

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

-----



**BÁO CÁO MÔN HỌC**

**Đề Tài: Xây dựng hệ thống truy vấn ảnh đơn giản**

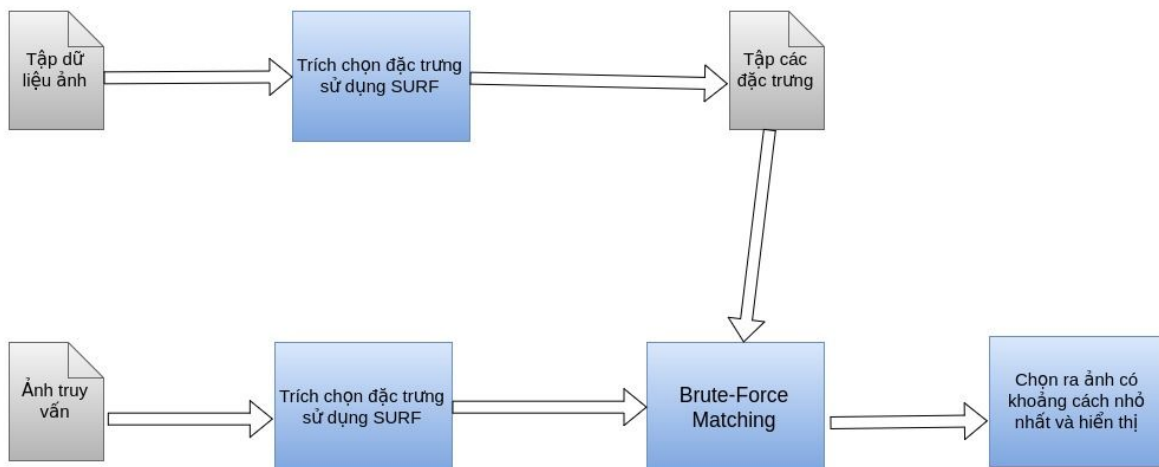
Giáo viên hướng dẫn: TS. Nguyễn Thị Oanh

Nhóm sinh viên thực hiện:

Đinh Hữu Hải Quân	20143632
Nguyễn Tiến Thạo	20144161
Bùi Tiến Thành	20144052

# I. Mô hình hệ thống

## 1. Giải pháp 1



Trong giải pháp đầu tiên để xây dựng hệ thống, nhóm sử dụng các kỹ thuật như:

- SURF (Speeded-Up Robust Features): sử dụng để trích chọn đặc trưng của các ảnh
- Brute-Force Matching: để so sánh các đặc trưng của ảnh

Quá trình phương pháp này như sau:

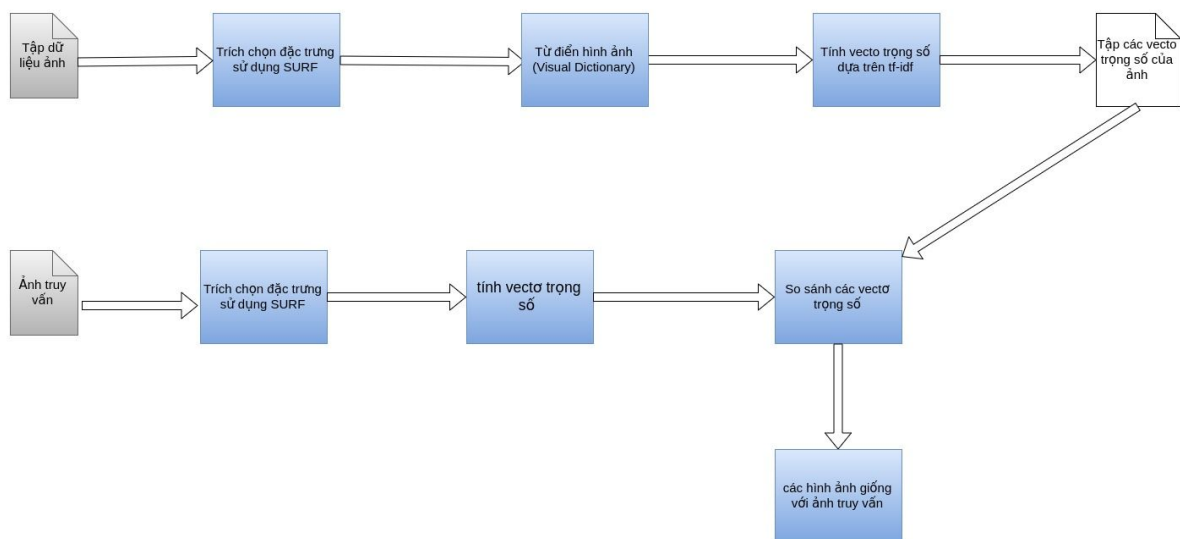
- Bước 1: Trích chọn được các đặc trưng của tập ảnh dữ liệu theo phương pháp SURF thì hàm **surf.detectAndCompute()** sẽ trả về cho chúng ta các **keypoints** và các **descriptors** tương ứng.
- Bước 2: Lưu các descriptors ra file (chứa trong thư mục **extract-feature**)
- Bước 3: Trích chọn đặc trưng của ảnh truy vấn theo SURF
- Bước 4: Sử dụng phương pháp Brute-Force Matching để so sánh các đặc trưng của ảnh truy vấn với các đặc trưng của các ảnh đã lưu ra file ở bước 2. Hàm **match()** sẽ trả về một mảng các đối tượng **DMatch** có chứa thuộc tính **distance** các đặc trưng giống nhau nhất. Dựa vào đó, ta tính tổng các distance này cho mỗi ảnh và sắp xếp chúng theo thứ tự.

- Bước 5: Chọn ra các ảnh có tổng các distance từ nhỏ đến lớn tương ứng với ảnh giống ít hay nhiều với ảnh truy vấn

**Ưu điểm:** Đơn giản, dễ làm

**Nhược điểm:** tốc độ truy vấn chậm

## 2. Giải pháp 2



Trong giải pháp thứ hai để xây dựng hệ thống, nhóm sử dụng các kỹ thuật như:

- SURF (Speeded-Up Robust Features): sử dụng để trích chọn đặc trưng của các ảnh
- Kmeans: để phân các đặc trưng thành các cụm
- Tf-idf: Để xây dựng vectơ trọng số
- Phương pháp đo khoảng cách euclidean: đo khoảng cách giữa các vectơ trọng số

Quá trình phương pháp này như sau:

- Bước 1: Trích chọn được các đặc trưng của tập ảnh dữ liệu theo phương pháp SURF thì hàm **surf.detectAndCompute()** sẽ trả về cho chúng ta các **keypoints** và các **descriptors** tương ứng.

- Bước 2: Gom các đặc trưng của tất cả ảnh lại và sử dụng thuật toán Kmeans để phân thành các cụm. Hệ thống đang sử dụng Kmeans với số cụm là 1000. Từ đó ta có một từ điển hình ảnh (Visual Dictionary)

- Bước 3: Sử dụng phương pháp Tf-idf để tính ra vector trọng số của mỗi ảnh theo Visual Dictionary đã tạo ở trên

- Bước 4: Lưu tập Visual Dictionary (file **voc.npy**), vector trọng số (file **img\_vs\_cluster.npy**), danh sách ảnh (file **img\_paths.npy**) ra File

- Bước 5: Trích chọn đặc trưng của ảnh truy vấn theo SURF

- Bước 6: Load dữ liệu từ các file đã lưu và tạo ra vector trọng số cho ảnh truy vấn

- Bước 7: Sử dụng phương pháp đo khoảng cách euclidean để tính khoảng cách giữa vector trọng số của ảnh truy vấn và các vector trọng số trong File đã lưu

- Bước 8: Sắp xếp khoảng cách đã tính và đưa ra ảnh theo thứ tự khoảng cách từ nhỏ đến lớn

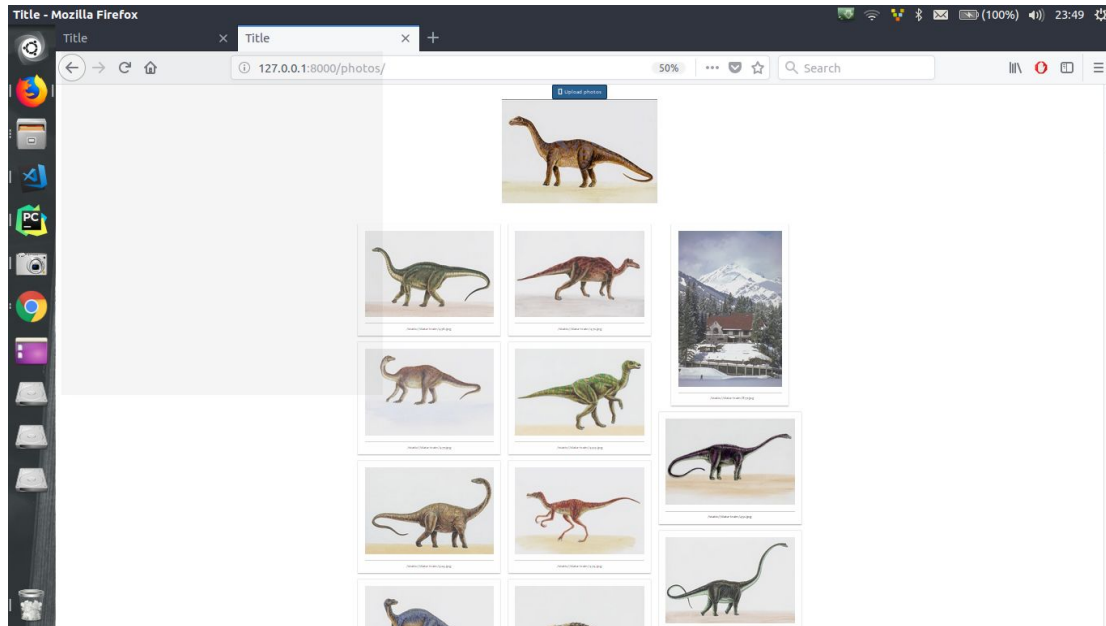
**Ưu điểm:** Tốc độ truy vấn nhanh hơn

**Nhược điểm:** quá trình thực hiện phức tạp hơn phương pháp đầu tiên và quá trình phân cụm có thể diễn ra lâu nếu dữ liệu lớn

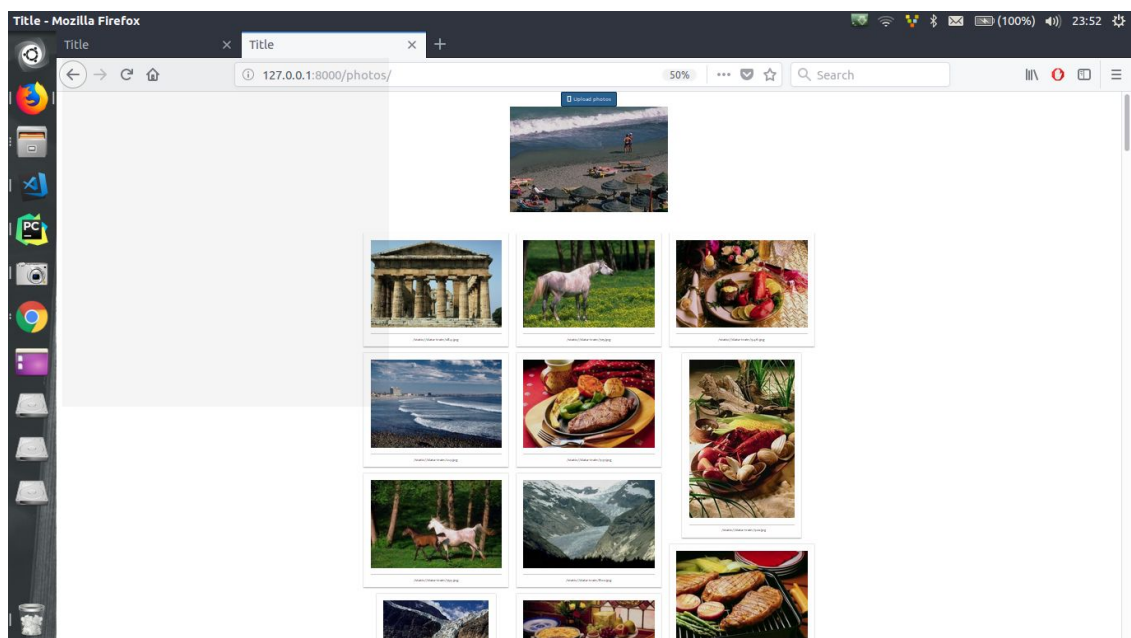
## II. Demo hệ thống

### 1. Giải pháp 1

- Trường hợp kết quả trả về tốt:

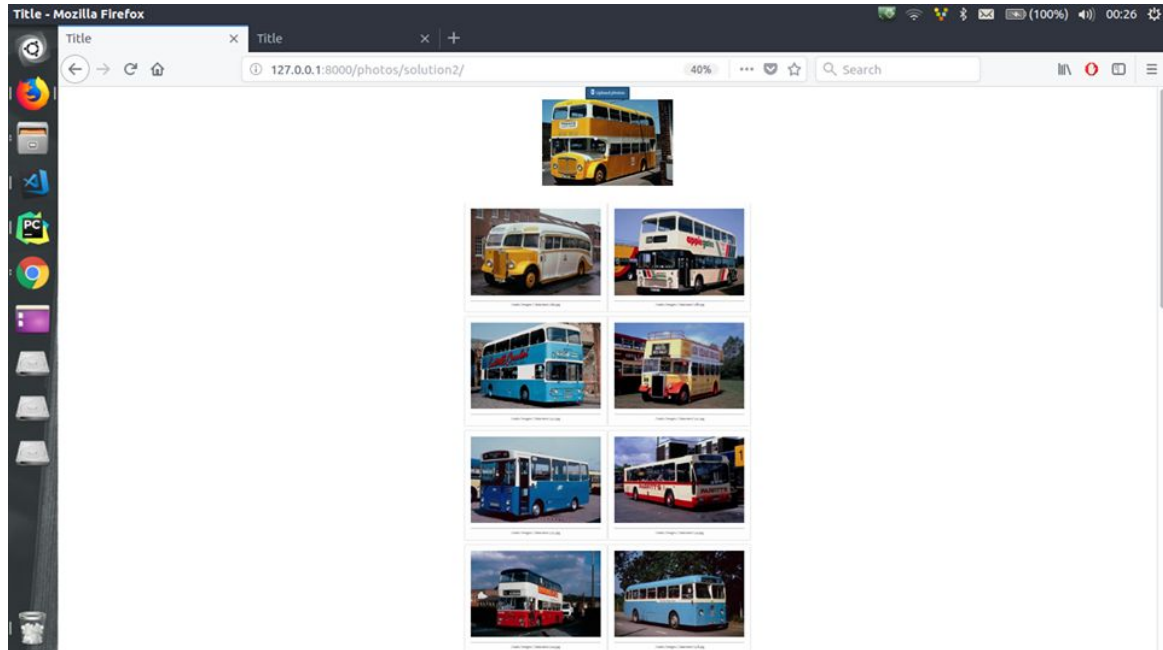


- Trường hợp kết quả trả về không tốt:

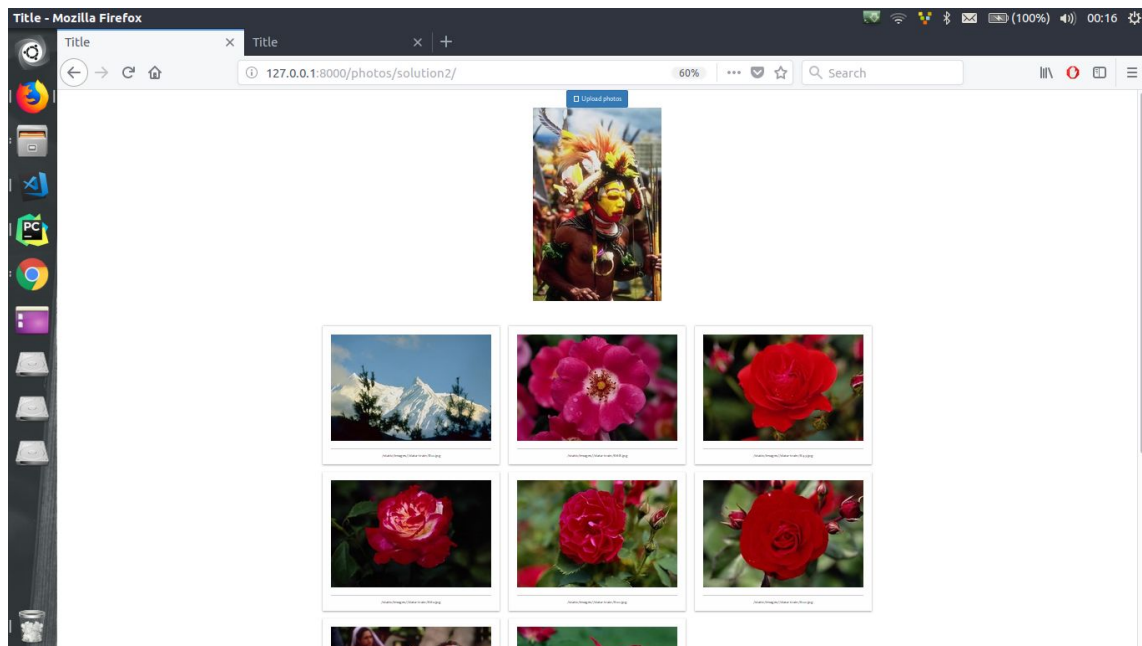


## 2. Giải Pháp 2

- Trường hợp kết quả trả về tốt:



- Trường hợp kết quả trả về không tốt:



### III. Tài liệu tham khảo

- <https://docs.opencv.org/3.4.0/>
- <https://ianlondon.github.io/blog/visual-bag-of-words/v>
- <https://docs.scipy.org/doc/scipy/reference/cluster.vq.html>