

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN

NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

Giảng viên hướng dẫn: **TS. Lê Đình Duy – TS. Nguyễn Tấn Trần Minh Khang**

Học viên: **Nguyễn Thanh Hòa** - Mã số học viên: **CH1601006**

Email: **hoant.11@grad.uit.edu.vn**

Lớp: **CH-KHMT khóa 2016**

Tháng 12/2017

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	3
LỜI CẢM ƠN	4
YÊU CẦU ĐỒ ÁN	5
I. MỤC TIÊU ĐỒ ÁN & PHÂN TÍCH THIẾT KẾ HỆ THỐNG	6
1. Mục tiêu đồ án	6
2. Lựa chọn nền tảng & thiết kế giao diện hệ thống	6
II. PHƯƠNG PHÁP VÀ KỸ THUẬT ÁP DỤNG	8
1. Quy trình xây dựng và cài đặt hệ thống	8
2. Phương pháp áp dụng	9
III. XÂY DỰNG HỆ THỐNG	10
1. Thu thập bộ dữ liệu và tạo image set	11
2. Lựa chọn phương pháp rút trích đặc trưng để tạo bagOfFeatures	12
3. Đánh chỉ mục tìm kiếm cho các ảnh (index images)	12
4. Tìm kiếm các ảnh phù hợp khi truy vấn	13
5. Xây dựng giao diện hệ thống bằng GUIDE và tích hợp chức năng	13
6. Xây dựng các tính năng mở rộng và hoàn thiện	14
IV. KẾT QUẢ & ĐÁNH GIÁ	17
1. Kết quả	17
2. Đánh giá	21
3. Hướng phát triển	21
BÁO CÁO MỞ RỘNG	22
TÀI LIỆU THAM KHẢO	24
PHỤ LỤC	25
1. SourceCode của ứng dụng và các bộ Dataset dùng trong đồ án	25
2. Hàm SURFBagOfFeaturesExtractor	25
3. Các Callback trong GUI của hệ thống	26

[illegible]

LỜI CẢM ƠN

Tác giả xin trân trọng cảm ơn **TS. Lê Đình Duy** và **TS. Nguyễn Tấn Trần Minh Khang** đã tận tình truyền đạt các kiến thức giá trị qua môn học *Nhận dạng thị giác và ứng dụng* để giúp đồ án này được hoàn thành tốt đẹp.

Xuất phát từ học viên chưa có nhiều kiến thức nền tảng về xử lý ảnh, thị giác máy tính, nhận dạng thị giác, tác giả đã được tiếp cận những kiến thức mới rất bổ ích và có khả năng ứng dụng thực tế qua những buổi học, thực hành trực tiếp. Tác giả đã vận dụng các kiến thức đã học cũng như tham khảo để thực hiện đồ án môn học với *dạng 4: Xây dựng hệ thống tìm kiếm hình ảnh (Image Retrieval) với các phương pháp dùng BoW*.

Do kiến thức tổng quan và thời gian nghiên cứu chuyên sâu, phần cứng máy tính còn khá hạn chế, nên một số vấn đề phức tạp chưa thể giải quyết hoàn chỉnh mặc dù đã tập trung nghiên cứu, thử nghiệm. Rất mong các thầy thông cảm.

Một lần nữa xin cảm ơn các thầy đã nhiệt tình hướng dẫn để đồ án này được hoàn thành cũng như truyền đạt các kiến thức, kinh nghiệm quý báu trong quá trình nghiên cứu khoa học và sắp đến.

Trân trọng.

Thành phố Hồ Chí Minh, tháng 12 năm 2017

Nguyễn Thanh Hòa

YÊU CẦU ĐỒ ÁN

Dạng 4:

Bài toán: *Tìm kiếm ảnh (image retrieval).*

Yêu cầu: *Học viên tìm hiểu các phương pháp dùng BoW.*

Ví dụ, tham khảo matlab: <https://www.mathworks.com/help/vision/ug/image-retrieval-with-bag-of-visual-words.html>

I. MỤC TIÊU ĐỒ ÁN & PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. Mục tiêu đồ án

Dựa vào các yêu cầu đã cho của đồ án, tác giả sẽ tiến hành cài đặt một hệ thống tìm kiếm hình ảnh với các mục tiêu sau:

- Sử dụng bộ dữ liệu **The Paris Dataset** đầy đủ gồm **6392** ảnh được cung cấp tại <http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/> [1].
Bộ ảnh này đã được xử lý, giải nén – đặt trong thư mục dataset_paris và upload tại <https://drive.google.com/file/d/1nqgNIB7zQeur0FZFMKx0b6gBihTbo8gR/view?usp=sharing>
- Cung cấp giao diện người dùng trực quan với module nhập ảnh query và danh sách ảnh kết quả.
- Cho phép xem chi tiết các ảnh kết quả, điều chỉnh số lượng kết quả trả về, tìm ảnh phù hợp nhất, đánh giá và so sánh mức độ khác biệt so với ảnh query.
- Cho phép người dùng lựa chọn một vùng ảnh thay vì toàn bộ ảnh làm query.
- Đánh giá kết quả của hệ thống.

Hệ thống tham khảo trên demo *Web-scale particular object search* của nhóm VGG tại <http://www.robots.ox.ac.uk/~vgg/research/parisings/index.html>

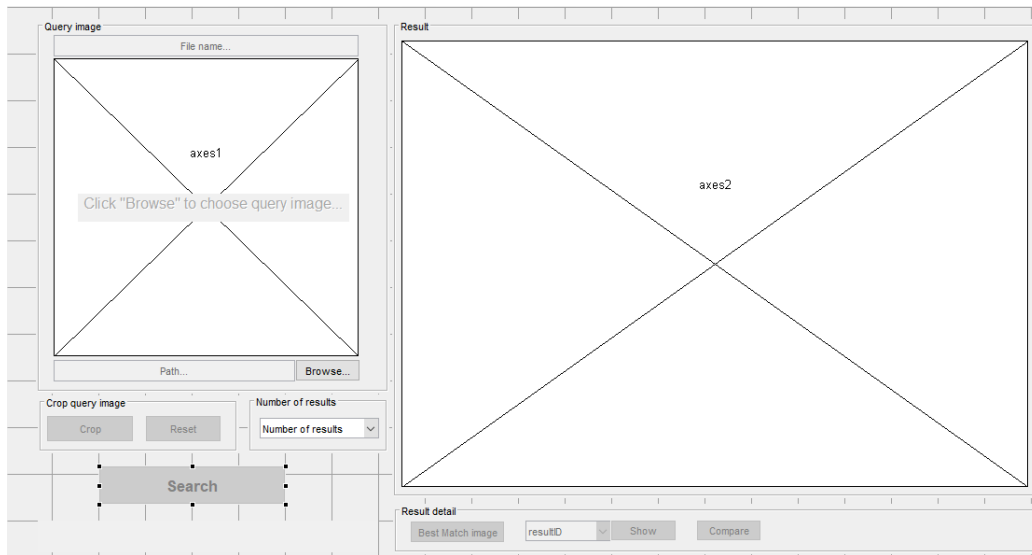
2. Lựa chọn nền tảng & thiết kế giao diện hệ thống

Tác giả lựa chọn xây dựng ứng dụng trên Matlab version 2017a với sự hỗ trợ của bộ công cụ **Computer Vision System Toolbox (CVST)**¹, đây là bộ công cụ rất mạnh mẽ của Matlab để hỗ trợ thiết kế và xây dựng các hệ thống liên quan đến thị giác máy tính, tìm kiếm nhận dạng ảnh số, video,...

Giao diện ứng dụng được xây dựng trên môi trường **GUIDE**² của Matlab và đáp ứng các yêu cầu đã cho của đồ án, trong đó:

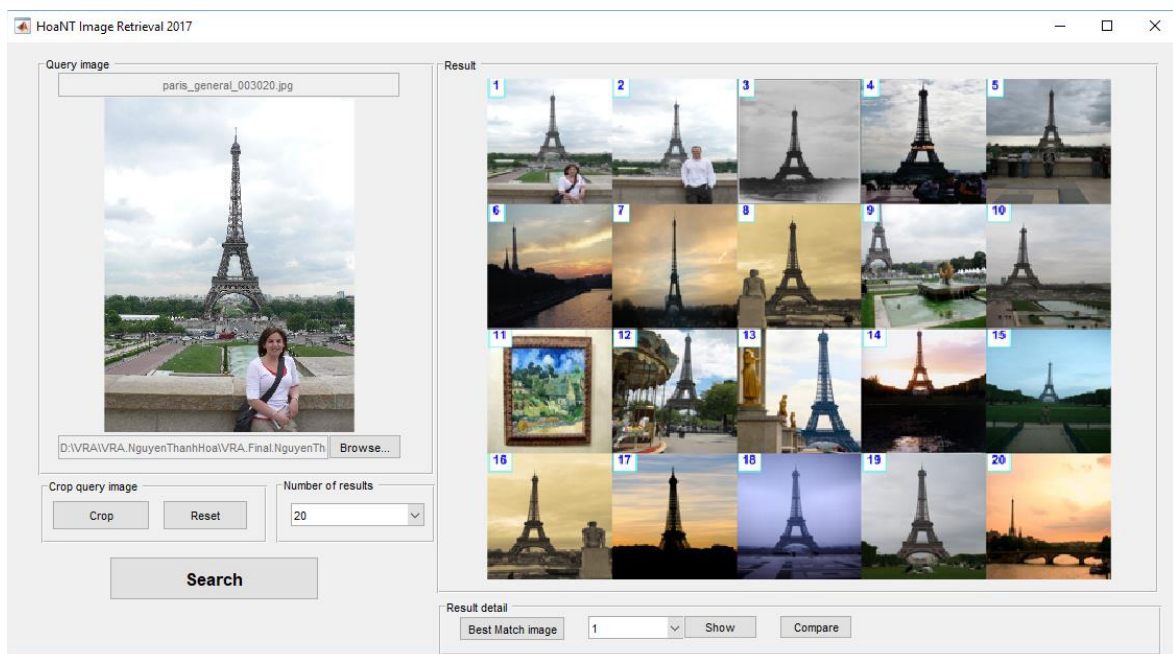
¹ <https://www.mathworks.com/help/vision/index.html>

² <https://www.mathworks.com/help/matlab/guide-or-matlab-functions.html>



Hình 1. Thiết kế giao diện ứng dụng

- Ảnh truy vấn (*Query image*) và ảnh kết quả (*Result images*) sử dụng **Axes** của Matlab GUIDE.
- Nhóm chức năng **Crop**: Hệ thống cho phép người dùng tìm kiếm dựa trên 1 vùng ảnh làm query thay vì toàn bộ ảnh (*yêu cầu của nhóm 2 thành viên*)
- Tùy biến: Tùy chọn số ảnh kết quả mong muốn trả về (*Number of results*), xem chi tiết từng ảnh kết quả, so sánh với ảnh query,...

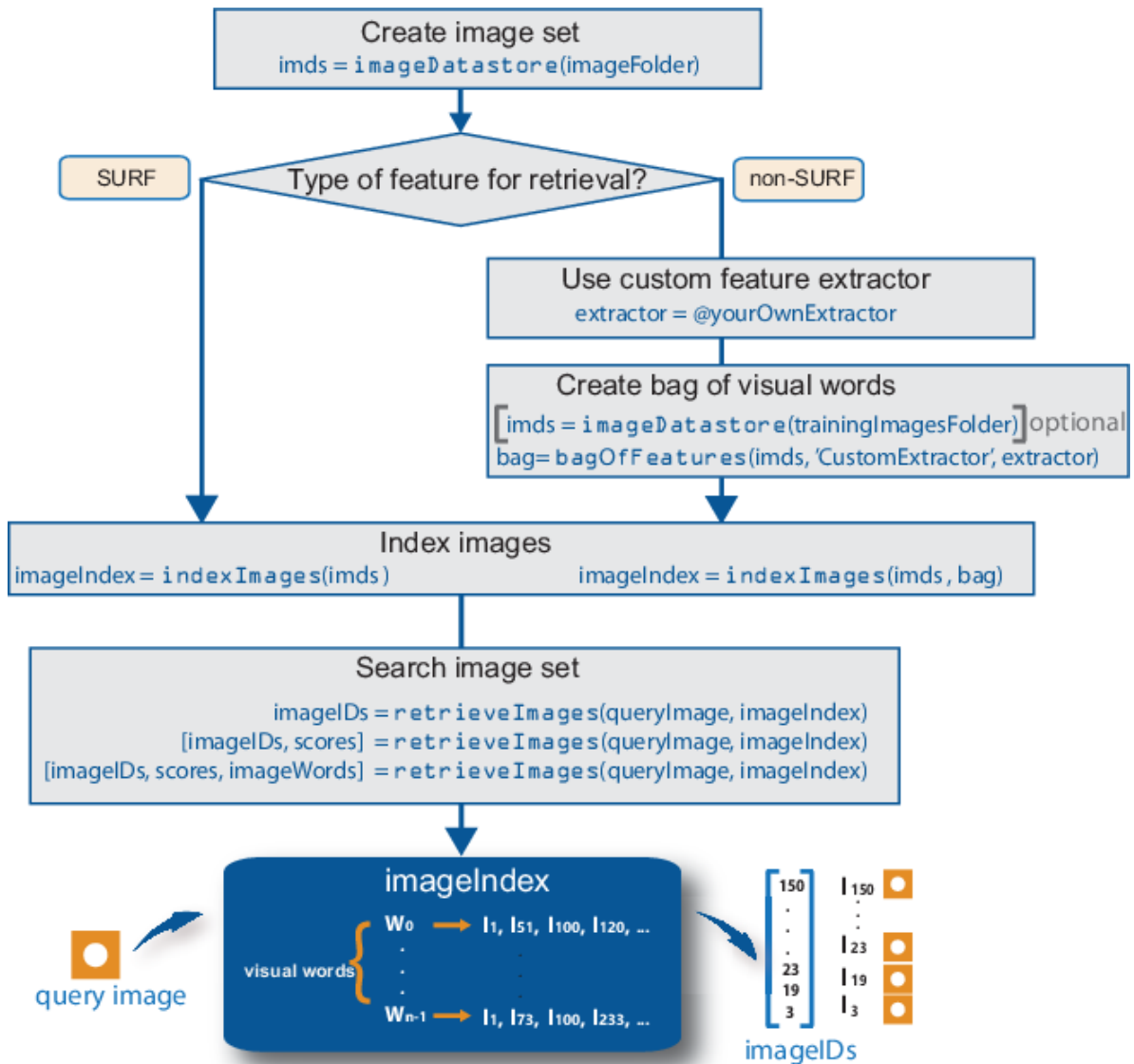


Hình 2. Kết quả hệ thống dự kiến

II. PHƯƠNG PHÁP VÀ KỸ THUẬT ÁP DỤNG

1. Quy trình xây dựng và cài đặt hệ thống

Tham khảo [2], quy trình để xây dựng hệ thống truy vấn hình ảnh (Image Retrieval) được thực hiện với bộ công cụ CVST sẽ được thực hiện theo quy trình sau:



Hình 3. Quy trình xây dựng hệ thống truy vấn hình ảnh bằng CVST [2]

Dựa trên quy trình chung này, tác giả xây dựng hệ thống theo các bước sau:

• Bước 1: Thu thập bộ dữ liệu và tạo image set

Sử dụng hàm `imageDataStore` để tạo image data set dựa trên bộ dữ liệu đã chọn cho đồ án này là **The Paris Dataset** (6392 ảnh ~ 2.44 GB). [1]

- **Bước 2: Lựa chọn phương pháp rút trích đặc trưng để tạo bagOfFeatures [3]**

Mặc định CVST sẽ sử dụng phương pháp **SURF** (Speeded Up Robust Features) để tạo bộ bagOfFeature. Ngoài ra, có thể tùy biến bộ rút trích đặc trưng khác qua việc thêm tham số 'CustomExtractor'.

- **Bước 3: Đánh chỉ mục tìm kiếm cho các ảnh (index images)**

Sử dụng hàm **indexImages** [4] để tạo chỉ mục tìm kiếm nhằm xác định visual words tương ứng với vị trí xuất hiện của nó trong image set ban đầu.

Thời gian thực hiện bước 2 và bước 3 khá lâu.

- **Bước 4: Tìm kiếm các ảnh phù hợp khi truy vấn**

Sử dụng hàm **retrieveImages** để tìm kiếm các ảnh phù hợp dựa vào ảnh query cung cấp và **indexImages** với các visual words đã tạo ở bước 2,3. Hệ thống sẽ trả về mặc định 20 ảnh kết quả nhưng có thể tùy biến số kết quả này với thông số NumResults

- **Bước 5: Xây dựng giao diện hệ thống bằng GUIDE và tích hợp chức năng**

Sau khi kiểm tra kết quả thành công, dựa trên phân tích thiết kế từ đầu, tác giả tiến hành xây dựng giao diện ứng dụng trên GUIDE và tích hợp các chức năng đã thử nghiệm thành công ở các bước trên vào giao diện.

- **Bước 6: Xây dựng các tính năng mở rộng và hoàn thiện**

Bổ sung các chức năng mở rộng: Cho phép chọn 1 vùng ảnh làm ảnh query thay vì toàn bộ ảnh, tùy chỉnh số ảnh trả về, gán nhãn cho các ảnh kết quả & cho phép xem chi tiết, so khớp với ảnh query,...

2. Phương pháp áp dụng

Lựa chọn phương pháp rút trích đặc trưng **SURF** khi xây dựng bộ BagOfFeatures:

```
% Create a custom bag of features using the 'CustomExtractor'
if ~exist('parisBagOfFeatures.mat', 'file') % Create a custom bag of
features only once
    bag = bagOfFeatures(trainingSet, ...
        'CustomExtractor', @SURFBagOfFeaturesColorExtractor, ...
        'VocabularySize', 10000);
    save('parisBagOfFeatures.mat', 'bag');
end
```

Đây là phương pháp được mặc định trong CVST để xây dựng `bagOfFeatures`³ vì tính phổ biến và hiệu quả, nhanh chóng. Trong đồ án, tác giả lựa chọn SURF kèm theo một số tùy biến để tăng hiệu quả trong việc tìm kiếm (tham khảo [2]):

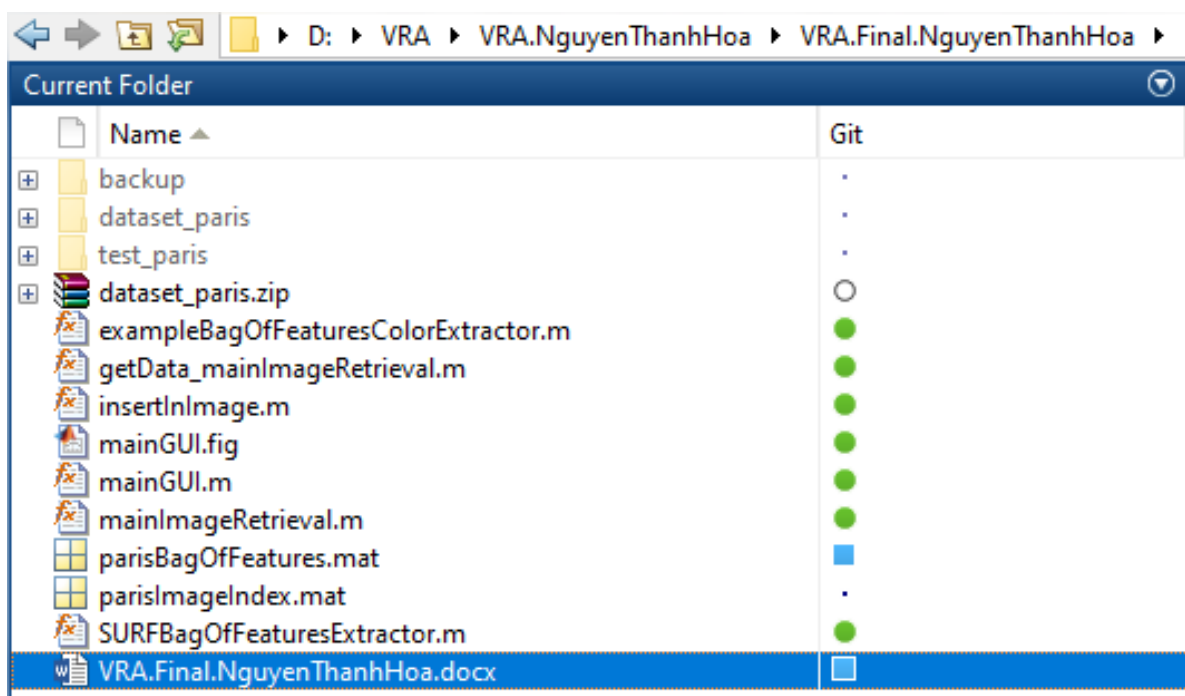
- Kích thước bộ từ điển (**VocabularySize**): **10000** (mặc định chỉ có 500)
- Tùy biến SURF sử dụng thông số **CustomExtractor** với nội dung cụ thể của **@SURFBagOfFeaturesColorExtractor** được thể hiện trong Phụ lục.

Ngoài ra, cũng có thể lựa chọn các phương pháp khác như Histogram of Gradients (HOG), Local Binary Patterns (LBP), Maximally Stable Extremal Regions (MSER),...

III. XÂY DỰNG HỆ THỐNG

Nội dung trong phần này sẽ trình bày cụ thể quá trình hiện thực các bước cài đặt và xây dựng hệ thống đã đưa ra tại phần trước trên Matlab 2017a.

Tổ chức code của hệ thống gồm:



Hình 4. Tổ chức các file của hệ thống

Trong đó:

- **dataset_paris**: chứa tập dữ liệu Paris dataset với 6392 ảnh (tập ảnh training)
- **test_paris**: chứa tập dữ liệu test dùng làm ảnh query kiểm tra kết quả (tập ảnh test)

³ <https://www.mathworks.com/help/vision/ref/bagoffeatures-class.html>

- File **parisBagOfFeatures.mat** và **parisImageIndex.mat** lưu trữ bộ BagOfFeatures và Image Index để dùng lần sau (chỉ cần chạy 1 lần đầu)
- **SURFBagOfFeaturesExtractor.m**: Bộ rút trích đặc trưng SURF
- **mainGUI** (.fig và .m): Giao diện và các Callback trên giao diện (dùng GUIDE)
- **mainImageRetrieval**: file thực thi chính của hệ thống
- **insertInImage**: Sử dụng module InsertInImage để chèn nhãn thứ tự cho ảnh (tham khảo [6])

1. Thu thập bộ dữ liệu và tạo image set

Bộ dữ liệu **The Paris Dataset** (6392 ảnh ~ 2.44 GB) được đặt tên tương ứng 12 chủ đề.

- | | |
|------------------------------|---------------------------|
| 1. La Defense Paris | 7. Notre Dame Paris |
| 2. Eiffel Tower Paris | 8. Pantheon Paris |
| 3. Hotel des Invalides Paris | 9. Pompidou Paris |
| 4. Louvre Paris | 10. Sacre Coeur Paris |
| 5. Moulin Rouge Paris | 11. Arc de Triomphe Paris |
| 6. Musee d'Orsay Paris | 12. Paris |

Để đảm bảo có bộ dữ liệu, trong mainImageRetrieval.m, đoạn code sau sẽ kiểm tra bộ dữ liệu đã có chưa và tải – giải nén nếu chưa có, sau đó dùng hàm imageDatastore để tạo image set dựa trên tập ảnh này được lưu tại thư mục **dataset_paris**

```
% Location of the compressed data set
url1 = 'http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/paris_1.tgz';
url2 = 'http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/paris_2.tgz';
% Store the output in a dataset_paris folder
outputFolder = fullfile('', 'dataset_paris'); % define output folder

% If it didn't exist dataset folder, download and untar dataset once
if ~exist(outputFolder, 'dir')
    waitbar(1/steps,h,'Downloading Paris Dataset...');
    untar(url1, outputFolder);
    untar(url2, outputFolder);
end
imageDataset =
imageDatastore(fullfile(outputFolder), 'LabelSource', 'foldernames');
```

2. Lựa chọn phương pháp rút trích đặc trưng để tạo bagOfFeatures

Như đã đề cập, tác giả sử dụng phương pháp SURF. Trong trường hợp sử dụng toàn bộ 6392 ảnh làm dữ liệu training thì trainingSet = imageImageSet:

```
trainingSet = imageDataset;
% Create a custom bag of features using the 'CustomExtractor' option
if ~exist('parisBagOfFeatures.mat', 'file') % Create a custom bag of features
only once
    disp(' Create a custom bag of features...');
    waitbar(3/steps,h,'Create a custom bag of features using the
CustomExtractor...');
    bag = bagOfFeatures(trainingSet, ...
        'CustomExtractor', @SURFBagOfFeaturesColorExtractor, ...
        'VocabularySize', 10000);
    save('parisBagOfFeatures.mat', 'bag');
end
% Load pre-trained bagOfFeatures if it exist
load('ParisBagOfFeatures.mat', 'bag');
```

Sử dụng hàm **save** để lưu lại BagOfFeatures để không mất thời gian chạy lại lần sau, chỉ cần dùng hàm **load** để nạp BagOfFeatures qua biến **bag**.

```
Creating Bag-Of-Features.
-----
* Image category 1: dataset_paris
* Extracting features using a custom feature extraction function: SURFBagOfFeaturesExtractor.
* Extracting features from 6392 images...done. Extracted 9589388 features.
```

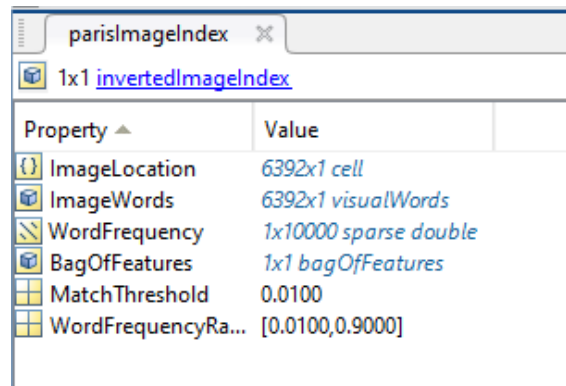
Hình 5. Với 6392 ảnh, có 9.589.388 feature được rút trích bằng SURF

3. Đánh chỉ mục tìm kiếm cho các ảnh (index images)

Sử dụng hàm indexImages và image set, bagOfFeatures đã tạo ở bước 1, 2 để đánh chỉ mục (Index) và lưu vào file để sử dụng lần sau.

```
if ~exist('parisImageIndex.mat', 'file') % Create a search index only once
    disp('Create a search index...');
    waitbar(4/steps,h,'Create a search index...');
    parisImageIndex = indexImages(imageDataset, bag);
    save('parisImageIndex.mat', 'parisImageIndex');
end
load('parisImageIndex.mat', 'parisImageIndex');
```

Quá trình thực hiện bước 2 và 3 mất nhiều thời gian nhất. Với tập dữ liệu Paris Dataset và dùng máy tính Windows 10, Core i7, RAM 8GB và Matlab 2017a mất khoảng 6 giờ để hoàn thành BagOfFeatures và ImageIndex.



Property	Value
ImageLocation	6392x1 cell
ImageWords	6392x1 visualWords
WordFrequency	1x10000 sparse double
BagOfFeatures	1x1 bagOfFeatures
MatchThreshold	0.0100
WordFrequencyRa...	[0.0100, 0.9000]

Hình 6. File `parisImageIndex.mat` sau khi hoàn thành bước 3

4. Tìm kiếm các ảnh phù hợp khi truy vấn

Sử dụng hàm `retrieveImages`⁴ để thực hiện truy vấn ảnh dựa trên ảnh query, `ImageIndex` đã xây dựng, số kết quả trả về.

```
[imageIDs,scores,queryWords] = retrieveImages(queryImage,
parisImageIndex, 'NumResults',numResult);
```

Các thông tin trả về bao gồm:

- + `imageIDs`: chứa danh sách ID các ảnh phù hợp
- + `scores`: thể hiện mức độ khác biệt giữa ảnh query và ảnh kết quả.
- + `queryWords`: chứa visual word của ảnh query.

Việc xử lý hiển thị kết quả hình ảnh và gán nhãn được thực hiện thông qua đoạn code sau:

```
% Show result
thumbnails = [];
result = imageDatastore.Files(imageIDs);
% Labeling result image with insertInImage
for i = 1:numel(result)
    img = imread(result{i});
    thumbnails = cat(4, thumbnails,...
        insertInImage(imresize(img, [200 200]),...
            @()text(10,10,num2str(i)),...
            {'fontweight','bold','color','b','fontsize',15,'margin',5,'edgecolor',[0 1
            1],'backgroundcolor','w'}));
end
```

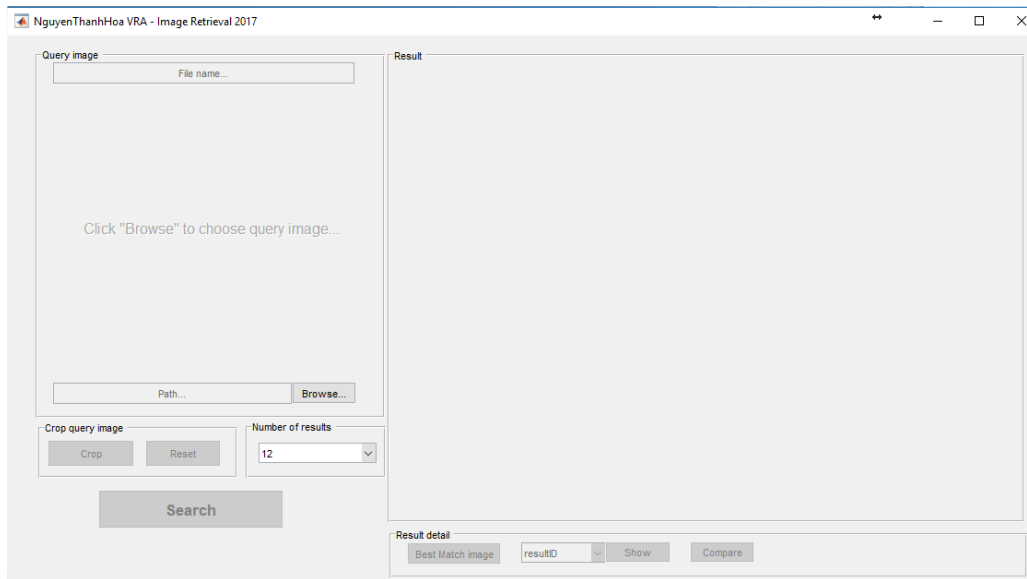
Trong đó có sử dụng module `insertInImage` [6] để chèn nhãn lên ảnh, sau đó chuyển biến `result` và `thumbnails` sang GUI để xử lý hiển thị kết quả trên **Axes** theo chế độ **montage**.

5. Xây dựng giao diện hệ thống bằng GUIDE và tích hợp chức năng

⁴ <https://www.mathworks.com/help/vision/ref/retrieveimages.html>

Giao diện chính của hệ thống xây dựng bằng GUIDE trên Matlab

- + Axes: 2 axes thể hiện Query Image và toàn bộ kết quả
- + Các thành phần còn lại sử dụng các object tương ứng được GUIDE hỗ trợ để thể hiện và tùy biến việc hiển thị phù hợp khi sử dụng thực tế.

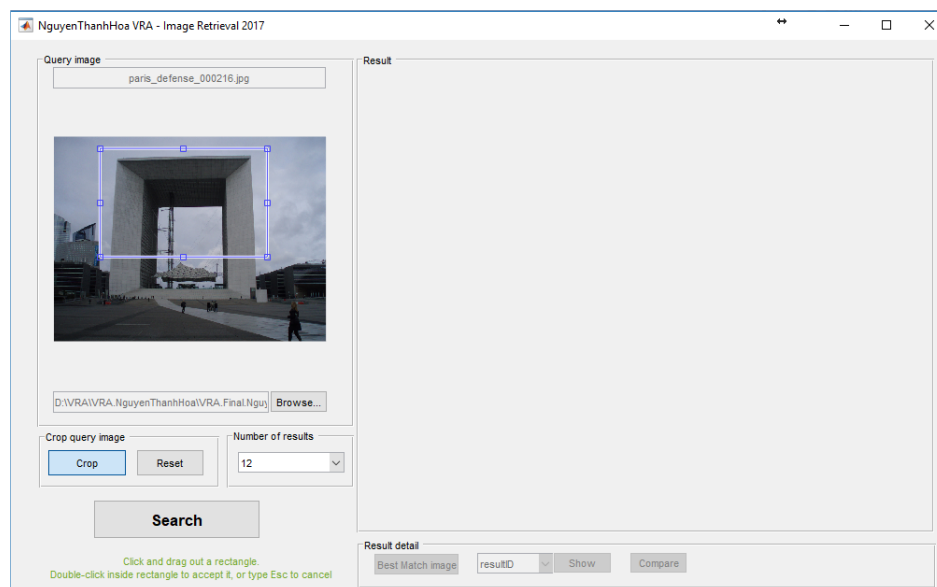


Hình 7. Giao diện hệ thống lúc khởi động

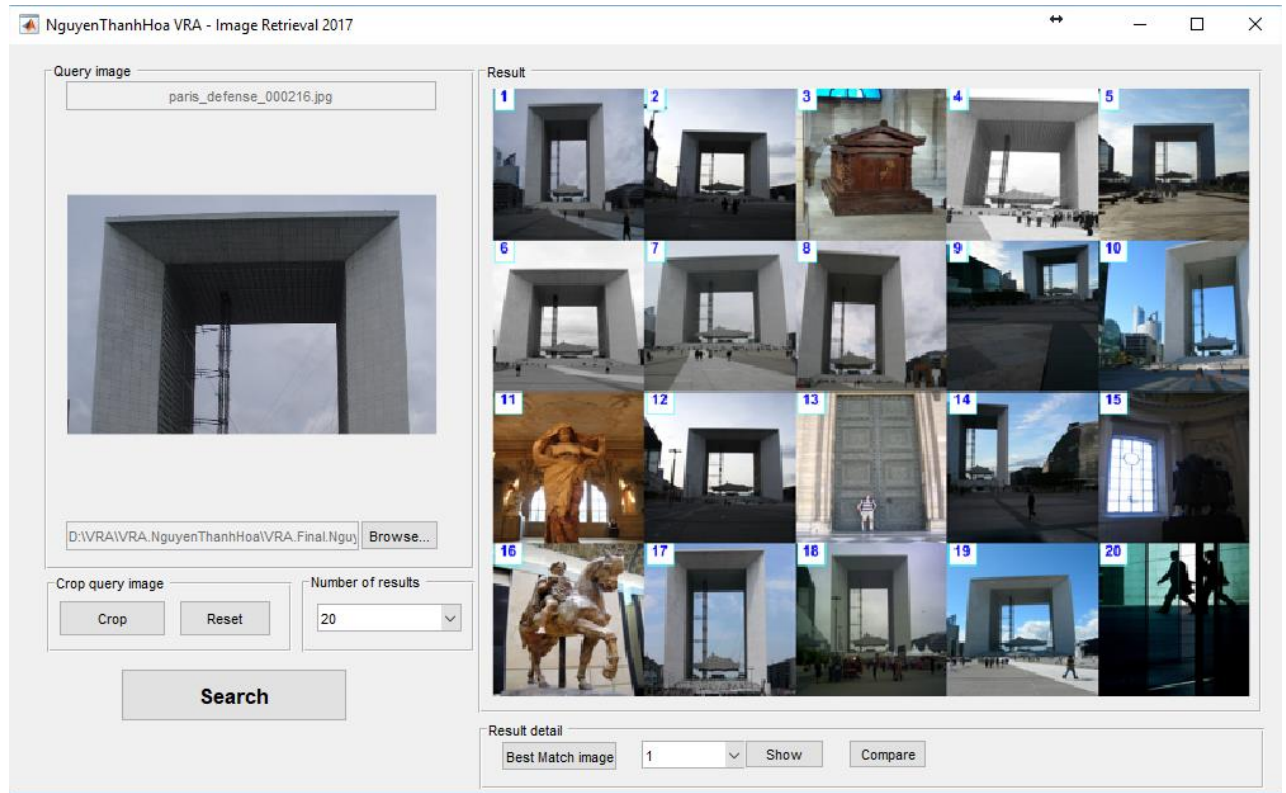
Xây dựng các Callback tương ứng (trình bày trong Phụ lục)

6. Xây dựng các tính năng mở rộng và hoàn thiện

- Chọn 1 vùng ảnh để làm ảnh query: sử dụng hàm **imrect** và **imcrop**



Hình 8. Chọn 1 vùng trên ảnh để làm ảnh query, doubleclick để Crop, ESC để hủy



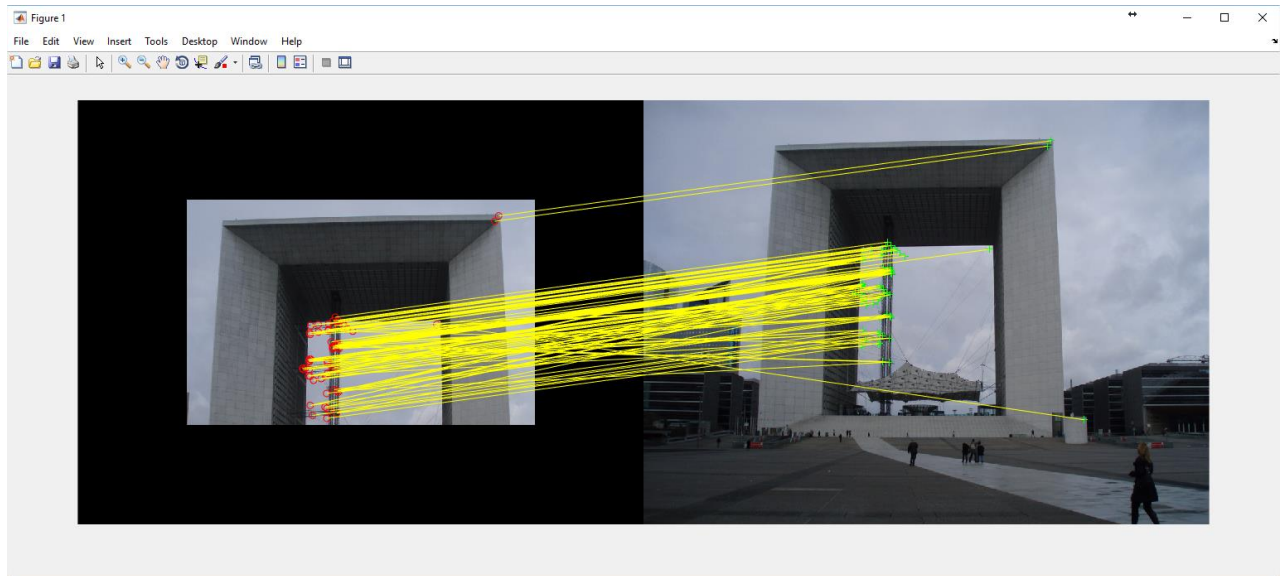
Hình 9. Kết quả tìm kiếm khi dùng 1 vùng ảnh làm query

- So sánh Visual Words của ảnh kết quả và ảnh query:

```
bestMatchWords = handles.parisImageIndex.ImageWords(currentImage);
bestImage = imread(handles.parisImageIndex.ImageLocation{currentImage});
queryWordsIndex = handles.queryWords.WordIndex;
bestMatchWordIndex = bestMatchWords.WordIndex;

tentativeMatches = [];
for i = 1:numel(handles.queryWords.WordIndex)
    idx = find(queryWordsIndex(i) == bestMatchWordIndex);
    matches = [repmat(i, numel(idx), 1) idx];
    tentativeMatches = [tentativeMatches; matches];
end
points1 = handles.queryWords.Location(tentativeMatches(:,1),:);
points2 = bestMatchWords.Location(tentativeMatches(:,2),:);
figure
showMatchedFeatures(handles.myImage,bestImage,points1,points2,'montage')
;
```

Dựa vào queryWords thu thập khi dùng hàm retrieveImages và visual word của ảnh kết quả để thực hiện đối chiếu.



Hình 10. Ví dụ đối chiếu kết quả ảnh query và ảnh tốt nhất trả về

- Tính toán Average Precision để đánh giá kết quả:

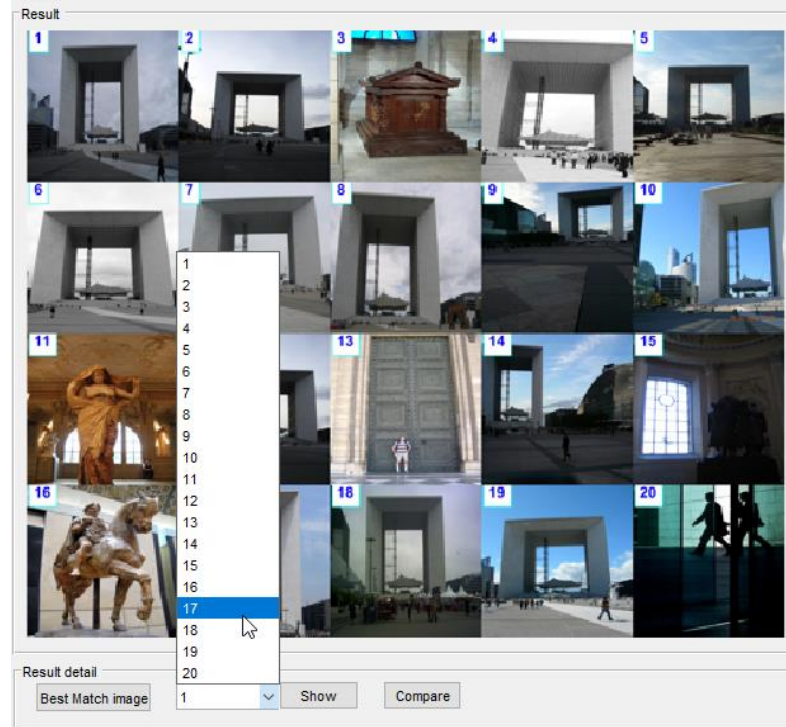
```
expectedID = [1 85];
[averagePrecision,actualIDs] = evaluateImageRetrieval(queryImage,...
    parisImageIndex,expectedID);

fprintf('Average Precision: %f\n\n',averagePrecision)
```

Để xác định được thông số này, cần cung cấp expectedID gồm danh sách ID các ảnh kết quả mong muốn được trả về tương ứng. (tham khảo [7])

- Hiển thị chi tiết ảnh kết quả mong muốn:

Do khi sử dụng 1 Axes để hiển thị toàn bộ ảnh theo chế độ montage, việc xử lý click vào từng ảnh để hiển thị chi tiết khá khó khăn, do đó bổ sung thêm việc gán nhãn tương ứng cho các ảnh kết quả và cho phép hiển thị ảnh mong muốn chi tiết ở figure mới:



Hình 11. Tùy chọn hiển thị ảnh kết quả

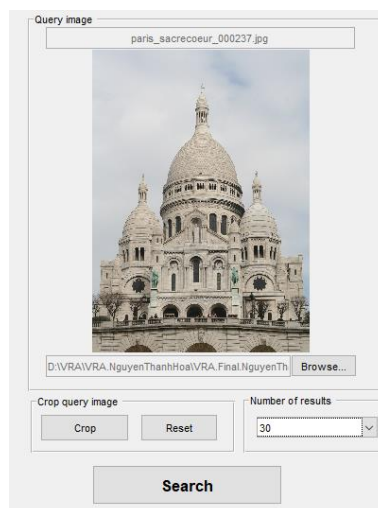
IV. KẾT QUẢ & ĐÁNH GIÁ

1. Kết quả

Minh họa cách hoạt động và sử dụng hệ thống:

Video minh họa chi tiết: <https://www.youtube.com/watch?v=AN5jPnmuRLQ>

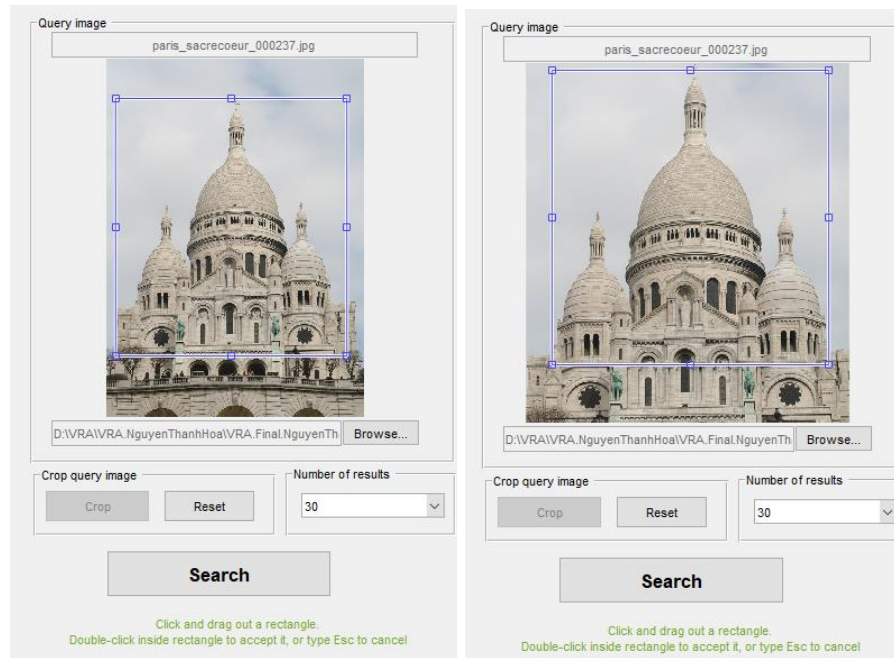
- **Bước 1: Chọn hình ảnh query** và số lượng kết quả trả về mong muốn



Hình 12. Chọn ảnh query

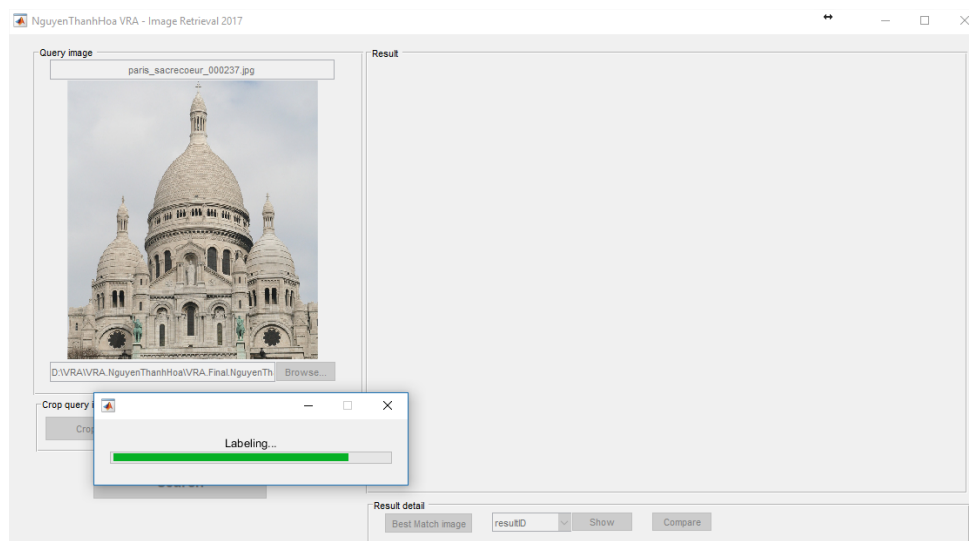
- **Bước 2:** (tùy chọn) **Thực hiện Crop ảnh** nếu muốn chọn 1 vùng bất kỳ trên ảnh làm query. Quá trình crop này có thể thực hiện liên tục nhiều lần.

Có thể điều chỉnh vùng chọn. Sau khi chọn, double-click vào vùng chọn để xác nhận, nhấn ESC để thoát hoặc bấm Reset để quay về ảnh ban đầu.

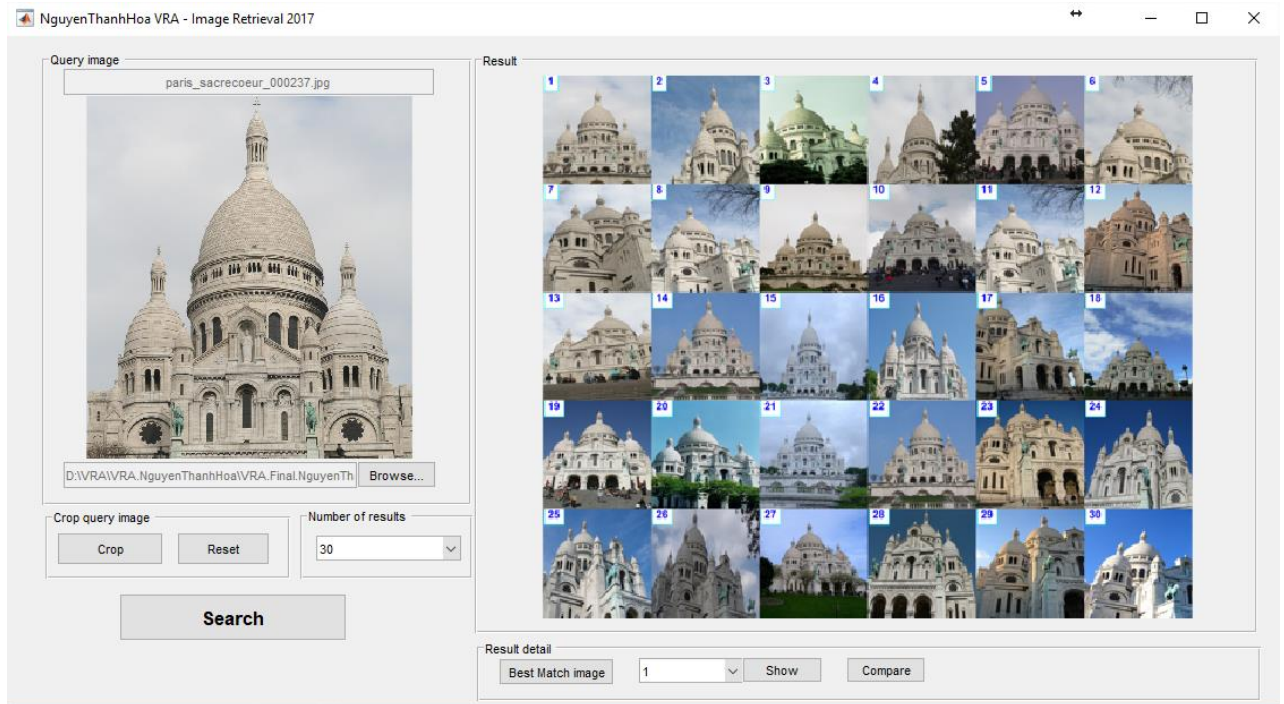


Hình 13. Ví dụ về chọn vùng ảnh làm query

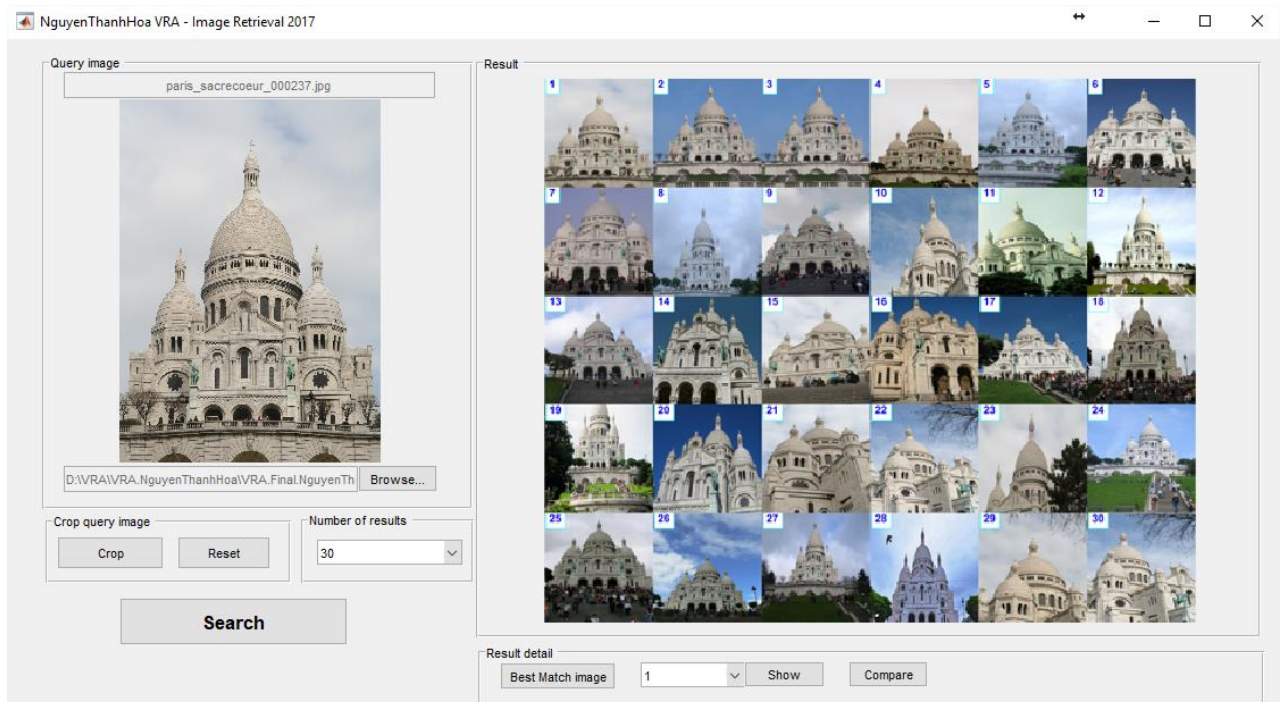
- **Bước 3:** Tìm kiếm dựa trên toàn bộ hoặc một phần hình ảnh query
Chọn Search để bắt đầu tìm kiếm



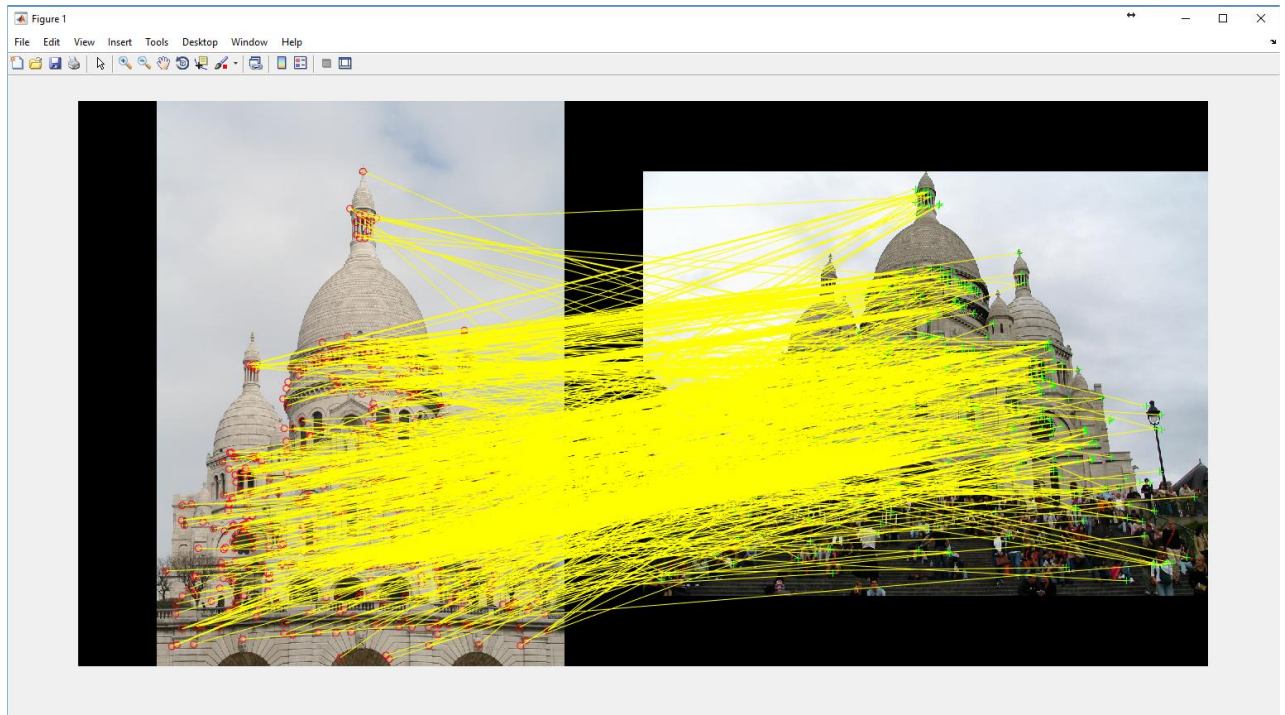
Hình 14. Quá trình tìm kiếm ảnh (trung bình khoảng 10s)



Hình 15. Kết quả 30 hình ảnh trả về

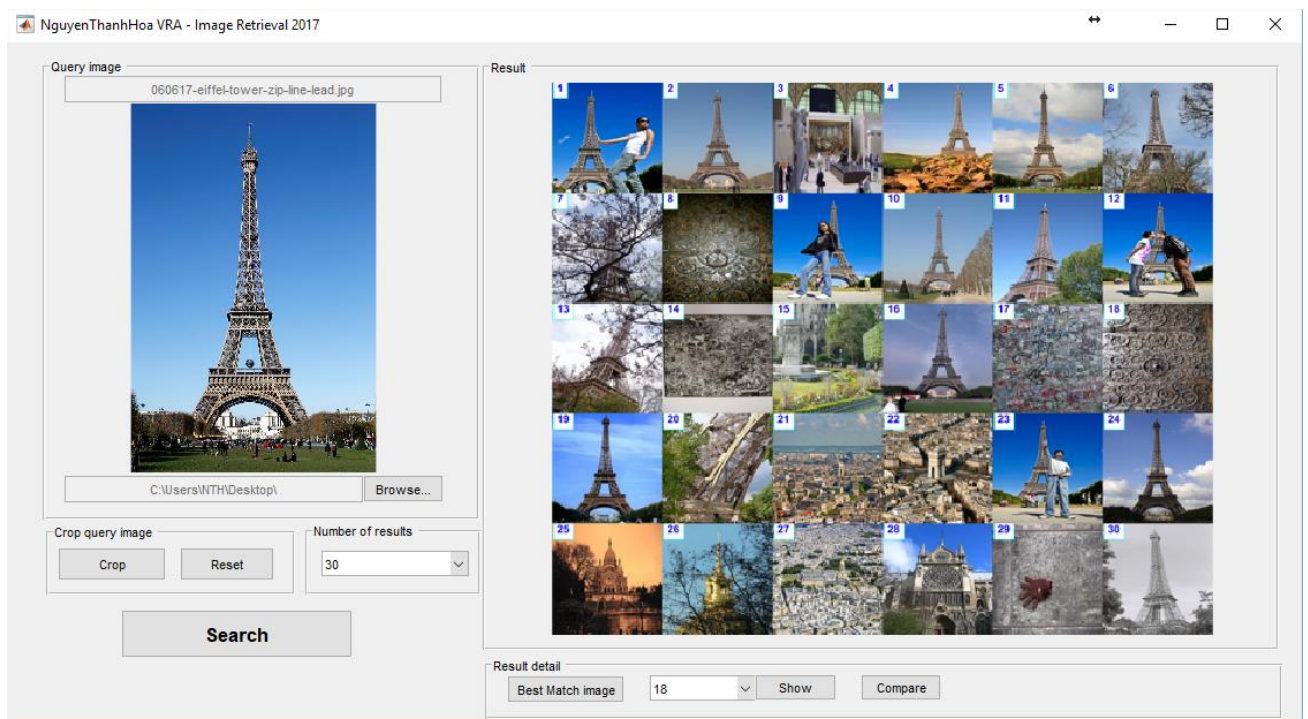


Hình 16. Kết quả nếu dùng ảnh gốc làm query



Hình 17. So khớp giữa ảnh query và 1 ảnh kết quả thứ 18

Sử dụng ảnh trong thư mục **test_image** gồm 60 ảnh không nằm trong dataset được download từ Internet để dùng làm ảnh query kiểm tra kết quả tìm kiếm trong dataset.



Hình 18. Tìm kiếm từ 1 hình ảnh khác download từ Internet

2. Đánh giá

Hệ thống đã đáp ứng được yêu cầu và mục tiêu ban đầu với kết quả hình ảnh truy vấn trả về nhanh chóng và phù hợp nhất dựa trên các hình ảnh đang có trong bộ dữ liệu train là Paris Dataset với 6392 ảnh. Hệ thống này thuộc dạng **Content Based Image Retrieval** [2] – truy vấn hình ảnh dựa trên nội dung có trên ảnh, cho kết quả chính xác và tốc độ truy vấn khá nhanh chóng và ổn định.

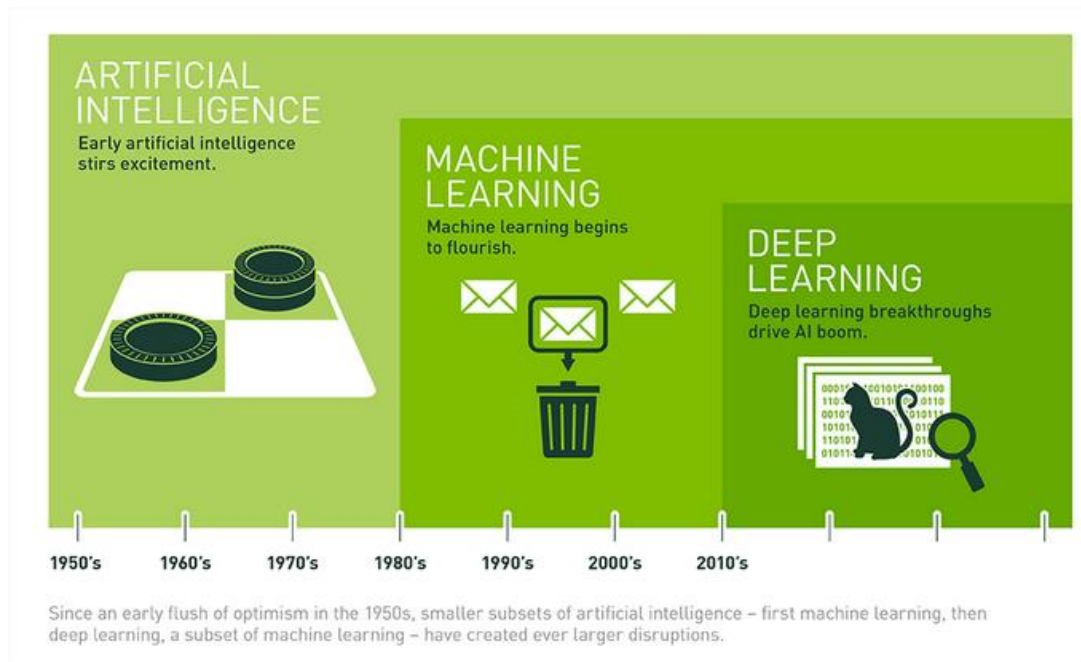
Với hệ thống, thời gian xây dựng bộ BagOfFeatures sẽ rất lâu nếu không được cung cấp sẵn – trong đồ án này bộ dữ liệu đã được xây dựng và đính kèm. Bên cạnh đó, hệ thống chỉ mới sử dụng bộ dữ liệu dựa trên phương pháp SURF để rút trích đặc trưng và chưa đưa ra nhiều thông số để đánh giá mức độ hiệu quả của các phương pháp khác nhau.

3. Hướng phát triển

- Sử dụng nhiều phương pháp khác để rút trích đặc trưng (Feature Extractor) để so sánh hiệu quả so với khi sử dụng SURF, từ đó lựa chọn phương pháp tối ưu nhất cho hệ thống.
- Tính toán các thông số để đánh giá (evaluate) hiệu quả của hệ thống với các query khác nhau và sử dụng Groundtruth files để so sánh kết quả với các phương pháp khác đã công bố.
- Mở rộng bộ dữ liệu Flickr 100k hoặc sử dụng bộ dữ liệu khác.

BÁO CÁO MỞ RỘNG

“Deep learning - xu hướng của tương lai”



Hình 19. Mối quan hệ giữa AI, Machine Learning và Deep Learning⁵

Deep Learning là một kỹ thuật của Machine Learning đang được giới công nghệ đặc biệt quan tâm và dự báo đây sẽ là một trong những xu hướng nghiên cứu của thời gian tới.

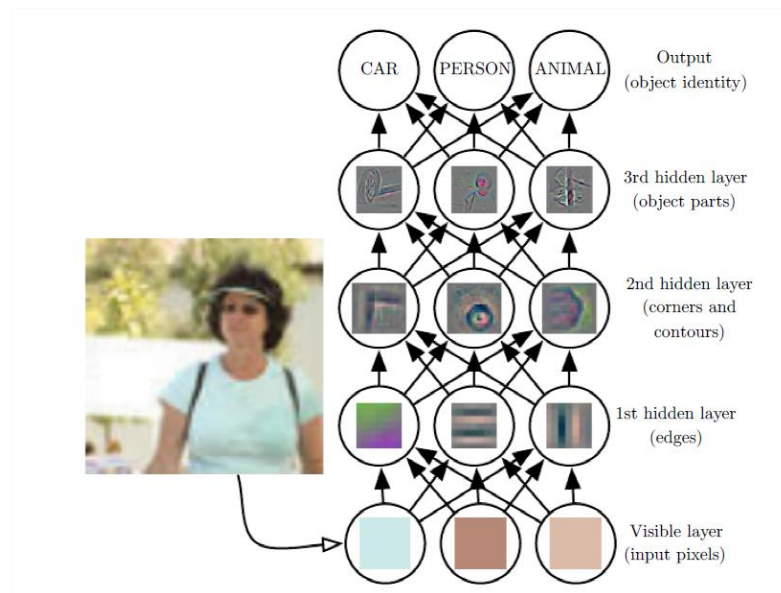
Theo Aaron Chavez, nhà khoa học tại AlchemyAPI: “The general idea of deep learning is to use neural networks to build multiple layers of abstraction to solve a complex semantic problem.”. Mục tiêu của **deep learning** nhằm cải thiện các vấn đề của Machine Learning như Computer Vision (thị giác máy tính), Natural Language Processing (xử lý ngôn ngữ tự nhiên),... nhằm giải quyết các thách thức về dữ liệu lớn, phức tạp và các loại dữ liệu phi cấu trúc.

NGUYÊN TẮC HOẠT ĐỘNG

Deep Learning xây dựng dựa trên 1 mạng lưới neural nhân tạo (neural network), tổ chức thành từng lớp (layer). Mỗi lớp này sẽ biểu diễn các thuộc tính (features) của lớp cao hơn.

⁵ Nguồn: <https://blogs.nvidia.com/blog/2016/10/17/deep-learning-help-business/>

Mạng neural nhân tạo có thể được hiểu là một hệ thống các chương trình và cấu trúc dữ liệu mô phỏng hoạt động của hệ thần kinh con người, gồm nhiều xử lý hoạt động song song.



Hình 20. Minh họa về hoạt động của Deep Learning [8]

Theo hình trên, các giá trị nhận được ở Layer 1 (Visible layer), lần lượt thông qua các lớp đến lớp cuối cùng. Layer 1 gọi là Input Layer, layer cuối cùng là Output layer, các layer ở giữa gọi là Hidden Layer.

Vấn đề của deep learning là cần phải được “train” (luyện tập) thông qua một số lượng rất lớn dữ liệu để có thể điều chỉnh, cải thiện dần các neural và có thể đưa ra kết luận chính xác hơn về vấn đề đang cần giải quyết.

Một số **ứng dụng** của Deep learning trong thực tế có thể kể đến như ⁶:

- Phân loại đối tượng (Object Classification) trong ảnh với số lượng ảnh lớn.
- Tái tạo ảnh màu từ ảnh đen trắng, tiếng động cho phim.
- Xác định ý nghĩa hình ảnh và tạo mô tả (caption) tự động.
- Phát hiện, chuẩn đoán bệnh,...

⁶ Tham khảo <https://machinelearningmastery.com/inspirational-applications-deep-learning/>

TÀI LIỆU THAM KHẢO

- [1] James Philbin, Andrew Zisserman, "The Paris Dataset," VGG, [Online]. Available: <http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/>. [Accessed 29 November 2017].
- [2] "Image Retrieval with Bag of Visual Words," 1 12 2017. [Online]. Available: <https://www.mathworks.com/help/vision/ug/image-retrieval-with-bag-of-visual-words.html>.
- [3] "Bag of visual words object," [Online]. Available: <https://www.mathworks.com/help/vision/ref/bagoffeatures-class.html>. [Accessed 22 November 2017].
- [4] "Create image search index," [Online]. Available: <https://www.mathworks.com/help/vision/ref/indeximages.html>. [Accessed 22 November 2017].
- [5] "Image Retrieval Using Customized Bag of Features," MathWorks, [Online]. Available: <https://www.mathworks.com/help/vision/examples/image-retrieval-using-customized-bag-of-features.html>. [Accessed 1 December 2017].
- [6] "Embed Text and Graphics in an Image," MathWorks, [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/38721-embed-text-and-graphics-in-an-image>. [Accessed 1 December 2017].
- [7] "Evaluate image search results," MathWorks, [Online]. Available: <https://www.mathworks.com/help/vision/ref/evaluateimageretrieval.html>. [Accessed 2 December 2017].
- [8] Ian Goodfellow and Yoshua Bengio and Aaron Courville, in *Deep Learning*, MIT Press, 2016, p. 6.

PHỤ LỤC

1. SourceCode của ứng dụng và các bộ Dataset dùng trong đồ án

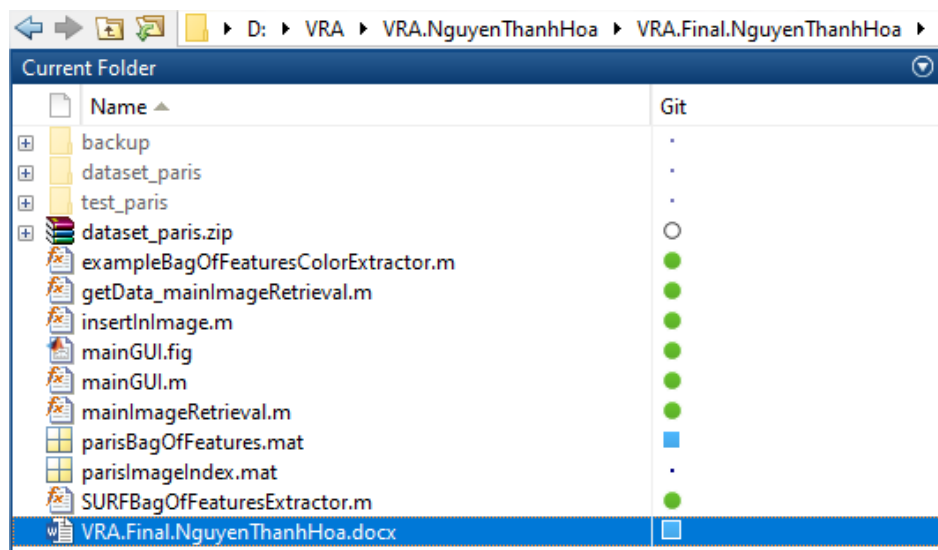
Tất cả các file liên quan đến đồ án được cập nhật tại Github > Thư mục

VRA.Final.NguyenThanhHoa: <https://github.com/hoait/VRA.NguyenThanhHoa>

Do giới hạn của Github upload file tối đa 100MB và toàn bộ repository tối đa 1 GB nên Dataset các các file .mat rút trích đặc trưng được upload riêng tại:

- Bộ dữ liệu ảnh train **Paris Dataset [1]** (6392 ảnh) – 2.5GB upload tại <https://drive.google.com/file/d/1nqgNIB7zQeur0FZFMKx0b6gBihTbo8gR/view?usp=sharing>.
- File `parisImageIndex.mat` (106 MB): <https://drive.google.com/file/d/1P5rmGU7N9QVseH9tElwEtv1VGpB2ZoqA/view?usp=sharing>

Các file này cần được đặt chung với thư mục VRA.Final.NguyenThanhHoa tại đường dẫn D:\VRA\VRA.NguyenThanhHoa\VRA.Final.NguyenThanhHoa tương tự hình sau để chương trình hoạt động tốt nhất:



Hình 21. Cấu trúc thư mục của project

2. Hàm SURFBagOfFeaturesExtractor

```
function [features, featureMetrics, vararginout] = SURFBagOfFeaturesExtractor(I)
% Step 1: Preprocess the Image
[height,width,numChannels] = size(I);
```

```

if numChannels > 1
    grayImage = rgb2gray(I);
else
    grayImage = I;
end
%% Step 2: Select Point Locations for Feature Extraction
% Define a regular grid over I.
gridStep = 8; % in pixels
gridX = 1:gridStep:width;
gridY = 1:gridStep:height;

[x,y] = meshgrid(gridX, gridY);
gridLocations = [x(:) y(:)];
% Use a feature detector such as detectSURFFeatures
% or detectMSERFeatures to select point locations. For instance:
multiscaleSURFPoints = detectSURFFeatures(grayImage);

%% Step 3: Extract features
% Finally, extract features from the selected point locations. By %
default,bagOfFeatures extracts upright SURF features.
features = extractFeatures(grayImage, multiscaleSURFPoints,'Upright',true);

%% Step 4: Compute the Feature Metric
% Alternatively, if a feature detector was used for point
% selection, the detection metric can be used. For example:
featureMetrics = multiscaleSURFPoints.Metric;
% Optionally return the feature location information.
if nargin > 2
    % Return feature location information
    varargout{1} = multiscaleSURFPoints.Location;
end

```

3. Các Callback trong GUI của hệ thống

- Browse và load ảnh query

```

function browseBtn_Callback(hObject, eventdata, handles)
    [filename, pathname] = uigetfile({'*.jpg'; '*.bmp'}, 'Select
    image...');

    handles.myImage = imread(strcat(pathname, filename));
    % Save original image (oImage) for reset after crop
    handles.oImage = handles.myImage;
    % Show image in axes1
    axes(handles.axes1);
    imshow(handles.myImage);
    set(handles.filenameLbl, 'string', filename);
    set(handles.pathLbl, 'string', pathname);
    % save the updated handles object
    guidata(hObject, handles);

```

```
% Adjust button (enable)
handles.browseText.Visible = 'Off';
handles.cropBtn.Enable = 'On';
handles.searchBtn.Enable = 'On';
handles.rstBtn.Enable = 'On';
```

- Search Callback

```
function searchBtn_Callback(hObject, eventdata, handles)
% hObject    handle to searchBtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Adjust button (enable)
handles.cropBtn.Enable = 'Off';
handles.browseBtn.Enable = 'Off';
handles.rstBtn.Enable = 'Off';
handles.searchBtn.Enable = 'Off';
handles.numResult =
str2double(handles.numPopup.String{handles.numPopup.Value});
[thumbnails, result] =
mainImageRetrieval(handles.myImage,handles.numResult);
montage(thumbnails,'Parent',handles.axes2);
handles.result = result;
% Adjust button (enable)
handles.cropBtn.Enable = 'On';
handles.browseBtn.Enable = 'On';
handles.rstBtn.Enable = 'On';
handles.searchBtn.Enable = 'On';
handles.numPopup.String
%Change result popup
temp = strtrim(cellstr(num2str([1:handles.numResult]')));
set(handles.resultPopup,'String',temp);
% save the updated handles object
guidata(hObject,handles);
```

- Crop Callback

```
function cropBtn_Callback(hObject, eventdata, handles)
% hObject    handle to cropBtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
crophlpMess = {'Click and drag out a rectangle.';
'Double-click inside rectangle to accept it, or type Esc to
cancel'};
% uiwait(helpdlg(crophlpMess,'Crop Image Tips'));
set(handles.hlpText,'string',crophlpMess);
handles.cropBtn.Enable = 'Off';
axes(handles.axes1);
```

```

gca = handles.axes1;
hRect = imrect(gca);
pos = wait(hRect);
if isempty(pos)
    delete(hRect);
else
    handles.myImage = imcrop(handles.myImage,pos);
    imshow(handles.myImage);
    % save the updated handles object
    guidata(hObject,handles);
end
handles.cropBtn.Enable = 'On';
set(handles.hlpText,'string','');

```

- Compare Callback

```

function compareBtn_Callback(hObject, eventdata, handles)
% hObject    handle to compareBtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
selectedValue =
str2double(handles.resultPopup.String{handles.resultPopup.Value});
currentImage = handles.imageIDs(selectedValue);

bestMatchWords = handles.parisImageIndex.ImageWords(currentImage);
bestImage =
imread(handles.parisImageIndex.ImageLocation{currentImage});
queryWordsIndex = handles.queryWords.WordIndex;
bestMatchWordIndex = bestMatchWords.WordIndex;

tentativeMatches = [];
for i = 1:numel(handles.queryWords.WordIndex)
    idx = find(queryWordsIndex(i) == bestMatchWordIndex);
    matches = [repmat(i, numel(idx), 1) idx];
    tentativeMatches = [tentativeMatches; matches];
end
points1 = handles.queryWords.Location(tentativeMatches(:,1),:);
points2 = bestMatchWords.Location(tentativeMatches(:,2),:);
figure
showMatchedFeatures(handles.myImage,bestImage,points1,points2,'monta
ge');

```

HẾT