

## MỘT GIẢI PHÁP TRONG XÂY DỰNG HỆ THỐNG HỖ TRỢ GIỮ XE THÔNG MINH

Nguyễn Thái Nghe<sup>1</sup>, Nguyễn Văn Đồng<sup>1</sup> và Võ Hùng Vĩ<sup>1</sup>

<sup>1</sup> Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

### Thông tin chung:

Ngày nhận: 27/05/2014

Ngày chấp nhận: 29/12/2014

### Title:

A solution for building intelligent parking support system

### Từ khóa:

Hệ thống giữ xe thông minh, nhận dạng biển số xe, đặc trưng Haar-like, mã vạch

### Keywords:

Intelligent parking support system, motorcycle license plate recognition, Haar-like feature, barcode

### ABSTRACT

In this study, we propose a solution for building an Intelligent Parking Support System (IPSS). The IPSS uses three recognition techniques including automatic recognition for motorcycle license plate (using cascade of Boosting with Haar-like features and Support Vector Machines - SVM), barcode recognition, and semi-recognition via surveillance cameras. Experimental results show that the models work well in three stages. In the license plate area recognition stage, the model gets 99% of accuracy when we use 750 images for training and 243 images for testing. In the letter area recognition stage, the model achieves 95.88% of accuracy when we use 11866 and 4755 images for training and testing, respectively. In the last stage, we train the SVM model on 2603 records and test it on 1550 records. The outcome result is the accuracy of 98.99%. This is a full-fledged system and can be applied in practice.

### TÓM TẮT

Bài viết này giới thiệu một giải pháp trong xây dựng “Hệ thống hỗ trợ giữ xe thông minh - IPSS”. Nhằm tăng độ an toàn cho bãi xe, IPSS kết hợp các kỹ thuật nhận dạng gồm: Nhận dạng và rút trích biên số tự động (dùng giải thuật Boosting phân tầng cùng với đặc trưng Haar-like và sau đó là giải thuật máy học SVM), nhận dạng mã vạch, và nhận dạng bán tự động qua các camera quan sát. Thực nghiệm cho thấy kết quả nhận dạng khá tốt ở cả ba giai đoạn: Định vị biển số đạt độ chính xác 99% khi dùng 750 ảnh để huấn luyện và 243 ảnh để kiểm thử. Định vị ký tự chính xác 95.88% khi dùng 11866 ảnh cho huấn luyện và 4755 ảnh để kiểm thử. Phân loại ký tự bằng giải thuật SVM chính xác 98.99% khi sử dụng 2603 phần tử cho huấn luyện và 1550 phần tử cho đánh giá. Hệ thống đã được cài đặt hoàn chỉnh và có thể đưa vào ứng dụng trong thực tế.

## 1 GIỚI THIỆU

Do mật độ dân số ở nước ta phân bố khá đồng tại các thành phố lớn trong khi các phương tiện giao thông công cộng còn chưa phát triển kịp nên đã bùng nổ các phương tiện giao thông cá nhân, đặc biệt là xe máy. Chính vì thế công tác quản lý, trông coi, kiểm soát xe máy, đặc biệt là phát hiện các hành vi của kẻ gian,... tốn nhiều thời gian và công sức.

Trong thực tế, đã có nhiều nhóm nghiên cứu và thương mại hóa bài toán xây dựng hệ thống hỗ trợ giữ xe, chẳng hạn như nhóm Mắt thần thuộc Trung tâm Công nghệ Mô phỏng - Học viện Kỹ thuật quân sự đã cho ra đời sản phẩm “Hệ thống quản lý bãi xe thông minh ứng dụng công nghệ nhận dạng biển số kết hợp với công nghệ thẻ” năm 2006; Công ty cổ phần đầu tư phát triển phần mềm Biển Bạc đã nghiên cứu và phát triển thành công phần

mềm STM01 trong quản lý phương tiện và phát hiện vi phạm giao thông năm 2010; “Bãi giữ xe thông minh” của công ty AIO, hệ thống này có tích hợp máy nhận dạng vân tay (từ hệ thống chấm công) do đó chỉ có chính người đưa xe vào mới có thể lấy xe ra. Điều này có thể gây bất tiện, chẳng hạn như ai đó mượn nhờ xe người khác hoặc ở các siêu thị, khu mua sắm khi người đưa xe vào là chồng, còn người lấy xe ra là vợ do người chồng bận khiêng hàng hóa,...; Hay hệ thống “PTP\_SP : Hệ thống giữ xe thông minh” của công ty Vương Thịnh Phát đang được cài đặt khá nhiều tại các siêu thị hiện nay. Tuy nhiên, hệ thống này chỉ sử dụng camera để ghi lại hình ảnh và biển số xe của người gửi để hỗ trợ cho nhân viên giữ xe quan sát chứ chưa nhận dạng tự động, và do vậy việc nhận dạng vẫn phải cần nhân viên thực hiện. Một số nghiên cứu khác cũng tập trung trên lĩnh vực bán tự động hoặc chỉ dừng lại ở mức kiểm tra thử nghiệm giải thuật.

Từ những thực tế trên, việc nghiên cứu và xây dựng một hệ thống hỗ trợ giữ xe thông minh - tên gọi tắt là IPSS (Intelligent Parking Support System) để có thể ứng dụng trong thực tế vẫn là cần thiết. Đặc biệt trong nghiên cứu này, chúng tôi kết hợp các kỹ thuật gồm việc nhận dạng biển số, nhận dạng ký tự, phân loại ký tự bằng giải thuật máy học SVM, sử dụng công nghệ barcode để kiểm tra thẻ xe vào ra. Bên cạnh phương pháp tự động, nhằm tăng độ an toàn tối đa, hệ thống này cũng hỗ trợ lưu trữ và truy xuất lại ảnh biển số xe, ảnh khuôn mặt của người gửi xe để nhân viên có thể so sánh đối chiếu.

## 2 NHỮNG NGHIÊN CỨU LIÊN QUAN TRONG LĨNH VỰC NHẬN DẠNG

Bài toán phát hiện các đối tượng trong ảnh như gương mặt người, nụ cười (smile),... rất được quan tâm. Các nhà sản xuất máy ảnh như Canon, Nikon, Samsung,... cũng đã tích hợp các giải thuật nhận dạng vào trong sản phẩm của mình về phát hiện mặt người. Trong những nghiên cứu trước đây về lĩnh vực nhận dạng biển số xe, các phương pháp có thể trình bày trong 3 nhóm: phương pháp hình thái học, phương pháp so khớp mẫu, và phương pháp máy học (J.-W. Hsieh *et al.*, 2002, K.-I. Kim *et al.*, 2002).

Với phương pháp biến đổi Hough, V. Kamat và S. Ganesan (1995) đã đề xuất như sau: Sử dụng biến đổi Hough để phát hiện các đường thẳng song song được xem như các “ứng cử viên” của biển số. Sau đó, các ràng buộc hình học của biển số được áp dụng để kiểm tra xem các ứng viên này có phải

là biển số hay không. Bên cạnh đó một số ràng buộc chuẩn như chiều cao/rộng của các ký tự phải bằng nhau, tâm các ký tự phải nằm trên một đường thẳng, chiều cao của mỗi ký tự phải lớn hơn phân nửa chiều cao biển số trích xuất được J.-W. Hsieh (2002) giới thiệu. Một tiếp cận khác là sử dụng thống kê về đường biên và các toán tử hình thái để trích xuất các ứng viên (biển số xe), sau đó loại bỏ các ứng viên không đúng bằng cách kiểm tra các ràng buộc. Những phương pháp này có ưu điểm là không phụ thuộc vào màu sắc của biển số xe, tuy nhiên chúng phụ thuộc khá nhiều vào bước trích đặc trưng biên cạnh (các đoạn thẳng ứng viên thu được thường ngắn hơn nhiều so với chiều dọc cũng như chiều ngang của biển số).

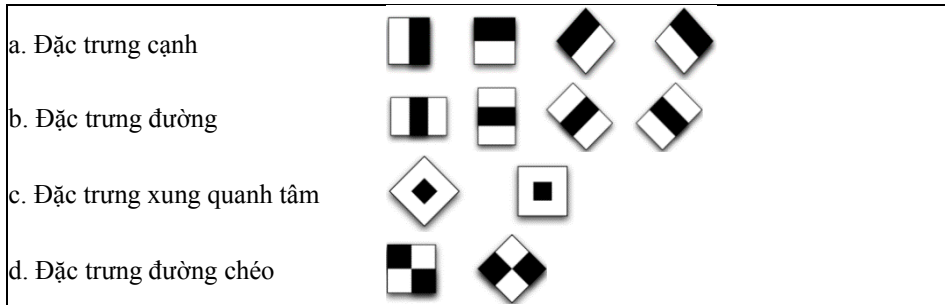
Trong phương pháp so khớp mẫu, các mẫu chuẩn của biển số xe sẽ được nhận dạng trước hoặc nhận dạng các tham số thông qua một hàm. Từ một ảnh đưa vào, tính các giá trị tương quan so với các mẫu chuẩn. Thông qua các giá trị tương quan đó để quyết định có tồn tại biển số xe trong ảnh hay không. Phương pháp này đơn giản, dễ cài đặt, tuy vậy nó chưa hiệu quả khi tỷ lệ, tư thế, và hình dáng thay đổi.

Ở phương pháp máy học, K.-I. Kim *et al.* (2002) giới thiệu phương pháp phát hiện biển số xe bằng cách áp dụng máy học SVM lên trên các “ứng viên” được xem như là các vùng có kết cấu rời rạc (discriminatory texture). Khuyết điểm của phương pháp này là phụ thuộc nhiều vào hình dáng và vẻ ngoài của biển số, do đó nó không hiệu quả trong điều kiện nhiễu, thiếu độ sáng và thay đổi phép quay. Để khắc phục các điểm bất lợi trên, Q.-N. Tran *et al.* (2008) đề xuất một nhận dạng bằng cách kết hợp những điểm mạnh của vectơ mô tả SIFT (Scale-Invariant Feature Transform) và máy học SVM thông qua việc trích các đặc trưng của biển số xe cũng như không phải biển số xe đưa vào cho giải thuật SVM học, từ đó có thể phân loại cho các đối tượng mới. Phương pháp này khắc phục được các thách thức chính trong phân loại ảnh như biến đổi tỉ lệ, thay đổi góc nhìn, phép quay, thay đổi về độ sáng, nhiễu. Tuy nhiên, nó có độ phức tạp cao và thời gian thực hiện khá lâu.

## 3 ĐẶC TRƯNG HAAR-LIKE, GIẢI THUẬT BOOSTING PHÂN TẦNG VÀ MÔ HÌNH SVM

### 3.1 Đặc trưng Haar-like

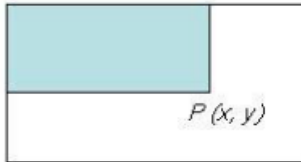
Các đặc trưng Haar-Like (P.-A. Viola và M.-J. Jones, 2001) là những hình chữ nhật được phân thành các vùng khác nhau như biểu diễn trong Hình 1:



Hình 1: Các đặc trưng Haar-Like

Giá trị của đặc trưng Haar-like được xác định bởi độ chênh lệch giữa tổng giá trị pixel mức xám nằm trong vùng đen so với vùng trắng:

$$f(x) = \frac{\text{Tổng vùng đen (các mức xám của pixel)}}{\text{Tổng vùng trắng (các mức xám của pixel)}} -$$



$$P(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Hình 2: Phương pháp tính Integral Image

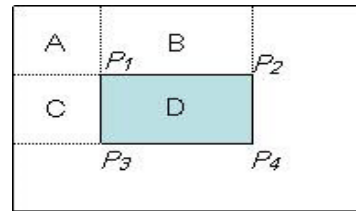
Như vậy để tính các giá trị của đặc trưng Haar-like, người ta phải tính tổng của các pixel trên ảnh. Nhưng để tính toán các giá trị của các đặc trưng Haar-like cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn, không đáp ứng được cho ứng dụng đòi hỏi tính run-time. Do đó, Viola và Jones (2001) đã đưa ra một khái niệm gọi là Integral Image - là một mảng hai chiều với kích thước bằng với kích thước của ảnh cần tính các đặc trưng

$$D = \underbrace{(x_4, y_4)}_{A+B+C+D} - \underbrace{(x_2, y_2)}_{A+B} - \underbrace{(x_3, y_3)}_{A+C} + \underbrace{(x_1, y_1)}_A$$

### 3.2 Mô hình phân lớp phân tầng (Cascade Classifier)

Trong mô hình phân lớp phân tầng, mỗi tầng là một bộ phân lớp. Một mẫu để được phát hiện là đối tượng thì nó cần phải đi qua hết tất cả các tầng của mô hình, các bộ phân lớp ở tầng sau được huấn

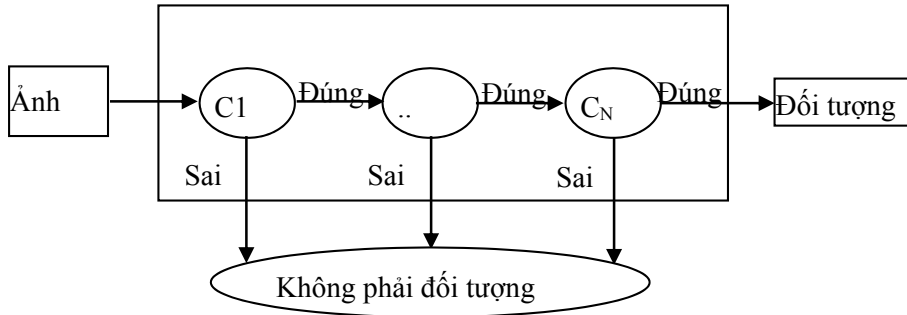
luyện bằng những mẫu mà bộ phân lớp trước nó nhận dạng sai. Mô hình cascade cho nhận dạng đối tượng được biểu diễn trong Hình 4. Đối tượng trong nghiên cứu này là ảnh biển số xe ở giai đoạn định vị biên số và là ảnh ký tự ở giai đoạn định vị ký tự.



Hình 3: Tính giá trị mức xám của vùng D trên ảnh

Sau khi đã tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện theo cách sau: Giả sử ta cần tính tổng các giá trị mức xám của vùng D như trong Hình 3, ta có thể tính:  $D = A + B + C + D - (A+B) - (A+C) + A$

Với  $A + B + C + D$  chính là giá trị tại điểm P4 trên Integral Image, tương tự như vậy  $A+B$  là giá trị tại điểm P2,  $A+C$  là giá trị tại điểm P3, và  $A$  là giá trị tại điểm P1. Vậy ta có thể viết lại biểu thức tính D:



**Hình 4: Mô hình phân lớp phân tầng**

Thuật toán này tiến hành phân ra thành các giai đoạn (stage), các “vùng ứng viên” phải đạt ngưỡng đúng (threshold) của tất cả các giai đoạn để được xác định là có chứa đối tượng. Số lượng giai đoạn và ngưỡng quyết định bởi dữ liệu học và các tham số khác do người dùng chỉ định. Để có thể nhận dạng thuật toán tiến hành lấy mẫu từ 2 tập dữ liệu: tập ảnh dương bao gồm ảnh có chứa đối tượng và tập ảnh âm không chứa đối tượng. Chuyển các ảnh này sang ảnh mức xám và tiến hành học dựa trên các ảnh này. Dữ liệu học càng nhiều thì nhận diện càng chính xác, bao gồm cả ảnh dương và âm, bù lại thì thời gian huấn luyện lâu hơn. Nếu vùng nghi ngờ được xác định đúng là đối tượng thì thuật toán trả về những khung hình chữ nhật (vì có thể có nhiều đối tượng trong 1 hình). Mô hình phân lớp phân tầng kết hợp với các đặc trưng Haar-like được xây dựng nhằm rút ngắn thời gian xử lý, giảm thiểu tỷ lệ nhận dạng sai của bộ nhận dạng (bạn đọc quan tâm xin tham khảo thêm trong tài liệu của P.-A. Viola và M-J. Jones (2001).

**3.3 Giải thuật máy học Support Vector Machines (SVM)**

SVM (C. Corinna and V. Vapnik, 1995) đã được áp dụng thành công trong rất nhiều ứng dụng như nhận dạng vân tay, nhận dạng mặt người, phân loại bệnh ung thư... Trong nghiên cứu này, chúng ta quan tâm tới SVM là một phương pháp phân lớp tuyến tính (linear classifier), với mục đích xác định một siêu phẳng (hyperplane) tối ưu để phân tách hai lớp của dữ liệu nghĩa là cho phép chia các điểm (phần tử) dữ liệu thành hai phần sao cho các điểm cùng một lớp nằm về một phía của siêu phẳng này. Đối với tập dữ liệu của bài toán có nhiều lớp chúng ta có thể đưa về bài toán 2 lớp bằng kỹ thuật 1-tất cả

(one vs rest) hoặc 1-1 (one vs one). Ví dụ, phân lớp nhị phân tuyến tính, cho m phần tử (điểm) dữ liệu trong không gian n chiều:  $\{(x_i, y_i)\}$ ,  $i=\{1..m\}$ , với các thuộc tính  $x_i \in R^n$  và nhãn (lớp)  $y_i \in \{1,-1\}$ . Sau quá trình tối ưu để tìm ra các tham số (các véc tơ hỗ trợ - support vectors (với số lượng  $N_{sv}$ ) và các nhân tử Lagrange  $\alpha_i$ ), SVM phân lớp phần tử dữ liệu mới đến x như sau:

$$predict(x) = sign \left( \sum_{i=1}^{N_{sv}} \alpha_i y_i x_i x - b \right)$$

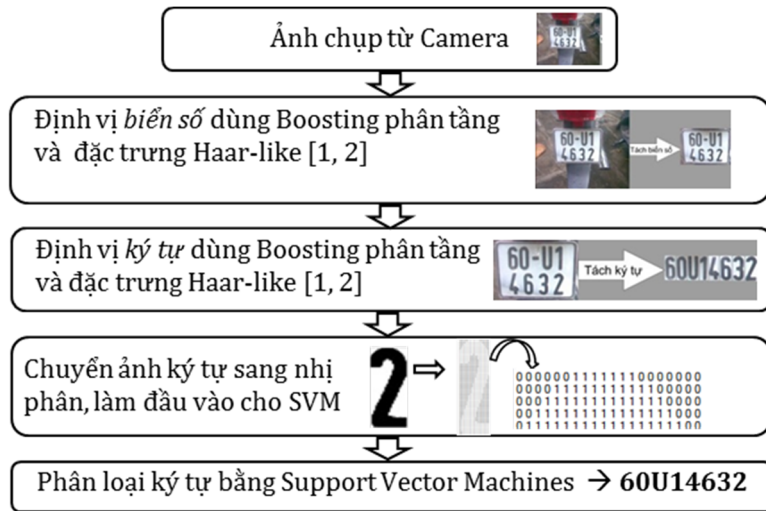
Trong đó:

$$b = \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} (w \cdot x_i - y_i)$$

$$w = \sum_i \alpha_i y_i x_i.$$

**4 ĐỀ XUẤT PHƯƠNG PHÁP NHẬN DẠNG VÀ RÚT TRÍCH KÝ TỰ TRONG BIÊN SỐ XE**

Chúng tôi đề xuất quy trình nhận dạng và rút trích biên số qua các bước chính như mô tả trong Hình 5. Ở đó, sau khi hệ thống nhận được ảnh chụp từ camera, hệ thống sẽ định vị và trích ra vùng chứa biên số. Từ vùng chứa biên số này, thực hiện tương tự để trích ra ảnh các ký tự có trong vùng biên số đó. Các ảnh ký tự này sau đó được chuyển sang dạng nhị phân để làm đầu vào cho bộ phân lớp SVM. Sau cùng, giải thuật SVM sẽ phân loại các giá trị nhị phân này thuộc vào các ký tự tương ứng nào (0..9, A..Z).



Hình 5: Quy trình nhận dạng và phân lớp biển số xe

Sau đây, chúng tôi sẽ mô tả chi tiết cách thực hiện cho mỗi bước trong hình trên.

**4.1 Định vị vùng biển số xe**

Giai đoạn này sử dụng đặc trưng Haar-like và Boosting phân tầng (mỗi bộ phân lớp trong Hình 3 là một mô hình Boosting với bộ phân lớp yếu là Cây quyết định) để định vị vùng biển số xe, sau đó trích ảnh biển số ra như minh họa trong Hình 6. Giải thuật định vị này có sẵn trong thư viện EmguCV<sup>1</sup>, vấn đề ta cần thực hiện là tiền và hậu xử lý ảnh để có được vùng biển số mong muốn. Chi tiết về cách thực hiện được trình bày qua các bước dưới đây.



Hình 6: Định vị và trích ảnh biển số xe

**Bước 1:** Chuẩn bị tập dữ liệu huấn luyện gồm 2250 ảnh màu. Trong đó, 750 ảnh chứa biển số xe (tập ảnh này là tập ảnh dương - positive) và 1500 ảnh không chứa biển số xe (hình bất kỳ như phong cảnh, vật dụng,... Tập ảnh này còn gọi là tập ảnh âm (negative) hay ảnh nền (background)). Lưu ý: Hai tập ảnh này nên cùng kích thước (trong nghiên cứu này là 640 x 480 pixels) và nên xoay ảnh theo nhiều góc để vét cạn các trường hợp thực tế.

**Bước 2:** Xác định tọa độ của các vùng chứa biển số xe, sau đó lưu toàn bộ chúng vào tập tin (ví dụ "location.txt") với nội dung:

```
<Đường dẫn>/<tên ảnh> <số biển số xe> x11
y11 w11 h11 x12 y12 w12 h12
...
<Đường dẫn>/<tên ảnh> <số biển số xe> x21
y21 w21 h21
```

Với: x11, y11 là tọa độ biển số xe; w11, h11 là chiều rộng và chiều cao biển số như minh họa trong Hình 7.

Ví dụ:

```
D:/bienso/1.jpg 1 132 112 303 216
D:/bienso/2.jpg 1 164 122 288 209
```

Ở giai đoạn này, chúng tôi đã phát triển một bộ công cụ để đọc ảnh, click-and-drag để chọn vùng biển số và xác định tọa độ đối tượng (sau đó lưu lại vào file location.txt như đã trình bày trên).



Hình 7: Xác định tọa độ vùng chứa biển số xe

<sup>1</sup> <http://www.emgu.com/>



**Bước 3:** Tạo các mẫu dùng cho huấn luyện (sử dụng chương trình `opencv_createsamples`<sup>2</sup> trong thư viện EmguCV), lưu lại vào tập tin vector dùng để huấn luyện ở bước tiếp theo. Cú pháp:

```
opencv_createsamples -vec vec.vec -info
D:\bienso\location.txt -num 750 -w 40 -h 30
```

Trong đó:

- `vec vec.vec`: file vector output được lưu với tên “vec.vec”. Đây là tập tin vector chứa ảnh thumbnail với kích cỡ được chỉ ra trong tham số `width` và `height` của tập tin `location.txt`. Dựa vào file này mà chương trình sẽ tìm ra các đặc trưng haar-like của đối tượng, từ đó tạo bộ nhận diện dựa trên đặc trưng này.

- `info D:\bienso\location.txt`: tập tin chứa tọa độ biên số đã tạo ở Bước 1.

- `num 750`: số ảnh dương (biển số) là 750.

- `w 40 -h 30`: tham số xác định kích thước ảnh tối thiểu. Nếu kích thước biển số nhỏ hơn 40x30 thì chương trình sẽ cho rằng không có biển số trong ảnh.

**Bước 4:** Tạo tập tin chỉ mục các ảnh không chứa biển số xe (ví dụ `bg.txt`), tập tin này đơn giản là chứa các đường dẫn và tên ảnh background (ảnh âm – ảnh bất kỳ không phải hoặc không chứa biển số xe và có cùng kích thước 640 x 480 pixels với ảnh chứa biển số xe), ví dụ:

```
<Đường dẫn>/<tên ảnh>
...
<Đường dẫn>/<tên ảnh>
```

**Bước 5:** Huấn luyện mô hình. Sử dụng chương trình `opencv_haartraining` (đi kèm với EmguCV) để huấn luyện, lưu tập tin mô hình với tên `bienso.xml`. Cú pháp như sau:

```
opencv_haartraining -data D:\bienso -vec
vec.vec -D:\bienso\bg.txt -npos 750 -nneg 1500 -
nstages 20 -mem 2000 -w 40 -h 30 -nonsym -
minhitrate 0.995 -maxfalsealarm 0.5
```

Trong đó:

- `data D:\bienso`: đường dẫn thư mục chứa các file thông số ngõ ra được cập nhật của tầng huấn luyện (cascade). Khi huấn luyện chương trình

sẽ tạo ra các thư mục từ 0 ⇒ `nstages`. Khi huấn luyện xong sẽ tạo ra file `bienso.xml` chứa mô hình để sử dụng về sau.

- `vec vec.vec`: chứa đường dẫn đến file vector đã tạo ra ở Bước 3.

- `bg D:\bienso\bg.txt`: đường dẫn đến file chứa danh sách các ảnh nền (ảnh âm) đã tạo ở Bước 4.

- `npos 750`: số lượng ảnh dương (biển số).

- `nneg 1500`: số lượng ảnh âm (ảnh nền).

- `nstages 20`: số lượng giai đoạn (stage) huấn luyện.

- `mem 2000`: dung lượng bộ nhớ RAM cần cho quá trình huấn luyện (MB).

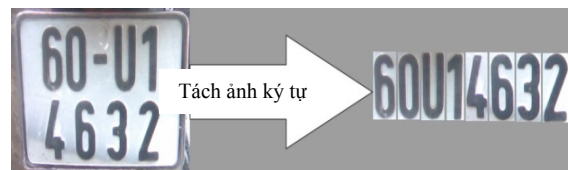
- `nonsym`: khai báo các đối tượng huấn luyện là không có tính đối xứng.

- `minhitrate 0.995`: là tỉ lệ dự đoán đúng tối thiểu.

- `maxfalsealarm 0.5`: tỉ lệ dự đoán sai tối đa.

## 4.2 Định vị ký tự

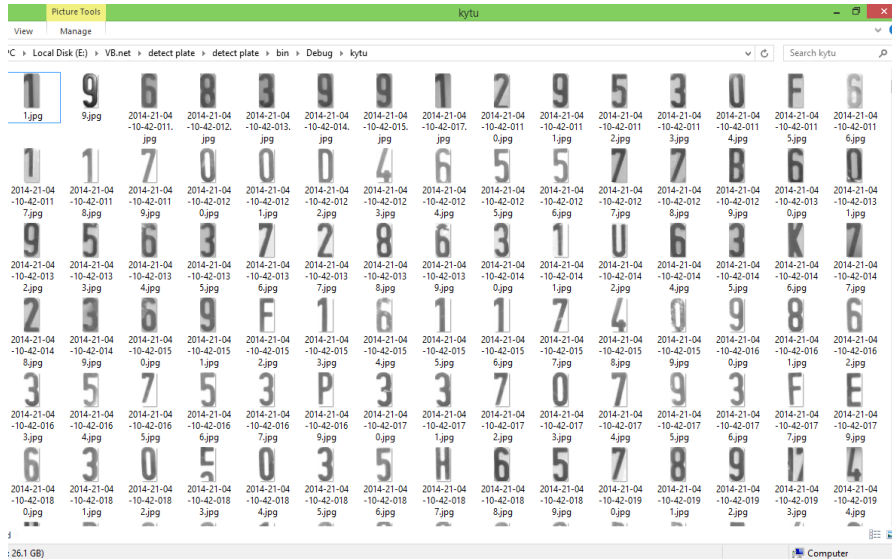
Hoàn toàn tương tự như ở giai đoạn định vị biển số, ta thực hiện việc định vị và trích ảnh các ký tự trong biển số xe như minh họa ở Hình 8 (lưu ý: sau bước này ta mới chỉ nhận được các ký tự dưới dạng ảnh).



**Hình 8: Định vị và trích ảnh ký tự trong biển số xe**

Chúng tôi sử dụng dữ liệu huấn luyện trong giai đoạn này gồm 5200 ảnh màu. Trong đó, 1500 ảnh chứa ảnh ký tự biển số xe và 3700 ảnh không chứa ký tự biển số xe (hình phong cảnh, vật dụng...). Các bước thực hiện hoàn toàn tương tự ở giai đoạn định vị biển số. Sau khi định vị được ảnh ký tự, tách và chuyển chúng về kích thước 20x48 pixels như minh họa trong Hình 9. Với mỗi ảnh ký tự thu được, chúng tôi đề xuất chuyển chúng về giá trị nhị phân để làm tập dữ liệu đầu vào cho bộ phân lớp SVM.

<sup>2</sup> [http://docs.opencv.org/doc/user\\_guide/ug\\_traincascade.html](http://docs.opencv.org/doc/user_guide/ug_traincascade.html)



Hình 9: Ảnh ký tự sau khi tách

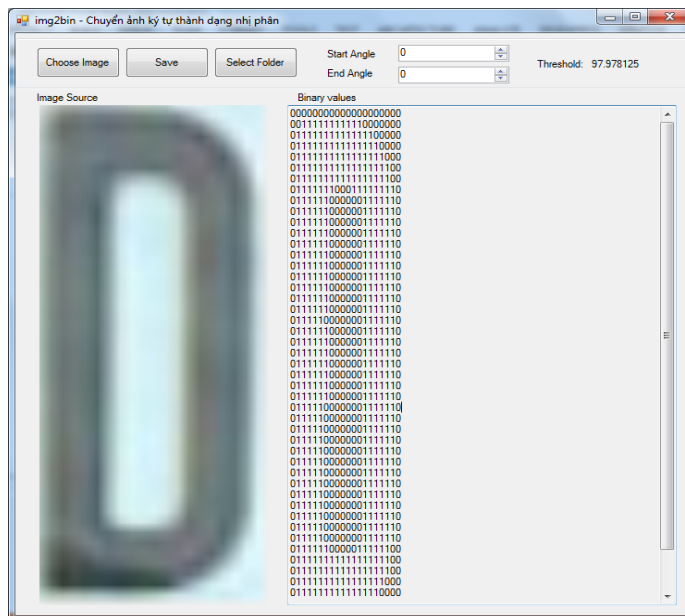
### 4.3 Chuyển ảnh ký tự sang dạng nhị phân

Ở bước này, chúng tôi chuyển các ảnh ký tự (20x48) thành giá trị nhị phân biểu diễn tương ứng cho ảnh, như minh họa trong Hình 11. Giải thuật được chúng tôi sử dụng ở đây rất đơn giản là với mỗi điểm ảnh, nếu mức sáng (intensity) của nó lớn hơn một ngưỡng nào đó (ví dụ, trong nghiên cứu này chúng tôi sử dụng giá trị trung bình mức xám của mỗi ảnh để làm giá trị ngưỡng) thì sẽ chuyển thành 1, ngược lại là 0.

Tuy nhiên, trong thực tế sẽ có lúc những ký tự trong biển số không thẳng đứng mà nghiêng về một bên nào đó, do đó tập dữ liệu phải bao gồm nhiều góc xoay khác nhau. Qua khảo sát thực tế cho thấy rằng góc nghiêng nằm trong khoảng từ  $-10^\circ$  đến  $10^\circ$  như trong Hình 10.



Hình 10: Các góc xoay của 1 ký tự



Hình 11: Chuyển ảnh ký tự thành ma trận nhị phân 20x48. Hình minh họa cho ảnh ký tự D, tuy nhiên ta có thể chuyển toàn bộ thư mục ảnh

Sau khi chuyển ảnh ký tự sang dạng nhị phân, tùy theo bộ phân lớp (classification) được sử dụng mà ta sẽ chuyển thành định dạng đầu vào tương ứng của bộ phân lớp đó. Trong nghiên cứu này, chúng tôi sử dụng LibSVM (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>) với định dạng dữ liệu đầu vào như sau:

```
<label><index1>:<value1><index2>:<value2>...
...
<label><index1>:<value1> <index2>:<value2>...
```

Ký tự	A	B	C	D	E	F	G	H	K	L	M	N	P	R	S	T	U	V	X	Y	Z
Số tương ứng	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Thực hiện tương tự cho tất cả các ảnh ký tự thu được ở phần 4.2 (minh họa trong Hình 9) ta thu được tập dữ liệu dùng để huấn luyện mô hình SVM.

**4.4 Xây dựng mô hình SVM**

Sử dụng bộ dữ liệu thu được ở phần 4.3 trước, huấn luyện mô hình SVM (có thể sử dụng công cụ LibSVM) với các tham số mặc định. Trong tương lai, chúng tôi sẽ cải tiến kết quả huấn luyện bằng cách tìm kiếm các siêu tham số (hyper-parameters) tốt nhất, như giá trị C, hàm kernel, giá trị gamma,..

**4.5 Nhận dạng và phân loại cho một ảnh biển số xe mới đến**

Khi có một ảnh biển số xe mới đến, ta sẽ sử dụng các mô hình đã có để định vị biển số, định vị ký tự, chuyển ký tự về dạng nhị phân và sau cùng là dùng SVM để nhận dạng (phân loại) các ký tự có trong biển số này.

**5 XÂY DỰNG HỆ THỐNG**

Sau các bước huấn luyện các mô hình, công việc còn lại là xây dựng hệ thống hoàn chỉnh để tích hợp các mô hình này vào.

Tương tự như bất kỳ một hệ thống thông tin quản lý nào, các chức năng chính trong hệ thống này bao gồm:

Trong đó, cột đầu tiên là nhãn (lớp đích cần phân loại – target class), tiếp theo là các vị trí có giá trị. Ví dụ trong Hình 11, các số nhị phân kích thước 20x48 có 960 vị trí và chữ D được biểu diễn theo định dạng của LibSVM như sau: 13 23:1 24:1 ... 869:1 870:1

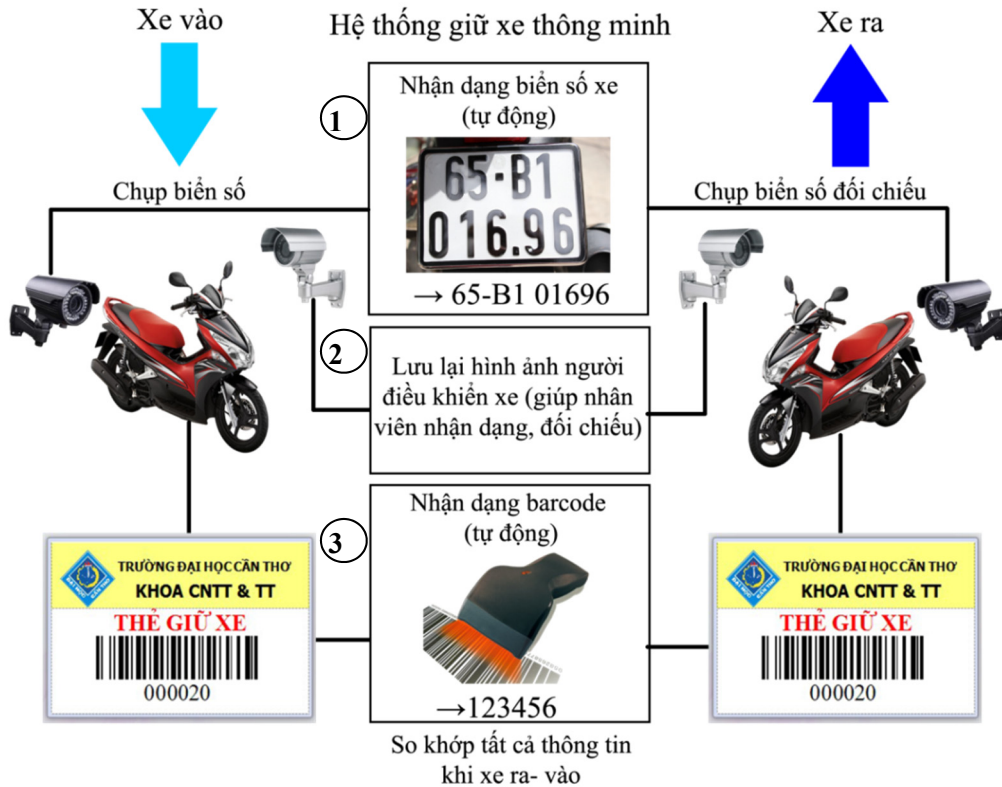
Với số 13 tương ứng ký tự D như quy ước trong bảng dưới đây:

- Quản lý xe vào/ra (lưu trữ hình ảnh biển số, khuôn mặt của người gửi xe, lưu trữ các ký tự trong biển số xe sau nhận dạng, thời gian vào/ra..).
- Quản lý thẻ xe (tạo thẻ xe kèm mã vạch, in thẻ, báo mất, báo hư,..).
- Quản lý thông tin nhân viên (gồm lịch trực).
- Quản lý tài khoản hệ thống (quyền, nhóm..).
- Tìm kiếm xe theo các thông tin như biển số xe, thời gian vào/ra...
- Thống kê lưu lượng xe ra/vào, doanh thu, thời gian làm việc của nhân viên theo khoảng thời gian...

**5.1 Sơ đồ tổng thể của hệ thống**

Sơ đồ tổng thể của hệ thống được biểu diễn trong Hình 12. Ở đó, khi xe vào sẽ dừng tại vạch quy định, lúc này nhân viên sẽ quét mã vạch trên thẻ giữ xe để lưu lại số trên thẻ. Cùng lúc đó, camera thứ nhất chụp ảnh gương mặt của người gửi xe, camera thứ 2 sẽ chụp ảnh biển số xe, cả 2 ảnh này đều được hệ thống lưu lại. Ảnh biển số xe sẽ được nhận dạng và chuyển thành các ký tự tương ứng. Như vậy, các thông tin quan trọng được hệ thống ghi lại là số thẻ xe, ảnh gương mặt người gửi xe, ảnh biển số xe, ký tự biển số, và thời gian vào/ra.





**Hình 12: Sơ đồ tổng thể của hệ thống**

Khi xe ra cũng dừng ngay vị trí quy định, nhân viên sẽ quét mã vạch trên thẻ nhận từ người gửi xe, hệ thống tự động truy xuất lại các thông tin khi xe vào để so sánh đối chiếu, nếu biển số giống nhau thì chương trình sẽ thông báo đúng và cho xe ra. Ngược lại sẽ cảnh báo để nhân viên so sánh đối chiếu các ảnh đã lưu.

Đối với xe đạp, do không có biển số nên thuộc tính này sẽ không được lưu lại. Các thông tin khác thì tương tự như trường hợp của xe máy.

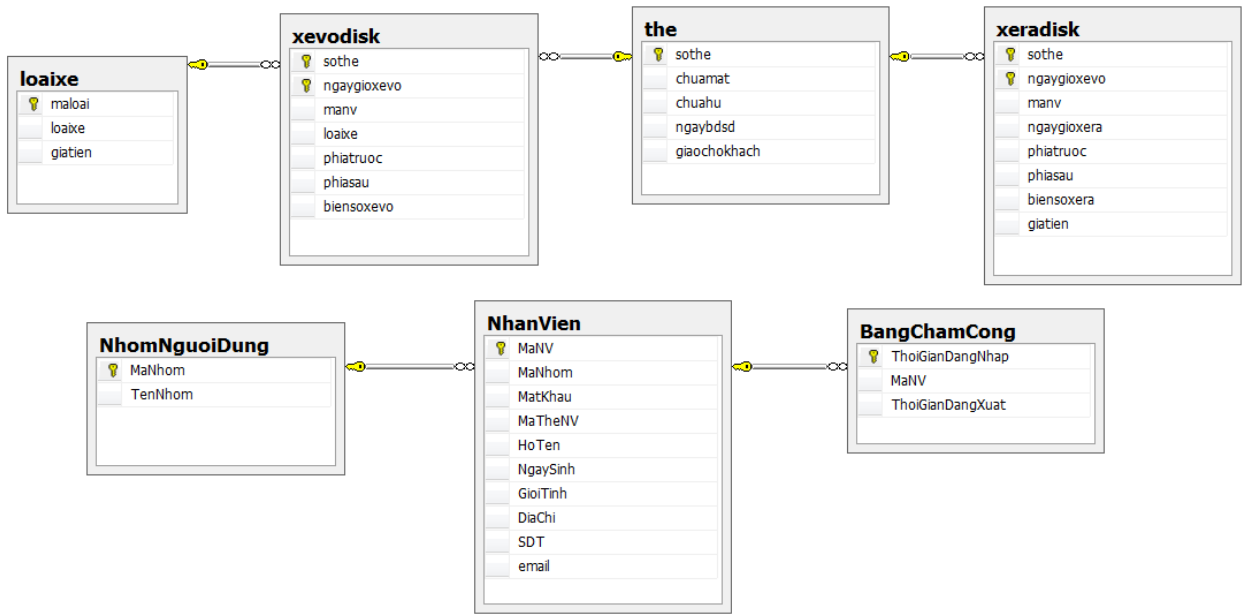
Cũng cần lưu ý rằng mô hình này là giải pháp bán tự động, do vậy vẫn phải có nhân viên giữ xe. Giải pháp này hỗ trợ tối đa cho nhân viên nhằm giảm sai sót trong nhận dạng bằng mắt qua tính năng nhận dạng biển số tự động và đưa ra các lời cảnh báo chứ không cứng nhắc trong việc “bắt lầm hay tha lầm”.

Ngoài ra, hệ thống không hoàn toàn phụ thuộc vào thẻ giữ xe barcode mà phải qua các đối chiếu

như đối chiếu tự động cho phần biển số (module ① trong Hình 12), đối chiếu qua quan sát hình ảnh (gương mặt, quần áo, hình biển số,.. – module ②) của người gửi xe (do nhân viên thực hiện) do đó việc in thẻ giả không phải là trở ngại lớn cho độ an toàn của hệ thống. Tuy nhiên, trong tương lai chúng tôi sẽ hướng đến sử dụng thẻ từ - công nghệ bảo mật tốt hơn nhưng giá thành cũng đắt đỏ hơn. So với các hệ thống hiện tại ở các siêu thị, bệnh viện,... (chỉ có module ②) thì hệ thống IPSS này hoàn toàn khả thi.

### 5.2 Lược đồ cơ sở dữ liệu (CSDL)

Do mỗi lượt xe vào/ra sẽ có 4 ảnh được lưu lại, chúng tôi chọn giải pháp chỉ lưu đường dẫn đến ảnh trong mà không lưu trực tiếp ảnh vào đó vào CSDL. Các bức ảnh sẽ được lưu riêng trên đĩa do đó kích thước CSDL cải thiện đáng kể trong khi tốc độ truy xuất cũng ở mức chấp nhận được. Lược đồ CSDL minh họa trong Hình 13.

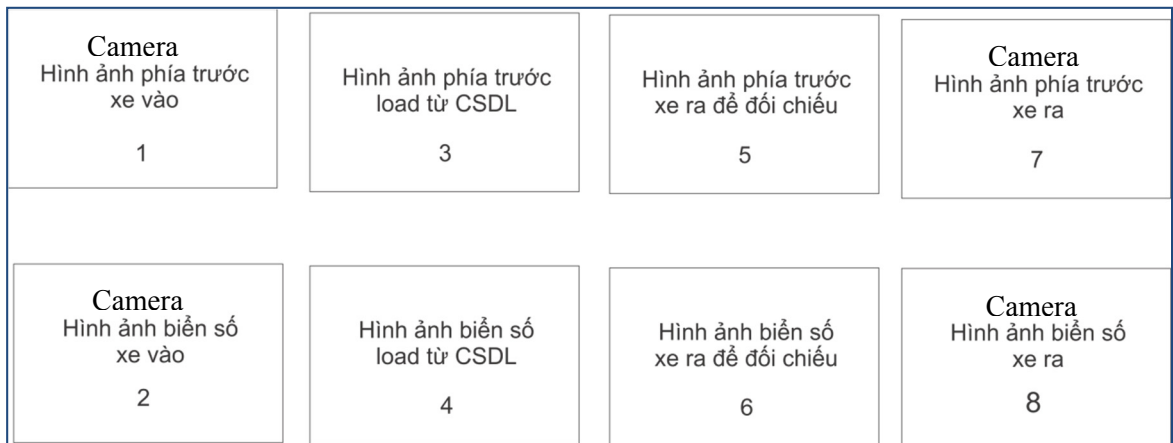


Hình 13: Lược đồ CSDL

**5.3 Thiết kế giao diện gửi/trả xe**

Chúng tôi chọn giải pháp 8 khung hình cho giao diện khi xe vào/ra như minh họa trong Hình 14. Trong đó, khung 1&2, khung 7&8 là hình ảnh trực tiếp từ camera. Khi xe vào, hình ảnh từ camera ở khung 1&2 sẽ được lưu lại và hiển thị trên khung

3&4. Khi xe ra, hệ thống sẽ load hình ảnh lúc xe vào và hiển thị ở khung 5&6. Giao diện này phù hợp với trường hợp một cổng vào ra. Nếu triển khai hệ thống dưới dạng một cổng vào và một cổng ra độc lập nhau, giao diện này có thể tùy biến lại còn 4 khung hình cho phù hợp.

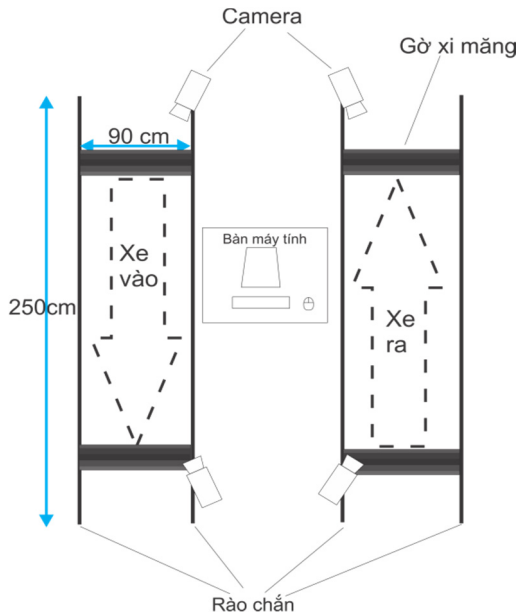


Hình 14: Giao diện chính của phần quản lý xe vào/ra

**5.4 Cài đặt và triển khai hệ thống**

Hệ thống được xây dựng trên nền tảng .NET (VB.NET), hệ quản trị cơ sở dữ liệu SQL Server, thư viện EmguCV và LibSVM.

Sơ đồ triển khai hệ thống minh họa trong Hình 15. Trong đó, các thiết bị cần thiết gồm: 01 bộ máy tính (màn hình càng lớn càng tốt, ít nhất là 19”), 01 máy quét mã vạch, 04 camera (hay webcam độ phân giải cao), các rào chắn và thiết bị bảo vệ khác.



Hình 15: Sơ đồ triển khai hệ thống

## 6 KẾT QUẢ ĐẠT ĐƯỢC

### 6.1 Độ chính xác của các mô hình

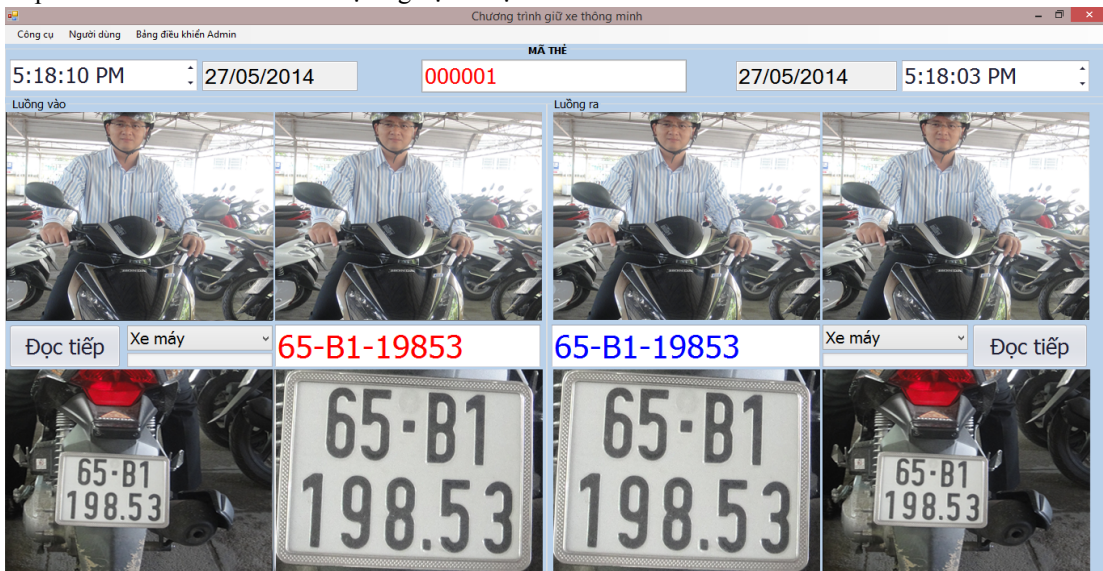
Trong giai đoạn định vị vùng biển số, chúng tôi sử dụng 750 ảnh biển số cho huấn luyện và 243 ảnh cho kiểm thử. Kết quả độ chính xác đạt 99%. Trong giai đoạn định vị ký tự, 11866 ảnh đã được dùng để huấn luyện và 4755 ảnh được dùng để kiểm thử, kết quả độ chính xác đạt 95.88%. Giai đoạn phân loại ký tự bằng SVM đạt độ chính xác 98.99% khi dùng 2603 phần tử cho huấn luyện và 1550 phần tử cho kiểm thử. Các thực nghiệm được

thực hiện theo nghi thức kiểm tra chéo 3 đường (3-fold cross validation).

Ngoài ra, sau khi xây dựng hệ thống xong chúng tôi cũng kiểm tra lại độ chính xác toàn bộ quy trình nhận dạng của hệ thống (gồm định vị và nhận dạng/phân lớp biển số) trên 215 ảnh biển số mới. Kết quả cho thấy tỉ lệ nhận dạng đúng hoàn toàn (định vị và phân lớp) đạt 190/215 ảnh (tương đương 88.37%) ở lần chụp thứ nhất của camera. Có 25/215 (11.62%) biển số được định vị đúng nhưng phân lớp sai 1 ký tự (đa phần là sai do điều kiện khách quan như biển số quá cũ do vậy ký tự trên biển số quá mờ hoặc bị chói ánh sáng, hoặc một vài trường hợp các ký tự có hình thái gần giống nhau như H và N; E và F). Để khắc phục tình trạng trên, trên form giao diện hệ thống (như minh họa trong Hình 16) chúng tôi cài đặt chức năng nhận dạng lại thông qua việc ấn phím nóng → hoặc click vào nút “Đọc tiếp” để cho phép người dùng kích hoạt lại việc chụp và nhận dạng biển số. Kết quả cho thấy 25 ảnh bị nhận dạng sai ở lần chụp thứ nhất đã giảm xuống còn 10 ảnh ở lần 2 và không còn ảnh nào bị sai ở lần chụp thứ 3. Như vậy, trong thực tế, đối với những biển số bị nhận dạng sai, người dùng sẽ ấn phím → để hệ thống chụp và nhận dạng lại (thao tác ấn phím này không quá 3 lần và chỉ mất khoảng 1 giây/lần do đó giải pháp này hoàn toàn khả thi).

### 6.2 Một số giao diện minh họa

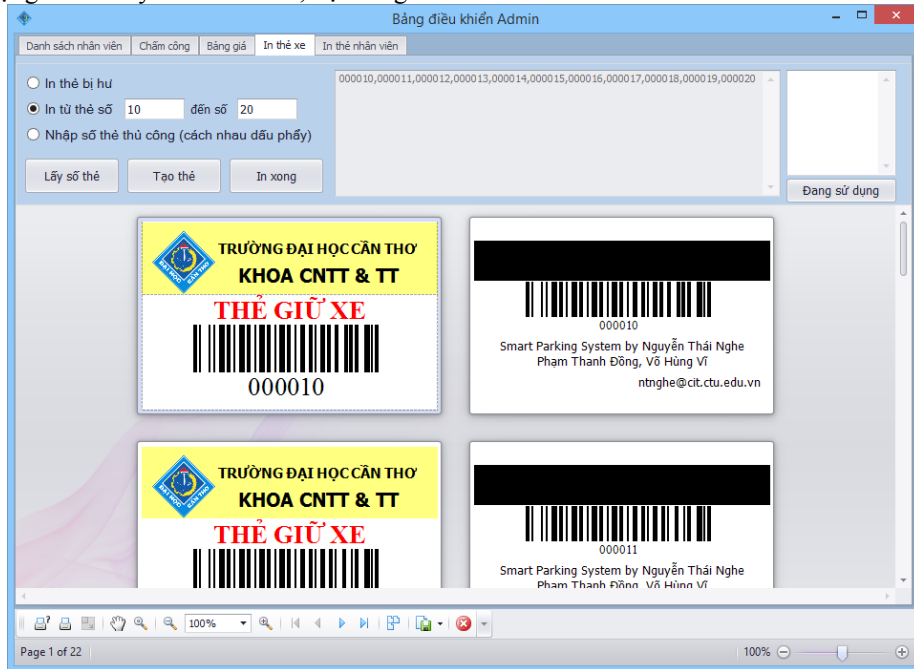
Giao diện chính khi xe vào ra được minh họa trong Hình 16, tương tự như đã trình bày trong phần thiết kế ở mục 5.3.



Hình 16: Giao diện gửi/trả xe

Giao diện minh họa cho phần quản lý thẻ xe kèm mã vạch được trình bày trong Hình 17. Ở đó, người sử dụng có thể tùy biến màu sắc, nội dung in

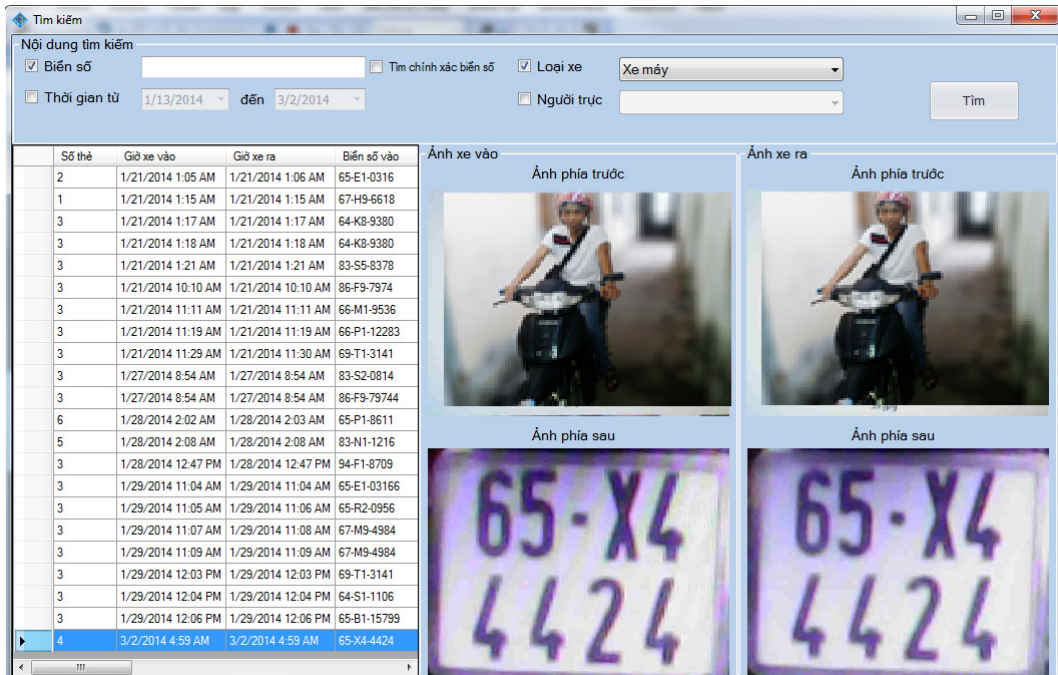
trên thẻ giữ xe cho phù hợp với từng đơn vị được triển khai.



Hình 17: Giao diện in thẻ xe với mã vạch

Giao diện minh họa cho phần tìm kiếm thông tin xe vào/ra theo tiêu chí nào đó (biển số, thời điểm,...) được trình bày như trong Hình 18. Thông

tin được truy xuất bao gồm thời điểm, biển số dạng ký tự, ảnh người gửi, ảnh biển số do vậy rất tiện lợi trong việc quản lý.



Hình 18: Giao diện Tìm thông tin xe vào/ra



Ngoài ra, hệ thống còn hỗ trợ nhiều tính năng khác tương ứng với 2 nhóm người dùng chính là admin và nhân viên.

– Nhóm nhân viên có các quyền như quản lý xe ra - vào, báo mất thẻ, báo hư thẻ, quản lý mật khẩu đăng nhập, tùy chọn camera, tùy chọn thư mục lưu ảnh, tìm kiếm xe vào ra theo các tiêu chí, thống kê...

– Nhóm admin có tất cả quyền của nhân viên và thêm các quyền như: cập nhật thông tin nhân viên, chấm công, quản lý giá tiền gửi cho từng loại xe, in thẻ xe...

## 7 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi đã đề xuất giải pháp xây dựng hệ thống hỗ trợ giữ xe thông minh IPSS. Đây là hệ thống kết hợp cả ba kỹ thuật nhận dạng nhằm tăng độ an toàn tối đa trong quản lý bãi xe, gồm: Nhận dạng biển số, nhận dạng mã vạch, và nhận dạng bán tự động qua các camera quan sát. Ngoài ra, IPSS còn hỗ trợ nhiều tính năng khác như in thẻ giữ xe (kèm mã vạch), tìm kiếm, thống kê, quản lý lịch trực,... IPSS được xây dựng trên nền tảng .NET + SQL Server, thư viện EmguCV (OpenCV) + LIBSVM.

Nghiên cứu sẽ tiếp tục được hoàn thiện thông qua việc cải thiện độ chính xác của mô hình định vị ký tự. Hiện tại, do bộ dữ liệu huấn luyện còn chưa đủ lớn nên một số ký tự gần giống nhau vẫn còn bị nhận dạng sai, như E và F, N và H, 1 và 7.

## LỜI CẢM Ạ

Nghiên cứu này được tài trợ bởi đề tài NCKH cấp Trường ĐHTC mã số T2014-08 và Phòng Xử lý dữ liệu thông minh – Khoa CNTT&TT, ĐHTC. Nhóm tác giả chân thành cảm ơn các góp ý của TS. Phạm Nguyễn Khang.

## TÀI LIỆU THAM KHẢO

1. Paul A. Viola and Michael J. Jones, 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition, pages 511-518.
2. Paul A. Viola and Michael J. Jones, 2004. Robust real-time face detection. International Journal of Computer Vision, pages 137-154.
3. Freund, Y. and Schapire, R., 1995. A decision-theoretic generalization of on-line

- learning and an application to boosting. Computational Learning Theory, pp. 23-37.
4. Q-N. Tran, T-N. Do, F. Poulet, N-K. Pham, 2008. Vehicle license plate classification. in proc. The National conference in computer science, pp. 79-85.
5. V. Kamat and S. Ganesan, 1995. An efficient implementation of the Hough transform for detecting vehicle license plates using dsp's. In RTAS '95: Proceedings of the Real-Time Technology and Applications Symposium, page 58, Washington, DC, USA. IEEE Computer Society.
6. J-W Hsieh, S-H Yu, and Y-S Chen, 2002. Morphology-based license plate detection from complex scenes. In Proceedings of the International Conference on Pattern Recognition, 3:30-176, IEEE Press, Washington, USA.
7. K. I. Kim, K. Jung, and J. H. Kim, 2002. Color texture-based object detection: An application to license plate localization. In proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines, Springer-Verlag, London, UK, pp. 293-309.
8. L. Dlagnekov, 2005. Car license plate, make, and model recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA.
9. H. Mahini, S. Kasaei, F. Dorri, and F. Dorri, 2006. An efficient features – based license plate localization method. In ICPR (2), pages 841–844. IEEE Computer Society.
10. C. Schmid and R. Mohr, 1997. Local gray value invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5): 530–535.
11. D-G. Lowe, 1999. Object Recognition from Local Scale Invariant Features. In Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, pp. 1150–1157.
12. D-G. Lowe, 2004. Distinctive Image Features from Scale Invariant Keypoints. In International Journal of Computer Vision, pp. 91–110.



13. M. Turk and A. Pentland, 1991. Face recognition using eigenfaces, IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591.
14. Schneiderman H. and Kanade T., 2000. A statistical method for 3D object detection applied to faces and car, In International Conference on Computer Vision.
15. Cortes, Corinna and Vladimir Vapnik, 1995. Support-Vector Networks. Machine Learning 20: 273–297.
16. Đỗ Thanh Nghị, 2011. Khai mô dữ liệu - Minh hoạ bằng ngôn ngữ R. Nhà xuất bản Đại học Cần Thơ.