

ỨNG DỤNG XỬ LÝ ẢNH ĐỂ ĐẾM SỐ LƯỢNG XE Ô TÔ TRONG NỘI THÀNH ĐÀ NẴNG

Đoàn Phước Miền¹, Tống Thị Kim Quy², Trần Thế Vũ³, Huỳnh Hữu Hưng⁴

¹Khoa Kỹ thuật & Công nghệ, Trường Đại học Trà Vinh

²Sinh viên cao học, Trường Đại học Bách khoa, Đại học Đà Nẵng

³Viện Nghiên cứu và Đào tạo Việt Anh, Đại học Đà Nẵng

⁴Khoa Công nghệ Thông tin, Trường Đại học Bách khoa, Đại học Đà Nẵng

phuocmien@tvu.edu.vn, tongthikimquy@gmail.com, vu.tran@vnuk.edu.vn, hhhung@dut.udn.vn

TÓM TẮT: Tắc nghẽn giao thông là một vấn đề lớn cần giải quyết triệt để đối với tất cả các thành phố lớn trong đó có thành phố Đà Nẵng. Để điều phối được mật độ giao thông, trước hết cần phải biết được số lượng xe trung bình tham gia lưu thông trên các đoạn đường trọng tâm trong thành phố. Trong bài báo này chúng tôi đề cập đến việc xây dựng ứng dụng đếm số lượng xe ô tô trong nội thành Đà Nẵng. Bài báo tập trung nghiên cứu vào cách dò tìm đối tượng dựa trên phương pháp trừ nền kết hợp với dò tìm đám loang (blob). Bên cạnh bài báo còn thực nghiệm các phương pháp khác để so sánh kết quả một cách khách quan nhất. Các kết quả đạt được bao gồm: (1) So sánh hiệu năng của Phương pháp bộ phân mảnh tương thích mức điểm ảnh (PBAS) với hai thuật toán trừ nền cơ bản nhất là phương pháp khác biệt khung hình tĩnh và phương pháp khác biệt khung hình, (2) Đánh giá ảnh hưởng của các tham số trong thuật toán trừ nền đến quá trình nhận dạng, (3) Đánh giá hệ thống với các video đầu vào khác nhau.

Từ khóa: Phương pháp trừ nền, dò tìm Blob, khung hình tĩnh, khung hình liên tiếp, xử lý ảnh...

I. GIỚI THIỆU

Ngày nay, việc sử dụng xe cơ giới ngày một tăng làm cho tình trạng kẹt xe đang ngày càng nghiêm trọng. Việc tắc nghẽn giao thông đã làm cho chi phí của người tham gia tăng cao, tốn thời gian di chuyển. Ùn tắc giao thông cũng khiến cho người điều khiển phương tiện cảm thấy ức chế, mệt mỏi và phát sinh một số vấn đề vi phạm giao thông như: lấn làn, vượt đèn đỏ.

Theo thống kê của Ủy ban An toàn giao thông quốc gia cho biết trong 8 tháng đầu năm 2017 cả nước có gần 13000 vụ tai nạn giao thông đường bộ làm cho hơn 5000 người chết và gần 11000 người bị thương. Theo đó, bình quân mỗi năm ở Việt Nam có khoảng 8670 người chết được ghi nhận tại hiện trường. Nếu theo Bộ Y tế, năm 2015 có khoảng 15386 người chết, nghĩa là gần gấp đôi số người chết được ghi nhận tại hiện trường do cảnh sát giao thông thống kê [14]. Theo thống kê của Bộ Giao thông Vận tải, tính riêng tháng 12/2014, có đến 21.952 xe ô tô, 194.482 xe mô tô được đăng ký mới toàn quốc [17]. Do đó, việc nghiên cứu phương pháp phát hiện giao thông tự động là rất cần thiết để giúp tạo ra các luồng giao thông trơn tru, an toàn và hiệu quả, dẫn đến việc phân tích giao thông một cách đúng đắn, điều chỉnh việc quản lý cũng như phân phối các tín hiệu điều khiển phù hợp.

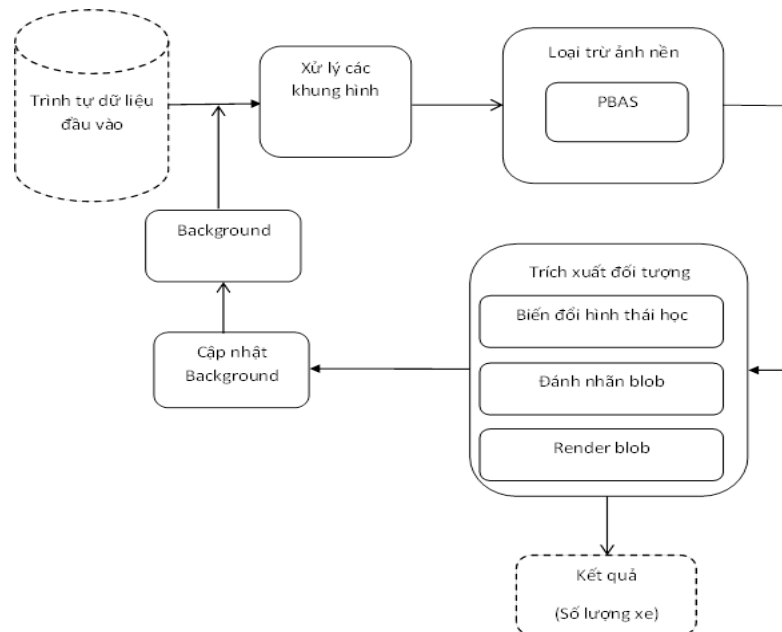
Các nhà nghiên cứu trong nước và trên thế giới hiện nay quan tâm rất nhiều đến hệ thống cảnh báo ùn tắc giao thông. Đã có rất nhiều phương pháp được đưa ra để giải quyết vấn đề ùn tắc thể hiện trong [1][2][3][5][8]. Tuy nhiên có hai phương pháp đang được quan tâm nhất đó là: dựa vào cảm biến tại chỗ [2] và sử dụng kỹ thuật nhận dạng hình ảnh để phân tích mật độ giao thông [3]. Phương pháp xử lý hình ảnh từ video và cách phát hiện đối tượng đã được nghiên cứu cho mục đích ước tính lưu lượng xe và lưu lượng giao thông [1]. Các phương pháp như phát hiện điểm, phát hiện cạnh được sử dụng trong quá trình dò tìm và theo dõi xe [7]. Có thể nói rằng một trong những nghiên cứu quan trọng nhất là phát hiện đối tượng trong ảnh [5] tạo tiền đề cho các hệ thống phát hiện đối tượng từ video [9]. Trong một số công trình như [10], phương pháp phân biệt hình ảnh phía trước và phía sau được sử dụng để trích xuất các phương tiện di chuyển trong chế độ định tuyến. Một số nghiên cứu như [11] và [12] đã chứng minh rằng việc sử dụng các vectơ đặc trưng từ vùng hình ảnh có thể cực kỳ hiệu quả cho các mục tiêu phát hiện xe. Một số nghiên cứu khác [13] trình bày ước tính kích thước phương tiện chính xác bằng cách sử dụng một tập hợp các hàm ánh xạ tọa độ. Hơn nữa, một số nhà nghiên cứu đã phát triển một loạt các thuật toán tăng cường để phát hiện đối tượng bằng cách sử dụng các phương pháp học máy có thể phát hiện và phân loại các đối tượng di chuyển theo cả loại và màu sắc như [15]. Trong [4] hiển thị tổng quan về các bước trừ nền: Bước đầu sẽ khởi tạo nền bằng N khung hình được thu nhận nền đầu tiên mà đối tượng đang đứng yên. Sau đó, phát hiện đối tượng chuyển động được thực hiện thông qua phát hiện foreground bao gồm phân loại các điểm ảnh như tiền cảnh hoặc nền bằng cách so sánh hình nền và khung hiện tại. Cuối cùng cần phải duy trì nền để cập nhật hình nền theo thời gian. Hai bước cuối sẽ được thực thi lặp lại liên tục trong suốt thời gian xử lý.

Mặc dù có rất nhiều nghiên cứu ngoài nước đã ứng dụng mạnh mẽ và vận hành thực tế, tuy nhiên không thể áp dụng y khuôn vào giao thông Việt Nam. Đặc thù giao thông ở Việt Nam rất phức tạp, như địa hình giao thông, phương tiện giao thông, văn hóa giao thông. Đặc biệt là phương tiện tham gia giao thông ở Việt Nam rất đa dạng như: Xe máy, xe đạp, xe đạp điện, xe ô tô 3 bánh, xe ô tô 4 bánh, xe tải, xe bò, xe công nông,... bên cạnh đó còn có người đi bộ, gia súc lưu thông trên đường. Bài báo này sẽ tập trung giải quyết vấn đề đếm số lượng xe ô tô lưu thông trên đường ở các điểm có camera trên địa bàn thành phố Đà Nẵng.

Bài báo bao gồm phần đầu sẽ giới thiệu về các công trình đã được nghiên cứu gồm những hệ thống đã và đang sử dụng, các phương pháp tiếp cận. Tiếp theo trong phần 2, chúng tôi sẽ trình bày chi tiết hệ thống đề xuất. Phần kết quả thực nghiệm sẽ được trình bày ngay sau đó và cuối cùng là phần kết luận.

II. HỆ THỐNG ĐỀ XUẤT

Dữ liệu đầu vào của hệ thống gồm các video thu được từ camera đặt ở góc đường trong nội thành Đà Nẵng. Sau khi có được dữ liệu này, hệ thống sẽ tiến hành xử lý để tách ra từng khung hình trên mỗi video. Mỗi khung hình sẽ được xử lý bởi phương pháp loại trừ hình nền nhằm biến đổi hình ảnh gốc (có màu sắc) thành mặt nạ nhị phân chỉ chứa hai loại điểm ảnh là đen và trắng. Trong đó, điểm ảnh đen tương ứng với background và điểm ảnh trắng tương ứng với foreground. Bước tiếp theo sử dụng hình ảnh thu được cùng với phương pháp nhận dạng đối tượng để khoanh vùng các đối tượng di chuyển. Ứng với chiều cao và chiều rộng tương ứng của các loại xe, ta có thể nhận dạng đối tượng có được có thỏa mãn yêu cầu hay không? Nếu có, tăng tổng số lượng xe đếm được và cuối cùng xuất tổng số xe ra màn hình. Mô hình đề xuất của chúng tôi như sau:



Hình 1. Mô hình hệ thống đề xuất

2.1. Tiền xử lý

Đầu vào là một video quay con đường với xe lưu thông, video này có thể là bất cứ loại file video ở định dạng nào. Ở bước này, ta tiến hành tách lấy từng khung hình từ video, biến đổi hình ảnh này thành một đối tượng hình ảnh số để làm đầu vào cho bước tiếp theo.

Chương trình đọc file video và tách thành từng frame, các frame này sẽ được xử lý và tạo thành đối tượng Mat để chứa các thông số tương ứng và là đầu vào cho bước tiếp theo. Mat về cơ bản là một lớp với hai phần dữ liệu: Ma trận tiêu đề (matrix header chứa các thông tin như kích cỡ ma trận, phương thức dùng để lưu trữ) và một con trỏ đến ma trận chứa giá trị điểm ảnh.

2.2. Loại trừ hình nền (Background Subtraction – PBAS)

Thuật toán tuân theo mô hình phi tham số, nghĩa là mỗi điểm ảnh x_i được mô hình hóa bằng một mảng các giá trị nền quan sát gần nhất[6]. Phương pháp này bao gồm các thành phần được mô tả như một máy trạng thái. Là một thành phần trung tâm, khối quyết định sẽ quyết định một điểm ảnh là foreground hay không dựa trên hình ảnh hiện tại và hình nền $B(x_i)$. Quyết định này dựa vào ngưỡng $R(x_i)$ của mỗi điểm ảnh. Hơn nữa, hình nền đã được cập nhật theo thời gian để cho phép thay đổi nền dần dần. Trong mô hình này, việc cập nhật dựa trên tham số học $T(x_i)$ của mỗi điểm ảnh. Ý tưởng chính của phương pháp này là hai ngưỡng của mỗi điểm ảnh tự động thay đổi dựa trên việc ước tính sự thay đổi của nền.

Mục đích của mỗi phương thức phân đoạn hình nền là để đưa ra được quyết định nhị phân, để xác định một điểm ảnh thuộc background hay foreground. Quá trình ra quyết định lấy hình ảnh đầu vào và so sánh nó một cách nào đó với hình nền. Trong trường hợp này, hình nền $B(x_i)$ được xác định bởi một mảng gồm N giá trị điểm ảnh được quan sát gần nhất.

$$B(x_i) = \{B_1(x_i), \dots, B_k(x_i), \dots, B_N(x_i)\} \quad (1)$$

Một điểm ảnh x_i được quyết định thuộc về background nếu giá trị điểm ảnh của nó $I(x_i)$ gần hơn một ngưỡng quyết định nhất định $R(x_i)$ đến ít nhất $\#_{min}$ của N giá trị background. Vì vậy, mặt nạ nhị phân được tính như sau:

$$F_{s,t} = \begin{cases} foreground & \text{nếu } \#(d(I_{s,t}, B_s(i)) < R_{s,t}) < \#_{min} \\ background & \text{cho các trường hợp còn lại} \end{cases} \quad (2)$$

Ở đây, $F = 1$ nghĩa là foreground. Do đó, có thể thấy rằng việc quyết định liên quan đến hai thông số: Khoảng cách ngưỡng $R(x_i)$ được xác định cho mỗi điểm ảnh riêng biệt và có thể thay đổi tự động. Số lượng tối thiểu $\#_{min}$ là một tham số cố định toàn cục.

Việc cập nhật hình nền B là cần thiết để tính toán những thay đổi trong background như: thay đổi ánh sáng, bóng và các hình nền di chuyển như cây cối ...

Vì khu vực foreground không thể được dùng cho việc cập nhật, các hình nền chỉ được cập nhật cho những điểm ảnh hiện tại đang là background (ví dụ $F(x_i) = 0$). Cập nhật có nghĩa rằng cho một số $k \in 1 \dots N$ (chọn một cách ngẫu nhiên), tương ứng giá trị hình nền $B_k(x_i)$ được thay thế bằng giá trị điểm ảnh hiện tại $I(x_i)$. Nó cho phép giá trị điểm ảnh hiện tại được "học" vào hình nền. Tuy nhiên, việc cập nhật này chỉ được thực hiện với xác suất $p = 1/T(x_i)$. Trái lại, sẽ không có cập nhật nào được thực hiện. Do đó, các tham số $T(x_i)$ định nghĩa tỉ lệ cập nhật. $T(x_i)$ càng cao, khả năng xảy ra cập nhật càng ít.

Chúng ta cũng cập nhật (với xác suất $p = 1/T(x_i)$) một điểm ảnh lân cận được chọn ngẫu nhiên $y_i \in N(x_i)$. Như vậy, hình nền $B_k(y_i)$ tại điểm ảnh lân cận này được thay thế bởi giá trị điểm ảnh hiện tại $V(y_i)$.

Trong một chuỗi video, có thể có khu vực với hình nền động cao (ví dụ: nước, cây trong gió ...) và các khu vực có ít hoặc thậm chí không có thay đổi (ví dụ: bức tường). Về mặt ý tưởng, đối với các khu vực có sự thay đổi cao, ngưỡng $R(x_i)$ nên được tăng để không bao gồm các đối tượng foreground. Trong khi đối với các vùng tĩnh, $R(x_i)$ nên là con số thấp bởi những sai lệch nhỏ như vậy dẫn đến quyết định cho foreground. Tóm lại, ngưỡng $R(x_i)$ cần phải có khả năng tự động điều chỉnh cho phù hợp. Để làm được điều này, cần phải có một thước đo độ động của background được thực hiện như sau:

Đầu tiên, bên cạnh việc lưu một mảng các giá trị điểm ảnh được quan sát gần nhất trong hình nền $B(x_i)$, ta cũng tạo một mảng $D(x_i) = \{D_1(x_i), \dots, D_N(x_i)\}$ của khoảng cách quyết định tối thiểu. Bất cứ khi nào một cập nhật của $B_k(x_i)$ được thực hiện, khoảng cách được quan sát tối thiểu hiện tại $d_{min}(x_i) = \min_k \text{dist}(I(x_i), B_k(x_i))$ được viết vào mảng này $D_k(x_i) \leftarrow d_{min}(x_i)$. Do đó, các lịch sử của khoảng cách quyết định nhỏ nhất được tạo ra. Trung bình của các giá trị này $\bar{d}_{min}(x_i) = \frac{1}{N} \sum_k D_k(x_i)$ là một thước đo của background động. Giả sử một background hoàn toàn tĩnh, $\bar{d}_{min}(x_i)$ sẽ bằng 0. Đối với các background động hơn, sẽ luôn có một độ lệch (nhỏ) của giá trị đang quan sát đối với những giá trị đã quan sát và vì thế, $\bar{d}_{min}(x_i)$ sẽ cao hơn. Với ước tính này của background động, ngưỡng quyết định có thể tự điều chỉnh như sau:

$$R(x_i) = \begin{cases} R(x_i) \cdot (1 - \frac{R_{inc}}{dec}), & \text{if } R(x_i) > \bar{d}_{min}(x_i) \cdot R_{scale} \\ R(x_i) \cdot (1 + \frac{R_{inc}}{dec}), & \text{else} \end{cases} \quad (3)$$

Trong đó, R_{inc}/dec , R_{scale} là những tham số cố định. Nó có thể được xem là bộ điều khiển động cho các biến trạng thái $R(x_i)$. Cho một bất biến $\bar{d}_{min}x_i$ không đổi, ngưỡng quyết định $R(x_i)$ tiếp cận kết quả của $\bar{d}_{min}x_i \cdot R_{scale}$. Vì vậy, một sự gia tăng trong background động (xảy ra đột ngột) dẫn đến sự tăng (chậm) của $R(x_i)$ đối với một ngưỡng quyết định $R(x_i)$ cao hơn.

Việc cập nhật tốc độ độc lập với các trạng thái foreground $F(x_i)$, cuối cùng, mỗi đối tượng sẽ được sáp nhập vào background tùy thuộc vào tham số "học" $T(x_i)$. Để giảm bớt các vấn đề, ý tưởng là đưa ra một bộ điều khiển động (thứ hai) cho $T(x_i)$. Như vậy, xác suất của việc "học" background tăng (chậm) khi một điểm ảnh là background và giảm (chậm) khi một điểm ảnh là foreground. Vấn đề ở đây là phân loại foreground sai thì sẽ được "học" từ từ vào background và vì vậy, vẫn còn foreground.

Có thể giả sử rằng các điểm ảnh phần lớn được phân loại sai là foreground trong các khu vực background động cao. Như vậy, sức mạnh của việc điều chỉnh trong bộ điều khiển có thể được điều chỉnh bằng cách sử dụng ước lượng động $\bar{d}_{min}x_i$. Ta định nghĩa:

$$T(x_i) = \begin{cases} T(x_i) + \frac{T_{inc}}{\bar{d}_{min}x_i}, & \text{if } F(x_i) = 1 \\ T(x_i) - \frac{T_{inc}}{\bar{d}_{min}x_i}, & \text{if } F(x_i) = 0 \end{cases} \quad (4)$$

Với T_{inc} , T_{dec} là các tham số cố định. Có các tham số khác nhau cho hai trường hợp, bởi vì ta giả sử rằng phần lớn thời gian, các điểm ảnh là background (đúng trong hầu hết các trường hợp). Lựa chọn các thông số độc lập cho background và foreground vì thế sẽ dẫn đến sự cân bằng của $T(x_i)$. Hơn nữa, ta định nghĩa cận trên và cận dưới

$T_{lower} < T < T_{upper}$, các giá trị này không thể vượt quá phạm vi ràng buộc. Bộ điều khiển trên đảm bảo rằng trong trường hợp background động cao (nghĩa là $\bar{d}_{min}x_i$ lớn), tham số "học" $T(x_i)$ vẫn không đổi hoặc chỉ thay đổi rất nhỏ. Trong trường hợp này, việc phát hiện foreground sai sẽ không tồn tại lâu, bởi vì xác suất để cập nhật $p = 1/T(x_i)$ không đạt được giá trị 0 rất nhanh. Trong một trường hợp lý tưởng khác của một background hoàn toàn tĩnh, bộ phân loại foreground là khá vững chắc, vì thế $T(x_i)$ tăng lên nhanh chóng, hình nền được kiểm định mà không cần nhiều cập nhật.

Những hình ảnh đầu vào $I(x_i)$ là một hình ảnh màu ba kênh (R, G, B). Ở đây, ta xử lý từng kênh độc lập và chạy thuật toán trong 3 luồng song song. Kết quả $F(x_i)$ là một bộ các toán tử của ba phân thức $F^R(x_i), F^G(x_i), F^B(x_i)$.

Đối với mỗi kênh màu, ngoài giá trị điểm ảnh, ta cũng sử dụng biên độ gradient. Vì vậy, đầu vào $I(x_i) = \{I^v(x_i), I^m(x_i)\}$ bao gồm giá trị điểm ảnh của chính nó $I^v(x_i)$ và biên độ gradient $I^m(x_i)$ tại điểm ảnh đó. Theo đó, mỗi phân tử của lịch sử background $B_k(x_i) = \{B_k^v(x_i), B_k^m(x_i)\}$ (ở phương trình (1)) cũng bao gồm hai phân tử ứng. Để tính khoảng cách trong phương trình (2), ta dùng phương trình sau:

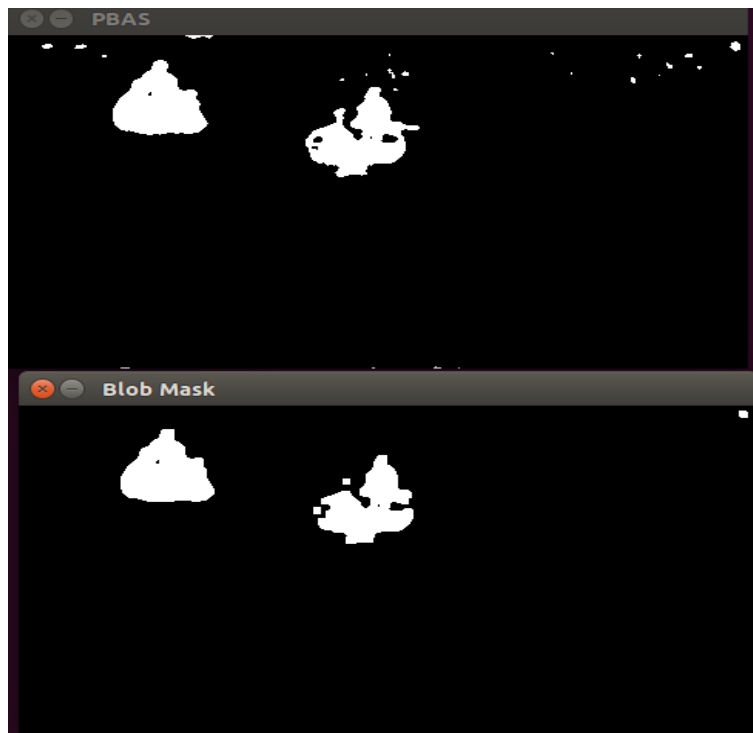
$$dist(I(x_i), B_k(x_i)) = \frac{\alpha}{\bar{I}^m} \cdot |I^m(x_i) - B_k^m(x_i)| + |I^v(x_i) - B_k^v(x_i)| \quad (5)$$

Trong đó, \bar{I}^m là trung bình độ lớn gradient trong khung hình quan sát cuối cùng. Vì vậy, $\frac{\alpha}{\bar{I}^m}$ chiếm tầm quan trọng của giá trị điểm ảnh so với cường độ gradient.

2.3. Trích xuất đối tượng

Kết quả của bước trừ nền là một mặt nạ nhị phân của khung hình hiện tại ảnh đầu vào. Mặt nạ nhị phân này sẽ tiếp tục được đưa vào khối BlobTracking để thực hiện tối ưu hóa bằng biến đổi hình thái học và sau đó phương pháp dò blob sẽ được áp dụng lên đầu ra của biến đổi hình thái học để có kết quả là các blob được đánh nhãn riêng biệt. Các phương tiện giao thông (ô tô, xe máy) sẽ được xác định dựa trên kích thước của chúng và được ghi nhận. Đầu ra cuối cùng của quá trình này là khung hình với các phương tiện giao thông đã được dựng lại cùng với kết quả đếm số lượng phương tiện tham gia giao thông.

Với cửa sổ PBAS là kết quả của quá trình trừ nền và Blob Mask là kết quả của quá trình biến đổi hình thái học, các điểm nhiễu đã bị loại bỏ đáng kể.



Hình 2. Mặt nạ nhị phân trừ nền và biến đổi hình thái học

Sau giai đoạn xử lý nhiễu ở bước biến đổi hình thái học, mặt nạ nhị phân mới *segmentated* sẽ tiếp tục được đưa vào bước xử lý chính và quan trọng nhất của quá trình trích xuất đối tượng là đánh nhãn để phân biệt các đối tượng và kết quả đầu ra sẽ là một mảng *blobs* bao gồm các blob trong khung hình hiện tại.

Với mỗi một blob được coi như một lớp đối tượng có các thuộc tính và phương thức. Do các phương tiện giao thông được lọc theo kích thước của chúng (có kích thước lớn hơn người, động vật) và để phục vụ render, có 5 thuộc

tính cần lưu ý là *area*, *minx*, *miny*, *maxx*, *maxy*. Trong đó *area* chính là số điểm ảnh biểu diễn cho đối tượng đó và các giá trị *min*, *max* là các giới hạn tọa độ của đối tượng xét theo mặt phẳng 2 chiều.

Một điểm ảnh P_n thuộc biên của đối tượng phải luôn có ít nhất 1 điểm lân cận là điểm ảnh đen, do vậy để tìm ra điểm biên tiếp theo của nó, có thể thực hiện quét các điểm lân cận của P_n bắt đầu từ điểm P_n bên cạnh điểm ảnh đen đã xác định trước theo chiều kim đồng hồ. Từ đó ta cũng có thể rút ra được một nguyên tắc, với P_n là điểm biên tiếp theo của P_{n-1} luôn có một điểm ảnh đen liền kề trước P_n theo vòng quét của điểm P_{n-1} . Như vậy, ta có 4 nguyên tắc sau để xác định một điểm đen liền kề với P_n như sau:

Nếu P_n là lân cận phải (R) hoặc dưới phải (DR) của P_{n-1} , luôn có một điểm ảnh đen liền kề là lân cận trên (U) của P_n ;

Nếu P_n là lân cận trái (L) hoặc trên trái (UL) của P_{n-1} , luôn có một điểm ảnh đen liền kề là lân cận dưới (D) của P_n ;

Nếu P_n là lân cận dưới (D) hoặc dưới trái (DL) của P_{n-1} , luôn có một điểm ảnh đen liền kề là lân cận phải (R) của P_n ;

Nếu P_n là lân cận trên (U) hoặc trên phải (UR) của P_{n-1} , luôn có một điểm ảnh đen liền kề là lân cận trái (L) của P_n .

2.4. Đếm xe

Để tránh đếm các blob tương ứng với đối tượng không mong muốn như người, động vật hoặc nhiễu thuộc background, ta cần xác định phạm vi kích thước của các phương tiện giao thông và áp dụng bộ lọc để loại bỏ các đối tượng không mong muốn đó. Thực chất, kích thước của một đối tượng được thể hiện qua giá trị *area* của một blob, với *area* chính là số điểm ảnh cấu thành nên blob đó. Sau khi có được tập các đối tượng mong muốn là các phương tiện giao thông, quá trình trích xuất và đếm số lượng sẽ được áp dụng để vẽ khung chữ nhật bao quanh đối tượng và hiện thị số lượng phương tiện đang xuất hiện trong khung hình.

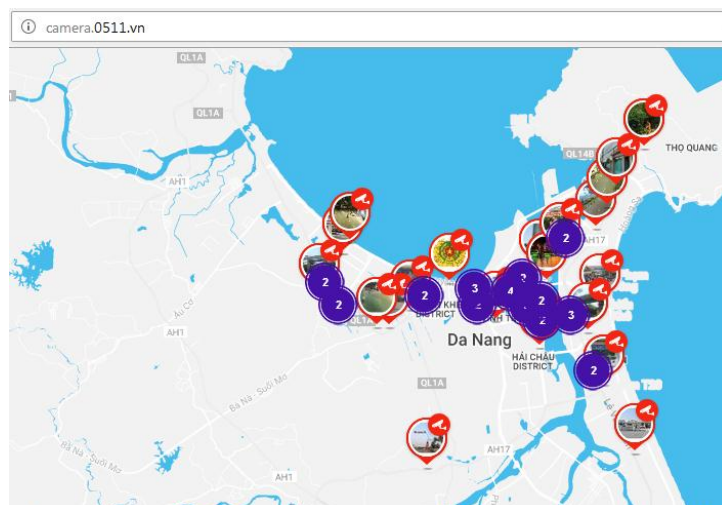
Tuy nhiên, như đã nêu trước đó, khuyết điểm của phương pháp này là khi một số đối tượng bị chồng lấn thì quá trình trích xuất đối tượng sẽ coi đó là 1 đối tượng dẫn đến nhận dạng và đếm không chính xác.

III. KẾT QUẢ THỰC NGHIỆM

Chúng tôi sẽ thực hiện bằng giải pháp đề xuất để nhận dạng xe ô tô. Sau đó, mở rộng các tham số để có thể đếm được các loại phương tiện giao thông khác. Đồng thời, tiến hành ba bài thực nghiệm để đánh giá hiệu năng của việc nhận dạng phương tiện giao thông dựa trên phương pháp trừ nền PBAS kết hợp với trích xuất đối tượng dùng kỹ thuật dò biên đã trình bày ở trên. Các bài thử nghiệm bao gồm: (1) So sánh hiệu năng của PBAS với hai thuật toán trừ nền cơ bản nhất là Static Frame Difference và Frame Difference, (2) Đánh giá ảnh hưởng của các tham số trong thuật toán trừ nền đến quá trình nhận dạng, (3) Đánh giá hệ thống với các video đầu vào khác nhau.

3.1. Nguồn dữ liệu Camera

Trong bài báo này, chúng tôi sẽ đề xuất xây dựng ứng dụng đếm số lượng xe sử dụng camera giao thông trong nội ô thành phố Đà Nẵng. Dữ liệu video trong bài báo chúng tôi đề cập từ nguồn camera công cộng tại địa chỉ <http://camera.0511.vn/> và được đọc trực tiếp theo thời gian thực.



Hình 3. Nguồn dữ liệu Camera

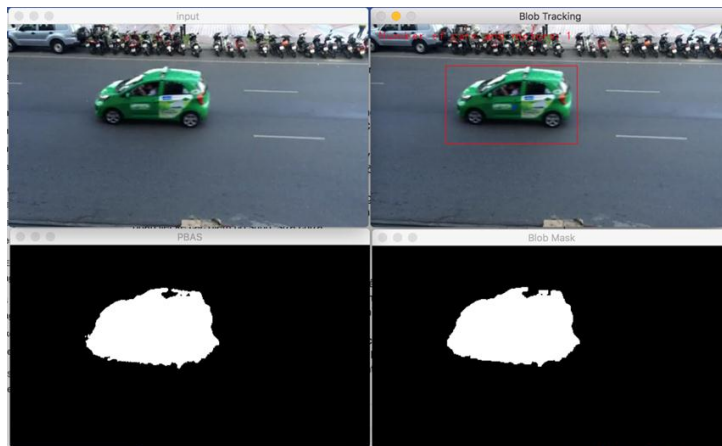
3.2. Môi trường và công cụ sử dụng

Chúng tôi sử dụng hệ điều hành Ubuntu phiên bản 14.04 sau đó nâng cấp lên phiên bản 16.04 để cài đặt các ứng dụng liên quan. Tốc độ Ram trên máy là 8GB, Bộ xử lý Corei5, Ổ cứng SSD 120GB.

Ngôn ngữ lập trình được sử dụng là Visual C++ kết hợp với thư viện Opencv3.0 và bộ nhận dạng phương tiện giao thông open source [16].

3.3. Thực nghiệm trên bài toán nhận dạng xe ô tô

Áp dụng các bước theo mô hình đề xuất, chúng tôi đã tiến hành thử nghiệm sử dụng đầu vào là một video. Kết quả chương trình đã nhận dạng được xe ô tô trong từng khung hình. Kết quả được thể hiện ở hình bên dưới.



Hình 4. Kết quả nhận dạng xe ô tô từ một video đầu vào

3.4. Thực hiện mở rộng nhận dạng các phương tiện giao thông khác

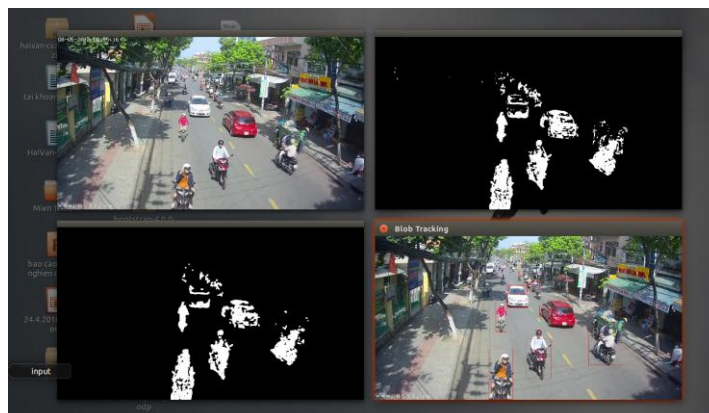
Chúng tôi thay đổi các thông số của kích thước khung hình mong muốn khi thực hiện bước lọc đối tượng (theo phương pháp Blob Tracking). Ứng với sự thay đổi đó, ta có thể dễ dàng giới hạn các đối tượng mà ta muốn nhận dạng. Hình dưới đây mô tả cụ thể các tham số nhỏ nhất (minArea) và lớn nhất (maxArea) của một đối tượng thỏa mãn yêu cầu tìm kiếm của ta. Theo đó, với minArea = 400, chương trình demo thực hiện nhận dạng các đối tượng tương ứng với kích thước của xe máy, xe đạp và xe ô tô. Trong khi đó, với minArea = 500, chương trình chỉ nhận dạng được xe ô tô.

```

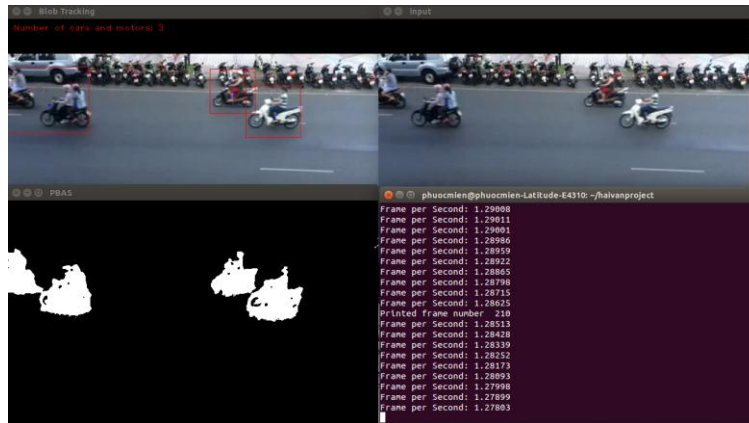
BlobTracking.xml
1  <?xml version="1.0"?>
2  <opencv_storage>
3  <!-- <minArea>5000</minArea> <!-- nhận dạng xe oto -->
4  <minArea>400</minArea> <!-- nhận dạng xe máy, xe đạp và xe oto -->
5  <maxArea>30000</maxArea>
6  <debugTrack>0</debugTrack>
7  <debugBlob>0</debugBlob>
8  <showBlobMask>1</showBlobMask>
9  <showOutput>1</showOutput>
10 </opencv_storage>

```

Hình 5. Các tham số cấu hình cho việc nhận dạng đối tượng blob



Hình 6. Kết quả nhận dạng mở rộng



Hình 7. Kết quả nhận dạng mở rộng xe mô tô 2 bánh

3.5. Đánh giá hiệu năng của các thành phần ảnh hưởng tới bài toán nhận dạng xe

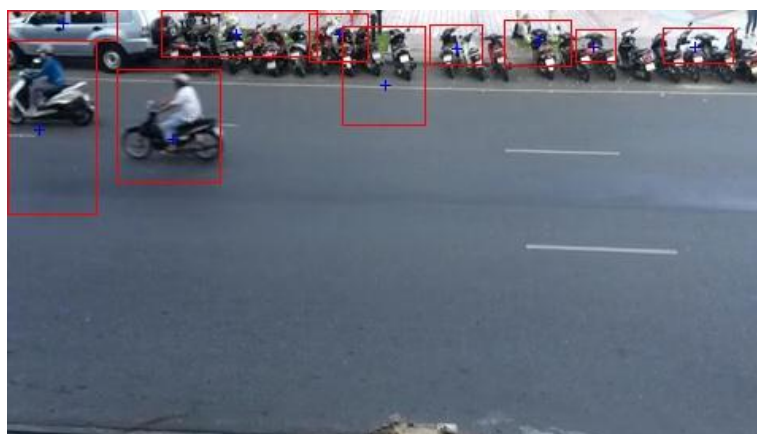
1. So sánh hiệu năng trừ nền của thuật toán PBAS với static frame difference và frame difference

Mục đích của bài thử nghiệm này chính là để so sánh hiệu năng của các thuật toán trừ nền PBAS với hai thuật toán trừ nền cơ bản nhất là Static Frame Difference và Frame Difference dựa trên thời gian xử lý, tỷ lệ nhận dạng thành công. Việc nhận dạng sẽ được thực hiện trên cùng một video đầu vào.

Bảng 1. So sánh hiệu năng của PBAS, Static Frame Difference và Frame Difference

Thuật toán	Frame per second (FPS)	Số đối tượng thực	Số blob tìm thấy	Số đối tượng/blob nhận dạng đúng	Tỷ lệ nhận đối tượng chính xác	Tỷ lệ nhận blob chính xác
Static Frame Difference	12.05	143	631	97	67,83 %	15,37 %
Frame Difference	14.5	143	125	93	65,03 %	67,88 %
PBAS	2.9	143	137	108	75,52 %	78,83 %

Như vậy có thể thấy do tính đơn giản về phương pháp xây dựng hình nền mà Static Frame Difference và Frame Difference có tốc độ xử lý nhanh hơn hẳn so với PBAS. Tuy nhiên, xét về độ chính xác của việc nhận dạng thì PBAS có ưu thế vượt trội. Lưu ý rằng, đặc biệt thuật toán Static Frame Different có số lượng blob tìm thấy sai lệch lớn so với số lượng đối tượng thực. Điều này là do khi điều kiện nền thay đổi (như độ sáng...) thì sẽ có sự sai khác so với nền tĩnh được chọn ban đầu dẫn đến các điểm ảnh thuộc nền ở các khung hình sau bị nhận dạng thành. Các trường hợp này còn gọi là positive sai.



Hình 8. Nhận dạng sai nền trong Static Frame Difference

Do đó khi đánh giá hiệu năng, ngoài tỷ lệ nhận đối tượng chính xác (số đối tượng nhận dạng được trên số đối tượng thực) ta cần quan tâm tới tỷ lệ nhận dạng blob chính xác với tỷ lệ này chính là số lượng blob đúng là đối tượng trên số lượng blob tìm thấy. Như ở thực nghiệm trên tuy Static Frame Difference có tỷ lệ nhận dạng đối tượng tốt hơn

Frame Difference nhưng khi đánh giá chung thì Frame Difference vẫn là thuật toán tốt hơn do số lượng false positive gần như không có.

2. Đánh giá ảnh hưởng của các tham số trong thuật toán trừ nền đến quá trình nhận dạng

Để đánh giá ảnh hưởng của các tham số, chúng tôi tiến hành lần lượt các thay đổi về giá trị N và giá trị ngưỡng R. Với giá trị N tương ứng 10, 20, 100. Kết quả cho thấy nếu giá N quá nhỏ thì tỷ lệ điểm hình nền bị xác định sai thành foreground là khá cao dẫn đến một số phần của nền bị nhận diện thành đối tượng. Còn nếu giá trị N quá cao thì ngược lại, các điểm ảnh thuộc foreground sẽ bị nhận thành nền dẫn đến tình trạng một đối tượng có thể bị nhận dạng thành hai đối tượng do bị cắt xén thành hai blob. Ngoài ra, N càng cao thì hệ thống sẽ mất nhiều thời gian xử lý hơn do một điểm ảnh sẽ phải có số lần so sánh với N nên nhiều hơn. Qua thực nghiệm, cho thấy khoảng giá trị $20 < N < 30$ là phù hợp với video test. Với ngưỡng R quyết định khoảng cách an toàn tối thiểu để một điểm ảnh không bị xác định là quá gần với đặc tính của nền. Thực nghiệm với các trường hợp $R=5, 18, 35$ cho thấy khi R quá nhỏ sẽ dẫn đến các điểm ảnh dễ dàng được coi là điểm ảnh foreground còn R quá lớn thì hàng loạt điểm ảnh thuộc đối tượng có màu sắc tương đối giống hình nền sẽ bị nhận thành nền. Kết quả này cũng phù hợp với lý thuyết khi ngưỡng R càng cao thì yêu cầu về khoảng cách giữa một điểm ảnh của khung hình $I(x_i)$ so $B_k(x_i)$ càng cao dẫn đến số lượng điểm ảnh không đáp ứng được yêu cầu đó càng nhiều và các điểm ảnh đó sẽ dễ bị đánh giá thành nền. Kết quả đánh giá cho thấy $15 < R < 20$ là hợp lý để đảm bảo mật độ phân không bị sai lệch quá nhiều so với ảnh gốc.

3. Đánh giá hệ thống sử dụng các video đầu vào khác nhau

Mục đích của bài thực nghiệm này là để đánh giá việc nhận dạng ở các góc camera khác nhau nên nền cũng sẽ khác nhau và cũng có một vài đối tượng chuyển động khác không phải là phương tiện ô tô (người đi chuyển trên vỉa hè) để đánh giá hiệu năng nhận dạng của hệ thống khách quan hơn.

Bảng 2. Đánh giá hệ thống với hai video đầu vào khác nhau

Video đầu vào	Frame per second (FPS)	Số đối tượng thực	Số blob tìm thấy	Số đối tượng/blob nhận dạng đúng	Tỷ lệ nhận đối tượng chính xác	Tỷ lệ nhận blob chính xác
v200048.ts	2.9	143	137	93	65,03 %	67,88 %
v200555.ts	3.1	47	44	38	80,85 %	86,36 %

Kết quả cho thấy hệ thống nhận dạng vẫn làm việc tương đối hiệu quả với video đầu vào v200555.ts. Ta sẽ thấy có một lưu ý là số blob tìm thấy lại nhỏ hơn đối tượng thực, điều này có thể giải thích do một số đối tượng có thể không được nhận dạng thành blob hoặc các đối tượng bị chồng lấn vào nhau dẫn đến bị trích xuất thành một blob.

IV. KẾT LUẬN

Bài báo đã đạt được một số kết quả khá quan trọng như: (1) Sử dụng các phương pháp nhận dạng như phân loại nổi tầng, trừ nền kết hợp với dò blob. (2) Hiện thị số lượng xe ở mỗi khung hình với độ chính xác cao bằng cách sử dụng phương pháp loại trừ hình nền và trích xuất đối tượng. (3) Đưa ra bảng đánh giá sự ảnh hưởng khác nhau của các yếu tố như thuật toán xử lý, tham số cấu hình, dữ liệu đầu vào ở các góc nhìn khác nhau. Mặc dù có một số kết quả như trên, tuy nhiên về tốc độ xử lý khi dùng thuật toán PBAS để trừ nền cần phải được cải thiện. Vấn đề các đối tượng chồng lấp vẫn chưa được giải quyết.

V. TÀI LIỆU THAM KHẢO

Tiếng anh:

- [1] T. R. Currin. "Turning Movement Counts. In Introduction to Traffic Engineering: A Manual for Data Collection and Analysis". Stamford Wadsworth Group, pp. 13–23, 2001 .
- [2] Subhashis Chaudhuri, Tom V. Mathew, Gopal R. Patil. "Real Time Traffic Counting System using video image processing". National Conference on Urban Mobility - Challenges, Solutions, and Prospects - IIT Madras, 2012.
- [3] K. Vidhya, A. BazilaBanu. "Density Based Traffic Signal System". International Journal of Innovative Research in Science, Engineering and Technology, 2014 International Conference on Innovations in Engineering and Technology (ICIET'14), 2014.
- [4] T. Bouwmans. "Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview". Science Direct journal on Computer Science Review, vol. 11 –12, pp. 31-66, may 2014.
- [5] Marco Treiber. "An Introduction to Object Recognition – Selected Algorithms for a Wide Variety of Applications". Siemens Electronics Assembly Systems GmbH & Co. KG Rupert-Mayer-Str. 44 81359 Munic, Germany, 2010.

- [6] Birgi Tamersoy. “Background Subtraction”. The University of Texas at Austin, September 2009.
- [7] Martin Hofmann, Philipp Tiefenbacher, Gerhard Rigoll. “The Pixel-Based Adaptive Segmenter”. Institute for Human-Machine Communication, 2009.
- [8] N. Chintalacheruvu, V. Muthukumar. “Video Based Vehicle Detection and Its Application in Intelligent Transportation Systems”. Journal of Transportation Technologies, Vol. 2, pp. 305-314, 2012.
- [9] P. G. Michalopoulos. “Vehicle Detection Video Through Image Processing: The Autoscope System”. IEEE Transactions on Vehicular Technology, Vol. 40, No. 1, 1991
- [10] Q. Cai, A. Mitiche, J. K. Aggarwal. “Tracking Human Motion in an Indoor Environment”. International Conference on Image Processing, USA, Vol. 1, pp. 215-218, 1995
- [11] D. G. Lowe. “Distinctive Image Features from Scaled-Invariant Keypoints”. International Journal of Computer Vision, pp. 91-110, 2004
- [12] D. A. Forsyth, J. Ponce. “Computer Vision: A Modern Approach”. Prentice Hall, 2003
- [13] A. H. S. Lai, G. S. K. Fung, N. H. C. Yung. “Vehicle Type Classification from Visual-Based Dimension Estimation”. IEEE Intelligent Transportation Systems Conference, pp. 201-206, 2001

Trang web:

- [14] Trang tuổi trẻ - Ủy ban An toàn giao thông quốc gia. “*Mỗi năm Việt Nam có 15.000 người chết vì tai nạn giao thông*”. <https://tuoitre.vn/moi-nam-vn-co-15000-nguoi-chet-vi-tai-nan-giao-thong-20170920154009747.htm>
- [15] Wikipedia - The Free Encyclopedia. “*Background Subtraction*”. https://en.wikipedia.org/wiki/Background_subtraction.
- [16] GitHub. “Andrews Sobral – Simple Vehicle Counting”. https://github.com/andrewssobral/simple_vehicle_counting.
- [17] Trang môi trường - Cổng thông tin điện tử Bộ Giao Thông Vận Tải, “*Năm 2014: Tai nạn giao thông giảm cả ba tiêu chí*”, <http://www.mt.gov.vn/moitruong/quy-chuan-chat-luong/35178/nam-2014--tai-nan-giao-thong-giam-ca-3-tieu-chi.aspx>.

TOWARDS VEHICLE COUNTING USING IMAGE PROCESSING: A CASE STUDY IN DA NANG CITY

Doan Phuoc Mien, Tong Thi Kim Quy, Tran The Vu, Huynh Huu Hung

ABSTRACT: *Traffic congestion is a major issue that needs to be addressed thoroughly for all major cities including Danang. In order to coordinate traffic flow, it is necessary first to know the average number of vehicles involved in the traffic in the city's main roads. In this paper we propose a methodology for vehicle counting. The paper focuses on vehicle detection based on the subtraction method associated with blob detection. Different methods are used to compare the results in the most objective way. The results are as follows: (1) Compare the performance of PBAS with two algorithms is Static Frame Difference and Frame Difference, (2) Evaluate the effect of parameters in the subtraction algorithm to the identification process, (3) Evaluate the system with different input videos.*

Keywords: *subtraction method, detection Blob, Static Frame Difference, Frame Difference, Image processing.*