KIỂM TRA

Họ tên sinh viên: Nguyễn Thành Đạt

MSSV: 61130137

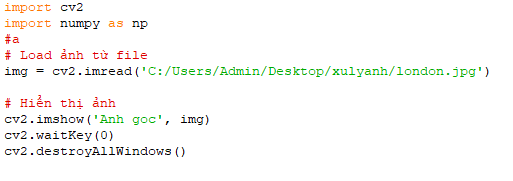
Câu 1

a)

Để load và xem một file ảnh bất kỳ trong Python, chúng ta có thể sử dụng thư viện OpenCV.

Đầu tiên, ta cần cài đặt OpenCV bằng cách sử dụng pip: pip install opencv-python

Sau đó, ta có thể sử dụng hàm cv2.imread() để đọc file ảnh và hàm cv2.imshow() để hiển thị ảnh lên màn hình.



Ở đât cv2.imread là hàm đọc file từ đường dẫn link ảnh có trong bộ nhớ, tham số đầu tiên của hàm cv2.imshow() là tên cửa sổ, tham số thứ hai là ảnh cần hiển thị. Hàm cv2.waitKey() dùng để chờ người dùng nhấn phím bất kỳ để đóng cửa sổ ảnh. Nếu không có lệnh này, cửa sổ ảnh sẽ bị đóng ngay lập tức sau khi hiển thị.

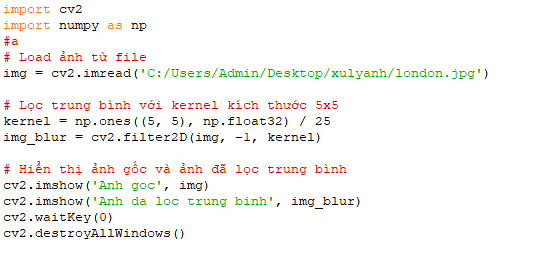


Ảnh được hiển thị sau khi load và view ảnh

b)

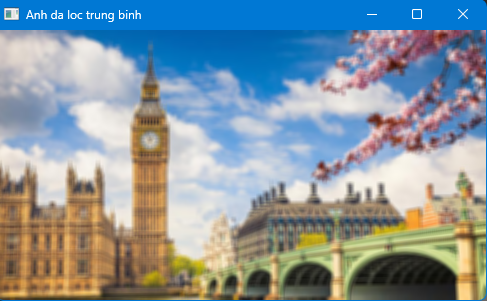
Để lọc trung bình một ảnh trong Python, ta có thể sử dụng hàm cv2.blur() của thư viện OpenCV. Hàm này thực hiện việc tính trung bình cộng của các pixel trong vùng lân cận của từng pixel trong ảnh.

Dưới đây là code ví dụ về cách lọc trung bình một ảnh và hiển thị kết quả:



Trong đó, hàm cv2.blur() được sử dụng để thực hiện bộ lọc trung bình trên ảnh. Tham số thứ nhất là ảnh cần lọc và tham số thứ hai là kích thước của kernel, tức là kích thước của cửa sổ lân cận được sử dụng để tính trung bình.

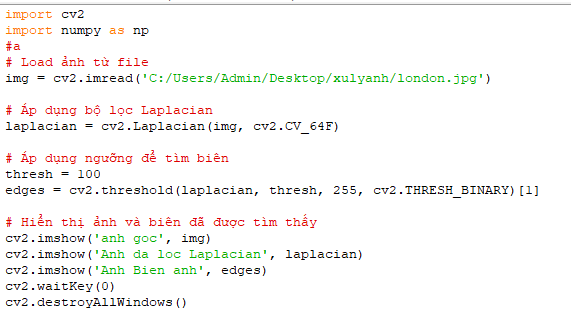
Sau đó, ta hiển thị ảnh gốc và ảnh đã được lọc bằng hàm cv2.imshow(). Ở đây, tham số đầu tiên là tên cửa sổ, tham số thứ hai là ảnh cần hiển thị.



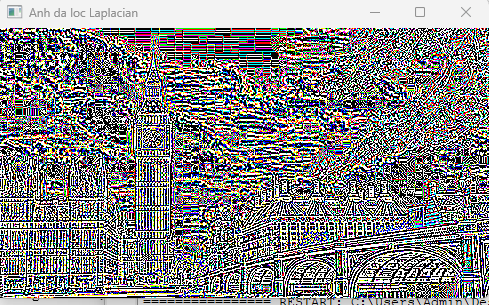
Kết quả ảnh sau khi lọc trung bình

c)

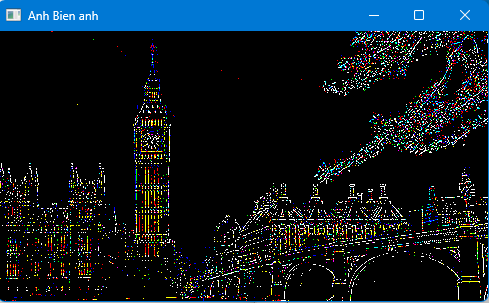
Để thực hiện lọc gradient tìm biên của ảnh Laplacian trong Python, chúng ta có thể sử dụng hàm cv2.Laplacian() để tính gradient của ảnh và sau đó áp dụng ngưỡng để tìm các biên.



Trong đó, hàm cv2.Laplacian() được sử dụng để áp dụng bộ lọc Laplacian trên ảnh và tính gradient. Sau đó, chúng ta áp dụng ngưỡng để tìm các biên bằng cách sử dụng hàm cv2.threshold(). Biên sẽ được gán màu trắng (giá trị 255) và các pixel không phải biên sẽ được gán màu đen (giá trị 0).



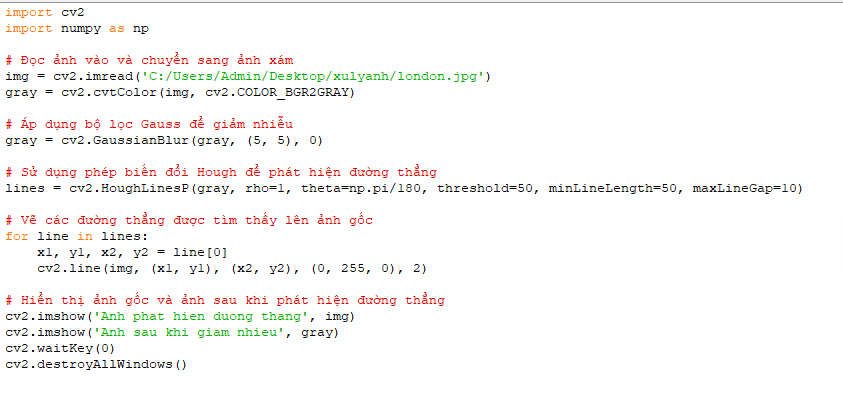
Ảnh sau khi lọc Laplacian



Biên của ảnh

d)

Để thực hiện phép biến đổi Hough để phát hiện đường thẳng trong ảnh bằng Python, ta cần sử dụng thư viện OpenCV. Dưới đây là các bước thực hiện:



B1: Import thư viện OpenCV và đọc ảnh đầu vào:

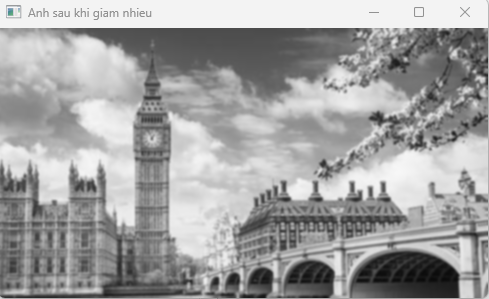
B2: Chuyển ảnh đầu vào sang ảnh xám.

B3: Áp dụng phép lọc Gaussian để loại bỏ nhiễu.

B4: Áp dụng phép biến đổi Canny để phát hiện cạnh.

B5: Thực hiện phép biến đổi Hough để phát hiện đường thẳng.

Trong đó, tham số đầu tiên của phép biến đổi Hough là edges, là ảnh được tạo ra từ phép biến đổi Canny ở bước 4. Tham số thứ hai là rho, khoảng cách từ gốc tọa độ đến đường thẳng. Tham số thứ ba là theta, góc giữa đường thẳng và trục ngang. Tham số cuối cùng là threshold, là ngưỡng số lượng điểm ảnh nằm trên đường thẳng để phát hiện một đường thẳng.



ảnh sau khi giảm nhiễu



Ảnh phát hiện đường thẳng

--

- Để phát hiện đường tròn, chúng ta sẽ sử dụng phép biến đổi Hough trong không gian tham số của đường tròn.

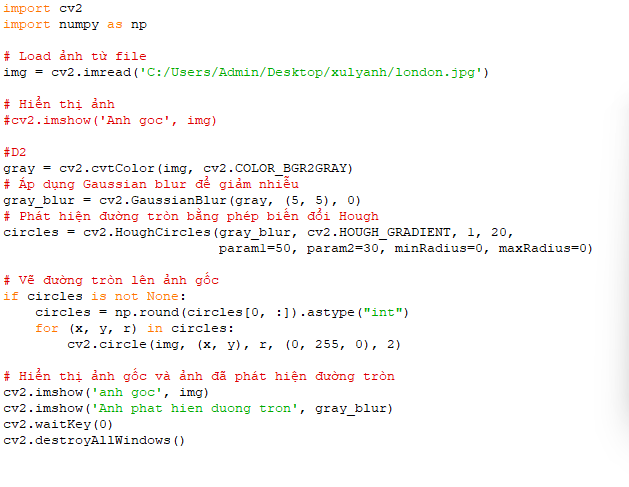
B1: Chuyển đổi ảnh sang ảnh xám (grayscale) và áp dụng bộ lọc Gaussian để giảm nhiễu.

B2: Sử dụng phép biến đổi Canny để phát hiện biên cạnh trên ảnh đã được xử lý.

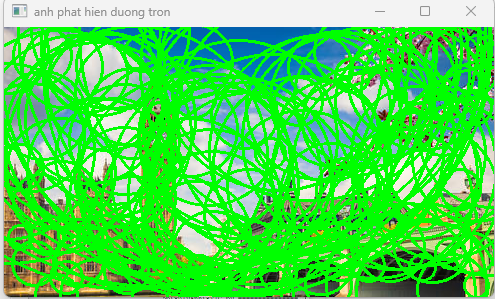
B3: Xác định các điểm biên cạnh trên ảnh, sau đó áp dụng phép biến đổi Hough trong không gian tham số của đường tròn để tìm các đường tròn trong ảnh.

B4: Áp dụng một ngưỡng trên giá trị tối đa của phép biến đổi Hough để loại bỏ các đường tròn không cần thiết.

B5: Vẽ các đường tròn tìm được trên ảnh gốc.



Đầu tiên đọc ảnh và chuyển đổi sang ảnh xám và áp dụng bộ lọc Gaussian để giảm nhiễu. Sau đó, chúng ta sử dụng phép biến đổi Canny để phát hiện biên cạnh trên ảnh đã được xử lý. Chúng ta sau đó áp dụng phép biến đổi Hough trong không gian tham số của đường tròn để tìm các đường tròn trong ảnh. Cuối cùng, chúng ta vẽ các đường tròn tìm được trên ảnh gốc.



Ảnh sau khi phát hiện và vẽ các đường tròn

Câu 2:

Để thực hiện kỹ xảo tấm màn xanh trong video và ghép nó với phong cảnh thực tế khi em đang đi trên đại lộ Hollywood mà không sử dụng thư viện moviepy, em có thể làm theo các bước sau:

Tạo một video thực tế khi em đang đi trên đại lộ Hollywood. Video này sẽ được sử dụng làm nền cho kỹ xảo tấm màn xanh.

Tạo một video chứa nội dung cần thêm vào bằng kỹ xảo tấm màn xanh. Video này sẽ có phông nền màu xanh chroma key để có thể loại bỏ phông nền này bằng kỹ thuật tấm màn xanh.

Sử dụng thư viện OpenCV để tải video thực tế và video cần thêm vào.

Đọc từng frame của video cần thêm vào bằng hàm cv2.imread() của OpenCV.

Sử dụng hàm cv2.cvtColor() của OpenCV để chuyển đổi màu của frame đó sang không gian màu HSV.

Sử dụng hàm cv2.inRange() của OpenCV để tạo ra một binary mask với các điểm ảnh nằm trong phạm vi màu xanh chroma key.

Sử dụng hàm cv2.bitwise\_not() của OpenCV để đảo ngược giá trị của binary mask để lấy phần còn lại của frame (không nằm trong phạm vi màu xanh chroma key).

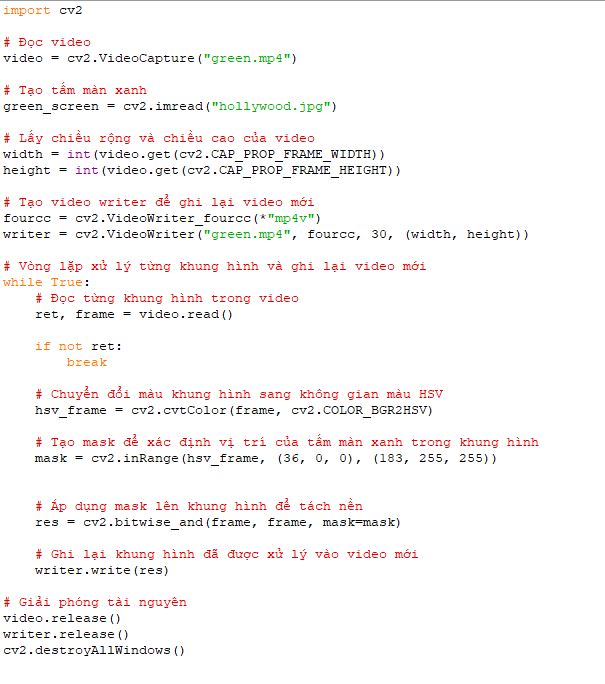
Sử dụng hàm cv2.resize() của OpenCV để điều chỉnh kích thước của frame cần thêm vào sao cho phù hợp với kích thước của video thực tế.

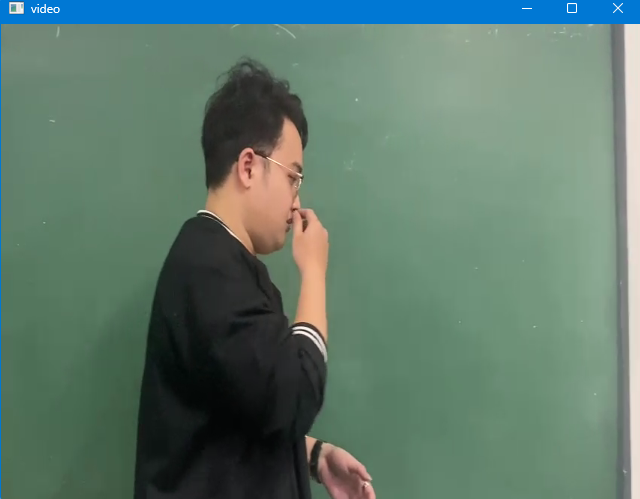
Sử dụng hàm cv2.bitwise\_and() của OpenCV để áp dụng binary mask lên frame cần thêm vào để loại bỏ phông nền màu xanh chroma key.

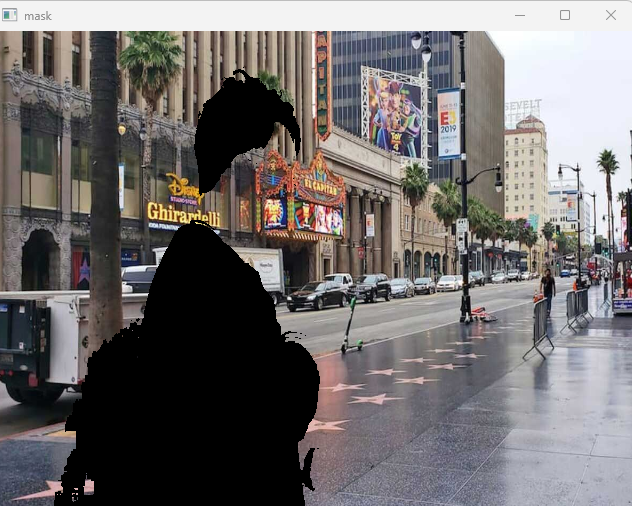
Sử dụng hàm cv2.bitwise\_and() của OpenCV để ghép frame đã loại bỏ phông nền với video thực tế.

Lặp lại các bước 4 đến 10 cho tất cả các frame của video cần thêm vào.

Sử dụng hàm cv2.VideoWriter() của OpenCV để ghi lại video đã hoàn thành.



video nền xanh

video xử lý đi trên đại lộ