



GIẢI THUẬT MỚI CHO BÀI TOÁN ĐỊNH VỊ VÀ NHẬN DẠNG BIỂN SỐ XE Ô TÔ

Trương Quốc Bảo¹ và Võ Văn Phúc²

¹ Khoa Công nghệ, Trường Đại học Cần Thơ

² Trung tâm đào tạo và dịch vụ tin học Nam Việt

Thông tin chung:

Ngày nhận: 16/01/2013

Ngày chấp nhận: 19/08/2013

Title:

A new algorithm for car license plate localization and recognition

Từ khóa:

Nhận dạng biển số, gán nhãn thành phần liên thông, phát hiện cạnh đỉnh, mạng Neron

Keywords:

License plate recognition, Connected component labeling, Vertical edge detection, Neural network

ABSTRACT

In this paper, we describe a new method to detect the car number plate. The proposed algorithms applied to locate the car license plate include connected-components labeling (CCLA), unwanted region elimination (UREA) and modified location license plate algorithms for improvement car license plate detection result. In addition, we also introduce a new neural network model for license plate recognition. Our proposed approach obtained better results for removing noise and locating characters in the plate if compared to the method applying vertical edge detection algorithm (VEDA). The promising experimental results demonstrated that our proposed method is efficient and stable enough for problem identification car license plate.

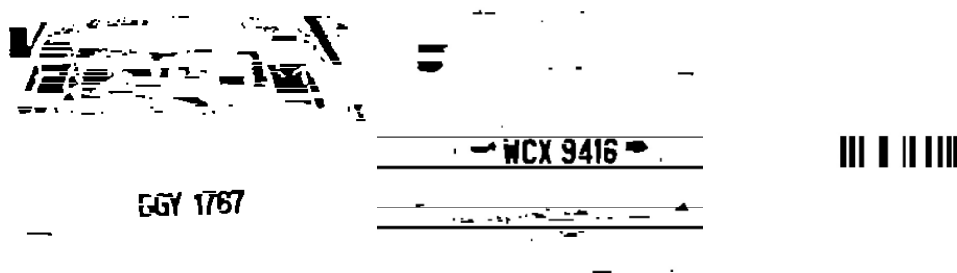
TÓM TẮT

Trong nghiên cứu này, chúng tôi đề xuất phương pháp mới giúp phát hiện biển số xe ô tô. Phương pháp được đề xuất bao gồm các giải thuật: giải thuật đánh nhãn cho các thành phần liên thông (CCLA), giải thuật loại bỏ vùng không mong muốn (UREA), giải thuật định vị biển số xe ô tô cải tiến. Ngoài ra, chúng tôi cũng đề xuất mô hình mạng nơ-ron cho việc nhận dạng các ký tự trên biển số. Kết quả đã loại nhiễu và định vị biển số xe tốt hơn so với phương pháp áp dụng giải thuật xác định cạnh đứng (VEDA). Những kết quả thực nghiệm chứng tỏ phương pháp chúng tôi đề xuất là hiệu quả và đủ ổn định cho bài toán nhận dạng biển số xe.

1 GIỚI THIỆU

Bài toán phát hiện và nhận dạng biển số xe ô tô đã có rất nhiều công trình nghiên cứu với các phương pháp tiếp cận khác nhau [1-2, 9-11]. Những nghiên cứu này chủ yếu xoay quanh vấn đề tìm cách tiếp cận và xây dựng mô hình thuật toán theo hướng giải quyết đơn lẻ các bài toán nhỏ và tổng hợp thành bài toán lớn. Trong đó, giai đoạn khử nhiễu để định vị biển số xe được xem là bước quan trọng nhất và hiện tại vẫn chưa tối ưu

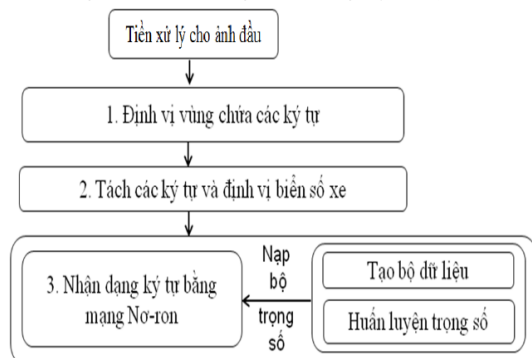
vì vùng ảnh tách biển số xe còn nhiều đối tượng nhiễu (Hình 1(a)), vùng biển số xe tách được cũng bị nhiễu (Hình 1(b), 1(c)) nên gây khó khăn cho việc tách ký tự [1-2, 9-10]. Để khắc phục nhược điểm này, chúng tôi đề xuất một phương pháp mới gồm sự kết hợp của các giải thuật: đánh nhãn cho các thành phần liên thông [3], loại bỏ vùng không mong muốn để xử lý loại nhiễu trước khi định vị biển số xe.



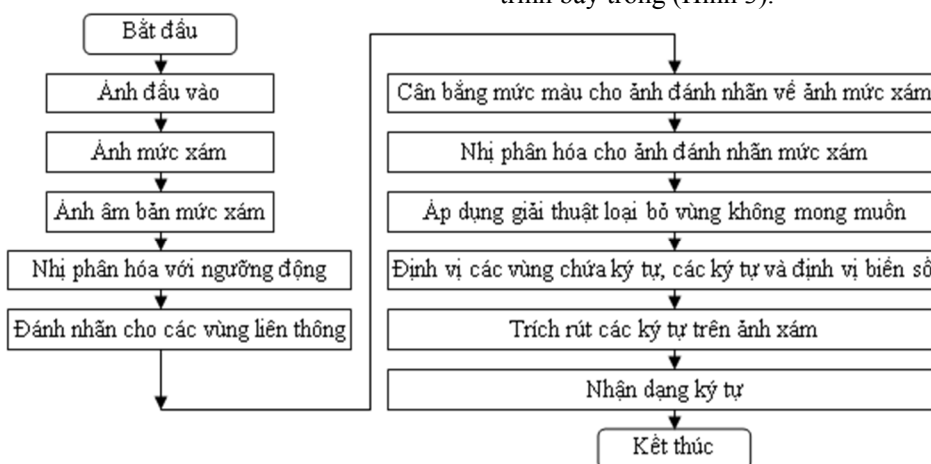
(a) Có nhiều đối tượng nhiễu (b) Nhiều vùng ký tự (c) Không tách được ký tự rõ ràng

Hình 1: Khử nhiễu, định vị biển số xe và tách ký tự bằng phương pháp phát hiện cạnh [1,2]

Bài toán định vị và nhận dạng biển số xe ô tô được chúng tôi chia làm ba bài toán con: định vị vùng chứa các ký tự từ ảnh đầu vào, tách các ký tự từ vùng ký tự và định vị biển số xe, rút trích bộ đặc trưng và nhận dạng cho từng ký tự (Hình 2).



Hình 2: Lược đồ tổng quát các bước thực hiện bài toán



Hình 3: Lược đồ chi tiết các bước thực hiện bài toán

Phần tiếp theo của bài viết trình bày quá trình định vị vùng chứa các ký tự từ ảnh đầu vào, phương pháp tách ký tự và định vị biển số xe

Giải thuật đánh nhãn cho các thành phần liên thông [3] sẽ được áp dụng trên ảnh nhị phân và đánh nhãn cho các vùng liên thông, sau đó sẽ loại bỏ những vùng không mong muốn ở bước tiếp theo.

Giải thuật định vị biển số xe ô tô cải tiến sẽ tìm các vùng ứng viên chứa các ký tự và các ký tự trên biển số xe. Biển số xe được xem là hợp lệ khi số ký tự ≥ 5 . Biển số xe được định vị bằng góc trái trên của ký tự đầu tiên và góc phải dưới của ký tự cuối cùng. Riêng biển số xe ô tô vuông được xem như một ký tự lớn có đặc điểm là có thể tách đôi ở khoảng giữa với một ngưỡng cố định.

Các ký tự được tách ra từ bản số xe sau đó sẽ được nhận dạng bằng 36 mạng Nơ-ron nhân tạo. Mỗi mạng được huấn luyện nhận dạng một ký tự. Lược đồ chi tiết cho bài toán của chúng tôi được trình bày trong (Hình 3).

được trình bày trong phần 3. Phần 4 đề xuất kỹ thuật rút trích đặc trưng và xây dựng mô hình nơ-ron cho quá trình nhận dạng ký tự của biển số

xe. Một số kết quả thực nghiệm và thảo luận được nêu trong mục 5. Kết luận và định hướng nghiên cứu tương lai được nêu tóm tắt trong phần 6.

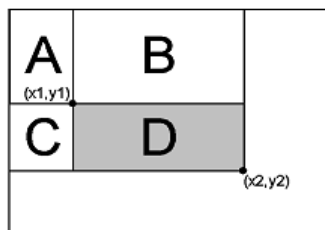
2 QUÁ TRÌNH ĐỊNH VỊ VÙNG CHỨA CÁC KÝ TỰ TỪ ẢNH ĐẦU VÀO

2.1 Tiền xử lý

Đầu tiên, các ảnh đầu vào sẽ được chuyển đổi thành ảnh mức xám. Ảnh xám sau đó được chuyển đổi thành ảnh âm bản để có biên số chữ trắng nền đen rồi đưa về ảnh nhị phân với ngưỡng động.

4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33



(a) Giá trị ảnh đầu vào, (b) Ảnh tích phân đã được tính, (c) Hình chữ nhật D cần tính tổng

Hình 4: Ảnh tích phân

Bước 3: Tính ngưỡng động qua tổng của D:

$$i'(x,y) = \begin{cases} 0 & \text{if } i(x,y) \cdot \text{count} \leq (\text{Sum} \cdot \frac{100-T}{100}) \\ 255 & \text{otherwise} \end{cases}$$

$$i'(x,y) = \begin{cases} 0 & \text{if } i(x,y) \cdot \text{count} \leq (\text{Sum} \cdot \frac{100-T}{100}) \\ 255 & \text{otherwise} \end{cases} \quad (3),$$

Trong đó: **i**: ảnh đầu vào; **i'**: ảnh nhị phân; **count**: số điểm ảnh trong hình chữ nhật D cần tính tổng.

Sum: là tổng của hình chữ nhật D tính theo công thức (2);

T: tham số cho hàm nhị phân hóa với ngưỡng động, thực nghiệm với $T=0.15$ [1-2].

2.2 Định vị vùng chứa các ký tự

Đây cũng được xem là bước định vị biên số xe và được thực hiện thông qua các giải thuật: đánh nhãn, loại bỏ vùng không mong muốn, định vị vùng chứa các ký tự của biên số xe.

2.2.1 Giải thuật đánh nhãn các vùng liên thông (Connected Component Labeling Algorithm-CCLA)

Mục đích của việc đánh nhãn cho các thành

Bước 1: Tính ảnh tích phân [1-2, 5-6, 8].

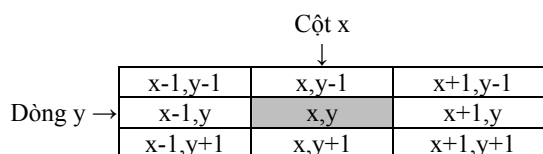
$$I(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (1), \text{ với } I \text{ là ảnh tích}$$

phân; **i** là ảnh đầu vào (Hình 4(b))

Bước 2: Tính tổng hình chữ nhật D (Hình 4(c)), với góc trái trên $(x1,y1)$ và góc phải dưới $(x2,y2)$:

$$\text{Sum} = \text{intImg}[x2,y2] - \text{intImg}[x2,y1-1] - \text{intImg}[x1-1,y2] + \text{intImg}[x1-1,y1-1] \quad (2),$$

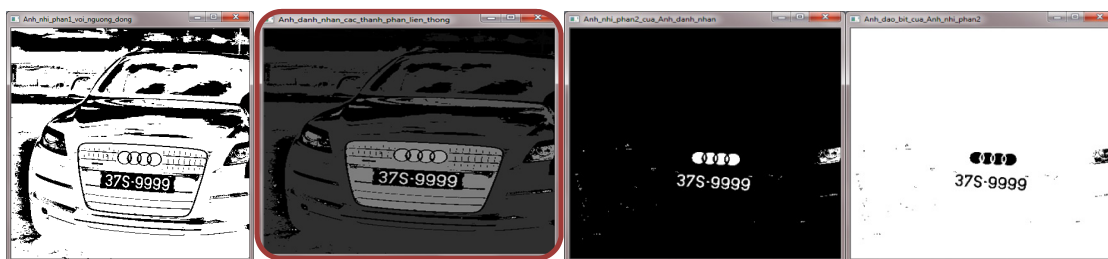
phần liên thông là tách các đối tượng trong ảnh [3]. Giải thuật được thực hiện theo 8 hướng của một điểm ảnh trên ảnh nhị phân (Hình 5). Ảnh đánh nhãn cần được cân bằng lại mức màu về ảnh mức xám với tỉ lệ $=255/\text{số nhãn}$.



Hình 5: Tám điểm xung quanh điểm ảnh có tọa độ (x,y)

2.2.2 Nhị phân hóa cho ảnh đánh nhãn mức xám để giảm bớt vùng liên thông

Ảnh đánh nhãn sẽ được nhị phân hóa với ngưỡng $T=128$ để loại bớt một nửa số vùng liên thông, với hy vọng tìm được ngay biên số xe ở nửa ảnh phía dưới. Nếu không tìm thấy vùng chứa các ký tự trong ảnh này, ngưỡng T sẽ được giảm dần về 0. Qua các kết quả thực nghiệm, chúng tôi đã đề xuất mỗi bước giảm của T là 24 để tăng tốc độ xử lý và tối đa vòng lặp xử lý lúc này sẽ là 5 (Hình 6).



(a) Ảnh nhị phân với ngưỡng động, (b) Ảnh đánh nhãn, (c) Ảnh nhị phân, (d) Ảnh NOT của (c)

Hình 6: Nhị phân hóa ảnh đánh nhãn (Label Image)

2.2.3 Loại bỏ vùng không mong muốn (Unwanted Region Elimination Algorithm- UREA)

Trên ảnh nhị phân của ảnh đánh nhãn có nhiều vùng không thật sự chứa ký tự trên biển số xe, chúng tôi xây dựng giải thuật sau nhằm loại bỏ các vùng không mong muốn để tăng hiệu suất của hệ thống:

Bước 1: Loại bỏ các vùng liên thông với biên của ảnh, vì ký tự của biển số xe không nằm trên vùng liên thông với biên.

Bước 2: Loại bỏ các vùng nhỏ hơn 15 pixels. Từ kết quả thực nghiệm, chúng tôi xác định rằng ký tự có thể nhận dạng được nếu có kích thước tối thiểu là 15 pixels.

Bước 3: Loại bỏ các vùng chứa đường thẳng mỏng 1 pixel có độ dài hơn 20 pixels. Nếu độ dày của ký tự là 1 pixel thì ký tự này rất nhỏ nên các đường thẳng của các ký tự không thể dài hơn 20 pixels.

Bước 4: Tương tự, loại bỏ các vùng có chứa đường thẳng mỏng 2 pixels có độ dài lớn hơn 35 pixels.

Bước 5: Loại các vùng với số lượng pixel quá lớn. Vùng được xem là vùng có số lượng pixel quá lớn khi số lượng pixel của vùng này lớn gấp 4 lần số lượng pixel của một ký tự bất kỳ trong biển số.

2.2.4 Giải thuật định vị vùng ứng viên là vùng biển số xe

Trong bước này, vùng ứng viên chứa các ký tự của biển số xe sẽ được chọn.

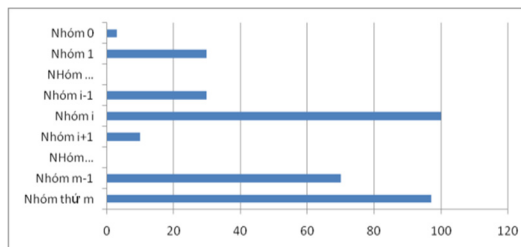
Bước 1: Duyệt qua từng dòng của ảnh, đếm số điểm ảnh màu đen trên mỗi dòng và lưu lại trong một mảng HsumBlackPoint(n), n=0,1,2.. height-1. Sau đó, ta chia ảnh thành các nhóm nhỏ như sau:

$$Num_{group} = \frac{Height}{N} \quad (4)$$

Trong đó: **Num_{group}**: là tổng số nhóm; **Height**: là chiều cao của ảnh;

N: là số dòng nhóm lại thành 1 nhóm. Trong bài viết này, đã thực nghiệm với N=10 [1-2].

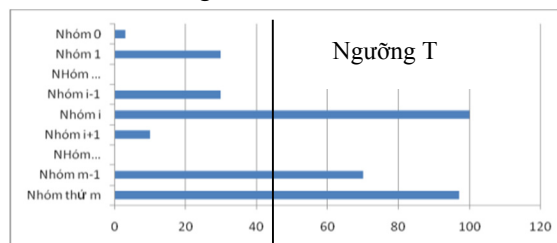
Bước 2: Tính tổng điểm ảnh màu đen trên từng nhóm, bằng cách cộng mười giá trị liên tiếp trong mảng HsumBlackPoint() lại, kết quả lưu trong mảng HsumBlackPointGroup (m), m=0,1,2... height/N.



Hình 7: Chia nhóm, tính tổng pixels đen theo nhóm

Bước 3: Loại bỏ bớt nhóm có ít pixel, duyệt qua mảng HsumBlackPointGroup (m). Nếu giá trị nhóm nào bé hơn ngưỡng T (Hình 8), sẽ được gán giá trị về 0.

Ngưỡng T được xác định bằng thực nghiệm với công thức: $(T \leq \text{MaxSum}/5 \parallel T < 20)$. Trong đó, $\text{MaxSum} = \text{Max}(\text{HsumBlackPointGroup}(m))$, m = 0, 1, 2,... height/N.



Hình 8: Loại bớt nhóm ít pixel với ngưỡng T

Bước 4: Loại bỏ các nhóm đơn độc không là vùng ứng viên chứa biển số. Một nhóm được gọi là đơn độc khi giá trị của nhóm lớn hơn 0 nhưng hai nhóm trước và sau có giá trị bằng 0.

Bước 5: Kết nối các vùng có khả năng là vùng ứng viên của biển số vuông.

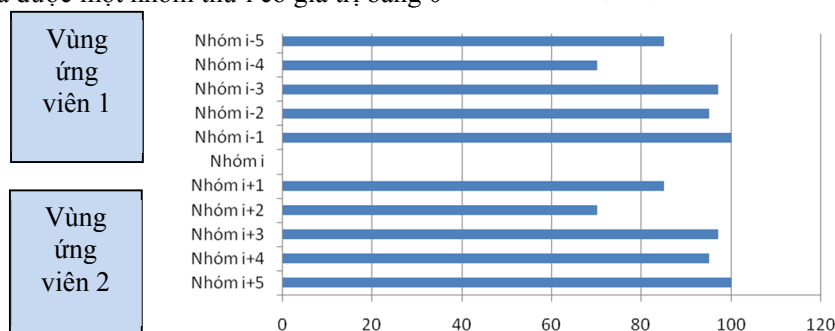
Đối với biển số vuông có thể bị tách thành hai nhóm liền kề nhau, cả hai đều là vùng ứng viên tốt với tỉ lệ điểm ảnh phù hợp. Do đó, một biển số vuông chứa được một nhóm thứ i có giá trị bằng 0

ở giữa, thì ít nhất có 5 nhóm trước nhóm i và 5 nhóm sau nhóm i , tất cả đều lớn hơn 0. Để tăng tính chính xác, ta thêm điều kiện:

$$\frac{\sum \text{Black pixels in region 1}}{\sum \text{Black pixels in region 2}} > 0,35 \quad \text{và}$$

$$\frac{\sum \text{Black pixels in region 2}}{\sum \text{Black pixels in region 1}} > 0,7$$

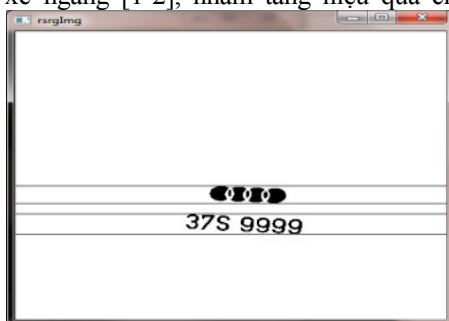
Nếu thỏa tất cả các điều kiện trên thì nhóm thứ i sẽ được gán giá trị là 1. (Hình 9).



Hình 9: Kết nối hai vùng ứng viên lớn kề nhau

Trong thuật toán này, các bước 3, 4, 5 là ba bước đã cải tiến bổ sung cho thuật giải định vị biển số xe ngang [1-2], nhằm tăng hiệu quả cho

việc chọn đúng vùng ứng viên và áp dụng thêm cho biển số xe vuông.



(a) Biển số ngang



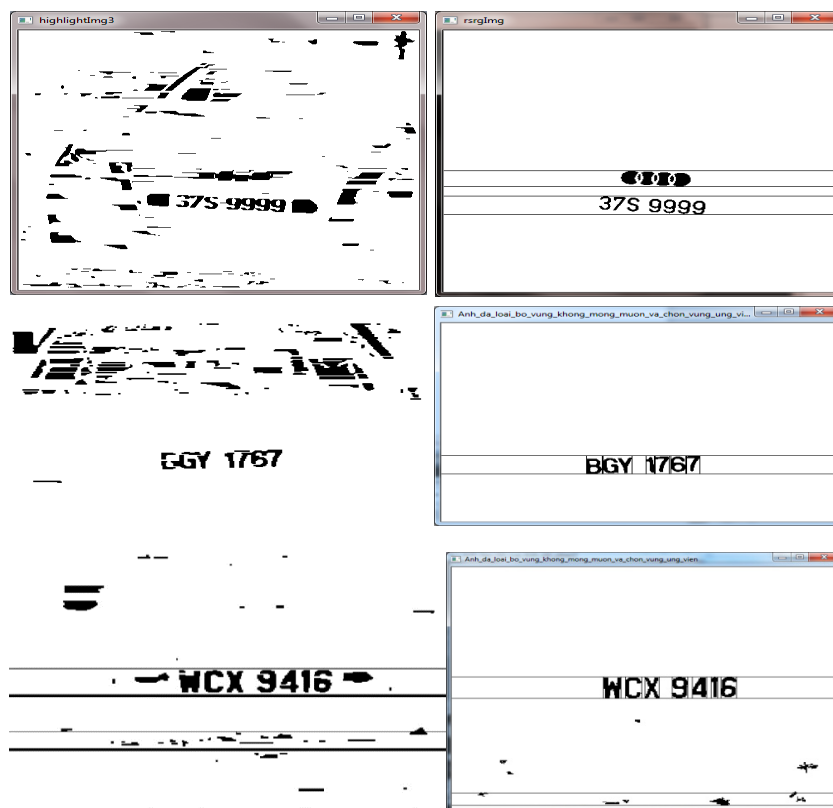
(b) Biển số vuông

Hình 10: Chọn vùng ứng viên chứa các ký tự trên biển số xe

2.2.5 So sánh khả năng khử nhiễu của “phương pháp phát hiện ký tự bằng ảnh đánh nhãn” với “phương pháp phát hiện cạnh (VEDA)”

Trong giai đoạn phát hiện vùng chứa các ký tự, chúng tôi đã so sánh cách tiếp cận được đề xuất với phương pháp sử dụng giải thuật phát hiện cạnh (edges) kết hợp hình thái học (morphology)

với hai giải thuật chính là: giải thuật phát hiện cạnh đứng (VEDA) và giải thuật loại bỏ dòng không mong muốn (ULEA), được nêu rõ trong tài liệu tham khảo [1-2]. Sau khi kiểm tra trên 150 mẫu ảnh đầu vào, 100% kết quả sử dụng “phương pháp phát hiện ký tự bằng ảnh đánh nhãn” của chúng tôi loại nhiễu hiệu quả hơn “phương pháp phát hiện cạnh (VEDA)” (Hình 11).



Hình 11: So sánh hai phương pháp khử nhiễu

Kết quả áp dụng phương pháp phát hiện cạnh (Trái) và phương pháp phát hiện ký tự bằng ảnh đánh nhãn (Phải)

3 QUÁ TRÌNH ĐỊNH VỊ BIỂN SỐ XE VÀ TÁCH KÝ TỰ

Sau khi thu được các vùng ứng viên, bước tiếp theo chúng ta cần xác định các vùng ứng viên này là biển số vuông hay biển số ngang. Trong mục 3.1. chúng tôi đề xuất khái niệm và giải thuật tìm **lát cắt hẹp nhất** để giải quyết vấn đề này. Nếu tồn tại lát cắt hẹp nhất thì ta có biển số vuông, ngược lại là biển số ngang. Tiếp theo ở mục 3.2 và 3.3. chúng tôi trình bày phương pháp tách các ký tự cho từng loại biển số tương ứng để tiến hành dạng các ký tự trên biển số.

3.1 Giải thuật tìm lát cắt hẹp nhất P trên vùng ngang V_i

Lát cắt hẹp nhất P là tổng số điểm đen nhỏ nhất theo dòng có thể chia cắt vùng ứng viên chiều ngang thành 2 phần trong khoảng $[(V_i.y2-V_i.y1)/4, (V_i.y2-V_i.y1)*3/4]$ theo ngưỡng T với $y1, y2$ lần lượt là vị trí chọn dòng đầu và cuối của vùng V_i và $[V_i.y2-V_i.y1]$ là chiều cao của

V_i . Khi ngưỡng $T = \max/5 < 13$, ta chọn $T=13$. Vùng ứng viên biển số ngang có 5 ký tự sẽ có số pixel trên các dòng lớn hơn 13 pixels nên không có lát cắt P. Ngược lại, nếu có lát cắt P sẽ là vùng ứng viên biển số vuông.

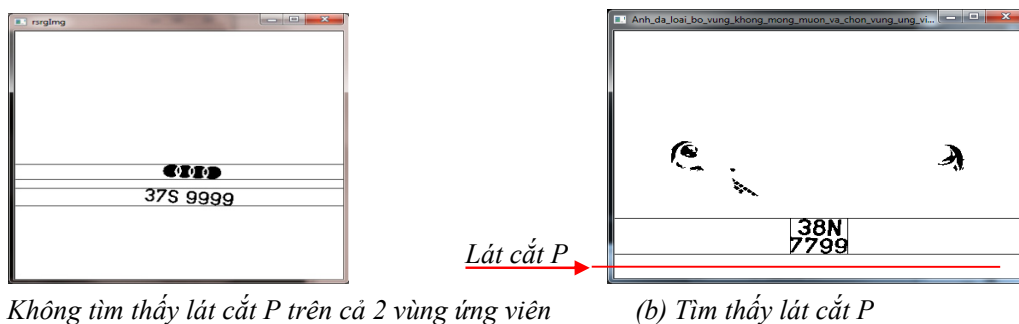
Bước 1: Tính ngưỡng $T = \max/5$ hoặc $T=13$, với max là tổng số điểm đen lớn nhất theo dòng.

Bước 2: Duyệt từ vị trí $(V_i.y2-V_i.y1)/4$ đến $(V_i.y2-V_i.y1)*3/4$, với $[V_i.y2-V_i.y1]$ là chiều cao của V_i , tìm dòng có số pixel nhỏ nhất L.

Bước 3: Nếu L bé hơn hoặc bằng T thì tồn tại lát cắt P. Ngược lại, không tồn tại lát cắt P.

Bước 4: Nếu tồn tại lát cắt P, kết luận đây là vùng ứng viên biển số vuông. Ngược lại, là vùng ứng viên biển số ngang.

Một ví dụ minh họa cho giải thuật tìm lát cắt hẹp nhất được trình bày ở Hình 12.



Hình 12: Tìm lát cắt hẹp nhất P

3.2 Giải thuật trích các ký tự cho biển số ngang

Nếu vùng ứng viên thuộc biển số ngang ta sử dụng thuật toán sau để tách các ký tự thuộc biển số (trong đó: y_1 , y_2 lần lượt là vị trí chọn dòng đầu và cuối của vùng, width và h lần lượt là chiều rộng và chiều cao của vùng đang xét).

Bước 1. Tính tổng pixel đen theo cột của vùng ngang từ (0, y_1) đến (width, y_2)

Bước 2. Tìm vùng ký tự (**box**) dựa trên lát cắt hẹp 1 pixel, tính chiều rộng w , chiều cao h và tính luôn chiều cao thực ch của vùng.

Bước 3. Điều chỉnh biên thực đối với các ký tự I, L, H, N, M, W, V, U

Bước 4. Loại vùng ký tự với tỉ lệ w/h không hợp lệ như: dấu chấm, dấu gạch nối...

Bước 4.1. Loại vùng ký tự có tỉ lệ $w/h > 2.5$ và vùng có $ch < \max(ch)/2$

Bước 4.2. Tách ký tự lần 2, với lát cắt = 2 pixels, cho ký tự kép có: $w/h \geq 1.3$ và $w/h \leq 2.5$

Bước 5. Lọc trên số lượng ký tự, cụm ký tự:

Bước 5.1. Nếu số ký tự $< 6 \Rightarrow$ Loại vùng ngang.

Bước 5.2. Loại box có tổng pixel đen < 15 pixels.

Loại box có tổng pixel đen/ $w.h > 90\%$ và tổng pixel đen/ $w.h < 10\%$

Bước 5.3. Nếu có nhiều cụm ký tự, loại cụm có số box < 6

Bước 5.4. Đếm lại số box, loại vùng ngang nếu số box < 6

Bước 6. Đếm lại số box sau khi đã loại các box không hợp lệ

Bước 7. Trả về vị trí biển số xe, vị trí các ký tự và vẽ lên ảnh đầu ra.

3.3 Giải thuật trích các ký tự cho biển số vuông

Ngược lại, nếu vùng ứng viên thuộc biển số vuông ta tiến hành tách các ký tự thuộc biển số như sau:

Bước 1: Tách vùng ứng viên biển số vuông thành 2 vùng tại vị trí lát cắt hẹp nhất P.

Bước 2: Áp dụng giải thuật chọn vùng ký tự cho biển số ngang trên 2 vùng con này.

Bước 3: Tính tổng hợp số ký tự trên 2 vùng con. Nếu số ký tự < 5 , loại vùng ứng viên ký tự này. Ngược lại, đây là biển số vuông.

Kết quả thực hiện giải thuật định vị ký tự trên biển ngang (Hình 13(a)) và vuông (Hình 13(b),(c)).



(a) Định vị ký tự trên biển số ngang (b) Biển số vuông được định vị trước (c) Định vị các ký tự

Hình 13: Định vị ký tự trên biển số ngang và vuông

4 QUÁ TRÌNH TRÍCH RÚT BỘ ĐẶC TRƯNG VÀ NHÂN DẠNG KÝ TỰ

4.1 Trích đặc trưng ảnh

Mẫu ký tự huấn luyện được véc tơ hóa dùng phương pháp chia lưới để tính trung bình mức xám của cửa sổ con kích thước 4x4. Ảnh mẫu có kích thước 32x24 sẽ được biến đổi thành véc tơ 48 chiều $\{x_1, x_2, \dots, x_{48}\}$ (Hình 14).

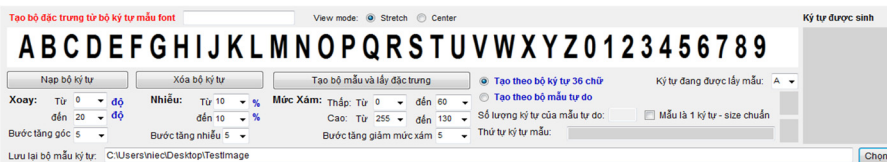


Hình 14: Cách chia lưới tính đặc trưng

4.2 Mô hình mạng Nơ-ron truyền thẳng nhiều lớp cho nhận dạng ký tự [4], [7]

Kiến trúc mạng: mạng nơ-ron truyền thẳng kết hợp giải thuật lan truyền ngược lỗi.

Mạng Nơ-ron gồm 3 lớp: lớp các giá trị đầu vào, lớp các nơ-ron ẩn, lớp các nơ-ron đầu



Hình 16: Chương trình tạo bộ mẫu ký tự

Tạo mẫu từ những ảnh ký tự được lấy trên các biển số xe thực tế:

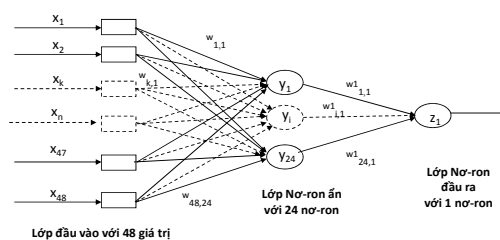


Từ bộ ảnh 375.342 ký tự mẫu, chúng tôi đã tạo bộ dữ liệu đặc trưng chung cho tất cả các ký tự, từ đó tạo ra 36 bộ dữ liệu đặc trưng cho từng ký tự để thực hiện huấn luyện. Trong quá trình huấn luyện chúng tôi còn tạo ra 2 bộ dữ liệu để kiểm tra và kiểm thử độc lập với bộ dữ liệu huấn luyện cho mỗi ký tự. Bộ dữ liệu kiểm tra huấn luyện có **142856** mẫu, bộ dữ liệu kiểm thử sau khi huấn luyện có **8518** mẫu.

4.4 Huấn luyện trọng số cho từng mạng

Mỗi mạng của 36 mạng được huấn luyện theo các thông số sau:

ra. Có 2 bộ trọng số tại các liên kết từ lớp đầu vào đến lớp ẩn và từ lớp ẩn đến lớp đầu ra.



Hình 15: Mô hình mạng Nơ-ron cho nhận dạng ký tự

Số lượng mạng: Mỗi mạng huấn luyện nhận dạng 1 ký tự, ta có tổng cộng 36 mạng cho 36 ký tự.

4.3 Xây dựng bộ dữ liệu cho nhận dạng ký tự

Trước khi tạo bộ dữ liệu chúng tôi đã tạo bộ ký tự mẫu cho tập huấn luyện như sau:

Tạo mẫu từ bộ font: Mẫu ký tự được điều chỉnh thêm tham số mức xám thấp cho màu nét chữ, mức xám cao cho nền, góc quay trái và phải, nhiều dóm cho các font Normal Arial, Bold Arial, **Arial Black**.

- Lớp đầu vào: 48 giá trị; Lớp các nơ-ron ẩn: 24 nơ-ron; Lớp các nơ-ron đầu ra: 1 nơ-ron
- Hệ số học: $\alpha=0.001$; Hằng số quán tính (bước đà): $\beta=0.7$.
- Số vòng huấn luyện: từ 300 đến 1500 vòng tùy từng ký tự.
- Độ chính xác huấn luyện mong muốn: 99.999%

4.5 Nhân dạng ký tự

Giải thuật nhận dạng được thực hiện qua bốn bước sau:

Bước 1: Duyệt qua từng ký tự của biến số.

Bước 2: Lấy đặc trưng của ký tự và cho chạy qua 36 mạng nơ-ron.

Bước 3: Tổng hợp 36 kết quả đầu ra của 36 mạng Nơ-ron, ta chọn kết quả lớn nhất.

Bước 4: Kết quả được chọn ở mạng nào thì ký tự ở mạng đó là kết quả.

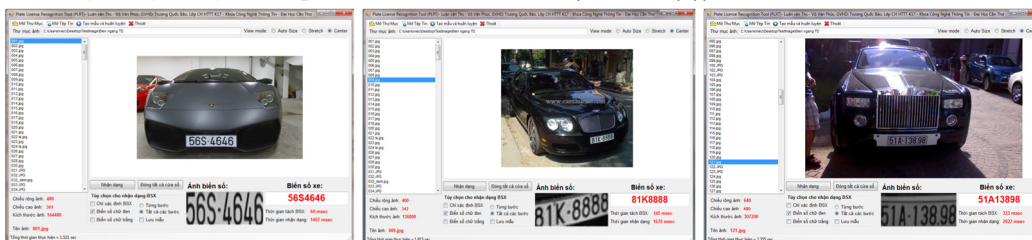
5 MỘT SỐ KẾT QUẢ THỰC NGHIỆM

Trong phần này, chúng tôi trình bày một số kết quả nhận dạng biển số xe thực nghiệm với ảnh biển số xe đầu vào được chụp trong nhiều điều kiện khác nhau nhằm minh họa hết khả năng và các điểm mạnh của hệ thống. Đồng thời thống kê độ chính xác của hệ thống trên các tập dữ liệu ảnh khác nhau được trình bày ở Bảng 1, các kết quả nêu ra trong bảng cũng được phân tích đầy đủ ở

phần 5.6 của mục này.

5.1 Ảnh biển số xe với các góc và hướng chụp khác nhau

Trong trường hợp góc chụp thẳng, ảnh ký tự thường rõ nên dễ xử lý và nhận dạng chính xác. Biển số thường dày và lớn nên giải thuật loại bỏ vùng không mong muốn dễ dàng loại đi (Hình 17(a)). Ngược lại, trường hợp góc chụp chéo, khó khăn là các ký tự có thể không tách được do góc nghiêng của ký tự quá lớn. Để giải quyết vấn đề này, chúng ta cần xoay lại ảnh biển số. Với điều kiện góc nghiêng của các ký tự trên biển số bé hơn 15 độ hệ thống sẽ tách và nhận dạng được (Hình 17(b)).



(a) Góc chụp thẳng

(b) Góc chụp chéo

(c) Chụp ngược chiều sáng

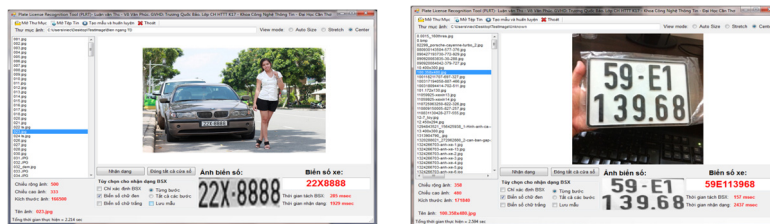
Hình 17: Ảnh biển số xe với các góc chụp khác nhau

Trường hợp ảnh chụp ngược chiều sáng sẽ có độ tương phản thấp, ảnh bị mờ, nền sáng nên khó xác định ngưỡng. Với giải thuật phân hóa với ngưỡng động đã giúp khôi phục vùng biển số (Hình 17(c)).

5.2 Ảnh biển số xe với các khoảng cách chụp khác nhau

Trong trường hợp ảnh chụp gần với kích

thước quá lớn, hệ thống sẽ điều chỉnh kích thước ảnh về 800x600 pixel rồi mới xử lý nhận dạng. Do tính chất bao đóng của ký tự và tỉ lệ qui ước xử lý dựa trên kích thước động của ký tự nên không ảnh hưởng trong trường hợp chụp gần hay chụp xa (Hình 18).



(a) Ảnh chụp xa

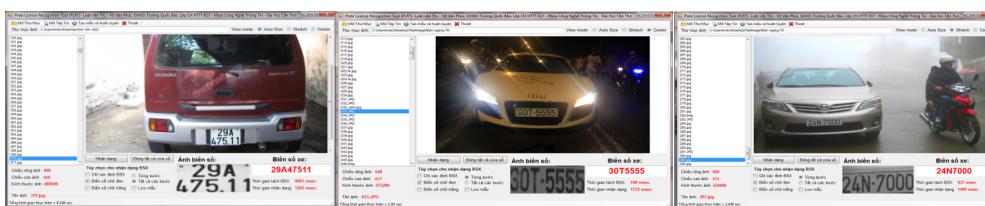
(b) Ảnh chụp gần

Hình 18: Ảnh biển số xe với các khoảng cách chụp khác nhau

5.3 Ảnh biển số xe được chụp với các điều kiện thời tiết khác nhau

Trong điều kiện thời tiết tốt kể cả trường hợp bị nhiễu (Hình 19(a)) hệ thống dễ dàng nhận dạng được biển số xe một cách chính xác. Ngược lại, về đêm hay trời mưa, ảnh biển số bị mờ, có độ tương phản thấp, ảnh sẽ có thêm vùng ánh sáng

trắng của đèn vào ban đêm hoặc nhiễu do mưa. Giải thuật nhị phân hóa với ngưỡng động đã giúp khôi phục vùng biển số rất tốt, vùng ánh sáng của đèn sẽ là vùng có số lượng pixel lớn và nhiễu của mưa sẽ là vùng có số lượng pixel nhỏ sẽ bị loại bởi giải thuật loại bỏ vùng không mong muốn (Hình 19(b),(c)).



(a) Ban ngày

(b) Ban đêm

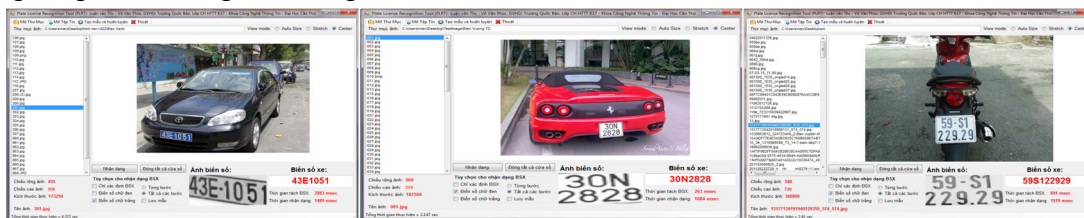
(c) Trời mưa

Hình 19: Ảnh biển số xe được chụp trong điều kiện thời tiết khác nhau

5.4 Ảnh các dạng biển số xe khác nhau

Ngoài biển số xe dạng ngang và vuông, hệ thống cũng có khả năng nhận dạng biển số xe với

nhều màu sắc khác nhau như biển số màu trắng, xanh, đỏ,... (Hình 20).



(a) Biển số xe màu xanh

(b) Biển số xe vuông

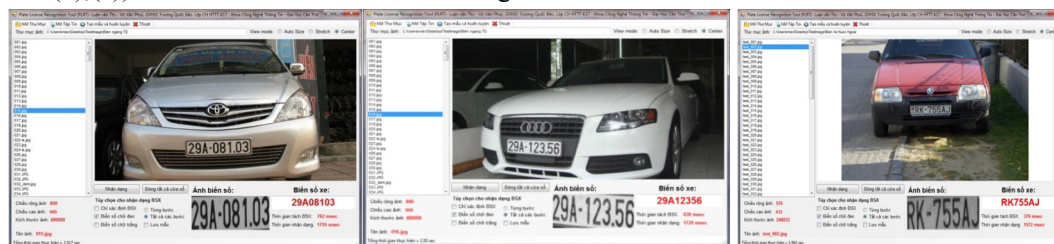
(c) Biển số xe mô tô

Hình 20: Ảnh các dạng biển số xe khác nhau

5.5 Ảnh dạng biển số xe bị nhiễu

Điểm mạnh lớn nhất của phương pháp tiếp cận của chúng tôi là loại được rất nhiều dạng nhiễu khác nhau cho biển số xe như: biển số có dấu chấm, dấu 2 chấm và dấu gạch nối (Hình 21(a)), biển số nước ngoài có vòng logo, huy hiệu (Hình 21(b),(c)), nhiễu ký tự trong hoặc ngoài biển số (Hình 21(d),(e)) hoặc ảnh biển số có chứa vùng

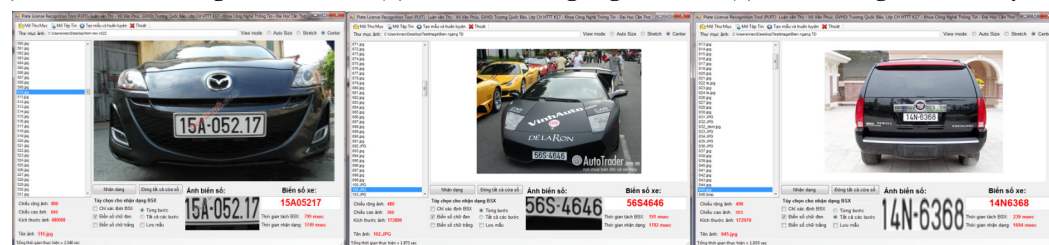
dạng hình vuông, hình chữ nhật giống như biển số xe (Hình 21(f)) mà một số phương pháp nhận dạng dựa trên cạnh có thể nhận nhầm vùng ứng viên biển số xe. Giải thuật loại bỏ vùng không mong muốn và tách ký tự sẽ dễ dàng loại bỏ các vùng dư thừa này nhờ vào các thông số tỷ lệ kích thước và số lượng pixels của các vùng ứng viên được xét.



(a) Nhiễu do chấm, gạch nối

(b) Nhiễu do vòng logo

(c) Nhiễu do gạch nối, huy hiệu



(d) Nhiễu do chữ trên biển số

(e) Nhiễu do chữ ngoài biển số

(f) Nhiễu hình vuông, chữ nhật

Hình 21: Các dạng nhiễu trên ảnh biển số xe

5.6 Thống kê kết quả kiểm tra độ chính xác của hệ thống nhận dạng biển số xe trên các tập ảnh

Trong bảng thống kê sau, lỗi định vị biển số

Bảng 1: Thống kê kết quả kiểm tra trên các tập ảnh

Tên mẫu	Số lượng	Định vị	Tách ký tự	Nhận dạng	Tổng hợp
Biển số ngang	450	425 (94,4%)	420 (98,8%)	414 (98,6%)	92,0%
Biển số vuông	150	135 (90%)	132 (97,8%)	129 (97,8%)	86%
Xe nước ngoài	60	58 (96,7%)	57 (98,2%)	55 (96,5%)	91,7%

Ngoài ra, quá trình tách ký tự có thể tách thiếu ký tự hoặc dư là do lấy ngưỡng cho ảnh nhị phân làm mất ký tự, loan vùng ký tự và biên biển số, do biển số nghiêng nhiều nên các ký tự kết dính không tách được. Đối với biển số vuông, ngoài phần lỗi như biển số ngang còn thêm một lỗi nữa là không tìm được lát cắt hẹp nhất P. Các mẫu kiểm tra xe nước ngoài bị lỗi cũng do các nguyên nhân trên.

Đối với phần nhận dạng lỗi chủ yếu do các ký tự tách được: bị mất nét, bị nhiễu quá nhiều, ký tự kép, thiếu và dư ký tự, hoặc không tách đúng ký tự.

6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

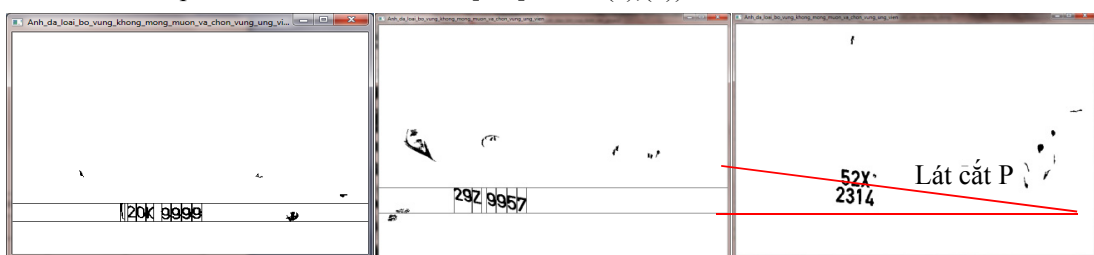
Chúng tôi đã xây dựng hoàn thiện công cụ định vị và nhận dạng biển số xe ô tô với bộ dữ liệu 200 ảnh tự chụp, 20 ảnh trích từ bài báo [1-2]

ngang chủ yếu do giải thuật thực hiện tách ký tự trước sau đó mới định vị, trong trường hợp tách ký tự bị thiếu hoặc dư sẽ định vị biển số không chính xác.

và các tập ảnh khác được thu thập từ mạng internet [12-15] sử dụng công cụ lập trình visual C++2008 và thư viện OpenCV 2.1. Phương pháp được đề xuất là sự kết hợp của 2 giải thuật: *đánh nhãn cho các thành phần liên thông* [3], *loại bỏ vùng không mong muốn*. Hệ thống hóa được các bước định vị, tách ký tự và nhận dạng biển số xe. Ngoài ra, hệ thống còn được xây dựng thêm các công cụ tạo nhanh bộ dữ liệu huấn luyện cho nhận dạng ký tự với nhiều nguồn ảnh mẫu, công cụ huấn luyện mạng nơ-ron nhận dạng ký tự.

Bên cạnh những ưu điểm đã đạt được vẫn còn những vấn đề cần được cải thiện trong tương lai:

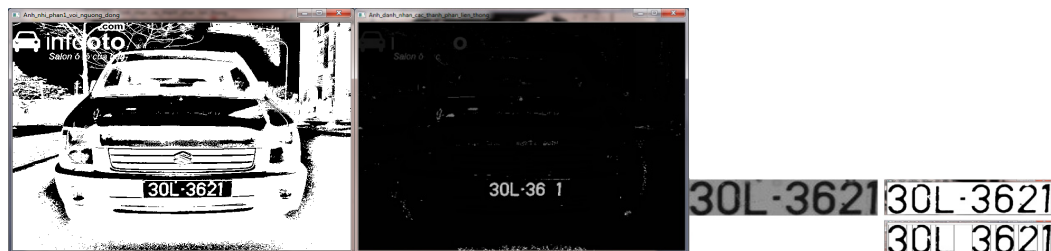
Giải thuật nhị phân hóa với ngưỡng động có thể làm dư ký tự hoặc mất ký tự. Lát cắt hẹp 1 pixel có thể làm mất nét ký tự đối với biển số chữ nhỏ, nét mỏng và nghiêng (Hình 22(a),(b)).



(a) Dư ký tự

(b) Ký tự kép

(c) Biển số vuông có độ nghiêng lớn hơn 15 độ



(d) Chữ nhiễu biên biển số

(e) Mất chữ do nhiễu biên

(f) Cải tiến bằng cách định vị lại ký tự

Hình 22: Ảnh minh họa một số vấn đề cần được cải tiến

Do ảnh đầu vào bị lệch trước khi tách ký tự nên một số ký tự trên biển số nghiêng sẽ bị dính

kép, với lát cắt hẹp 2 pixels có thể không tách được và nhận dạng sẽ thiếu ký tự. Hệ thống nhận

dạng tốt đối với biên số có độ nghiêng chữ dưới 15 độ (Hình 22(c)).

Do ký tự được tách cùng lúc với biên số xe nên ký tự bị nhiễu biên có thể bị mất khi áp dụng giải thuật đánh nhãn và nhị phân hóa. Lỗi này có thể khắc phục bằng cách định vị lại ký tự sau khi đã định vị biên số xe (Hình 22(d),(e),(f)).

TÀI LIỆU THAM KHẢO

1. Abbas M. Al-Ghaili, Syamsiah Mashohor, Alyani Ismail, and Abdul Rahman Ramli, 2008. "A New Vertical Edge Detection Algorithm and its Application", International Conference on Computer Engineering & Systems, ICCES 2008, pp.204-209.
2. Abbas M. Al-Ghaili, Syamsiah Mashohor, Abdul Rahman Ramli and Alyani Ismail, 2010. "Car License Plate Detection Method for Malaysian Plates-Styles by Using a Web Camera", Pertanika J. Sci. & Technol. Vol. 18, No. 2, pp. 303-319.
3. Alain Boucher-IFI, 2012. "Image processing & Computer vision", Can Tho University. slide 9 – "Binary Image" of course, pp. 10-22.
4. Bekir Karlik and A.Vehbi Olgac, 2011. "Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks", International Journal of Artificial Intelligence And Expert Systems (IJAE), Vol. 1, No. 4, pp. 111-122.
5. D. Bradley and G.Roth, 2007. "Adaptive thresholding using the integral image", J. Graphics Tools, Vol. 12, No. 2, pp. 13–21.
6. F.Shafait, D.Keysers, and T.M.Breuel, 2008. "Efficient Implementation of Local Adaptive Thresholding Techniques using Integral Images", International Conference on Document Recognition and Retrieval XV, Electronic Imaging, SPIE-IS&T Vol. 6815, pp.1-6.
7. Haris Al-Qodri Maarif, and Sar Sardy, 2006. "Plate Number Recognition by Using Artificial Neural Network", J. Electronic and Electrical (Network), Vol. 3, No. 1, pp. 176-182.
8. Neeta Nain, Gaurav Jindal, Ashish Garg and Anshul Jain, 2008. "Dynamic Thresholding Based Edge Detection", Proceedings of the World Congress on Engineering 2008, Vol. 1, , No. 1, pp. 2-7.
9. P.Sa-ngamuang, C.Thamnittasana, T.Kondo, 2007. "Thai Car License Plate Recognition Using Essential-Elements-Based Method", Proceedings of Asia-Pacific Conference on Communications, pp. 41-44.
10. Saeed Rastegar, Reza Ghaderi, Gholamreza Ardeshir & Nima Asadi, 2009. "An intelligent control system using an efficient License Plate Location and Recognition Approach", International Journal of Image Processing (IJIP), Vol. 3, No. 5, pp. 252-264.
11. Vahid Abolghasemi and Alireza Ahmadyfard, 2007. "A Fast Algorithm for License Plate Detection", Proceedings of the 9th international conference on Advances in visual information systems, pp. 468–477.
<http://otocuvn.com.vn/>
<http://choxe.net/oto/>
<http://www.infoto.com/>
<http://www.bonbanh.com/>