**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH**

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN THỊ GIÁC MÁY**

**Nhóm 4: GVHD**: Nguyễn Đức Thành

Diệp Vĩ Cường – 1510347

Nguyễn Anh Tuấn – 1513834

Nguyễn Minh Nhựt – 1512374

Võ Thanh Phong – 1512464



## Đề tài: Dùng các ảnh hay video chụp đèn tín hiệu giao thông, nhận dạng đèn xanh vàng đỏKết quả hình ảnh cho traffic light recognition

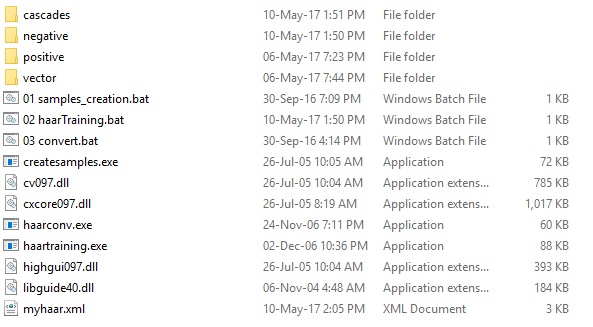
## Ý tưởng

Ứng dụng sẽ sử dụng song song hai chức năng là khoanh vùng đèn giao thông và xác định màu của đèn. Đầu tiên, dựa vào tập huấn luyện xác định vị trí của đèn, hiển thị lên cho người dùng. Sau đó, duyệt trên các vùng đã xác định, kiểm tra xem có tồn tại một trong ba màu (xanh, đỏ, vàng) hay không. Nếu có thì sẽ hiển thị thông báo bằng cách vẽ một hình tròn có màu tương ứng dưới góc phải màn hình. Nếu ba màu trên không tồn tại trong vùng xác định thì sẽ không có thông báo nào cả để tránh hiện tượng nhận sai các vùng bị nhiễu.

## Thực hiện

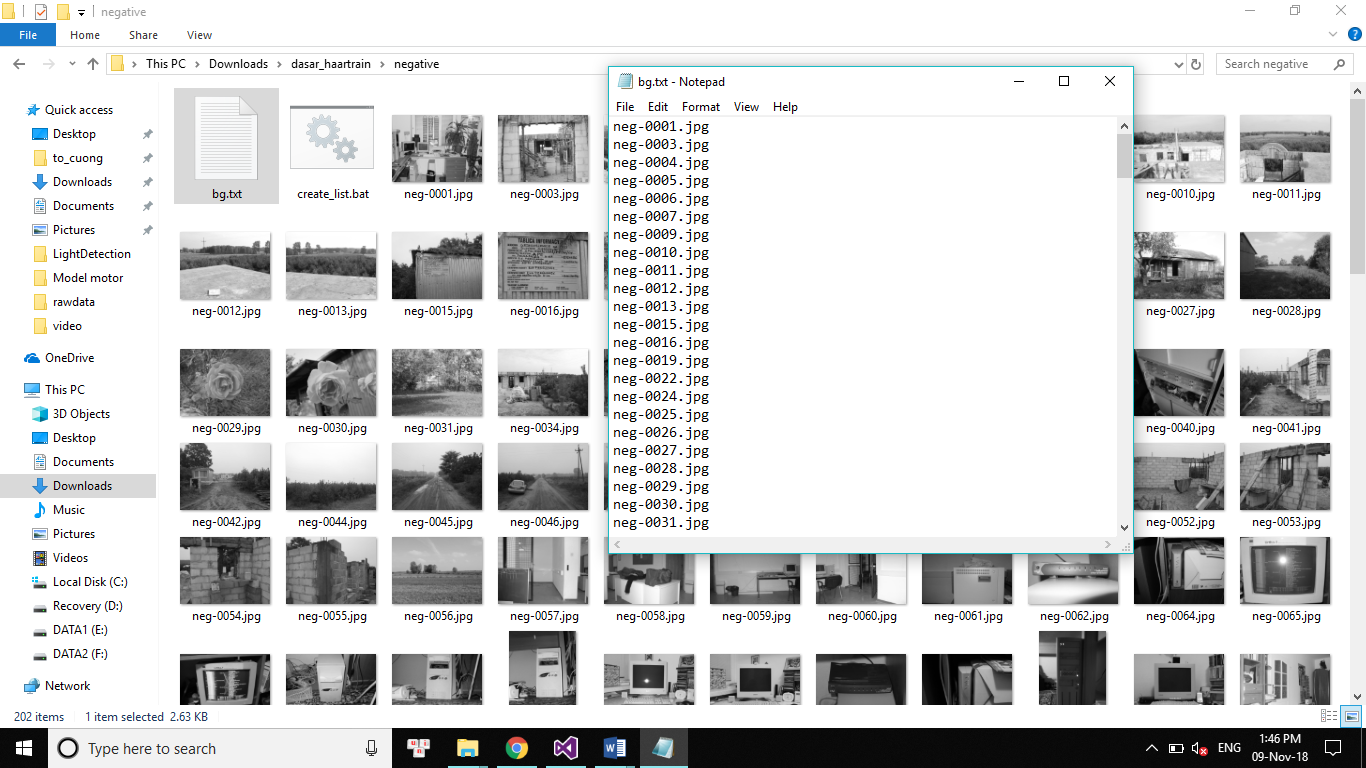
### 1. Huấn luyện

Tải tool huấn luyện từ link bên dưới: http://www.mediafire.com/file/1aq02tpidk105fv/dasar\_haartrain.rar



Hình 1.1 Các tập tin trong thư mục Haar training.

* Bước 1: Tạo danh sách Negative.
* Chuyển tất cả dữ liệu hình Negative vào thư mục **rawdata** trong thư mục **negative**.
* Chạy file **create\_list.bat** để tạo file **bg.txt** chứa tên của các file ảnh

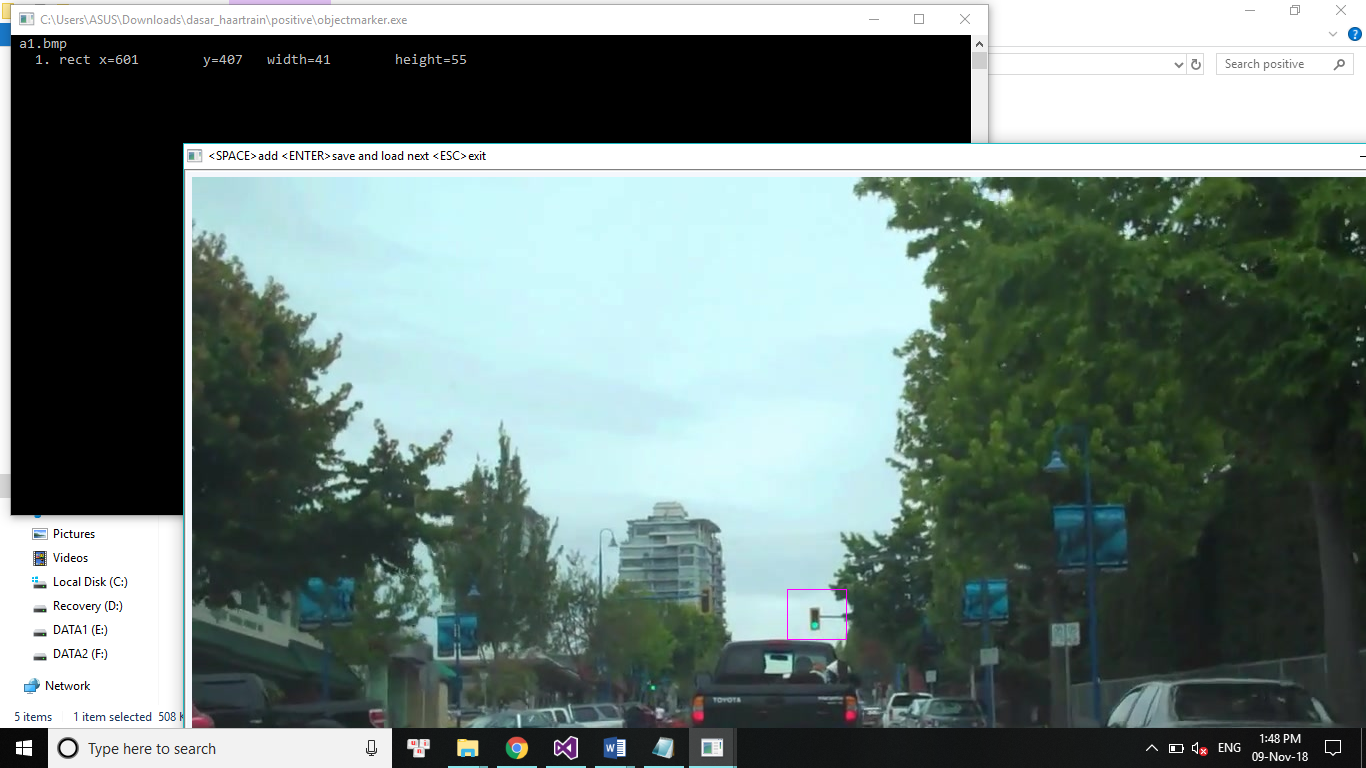


Hình 1. Tạo danh sách Negative.

* Bước 2: Tạo danh sách Positive.

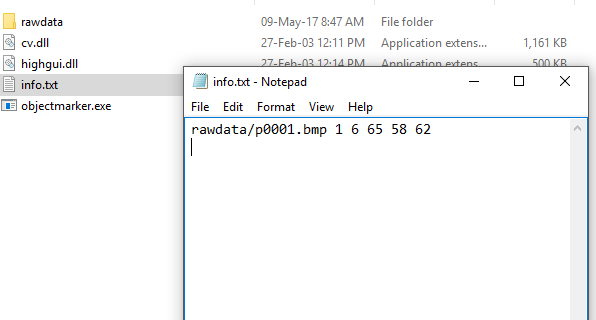
- Chuyển tất cả dữ liệu hình Positive vào thư mục **rawdata** trong thư mục **positive**.

- Chạy tập tin **objectmarker.exe** để lưu lại vị trí của đối tượng trong hình.



Hình 1. Tạo danh sách Positive.

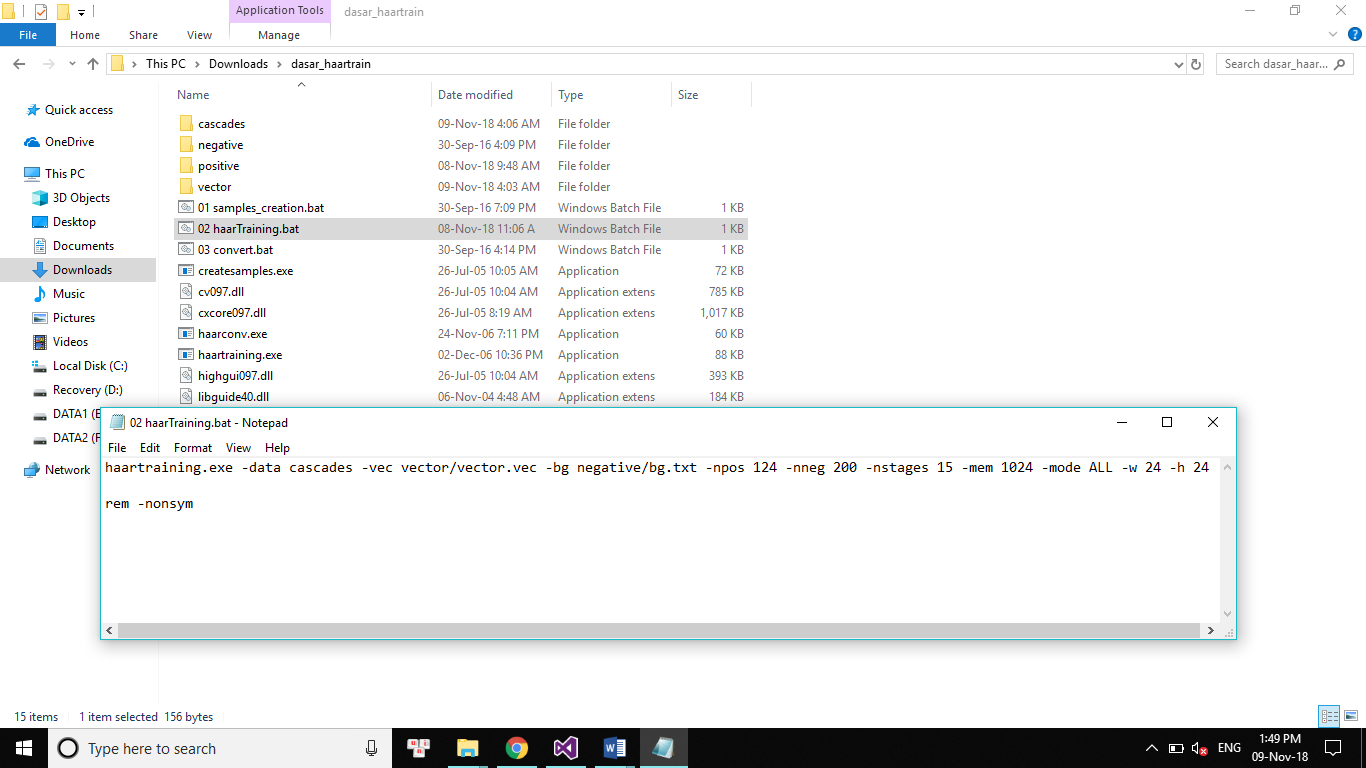
- Tập dữ liệu positive sẽ được lưu lại thành một danh sách trong tập tin **info.txt**.



Hình 1. Danh sách Positive sau khi tạo xong.

* Bước 3: Chỉnh sửa thông số huấn luyện.

- Nhấp chuột phải vào tập tin **haarTraining.bat** chọn **edit** để chỉnh sửa thông số huấn luyện.



Hình 1. Chỉnh sửa thông số huấn luyện.

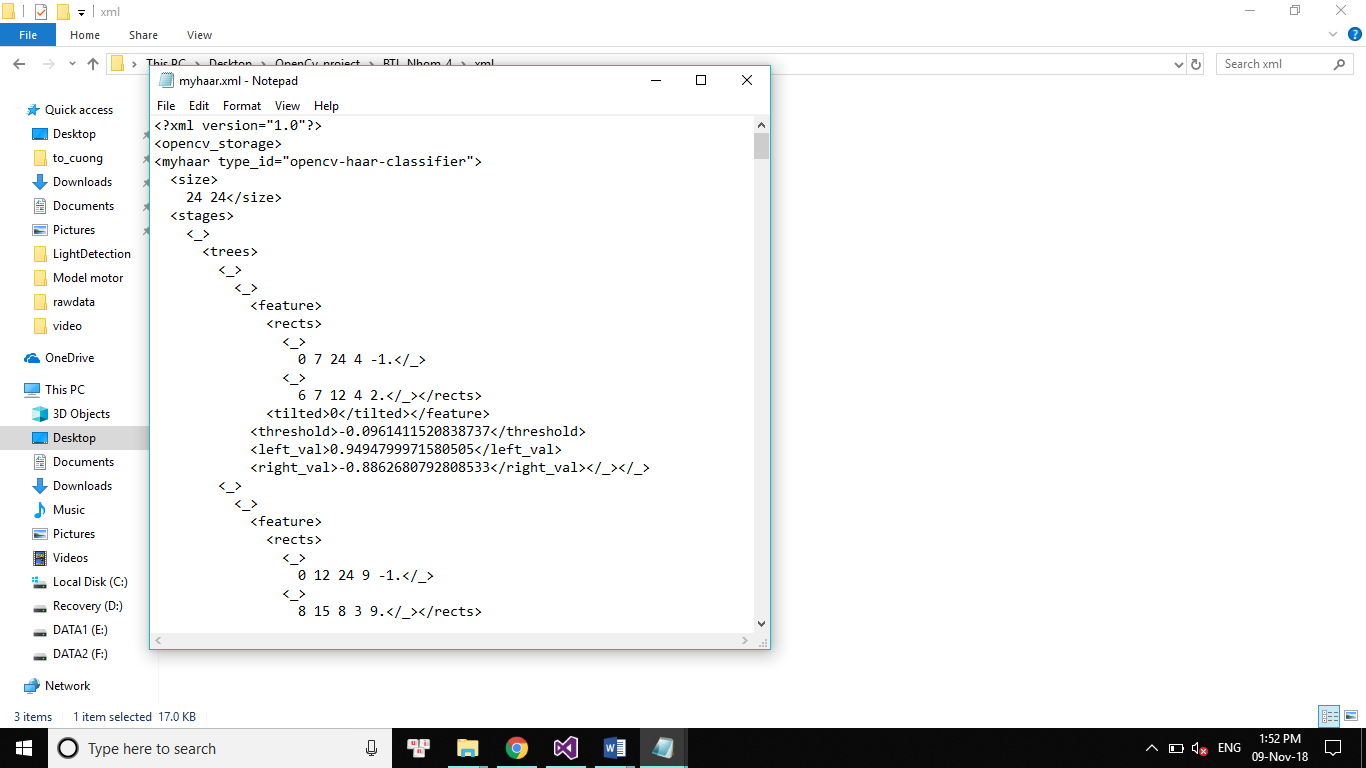
-npos <số lượng ảnh positive>.

-nneg <số lượng ảnh negative>.

* Bước 4: Huấn luyện.

- Chạy lần lượt 3 tập tin theo thứ tự samples\_creation.bat, haarTraining.bat và convert.bat.

- Kết quả cuối cùng sẽ tạo ra một tập tin .xml chứa thông tin dữ liệu hình ảnh đã được mã hóa.



Hình 1. Thông số trong tập tin xml sau khi luyện.

### 2.Nhận biết màu

#### Dải màu hsv của đèn giao thông

Sử dụng hàm **inRange** trong openCV để lấy các vùng màu cần thiết trong hình hoặc khung hình (lưu ý: phải chuyển sang dải màu HSV trước khi dùng hàm **inRange**). Các vùng màu cần lấy lần lượt là:

- Màu đỏ:

<1>: Scalar(0, 100, 100) -> Scalar(10, 255, 255)

<2>: Scalar(160, 100, 100) -> Scalar(179, 255, 255)

- Màu vàng:

<1>: Scalar(20, 100, 100) -> Scalar(30, 255, 255)

- Màu xanh lá:

<1>: Scalar(30, 59, 50) -> Scalar(79, 158, 171)

<2>: Scalar(0, 255, 0) -> Scalar(229, 255, 204)

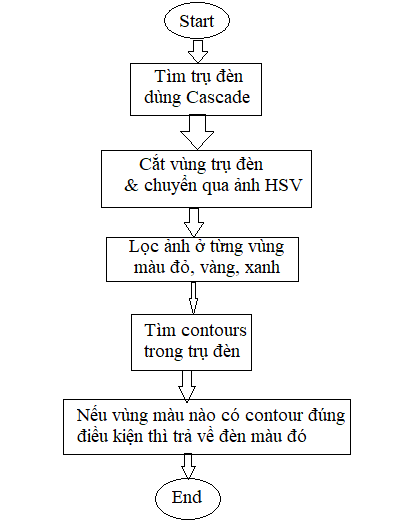
#### Tìm vị trí màu trên ảnh

Dùng hàm **findContours** của openCV để nhận biết viền trong mỗi threshold của từng màu. Sau đó, sử dụng lớp đối tượng **Moments** của openCV để xác định vị trí và độ lớn của các cụm màu trong threshold.

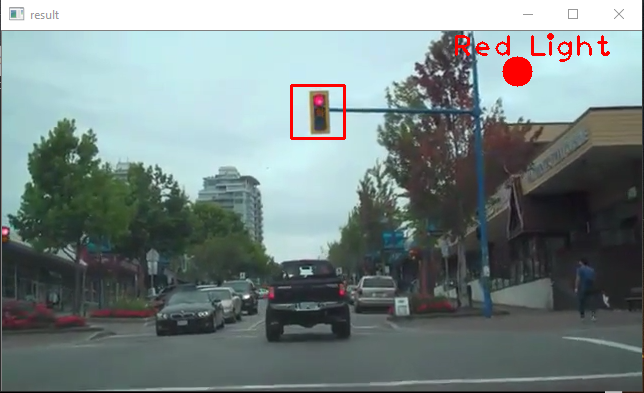
#### Kết hợp nhận diện màu với khoanh vùng

Sử dụng vòng lặp, cứ mỗi khoanh vùng đối tượng thì kiểm tra bên trong có cụm màu nào không. Nếu có thì trả về màu nhận biết được (khi kiểm tra màu nên sử dụng lệnh **else if** để lấy duy nhất một màu đèn, tránh hiện tượng 1 lúc hiển thị 3 màu đèn).

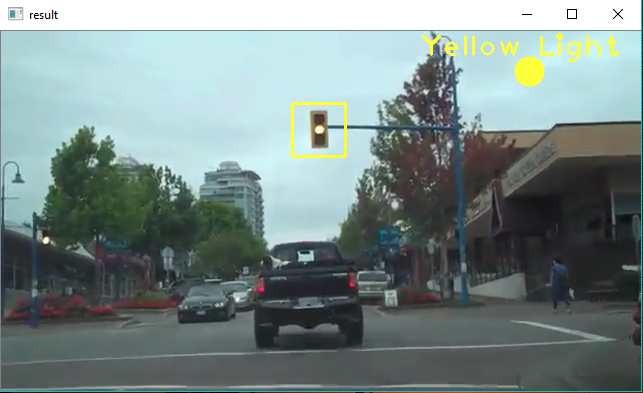
#### Lưu đồ giải thuật



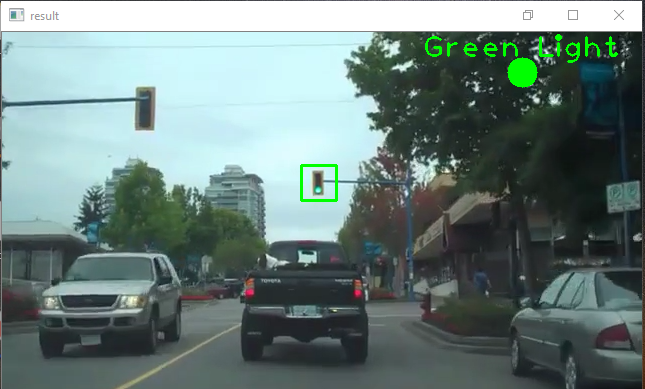
## Kết quả :



Hình 3. Nhận diện tín hiệu đèn đỏ



Hình 3.2 Nhận diện tín hiệu đèn vàng.



Hình 3.3 Nhận diện tín hiệu đèn xanh.

## IV. Nhận xét

- Tỉ lệ nhận dạng trên 90%. Hầu hết đều nhận và khoanh vùng được đèn.

- Dù đã khử nhiễu rất nhiều nhưng vẫn chưa triệt để, nguyên nhân là do tập dữ liệu hình còn ít nên ứng dụng vẫn chưa đủ một số dữ kiện cho các hình dạng khó như: cây cối, cột đèn đường,… và các cấu trúc vật thể khác có hình dạng giống như đèn giao thông.

## KẾT LUẬN



## Ưu điểm

* Nhận biết được hình dạng đèn giao thông và phân biệt được trạng thái của màu đèn.
* Khử nhiễu tốt.

## Nhược điểm

* Thuật toán khá nặng dẫn đến việc xử lý lâu khi áp dụng trên video và real time.
* Vẫn còn nhiễu tại một số điểm nhỏ dẫn đến sai sót đôi chút.
* Tập huấn luyện hình ảnh vẫn khá ít.

## Hướng phát triển

* Tối ưu và cải thiện thuật toán để quá trình xử lý tốt hơn.
* Tăng số lượng tập huấn luyện (khoảng trên 1000 hình positive, và trên 1000 hình negative) để giảm nhiễu và nhận diện chính xác hơn.
* Cập nhật thêm tính năng nhận biết thời gian chờ của đèn giao thông.