



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

CHUNG KIM KHÁNH (19127644), LA TRƯỜNG PHI (19127506)

BÁO CÁO

MỞ RỘNG KHẢ NĂNG TRUY VẤN ĐỐI TƯỢNG TRONG BỘ DỮ LIỆU ẢNH KHÔNG LỒ

Truy vấn thông tin thị giác

Thành phố Hồ Chí Minh - 2022

MỤC LỤC

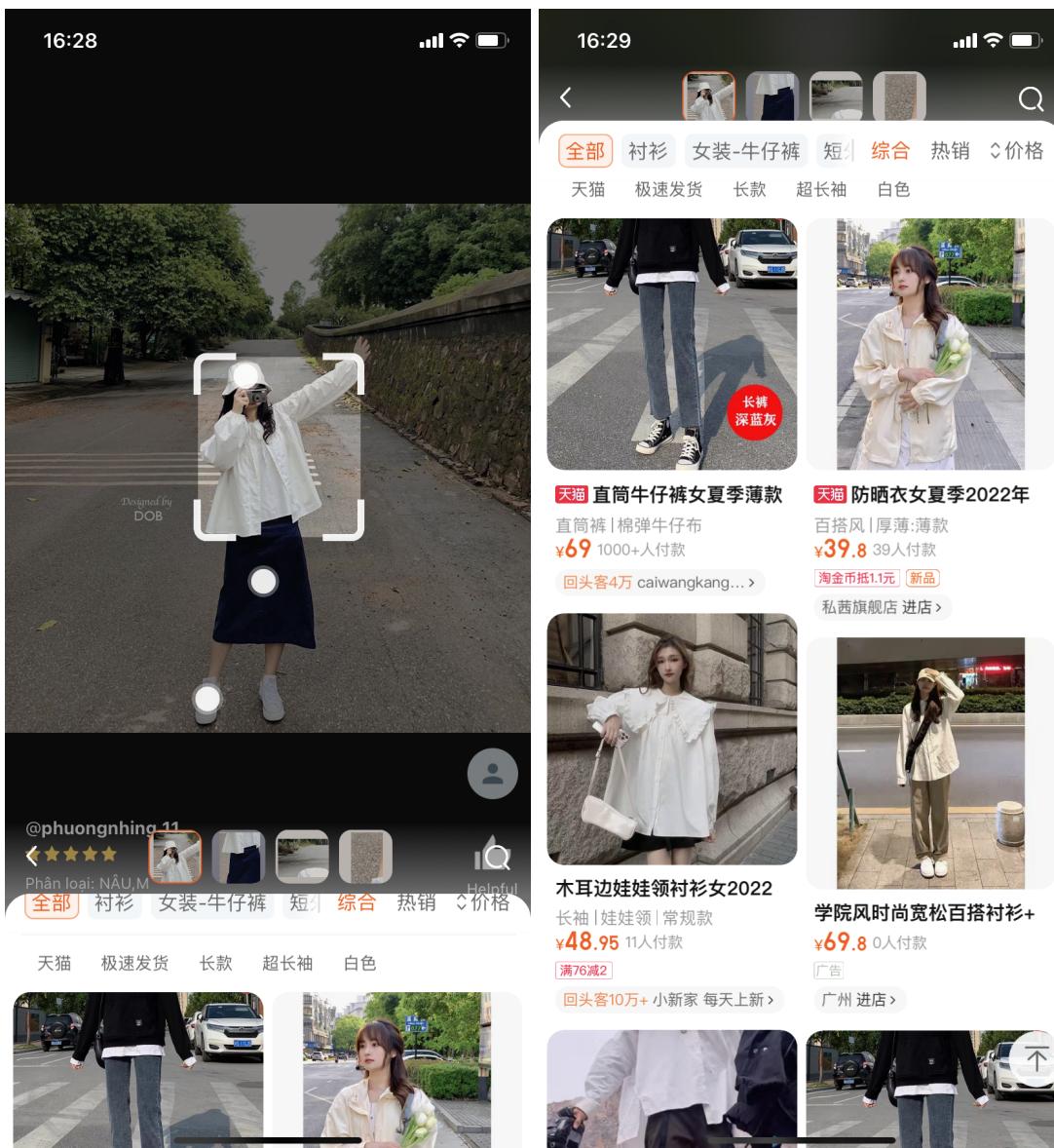
Truy vấn với quy mô lớn	3
Động lực nghiên cứu	3
Mở rộng quy mô lượng tử hoá (Scaling quantization)	5
Đặt vấn đề	5
Phân cụm K-means [4]	6
Phân cấp K-means (HKM)	7
Xấp xỉ K-means (AKM)	7
Đánh giá kết quả	10
Độ chính xác của xấp xỉ AKM	10
Hiệu suất truy xuất bằng cách sử dụng các từ vựng trực quan do AKM tạo ra	11
Khả năng khai quát hóa của các từ vựng lớn được tạo ra bởi AKM (hoặc các phương pháp khác).	13
Sắp xếp lại không gian (Spatial re-ranking)	15
Mô hình Bag of Visual Word [3]	15
Đặt vấn đề	15
Phương pháp RANdom SAmples Consensus (RANSAC)	16
Mất mát trong quá trình lượng tử hoá (Lost in quantization)	16
Nguồn gốc của lỗi lượng tử hoá	17
Phép gán mềm không gian bộ mô tả	17
Thiết lập	18
Trọng số TF-IDF và gán mềm	18
Xếp hạng lại không gian và gán mềm	18
Đánh giá thực nghiệm	19
Biến thể tham số	19
So sánh với các phương pháp khác	20
Ảnh hưởng của kích thước từ vựng	21
Xác minh không gian và Tăng cường đến 100K hình ảnh	21
Tổng kết	21

Tăng cường tìm sót (Boosting Recall)	22
Đặt vấn đề	22
Method	22
Model type	25
Methods	25
Query expansion baseline (Mở rộng truy vấn cơ sở)	25
Transitive closure expansion (Mở rộng đóng bắc cầu)	25
Average query expansion (Mở rộng truy vấn trung bình)	25
Recursive average query expansion (Mở rộng truy vấn trung bình đệ quy)	26
Multiple image resolution expansion (Mở rộng nhiều độ phân giải hình ảnh)	26
Cải thiện truy vấn Bag-of-Words (Làm thêm)	27
Hệ thống truy xuất cơ sở (Baseline retrieval system)	27
RootSIFT: Khoảng cách Hellinger cho SIFT	28
Hellinger kernel	29
SIFT	29
RootSIFT	29
Hiệu suất truy xuất (mAP) của các phương pháp đề xuất khác nhau	30
Kết luận	30
Mở rộng truy vấn mang tính phân biệt (Discriminative query expansion)	31
Chi tiết triển khai	33
Thảo luận	33
Tăng cường đặc trưng cơ sở dữ liệu (Database-side feature augmentation)	34
Mô tả	35
Chi tiết triển khai	35
Kết luận và khuyến nghị cho thiết kế hệ thống truy xuất	36
Tài liệu tham khảo	36

I. Truy vấn với quy mô lớn

1. Động lực nghiên cứu

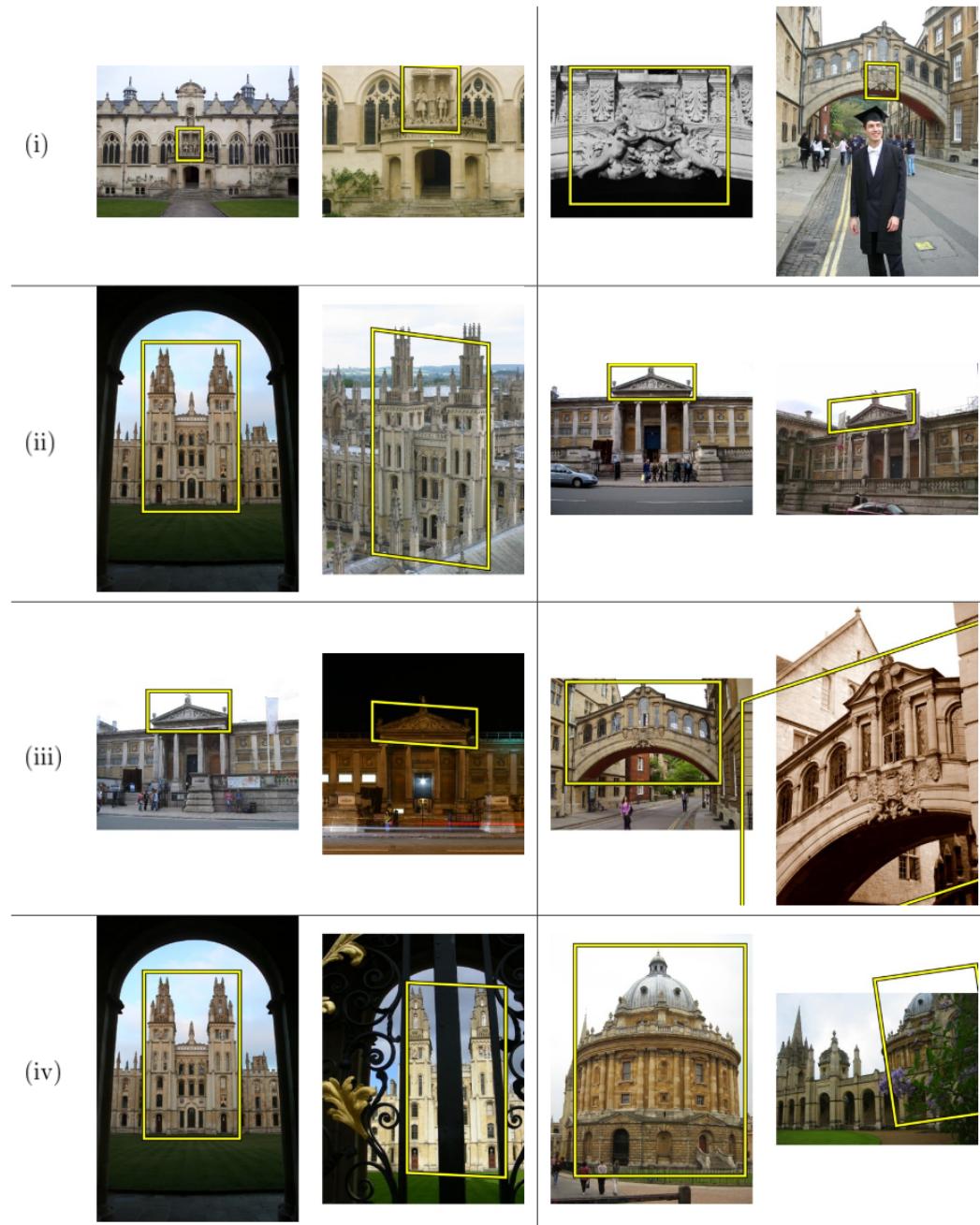
Trong thực tế, các doanh nghiệp như Google, Taobao, Facebook,... luôn có một kho dữ liệu lớn lưu trữ các hình ảnh. Bài toán được đưa ra ở đây là người dùng có thể chọn một đối tượng cụ thể bất kỳ trên một hình ảnh bất kỳ do người dùng đưa ra (ảnh đó có thể có trong kho dữ liệu hoặc không có). Và ta sẽ truy xuất ra được các hình ảnh đang có trong kho dữ liệu cũng có đối tượng đó.



Bài toán đối mặt với 2 thách thức chính:

NHÓM 1 - INNOVATION

Thách thức đầu tiên là về điều kiện hình ảnh trong thực tế (cụ thể là các điều kiện tự nhiên). Trong hình bên dưới là một số dữ liệu ảnh được lấy từ kho dữ liệu Oxford.



Ở đây là các ví dụ về các loại thách thức (khó khăn) được lấy ra trong bộ dữ liệu là:

- (i) Thay đổi về tỷ lệ của đối tượng đổi với khung hình (scale changes)
- (ii) Thay đổi góc nhìn (viewpoint changes)

(iii) Thay đổi về điều kiện ánh sáng (lighting changes)

(iv) Che khuất một phần đối tượng (partial occlusions)

Thách thức thứ 2, đối với tập dữ liệu hình ảnh cực lớn nhưng người dùng muốn kết quả truy xuất nhanh dưới 1 giây cho mỗi truy xuất. Tác giả muốn phương pháp hoạt động được trên nhiều loại thiết bị khác nhau nhưng không có phần nào của quá trình xử lý hoặc truy xuất sẽ mất thời gian hơn tuyến tính trong kích thước của kho dữ liệu. Trong bài báo này, không có phương pháp chính xác tuyệt đối để xác định đối tượng mà người dùng cần tìm có trong hình ảnh của kho dữ liệu không. Mỗi phương pháp sẽ có quy tắc xếp hạng khác nhau.

15 năm qua, Truy xuất văn bản đã giải quyết và khắc phục được những vấn đề được nêu trên. Cụ thể là sự phát triển vượt bậc của Google (Mô hình giỏ từ - Chapter 2). Đây như một bước thiết lập các ký tự cần truy xuất để giảm lượng tài liệu cần được truy xuất.

Điểm khác nhau giữa truy xuất văn bản và truy xuất hình ảnh.

Truy xuất văn bản	Truy xuất hình ảnh
<p>Không cần mã hoá dữ liệu</p> <p>Ví dụ: Một người gõ tìm kiếm 3 từ bất kỳ thì nó sẽ truy xuất ra các tài liệu có chứa 3 từ đó.</p>	<p>Mã hoá dữ liệu -> nhiều ý nghĩa nội dung</p> <p>Ví dụ: Tìm kiếm chiếc G63 sơn màu hồng được thiết kế độc quyền từ 1 hình ảnh người dùng chụp được. Rất khó để tìm được chính xác chiếc G63 của người dùng đưa ra.</p>

Trong chương này, ta sẽ đề cập đến 4 nội dung chính sau:

- (i) cải thiện khả năng mở rộng và chất lượng của lượng tử hóa từ trực quan (visual word quantization)
- (ii) đưa thông tin không gian vào bảng xếp hạng
- (iii) khắc phục các hiệu ứng lượng tử hóa được thấy khi sử dụng các từ vựng rất lớn
- (iv) các định hướng cho nghiên cứu trong tương lai.

2. Mở rộng quy mô lượng tử hóa (Scaling quantization)

a) Đặt vấn đề

Được đề cập ở chương 2, các hệ thống truy xuất dựa trên hình ảnh sẽ trích xuất các vectors đặc trưng sau đó phân cụm chúng thành một kho vocabulary gồm các visual words. Dẫn đến việc nhiều hình ảnh sẽ có nhiều vector đặc trưng, nhiều dữ liệu.

Phương pháp K-means có hiệu quả nhưng lại tốn kém khi mở rộng quy mô cho các vocabularies lớn. Nhiều nghiên cứu gần đây áp dụng phương pháp phân cấp cụm để tăng kích thước của các visual words từ đó tăng độ chính xác. Việc không dùng phương pháp phân cấp sẽ cải tiến phương pháp K-means. Tác giả đề xuất phương pháp xấp xỉ lân cận.

Bảng so sánh tổng quan các phương pháp:

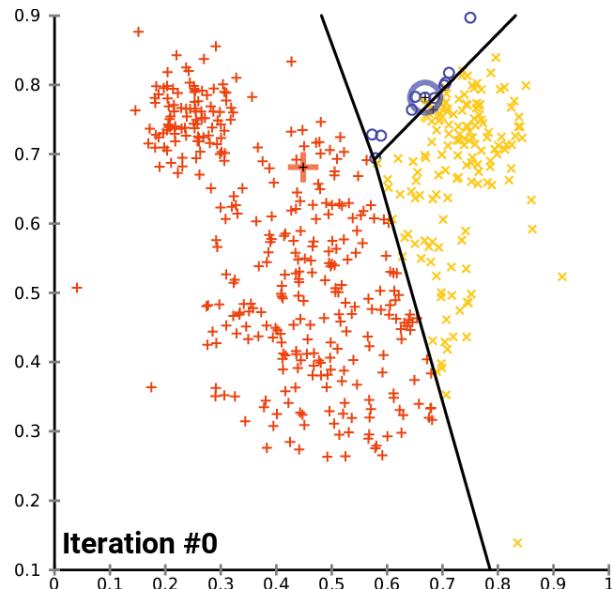
Tiêu chuẩn	K-means	HKM	AKM
Tốc độ	$O(K^2)$	$O(N \log K)$	$O(N \log K)$
Độ chính xác	Cao	Thấp	Cao

b) Phân cụm K-means [4]

- **Input:** Dữ liệu X (chưa có nhãn) và K cụm dữ liệu (cluster)
- **Output:** Điểm trung tâm của mỗi cụm (center)

Phân dữ liệu theo từng cụm khác nhau sao cho ở mỗi cụm, các dữ liệu có cùng tính chất

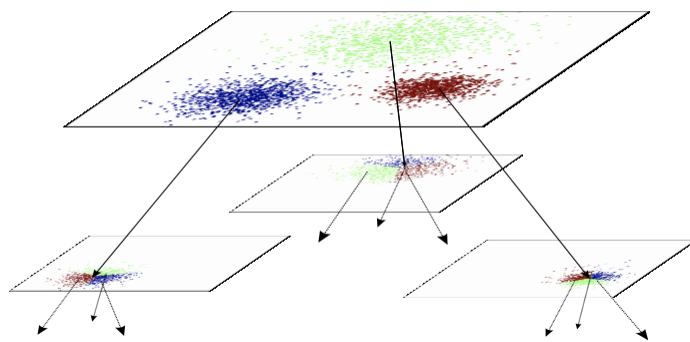
Sau phân cụm, ta thấy, đường phân định giữa các cụm (cluster) là đường trung trực giữa các điểm trọng tâm cụm (center) gần nhau.



=> Sử dụng toàn bộ các dữ liệu để xử lý.

c) Phân cấp K-means (HKM)

Nistér và Stewénius đề xuất tạo một "đồ thị nhánh cây" bằng cách sử dụng lược đồ phân cụm k-means phân cấp (còn gọi là lượng tử hóa vector có cấu trúc cây). Thay vì giải quyết một chủ đề phân cụm với một số lượng lớn các cụm, một hệ thống phân cấp có tổ chức cây gồm các chủ đề phân cụm nhỏ hơn sẽ được giải quyết dễ dàng hơn (nhanh hơn).



Ví dụ: ta có, Hệ số nhánh = 3, Level 2. Suy ra ta có Số vector = 32

Phương pháp này làm giảm thiểu sự ảnh hưởng (tầm quan trọng) của lỗi trong quá trình lượng tử hóa (quantization) đối với các trường hợp điểm nằm gần vùng ranh giới giữa các cụm.

Độ phức tạp thuật toán: $O(N \log K)$ → Lớn hơn 1 triệu visual words và nhiều điểm đặc trưng.

Tuy nhiên, chúng ta sẽ thấy rằng các cụm từ do HKM tạo ra không tạo ra visual vocabularies tốt để truy xuất. Khi đối tượng đã được phân vào sai cụm ngay từ ban đầu thì sẽ dẫn đến các lỗi sai sau của cụm đó. Vì vậy, kết quả có độ chính xác thấp.

d) Xáp xỉ K-means (AKM)

Phương pháp này sẽ cải thiện độ chính xác của HKM nhưng vẫn giữ nguyên độ phức tạp thuật toán đối với tập dữ liệu lớn. Trong k-means truyền thống, phần lớn thời gian dành cho việc tính toán neighbour gần nhất giữa điểm đặc trưng (feature points) và cụm trung tâm (cluster center).

=> AKM thay bằng phương pháp xấp xỉ lân cận gần nhất, sử dụng một rừng gồm một số cây k-d ngẫu nhiên (a forest of several randomized k-d trees).

K-d tree/K-d forest

BUILD-TREE($node, ys$)

```

1 if LEN( $ys$ ) = 1
2   then  $node \leftarrow$  NEW-LEAF-NODE( $ys[0]$ )
3   else  $dim, val \leftarrow$  CHOOSE-PSEUDORANDOM-SPLIT( $ys$ )
4      $ls, rs \leftarrow$  PARTITION-POINTS( $ys, dim, val$ )
5      $node \leftarrow$  NEW-INTERNAL-NODE( $dim, val$ )
6     BUILD-TREE( $node_{left}, ls$ )
7     BUILD-TREE( $node_{right}, rs$ )

```

SEARCH-TREE-ONCE($pq, node, dsq, query$)

```

1 while IS-INTERNAL-NODE( $node$ )
2   do
3      $disc \leftarrow (query[node_{dim}] - node_{val})$ 
4     if  $disc \leq 0$ 
5       then  $node \leftarrow node_{left}$ 
6         PRIORITY-QUEUE-INSERT( $pq, (node_{right}, dsq + disc^2)$ )
7       else  $node \leftarrow node_{right}$ 
8         PRIORITY-QUEUE-INSERT( $pq, (node_{left}, dsq + disc^2)$ )
9   return ( $node_{point}, DISTANCE(query, node_{point})$ )

```

SEARCH-KNN($forest, query, k$)

```

1  $pq \leftarrow$  NEW-PRIORITY-QUEUE()
2  $dists \leftarrow []$ 
3 for each  $tree \in forest$ 
4   do SEARCH-TREE-ONCE( $pq, tree, 0, query$ )
5 while LEN( $dists$ )  $\leq nchecks$ 
6   do  $node, dsq \leftarrow$  POP( $pq$ )
7     APPEND( $dists, SEARCH-TREE-ONCE(pq, node, dsq, query))$ 
8 return SELECT-K-SMALLEST( $dists, k$ )

```

Đây là mã giả của rừng k-d (k-d forest), là một số hoạt động chính được sử dụng để tìm kiếm xấp xỉ (approximate search) của thuật toán rừng k-d. “Build-Tree” xây dựng một cây duy nhất trong rừng theo độ phức tạp về thời gian $O(N \log N)$ và $O(N)$ về không gian. Search-Tree-Once đi xuống từ một nút (node) trên cây đến một lá, lấy bin đầu tiên tốt nhất và thêm các lựa chọn thay thế vào hàng đợi ưu tiên. Nó tính một khoảng cách tại nút lá được tìm thấy. Tìm kiếm-KNN lặp lại Search-Tree-Once với các nút tốt nhất và trả về k-NN (gần đúng) cho điểm truy vấn. Lưu ý rằng hàng đợi ưu tiên là chung cho tất cả các cây trong rừng.

Sự kết hợp của các cây này tạo ra một phân vùng chồng chéo của không gian đối tượng và giúp giảm thiểu hiệu ứng lượng tử hóa, trong đó các đối tượng nằm gần ranh giới phân vùng được gán cho một lân cận gần nhất không chính xác.

Một điểm dữ liệu mới được gán cho (xấp sỉ) cluster center gần nhất như sau:

- Ban đầu, mỗi cây được hạ xuống một lá và khoảng cách đến các ranh giới phân biệt được ghi lại trong một hàng đợi ưu tiên duy nhất cho tất cả các cây.
- Sau đó lặp đi lặp lại chọn nhánh tốt nhất từ tất cả các cây và tiếp tục thêm các nút chưa nhìn thấy vào hàng đợi ưu tiên.
- Dừng lại sau khi một số lượng cố định của đường đi trên cây đã được khám phá.

=> Bằng cách này, ta có thể sử dụng nhiều cây hơn mà không làm tăng đáng kể thời gian tìm kiếm.

Lưu ý: Trong thực tế, chúng ta lưu trữ nhiều điểm trong các nút lá của rừng k-d. Điều này làm giảm thời gian xây dựng và yêu cầu bộ nhớ và tăng vị trí bộ nhớ cache trong quá trình tìm kiếm.

AKM giảm thiểu chính xác hàm chi phí K-means

Flat K-means cố gắng giảm thiểu hàm chi phí sau đây trên các cluster centers:

$$L(k, a_i) = \sum_{i=1}^N \|x_i - c_{a_i}\|^2$$

Trong đó, L không lồi và việc tìm giá trị tối ưu toàn cục là NP-hard. K-mean tìm mức tối thiểu cục bộ của hàm này bằng cách sử dụng thuật toán kiểu EM, trong đó L được tối thiểu hóa lặp đi lặp lại w.r.t. đến các assignments, a_i (E-step), sau đó thu nhỏ w.r.t. đến các trung tâm cụm, c_k (M-step). Các bản cập nhật sau đây được tìm thấy thông qua sự khác biệt của L:

$$a_i^t = \operatorname{argmin}_{k_i} \|x_i - c_{k_i}^{t-1}\|^2$$

$$c_k^t = \frac{1}{|\{\forall i : a_i^t = k\}|} \sum_{\forall i : a_i^t = k} x_i$$

trong đó, giá trị t biểu thị giá trị của các biến tại lần lặp thứ t.

Chuỗi tổn thất (sequence of losses) không tăng: $L^1 \geq L^2 \geq \dots \geq L^T$ và có xu hướng tối thiểu cục bộ là L.

Lý thuyết thường khác xa với một số trường hợp trong thực tế. Tuy nhiên, một số công trình đã kiểm tra các thuộc tính của K-means bao gồm cách chọn khởi tạo tốt hơn,... Chỉ chấp nhận một phép gán gần đúng nếu nó làm giảm hàm chi phí, nếu không chúng ta sử dụng phép gán từ lần lặp trước. Điều này đảm bảo rằng chuỗi tần thắt không tăng và do đó sẽ hội tụ về mức tối thiểu cục bộ của hàm chi phí chính xác.

Qua quá trình kiểm thử của tác giả, xấp xỉ trong AKM có thể được coi là một dạng tối ưu hóa ngẫu nhiên có thể cho phép chúng ta tiến gần hơn đến mức tối thiểu toàn cục.

Phân tán Clustering

AKM thích hợp cho việc song song hóa trong các kiến trúc lập trình phân tán dữ liệu. Cụ thể, trên thực tế, mặc dù AKM nhanh hơn nhiều so với K-means, đặc biệt là đối với giá trị K lớn, việc thu thập một số lượng lớn các điểm đặc trưng vẫn có thể mất nhiều thời gian. Theo thực nghiệm, đối với $K = 1,0 \times 10^6$, $N = 17,0 \times 10^6$, một lần lặp lại AKM mất khoảng 5 giờ để hoàn thành. Do đó, trên một CPU đơn, quá trình chạy 30 lần lặp lại mất khoảng 6-7 ngày để hoàn thành.

Thuật toán AKM có thể dễ dàng song song hóa, trong đó mỗi nút trong một cluster bộ nhớ phân tán có thể lưu trữ một phần của các điểm đặc trưng với tổng một phần được tích lũy sau mỗi lần lặp. Vì vậy, thuật toán AKM mở rộng gần như tuyến tính lên đến khoảng 50 CPU cores.

e) Đánh giá kết quả

Ta kiểm tra ba khía cạnh chính:

A. Độ chính xác của xấp xỉ AKM

Dưới đây là độ chính xác của rừng k-d ngẫu nhiên để thực hiện các truy vấn lân cận gần nhất cho các bộ mô tả SIFT.

Search accuracy in a random set of 100,000 SIFT descriptors					
Checks\Trees	1	2	4	8	
128	32.7% 43.2% 49.0 μ s	36.1% 46.4% 50.6 μ s	40.0% 50.1% 53.0 μ s	43.3% 53.9% 56.1 μ s	
256	45.3% 57.5% 92.6 μ s	50.0% 61.6% 95.1 μ s	53.0% 64.9% 99.0 μ s	57.5% 68.5% 104.0 μ s	
512	55.5% 68.8% 179.2 μ s	64.5% 76.5% 184.0 μ s	66.8% 78.1% 190.5 μ s	71.5% 81.5% 199.5 μ s	
1024	67.5% 79.6% 352.7 μ s	73.8% 85.5% 361.9 μ s	79.1% 88.2% 373.7 μ s	83.2% 90.7% 388.5 μ s	

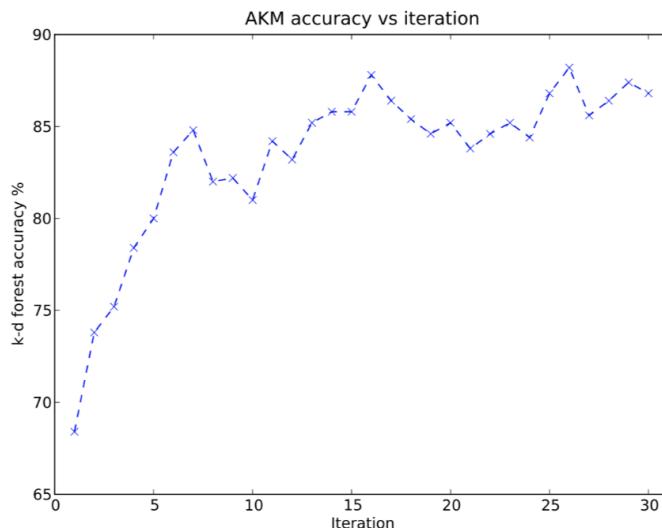
Search accuracy in a random of set of 1,000,000 SIFT descriptors					
Checks\Trees	1	2	4	8	
128	20.9% 27.6% 72.5 μ s	25.4% 32.6% 74.6 μ s	27.8% 36.4% 77.3 μ s	28.8% 37.4% 82.5 μ s	
256	28.5% 36.5% 128.5 μ s	34.6% 42.6% 131.7 μ s	37.8% 47.3% 136.6 μ s	40.6% 50.3% 143.2 μ s	
512	36.8% 46.8% 240.4 μ s	45.3% 54.5% 245.5 μ s	50.6% 60.3% 252.9 μ s	52.9% 62.7% 263.8 μ s	
1024	48.3% 60.1% 462.1 μ s	56.9% 67.8% 471.6 μ s	61.4% 71.4% 483.9 μ s	63.6% 73.6% 500.7 μ s	

Ở đây chỉ ra độ chính xác của rừng k-d ngẫu nhiên cho tìm kiếm lân cận gần nhất cho các tập hợp các bộ mô tả SIFT được lấy mẫu ngẫu nhiên có kích thước khác nhau từ tập dữ liệu Oxford. Kết quả được tính trung bình trên một tập hợp 5.000 bộ mô tả truy vấn đã được tổ chức. Ba con số được liệt kê là tỷ lệ phần trăm hàng xóm gần nhất được trả lại, phần trăm điểm trong 5% khoảng cách hàng xóm gần nhất được trả lại và thời gian tìm kiếm trên mỗi truy vấn. Đối với bộ mô tả SIFT 1M, việc thực hiện 1024 phép tính khoảng cách (ít hơn 0,1% của toàn bộ) trên 8 cây vẫn cho chúng tôi độ chính xác 63%. Thời gian được tạo ra trên chip Intel Xeon hàng hóa.

=> Trên thực tế, độ chính xác rừng k-d thực sự tăng lên khá nhiều trong quá trình chạy AKM. Đáng ngạc nhiên, hiệu ứng này được quan sát thấy ngay cả đối với các từ vựng rất lớn ($K = 2M$) với độ chính xác cuối cùng tương tự.

B. Hiệu suất truy xuất bằng cách sử dụng các từ vựng trực quan do AKM tạo ra

AKM được so sánh với K-means trong trường hợp chạy K-means vẫn có thể chạy (N và K nhỏ). Đối với các điểm dữ liệu này, hiệu suất cũng giống như trong phương sai trong phép đo.



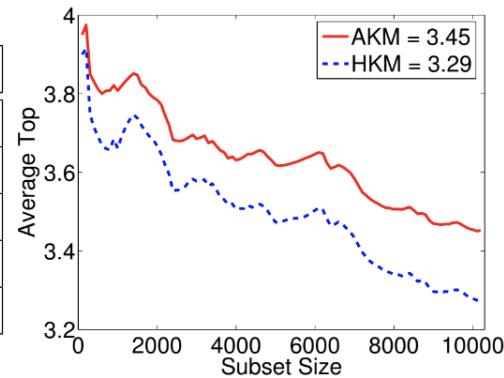
Đồ thị xác định độ chính xác của rừng k-d ngẫu nhiên khi quá trình chạy AKM diễn ra. Sau mỗi lần lặp, một mẫu nhỏ các điểm dữ liệu được sử dụng để ước tính độ chính xác của rừng k-d trong việc trả về lân cận gần nhất. Sau khi chạy AKM, NN accuracy dường như tăng lên đáng kể. Các kết quả này dành cho việc phân cụm trên tập dữ liệu Oxford với $K = 1M$ cluster centers.

Clustering Parameters		mAP	
Number of descriptors	Vocabulary size	K-means	AKM
800K	10K	0.355	0.358
1M	20K	0.384	0.385
5M	50K	0.464	0.453
16.7M	1M		0.618

Đây là so sánh giữa K-means với AKM cho các số khác nhau của bộ mô tả đào tạo và các cluster centers.

=> Ta thấy, AKM vượt trội hơn so với những công nghệ tiên tiến trước đây.

Method	# scoring levels	Average top
HKM	1	3.16
HKM	2	3.07
HKM	3	3.29
HKM	4	3.29
AKM		3.45



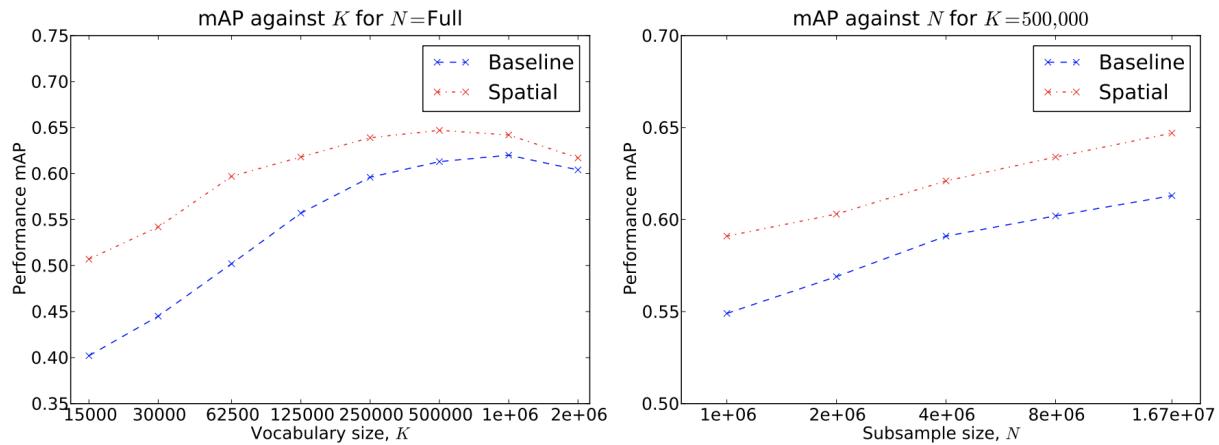
Đây là so sánh giữa AKM và HKM trên Recognition Benchmark của việc sử dụng các bộ mô tả để đào tạo và kiểm tra được cung cấp trong các bài báo trước đó. Các con số cho kết quả HKM giống với kết quả do tác giả thực hiện. Điều duy nhất thay đổi là phương pháp lượng tử hóa. K = 1M cho cả hai phương pháp. Điểm trung bình cao nhất được liệt kê là số lượng kết quả chính xác trung bình được trả về trong bốn kết quả đầu tiên. Đối với mỗi truy vấn trong tập dữ liệu của UK, có chính xác 3 kết quả phù hợp khác. Do đó, điểm số này nằm ngoài tối đa 4.

Method	Dataset	mAP	
		Bag-of-words	Spatial
FQM [157]	OXFORD	0.164	
HKM-1	OXFORD	0.439	0.469
HKM-2	OXFORD	0.418	
HKM-3	OXFORD	0.372	
HKM-4	OXFORD	0.353	
AKM	OXFORD	0.618	0.647
AKM	OXFORD-100K	0.490	0.541
AKM	OXFORD-1M	0.393	0.465

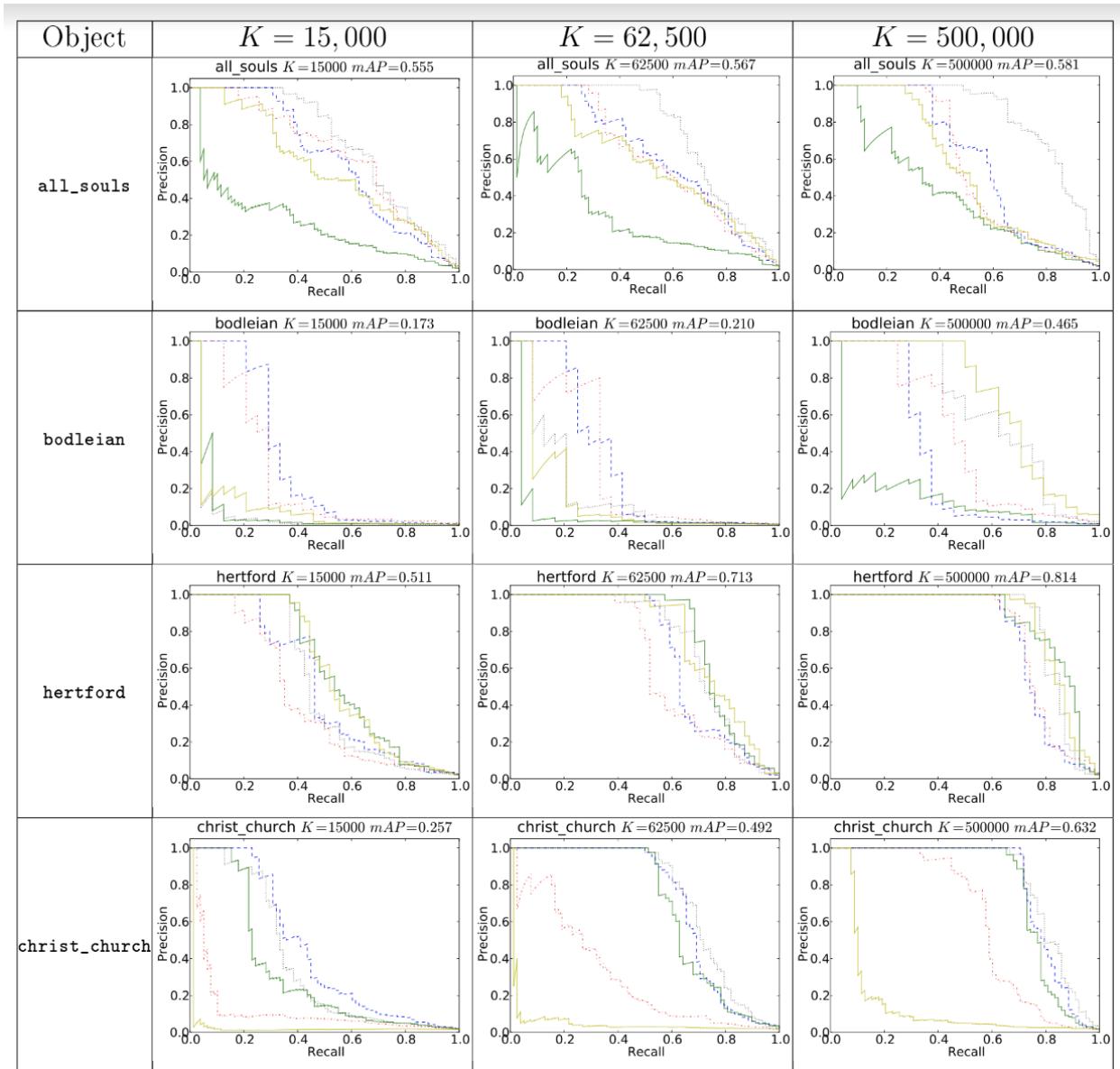
So sánh từ vựng qua ba tập dữ liệu. Đổi với phương pháp HKM, số lượng cấp độ được sử dụng để cho score được liệt kê trong tên phương pháp. Tất cả các phương pháp sử dụng K = 1M trung tâm cụm, được tạo từ tất cả các bộ mô tả trong tập dữ liệu Oxford. Đồng thời cũng hiển thị kết quả cho một phương pháp lượng tử hóa cố định.

C. Khả năng khái quát hóa của các từ vựng lớn được tạo ra bởi AKM (hoặc các phương pháp khác).

N\K	15,000	30,000	62,500	125,000	250,000	500,000	1,000,000	2,000,000
1M	0.380/0.497	0.413/0.516	0.439/0.536	0.486/0.573	0.530/0.588	0.549/0.591		
2M	0.393/0.506	0.425/0.528	0.466/0.551	0.503/0.579	0.556/0.600	0.569/0.603	0.591/0.619	
4M	0.387/0.502	0.419/0.525	0.466/0.569	0.525/0.590	0.557/0.608	0.591/0.621	0.594/0.616	0.594/0.607
8M	0.393/0.518	0.425/0.549	0.477/0.583	0.540/0.612	0.589/0.644	0.602/0.634	0.600/0.622	0.601/0.618
Full	0.402/0.507	0.445/0.542	0.502/0.597	0.557/0.618	0.596/0.639	0.613/0.647	0.620/0.642	0.604/0.617



Hiệu suất truy xuất trên tập dữ liệu Oxford khi lượng dữ liệu được phân nhóm với N là số lượng cụm, K là khác nhau. Mặc dù mức tăng hiệu suất ít hơn so với việc thay đổi K. Điều này cho thấy rằng nếu thời gian tính toán ngắn, việc phân nhóm sử dụng K lớn và N nhỏ sẽ mang lại hiệu suất truy xuất tốt.



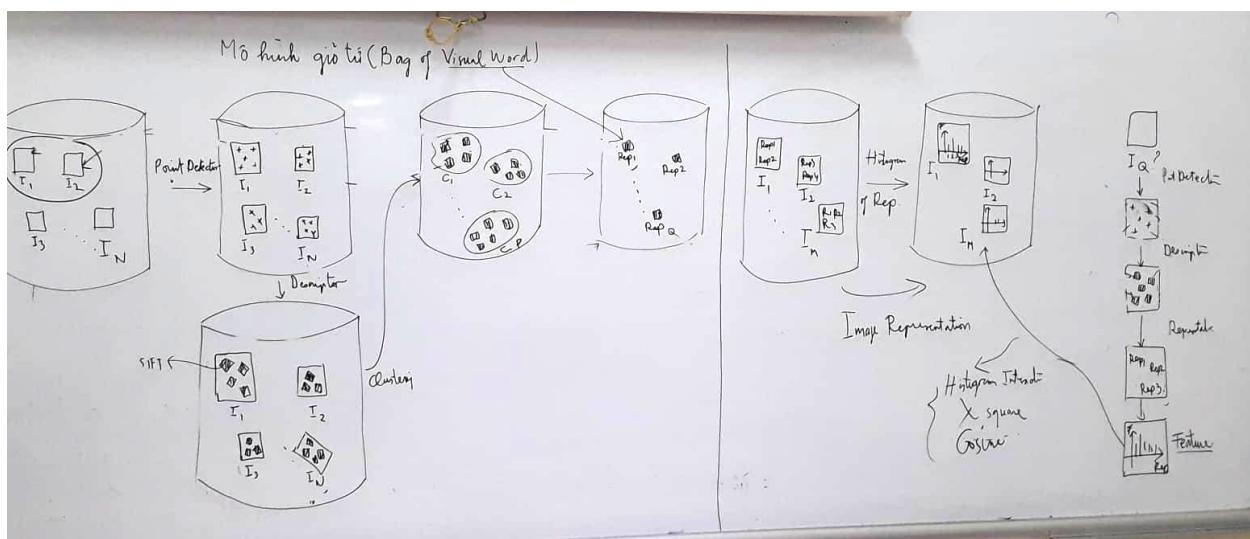
Nhìn vào các đường precision-recall cho các đối tượng truy vấn cụ thể trong tập dữ liệu Oxford để kiểm tra hiệu suất khi K tăng lên. Ở cấp độ này, chúng ta có thể thấy rằng các đối tượng khác nhau được phát hiện theo những cách khác nhau bởi kích thước của từ vựng. Điều này có thể liên quan đến mức độ tương tự trực quan của một đối tượng cụ thể với các đối tượng khác trong tập dữ liệu. Các đối tượng dễ bị nhầm lẫn có thể cần một từ vựng phân biệt trước khi việc truy xuất có thể trả về kết quả tốt.

3. Sắp xếp lại không gian (Spatial re-ranking)

a) Mô hình Bag of Visual Word [3]

Ý tưởng của Bag of Visual Word là chuyển đổi hình ảnh thành các đặc trưng (features) về ngữ nghĩa. Những đặc trưng bao gồm các keypoints và mô tả. Keypoint là các điểm trọng tâm (“stand out” points) của hình ảnh. Vì vậy, dù cho hình ảnh có xoay, thu nhỏ hoặc mở rộng thì các keypoints vẫn không đổi. Các keypoints và mô tả dùng để tạo các bộ từ vựng (vocabularies) và mô tả hình ảnh dưới dạng biểu đồ tần số (frequency histogram) của các đặc trưng trong từng bức ảnh. Từ các biểu đồ tần số ta có thể tìm thấy một hình ảnh tương tự hoặc danh mục của ảnh.

Cách xây dựng mô hình Bag of Visual Word:



Xác định các đặc trưng, bộ mô tả của mỗi ảnh trong bộ dữ liệu và xây dựng một bộ từ điển trực quan (visual dictionary). Việc này có thể thực hiện bằng các thuật toán như SIFT, KAZE,...

Sau đó, ta thực hiện cluster các mô tả (ta có thể sử dụng thuật toán K-Means, DBSCAN,...). Trọng tâm (center) của mỗi cluster sẽ được dùng như một bộ từ điển từ vựng trực quan (visual dictionary's vocabularies).

Cuối cùng, đối với mỗi ảnh, ta sẽ tạo được biểu đồ tần số từ các từ vựng và tần số của các từ vựng trong ảnh.

b) Đặt vấn đề

Trong phần này là đề xuất và mô tả một phương pháp xếp hạng lại không gian để cải thiện hiệu suất truy xuất so với mô hình Bag-of-words (BoW) tiêu chuẩn. Mô hình BoW bỏ qua yếu tố không gian, hệ thống truy xuất xem mọi tài liệu hình ảnh như một tập hợp các từ trực quan không có thứ tự.

c) Phương pháp RANdom SAmple Consensus (RANSAC)

Thuật toán RANdom SAmple Consensus (RANSAC) tiêu chuẩn để ước tính ma trận cơ bản đầy đủ 3-D hoặc phép đồng nhất xạ ảnh 2-D giữa hai hình ảnh là quá chung chung và chạy rất chậm. Vậy nên tác giả đã đề xuất các sửa đổi sau:

1. Tải nhận dạng, vị trí và hình dạng của từng từ trực quan của tài liệu. Nghiên cứu gần đây đã chỉ ra rằng hình dạng elip có thể được lượng tử hóa nếu cần nén nhiều hơn. Khi các từ vựng phân biệt được sử dụng, số lượng tương ứng phù hợp có thể khá ít, điều này cũng thúc đẩy tốc độ xếp hạng lại.
2. Sử dụng các phép biến đổi phẳng đơn giản giữa hai hình ảnh (DOF, 4 DOF và 5 DOF, cùng với các loose thresholds trên khoảng cách bên trong). Do đó, ta vẫn có thể xấp xỉ kết nối một số cảnh có biến dạng phối cảnh đáng kể bằng cách sử dụng loose bound đối với lỗi truyền tải.
3. Sử dụng hình elip cũng như vị trí của các interest points. Phép kết nối ellipse-ellipse có thể xác định các phép biến đổi lên đến 5 DOF. Điều này có nghĩa là chúng ta có thể tạo ra một giả thuyết từ một correspondence đơn ($N = 1$ như ở trên). Trong thực tế, sau đó có thể chỉ cần liệt kê tất cả các correspondences và chọn cái có số lượng nội dung cao nhất.
4. Khi đã tìm thấy một phép đồng nhất tốt, ta lấy các giá trị nội tại được tìm thấy và sử dụng chúng để ước tính lại một phép đồng nhất hoàn chỉnh (6 DOF), sử dụng phương pháp bình phương nhỏ nhất.

Tính toán một số đồng dạng khác nhau:

3 DOF: Cho phép dịch chuyển và chia tỷ lệ đồng nhất.

4 DOF: Cho phép dịch chuyển và chia tỷ lệ dị hướng.

5 DOF: Cho phép dịch chuyển, chia tỷ lệ dị hướng và cắt bảo quản theo phương thẳng đứng.

4. Mất mát trong quá trình lượng tử hóa (Lost in quantization)

Phần này khám phá các kỹ thuật ánh xạ từng đối tượng địa lý trực quan (ví dụ: bộ mô tả SIFT) với một tập hợp các từ trực quan có trọng số, thu được bằng cách chọn nhiều từ dựa trên sự gần gũi trong không gian bộ mô tả. Điều này cho phép so khớp các tính năng đã bị mất trong giai đoạn lượng tử hóa của hệ thống truy xuất của tác giả. Tác giả mô tả cách biểu diễn này có thể được kết hợp vào kiến trúc tf-idf chuẩn và cách chỉnh sửa không gian. Cách tiếp cận

của tác giả giải quyết cụ thể vấn đề về recall từ một truy vấn ban đầu và do đó bổ sung cho các phương pháp mở rộng truy vấn (xem II). Bộ mô tả chiều cao được ánh xạ tới một tổ hợp có trọng số của các từ trực quan, thay vì được gán cứng cho một từ như trong hệ thống cơ sở.

a) Nguồn gốc của lỗi lượng tử hóa

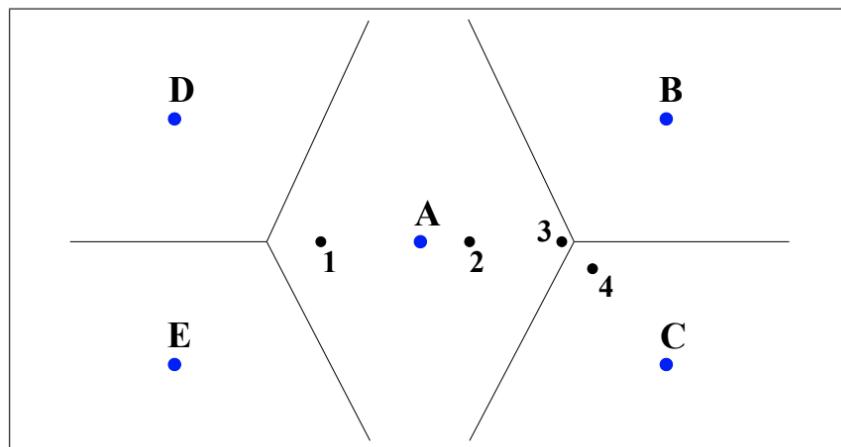
Trong Bag of Visual Words, hai đặc trưng hình ảnh được coi là giống hệt nhau nếu chúng được gán cho cùng một từ trực quan - visual word (cluster center). Mặt khác, hai đặc trưng được gán cho các cụm khác nhau (thậm chí rất gần) được coi là hoàn toàn khác nhau. Trên thực tế, lượng tử hóa cung cấp một giá trị gần đúng rất thô cho khoảng cách thực tế giữa hai đặc trưng: bằng không nếu được gán cho cùng một từ trực quan và ngược lại. Trong thực tế, gán cứng (hard assignment) dẫn đến lỗi vì sự thay đổi trong bộ mô tả tính năng.

Sự thay đổi này phát sinh từ nhiều nguồn: nhiều hình ảnh, độ sáng cảnh khác nhau, sự không ổn định trong quá trình phát hiện đặc trưng và những thay đổi không phải của vùng đo.

b) Phép gán mềm không gian bộ mô tả

Thuật ngữ “soft-assignment” thường được sử dụng trong so sánh biểu đồ. Nó mô tả các kỹ thuật xác định một giá trị liên tục với sự kết hợp có trọng số của các bins lân cận hoặc làm mịn biểu đồ để số lượng trong bin được truyền sang các bins lân cận.

Lợi ích của gán mềm (soft-assignment). Ta có:



Điểm A-E đại diện cho các cluster centers (từ trực quan), và điểm 1 đến 4 là các đối tượng địa lý. Chứng minh hai lợi ích của phân công mềm:

- (i) Trong gán cứng (hard-assignment), các tính năng 3 và 4 sẽ không bao giờ được kết nối (match) vì chúng được gán cho các từ trực quan khác nhau mặc dù nằm gần trong không gian bộ mô tả. Sử dụng phép gán mềm, các từ 3 và 4 sẽ được gán cho A, B và C (với các trọng số nhất định) và có thể được đối sánh chặt chẽ khi chúng ở gần nhau trong không gian bộ mô tả.

(ii) Trong phép gán cứng, tất cả các tính năng 1 đến 3 đều được gán cho từ A như nhau và không có cách nào để phân biệt rằng 2 và 3 gần hơn 1 và 3. Phép gán mềm cung cấp một cách ghi lại thông tin này và sau đó có trọng số hơn đến các trận đấu gần hơn và ít hơn các trận đấu xa hơn.

c) Thiết lập

Trọng số TF-IDF và gán mềm

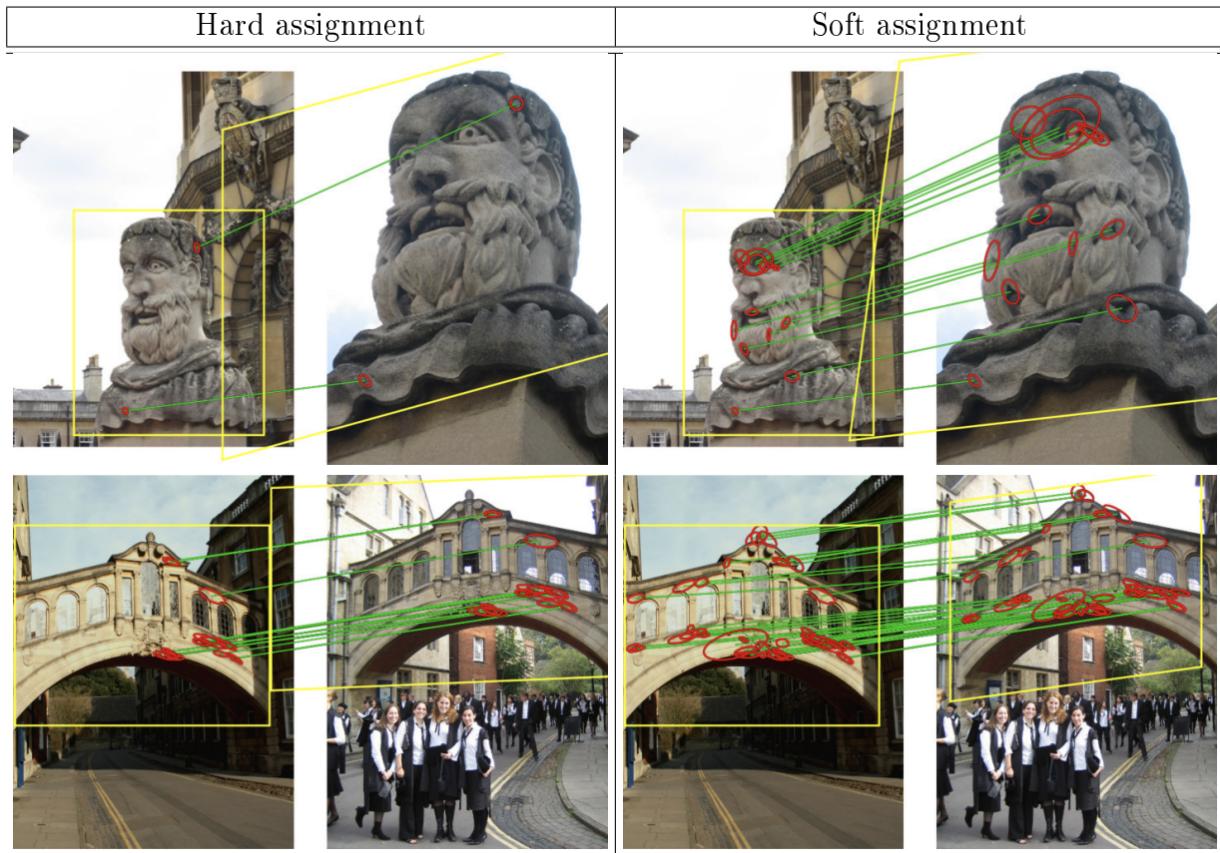
Sơ đồ trọng số tf-idf thường chỉ được áp dụng cho các số nguyên từ trực quan trong hình ảnh. Nó yêu cầu một số sửa đổi để xử lý một bộ mô tả được đại diện bởi một vectơ r trọng số. Ta điều chỉnh sơ đồ trọng số này cho phân cụm mềm như sau. Đối với thuật ngữ tần số, chúng ta chỉ cần sử dụng giá trị trọng số chuẩn hóa cho mỗi từ trực quan. Đối với phép đo đặc trưng tài liệu nghịch đảo, việc đếm số lần xuất hiện của một từ trực quan là một, bắt kể trọng lượng của nó nhỏ đến mức nào, sẽ cho kết quả tốt nhất.

Xếp hạng lại không gian và gán mềm

Khi chuyển từ chuyển từ gán cứng sang mềm, một trong những cân nhắc về hiệu suất chính là tiềm năng tăng trưởng số lượng tương ứng dự kiến giữa hai hình ảnh (được coi là tập hợp các tính năng chia sẻ ít nhất một phép gán từ trực quan). Sự phát triển này phát sinh từ việc mỗi đặc điểm hình ảnh được gán cho nhiều từ trực quan hơn để nó có khả năng khớp với nhiều đặc điểm hơn trong hình ảnh khác.

Tuy nhiên, vì vốn từ vựng về hình ảnh là cụ thể và lớn (1 triệu), nên xác suất hai đặc điểm không liên quan được gán cho cùng một từ trực quan là nhỏ. Do đó, sự tăng trưởng gần như tuyến tính về số lượng tương ứng dự kiến khi số lượng lân cận gần nhất được lấy tăng lên.

Tập hợp các correspondences dự kiến phù hợp với một phép đồng nhất khác được xác định bằng cách ước lượng kiểu RANSAC. Điều này yêu cầu score cho mỗi phép biến đổi được giả định và score có thể sử dụng vectơ có trọng số được liên kết với từng đặc trưng, thay vì chỉ đơn giản là đếm số lượng correspondences nội số.



=> Soft assignment cho kết quả tốt hơn Hard assignment

d) Đánh giá thực nghiệm

Biến thèm tham số

Điều chỉnh các tham số gán mềm. Các kết quả này dành cho 1 triệu từ vựng, được thực nghiệm trên Oxford.

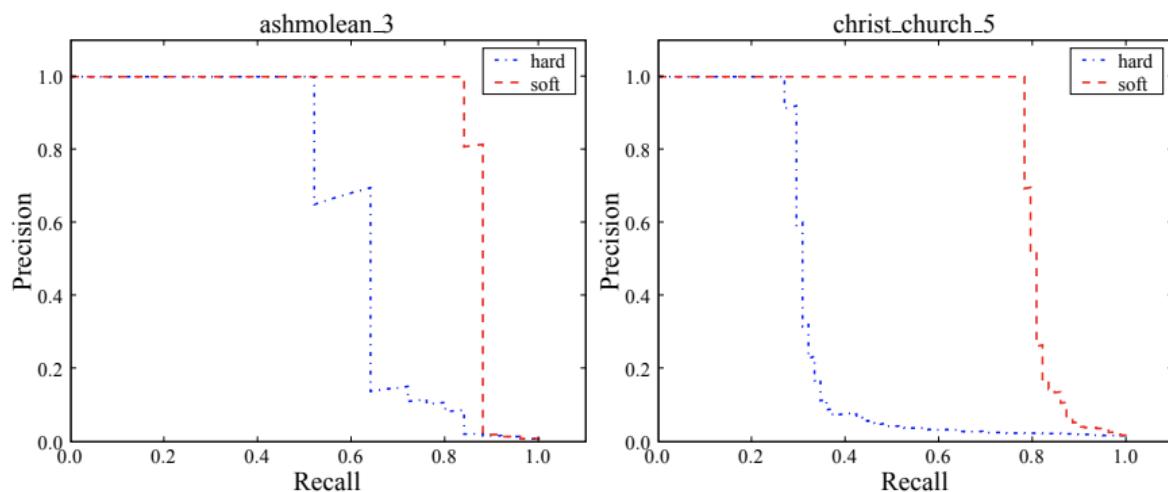
r	σ^2	Training data	
		OXFORD	PARIS
3	5,000	0.671	0.495
3	6,250	0.673	0.494
3	7,500	0.672	0.493
5	5,000	0.674	0.502
5	6,250	0.673	0.499
5	7,500	0.673	0.496

Hiệu suất của hệ thống do hai thông số này thay đổi rất ít. Đặc biệt, việc gán mềm cho hơn 4 lân cận gần nhất không mang lại thêm bất kỳ lợi ích gì. Điều này có thể được cho là do sự nhầm lẫn gia tăng trong quá trình so khớp. Do đó, trong các thử nghiệm tiếp theo sẽ sử dụng $r = 3$, $\sigma^2 = 6.250$.

So sánh với các phương pháp khác

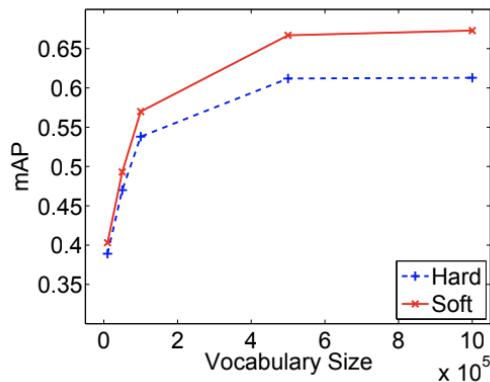
	Training data	
Method	OXFORD	PARIS
HKM [105] (1 level)	0.422	0.401
HKM [105] (2 level)	0.410	0.340
Hard [112]	0.614	0.403
Soft	0.673	0.494

=> Từ vựng được gán mềm hoạt động tốt nhất khi được đào tạo trên bộ dữ liệu Oxford hoặc Paris.



=> Trong cả hai trường hợp (Recall và Precision), hiệu suất đạt được là lớn.

Ảnh hưởng của kích thước từ vựng



=> Gán mềm tạo ra một mạng lưới lớn hơn nhiều khi các từ vựng lớn hơn được sử dụng do khả năng của gán mềm khắc phục một số lượng tử hóa không gian khi các từ vựng lớn được sử dụng.

Xác minh không gian và Tăng cường đến 100K hình ảnh

SP		Testing OXFORD		Testing OXFORD-100K	
		Training data		Training data	
		OXFORD	PARIS	OXFORD	PARIS
Hard		0.614	0.403	0.498	0.290
Hard	x	0.653	0.460	0.565	0.385
Soft		0.673	0.493	0.534	0.343
Soft	x	0.731	0.598	0.620	0.480

=> Từ vựng được gán mềm luôn vượt trội hơn từ vựng được gán cứng ngay cả khi được kết hợp với cấu trúc không gian hoặc tăng cường số lượng dữ liệu.

5. Tổng kết

Trong chương này đã trình bày ba phương pháp giúp mở rộng độ chính xác, khả năng mở rộng và tốc độ truy xuất đối tượng cụ thể trong các tập dữ liệu lớn:

- Large vocabularies do AKM tạo ra, đã được chứng minh là cải thiện đáng kể độ chính xác của việc truy xuất so với phương pháp phân cụm HKM hiện đại trước đây.
- Phương pháp để spatial re-ranking (xếp hạng lại không gian) sử dụng các tương ứng hình elip và các phép biến đổi giới hạn để đưa ra giả thuyết về các đồng phân phẳng cực kỳ nhanh chóng.

- Kỹ thuật soft-assignment được thiết kế để khắc phục một số tác hại của lỗi quantization gia tăng khi các vocabularies lớn được sử dụng, trong khi vẫn giữ được các lợi thế phân biệt.

II. Tăng cường tìm sót (Boosting Recall)

1. Đặt vấn đề

"Khi chúng ta tìm kiếm một bộ sưu tập tài liệu, chúng ta cố gắng truy xuất các tài liệu có liên quan mà không cần truy xuất các tài liệu không liên quan. Vì chúng ta không có Oracle nào sẽ cho chúng ta biết mà không sai về tài liệu nào có liên quan và không liên quan nên chúng ta phải sử dụng kiến thức không hoàn hảo để đoán cho bất kỳ tài liệu nào cho dù nó có liên quan hay không liên quan." Van Rijsbergen [158].

Trong Chương I, chúng ta đã thấy rằng một hệ thống truy xuất đối tượng cụ thể có thể mở rộng có thể được xây dựng bằng các phương pháp lấy cảm hứng từ các hệ thống truy xuất văn bản. Tuy nhiên, các phần mở rộng được giới thiệu chủ yếu liên quan đến việc tăng độ chính xác của việc truy xuất đảm bảo rằng 10 kết quả hàng đầu hoặc 20 kết quả được trả lại cho người dùng là chính xác, mà không kiểm tra việc thu hồi truy vấn. Các phương pháp như sử dụng các từ vựng rất lớn làm tăng cơ hội mà các từ trực quan cụ thể sẽ chỉ xảy ra trên một đối tượng cụ thể, nhưng chúng ta nhất thiết phải mất một số biến khi làm điều này. Ví dụ, rất cụ thể các từ trực quan sẽ không lặp đi lặp lại trên cùng một đối tượng trong các điều kiện hình ảnh rất cực đoan. Reranking không gian cũng chủ yếu tăng độ chính xác do thực tế là chúng ta chỉ có thể chạy nó trên các tài liệu được xếp hạng hàng đầu từ TF-IDF và nó chỉ phù hợp với các tính năng với cùng một từ trực quan. Việc phân loại mềm khắc phục một số vấn đề này nhưng vẫn không thể phù hợp với các mô tả trong điều kiện hình ảnh rất cực đoan (có thể ở xa trong không gian mô tả). Trong chương này, chúng tôi trình bày một phương thức mà chúng tôi gọi là total recall, chủ yếu liên quan đến việc tăng cường độ sót truy xuất đối tượng cụ thể. Đáng ngạc nhiên, nó có thể đạt được kết quả recall cao trong khi vẫn duy trì độ chính xác gần như hoàn hảo. Nó đạt được điều này bằng cách tận dụng thông tin thứ hai về các đối tượng hoàn toàn chứa trong kho văn bản đang được tìm kiếm khi nó chứa nhiều lần xuất hiện của đối tượng được truy vấn.

2. Method

Ta có phương pháp cơ bản được đề xuất như sau: (xem hình 2.1):

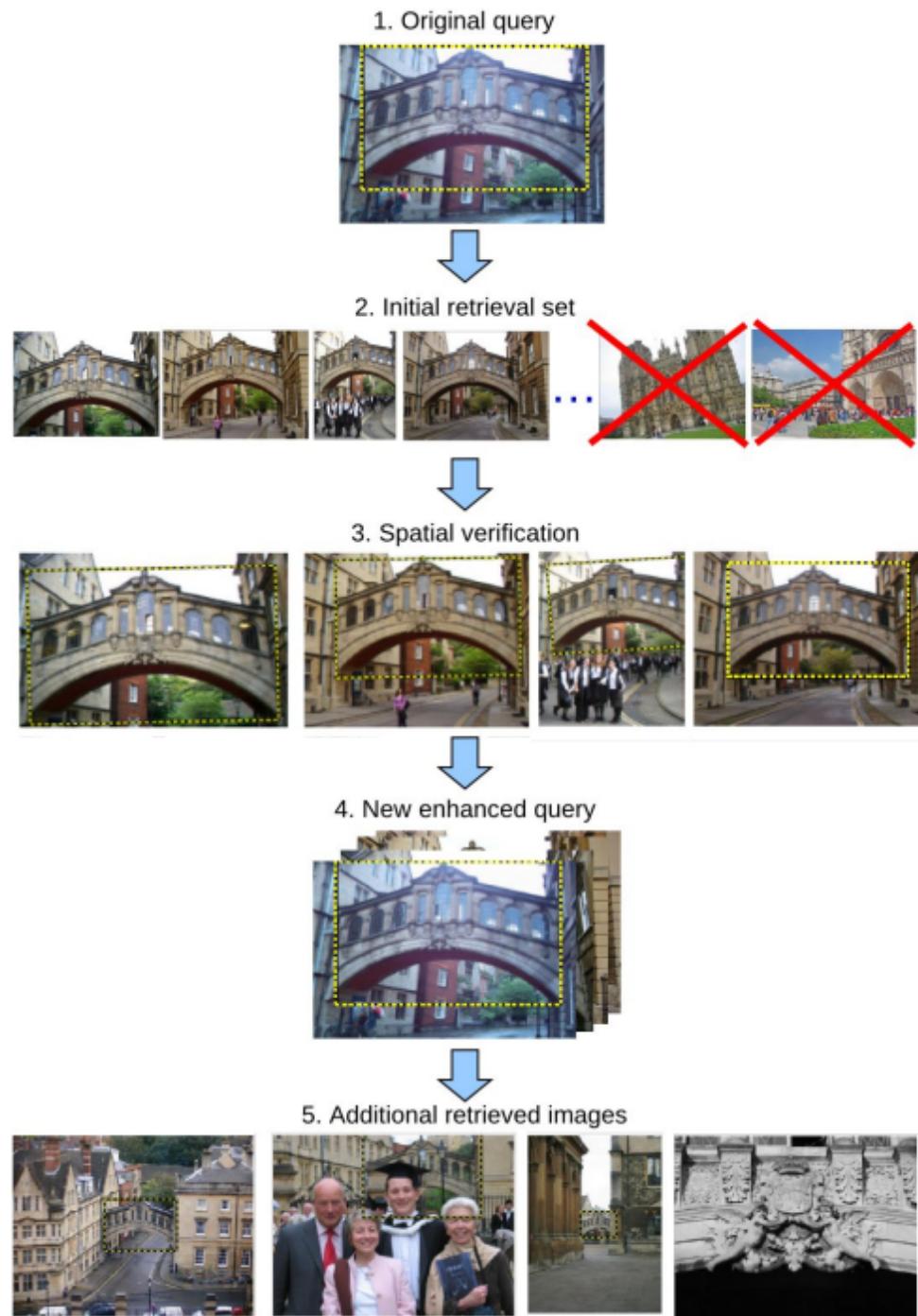
1. Đưa ra một vùng truy vấn, tìm kiếm tập hình ảnh và truy xuất một tập hợp các vùng hình ảnh khớp với đối tượng truy vấn. Điều này sử dụng các kỹ thuật được thảo luận trong Chương I và các lợi ích về độ chính xác đạt được bằng các dữ liệu đặc biệt lớn và xếp lại không gian. Cụ thể, được đưa ra một vùng truy vấn, Q, bao gồm một tập hợp các hình trực quan, truy vấn ban đầu trả về một danh sách kết quả được xếp hạng, R_i , cùng với các tập kết quả truy vấn đồng tính và số lượng inliers, (H_i, N_i) , phù hợp với t_o .

2. Kết hợp các kết quả được xác minh không gian, cùng với truy vấn ban đầu, để tạo thành một truy vấn được bổ sung chi tiết hoặc mô hình tiềm ẩn của đối tượng quan tâm. Truy vấn được làm giàu này bao toàn các quan hệ hình học của đối tượng được đề cập. Để kết hợp các kết quả đã kiểm tra, phép đồng nhất được phát hiện, H_i , được sử dụng để chiếu vùng truy vấn (hình chữ nhật do người dùng chọn) vào hình ảnh kết quả. Tất cả các dữ liệu trực quan trong vùng được chiếu này (bao gồm các dữ liệu đã khớp ban đầu cũng như các dữ liệu không khớp) sau đó được chiếu lại (bao gồm cả hình elip, v.v.) thành một hình ảnh truy vấn mới cùng với các hình ảnh truy vấn ban đầu. Thành phần

[x, y] của dữ liệu trực quan được biến đổi theo H_i^{-1} . Thành phần elip, $C = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ biến đổi theo $H^T C H^1$. Bởi vì mở rộng truy vấn khá nhạy cảm với các kết quả xác thực sai, chúng tôi chỉ bao gồm các tài liệu có $n_i > 20$. Theo kinh nghiệm, ngưỡng này bỏ qua hầu hết tất cả các kết quả dương tính giả trong khi vẫn cung cấp đủ tài liệu đã kiểm tra để thực hiện mở rộng tốt. Sự phân biệt không gian này được sử dụng cho tất cả các mô hình ngoại trừ phương pháp "Đường cơ sở mở rộng truy vấn"

3. Sử dụng truy vấn được làm giàu để truy vấn lại kho dữ liệu và truy xuất một bộ kết quả mới.

4. Lặp lại quá trình khi cần thiết, xen kẽ giữa sàng lọc mô hình và tái truy vấn



Hình 2.1: **Tổng quan về phương pháp mở rộng truy vấn**

3. Model type

Trong phần này mô tả một số phương pháp để tính toán các mô hình đối tượng tiềm ẩn. Chúng dựa trên các mô hình tổng quát của các đặc trưng và độ hội tụ của chúng, với mức độ phức tạp khác nhau.

Mỗi phương thức bắt đầu bằng cách tính toán truy vấn ban đầu Q0 bao gồm tất cả các dữ liệu trực quan nằm trong vùng truy vấn. Một mô hình tiềm ẩn sau đó được xây dựng từ các hình ảnh được xác minh và được trả về từ Q0, và một truy vấn mới Q1 hoặc một số truy vấn mới được phát hành. Điều này ngay lập tức làm tăng hai vấn đề:

(i) Việc này nên thực hiện bao lâu để một mô hình tiềm ẩn mới từ được xây dựng từ Q1 và một truy vấn kết quả truy vấn khác,...?

(ii) Làm thế nào các danh sách được xếp hạng được trả lại từ Q₀, Q₁,... được kết hợp? Chúng tôi khám phá cả hai câu hỏi này.

a) Methods

Các phương pháp có thể được chia thành những phương pháp có vấn đề về một query mới và các phương thức phát hành nhiều query. Trong trường hợp sau, cần phải kết hợp các danh sách xếp hạng được trả lại cho mỗi query

Query expansion baseline (Mở rộng truy vấn cơ sở)

Phương pháp này là một ứng dụng của việc mở rộng truy vấn như được sử dụng trong văn bản tự nhớ. Chúng tôi lấy kết quả M = 5 trên cùng từ kết quả truy vấn ban đầu (không có xác minh không gian), trung bình các vector tần số thuật ngữ được tính toán từ toàn bộ hình ảnh kết quả và truy vấn lại một lần. Kết quả của Q1 được thêm vào các kết quả của Q0 (Top 5).

Transitive closure expansion (Mở rộng đóng bắc cầu)

Một hàng đợi ưu tiên của hình ảnh được xác minh được khóa bởi số lượng inliers. Sau đó, một hình ảnh được chụp từ đầu hàng đợi và khu vực tương ứng với vùng truy vấn ban đầu được sử dụng để thực hiện một truy vấn mới. Kết quả được xác minh của truy vấn mở rộng được chèn vào hàng đợi chưa được chèn trước đó (một lần nữa theo thứ tự số lượng inliers). Quy trình lặp lại cho đến khi hàng đợi trống. Các hình ảnh trong kết quả cuối cùng theo cùng một thứ tự mà chúng nhập hàng đợi.

Average query expansion (Mở rộng truy vấn trung bình)

Một truy vấn mới được xây dựng bằng cách tính trung bình các kết quả xác định của truy vấn ban đầu. Đầu tiên, top m < 50 kết quả được xác minh hàng đầu trả về bởi công cụ tìm kiếm được chọn. Một Q_{Avg} truy vấn mới sau đó được hình thành bằng cách lấy trung bình của truy vấn ban đầu Q0 và kết quả m

$$d_{avg} = \frac{1}{m+1} \left(d_0 + \sum_{i=1}^m d_i \right),$$

trong đó d_0 là một vectơ TF được chuẩn hóa của vùng truy vấn và d_i là vectơ TF được chuẩn hóa của kết quả thứ i . Đối với mức trung bình này, chúng tôi lấy sự kết hợp của các đặc trưng của truy vấn ban đầu, kết hợp với các khu vực được chiếu ngược vào khu vực truy vấn bằng cách chuyển đổi ước tính của RANSAC. Đây là hình thức đơn giản nhất của mô hình tiềm ẩn vì không tính đến tính ổn định của các đặc trưng hoặc độ phân giải của hình ảnh. Một lần nữa, chúng tôi đã truy vấn lại một lần và kết quả của Q_{AVG} được thêm vào các kết quả của Q_0 .

Recursive average query expansion (Mở rộng truy vấn trung bình đệ quy)

Phương pháp này cải thiện phương pháp mở rộng truy vấn trung bình, bằng cách tạo các truy vấn đệ quy Q_i từ tất cả các kết quả xác định không gian được trả về. Phương pháp dừng lại một lần khi có nhiều hơn 30 hình ảnh được xác minh được tìm thấy hoặc sau khi không có hình ảnh mới nào được xác định tích cực.

Multiple image resolution expansion (Mở rộng nhiều độ phân giải hình ảnh)

Mô hình trong trường hợp này cũng tính đến xác suất quan sát một đặc trưng được đưa ra một hình ảnh của một đối tượng và độ phân giải của nó. Các đặc trưng bao gồm một khu vực nhỏ của đối tượng chỉ được nhìn thấy trong hình ảnh hoặc hình ảnh cận cảnh với độ phân giải cao. Tương tự, các đặc trưng bao gồm toàn bộ đối tượng không được nhìn thấy trên các chế độ xem chi tiết.

Hình ảnh tiềm ẩn được xây dựng như trước đây bằng cách chiếu ngược các vùng xác minh của Q_0 sử dụng các phép biến đổi H_i . Số lượng pixel của vùng được chiếu phụ thuộc vào độ phân giải của mỗi hình ảnh kết quả. Hình ảnh có độ phân giải trung bình được chọn làm hình ảnh tham chiếu độ phân giải và sự thay đổi tương đối của độ phân giải (đối với hình ảnh tham chiếu độ phân giải) được tính cho mỗi hình ảnh kết quả. Các dải phân giải được cung cấp bởi sự thay đổi độ phân giải tương đối như $(0, 4/5), (2/3, 3/2)$ và $(5/4, \infty)$.



Hình 2.2: Mở rộng truy vấn trong thực tế.

III. Cải thiện truy vấn Bag-of-Words (Làm thêm)

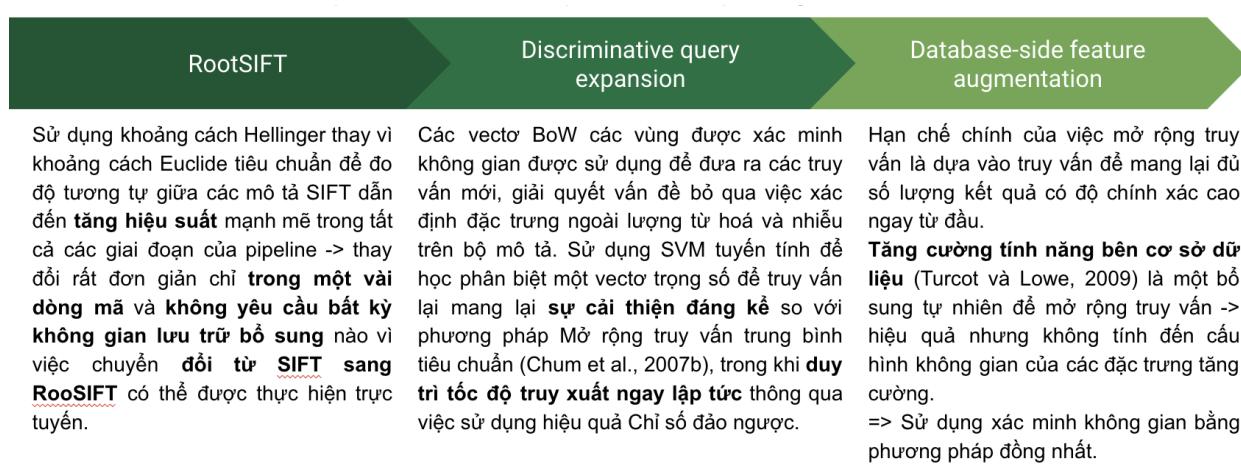
Phương pháp tiếp cận tiêu chuẩn của bài toán truy xuất đối tượng quy mô lớn gần thời gian thực:

- Biểu diễn một hình ảnh bằng cách sử dụng một Giỏ từ trực quan (BoW)
- Hình ảnh được xếp hạng bằng cách sử dụng thuật ngữ tần số nghịch đảo tần số tài liệu (tf-idf) -> Thông qua một chỉ mục nghịch đảo.

Một đối tượng trong hình ảnh đích có thể không được truy xuất vì một số lý do sử dụng tiêu chuẩn pipeline này:

- Dùng phát hiện đặc trưng.
- Mô tả nhiễu.
- Các thước đo không phù hợp để so sánh với bộ đặc tả.
- Mất do định lượng mô tả.

Chương này đề cập 3 nội dung chính sau:



Trong mỗi trường hợp, các phương pháp này tăng đáng kể hiệu suất truy xuất và có thể đơn giản được “kết nối vào” kiến trúc truy vấn đối tượng tiêu chuẩn của Philbin et al. (2007) (BoW, chỉ số đảo ngược, tf-idf, xếp hạng lại tính nhất quán trong không gian) mà không làm tăng thời gian xử lý.

Ví dụ: RootSIFT và mở rộng truy vấn phân biệt thậm chí không làm tăng yêu cầu lưu trữ.

1. Hệ thống truy xuất cơ sở (Baseline retrieval system)

Tác giả tuân theo framework truy vấn BoW tiêu chuẩn được mô tả trong (Philbin và cộng sự, 2007).

Sử dụng interest points affine-Hessian (Mikolajczyk và Schmid, 2004b), một vocabulary gồm 1 triệu vision words thu được bằng cách sử dụng giá trị K-means gần đúng (AKM) và xếp hạng lại không gian của 200 kết quả tf-idf hàng đầu bằng cách sử dụng phép biến đổi affine.

Việc triển khai hệ thống gần đây nhất của tác giả đạt được mAP là 0,672 trên bộ dữ liệu 5K của Oxford so với 0,657 ban đầu của Philbin và cộng sự (2007) => Đây là hệ thống cơ sở mà chúng tôi sẽ so sánh khi tác giả giới thiệu các phương pháp mới trong phần tiếp theo.

Triển khai gần đây nhất của tác giả về phương pháp mở rộng truy vấn trung bình từ Chum et al. (2007b) đạt được mAP là 0,726 trên Oxford 105K so với 0,711 ban đầu (Chum và cộng sự, 2007b).

Lưu ý: mặc dù bài báo gốc đã mô tả một số phương pháp để mở rộng truy vấn (ví dụ: đóng bắc cầu - transitive closure, nhiều độ phân giải hình ảnh - multiple image resolution), phương pháp trung bình đã trở thành tiêu chuẩn để so sánh với (Chum et al., 2011, Mikulik et al., 2010, Philbin et al., 2008) => nó có hiệu suất tương tự như các phương thức khác và thời gian chạy nhanh hơn vì các phương pháp khác liên quan đến việc đưa ra một số truy vấn mới. Do đó, tác giả sử dụng nó làm đường cơ sở để mở rộng truy vấn trong các so sánh tiếp theo.

Vì lý do nhất quán (sử dụng cùng một từ vựng trực quan và các tham số khác nhau của việc sắp xếp lại không gian và mở rộng truy vấn), tác giả so sánh các cải tiến của tác giả với việc triển khai hệ thống cơ sở gần đây nhất của tác giả.

2. RootSIFT: Khoảng cách Hellinger cho SIFT

Phương pháp này nổi tiếng với các lĩnh vực như phân loại kết cấu và phân loại hình ảnh, rằng việc sử dụng khoảng cách Euclid để so sánh histograms thường mang lại hiệu suất kém hơn so với sử dụng các thước đo như χ^2 (Chi - bình phương) hoặc Hellinger.

SIFT ban đầu được thiết kế để sử dụng với khoảng cách Euclidean (Lowe, 2004), nhưng vì nó là histogram nên câu hỏi tự nhiên nảy sinh là liệu nó có được lợi khi sử dụng các thước đo khoảng cách histogram thay thế hay không?

=> **Tác giả cho thấy rằng việc sử dụng nhân Hellinger thực sự mang lại một lợi ích to lớn.**

Giả sử x và y là n vectơ có đơn vị Euclid chuẩn ($\|x\|_2 = 1$), thì khoảng cách Euclid $d_E(x, y)$ giữa chúng liên quan đến độ tương tự của chúng (kernel) $S_E(x, y)$ là:

$$d_E(x, y)^2 = \|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^T y = 2 - 2S_E(x, y)$$

trong đó $S_E(x, y) = x^T y$, và bước cuối cùng sau từ $\|x\|_2^2 = \|y\|_2^2 = 1$. Ở đây tác giả quan tâm đến việc thay thế Euclidean similarity/kernel bằng Hellinger kernel.

=> Hữu ích khi sử dụng kết nối tiêu chuẩn giữa khoảng cách (số liệu) và kernels. Khoảng cách Euclid là sự dị biệt.

a) Hellinger kernel

Hellinger kernel, còn được gọi là hệ số Bhattacharyya, cho hai histograms chuẩn hóa L1, x và y (Ví dụ: $\sum_{i=1}^n x_i = 1$ và $x_i \geq 0$), được định nghĩa là:

$$H(x, y) = \sum_{i=1}^n \sqrt{x_i y_i}$$

Giả sử giá trị x,y bé thì nó sẽ ra kết quả số bé hơn x,y (0.9x0.8) khi căn ra thì sẽ ra giá trị lớn hơn (Căn của số bé hơn 1 sẽ ra lớn hơn số ban đầu). Điều này dẫn đến việc có quan tâm đến những tham số nhỏ nhặt.

Euclid thì không làm được. Cụ thể là có thành phần giá trị lớn, giá trị bé. Dẫn đến các giá trị bé sẽ không có tiếng nói trong kết quả chung cuộc.

Trong Chi-bình phương có chia cho mẫu số vì vậy sẽ tạo nên độ dị biệt cho mẫu số.

=> Hellinger là sự tương đồng

b) SIFT

Các vectơ SIFT có thể được so sánh bởi một nhân Hellinger bằng cách sử dụng một thao tác đại số đơn giản theo hai bước:

(i) L1 chuẩn hóa vectơ SIFT (ban đầu nó có đơn vị L2 chuẩn).

(ii) căn bậc hai mỗi phần tử.

Sau đó, $S_E(\sqrt{x}, \sqrt{y}) = \sqrt{x^T \sqrt{y}} = H(x, y)$, và các vectơ kết quả là L2 chuẩn hóa vì $S_E(\sqrt{x}, \sqrt{x}) = \sum_{i=1}^n x_i = 1$.

=> Định nghĩa một bộ đặc tả mới là RootSIFT - phần tử căn bậc hai của các vectơ SIFT chuẩn hóa L1.

Điểm mâu chốt: so sánh các bộ đặc tả RootSIFT sử dụng khoảng cách Euclidean tương đương với việc sử dụng nhân Hellinger để so sánh các vectơ SIFT ban đầu: $d_E(\sqrt{x}, \sqrt{y})^2 = 2 - 2H(x, y)$.

c) RootSIFT

RootSIFT được sử dụng trong đường dẫn truy xuất đối tượng cụ thể bằng cách thay thế SIFT bằng RootSIFT tại mọi điểm. Các bộ đặc tả RootSIFT được so sánh bằng cách sử dụng khoảng cách Euclidean.

=> Mọi bước có thể được sửa đổi dễ dàng:

- K-means để xây dựng từ vựng trực quan (vì nó dựa trên khoảng cách Euclid) -> vẫn có thể được sử dụng.
- Các phương pháp lân cận gần nhất (cần thiết cho các hệ thống có từ vựng rất lớn) -> vẫn có thể được sử dụng.
- Gán mềm các bộ mô tả cho các từ trực quan (Jégou và cộng sự, 2010a, Philbin và cộng sự, 2008)
- Mở rộng truy vấn và các phần mở rộng khác chỉ yêu cầu khoảng cách Euclidean trên SIFT (Jégou và cộng sự, 2008, 2010b, Mikulik và cộng sự, 2010, Philbin và cộng sự, 2010).

d) Hiệu suất truy xuất (mAP) của các phương pháp đề xuất khác nhau

Retrieval Method	SIFT		RootSIFT	
	Ox5k	Ox105k	Ox5k	Ox105k
Philbin et al. (2007): tf-idf ranking	0.636	0.515	0.683	0.581
Philbin et al. (2007): tf-idf with spatial reranking	0.672	0.581	0.720	0.642
Chum et al. (2007b): average query expansion (AQE)	0.839	0.726	0.850	0.756
Turcot and Lowe (2009): database-side feature augmentation (AUG)	0.776	0.711	0.827	0.759
This chapter: discriminative query expansion (DQE)	0.847	0.752	0.861	0.781
This chapter: spatial database-side feature augmentation (SPAUG)	0.785	0.723	0.838	0.767
This chapter: SPAUG + DQE	0.844	0.795	0.881	0.823

Tác giả sử dụng cách triển khai của tác giả đối với tất cả các phương pháp đã được liệt kê (Chum và cộng sự, 2007b, Philbin và cộng sự, 2007, Turcot và Lowe, 2009) để so sánh chúng một cách nhất quán bằng cách **sử dụng các từ vựng và bộ thông số trực quan giống nhau**.

=> RootSIFT **tốt hơn** đáng kể so với SIFT (**Nhưng thua rất nhiều so với ứng dụng trí tuệ nhân tạo**) cho tất cả các phương pháp được kiểm tra. Các từ vựng được tạo bằng cách sử dụng bộ mô tả Oxford 5K và tất cả các phương pháp ngoại trừ “xếp hạng tf-idf” đều sử dụng **sắp xếp lại không gian** của 200 kết quả hàng đầu.

Lưu ý: đối với AUG và SPAUG, tác giả tính toán lại idf ở phần sau.

e) Kết luận

Phép biến đổi RootSIFT có thể được coi là **một feature map rõ ràng** từ không gian SIFT ban đầu -> không gian RootSIFT. Sao cho việc thực hiện tích vô hướng (tức là một nhân tuyến tính) trong không gian RootSIFT tương đương với việc tính toán Hellinger kernel trong không gian gốc.

=> Cách tiếp cận này đã được khám phá trong bối cảnh của kernel map cho bộ phân loại SVM bởi (Perronnin và cộng sự, 2010b, Vedaldi và Zisserman, 2010). Những feature maps rõ ràng

có thể được xây dựng cho các kernels phụ khác, chẳng hạn như χ^2 (Chi - bình phương). Tuy nhiên, tác giả nhận thấy có chút khác biệt về hiệu suất so với Hellinger kernel khi được sử dụng trong hệ thống truy xuất đối tượng cụ thể.

Tác dụng của ánh xạ RootSIFT:

Giảm các giá trị **bin lớn** hơn so với các giá trị **bin nhỏ hơn**

Lý do: Khoảng cách Euclid giữa các vectơ SIFT ban đầu có thể **bị chi phối bởi các giá trị lớn** này. Sau khi ánh xạ, khoảng cách có tầm ảnh hưởng mạnh hơn với các giá trị bin nhỏ hơn.

Các công trình trước đây đã so sánh các vectơ SIFT với các khoảng cách khác Euclidean, nhưng không sử dụng feature map rõ ràng. Vì vậy, lợi ích của việc có thể tiếp tục sử dụng các thuật toán với khoảng cách Euclid (ví dụ: K-mean) là không rõ ràng.

Ví dụ:

Tác giả	Nội dung
Johnson (2010)	sử dụng phân kỳ của Jeffrey để so sánh các vectơ SIFT tập trung vào nén bộ mô tả
Pele và Werman (2008)	sử dụng biến thể của Khoảng cách của Earth Mover
Pele và Werman (2010)	số liệu bậc hai χ^2 (Chi - bình phương)

3. Mở rộng truy vấn mang tính phân biệt (Discriminative query expansion)

Mở rộng truy vấn có thể cải thiện đáng kể hiệu suất của hệ thống truy xuất. Phương pháp mở rộng truy vấn trung bình tiến hành như sau:

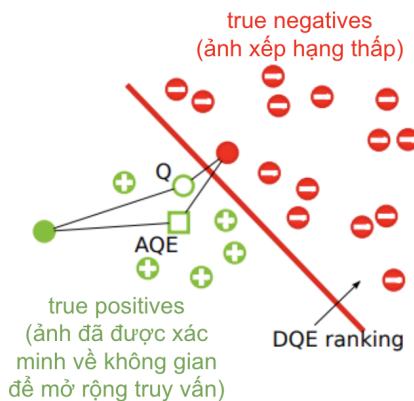
- Cho một vùng truy vấn, hình ảnh được xếp hạng bằng cách sử dụng điểm tf-idf.
- Xác minh không gian được thực hiện trên một danh sách ngắn các kết quả được xếp hạng cao, đồng thời cung cấp vị trí (ROI) của đối tượng truy vấn trong t hình ảnh được truy xuất.
- Các vectơ BoW tương ứng với các từ trong các ROI này được tính trung bình cùng với query BoW.
- Vectơ BoW mở rộng truy vấn kết quả này được sử dụng để tái truy vấn cơ sở dữ liệu.

Cách tiếp cận phân biệt để mở rộng truy vấn trong đó dữ liệu phủ định được tính đến và huấn luyện phân loại:

- Các vectơ BoW được sử dụng để làm giàu truy vấn **được thu thập theo cách chính xác** giống như đối với mở rộng truy vấn trung bình. Chúng cung cấp dữ liệu đào tạo tích cực và hình ảnh có điểm tf-idf thấp cung cấp dữ liệu đào tạo tiêu cực.
- Một SVM tuyến tính được đào tạo bằng cách **sử dụng các vectơ BoW dương và âm** này để **thu được vectơ trọng số w**.
- Vectơ trọng số đã học được sử dụng để **xếp hạng hình ảnh** theo khoảng cách của chúng từ ranh giới quyết định, tức là nếu hình ảnh được biểu diễn bằng vectơ BoW x , thì hình ảnh được sắp xếp theo giá trị $w^T x$. -> Sử dụng chỉ số đảo ngược giống như khi tính toán điểm tf-idf - cả hai phép toán chỉ là tích vô hướng giữa một vectơ và x .
- Đối với điểm tf-idf:

Phương pháp	Mở rộng truy vấn trung bình	Mở rộng truy vấn phân biệt (DQE)
Vector	BoW có trọng số idf truy vấn trung bình	Trọng số đã học w

Lưu ý: Để DQE hoạt động hiệu quả -> vector trọng số phải thừa thớt.



Q và AQE lần lượt biểu thị truy vấn và vectơ BoW mở rộng truy vấn trung bình. Xếp hạng tf-idf AQE sắp xếp hình ảnh dựa trên khoảng cách của chúng đến vectơ AQE, trong khi xếp hạng DQE sắp xếp hình ảnh theo khoảng cách đã ký từ ranh giới quyết định. Như minh họa ở đây, DQE xếp hạng chính xác hai hình ảnh có nhãn không xác định trong khi AQE thì không.

Bằng cách lựa chọn cẩn thận dữ liệu phủ định, vectơ trọng số thu được này ít nhất là thừa thớt như vectơ được sử dụng trong mở rộng truy vấn trung bình. Do đó, phương pháp này

ít nhất cũng hiệu quả về mặt tính toán như mở rộng truy vấn trung bình với chi phí đào tạo SVM tuyến tính không đáng kể. Hình minh họa bằng sơ đồ cách dữ liệu phủ định có thể mang lại lợi ích cho DQE so với việc mở rộng truy vấn trung bình.

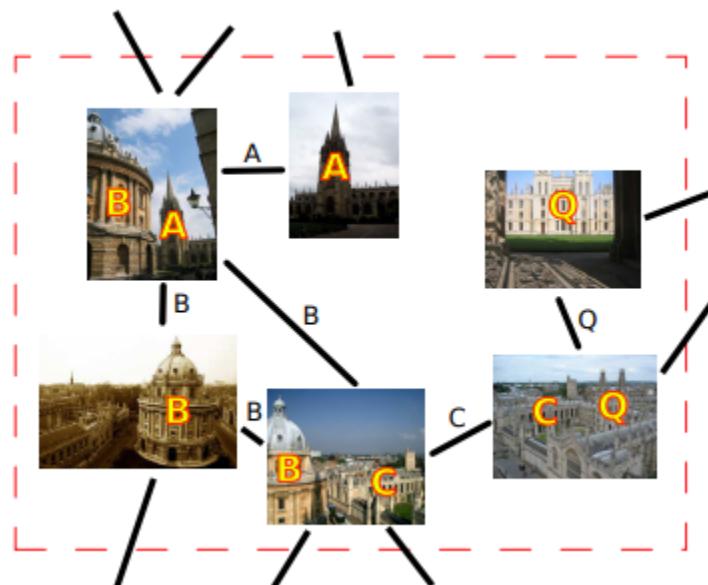
Retrieval Method	SIFT		RootSIFT	
	Ox5k	Ox105k	Ox5k	Ox105k
Philbin et al. (2007): tf-idf ranking	0.636	0.515	0.683	0.581
Philbin et al. (2007): tf-idf with spatial reranking	0.672	0.581	0.720	0.642
Chum et al. (2007b): average query expansion (AQE)	0.839	0.726	0.850	0.756
Turcot and Lowe (2009): database-side feature augmentation (AUG)	0.776	0.711	0.827	0.759
This chapter: discriminative query expansion (DQE)	0.847	0.752	0.861	0.781
This chapter: spatial database-side feature augmentation (SPAUG)	0.785	0.723	0.838	0.767
This chapter: SPAUG + DQE	0.844	0.795	0.881	0.823

=> Có thể thấy rằng DQE luôn tốt hơn so với AQE. Hiệu suất đạt được đặc biệt rõ ràng khi tăng kích thước tập dữ liệu - đối với Oxford 5K DQE tốt hơn AQE 1% và 1,3% đối với SIFT và RootSIFT, trong khi đối với Oxford 105k mAP cải thiện 3,6% và 3,3%.

Chi tiết triển khai

- Vectơ BoW tương ứng với mỗi hình ảnh trước tiên được cắt bớt để chỉ bao gồm các từ xuất hiện trong ít nhất một ví dụ tích cực.
- Bộ phân loại là một SVM tuyến tính được đào tạo với LIBSVM

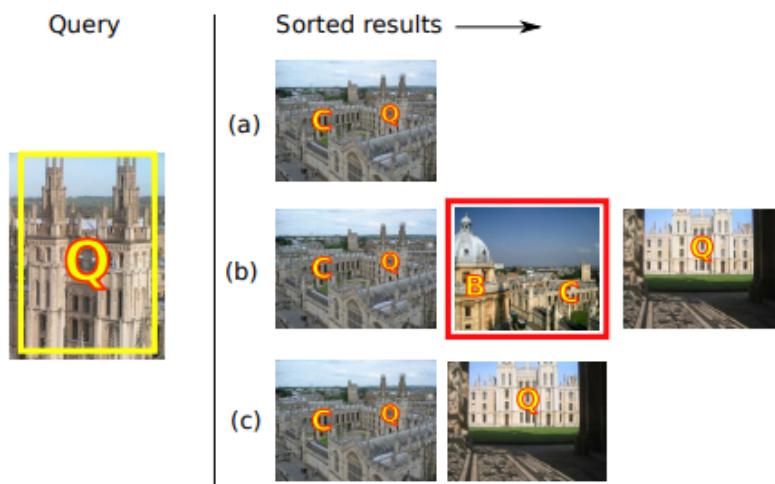
Thảo luận



Những từ khó hiểu này sau đó sẽ bị loại bỏ khi truy vấn lại, mặc dù “mở rộng” thực tế vẫn được thực hiện bằng cách lấy trung bình các vectơ BoW và áp dụng sơ đồ xếp hạng tf-idf. DQE đi xa hơn ở hai khía cạnh: đầu tiên nó học trọng số cho các từ tích cực (thay vì chỉ đơn giản là trung bình), và thứ hai nó có trọng số tiêu cực cho các từ khó hiểu (thay vì chỉ đơn giản là bỏ qua chúng).

4. Tăng cường đặc trưng cơ sở dữ liệu (Database-side feature augmentation)

Hiệu suất truy vấn của các phương pháp tăng cường:



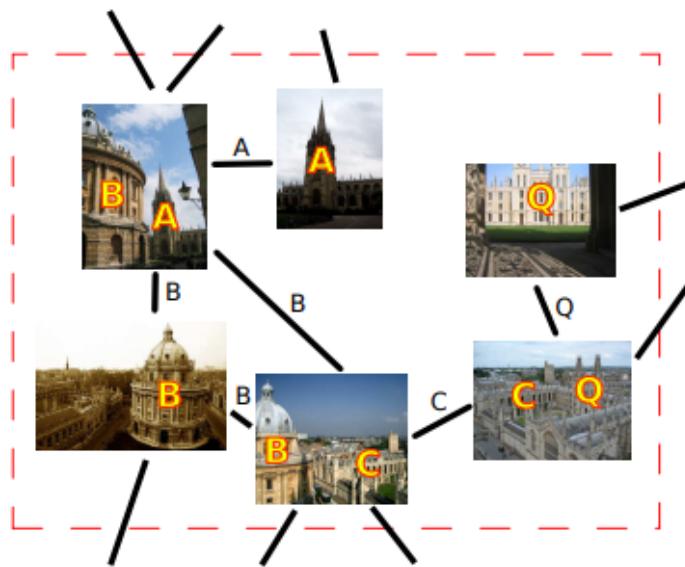
Đối tượng được truy vấn được tô sáng màu vàng trên hình ảnh ngoài cùng bên trái.

(a) Kết quả truy vấn tf-idf khi không dùng thông tin đồ thị ảnh, một hình ảnh trong thách thức của hệ thống truy vấn đã không được truy vấn

(b) Ảnh được truy vấn bằng việc sử dụng phương pháp của Turcot and Lowe (2009) và đồ thị được chỉ ra trong hình: Độ sót được tăng cường nhưng độ chính xác giảm trong các trường hợp dương tính giả (được đánh dấu màu đỏ)

(c) Phương pháp của chúng tôi cho thấy việc độ sót tăng lên khi duy trì độ chính xác cao vì hình ảnh chỉ được tăng cường với các từ trực quan từ các khu vực lân cận có liên quan.

Trừ khi hai hình ảnh gần giống nhau, một số lượng lớn các từ bổ sung sẽ thực sự không được nhìn thấy



Mô tả

Mặc dù rõ ràng có lợi ích truy xuất trong việc sử dụng tính đồng nhất không gian, nhưng sẽ có chi phí về yêu cầu lưu trữ bổ sung. Phương pháp gia tăng ban đầu của Turcot và Lowe (2009) không phát sinh chi phí này vì nó không cần phải tăng thêm vectơ BoW một cách rõ ràng trước khi tìm kiếm. Thay vào đó, tại thời điểm chạy, điểm tf-idf của sản phẩm vô hướng giữa truy vấn và hình ảnh tập dữ liệu được tính toán hiệu quả bằng cách sử dụng chỉ mục đảo ngược như bình thường, và sau đó nó được tăng lên bằng cách chỉ cần tính tổng điểm của các hình ảnh lân cận (lân cận theo biểu đồ phù hợp). Điều này tương đương với việc tăng các vectơ trước khi tính điểm tf-idf do tính phân phối của tích vô hướng. Tuy nhiên, điều này chỉ có thể thực hiện được vì tất cả các từ trực quan trong các hình ảnh lân cận đều được sử dụng để tăng thêm; tiện ích mở rộng của chúng tôi yêu cầu tăng cường rõ ràng vì một hình ảnh có thể đóng góp các từ khác nhau cho các hàng xóm khác nhau theo sự chồng chéo không gian. Điều này làm tăng yêu cầu lưu trữ vì chỉ số đảo ngược tăng lên, tuy nhiên nó đáng giá do cải thiện hiệu suất truy xuất. Lưu ý rằng đối với một bộ lưu trữ vectơ BoW cụ thể chỉ tăng khi một từ bổ sung chưa xuất hiện ở bất kỳ đâu trong hình ảnh tăng cường, như thể khi đó chỉ cần tăng số lượng của từ đó mà không ảnh hưởng đến kích thước chỉ mục đảo ngược.

Việc tăng cường xác minh không gian thêm trung bình 4,4 từ cho mỗi từ hiện có cho bộ dữ liệu Oxford 105K. Con số này ít hơn 28% so với phương pháp tăng cường ban đầu (Turcot và Lowe, 2009), và minh họa rằng phương pháp ban đầu thực sự giới thiệu một số lượng lớn các từ trực quan không liên quan và có thể gây bất lợi.

Chi tiết triển khai

Tác giả sử dụng cách tiếp cận của Philbin và Zisserman (2008) để xây dựng một biểu đồ hình ảnh phù hợp trong một bộ dữ liệu ngoại tuyến. Mỗi hình ảnh trong bộ dữ liệu được sử dụng làm truy vấn trong một hệ thống truy xuất đối tượng tiêu chuẩn (Philbin et al., 2007) và

một cạnh được xây dựng cho mỗi hình ảnh được xác minh theo không gian. Một phương pháp xây dựng đồ thị thay thế sử dụng băm (Chum và Matas, 2010a) có thể được sử dụng cho các bộ dữ liệu quy mô rất lớn trong đó truy vấn sử dụng mỗi hình ảnh lần lượt là không thực tế. Khi xây dựng biểu đồ, chúng tôi không bao gồm các hình ảnh truy vấn được sử dụng để đánh giá bộ dữ liệu để mô phỏng kịch bản thực tế trong đó hình ảnh truy vấn không được biết đến vào thời điểm tiền xử lý.

5. Kết luận và khuyến nghị cho thiết kế hệ thống truy xuất

RootSIFT: Sử dụng rootsift thay vì SIFT hiệu suất truy xuất được cải thiện trong mỗi thử nghiệm được tiến hành. Chúng tôi thực sự khuyên bạn nên được sử dụng vì nó cung cấp hiệu suất tăng miễn phí - rất dễ thực hiện, không tăng yêu cầu lưu trữ vì SIFT có thể được chuyển đổi thành Rootsift với chi phí tính toán không đáng kể.

Mở rộng truy vấn phân biệt (DQE): DQE luôn vượt trội so với mở rộng truy vấn trung bình (AQE). Nó hiệu quả như AQE vì đào tạo SVM là không đáng kể và trình điều khiển lại đòi hỏi các tài nguyên tính toán như nhau. Độ phức tạp thực hiện chỉ tăng nhẹ so với AQE do giai đoạn đào tạo bổ sung, tuy nhiên điều này không đáng kể vì nhiều gói SVM được công khai. Vì không có lập luận chống lại DQE, chúng tôi khuyên bạn nên sử dụng thay vì AQE trong mọi tình huống.

Tăng cường đặc trưng bên cơ sở dữ liệu (AUG): Aug là một phương pháp hữu ích để tăng độ sót. Nó không đòi hỏi tính toán trong thời gian chạy, nhưng nó đòi hỏi một giai đoạn xây dựng đồ thị tiền xử lý dài. Chúng tôi khuyên nó nên được sử dụng vì nó là một bổ sung tự nhiên để mở rộng truy vấn. Phần mở rộng của chúng tôi cho phương pháp cơ bản cải thiện độ chính xác nhưng tăng yêu cầu lưu trữ; Sự đánh đổi này nên được ghi nhớ khi quyết định có nên sử dụng nó hay không.

Gán mềm (soft assignment) bộ mô tả truy vấn: Việc gán mềm các bộ mô tả cho các từ trực quan làm giảm bớt các vấn đề do lượng tử hóa bộ mô tả gây ra ở một mức độ nào đó, nhưng trong cách triển khai ban đầu của Philbin et al. (2008) phép gán mềm đã được áp dụng cho cơ sở dữ liệu, do đó dẫn đến sự gia tăng lớn về yêu cầu lưu trữ vì các vectơ BoW đại diện cho hình ảnh do đó dày đặc hơn (so với việc sử dụng phép gán cứng). Thay vào đó, tác giả đề xuất Jégou et al. (2010a) và Jain et al. Cách tiếp cận của (2011) chỉ gán mềm các bộ mô tả truy vấn (không phải hình ảnh cơ sở dữ liệu) do đó không thay đổi các yêu cầu lưu trữ và chỉ làm tăng một chút thời gian xử lý truy vấn.

IV. Tài liệu tham khảo

- [1] [2010] Scale Object Retrieval Phd Thesis
- [2] [2013] Advancing Large Scale Object Retrieval

NHÓM 1 - INNOVATION

[3] [Bag of Visual Words in a Nutshell](#), Bethea Davida, Jul 3, 2018

[4] [k-means clustering](#), wikipedia