ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**Phòng Đào tạo Sau đại học & Khoa học công nghệ**

**ĐỒ ÁN MÔN HỌC**

**NHẬN DẠNG THÔNG TIN THỊ GIÁC**

GVHD:

Lê Đình Duy

Nguyễn Tấn Trần Minh Khang

Học viên:

Lê Hoàng Dũng - Mã HV: CH1501022

TP. Hồ Chí Minh – tháng 12/2017

**MỤC LỤC**

[MÔ TẢ ĐỒ ÁN 1](#_Toc502175474)

[PHẦN 1: HỆ THỐNG TRUY VẤN HÌNH ẢNH 2](#_Toc502175475)

[I. Tổ chức thư mục 2](#_Toc502175476)

[II. Phương pháp xây dựng hệ thống 5](#_Toc502175477)

[1. Rút trích đặc trưng (file index.py) 5](#_Toc502175478)

[2. Kĩ thuật gom cụm (Clustering) (file index.py) 6](#_Toc502175479)

[3. Tính khoảng cách TF-IDF (file index.py) 8](#_Toc502175480)

[4. Truy vấn ảnh (file search.py) 8](#_Toc502175481)

[5. Xây dựng server (file server.py) 10](#_Toc502175482)

[6. Xây dựng website 11](#_Toc502175483)

[7. Demo chương trình 11](#_Toc502175484)

[TÀI LIỆU THAM KHẢO 12](#_Toc502175485)

[MỘT SỐ SOURCE CODE THAM KHẢO 12](#_Toc502175486)

# MÔ TẢ ĐỒ ÁN

**Xây dựng hệ thống truy vấn hình ảnh (Image Retrieval System)**

* Github:
* Youtube:

**Mục tiêu, nội dung đồ án**

Tìm hiểu và áp dụng các phương pháp dùng BoW để xây dựn một hệ thống truy vấn hình ảnh. Hệ thống cung cấp một giao diện người dùng website. Tích hợp web service cung cấp các API để phục vụ cho việc truy vấn

Hệ thống sử dụng bộ dữ liệu train chuẩn là oxford5k (tham khảo <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/oxbuild_images.tgz> ). Vì lí do hạn chế phần cứng nên sẽ giới hạn 1000 bức ảnh train (random).

Ngoài chức năng retrieval bằng ảnh, hệ thống còn có tính năng cho phép người dùng lựa chọn vùng truy vấn (cropped), hoặc chọn loại đặc trưng sẽ dùng.

Các phương pháp được sử dụng: rút trích các đặc trưng như SIFT, SURF, ROOTSIFT, phương pháp gom cụm Kmeans và tính độ tương đồng bằng TF-IDF

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

# PHẦN 1: HỆ THỐNG TRUY VẤN HÌNH ẢNH

## **Tổ chức thư mục**

Tập tin đồ án gồm các thư mục sau:



* *Bao cao*: File báo cáo đồ án, phương pháp, kĩ thuật xây dựng hệ thống
* *Sources*: Chứa source code gồm
  + *web-client*: xây dựng web bằng ngôn ngữ NodeJS
  + *python-backend*: xây dựng một server REST API truy vấn ảnh bằng ngôn ngữ Python
  + *README.md* : Thông tin cài đặt và chạy thử hệ thống

**Hướng dẫn cài đặt hệ thống**

Môi trường cài đặt:

* Web:
  + NodeJS version 8.9.1
  + Cách build hệ thống:
    - cd web-client

Cài đặt packages

* + - npm install

Khởi tạo server (localhost:3000)

* + - npm run watch
* Server
  + Python: version 2.7.14
  + Cách build hệ thống:
    - cd python-backend

Cài đặt packages

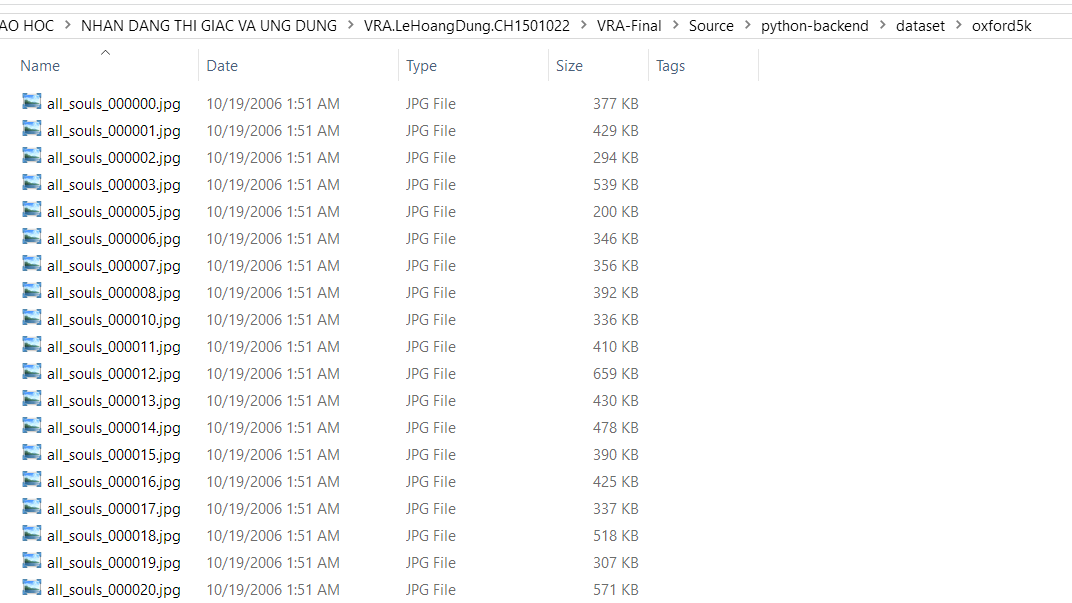
* + - python -m pip install -r requirements.txt

Khởi tạo data huấn luyện

* + - python index.py

Chạy server (localhost:5000)

* + - python server.py
* Thư mục dataset:
  + Copy ảnh bỏ vào thư mục: *python-backend/dataset/oxford5k/*



**Một số configuration trong hệ thống**

* Web:
  + File “config.json”:
    - **pyAPI.search**: link API đến server Python để truy vấn ảnh, giá trị default là <http://localhost:5000/search>
    - **pyAPI.upload**: link API đến server Python để upload ảnh, giá trị default là <http://localhost:5000/file-upload>
    - **externalUpload**: cho phép upload ảnh lên server Python, giá trị default là false
    - **searchByAjax**: cho phép gọi ajax thay vì submit ảnh lên server Python, giá trị default là true
* Server
  + File “config.py”:
    - **Settings.SEARCH\_THRESHOLD**: giá trị threshold để đánh giá ảnh trả về là phù hợp 🡪 trả về tất cả các ảnh có độ phù hợp (score) >= threshold
    - **Settings.MAX\_FILES**: số lượng file lớn nhất dùng để huấn luyện
    - **Settings.ROOT\_DATASET\_FOLDER**: thư mục lưu trữ ảnh huấn luyện, chứa các thư mục con là loại ảnh huấn luyện
    - **Settings.TRAIN\_DATASET**: tên thư mục loại ảnh được huấn luyện
    - **Settings.FEATURE**: loại đặc trưng dùng để rút trích
    - **Settings.FEATURE\_FILE**: tên file lưu trữ features
    - **KmeansTYPE**: phương pháp kmean sẽ dùng (type=1, 2, 3)
    - **Kmeans.NUM\_WORDS**: số lượng K clusters
    - **Kmeans.ITER**: số lần lặp Kmeans
    - **Features**: các loại đặc trưng sẽ dùng
    - **Resources**: chứa các cài đặt lên quan đến thư mục ảnh, phục vụ cho việc truy vấn đường dẫn file ảnh bằng các URL

## **Phương pháp xây dựng hệ thống**

1. **Rút trích đặc trưng (file index.py)**

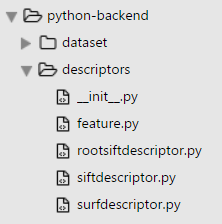
Hệ thống sử dụng ngôn ngữ Python để cài đặt hệ thống truy vấn ảnh. áp dụng một số thư viện của Python để hỗ trợ cho việc xử lý ảnh:

* + - * **Numpy**: chứa các hàm xử lý mảng, ma trận
      * **Scipy**: chứa các hàm xử lý ảnh, xử lý mảng, gom cụm
      * **OpenCV**: chứa các hàm xử lý ảnh, đọc ảnh, rút trích đặc trưng, hiển thị ảnh,..
      * **Pillow**: chứa các hàm để encode, lấy color,…
      * **Sklearn**: xử lý ảnh, gom cụm (clustering), đọc và lưu file

Quá trình huấn luyện dữ liệu train:

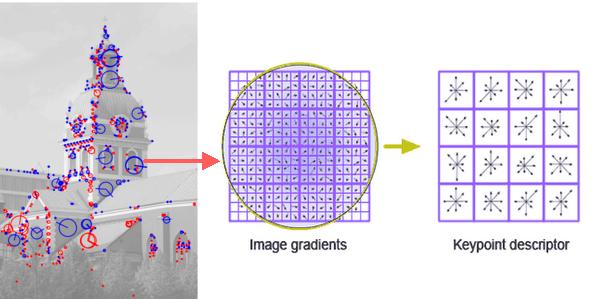
Nạp ảnh 🡪 Đọc và rút trích descriptors 🡪 clustering 🡪 tính TF-IDF 🡪 lưu kết quả tập tin

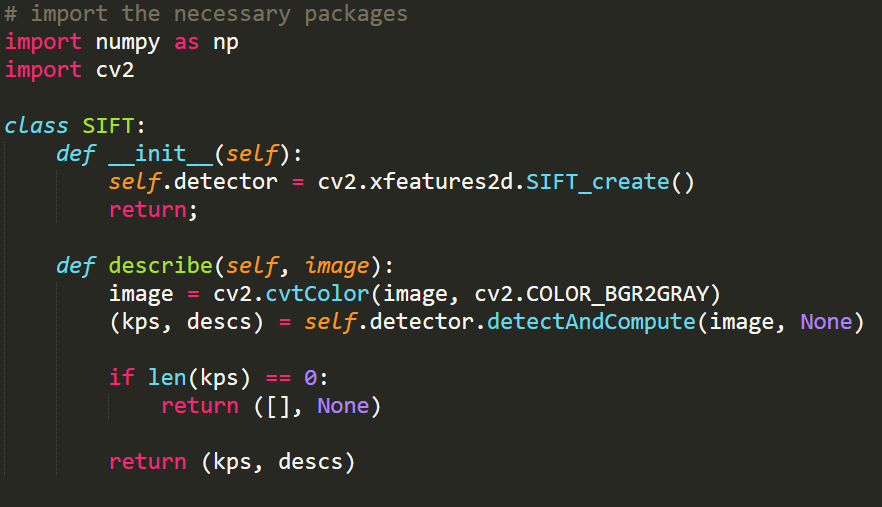
Áp dụng một số kĩ thuật rút trích đặc trưng như SIFT, SURF, ROOTSIFT. Các đặc trưng này được lựa chọn có sẵn trong thư viện của OpenCV và một số khác tham khảo từ trên internet. Các phương pháp được cài đặt trong thư mục *python-backend/descriptors*

**

**Đặc trưng SIFT:**

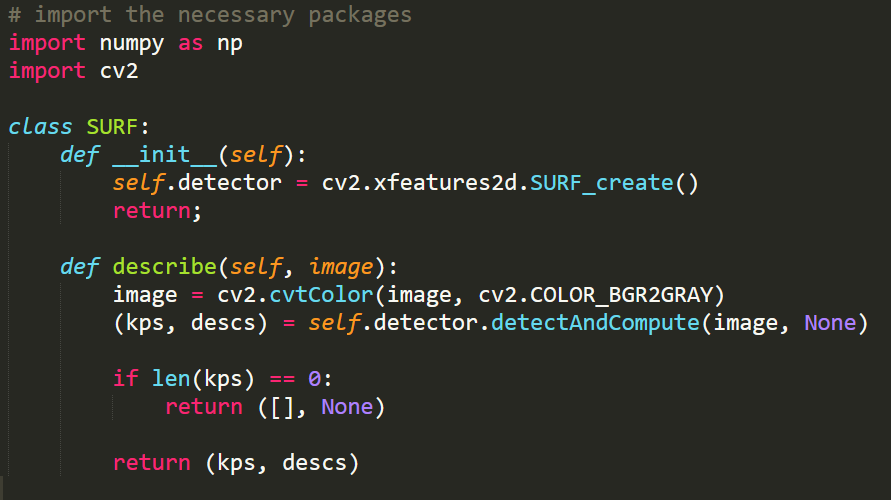
* Từ ảnh tìm ra các điểm ảnh đặc biệt, gọi là feature point hay keypoint. Đầu vào và đầu ra của phép biến đổi SIFT: ảnh -> SIFT -> các keypoint. Keypoint phụ thuộc rất ít vào cường độ sáng, nhiễu, che khuất (một phần ảnh bị che), góc xoay (ảnh bị xoay trong mặt phẳng 2D), thay đổi của tư thế (pose thay đổi trong không gian 3D)
* Để có thể phân biệt keypoint này với keypoint khác cần tìm ra tham số gì đó, gọi là descriptor. 2 keypoint khác nhau thì phải descriptor khác nhau. Thường thì descriptor là chuỗi số gồm 128 số (vector 128 chiều).
* Sau khi áp dụng biến đổi SIFT, ứng với mỗi keypoint, thu được (1) toạ độ keypoint (2) scale và orientation của keypoint (3) descriptor. Các mũi tên trong hình dưới vẽ nhờ vào scale và orientation





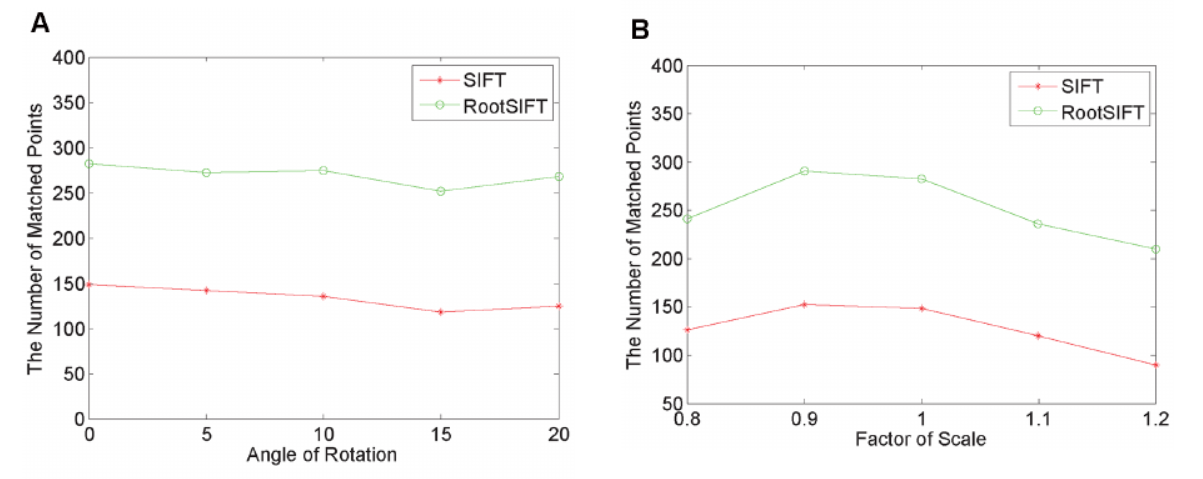
**Đặc trưng SURF**

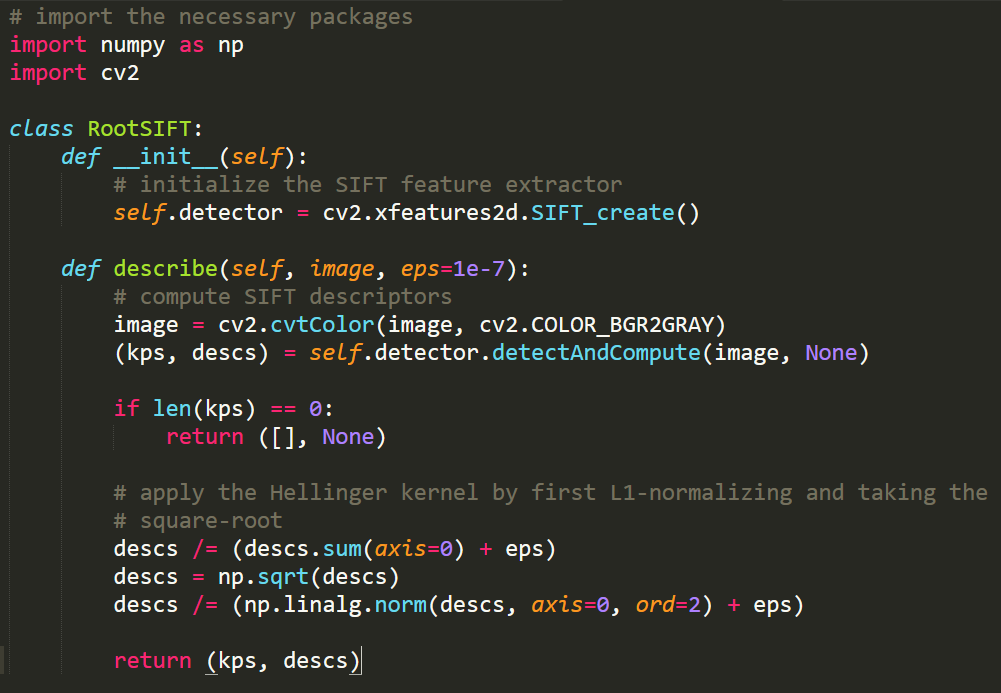
* SURF cũng gồm các bước như ở SIFT:
  + Scale-space extrema detection.
  + Keypoint localization.
  + Orientation assignment.
  + Keypoint descriptor.
* Nhưng, ở từng bước SURF sẽ có những sự cải thiện để cải thiện tốc độ xử lý mà vẫn đảm bảo độ chính xác trong việc detection.
* Ở SIFT, việc tìm Scale-space dựa trên việc tính gần đúng LoG (Laplace of Gaussian) dùng DoG (Difference of Gaussian), trong khi đó SURF sử dụng Box Filter, tốc độ xử lý sẽ được cải thiện đáng kể với việc dùng ảnh tích phân (integral image)
* Ở bước Orientation Assignment, SURF sử dụng wavelet response theo 2 chiều dọc và ngang, sau đó tính hướng chính bằng cách tính tổng các response đó, có một điều đáng chú ý là wavelet response cũng dễ dàng tính được với ảnh tích phân (integral image)



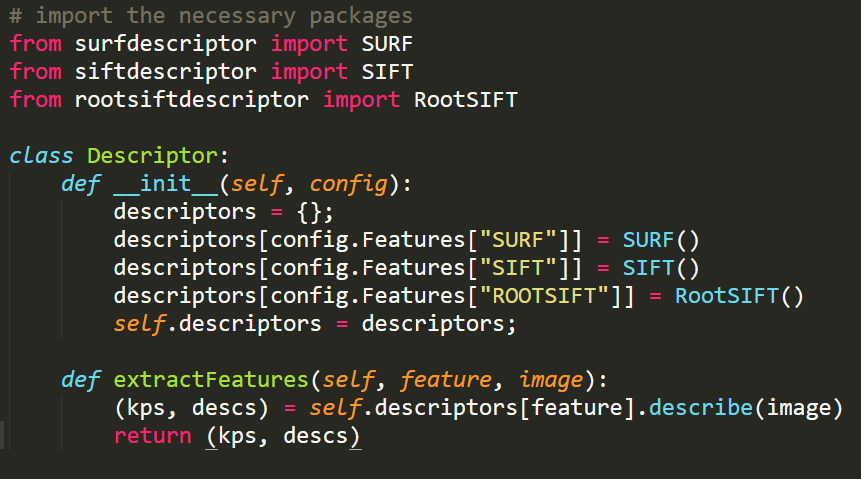
**Đặc trưng ROOTSIFT**

* ROOTSIFT là một cải tiến khác của SIFT, vẫn sử dụng khoảng cách Euclidean giống như SIFT. Tuy nhiên khoảng cách Euclide trong ROOTSIFT của các vector đặc trưng được ánh xạ tới mô hình nhân Hellinger (Hellinger kernel).
* Vectơ đặc trưng của RootSIFT được tính toán bởi nhân Hellinger, và sự kết hợp giữa hai điểm đặc trưng sau đó được đánh giá bởi tỷ số giữa khoảng cách điểm gần nhất và gần nhất thứ hai.
* Hiệu suất của việc nhận dạng được cải thiện bằng cách thay thế khoảng cách Euclide bằng nhân Hellinger, đã được chứng minh qua thực nghiệm





**Xây dựng bộ feature trong hệ thống:**

**

Thực hiện train 1000 ảnh với lần lượt các đặc trưng, nhóm nhận thấy ROOTSIFT có độ chính xác cao nhất trong các loại (trả về nhiều kết quả). SIFT và SURF cho kết quả gần giống nhau.

Kết quả: ROOTSIFT > SURF > SIFT

Để thực hiện huấn luyện ảnh, chỉnh các cài đặt trong file config.py:

* **Settings.MAX\_FILES**: giới hạn kích thước ảnh được train
* **Settings.TRAIN\_DATASET**: thư mục con của thư mục datasetchứa ảnh train
* **Settings.FEATURE**: loại đặc trưng để train

Chạy lệnh:

*python index.py*

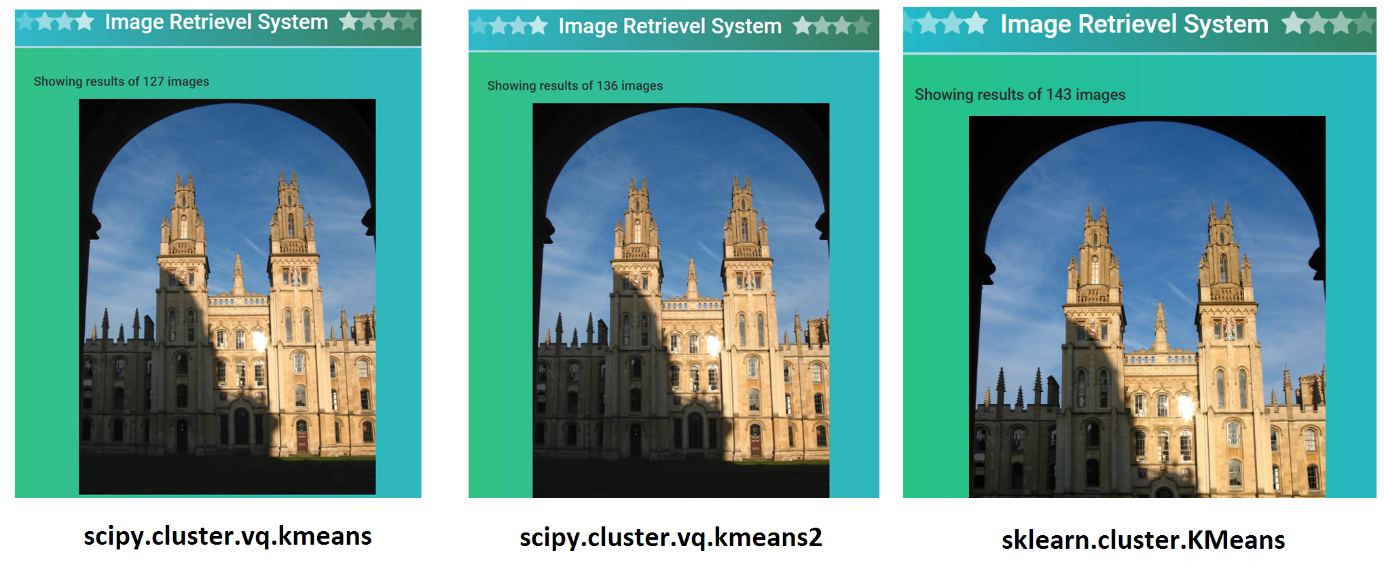
1. **Kĩ thuật gom cụm (Clustering) (file index.py)**

Hệ thống sử dụng thư viện **sklearn** và **scipy** để phục vụ cho việc clustering. Bởi vì các thư viện này cung cấp phương pháp KMeans khác nhau. Thực hiện lần lượt các phương pháp và nhận thấy KMeans của thư viện **sklearn** cho thời gian khá lâu so với **scipy**, nhưng bù lại có độ chính xác cao hơn

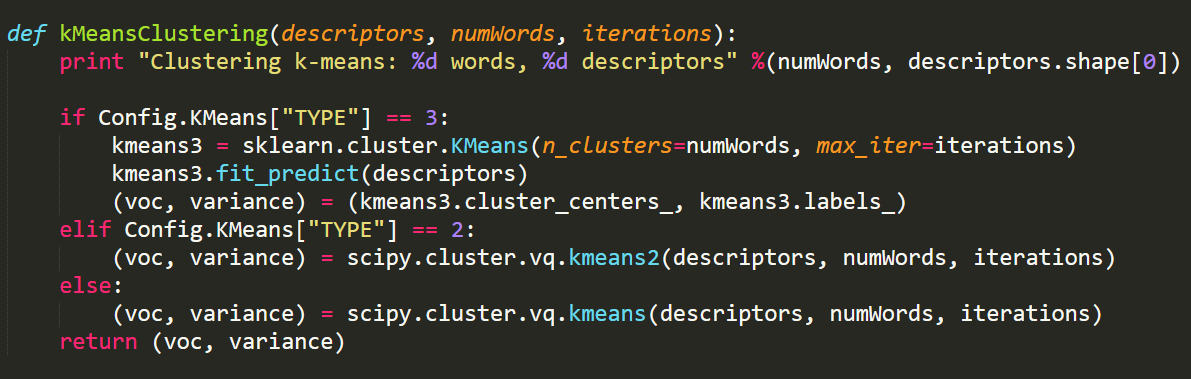
Ảnh query:



Kết quả của 3 phương pháp kmeans (K = 100, score threshold = 0.7)



Kết quả gom cụm ta sẽ thu được các center-clusters hay còn gọi là vocabularies



Một số cài đặt cho Kmeans

* **KMeans.TYPE**: phương pháp Kmeans được chọn(1, 2, 3)
* **KMeans.NUM\_WORDS**: số cụm (số K)
* **KMeans.ITER**: số lần lặp của thuật toán kmeans

1. **Tính khoảng cách TF-IDF (file index.py)**

Dựa vào kết quả bộ từ điển từ việc gom cụm, và danh sách các descriptors đã rút trích, tính toán khoảng cách vector dựa vào công thức TF-IDF và dạng chuẩn hóa L2



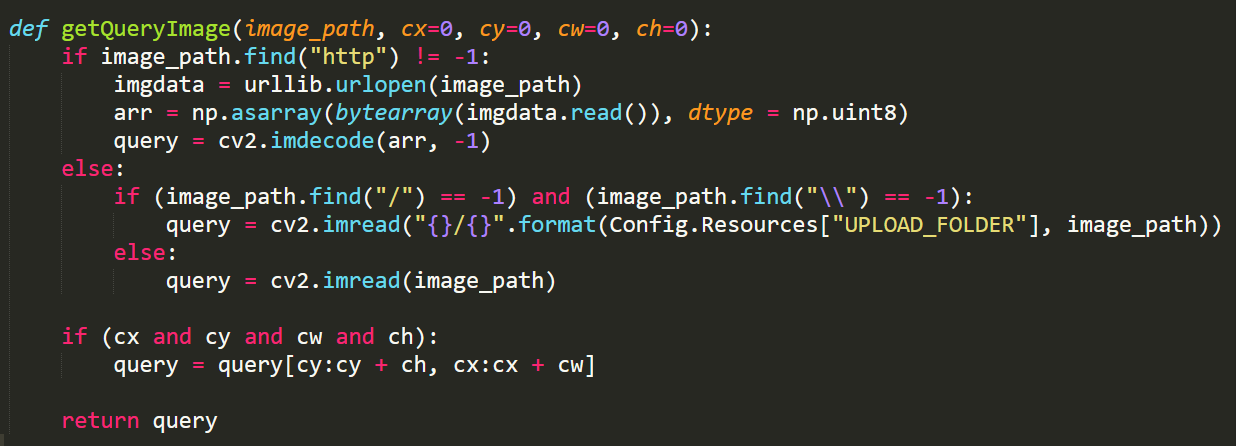
Kết quả được lưu trong tập tin:

*{tên thư mục train}\_{loại đặc trưng}\_ {loại kmean}\_features.bin*

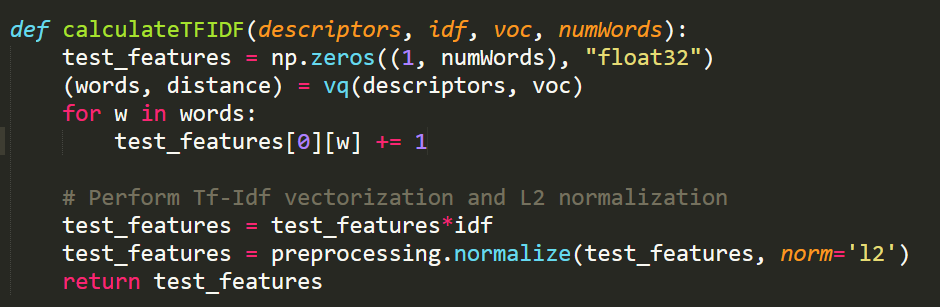
1. **Truy vấn ảnh (file search.py)**

Khi người dùng nhập ảnh query, ảnh sẽ được đọc và xử lý thông qua thư viện OpenCV.

Hàm **getQueryImage** có chức năng đọc ảnh và crop, nếu có đủ các tham số (x,y,w,h).

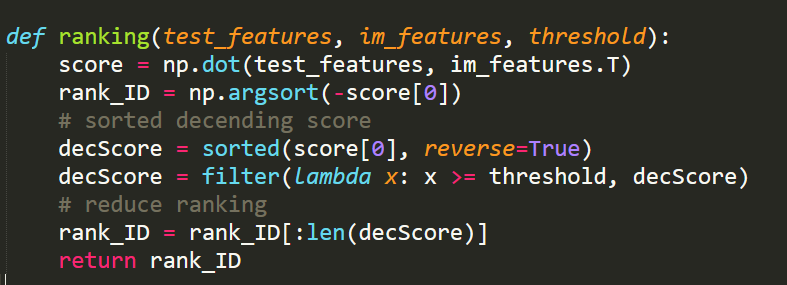


Tiếp đến ảnh query (hoặc crop) được rút trích đặc trưng và tính toán khoảng cách dựa vào giá trị TF-IDF và bộ từ điển lấy ra từ tập tin \*features.bin.

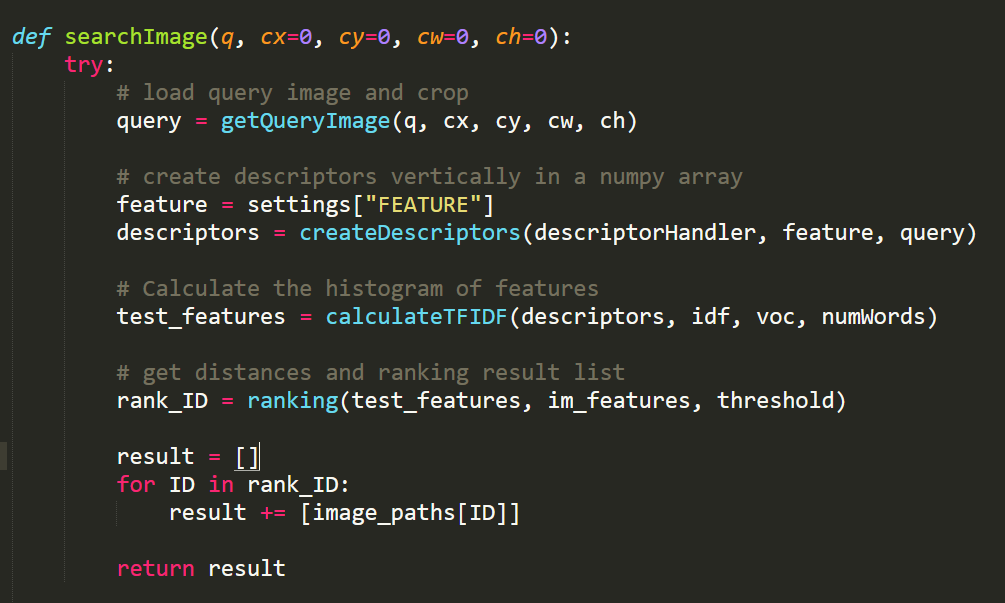


Sau đó ảnh trả về sẽ được lọc dựa trên giá trị THRESHOLD cho trước và sắp xếp (hàm **ranking**) theo thứ hạng từ cao đến thấp, sao cho ảnh gần giống query nhất sẽ xuất hiện ở vị trí đầu tiên

Để tính score, ta dựa vào biến **im\_features** chứa các đặc trưng rút ra từ tập train. Biến này có kích thước là (nRow, nCol) với nRow là số lượng ảnh train, nCol là số features rút trích. Nghĩa là mỗi dòng trong **im\_features** tương ứng với features của 1 ảnh. Tương tự **test\_features** là feature của ảnh query (chỉ có 1 dòng). Score được tính nhờ vào phép nhân ma trận **test\_features x invert(im\_features)**. Kết quả trả về là 1 danh sách có n ảnh result (n là tổng số ảnh train) mà mỗi phần tử có giá trị là độ tương đồng với ảnh query. Dựa vào danh sách này để trả về danh sách ảnh kết quả



Hàm **searchImage** gồm toàn bộ quy trình để tìm kiếm ảnh phù hợp câu query, tham số là query và các thông số crop ảnh (cx, cy, cw, ch)



1. **Xây dựng server (file server.py)**

Sử dụng thư viện Flask của Python và các thư viện con (Flask-Cors, Flask-Jsonpify, Flask-RESTful, Flask-Uploads) của nó để xây dựng một web service cung cấp API để truy vấn ảnh

Việc cài đặt server được thực hiện trong tập tin server.py. Server được chạy trên localhost: 5000

Chạy lệnh:

*python server.py*

1. **Xây dựng website**

Website được cài đặt bằng NodeJS với các plugin jQuery (Jcrop) để crop ảnh.

Để kết nối đến server Python, cần cung cấp các cài đặt trong file config.json

* + **pyAPI.search**: link API đến server Python để truy vấn ảnh, giá trị default là <http://localhost:5000/search>
  + **pyAPI.upload**: link API đến server Python để upload ảnh, giá trị default là <http://localhost:5000/file-upload>
  + **externalUpload**: cho phép upload ảnh lên server Python, giá trị default là false
  + **searchByAjax**: cho phép gọi ajax thay vì submit ảnh lên server Python, giá trị default là true

Chạy lệnh (localhost: 3000):

npm run watch

1. **Demo chương trình**

# PHẦN 2: VẤN ĐỀ MỞ RỘNG

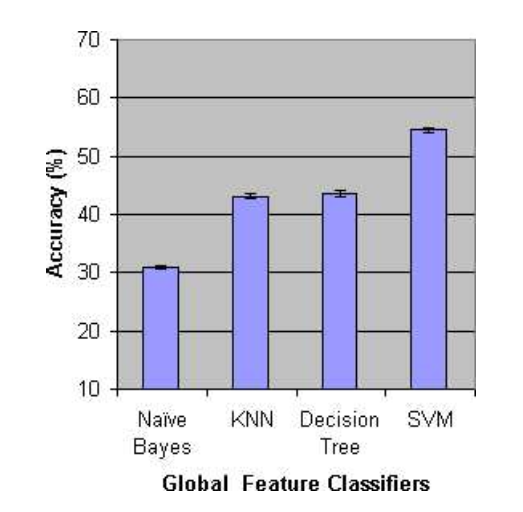
**Phương pháp kết hợp local feature và global feature trong bài toán nhận dạng thị giác Object Class Recognition**

## **Global feature**

Global feature mô tả bức ảnh như một tổng thể để khái quát toàn bộ đối tượng mà trong đó các local features mô tả những phần của bức ảnh. Global features gồm các mô tả về đường biên (contour representations), hình dạng (shape descriptors), và đặc điểm cấu trúc (texture features). Ví dụ: Shape Matrices, Invariant Moments (Hu, Zerinke), Histogram Oriented Gradients (HOG) và Co-HOG

Có rất nhiều hệ thống nhận dạng ảnh sử dụng global feature để mô tả nội dung một bức ảnh. Bởi vì chúng tạo ra các hình ảnh rất nhỏ gọn, ở đó mỗi hình ảnh tương ứng với một điểm trong một không gian đặc trưng cao. Kết quả là, bất kỳ bộ phân loại chuẩn nào cũng có thể được sử dụng.

Tuy nhiên, phần lớn các global features thường giả định rằng mỗi hình ảnh chỉ chứa một loại đối tượng. Do đó chúng rất hạn chế trong trường hợp bức ảnh chứa nhiều loại đối tượng được sắp xếp trộn lẫn



## **Local feature**

## **Kết hợp local feature và global feature trong bài toán nhận dạng**

## **Kết luận**

# TÀI LIỆU THAM KHẢO

1. Slide bài giảng môn Tìm kiếm thông tin thị giác
2. Slide bài giảng môn Nhận dạng thông tin thi giác
3. TF-IDF and L2-norm.

Link: <http://blog.christianperone.com/2011/10/machine-learning-text-feature-extraction-tf-idf-part-ii/>

1. Implementing RootSIFT in Python and OpenCV.

Link: <https://www.pyimagesearch.com/2015/04/13/implementing-rootsift-in-python-and-opencv/>

1. K Means Algorithm. Link: <http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>
2. Jcrop jQuery (crop ảnh). Link: <https://github.com/tapmodo/Jcrop>
3. B-lazy.js jQuery (tích hợp lazy load cho ảnh kết quả). Link: <http://dinbror.dk/blazy/>
4. Dropzone jQuery (cung cấp giao diện upload ảnh). Link: <http://www.dropzonejs.com/>
5. <http://zeus.robots.ox.ac.uk/oxfordbuildings/>
6. <https://en.wikipedia.org/wiki/Hellinger_distance>

# MỘT SỐ SOURCE CODE THAM KHẢO

1. <https://www.pyimagesearch.com/2014/12/01/complete-guide-building-image-search-engine-python-opencv/>
2. <https://github.com/devashishp/Content-Based-Image-Retrieval>
3. <https://gist.github.com/anabranch/48c5c0124ba4e162b2e3>