

BÁO CÁO

Đồ án cuối cùng

I. Thông tin sinh viên thực hiện:

1) Sinh viên 1:

- Tên: Lê Phú Đạt
- MSSV: 1653120
- Lớp: 16CLC2
- Email: 1653120@student.hcmus.edu.vn
- SĐT: 0362376029

2) Sinh viên 2:

- Tên: Nguyễn Anh Tú
- MSSV: 1653142
- Lớp: 16CLC2
- Email: 1653142@student.hcmus.edu.vn
- SĐT: 0357527794

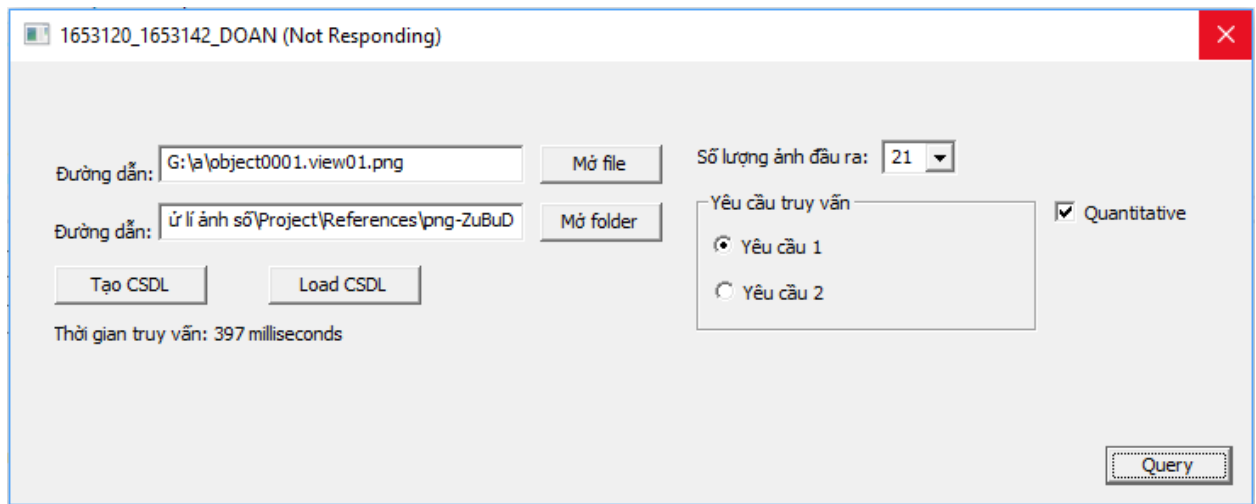
II. Nội dung:

1) Mức độ hoàn thành:

STT	Yêu cầu	Ghi chú	Hoàn Thành
<p>Mỗi nhóm gồm 2 sinh viên viết chương trình có giao diện mô phỏng hệ thống truy vấn ảnh trên CSDL ảnh ZuBuD. Sinh viên có thể download dữ liệu từ trang web môn học hoặc ở link sau: http://www.vision.ee.ethz.ch/showroom/zubud/</p> <p>Môi trường lập trình: C/C++, thư viện OpenCV. Sinh viên được sử dụng tất cả API mà thư viện OpenCV hỗ trợ. Đối với chương trình mô phỏng, sinh viên có thể xây dựng chương trình cho desktop hoặc mobile.</p>			
1	<ul style="list-style-type: none">- Cho phép người dùng truy vấn ảnh sử dụng lược đồ màu.- Đánh giá kết quả thực hiện của hệ thống truy vấn	<ul style="list-style-type: none">- Các tấm ảnh được hiển thị trên màn hình giảm dần theo mức độ giống và giá trị so khớp tương ứng. Độ do sinh viên tự chọn và giải thích.- So sánh kết quả sử dụng lược đồ màu lượng hóa và không lượng hóa.- Sinh viên nghiên cứu độ đo MAP (tài liệu tham khảo [1]). Sử dụng độ đo MAP để đánh giá độ chính xác kết quả truy vấn lần lược theo số lượng kết quả trả về là 3, 5, 11 và 21.	100%
2	<ul style="list-style-type: none">- Tìm hiểu một đặc trưng để tăng hiệu quả để nâng cao độ chính xác của hệ thống.- So sánh kết quả truy vấn của các phương pháp	<ul style="list-style-type: none">- Sinh viên có thể sử dụng các đặc trưng liên quan đến hình dáng, biên cạnh. Sinh viên có thể kết hợp các đặc trưng để tăng độ chính xác của hệ thống. Hoặc sử dụng các phương pháp để nâng cao chất lượng ảnh trước khi rút trích đặc trưng.- Sử dụng độ đo MAP để đánh giá độ chính xác kết quả truy vấn lần lược theo số lượng kết quả trả về là 3, 5, 11 và 21.	100%
3	<ul style="list-style-type: none">- Tìm hiểu và cài đặt cải tiến tốc độ tìm kiếm ảnh trong CSDL.	<ul style="list-style-type: none">- Sinh viên cần ghi rõ thời gian truy vấn trên giao diện của chương trình	100%

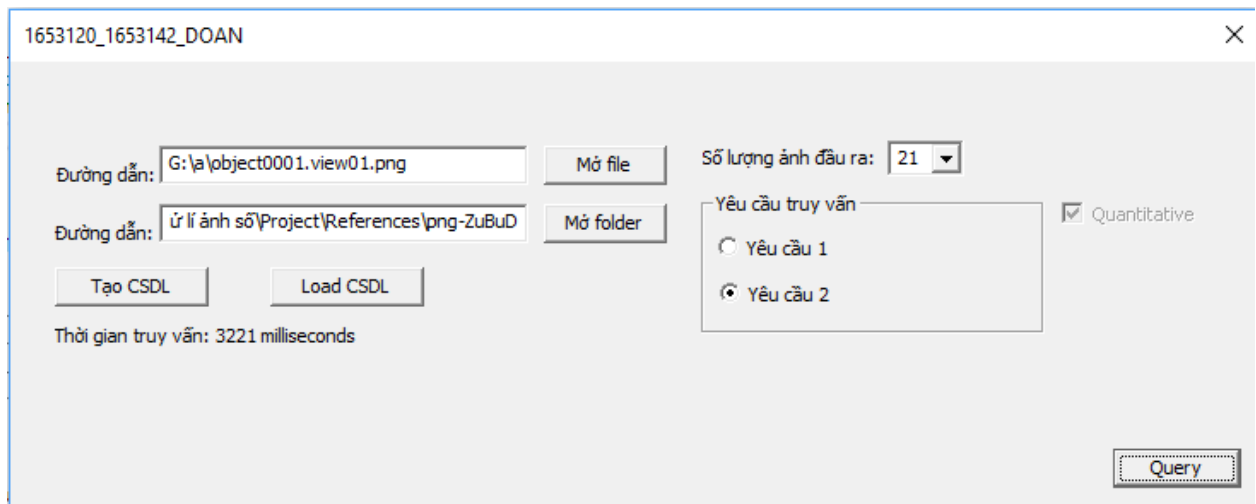
2) Kết quả.

a. Thời gian chạy



The screenshot shows a software window titled "1653120_1653142_DOAN (Not Responding)". It contains two text input fields for file paths, both with "Mở file" and "Mở folder" buttons. The first path is "G:\a\object0001.view01.png" and the second is "Ứ lí ảnh số\Project\References\png-ZuBuD". There are also "Tạo CSDL" and "Load CSDL" buttons. On the right, there is a dropdown menu for "Số lượng ảnh đầu ra:" set to "21", a "Yêu cầu truy vấn" section with radio buttons for "Yêu cầu 1" (selected) and "Yêu cầu 2", and a checked "Quantitative" checkbox. The "Thời gian truy vấn:" is displayed as "397 milliseconds". A "Query" button is in the bottom right corner.

Hình 1: Kết quả truy vấn với yêu cầu 1



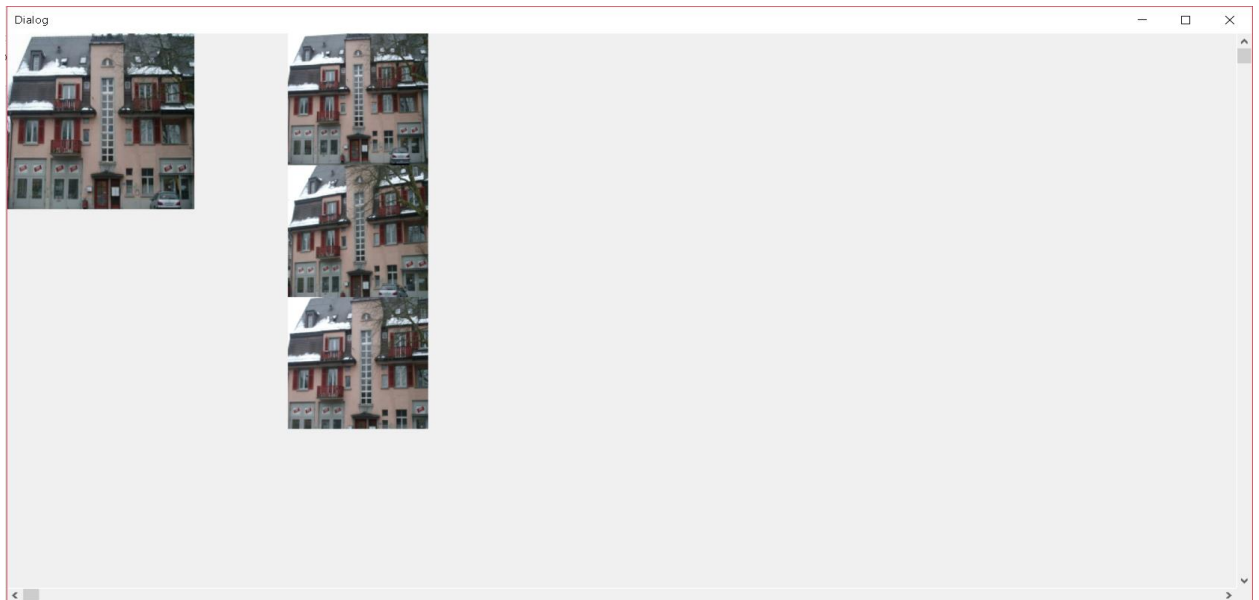
This screenshot is similar to the previous one but shows the application window titled "1653120_1653142_DOAN" without the "(Not Responding)" status. In the "Yêu cầu truy vấn" section, the radio button for "Yêu cầu 2" is now selected. The "Thời gian truy vấn:" is displayed as "3221 milliseconds". All other elements, including file paths, buttons, and the "Quantitative" checkbox, remain the same.

Hình 2: Kết quả truy vấn với yêu cầu 2

b. Độ đo MAP:

- Ảnh truy vấn: qimg0001.JPG trong QueryImages của ZuBuD.
- Database: 1005 ảnh của ZuBuD.
- Yêu cầu 1: Histogram
 - ✓ TRƯỜNG HỢP 1: KHI ĐÃ LƯỢNG HÓA MÀU

Số lượng ảnh trả về: 3

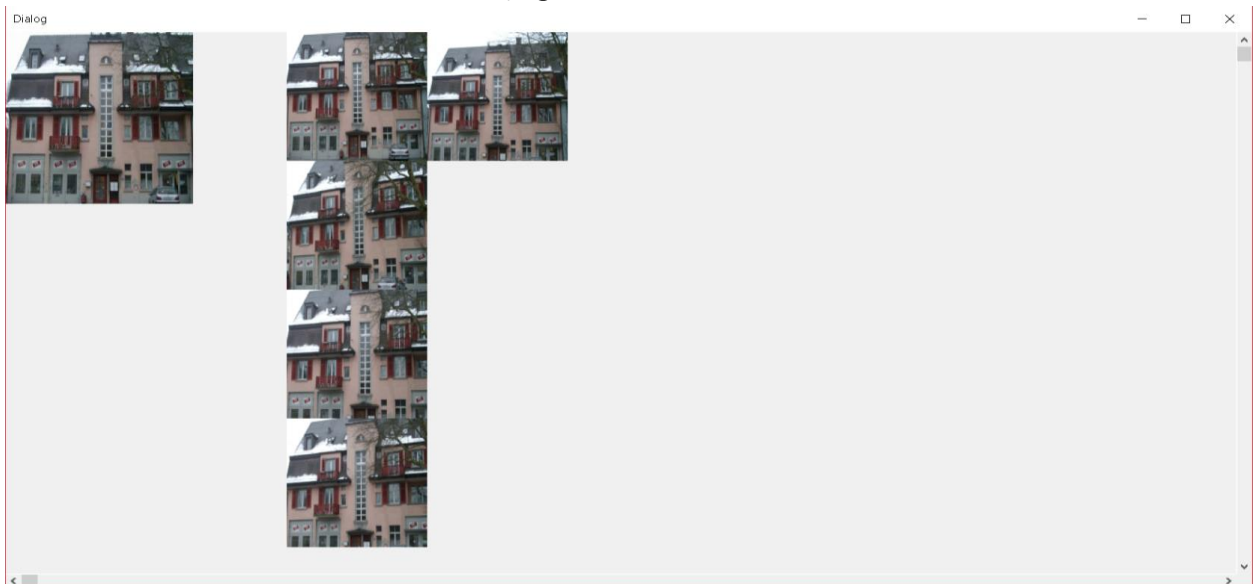


Số lượng ảnh trả về đúng: 3

Number	Recall	Relevant	Precision
1	0.33	1	1
2	0.67	1	1
3	1	1	1

$$AP = \frac{1+1+1}{3} = 1$$

Số lượng ảnh trả về: 5

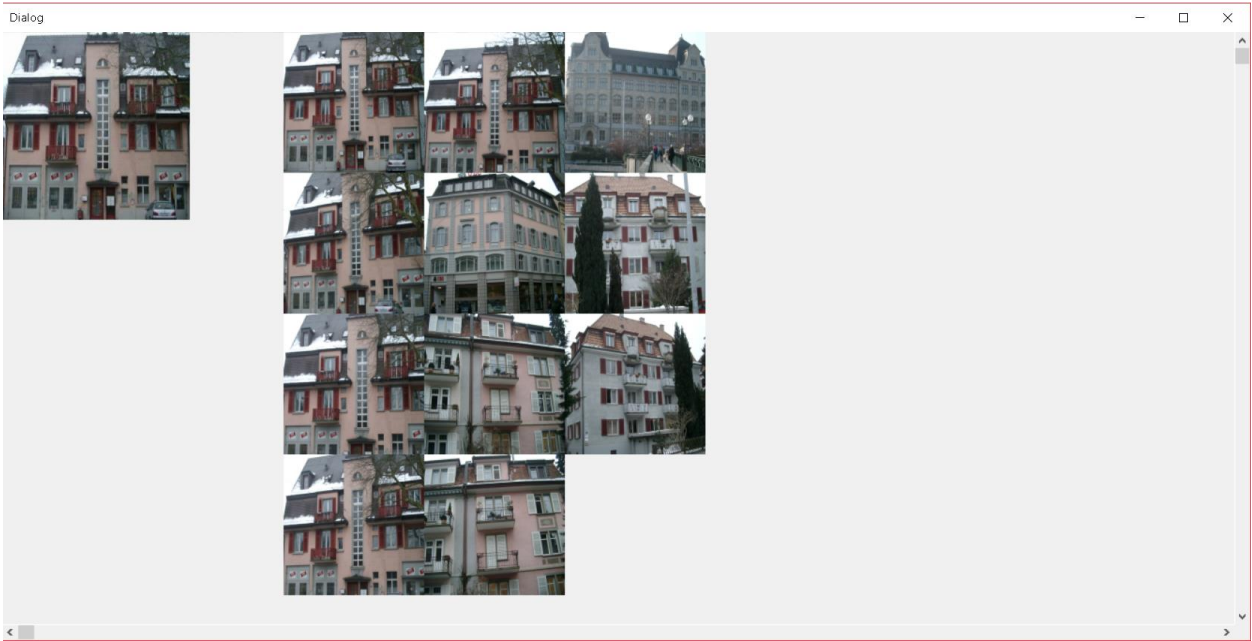


Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	1	1	1

$$AP = \frac{1+1+1+1+1}{5} = 1$$

Số lượng ảnh trả về: 11



Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	1	1	1
6	1	0	0.83
7	1	0	0.71
8	1	0	0.625
9	1	0	0.56
10	1	0	0.5
11	1	0	0.45

$$AP = \frac{1+1+1+1+1}{5} = 1$$

Số lượng ảnh trả về: 21



Số lượng ảnh trả về đúng: 5

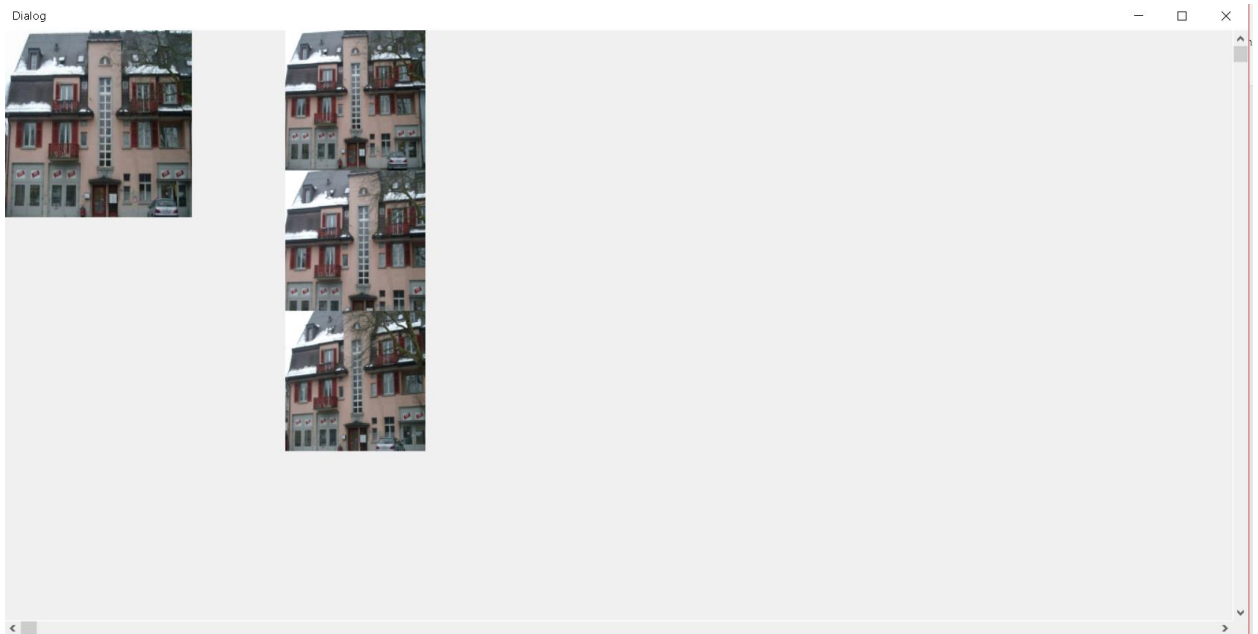
Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	1	1	1
6	1	0	0.83
7	1	0	0.71
8	1	0	0.625
9	1	0	0.56
10	1	0	0.5
11	1	0	0.45
12	1	0	0.42
13	1	0	0.38
14	1	0	0.36
15	1	0	0.33
16	1	0	0.31
17	1	0	0.29

18	1	0	0.28
19	1	0	0.26
20	1	0	0.25
21	1	0	0.24

$$AP = \frac{1+1+1+1+1}{5} = 1$$

$$\text{Vậy MAP} = \frac{1+1+1+1}{4} = 1$$

✓ TRƯỜNG HỢP 2: KHI CHƯA LƯỢNG HÓA MÀU
Số lượng ảnh trả về: 3

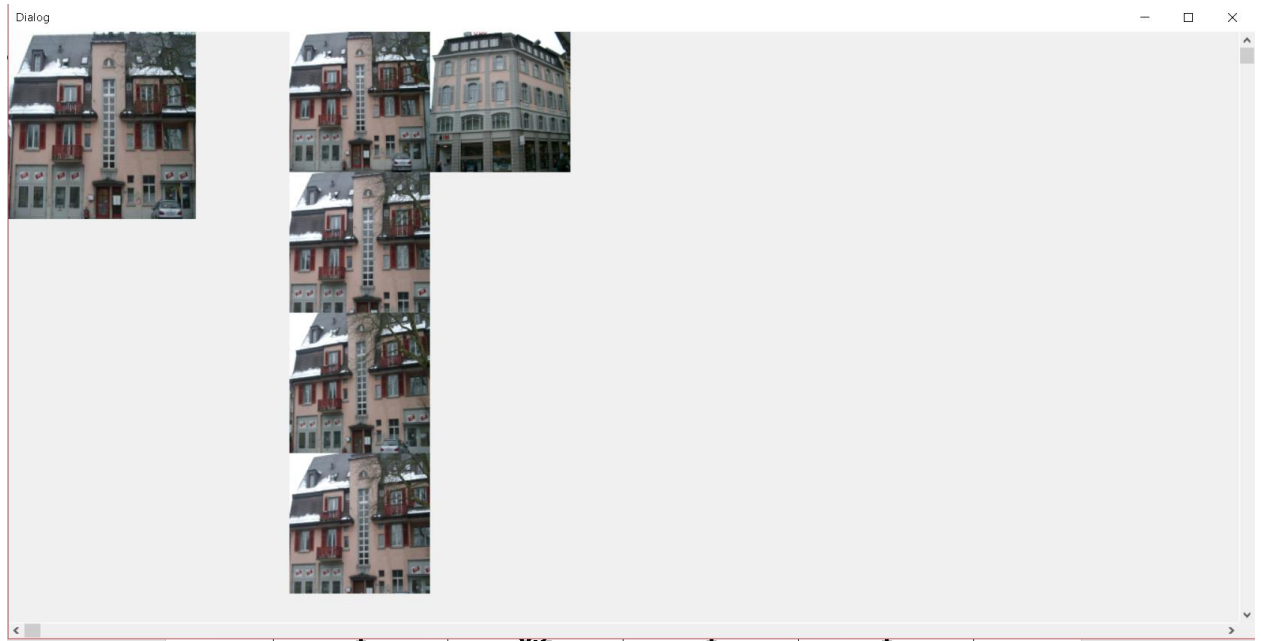


Số lượng ảnh trả về đúng: 3

Number	Recall	Relevant	Precision
1	0.33	1	1
2	0.67	1	1
3	1	1	1

$$AP = \frac{1+1+1}{3} = 1$$

Số lượng ảnh trả về: 5

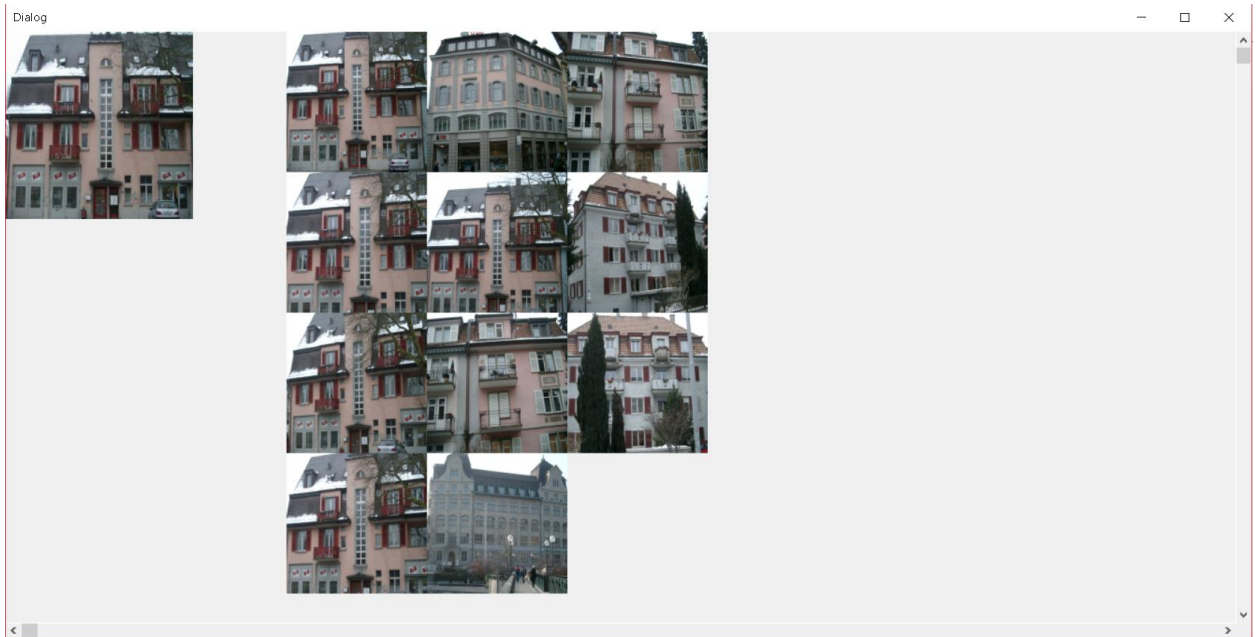


Số lượng ảnh trả về đúng: 4

Number	Recall	Relevant	Precision
1	0.25	1	1
2	0.5	1	1
3	0.75	1	1
4	1	1	1
5	1	0	0.8

$$AP = \frac{1+1+1+1}{4} = 1$$

Số lượng ảnh trả về: 11



Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	0.8	0	0.8
6	1	1	0.83
7	1	0	0.71
8	1	0	0.625
9	1	0	0.56
10	1	0	0.5
11	1	0	0.45

$$AP = \frac{1+1+1+1+0.83}{5} = 0.97$$

Số lượng ảnh trả về: 21



Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	0.8	0	1
6	1	1	0.83
7	1	0	0.71
8	1	0	0.625
9	1	0	0.56
10	1	0	0.5
11	1	0	0.45
12	1	0	0.42
13	1	0	0.38
14	1	0	0.36
15	1	0	0.33
16	1	0	0.31
17	1	0	0.29
18	1	0	0.28
19	1	0	0.26
20	1	0	0.25

21	1	0	0.24
----	---	---	------

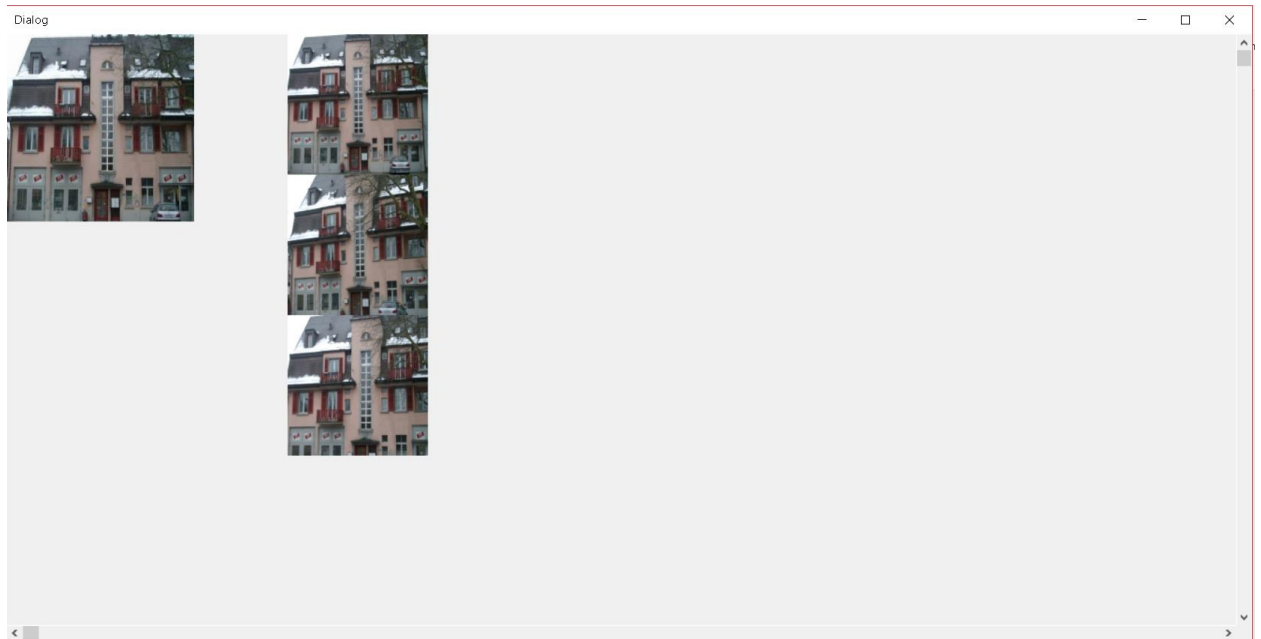
$$AP = \frac{1+1+1+1+0.83}{5} = 0.97$$

$$\text{Vậy MAP} = \frac{1+1+0.97+0.97}{4} = 0.985$$

Kết luận: MAP của histogram khi lượng hóa lớn hơn hẳn so với MAP của histogram khi chưa lượng hóa ($1 > 0.985$). Vậy sau khi lượng hóa màu thì sẽ tìm ra ảnh 1 cách chính xác.

- Yêu cầu 2: Sử dụng các đặc trưng

Số lượng ảnh trả về: 3

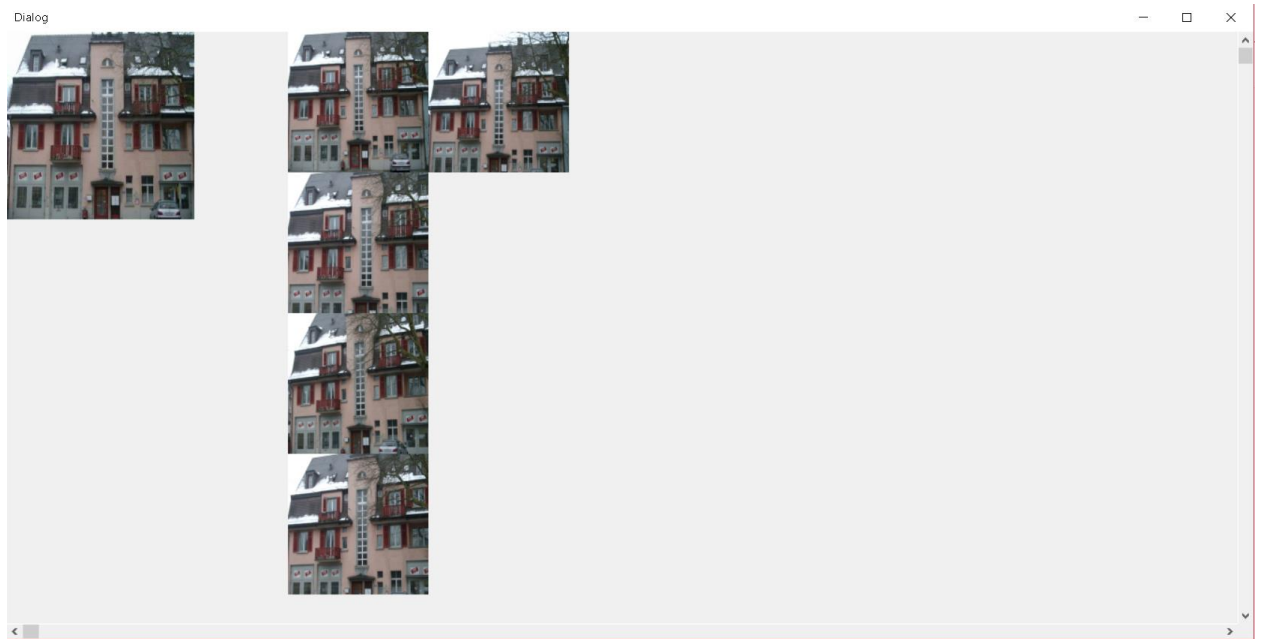


Số lượng ảnh trả về đúng: 3

Number	Recall	Relevant	Precision
1	0.33	1	1
2	0.67	1	1
3	1	1	1

$$AP = \frac{1+1+1}{3} = 1$$

Số lượng ảnh trả về: 5

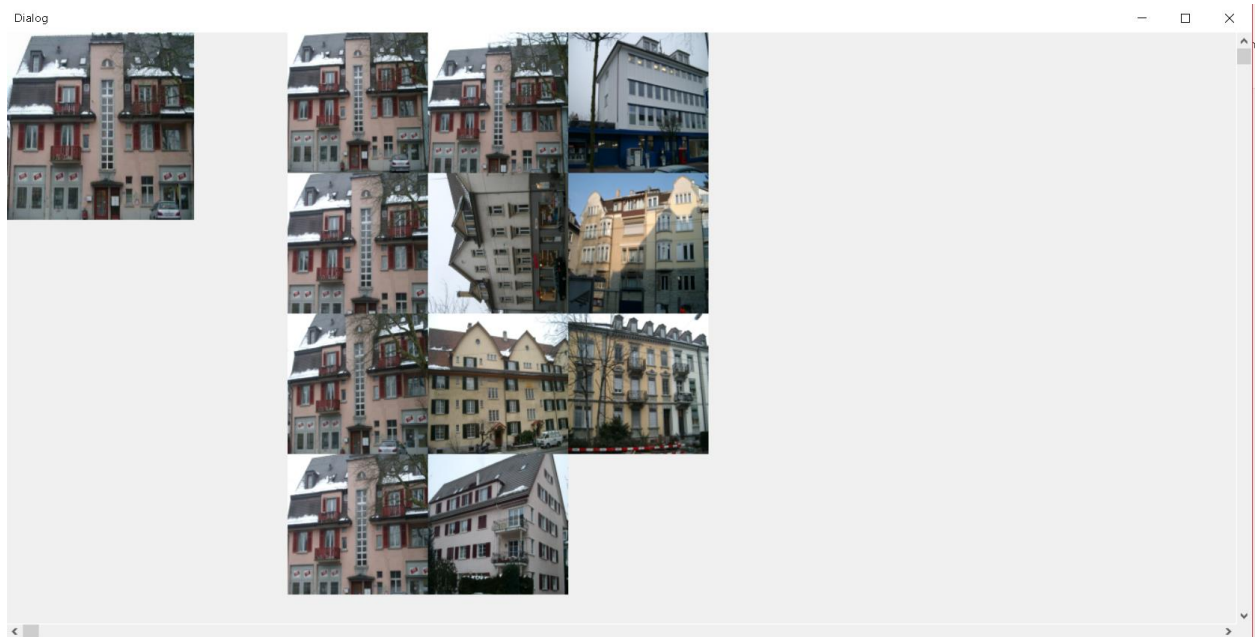


Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	1	1	1

$$AP = \frac{1+1+1+1+1}{5} = 1$$

Số lượng ảnh trả về: 11



Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	1	1	1
6	1	0	0.83
7	1	0	0.71
8	1	0	0.625
9	1	0	0.56
10	1	0	0.5
11	1	0	0.45

$$AP = \frac{1+1+1+1+1}{5} = 1$$

Số lượng ảnh trả về: 21



Số lượng ảnh trả về đúng: 5

Number	Recall	Relevant	Precision
1	0.2	1	1
2	0.4	1	1
3	0.6	1	1
4	0.8	1	1
5	1	1	1
6	1	0	0.83
7	1	0	0.71
8	1	0	0.625
9	1	0	0.56
10	1	0	0.5
11	1	0	0.45
12	1	0	0.42
13	1	0	0.38
14	1	0	0.36
15	1	0	0.33
16	1	0	0.31
17	1	0	0.29
18	1	0	0.28
19	1	0	0.26

20	1	0	0.25
21	1	0	0.24

$$AP = \frac{1+1+1+1+1}{5} = 1$$

$$\Rightarrow \text{Vậy MAP} = \frac{1+1+1+1}{4} = 1$$

Kết luận: MAP của bộ truy vấn khi sử dụng đặc trưng so với khi so sánh bằng histogram đã lượng hóa thì hiệu quả tương đương ($1 = 1$) nhưng với histogram khi chưa lượng hóa thì hiệu quả cao hơn ($1 > 0.985$). Vậy khi sử dụng đặc trưng cũng sẽ tăng chính xác kết quả tìm kiếm.

3) Một số vấn đề cần lý giải về đề án:

a. Thuật toán Shell sort:

- Mục đích: cải thiện thời gian sắp xếp mảng.
- Giải thuật:
 - ✓ Bước 1: Khởi tạo giá trị h
 - ✓ Bước 2: Chia list thành các sublist nhỏ hơn tương ứng với h
 - ✓ Bước 3: Sắp xếp các sublist này bởi sử dụng sắp xếp chèn (Insertion Sort)
 - ✓ Bước 4: Lặp lại cho tới khi list đã được sắp xếp

b. Thuật toán ORB:

- Đối với yêu cầu 2, ta sẽ sử dụng thuật toán ORB (Oriented FAST and Rotated BRIEF) để detect các đặc trưng (cụ thể ở đây là đặc trưng cạnh) dựa trên 1 thuật toán được gọi là Harris Corner Detector. Harris Corner là 1 thuật toán rút trích ra các góc và suy ra các đặc trưng của 1 ảnh mà ở đó các góc được tạo bởi 2 hướng cạnh khác nhau.
- Về thuật toán ORB, có thể thấy trong tên sẽ có tên của 2 thuật toán khác đó là FAST và BRIEF. Tại sao nó lại chứa tên của 2 thuật toán như vậy? Thì ta có thể lý giải như sau, đầu tiên ORB sẽ detect ra các keypoints của 1 ảnh (keypoints là các điểm nổi bật của 1 ảnh) bằng thuật toán FAST. Nhưng thuật toán FAST là 1 thuật toán thiên về detect nên sẽ không thể nào rút trích đặc trưng nếu chỉ có 1 mình nó nên ta sẽ cần tới 1 thuật toán khác để rút trích các đặc trưng thông

qua các keypoints. Ở đây, ta sẽ sử dụng thuật toán BRIEF để tạo ra các descriptors thực hiện việc tính toán dựa trên các keypoints và rút trích đặc trưng. Sau đó ta sẽ tìm ra các keypoints matching giữa 2 ảnh.

- Quá trình cài đặt thuật toán như sau:

- ✓ Đầu tiên ta sẽ detect các keypoints của 2 ảnh trước bằng cách tạo 1 biến con trỏ kiểu ORB:

```
Ptr<ORB> detector = ORB::create();
```

- ✓ Sau đó ta tạo ra vector kiểu KeyPoint để lưu trữ keypoints của ảnh:

```
vector<KeyPoint> keypoints;
```

- ✓ Và sau đó khai triển hàm detect để detect các keypoints:

```
detector->detect(src, keypoints);
```

- ✓ Sau khi detect xong các keypoints việc tiếp theo chúng ta sẽ làm là phải rút trích được đặc trưng dựa trên các keypoints đã được detect. Thì ta sẽ tạo ra 1 biến con trỏ kiểu ORB nhưng lần này sẽ là con trỏ để rút trích đặc trưng nên kiểu dữ liệu con trỏ sẽ là DescriptorExtractor thay vì FeatureDetector như trên:

```
Ptr<DescriptorExtractor> descriptor =  
ORB::create();
```

- ✓ Sau đó ta sẽ tạo ra biến descriptors kiểu Mat để chuẩn bị cho việc lưu trữ các đặc trưng:
- ✓ Ta sẽ khai triển hàm compute để tính toán và rút trích các đặc trưng và lưu vào 2 biến descriptors:

```
descriptor->compute(src, keypoints,  
descriptors);
```

- ✓ Việc tiếp theo là ta sẽ tìm ra các đặc trưng giống nhau giữa 2 ảnh bằng cách tạo ra 1 biến matcher khởi tạo bằng thuật toán FlannBased.
- ✓ Sau đó ta sẽ tạo ra 1 vector 2 chiều kiểu DMatch để lưu trữ các keypoints được tìm thấy và khai triển hàm knnMatch để

lưu trữ các keypoints đã được tìm ra của 2 descriptors và lưu vào vector đã tạo:

```
vector<vector<DMatch>> matches; //Tạo 1 mảng vector 2 chiều lưu  
matcher.knnMatch(descriptorsSrc, descriptorsDst, matches, 2);  
vector<DMatch> good_matches;
```

- ✓ Vì cách tìm kiếm này có khuyết điểm là so sánh quá chính xác hầu như 2 ảnh nào cũng cho ra số keypoints matching bằng với keypoints của ảnh gốc tức là 2 ảnh được cho là giống nhau. Nên chúng ta cần lọc các keypoints matching bằng 1 phương pháp gọi là Ratio Test.
- ✓ Ratio Test thực hiện như sau: Xác suất để 2 điểm matches là chính xác có thể được xác định bằng cách lấy tỉ lệ khoảng cách từ điểm gần nhất tới khoảng cách của điểm thứ 2 gần nhất. Chúng ta sẽ loại bỏ các matches có tỉ lệ khoảng cách nhỏ hơn 0.75.

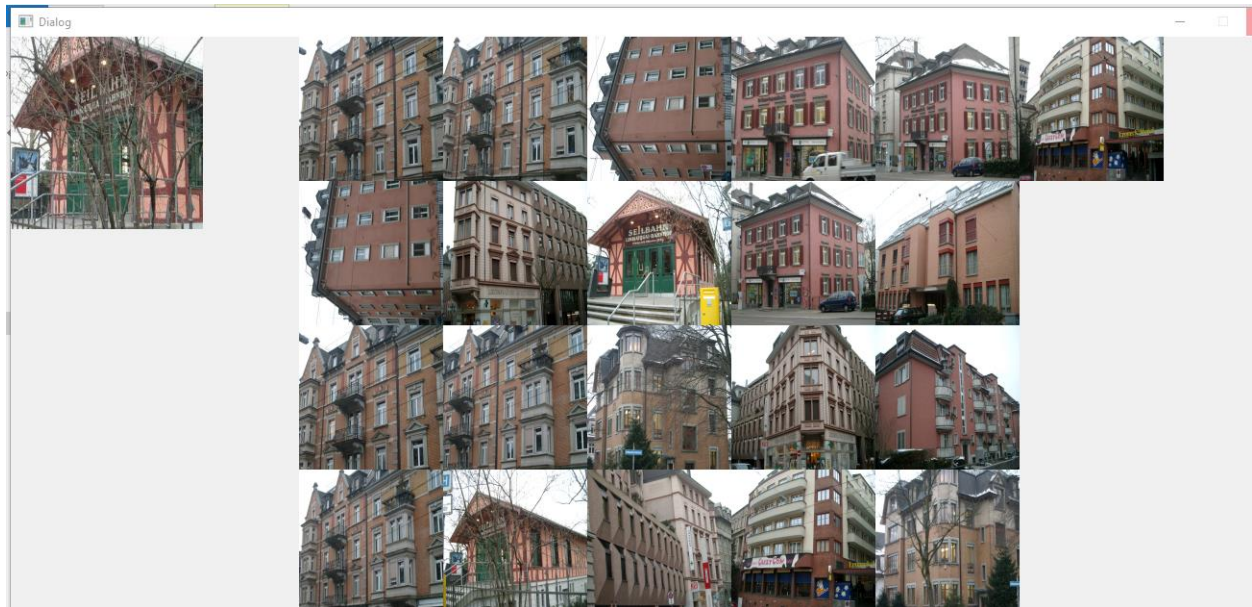
```
vector< DMatch > good_matches;  
for (unsigned int i = 0; i < matches.size(); ++i)  
{  
    if (matches[i][0].distance < matches[i][1].distance * 0.75)  
        good_matches.push_back(matches[i][0]);  
}  
return good_matches.size(); // Trả về số lượng điểm good match .
```

- ✓ Bằng cách so sánh tỉ lệ 2 neighbors gần nhất với 1 tỉ lệ là 0.75 thì ta sẽ lọc ra được các matches tốt hơn với ban đầu.
- ✓ Với cách detect và rút trích đặc trưng như trên ta có thể dễ dàng so sánh mức độ giống của 1 loạt ảnh với 1 ảnh được truy vấn dựa trên số keypoints matching.

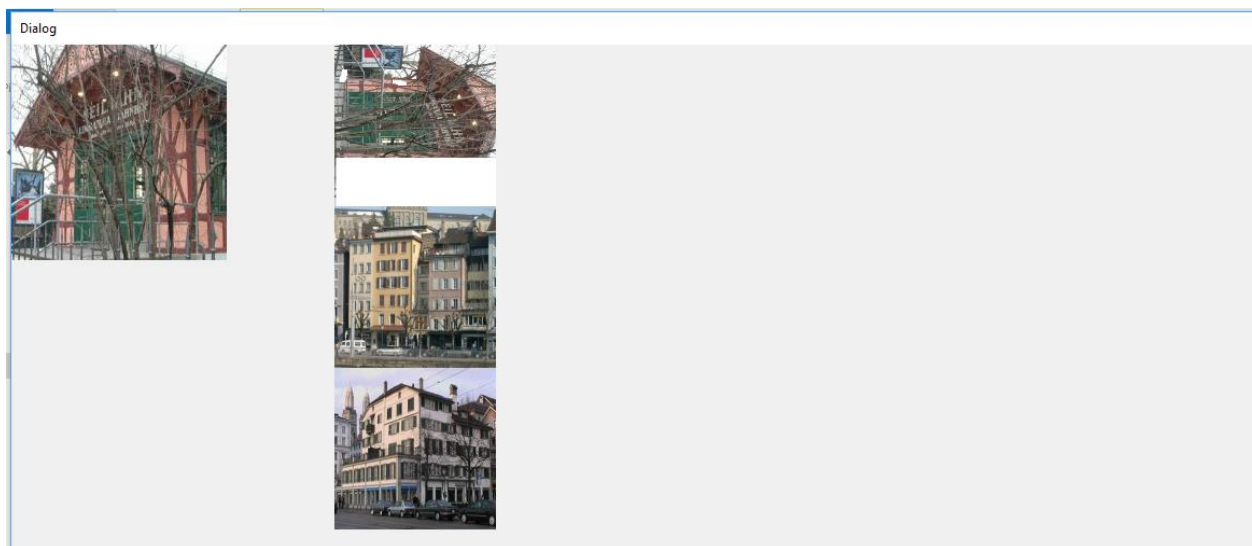
c. So sánh giữa 2 phương pháp:

Sau đây là các case chứng tỏ thuật toán ORB tốt hơn so với so sánh lượng hóa màu:

Yêu cầu 1 tức lượng hóa màu sử dụng số lượng trả về là 21 nhưng vẫn không tìm ra ảnh đã bị rotate với cắt bớt 1 phần.

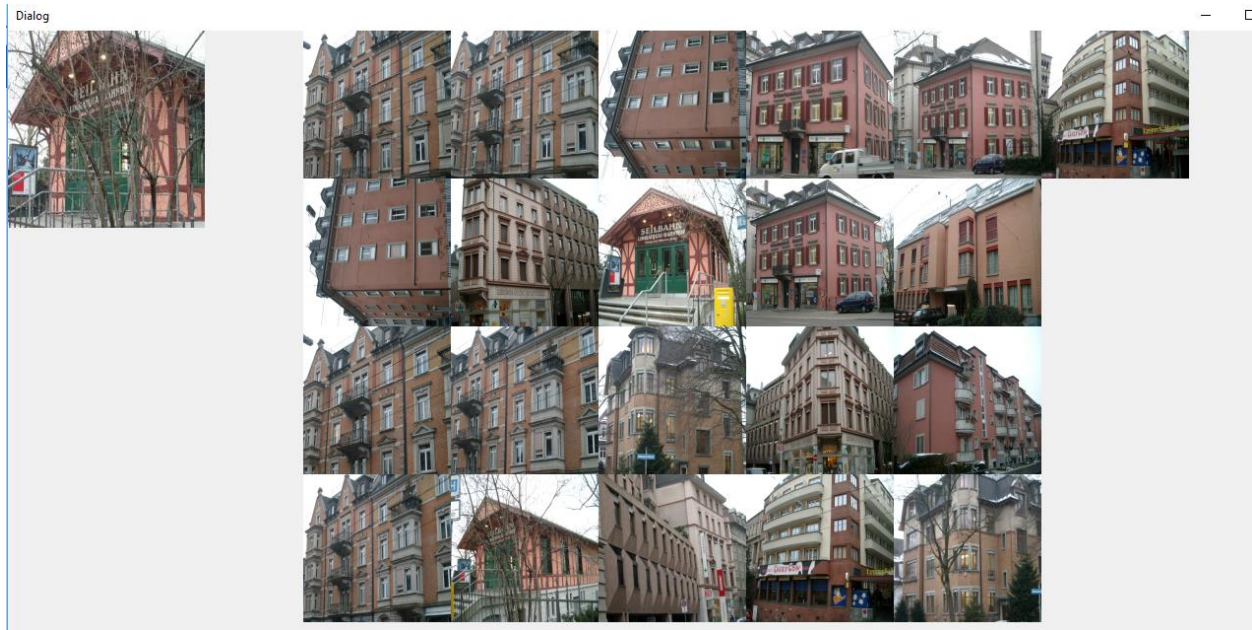


Nhưng với yêu cầu 2 thì lại detect ra. Vì thế mạnh của yêu cầu 1 nằm ở giá trị và tần số các điểm sáng đó cũng là 1 điểm yếu của yêu cầu 1 khi sử dụng lượng hóa màu.

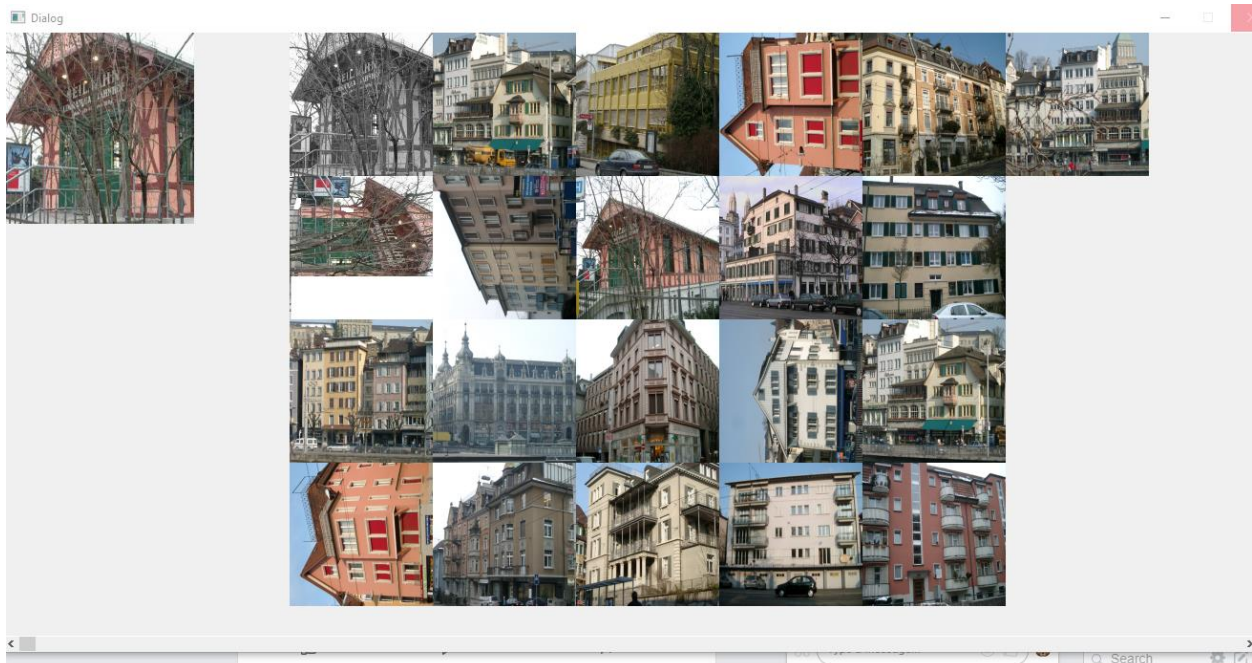


Tiếp theo, ta thử so sánh với database có 1 ảnh y chang ảnh được truy vấn nhưng đã được chuyển về dạng GrayScale:

Như đã thấy lần này yêu cầu 1 sử dụng số lượng 21 ảnh trả về nhưng vẫn không tìm ra



Nhưng với yêu cầu 2 thì lại tìm ra dễ dàng chỉ với 3 kết quả trả về



Nhưng đây chỉ là kết quả khi sử dụng Ratio Test để cải thiện các điểm matches của thuật toán ORB.

Khi chưa sử dụng Ratio Test thì hầu như độ chính xác của ORB không cao so với các thuật toán khác như là :SIFT hoặc SURF.Vì Scale Invariance của ORB không quá mạnh.Ngoài ra,

khi background bị nhiễu thì kết quả trả về sẽ nhận được vô số các đặc trưng không chính xác.

Khi so sánh các ảnh không cùng size thì xác suất tìm được ảnh chính xác hầu như bằng 0. Vì vậy ta phải cải tiến thêm cho ORB bằng cách sử dụng Ratio Test.

Mặt khác, thuật toán này chạy khá nhanh so với hầu hết các thuật toán detect đặc trưng các. Đây cũng là 1 thế mạnh của thuật toán ORB.