**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

TRƯỜNG CƠ KHÍ

KHOA CƠ ĐIỆN TỬ

**A close up of a sign

Description automatically generated**

**BÁO CÁO PROJECT**

**XỬ LÝ ẢNH**

***ĐỀ TÀI:***

**ROAD LANE LINE DETECTION**

***GVHD*:** TS.Nguyễn Xuân Hạ

TS. Mạc Thị Thoa

***SINH VIÊN THỰC HIỆN:***

Đặng Xuân Khang 20184496

Trần Ngọc Hiệp 20184440

Ngô Minh Hải 20194996

**Hà Nội, 8/2022**

Mục lục

[**I. Introduction** 3](#_Toc110452885)

[**II. Algorithm** 5](#_Toc110452886)

[2.1. Pipeline 5](#_Toc110452887)

[2.2. Perspective Tranform 6](#_Toc110452888)

[2.3. Pre-precessing 7](#_Toc110452889)

[2.4. Sliding Window 9](#_Toc110452890)

[2.5. Fit using Polynomial Regression 10](#_Toc110452891)

[2.5.1. Linear Regression: y = a.x+ b 10](#_Toc110452892)

[2.5.2. Polynomial Regression 11](#_Toc110452893)

[**III. Tài liệu tham khảo** 12](#_Toc110452894)

[**IV. Phân Công nhiệm vụ** 13](#_Toc110452895)

[1. Trần Ngọc Hiệp 13](#_Toc110452896)

[2. Đặng Xuân Khang 13](#_Toc110452897)

[3. Ngô Minh Hải 13](#_Toc110452898)

# **I. Introduction**

Chủ đề an toàn giao thông luôn là một trong những điểm nóng trên thế giới bởi đây chính là tai nạn giao thông nằm trong top những nguyên nhân gây tử vong hàng đầu trên toàn cầu. Con số 1,35 triệu người chết mỗi năm công bố trong báo cáo dài 424 trang của WHO đã cho thấy sự nguy hiểm tiềm ẩn từ tai nạn giao thông lớn đến nhường nào. Do đó, c**ần phải nỗ lực đáng kể để thúc đẩy việc áp dụng các biện pháp tăng cường an toàn đường bộ.**



*Hình 1. Tai nạn giao thông đường bộ*

Trong thời đại công nghệ 4.0, việc tự động hóa các hoạt động của máy móc đang bùng nổ và được áp dụng vào hàng loạt những công việc khác nhau. Ngay trong lĩnh vực giao thông, cũng đã có vô số mô hình tự động nhằm đảm bảo tính thuận tiện và an toàn cho người tham gia. Những năm gần đây, **Advanced Driver Assistance Systems (ADAS)** đã nổi lên như một lĩnh vực nghiên cứu quan trọng độ an toàn cho xe tự lái, thu hút được sự quan tâm của nhiều nhà nghiên cứu và các tập đoàn lớn.

Theo nghiên cứu của Hiệp hội Bảo hiểm An toàn Đường bộ Hoa Kỳ, hệ thống tránh va chạm có thể giảm 27% tai nạn ô tô từ phía sau; hệ thống cảnh báo lệch làn đường có thể giảm 21% số vụ tai nạn thương vong, hệ thống phát hiện điểm mù có thể giảm 14% tai nạn va chạm trong làn đường. Ngoài ra còn có hệ thống phát hiện tài xế buồn ngủ, kiểm soát đổ đèo, hệ thống nhìn ban đêm, hệ thống hỗ trợ đỗ xe,...



*Hình 2. Hệ thống ADAS*

Bởi tính cấp thiết của lĩnh vực này, nhóm chúng em đã quyết định làm đề tài phát hiện làn đường Road Lane Line Dectection.

Ảnh có chứa văn bản

Mô tả được tạo tự động

*Hình 3. Road Lane Line Detection*

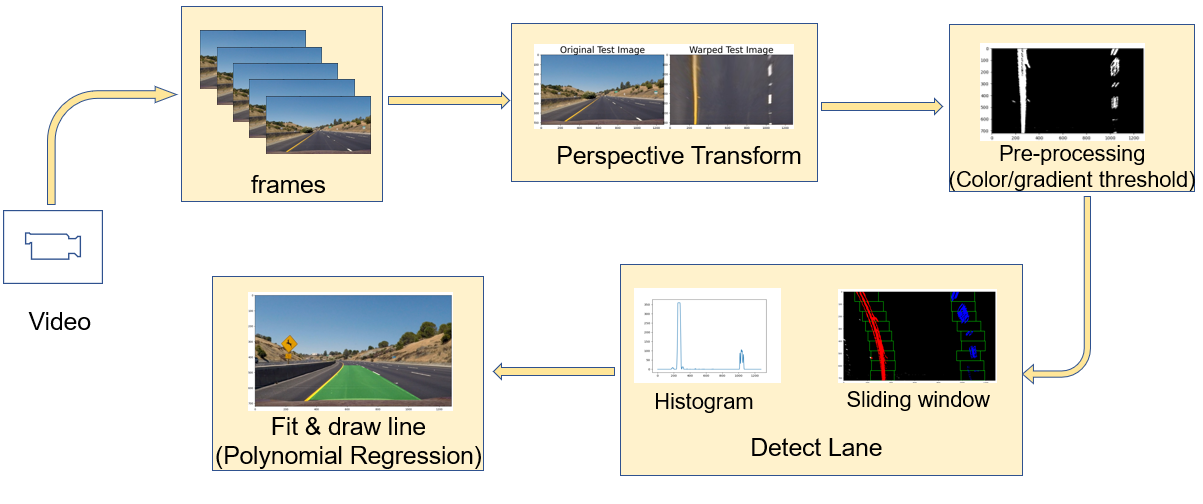
Đây cũng là một đề tài mà nhóm đành giá là rất thù vị và thực tế vì còn có thể được áp dụng tạo bản đồ cho robot tự hành (AGV) trong nhà máy hay thậm chí là giao bưu kiện tự động,…

*Hình 4. Robot AVG Hình 5. Robot giao hàng tự động*

# **II. Algorithm**

## 2.1. Pipeline



*Hình 6. Pipeline*

Các bước cụ thể được đưa ra như sau:

* B1: Từ video thu được trên camera, trích xuất từng khung hình ra thành hình ảnh riêng biệt, đơn lẻ.
* B2: Trên từng khung hình, thực hiện biến đổi góc nhìn (Perspective Transform) để thu được hình hảnh dưới dạng nhìn từ trên xuống (Bird-eye view).
* B3: Tiền xử lý ảnh (Xử lý màu & sử dụng Toán tử Sobel).
* B4: Phát hiện làn đường bằng Histogram và Sliding Window.
* B5: Cân chỉnh và vẽ hình ảnh làn đường về hình ảnh gốc ban đầu.
* B6: Lặp lại từ B1 với khung hình tiếp theo.

## 2.2. Perspective Tranform

Đầu tiên, video sẽ được cắt ra thành các frames, chúng ta sẽ xử lý từng frame một. Ta cần cài đặt kích thước ảnh, trích ra khung ảnh cần xử lý. Ở đây, hình ảnh từ camera có kích thước 1280 x 720 pixels. Khung ảnh cần xử lý được định vị bởi hình chữ nhật có các đỉnh với tọa độ:

* A (725 ; 455)
* B (555 ; 455)
* C (1280 ; 680)
* D (0 ; 680)

Text

Description automatically generated

Sau đó ta gán các tạo độ trên thành một khung ảnh mới hình chữ nhật.

Graphical user interface

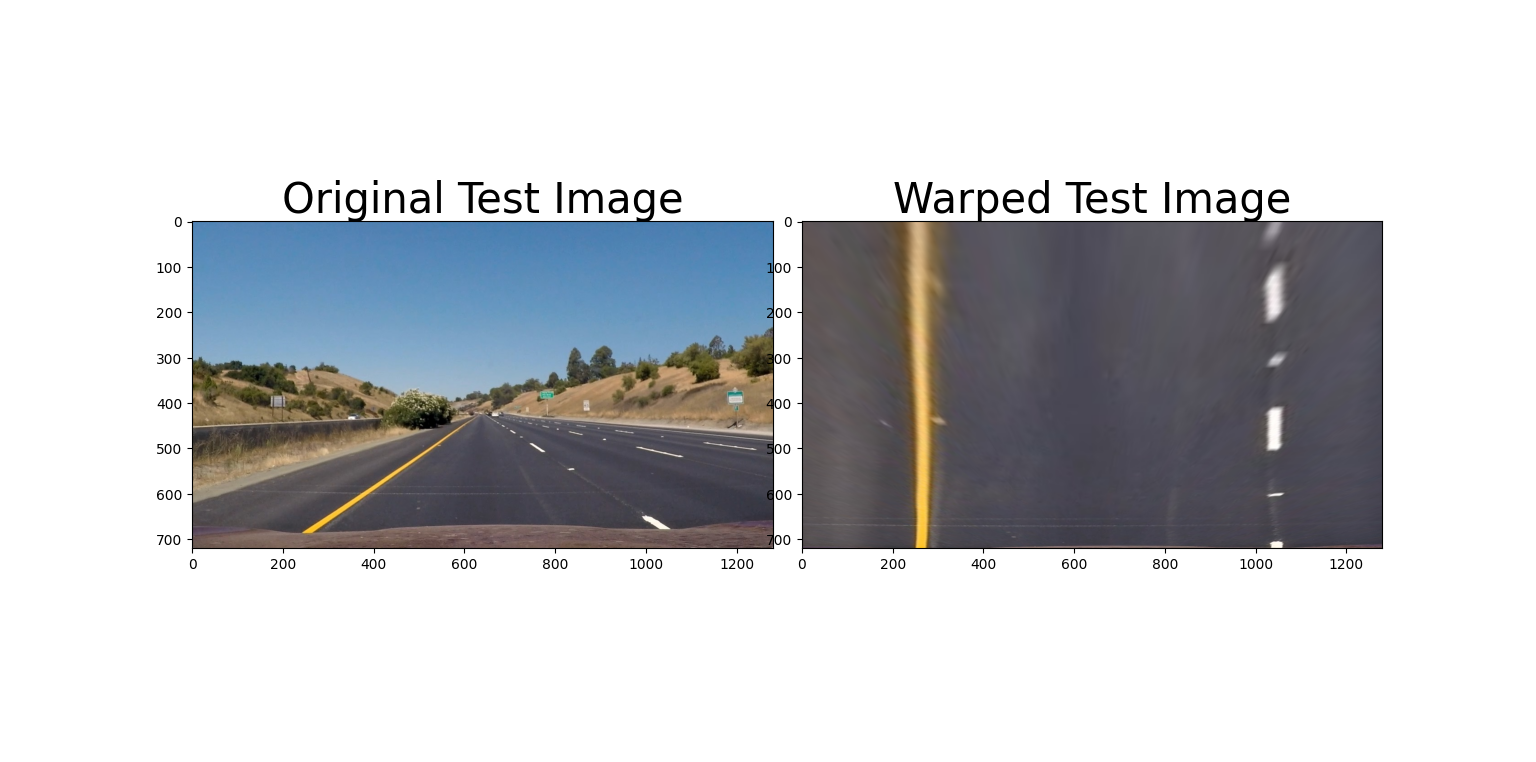
Description automatically generated with medium confidence

Sử dụng hàm **Warp** để đưa ảnh từ dạng hình thang ban đầu về hình chữ nhật, thu được ảnh mới dưới góc nhìn “from a bird’s eye perspective” bằng lệnh cv2.getPerspectiveTransform và cv2.warpPerspective trong thư viện OpenCV để dễ dàng sử dụng biểu đồ histogram ở phần sau cũng như dễ dàng tính độ cong của làn đường để dự đoán góc lái, …

*Text

Description automatically generated*

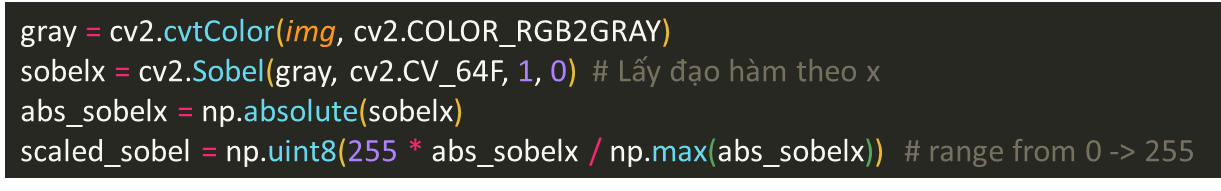
Kết quả sau khi thực hiện:



*Hình 7. Warped Image*

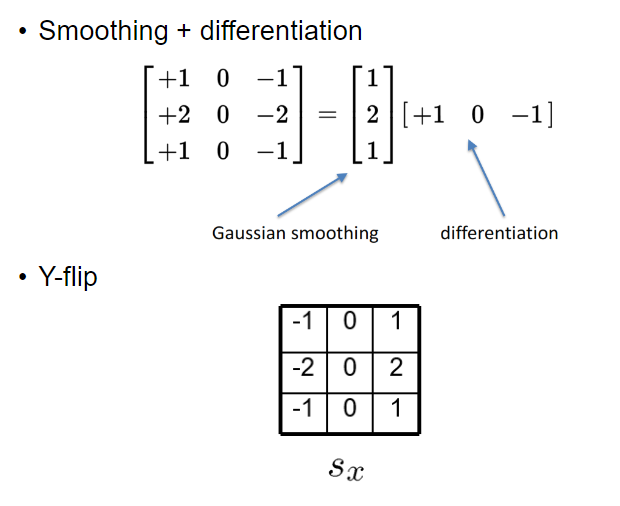
## 2.3. Pre-precessing

Sau đó đến bước tiền xử lý dữ liệu, chúng ta sẽ sử dụng thuật toán Sobel để phát hiện cạnh theo chiều ngang vì dấu hiệu nhận biết làn đường chỉ thay đổi mức độ sáng theo chiều ngang là chủ yếu, đồng thời ta đặt ngưỡng cho đạo hàm từ 20 đến 100.

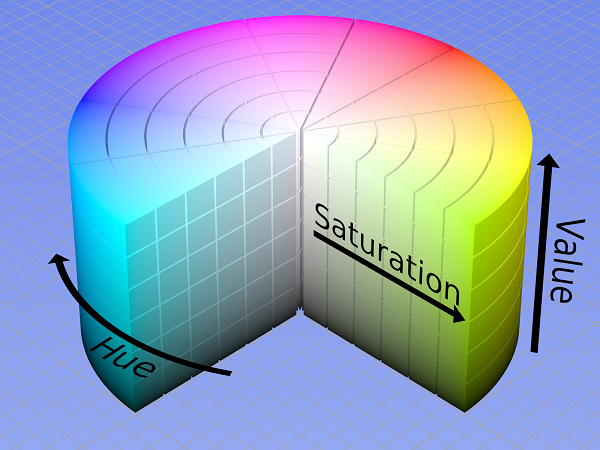




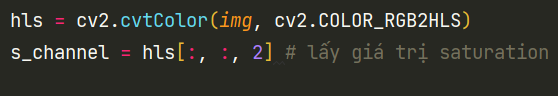
Toán tử Sobel:



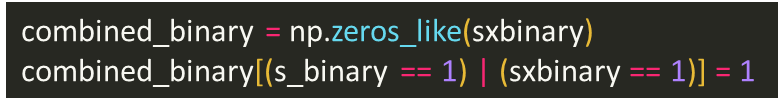
Cùng với đó, chúng ta sẽ phát hiện làn đường bằng cách lấy ngưỡng từ 170 tới 255 của các giá trị saturation trong thang màu HLS (hue, highlight, saturation) của ảnh vì vạch kẻ đường thường có màu đậm hơn mặt đường.

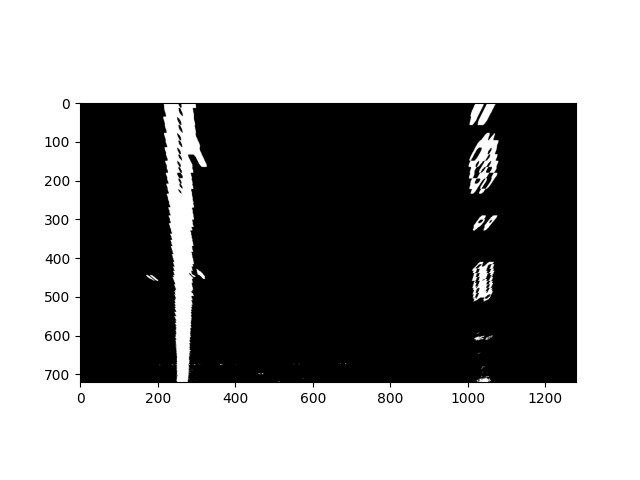


*Hình 8. Kênh màu HLS*



Và chúng ta sẽ kết hợp 2 kết quả đó lại với nhau, kết quả thu được như hình dưới đây:





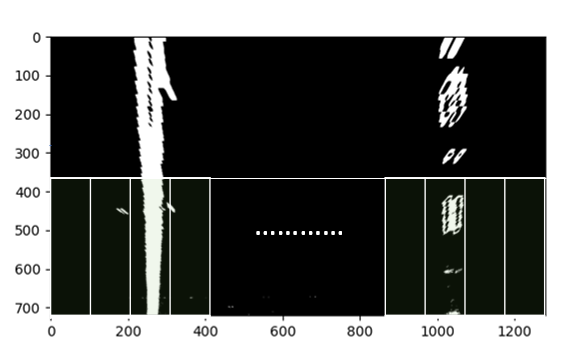
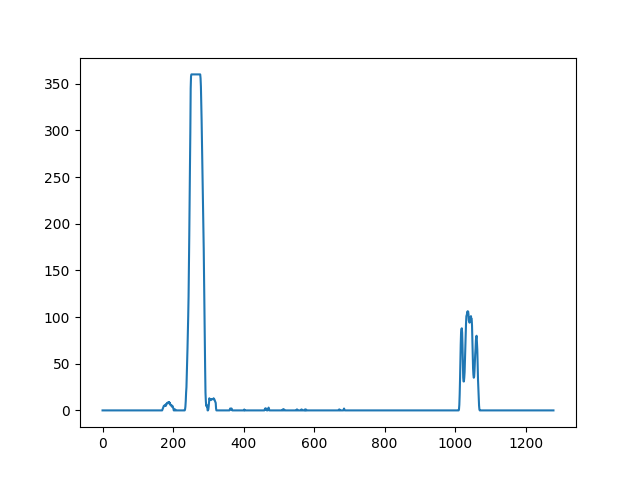
*Hình 9. Preprocessing Image*

## 2.4. Sliding Window

Và bây giờ chúng ta sẽ định vị vị trí làn đường từ ảnh kết quả của phần trước. Đầu tiên, chúng ta sẽ vẽ biểu đồ histogram mô tả tổng các giá trị pixel của các cột trong nửa dưới của ma trận ảnh sử dụng lệnh:

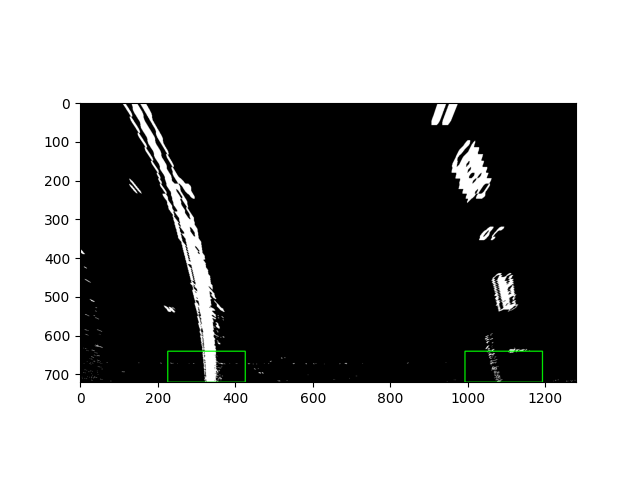
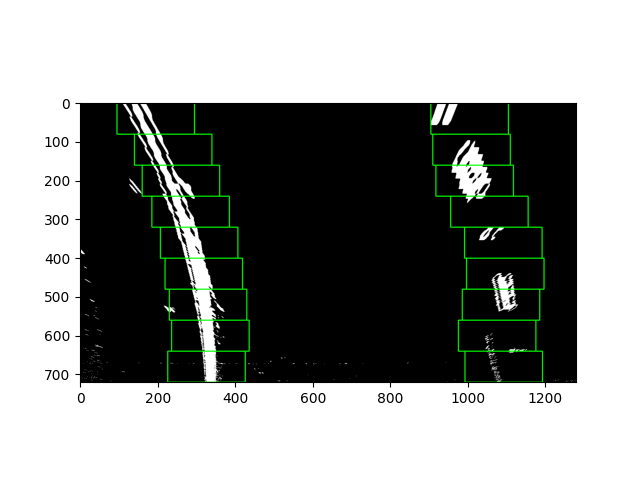


thu được kết quả như sau:

*Hình 10. Image Hình 11. Biểu đồ Histogram*

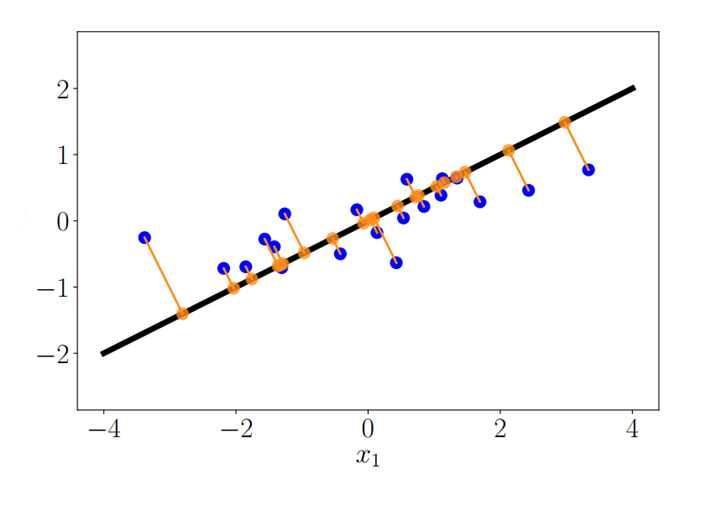
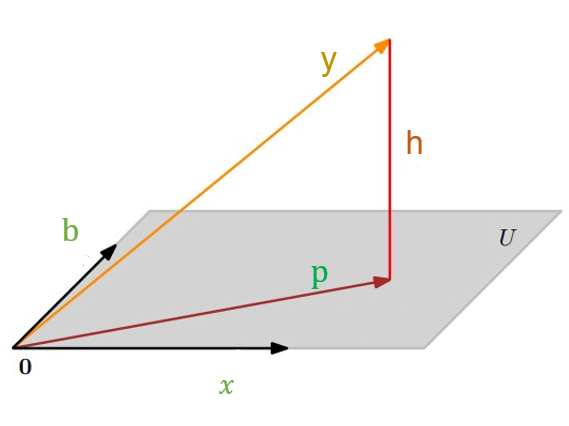
Ta thấy biểu đồ có 2 giá trị cực đại nhô lên hẳn so với những giá trị khác, đó chính là toạ độ x của vị trí bắt đầu của 2 vạch kẻ đường bên trái và bên phải, ta sẽ lấy ra 2 giá trị x đó. Tại mỗi toạ độ x, ta đặt 1 cửa sổ trượt kích thước 200\*80 với x là tâm cạnh chiều dài. Những vị trí có giá trị pixel khác 0 trong ô này chính là những vị trí xuất hiện vạch kẻ đường sẽ được lưu lại để vẽ làn đường trong bước tiếp theo. Sau đó, ta tính toạ độ trung tâm của các pixel khác 0 để lấy làm giá trị x cho ô tiếp theo. Cứ như vậy, ta sẽ thu được một list chứa tất cả các toạ độ vị trí của các pixel được cho là xuất hiện làn đường và ta sẽ vẽ một đường cong parapol đi qua các điểm đó sao cho tổng khoảng cách từ các điểm đó tới đường cong là nhỏ nhất bằng cách sử dụng thuật toán hồi quy đa thức (Polynomial Regression).

*Hình 12. Sliding window*

## 2.5. Fit using Polynomial Regression

### 2.5.1. Linear Regression: y = a.x+ b

*Hình 12+13. Linear Regression. “****Mathematics for Machine Learning, Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong****”.*

Để giải quyết bài toán hồi quy đa thức, trước hết ta giải quyết bài toán hồi quy tuyến tính. Cho tập hợp các điểm có toạ độ (x;y). Ta cần tìm phươmg trình đường thẳng y = a.x+ b sao cho tổng khoảng cách từ tất cả các điểm tới đường thẳng đó là nhỏ nhất. Ở hình bên phải, vector x là vector chứa tất cả các giá trị x, vector b là vector kích thước bằng độ dài của vector x với tất cả các giá trị bằng b. Vector x được co giãn a lần, vector b cũng được co giãn theo b. Vì vậy, tổng (ax+b) sẽ là tất cả các vector tâm O nằm trên mặt phẳng U chứa hai vector x và b với mọi (a, b). Vector y là vector chứa tất cả các giá trị y. Để tổng khoảng từ các điểm đến đường thẳng là nhỏ nhất thì ta cần tìm hai giá trị a, b sao cho vector p = (a.x + b) phải là hình chiếu của vector y lên mặt phẳng U.

Gọi A = , θ = . Vì h ⊥ (U) nên:

AT.h = 0

=> AT.(y-p) = 0

=> AT.y = AT.p

=> AT.y = AT.A.θ

=> θ = (ATA)-1.ATy

Vector θ chính là vector hệ số (a, b) cần tìm.

### 2.5.2. Polynomial Regression

Với phương trình bậc 2: y = a + bx1 + c thì A = , θ = thì công thức tính θ trên vẫn đúng.

Ở bài toán phát hiện làn đường, ta áp dụng thuật toán hồi quy đa thức để vẽ một parapol đi qua các điểm pixel được phát hiện là vạch kẻ đường thông qua câu lệnh np.polyfit(y, x, 2) của thư viện numpy. Sau đó ta vẽ làn đường độ đậm 0.3 lên frame ban đầu sẽ được kết quả như sau:

Ảnh có chứa văn bản, bầu trời, ngoài trời, con đường

Mô tả được tạo tự động

*Hình 14. Results Image*

Làm tương tự với các frame khác và imshow chúng ta sẽ được kết quả là một video phát hiện làn đường liên tục.

# **III. Tài liệu tham khảo**

[1] “Tìm hiểu về hệ thống hỗ trợ lái xe ADAS”, <https://vinfastauto.com/vn_vi/tim-hieu-ve-he-thong-ho-tro-lai-xe-adas>

[2] “**Advanced Lane Dectection**”, <https://kushalbkusram.medium.com/advanced-lane-detection-fd39572cfe91>

[3]**Mathematics for Machine Learning**, Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong*.*

[4] Prajakta R. Yelwande, Prof. Aditi Jahagirdar, “**Real-time Robust Lane Detection and Warning System using Hough Transform Method**”, International Journal of Engineering Research & Technology (IJERT), Vol. 8 Issue 08, August-2019.

# **IV. Phân Công nhiệm vụ**

1. Trần Ngọc Hiệp

Tìm hiểu về đề tài và thuật toán warped ảnh.

Làm slide và làm báo cáo phần đó.

1. Đặng Xuân Khang

Tìm hiểu về tiền xử lý ảnh, thang màu hls, gray, thuật toán sobel để phát hiện cạnh.

Làm slide và làm báo cáo phần đó.

1. Ngô Minh Hải

Tìm hiểu về cửa sổ trượt, thuật toán học có giám sát Linear Regression, Polynomial Regression.

Làm và tổng hợp slide, báo cáo.

Làm bản tóm tắt nội dung.