MỘT PHƯƠNG PHÁP PHÁT HIỆN ĐIỂM MÙ VỚI ĐỘ TIN CẬY CAO VÀ THỜI GIAN THỰC CHO CÁC HỆ THỐNG HỖ TRỢ LÁI XE THÔNG MINH

Vũ Hồng Sơn*, Đoàn Văn Tuấn, Nguyễn Tiến Dũng

Tóm tắt: *Các hệ thống hỗ trợ lái xe tiên tiến (Advanced Driver Assistance Systems: ADAS) đóng một vai trò quan trọng trong hệ thống an toàn chủ động chỉ có camera và các phương tiện tự động thông minh. Đối với các ứng dụng này, các yêu cầu về hiệu suất phát hiện đáng tin cậy và thời gian thực là các yếu tố cấp thiết. Bài báo này đề xuất giải pháp tối ưu tốc độ phát hiện ô tô và giảm các cảnh báo lỗi cho các hệ thống phát hiện điểm mù. Theo đó, trước tiên chúng tôi đề xuất bộ phân tầng Cascade – AdaBoost cùng với tập dữ liệu mẫu và thuật toán đào tạo của chúng tôi. Ngoài ra, để cải thiện tốc độ phát hiện, một kĩ thuật lựa chọn vùng quan tâm (Region of Interest: ROI) cũng được sử dụng để tránh trích xuất các vùng có khả năng tạo ra các cảnh báo lỗi như là bầu trời hoặc các vùng không phù hợp với phối cảnh. Phương pháp đề xuất đã tăng tốc độ phát hiện lên ít nhất 1,9 lần và giảm cảnh báo lỗi 2,24 lần so với phương pháp truyền thống ở các ảnh có độ phân giải cao (720 x 480) với tỷ lệ phát hiện đạt 99,4% và tỷ lệ cảnh báo lỗi nhỏ là 4,08%. Phương pháp đề xuất này có thể được ứng dụng cho các xe tự hành thông minh thời gian thực.*

Từ khóa: Hệ thống hỗ trợ lái xe tiên tiến (ADAS); Xe tự hành; Cascade – AdaBoost; Vùng quan tâm (ROI).

1. MỞ ĐẦU

Một trong những vấn đề xã hội lớn toàn cầu hiện nay là các vụ tai nạn giao thông đường bộ. Theo số liệu báo cáo của WHO, mỗi năm ước tính có khoảng 1,2 triệu người chết và 50 triệu người bị thương [1]. Có thể nhiều vụ tai nạn đã không xảy ra nếu các xe ô tô được trang bị hệ thống hỗ trợ lái xe tiên tiến (ADAS), ở đó, hệ thống ADAS sẽ hỗ trợ cho con người để có thể xử lý tốt hơn các tình huống liên quan tới tai nạn giao thông. Một cách tiếp cận khác đó là xu hướng chuyển dần từ các biện pháp an toàn thụ động như là sử dụng túi khí, dây bảo vệ và cấu trúc khung xe sang các biện pháp an toàn chủ động như kiểm soát ổn định hệ thống điện tử (Electronic Stability Control), hệ thống chống bó cứng phanh (Anti-lock Braking Systems), và các hệ thống hỗ trợ lái xe tiên tiến (ADAS) khác như tránh va chạm nhau, hỗ trợ giữ làn đường (Line Keeping Assistant). Các hệ thống hỗ trợ giảm thiểu tai nạn này đã trở nên phổ biến và được trang bị trên các xe ô tô tiên tiến vì người tiêu dùng đã quan tâm nhiều hơn đến sự an toàn.

Một hệ thống ADAS điển hình thu thập các thông tin liên quan từ các cảm biến sau:

- Cảm biến bên trong xe sẽ phát hiện và ứng phó các tình huống nguy hiểm bằng cách phân tích các hành vi của lái xe;

- Cảm biến bên ngoài xe (cảm biến siêu âm, rada và camera) sẽ cảnh báo sớm các tình huống nguy hiểm có thể xảy ra;

- Mạng cảm biến được sử dụng với nhiều cảm biến và được kết nối, đồng bộ với camera giao thông.

Hệ thống ADAS cung cấp các thông tin từ môi trường xung quanh chiếc xe để hỗ trợ lái xe, cũng như đưa ra các cảnh báo cho người lái xe nhằm tránh các tai nạn giao thông. Việc đồng bộ quá trình điều khiển xe với các thông tin từ hệ thống ADAS là cần thiết [2]. Trong hệ thống kiểm soát hành trình thích nghi (Adaptive Cruise Control) bao gồm 3 cảm biến rada, trong đó có 2 cảm biến rada gần để phát hiện các vật trên làn đường bên cạnh và một

cảm biến rada xa để phát hiện các vật trên đường. Các hệ thống hỗ trợ lái xe (Driver Assistant Systems) và bảo vệ xe chủ động (Active Safety Vehicles) đã được trang bị cho các xe hiện nay. Nếu mỗi chức năng của hệ thống ADAS cần bộ cảm biến và nền tảng xử lý riêng sẽ gây ra sự lãng phí lớn cũng như phức tạp khi lắp ráp. Để đơn giản và giảm chi phí, một hệ thống đa nhiệm được ra đời, nó có thể tích hợp nhiều chức năng xử lý của ADAS trong cùng một hệ thống xử lý [3]. Các thông tin về biên của làn đường là hết sức cần thiết cho các xe tự hành. Đã có nhiều nghiên cứu phát hiện biên của làn đường sử dụng các cảm biến khác nhau như cảm biến rada, camera. Tuy nhiên, rất ít các nghiên cứu giải quyết được vấn đề về biên làn đường tùy biến hoặc không có cấu trúc cụ thể [4]. Theo xu hướng thị trường hiện nay là tích hợp hai chức năng (Adaptive Cruise Control)và giữ làn đường (Lane Keeping Assist). Cấu trúc của bộ tích hợp giống như trường hợp dùng một camera: Khởi tạo dựa trên chức năng phân bố cạnh (Edge Distribution Function), vùng quan tâm (ROI) cho các lớp trái/phải và khoảng cách, rút trích các đặc trưng dựa trên bộ lọc có thể được mô hình hóa cho từng ROI cụ thể. Các kết quả chỉ ra rằng, kĩ thuật ROI thích nghi đơn giản có thể khắc phục tình trạng tắc nghẽn làn đường và xáo trộn các phương tiện lân cận [5]. Một cách tiếp cận khác là dùng hệ thống camera thị giác nổi. Cách tiếp cận này có thể phát hiện được vị trí, hướng và trạng thái chuyển động một cách đầy đủ như tốc độ, gia tốc từ đám mây điểm 3 chiều. Đám mây điểm này biểu diễn mô hình hóa vật 3 chiều và được tính bằng cách phân tíchcác chuỗi hình ảnh theo cả không gian và thời gian. Dùng thị giác nổi để nhận dạng bản đồ độ sâu của vật [7] và ước lượng tốc độ của vật [8]. Thông tin từ cảm biến rada dùng để phát hiện các đối tượng dựa trên hình ảnh ở khoảng cách xa. Phương pháp tiếp cận dựa trên hình ảnh nhanh có thể được thực hiện dựa trên các hệ thống nhúng kết hợp với sự phát triển của các thuật toán tiên tiến [10].

Bài báo này đề xuất giải pháp tối ưu tốc độ phát hiện ô tô và giảm các cảnh báo lỗi cho các hệ thống phát hiện điểm mù. Theo đó, trước tiên chúng tôi đề xuất bộ phân tầng Cascade – AdaBoost cùng với tập dữ liệu mẫu và thuật toán đào tạo của chúng tôi. Ngoài ra, để cải thiện tốc độ phát hiện, một kĩ thuật lựa chọn vùng quan tâm (ROI) là được sử dụng để tránh trích suất các vùng có khả năng tạo ra các cảnh báo lỗi như là bầu trời hoặc các vùng không gian không phù hợp với phối cảnh. Các tọa độ của các mẫu được phát hiện trong vùng ROI này, sau đó được chúng tôi áp dụng thêm một kĩ thuật tự động nhận dạng để kết luận mẫu được phát hiện có đúng là phương tiện đang di chuyển trên đường hay không, kĩ thuật này được gọi là đường cơ sở kiểm tra tọa độ của mẫu được phát hiện.

Phương pháp đề xuất đã tăng tốc độ phát hiện lên ít nhất 1,9 lần và giảm cảnh báo lỗi tới 2,24 lần so với phương pháp truyền thống ở các ảnh có độ phân giải cao (720x480) với tỉ lệ phát hiện đạt 99,4% và tỉ lệ phát hiện lỗi nhỏ là 4,08%. Phương pháp đề xuất này có thể được ứng dụng cho các xe tự hành thông minh thời gian thực.

2. CÁC NGHIÊN CỨU LIÊN QUAN

2.1. Thuật toán Adaboost

Adaboost (*Adaptive Boost*) là một cách tiếp cận *boosting* được *Freund* và *Schapire* [11] đưa ra vào năm 1995. Adaboost hoạt động trên nguyên tắc kết hợp tuyến tính các phân lớp yếu (*weak classifiers*) để tạo thành một bộ phân loại mạnh (*strong classifier*).

Là một cải tiến của tiếp cận *boosting*, Adaboost sử dụng thêm khái niệm trọng số (*weight*) để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi *weak classifier* được xây dựng, thuật toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng *weak classifier* kế tiếp: *tăng* trọng số của các mẫu bị nhận dạng *sai* và *giảm* trọng số của các mẫu được nhận dạng *đúng* bởi *weak classifier* vừa xây dựng. Bằng cách này, các *weak classifier* sau có thể tập trung vào các mẫu mà các *weak classifiers* trước nó

chưa làm tốt. Sau cùng, các *weak classifiers* sẽ được kết hợp tùy theo mức độ "tốt" của chúng để tạo dựng nên *strong classifier*.

Các *weak classifiers* $h_k(x)$ được biểu diễn như sau:

$$h_k(x) = \begin{cases} 1, p_k f_k(x) < p_k \theta_k \\ 0, otherwise \end{cases} \quad (1)$$

trong đó:

- $x = (x_1, x_2, ...x_n)$: Vector đặc trưng mẫu,
- θ : Ngưỡng,
- f_k : Hàm lượng giá vector đặc trưng mẫu,
- p_k : Hệ số quyết định chiều của bất phương trình.

Công thức trên có thể được diễn giải như sau: nếu giá trị vector đặc trưng của mẫu cho bởi hàm lượng giácủa bộ phân loại vượt qua một ngưỡng cho trước thì mẫu là *object* (đối tượng cần nhận dạng), ngược lại thì mẫu là *background*.

Thuật toán AdaBoost:

1. Cho một tập huấn luyện gồm *n* mẫu có đánh dấu $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ với $x_k \in X = (x_{k1}, x_{k2}, ...x_{kn})$ là vector đặc trưng và $y_k \in \{-1, 1\}$ là nhãn của mẫu.

2. Khởi tạo trọng số ban đầu cho các mẫu:

$$w_{1,k} = \frac{1}{n} \quad (2)$$

3. Xây dựng T weak classifiers:

Lặp $t = 1, ..., T$

Với mỗi đặc trưng trong vector đặc trưng, xây dựng một *weak classifier* h_j với ngưỡng θ_j và lỗi ε_j.

$$\varepsilon_j = \sum_k^n w_{t,k} \left| h_j(x_k) - y_k \right| \quad (3)$$

Chọn ra h_j với ε_j nhỏ nhất, ta được h_t.

$$h_t : X \to \{1, -1\} \quad (4)$$

Cập nhật lại trọng số

$$w_{t+1,k} = \frac{w_{t,k}}{Z_t} \times \begin{cases} e^{-\alpha_t}, h_t(x_k) = y_k \\ e^{\alpha_t}, h_t(x_k) \neq y_k \end{cases} \quad (5)$$

Trong đó:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_j}{\varepsilon_j} \right) \quad (6)$$

Z_t : Hệ số dùng để đưa $w_{t+1,k}$ về đoạn [0, 1].

4. *Strong classifier* xây dựng được:

$$H(x) = \text{dấu} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right) \quad (7)$$

Quá trình huấn luyện bộ phân loại được thực hiện bằng một vòng lặp mà ở mỗi bước

lặp, thuật toán sẽ chọn ra *weak classifier* h_t thực hiện việc phân loại với lỗi ε_t nhỏ nhất (do đó sẽ là bộ phân loại tốt nhất) để bổ sung vào *strong classifier*. Mỗi khi chọn được 1 bộ phân loại h_t, Adaboost sẽ tính giá trị α_t theo công thức ở trên. α_t cũng được chọn trên nguyên tắc làm giảm thiểu giá trị lỗi ε_t. Hệ số α_t nói lên mức độ quan trọng của h_t:

Trong công thức của bộ phân loại *H(x)*:

$$H(x) = \text{dấu}\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right) \tag{8}$$

Ta thấy, tất cả các bộ phân loại h_t đều có đóng góp vào kết quả phân loại của *H(x)*, và mức độ đóng góp của chúng phụ thuộc vào giá trị α_t tương ứng: h_t với α_t càng lớn thì nó càng có vai trò quan trọng trong *H(x)*.

Trong công thức tính α_t:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_j}{\varepsilon_j}\right) \tag{9}$$

Dễ thấy, giá trị α_t tỉ lệ nghịch với ε_j. Bởi vì h_t được chọn với tiêu chí đạt ε_t nhỏ nhất, do đó, nó sẽ đảm bảo giá trị α_t lớn nhất. Sau khi tính được giá trị α_t, Adaboost tiến hành cập nhật lại trọng số của các mẫu: tăng trọng số các mẫu mà h_t phân loại sai, giảm trọng số các mẫu mà h_t phân loại đúng. Bằng cách này, trọng số của mẫu phản ánh được mức độ khó nhận dạng của mẫu đó và h_{t+1} sẽ ưu tiên học cách phân loại những mẫu này.

Vòng lặp xây dựng *strong classifier* sẽ dừng lại sau *T* lần lặp. Trong thực tế, người ta ít sử dụng giá trị *T* vì không có công thức nào đảm bảo tính được giá trị *T* tối ưu cho quá trình huấn luyện. Thay vào đó, người ta sử dụng giá trị *max false positive* hay *max false alarm*. Tỉ lệ này của bộ phân loại cần xây dựng không được phépvượt quá giá trị này. Khi đó, qua các lần lặp, *false alarm* của *strong classifier* $H_t(x)$ xây dựng được sẽ giảm dần, và vòng lặp kết thúc khi tỉ lệ này thấp hơn *max false alarm*.

2.2. Vùng quan tâm (ROI)

Vùng quan tâm (*Region of Interest*, viết tắt ROI) là vùng ảnh ta sẽ áp dụng các xử lý/biến đổi trên đó và bỏ qua các vùng còn lại.

Không phải lúc nào chúng ta cũng cần xử lý trên toàn bộ không gian của ảnh. Thử tưởng tượng với một ảnh có kích thước 1024×1024 điểm ảnh, thì ta đã có hơn một triệu điểm ảnh cần phải xử lý. Do đó, để tiết kiệm chi phí tính toán, ta thu hẹp vùng quan tâm từ toàn ảnh về một vùng diện tích nhỏ hơn chỉ là một phần của ảnh gốc. Kĩ thuật này giúp giảm không gian tìm kiếm trên toàn bộ ảnh và giảm các cảnh báo lỗi ở các vùng không phù hợp với phối cảnh.

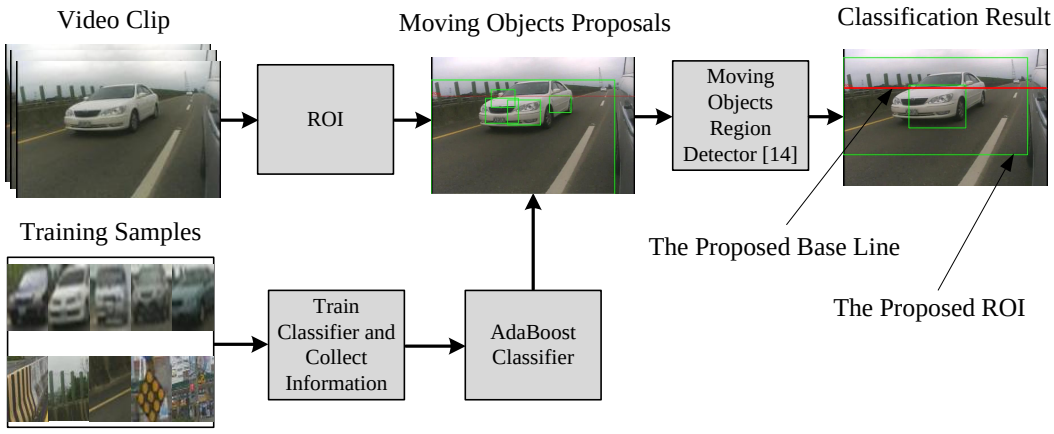
Áp dụng ROI lên ảnh ta lấy ảnh gốc nhân với mặt nạ ROI.

3. ĐỀ XUẤT THUẬT TOÁN

3.1. Mô tả thuật toán đề xuất

Chúng tôi đề xuất một giải pháp để xử lý thông tin nhanh trên ảnh và đáng tin cậy bằng cách kết hợp thuật toán Adaboot và vùng quan tâm (ROI). Mô hình đề xuất này có khả năng làm giảm thời gian xử lý thông tin trên toàn bộ ảnh. Ngoài ra, trong quá trình thực nghiệm chúng tôi phát hiện ra rằng, có rất nhiều các cảnh báo lỗi xuất hiện trên làn đường và ở các vùng không chứa đối tượng quan tâm; để giảm các cảnh báo lỗi này cho hệ thống,

một đường cơ sở được đề xuất bằng cách kẻ một đường ngang tại trung tâm của cảnh. Điều này giúp cho quá trình xử lý thông tin lọc ra được các vùng chứa các cảnh báo lỗi trên toàn bộ ảnh. Mô hình thuật toán được đề xuất trong bài báo là được mô tả chi tiết ở hình 1 và được tổng kết lại ở Algorithm1.



Hình 1. *Sơ đồ khối được đề xuất của hệ thống.*

Algorithm 1: The Proposed Algorithm
1. Video Clip
2. The Proposed ROI
3. Adaboost Classifier (Proposing the potential candidate regions)
4. Moving Objects Region Detector
4.1. Input: Edge detection
4.2. Find contours of the detected objects
4.3. Calculate *area*, *width* and *height* of bounding rectangles *box* for the detected objects
4.4. **for** (size_t i = 0; i < contours.size(); ++i) **do**
4.5. **if** ((*area*>*threshold1*) && (*box*.width >*threshold1*) && (*box*.height >*threshold1*)) **then**
4.6. **for** (size_t j = 0; j < contours.size(); ++j) **do**
4.7. **if** ((*area*>*threshold2*) && (*box*.width >*threshold2*) && (*box*.height >*threshold2*)) **then**
4.8. - Calculate the distance *d* from the defined location to neighboring locations
4.9. **if** (*d*<*threshold3*) **then**
4.10. - Cluster the neighboring locations round the defined location
4.11. **end if**
4.12. **end if**
4.13. **end for**
4.14. **end if**
4.15. **end for**
5. The Proposed Base Line (Removing the false alarms)
6. Classification Result

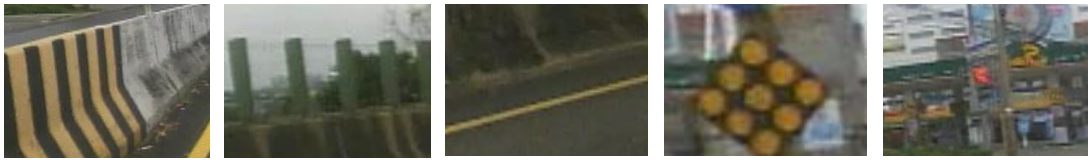
3.2. Tập huấn luyện được đề xuất

Ở các thí nghiệm của chúng tôi, tập dữ liệu đào tạo bao gồm hai tập mẫu dữ liệu, đó là tập mẫu dương (*positive sample*) và tập mẫu âm (*negative sample)*; tập mẫu dương chứa

các đối tượng cần được nhận dạng và tập mẫu âm không bao gồm các đối tượng quan tâm. Tập dữ liệu mẫu dương gồm có 2508 mẫu đào tạo, kích thước mỗi mẫu là 24x18-pixel; và tập dữ liệu mẫu âm gồm 4433 mẫu đào tạo. Một số mẫu dương và mẫu âm trong tập đào tạo là được minh họa lần lượt ở hình 2 và 3.

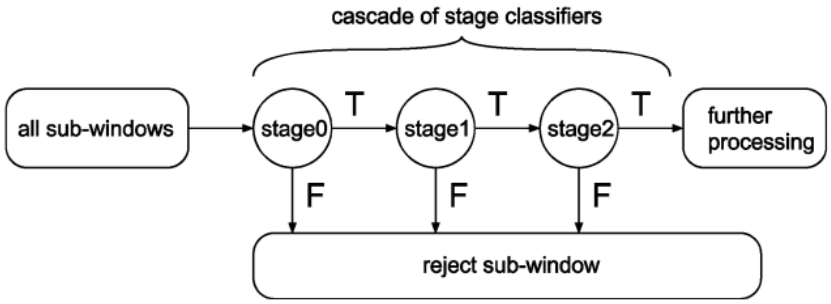


Hình 2. *Các mẫu đào tạo dương (positive sample).*



Hình 3. *Các mẫu đào tạo âm (negative sample)*

Trong quá trình huấn luyện, các bộ phân loại yếu (*weak classifiers*) sẽ học từ các mẫu dương trong tập huấn luyện và các mẫu âm là các vùng ảnh (*sub window*) trích ra từ các mẫu âm trong tập huấn luyện. Để đạt được các kết quả phát hiện nhanh, Viola và Jones đã đề xuất một cấu trúc phân tầng cho mỗi bộ phân loại, mỗi bộ phân loại này gồm có một chuỗi các bộ phân loại được phân cấp theo trọng số. Cấu trúc phân tầng này là được mô tả chi tiết ở hình 4. Hình 4 mô tả một cấu trúc bao gồm nhiều cấp bộ phân loại, mỗi cấp bộ phân loại được cấu thành bởi nhiều phân lớp yếu. Các cửa sổ đề xuất liên tiếp được định giá bởi các cấp bộ phân loại (để xác định cửa sổ này có chứa các đối tượng quan tâm hay không?) và quá trình xử lý tiếp theo chỉ được thực hiện khi giai đoạn trước đó trả về kết quả dương.



Hình 4. *Cấu trúc phân tầng các cấp bộ phân loại. Mỗi cấp bộ phân loại bao gồm nhiều phân lớp yếu. Nếu một cấp trả về kết quả sai, cửa sổ đề xuất sẽ bị từ chối, và quá trình xử lý sẽ dừng lại.*

Cấu trúc phân tầng này làm giảm thời gian xử lý đáng kể do bộ phân loại được đào tạo để từ chối nhanh các cửa sổ đề xuất không chứa đối tượng quan tâm; và để tập trung các tính toán, xử lý trên các cửa sổ đề xuất có khả năng chứa các đối tượng quan tâm. Bài báo này đề xuất mô hình đào tạo gồm 15 cấp cho bộ phân loại, mỗi cấp bao gồm các phân lớp yếu khác nhau, các thông tin chi tiết là được mô tả ở bảng 1.

Ngoài ra, trong quá trình khảo sát trên tập dữ liệu của chúng tôi, chúng tôi phát hiện ra rằng trong nhiều trường hợp các cảnh báo lỗi xuất hiện trên mặt đường. Vì vậy, chúng tôi đề xuất thêm một kỹ thuật lọc các cửa sổ được đề xuất căn cứ vào tọa độ của chúng. Kĩ thuật này được minh họa ở hình 1 bằng đường thẳng màu đỏ.

Bảng 1. *Số lượng các phân lớp yếu ở mỗi cấp của bộ phân loại.*

Số thứ tự các cấp (Stage number *n*)	0	1	2	3	4	5	6	7
Các phân lớp yếu (Weak classifiers)	5	8	12	16	18	23	29	30
Số thứ tự các cấp (Stage number *n*)	8	9	10	11	12	13	14	
Các phân lớp yếu (Weak classifiers)	24	23	25	19	14	10	6	

4. KẾT QUẢ THỰC NGHIỆM

Để đánh giá thuật toán đề xuất, chúng tôi tập hợp các tập dữ liệu có độ phân giải cao để đánh giá và giám sát. Tập dữ liệu bao gồm 16383 khung hình và có độ phân giải là 720x480 pixel. Các kết quả thực nghiệm được thực hiện trên hệ điều hành Windows 8-64bit, Intel Core i7-3770 CPU @ 3.40GHz, bộ nhớ 16 GB. Ngoài ra, chúng tôi định nghĩa ba tham số để đánh giá thuật toán được đề xuất: tỷ lệ phát hiện (DR:*Detection Rate*), tỷ lệ lỗi (MR: *Miss Rate*), tỷ lệ phát hiện lỗi (FDR: *False Detection Rate*). Phương trình các tham số được biểu diễn như sau:

$$DR = \frac{TP}{TPC} * 100\% \tag{10}$$

$$MR = 100\% - DP \tag{11}$$

$$FDR = \frac{FP}{TP + FP} * 100\% \tag{12}$$

Trong đó, *TPC (Total Pedestrion Collections)* là biểu diễn cho tổng số các mẫu xe ô tô, *TP(True Positives)* là các mẫu dương minh họa cho số lượng các mẫu xe ô tô được phát hiện, *FP(Fales Positives)* là các mẫu lỗi minh họa cho số lượng các mẫu không phải ô tô được phát hiện là ô tô.

Các kết quả thí nghiệmở bảng 2 chỉ ra rằng, cách tiếp cận của chúng tôi có thể tăng tốc lên ít nhất 1.9 lần khi so sánh với phương pháp truyền thống, với tỷ lệ cảnh báo lỗi giảm 5.05%. Các kết quả thí nghiệm là được minh họa chi tiết ở hình 5.

Bảng 2. *So sánh hiệu suất của thuật toán đề xuất và phương pháp truyền thống.*

Thuật toán	**Tổng số khung hình**	**Tỷ lệ phát hiện**	**Tỷ lệ lỗi**	**Tỷ lệ cảnh báo lỗi**	**Tốc độ xử lý**
Adaboost	16383	99.40%	0.6%	9.13%	29fps
Thuật toán đề xuất	16383	99.40%	0.6%	4.08%	55fps

5. KẾT LUẬN

Bài báo đề xuất một mô hình xử lý và phát hiện phương tiện đang di chuyển cho các điểm mù của hệ thống hỗ trợ lái tiến tiến (ADAS) ứng dụng cho xe ô tô thông minh dựa trên hình ảnh. Các thông tin xung quanh chiếc xe được thu nhận qua các camera hành trình

và được xử lý trên hệ thống nhúng với sự kết hợp của thuật toán Adaboost và các kĩ thuật được đề xuất trong đề tài. Các kết quả thí nghiệm chỉ ra rằng, cách tiếp cận trong bài báo có thể tăng tốc lên ít nhất 1.9 lần khi so sánh với phương pháp truyền thống, với tỷ lệ cảnh báo lỗi giảm 5.05%. Phương pháp đề xuất này có thể được ứng dụng cho các xe tự hành thông minh thời gian thực.

Lời cảm ơn: *Công trình này được hỗ trợ một phần bởi Bộ Giáo dục và Đào tạo, dưới mã đề tài: B2020-SKH-02.*

TÀI LIỆU THAM KHẢO

[1]. M. Lu, K. Wevers, and R. V. D. Heijden, *'Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety'*, Transportation Planning and Technology, vol. 28, no. 3, pp. 167–187, Jun. 2005.

[2]. K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, *'Three Decades of Driver Assistance Systems: Review and Future Perspectives'*, IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 4, pp. 6–22, 2014.

[3]. A. Ziebinski and S. Swierc, *'Soft Core Processor Generated Based on the Machine Code of the Application'*, Journal of Circuits, Systems and Computers, vol. 25, no. 04, p. 1650029, Apr. 2016.

[4]. J. Han, D. Kim, M. Lee, and M. Sunwoo, *'Road boundary detection and tracking for structured and unstructured roads using a 2D lidar sensor'*, International Journal of Automotive Technology, vol. 15, no. 4, pp. 611–623, Jun. 2014.

[5]. H. G. Jung, Y. H. Lee, H. J. Kang, and J. Kim, *'Sensor fusion-based lane detection for LKS+ACC system'*, International Journal of Automotive Technology, vol. 10, no. 2, pp. 219–228, Apr. 2009.

[6]. J. Arróspide and L. Salgado, *'On-road visual vehicle tracking using Markov chain Monte Carlo particle filtering with metropolis sampling'*, International Journal of Automotive Technology, vol. 13, no. 6, pp. 955–961, Oct. 2012.

[7]. S. Budzan and J. Kasprzyk, *'Fusion of 3D laser scanner and depth images for obstacle recognition in mobile applications'*, Optics and Lasers in Engineering, vol. 77, pp. 230–240, Feb. 2016.

[8]. M. Błachuta, R. Czyba, W. Janusz, and G. Szafrański, *'Data Fusion Algorithm for the Altitude and Vertical Speed Estimation of the VTOL Platform'*, Journal of Intelligent & Robotic Systems, vol. 74, no. 1–2, pp. 413–420, Apr. 2014.

[9]. A. Barth and U. Franke, *'Estimating the Driving State of Oncoming Vehicles From a Moving Platform Using Stereo Vision'*, IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 4, pp. 560–571, Dec. 2009.

[10]. D. Pamuła and A. Ziębiński, *'Securing video stream captured in real time'*, Przegląd Elektrotechniczny, vol. R. 86, nr 9, pp. 167–169, 2010.

[11]. Y. Freund, R. Schapire, *'A decision-theoretic generalization of on-line learning and an application to boosting'*, Journal of Computer and System Sciences, 55(1):119 – 139, 1997.

[12]. H.S. Vu, V.T. Doan, T.D. Nguyen, *'Real-Time Pedestrian Detection Using Motion Segmentation and Cascade-AdaBoost Classifier'*, Journal of Military Science and Technology, Special Issue, No.54A, pp. 10 – 19, 2018.

[13]. H.S. Vu, *'A high dynamic range imaging algorithm: implementation and evaluation'*, Science and Technology Development Journal, 22(3), pp. 293-307, 2019.

[14]. H.S. Vu, J.X. Guo, K.H. Chen, S.J. Hsieh, and D.S. Chen, *'A Real-Time Moving Objects Detection and Classification Approach for Static Cameras'*, Proc. IEEE Int. Conf. Consumer Electronics-Taiwan, pp. 255 – 256, May 2016. Nantou, Taiwan.

ABSTRACT

A REAL-TIME AND RELIABLE BLIND SPOT DETECTION METHOD FOR INTELLIGENT DRIVER ASSISTANCE SYSTEMS

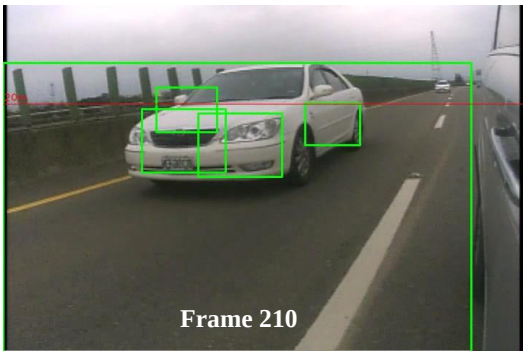
Advanced driver assistance systems (ADASs) play an important role in camera-only active safety systems and intelligent autonomous vehicles. For these applications, real-time and reliable detection performance is required. This paper aims to optimize the automobile detection speed and false alarm decrement for blind spot detection systems. Accordingly, we first propose the cascade-AdaBoost classifier along with our own sample datasets and training algorithm. In addition, in order to improve the detection speed, a regions of interests (ROIs) technique is also used to avoid extracting regions such as sky or regions of interests inconsistent with perspective, which generate the potential number of false alarms. The proposed method respectively achieves a speedup of at least 1.90x and 2.24x false alarm decrement compared to the conventional approach for high resolution images (720 x 480), with detection rate of 99.40% and a minor false detection rate of 4.08%. This makes the proposed method possible to be applied to real-time intelligent autonomous vehicles.

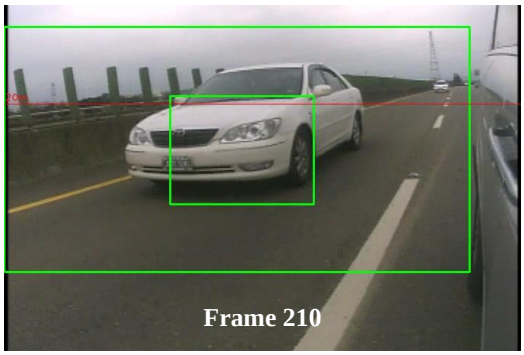
Keywords: ADAS; Autonomous; Adaboost; ROI.

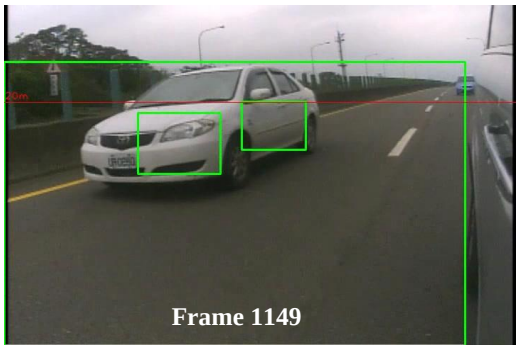
Nhận bài ngày 22 tháng 5 năm 2020
Hoàn thiện ngày 19 tháng 6 năm 2020
Chấp nhận đăng ngày 15 tháng 10 năm 2020

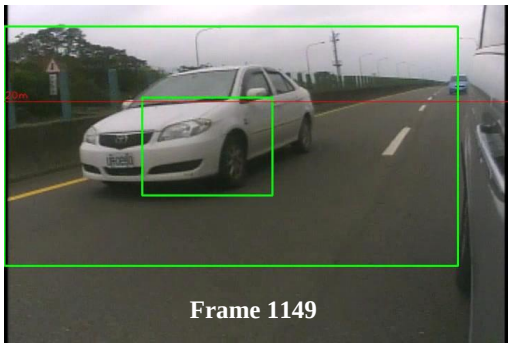
Địa chỉ: Trường Đại học Sư phạm Kỹ thuật Hưng Yên.
*Email: hongson.ute@gmail.com.

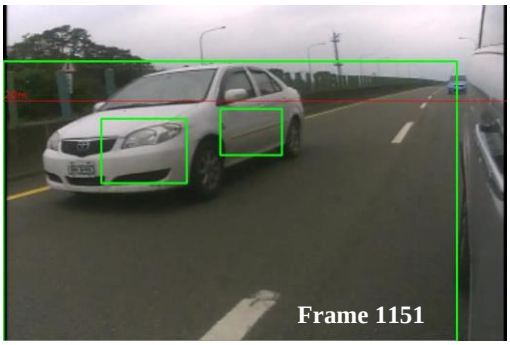
PHỤ LỤC

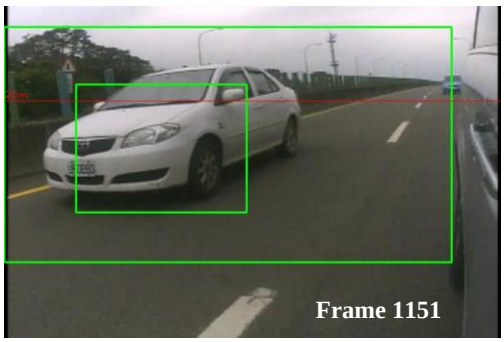

Frame 210


Frame 210


Frame 1149

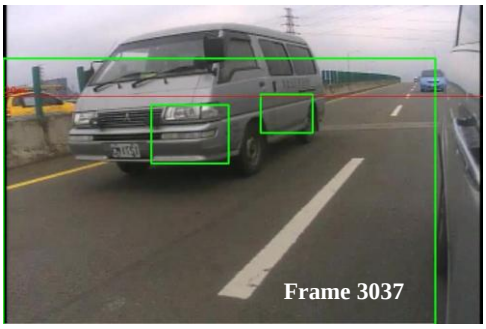

Frame 1149

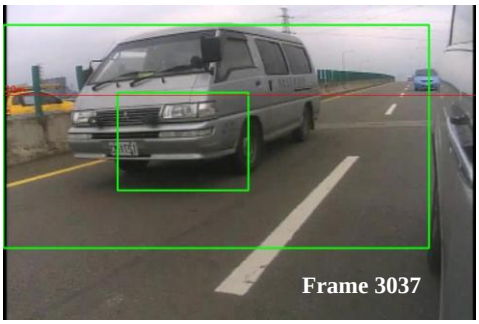

Frame 1151

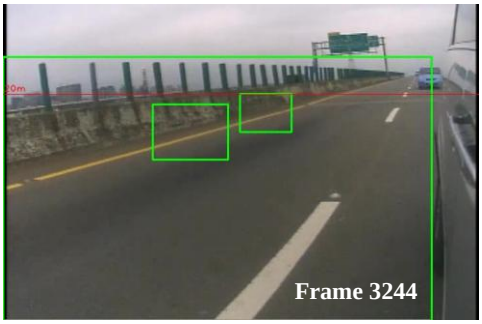

Frame 1151


Frame 3226

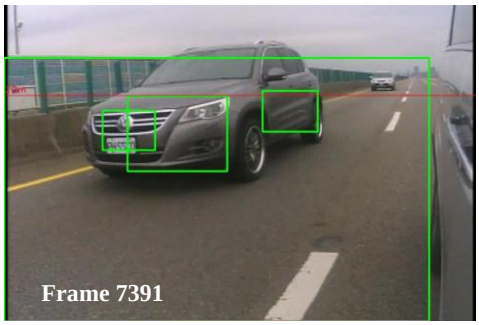

Frame 3226

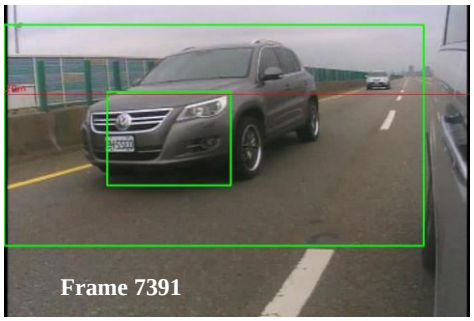

Frame 3037

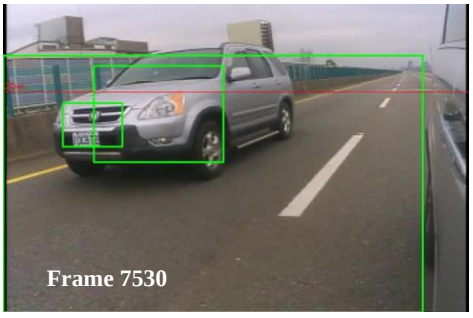

Frame 3037

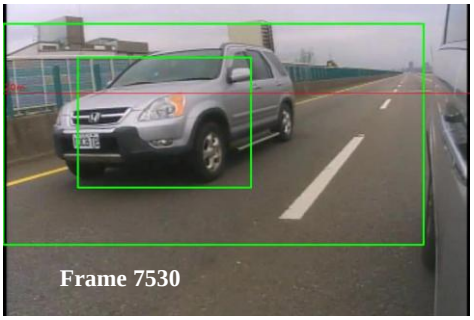

Frame 3244

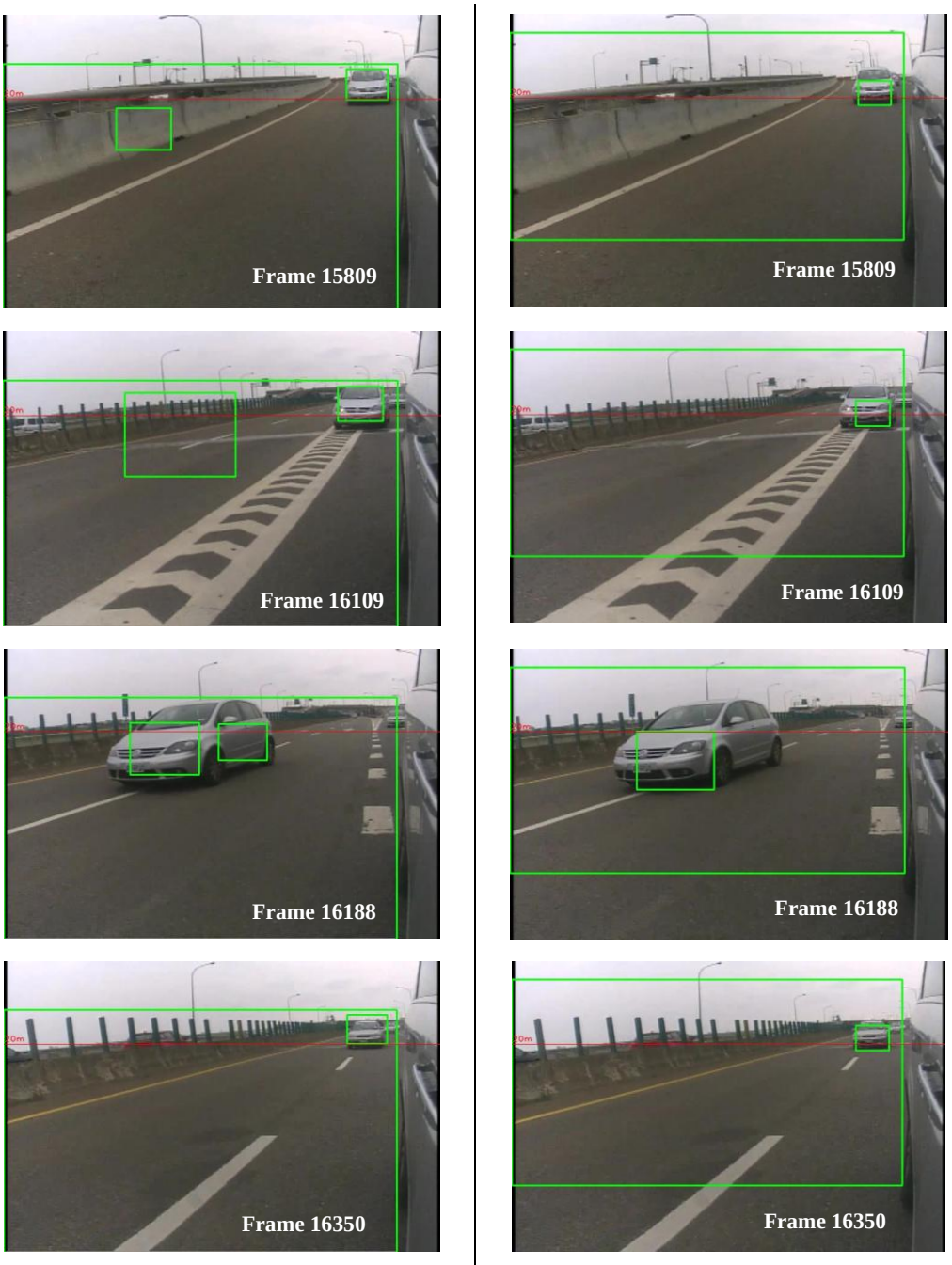

Frame 3244


Frame 7391


Frame 7391


Frame 7530


Frame 7530



Thuật toán truyền thống. *Thuật toán được đề xuất.*

Hình 5. *So sánh các kết quả thí nghiệm sử dụng thuật toán được đề xuất trong đề tài và thuật toán truyền thống.*