0.676778
51.	pitt_rivers_000119	0.803102
52.	pitt_rivers_000153	0.197441
53.	radcliffe_camera_000095	0.442341
54.	radcliffe_camera_000519	0.715753
55.	radcliffe_camera_000523	0.795492
	all_souls_000013	0.4188

4.2.2 Kết quả sử dụng Harris affine region detector kết hợp với SIFT descriptor

STT	Truy vấn	Độ chính xác
1.	all_souls_000013	0.224167
2.	all_souls_000026	0.182582
3.	all_souls_000051	0.824138
4.	ashmolean_000000	0.650303
5.	ashmolean_000007	0.193022
6.	ashmolean_000058	0.0576353
7.	ashmolean_000269	0.07369
8.	ashmolean_000305	0.393625
9.	balliol_000051	0.442062
10.	balliol_000167	0.0970889
11.	balliol_000187	0.171442
12.	balliol_000194	0.38133
13.	bodleian_000107	0.0810723
14.	bodleian_000108	0.284904
15.	bodleian_000132	0.827323
16.	bodleian_000163	0.130509

17.	bodleian_000407	0.047382
18.	christ_church_000179	0.471288
19.	christ_church_000999	0.456355
20.	christ_church_001020	0.275744
21.	cornmarket_000019	0.645183
22.	cornmarket_000047	0.690008
23.	cornmarket_000105	0.217892
24.	cornmarket_000131	0.516173
25.	hertford_000015	0.627591
26.	hertford_000027	0.638338
27.	hertford_000063	0.715333
28.	keble_000028	0.719866
29.	keble_000055	0.354188
30.	keble_000214	0.745299
31.	keble_000227	0.473019
32.	keble_000245	0.623046
33.	magdalen_000058	0.0720953
34.	magdalen_000078	0.0225745
35.	magdalen_000560	0.0623021
36.	oxford_000317	0.228505
37.	oxford_000545	0.349756
38.	oxford_001115	0.0511752
39.	oxford_001752	0.578468
40.	oxford_001753	0.481105
41.	oxford_002416	0.330567
42.	oxford_002562	0.0611148
43.	oxford_002734	0.376128
44.	oxford_002904	0.641943
45.	oxford_002985	0.286051
46.	oxford_003335	0.0717509
47.	oxford_003410	0.341288
48.	pitt_rivers_000033	0.281268
49.	pitt_rivers_000058	0.657479
50.	pitt_rivers_000087	0.686725
51.	pitt_rivers_000119	0.783144
52.	pitt_rivers_000153	0.371431
53.	radcliffe_camera_000095	0.409401
54.	radcliffe_camera_000519	0.72853
55.	radcliffe_camera_000523	0.761755
	Trung bình	0.3976

4.2.3 Kết quả sử dụng Hessian affine detector kết hợp với rootSIFT descriptor

STT	Truy vấn	Độ chính xác
1.	all_souls_000013	0.161571
2.	all_souls_000026	0.227066
3.	all_souls_000051	0.680386
4.	ashmolean_000000	0.618276
5.	ashmolean_000007	0.428772
6.	ashmolean_000058	0.101567
7.	ashmolean_000269	0.272393
8.	ashmolean_000305	0.572379
9.	balliol_000051	0.441722
10.	balliol_000167	0.0881901
11.	balliol_000187	0.200536
12.	balliol_000194	0.333437
13.	bodleian_000107	0.0558741
14.	bodleian_000108	0.430594
15.	bodleian_000132	0.770516
16.	bodleian_000163	0.131725
17.	bodleian_000407	0.136099
18.	christ_church_000179	0.61396
19.	christ_church_000999	0.450952
20.	christ_church_001020	0.600996
21.	cornmarket_000019	0.723745
22.	cornmarket_000047	0.653463
23.	cornmarket_000105	0.18844
24.	cornmarket_000131	0.367656
25.	hertford_000015	0.590053
26.	hertford_000027	0.643028
27.	hertford_000063	0.740821
28.	keble_000028	0.0990078
29.	keble_000055	0.0578283
30.	keble_000214	0.148485
31.	keble_000227	0.1131
32.	keble_000245	0.447946
33.	magdalen_000058	0.0851995
34.	magdalen_000078	0.0338103
35.	magdalen_000560	0.0967518
36.	oxford_000317	0.231279
37.	oxford_000545	0.41466
38.	oxford_001115	0.0435121
39.	oxford_001752	0.656768

40.	oxford_001753	0.194062
41.	oxford_002416	0.562349
42.	oxford_002562	0.0650981
43.	oxford_002734	0.597706
44.	oxford_002904	0.703219
45.	oxford_002985	0.304894
46.	oxford_003335	0.0724526
47.	oxford_003410	0.376038
48.	pitt_rivers_000033	0.320139
49.	pitt_rivers_000058	0.678913
50.	pitt_rivers_000087	0.647755
51.	pitt_rivers_000119	0.699112
52.	pitt_rivers_000153	0.169379
53.	radcliffe_camera_000095	0.600675
54.	radcliffe_camera_000519	0.703102
55.	radcliffe_camera_000523	0.690922
	Trung bình	0.3825

4.2.4 Kết quả sử dụng Hessian affine detector kết hợp với SIFT descriptor

STT	Truy vấn	Độ chính xác
1.	all_souls_000013	0.18818
2.	all_souls_000026	0.238188
3.	all_souls_000051	0.69279
4.	ashmolean_000000	0.355983
5.	ashmolean_000007	0.39601
6.	ashmolean_000058	0.179962
7.	ashmolean_000269	0.238509
8.	ashmolean_000305	0.425584
9.	balliol_000051	0.292163
10.	balliol_000167	0.0882495
11.	balliol_000187	0.150228
12.	balliol_000194	0.301795
13.	bodleian_000107	0.0548716
14.	bodleian_000108	0.441438
15.	bodleian_000132	0.742294
16.	bodleian_000163	0.195686
17.	bodleian_000407	0.046547
18.	christ_church_000179	0.61462
19.	christ_church_000999	0.470798
20.	christ_church_001020	0.590196
21.	cornmarket_000019	0.708204

22.	cornmarket_000047	0.685717
23.	cornmarket_000105	0.210263
24.	cornmarket_000131	0.313666
25.	hertford_000015	0.630002
26.	hertford_000027	0.646122
27.	hertford_000063	0.7856
28.	keble_000028	0.107071
29.	keble_000055	0.0672842
30.	keble_000214	0.372401
31.	keble_000227	0.32673
32.	keble_000245	0.457363
33.	magdalen_000058	0.0934552
34.	magdalen_000078	0.0292071
35.	magdalen_000560	0.0822919
36.	oxford_000317	0.205883
37.	oxford_000545	0.28414
38.	oxford_001115	0.0461578
39.	oxford_001752	0.647326
40.	oxford_001753	0.189825
41.	oxford_002416	0.57109
42.	oxford_002562	0.0494932
43.	oxford_002734	0.570385
44.	oxford_002904	0.664474
45.	oxford_002985	0.30101
46.	oxford_003335	0.0491942
47.	oxford_003410	0.435058
48.	pitt_rivers_000033	0.402751
49.	pitt_rivers_000058	0.675313
50.	pitt_rivers_000087	0.592403
51.	pitt_rivers_000119	0.644321
52.	pitt_rivers_000153	0.172967
53.	radcliffe_camera_000095	0.590365
54.	radcliffe_camera_000519	0.671988
55.	radcliffe_camera_000523	0.685917
	Trung bình	0.3758

4.2.5 Kết quả sử dụng SIFT detector kết hợp với rootSIFT descriptor

STT	Truy vấn	Độ chính xác
1.	all_souls_000013	0.18989
2.	all_souls_000026	0.31584
3.	all_souls_000051	0.640624
4.	ashmolean_000000	0.456072
5.	ashmolean_000007	0.43325

6.	ashmolean_000058	0.190135
7.	ashmolean_000269	0.089287
8.	ashmolean_000305	0.43921
9.	balliol_000051	0.491373
10.	balliol_000167	0.170679
11.	balliol_000187	0.117243
12.	balliol_000194	0.407462
13.	bodleian_000107	0.058318
14.	bodleian_000108	0.105241
15.	bodleian_000132	0.859104
16.	bodleian_000163	0.0447695
17.	bodleian_000407	0.041798
18.	christ_church_000179	0.563823
19.	christ_church_000999	0.52988
20.	christ_church_001020	0.663498
21.	cornmarket_000019	0.555341
22.	cornmarket_000047	0.647939
23.	cornmarket_000105	0.245256
24.	cornmarket_000131	0.218317
25.	hertford_000015	0.681626
26.	hertford_000027	0.44887
27.	hertford_000063	0.705647
28.	keble_000028	0.381751
29.	keble_000055	0.116539
30.	keble_000214	0.303537
31.	keble_000227	0.321479
32.	keble_000245	0.615568
33.	magdalen_000058	0.0592466
34.	magdalen_000078	0.0287671
35.	magdalen_000560	0.102857
36.	oxford_000317	0.401843
37.	oxford_000545	0.457827
38.	oxford_001115	0.0642692
39.	oxford_001752	0.663064
40.	oxford_001753	0.321384
41.	oxford_002416	0.213572
42.	oxford_002562	0.120632
43.	oxford_002734	0.792364
44.	oxford_002904	0.909963
45.	oxford_002985	0.323559
46.	oxford_003335	0.0472882
47.	oxford_003410	0.431624

48.	pitt_rivers_000033	0.706679
49.	pitt_rivers_000058	0.837192
50.	pitt_rivers_000087	0.810138
51.	pitt_rivers_000119	0.838646
52.	pitt_rivers_000153	0.841336
53.	radcliffe_camera_000095	0.908673
54.	radcliffe_camera_000519	0.738642
55.	radcliffe_camera_000523	0.794936
	Trung bình	0.4266

4.2.6 Kết quả sử dụng SURF features

Sử dụng hàm trainSURF và QuerySURF

STT	Truy vấn	Độ chính xác
1.	all_souls_000013	0.0221217
2.	all_souls_000026	0.068254
3.	all_souls_000051	0.203893
4.	ashmolean_000000	0.502426
5.	ashmolean_000007	0.116789
6.	ashmolean_000058	0.0436029
7.	ashmolean_000269	0.04
8.	ashmolean_000305	0.361934
9.	balliol_000051	0.0833333
10.	balliol_000167	0.151641
11.	balliol_000187	0.0833333
12.	balliol_000194	0.0833333
13.	bodleian_000107	0.0416667
14.	bodleian_000108	0.278472
15.	bodleian_000132	0.0174054
16.	bodleian_000163	0.511875
17.	bodleian_000407	0.214302
18.	christ_church_000179	0.211629
19.	christ_church_000999	0.0246795
20.	christ_church_001020	0.182313
21.	cornmarket_000019	0.333333
22.	cornmarket_000047	0.111111
23.	cornmarket_000105	0.111111
24.	cornmarket_000131	0.111111
25.	hertford_000015	0.212052
26.	hertford_000027	0.0275077
27.	hertford_000063	0.310693
28.	keble_000028	0.479464
29.	keble_000055	0.539286

30.	keble_000214	0.487103
31.	keble_000227	0.428571
32.	keble_000245	0.214719
33.	magdalen_000058	0.0149691
34.	magdalen_000078	0.0185185
35.	magdalen_000560	0.068445
36.	oxford_000317	0.0829174
37.	oxford_000545	0.111111
38.	oxford_001115	0.0350151
39.	oxford_001752	0.257938
40.	oxford_001753	0.232724
41.	oxford_002416	0.11927
42.	oxford_002562	0.0356582
43.	oxford_002734	0.196868
44.	oxford_002904	0.0221624
45.	oxford_002985	0.217949
46.	oxford_003335	0.0400079
47.	oxford_003410	0.00160256
48.	pitt_rivers_000033	0.166667
49.	pitt_rivers_000058	0
50.	pitt_rivers_000087	0
51.	pitt_rivers_000119	0
52.	pitt_rivers_000153	0
53.	radcliffe_camera_000095	0.00452489
54.	radcliffe_camera_000519	0.0244142
55.	radcliffe_camera_000523	0
	Trung bình	0.1502

5 Lưu ý khi chạy chương trình.

5.1 Runtime

- Cài đặt visual studio 2017 và phiên bản .net 4.7

- Cài đặt matlab 2017a trở lên hoặc runtime tương ứng.

5.2 Các tập tin cần thiết

Trong thư mục oxford/feat có chứa các dữ liệu đã được train tương ứng với các dectector và descriptor, sử dụng các file này thay thế cho các file nằm trực tiếp trong oxford/feat để sử dụng dữ liệu mong muốn.. Toàn bộ chương trình có thể tải tại: <https://github.com/nguyenmanhphuc/VRA-classroom>. Cụ thể:

- Source code: <https://github.com/nguyenmanhphuc/VRA-classroom/tree/master/VRA-final>
- Báo cáo: <https://github.com/nguyenmanhphuc/VRA-classroom/blob/master/VRA.Final.NguyenManhPhuc.pdf>,
<https://github.com/nguyenmanhphuc/VRA-classroom/blob/master/VRA.Final.NguyenManhPhuc.docx>

Do các hình ảnh của dataset quá nặng nên không được tải lên cùng với source chương trình. Trước khi chạy chương trình có thể tải về tại: <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/> và chép các hình giải nén vào thư mục oxford/images trong project VRA-final.

6 Tài liệu tham khảo

1. Thư viện lập trình Vlfeat: <http://www.vlfeat.org/index.html>, truy cập lần cuối vào ngày 10/12/2017.
2. Instance search, <https://github.com/nvtiep/Instance-Search>, truy cập lần cuối ngày 10/12/2017
3. Le Tan Hung, Các phép biến đổi Đồ hoạ Affine Transformations, <http://www.math.hcmuns.edu.vn/~tatuana/Do%20Hoa/Bai4B.pdf>, truy cập ngày 09/12/2017
4. Mikolajczyk, K. and Schmid, C. 2004. Scale & affine invariant interest point detectors. International Journal on Computer Vision 60(1):63-86
5. Scale Invariant Feature Transform (SIFT), <http://www.vlfeat.org/api/sift.html>, truy cập lần cuối ngày 10/12/2017
6. <https://www.pyimagesearch.com/2015/04/13/implementing-rootsift-in-python-and-opencv/>, truy cập lần cuối ngày 28/12/2017

