

## Giải Pháp Đọc Chỉ Số Công Tơ Từ Ảnh Số

### *A Solution For Automatic Electrical Metter Reading From Images*

**Lê Ngọc Sơn**  
Trưởng Đại học Bách Khoa Hà Nội  
e-mail: [ngocson9225@gmail.com](mailto:ngocson9225@gmail.com)

**Trần Hoài Linh**  
Trưởng Đại học Bách Khoa Hà Nội  
e-mail: [linh.tranhoai@hust.edu.vn](mailto:linh.tranhoai@hust.edu.vn)

#### Tóm tắt

Hiện nay một số lượng lớn công tơ đang sử dụng vẫn là công tơ thế hệ cũ nên không truy xuất được số liệu từ xa mà vẫn phải sử dụng các nhân công đi thu thập thủ công. Các nhân công sẽ chụp ảnh bề mặt công tơ và sau đó tự nhập các chỉ số công tơ vào cơ sở dữ liệu. Để cải thiện thời gian cần thiết để xử lý một công tơ, ta thấy cần có nhu cầu xây dựng và lập trình các chương trình tự động xử lý ảnh để trích xuất các chỉ số. Một trong những ưu điểm của thuật toán là có khả năng đáp ứng thời gian thực, nhận dạng chỉ số công tơ trong nhiều điều kiện ánh sáng, bụi bẩn và góc quay. Các thuật toán được thử nghiệm trên máy tính bảng hiệu Asus với các hình ảnh được thu thập chính bằng camera của máy và đã đạt được độ chính xác cao.

**Từ khóa:** công tơ điện tử, đọc chỉ số, nhận dạng, xử lý ảnh, mạng nơ-ron.

#### Abstract:

Nowadays, many electric meters are still of old generation, which cannot be accessed from distance. In order to read their indications, we need to use man power. The readers would take pictures of the meters' surfaces, read the numbers from the images and manually input these numbers into the databases. To improve the time, we need a solution of automatic image processing to read out the numbers. The solution has a number of advantages: real-time response, automatic handling of different lighting conditions, different angles of images,... The proposed solutions were successfully implemented in a Asus table to process the images collected by the tablet's built-in camera with high accuracies.

**Keywords:** Electronic metter, meter reading, pattern recognition, image processing, neural network.

#### Chữ viết tắt

MLP	Multilayer perceptron
OCR	Optical Character Recognition
SVM	Support vector machine

### 1. Giới thiệu chung

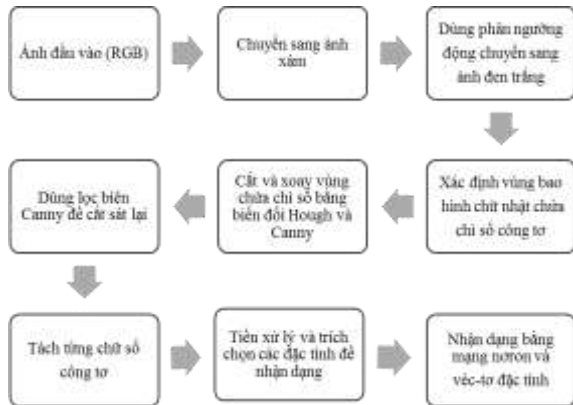
Hiện nay một số lượng lớn các công tơ sử dụng tại Việt Nam vẫn là loại công tơ cơ cổ điển, không tương tác được với các hệ thống thu thập và quản lý hiện đại một cách tự động. Việc đọc và ghi lại số liệu trên công tơ này hiện tại còn thô sơ, tốn nhiều thời gian và công sức. Hơn nữa việc ghi lại số liệu còn mang tính chủ quan nên nhiều khi không chính xác, làm ảnh hưởng đến uy tín của nhà cung cấp điện và gây thiệt hại về kinh tế cho người dân. Vì thế nhóm tác giả

muốn đề xuất một giải pháp đọc ghi chỉ số tự động. Giải pháp cho phép sử dụng camera của thiết bị sẵn có (máy tính bảng, điện thoại) để chụp ảnh mặt công tơ hoặc có thể thiết kế riêng một phần cứng có thể sử dụng hỗ trợ cơ khí để người công nhân không phải trèo lên. Khi người ghi công tơ cài đặt ứng dụng này trên điện thoại của mình di chuyển đến các cột công tơ, dựa theo vị trí GPS phần mềm sẽ tự động hiển thị vị trí các cột công tơ gần đó kèm theo tên chủ hộ. Người đi ghi công tơ sẽ chọn công tơ và chụp hình mặt công tơ đó. Ảnh gốc chụp về sẽ được lưu trữ cùng với các thông tin về thời gian địa điểm (theo GPS), đồng thời phần mềm hỗ trợ nhập tự động chỉ số công tơ vào cơ sở dữ liệu trên máy điện thoại dựa trên thuật toán xử lý ảnh tích hợp trên đó. Cuối ca làm việc, người dùng có thể kết nối với một máy tính chủ qua Wifi để tải các ảnh chụp và các kết quả ghi công tơ tự động lên cơ sở dữ liệu ở trên máy chủ. Các thông tin này sẽ được sử dụng để nâng cao chất lượng quản lý và chăm sóc khách hàng.

### 2. Phân tích giải pháp

Một trong những vấn đề quan trọng trong giải pháp là xây dựng được phần thuật toán xử lý ảnh. Thuật toán đã được nhóm tác giả chia làm hai bước chính: bước đầu tiên là từ ảnh đầu vào xác định vùng chứa chỉ số công tơ, bước thứ hai là tách từng chữ số trong ảnh đầu vào trích chọn vecto đặc tính sau đó đưa vào mô hình nhận dạng. Ở bước đầu tiên thuật toán nhận thấy vùng chứa chỉ số công tơ là một vùng hình chữ nhật có một số đặc điểm cố định nhóm tác giả đã tiến hành một số giải pháp: sử dụng trích chọn đặc trưng Haar-like và bộ lọc Adaboost một cách tương tự như tài liệu tham khảo [8]; sử dụng biến đổi Hough để xác định hai cặp đường thẳng song song là biên của vùng chứa chỉ số như tài liệu tham khảo [9]. Tuy nhiên ở phương pháp thứ nhất kết quả thu được không tốt do vùng chứa chỉ số công tơ thường bám bụi nhiều, phương pháp thứ hai tốc độ chạy khá chậm và không đáp ứng được yêu cầu về thời gian thực. Vì thế nhóm tác giả đã đề xuất giải pháp loại bỏ được cả hai vấn đề trên. Phương pháp sử dụng thuật toán phân ngưỡng động (*Adaptive threshold*) [11] để nhị phân ảnh đầu vào nhằm làm giảm ảnh hưởng của nhiễu và ánh sáng. Việc chọn các thông số mask để thực hiện thuật toán này được nhóm tác giả đã thử chọn quanh dải kích cỡ bằng 1/4 kích cỡ ảnh xám cần phân ngưỡng. Sau đó dựa vào đặc điểm hình thái của vùng chứa công tơ: diện tích, tỉ lệ chiều dài chiều rộng, tỉ lệ giữa số pixel đen chia pixel trắng, số chữ số... để có thể xác định vùng đó có chứa chỉ số công tơ. Để có được

các giá trị về khoảng của các thông số nhóm tác giả đã tiến hành thu thập 600 ảnh mẫu công tơ ở các điều kiện ánh sáng, góc chụp khác nhau và tính toán ra các đặc điểm. Ở bước thứ hai ta sử dụng các phép chiếu, phép phân tích blob để tách riêng từng chữ số. Từ mỗi chữ số ta trích rút vectơ đặc tính đầu vào theo điểm crossing, hình chiếu, vị trí và theo từng vùng. Sau đó vectơ được đưa vào mô hình mạng nơron truyền thẳng (MLP). Mô hình tổng quát của thuật toán mô tả như trên H. 1.



H. 1 Sơ đồ khối tổng thể các bước chính trong giải pháp đọc tự động các chỉ số công tơ từ ảnh số

### 3. Các thiết kế chi tiết của thuật toán

#### 3.1. Xác định vùng chứa chỉ số công tơ

##### 1. Thu nhận và phân ngưỡng ảnh thu được

Sau khi thu được ảnh màu RGB từ camera ta tiến hành chuyển ảnh RGB sang ảnh xám rồi chuyển ảnh đen trắng bằng phân ngưỡng động (Adaptive threshold) như tài liệu tham khảo [11] (nhóm tác giả cũng đã sử dụng phương pháp trong bài báo này để tăng tốc độ tính toán) với mask 51x51 và thu được kết quả như sau:



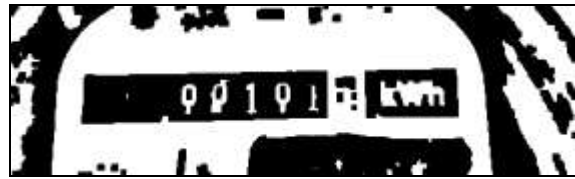
H. 2 Kết quả phân ngưỡng với phân ngưỡng động

Do công tơ có mặt kính nên dẫn đến độ sáng của bề mặt công tơ không đồng đều nên phân ngưỡng toàn cục (Ostu, Histogram hay phép phân ngưỡng trực tiếp từ ảnh màu thông qua mô hình HSI như trong [10]) đều cho kết quả không tốt. Trái lại trong những trường hợp này phân ngưỡng động lại cho kết quả khá tốt bởi lẽ nó chỉ phụ thuộc vào độ tương phản của ảnh. Chính vì thế phân ngưỡng động được quyết định chọn để sử dụng cho bài toán này.

##### 2. Tìm vùng chứa chỉ số công tơ

Trước khi thực hiện việc tìm chỉ số công tơ vùng chứa chỉ số được làm nổi bật và cô lập với cái vùng khác bằng các phép co ảnh vào dẫn ảnh với mask 5x5.

Kết quả đạt được sau khi thực hiện phép co và dẫn ảnh [5,6]:



H. 3 Kết quả sau khi co ảnh và dẫn ảnh

Sau phép co ảnh và dẫn ảnh rõ ràng vùng chứa chỉ số công tơ đã nằm tách biệt với các vùng khác. Bước tiếp theo ta sẽ dựa vào đặc điểm hình học của vùng chứa chỉ số công tơ để tìm ra nó.

- Sử dụng thuật toán blob để đánh số các thành phần liên thông màu đen có trong ảnh.
- Với mỗi thành phần liên thông ta xét hình chữ nhật có hai cạnh song song với hai trục tọa độ bao quanh nó.
- Sau khi ta xác định được các hình chữ nhật, dựa vào hình thái của vùng chứa chỉ số công tơ ta sẽ duyệt và tìm ra vùng có khả năng chứa chỉ số công tơ dựa trên các tiêu chí đã được tính toán:
  - Diện tích hình chữ nhật bao >500 pixel
  - Số chữ số (số thành phần liên thông màu trắng) >4. Việc đếm số chữ số này được thực hiện bằng cách phân ngưỡng ảnh và đếm số thành phần liên thông màu trắng > ngưỡng K nào đó.
  - $1.5 < \text{Chiều dài} / \text{Chiều rộng} < 7$
  - Diện tích pixel đen / Tổng pixel trong ảnh > 0.3

Do điều kiện loại khá chặt nên sau bước này là đã tìm được vùng bao chứa chỉ số công tơ.

##### 3. Xoay lại vùng chứa chỉ số công tơ

Ở bước trên đã cắt thô ra được vùng chứa chỉ số công tơ xong các chữ số trong vùng này còn bị xoay nên gây khó khăn cho việc cắt sát ảnh và nhận dạng sau này. Vì thế bước này ta sẽ tiến hành xoay lại vùng chứa số công tơ cho ngay ngắn.

- Đầu tiên ta sử dụng phép lọc biên Canny với hai ngưỡng 25 và 50 để xác định các điểm biên trên ảnh.



H. 4 Vùng chứa chỉ số công tơ sau lọc Canny

- Để thấy các đường viền của vùng chứa chỉ số công tơ tạo thành các đường thẳng. Để xác định các đường thẳng này ta sẽ dùng biến đổi Hough với các thông số như sau:
  - Mọi đường thẳng sẽ bị bỏ qua nếu nó ít hơn <50 pixel trắng.

- Nếu khoảng cách từ 1 điểm tới đường thẳng đó  $< 5$  pixel có thể xem điểm đó thuộc đường thẳng.



H. 5 Đường thẳng tìm được sau phép biến đổi Hough

3. Sau khi xác định được các thẳng ta tính góc nghiêng của các đường thẳng so với trục Ox. Lấy trung bình góc nghiêng đó làm góc để xoay lại ảnh.

#### 4. Cắt sát vùng chứa chỉ số công tơ

Sau khi xoay lại ảnh ta sẽ tiến hành cắt sát vùng chứa chỉ số công tơ. Để thực hiện việc này trước hết ta sử dụng lọc biên Canny để dò biên của ảnh, tiếp theo ta xác định vị trí của hai cặp đường thẳng song song là biên vùng chứa chỉ số. Việc tiếp theo đơn giản là cắt lại vùng chứa chỉ số này.



H. 6 Vùng chứa chỉ số công tơ sau lọc Canny



H. 7 Vùng chứa chỉ số công tơ sau khi cắt sát

#### 5. Tách từng chữ số trong ảnh

1. Để tiến hành tách các chữ số trong ảnh ta chỉnh lại kích cỡ của vùng chứa chỉ số công tơ 40x200 sau đó tiến hành phân ngưỡng động với cửa số 11x11.



H. 8 Vùng chứa chỉ số công tơ sau khi lấy ngưỡng động

2. Ảnh sau khi phân ngưỡng còn rất nhiều ta cần loại bỏ nhiễu trong ảnh bằng cách: Ta sử dụng thuật toán blob để đánh số các thành phần liên thông trong ảnh, sau đó ta loại bỏ những thành phần liên thông  $< 30$  pixel và những thành phần nằm sát viền ảnh (cách viền ảnh  $\leq 3$  pixel).



H. 9 Ảnh chỉ số công tơ sau khi loại bỏ nhiễu

3. Xác định các hình chữ nhật bao các thành phần liên thông. Dựa vào đặc điểm hình học các chữ số để có thể tìm ra các vùng có khả năng là chữ số:
  - $30 < \text{Số pixel trắng} < 300$
  - Chiều rộng nhỏ  $< 20$  và chiều cao  $< 30$
  - Đỉnh trái trên có hoành độ  $x > 70$
  - Đường thẳng  $y = 20$  phải cắt hình chữ nhật
4. Sắp xếp lại các vùng có khả năng chứa số và chọn 5 vùng có diện tích lớn nhất ta được các vùng chứa số.



H. 10 Ảnh chọn ra các vùng có khả năng chứa chữ số

## 2.2. Nhận dạng với mạng nơron

Đây là giai đoạn cuối cùng trong khâu xử lý ảnh. Ta sẽ tiến hành tính toán vec-tơ đặc tính đầu vào để huấn luyện mạng nơron

### 1. Trích chọn các thành phần đặc trưng từ các số tìm được

Để tiến hành trích chọn các thành phần đặc trưng trong các số vừa tìm được ta phải tiến hành hiệu chỉnh và loại nhiễu các số vừa cắt được.

1. Chỉnh lại kích cỡ ảnh là 12x20. Đây là kích cỡ gần với tỉ lệ kích cỡ đối tượng thật sẽ mang lại nhiều thông tin cho quá trình trích chọn đặc tính. Đồng thời chiều dài và chiều rộng đều chia hết cho 4 thuận lợi cho việc trích chọn đặc tính theo vùng về sau. Tiếp đó nhóm tác giả phân ngưỡng các số vừa cắt với ma trận cửa số 9x9.
2. Loại các nhiễu trong ảnh là các thành phần liên thông gồm các pixel trắng không phải số bằng cách lấy thành phần liên thông lớn nhất trong ảnh.
3. Tiến hành đưa chữ số sát lên vùng góc trái trên của khung.



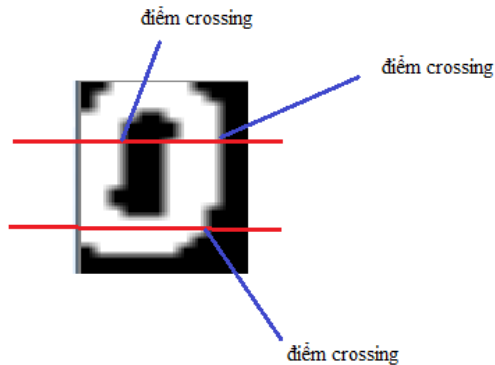
H. 11 Ví dụ ảnh chữ số sau khi loại nhiễu

Sau khi loại bỏ nhiễu trên ảnh các số ta tiến hành trích chọn đặc trưng trên ảnh:

1. Trích chọn đặc trưng theo vùng:
  - Chia ảnh thành 15 vùng mỗi vùng có kích cỡ là  $4 \times 4$ . Sau đó ta tính tổng các pixel trắng trong mỗi vùng và tính tổng pixel trắng trên toàn ảnh.
  - Ta có 16 thuộc tính cho việc trích chọn thuộc tính theo vùng. Tính tổng số pixel trắng theo từng dòng và từng cột ta có  $12 + 20$  thuộc tính nữa.
2. Trích chọn theo đặc trưng theo số điểm crossing: Ta định nghĩa điểm crossing là điểm mà tại đó có

sự chuyển vùng từ pixel trắng sang pixel đen theo hàng hoặc cột:

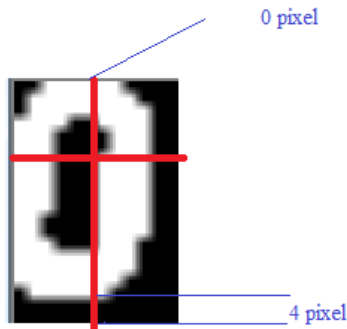
- Tính toán số điểm crossing theo dòng ta có 12 thuộc tính
- Tính toán số điểm crossing theo cột ta có 20 thuộc tính



H. 12 Trích chọn đặc trưng theo số điểm crossing

3. Trích chọn đặc trưng theo vị trí:

- Tính toán khoảng cách đến pixel trắng đầu tiên theo từng dòng hướng từ bên trái sang ta có 12 thuộc tính.
- Tính toán khoảng cách đến pixel trắng đầu tiên theo từng dòng hướng từ bên phải sang ta có 12 thuộc tính.
- Tính toán khoảng cách đến pixel trắng đầu tiên theo từng cột hướng từ bên trên xuống dưới ta có 20 thuộc tính.
- Tính toán khoảng cách đến pixel trắng đầu tiên theo từng dòng hướng từ từ bên dưới lên trên ta có 20 thuộc tính.



H. 13 Trích chọn đặc trưng theo vị trí

Sau quá trình trích chọn đặc trưng ta có 144 thuộc tính cho một ảnh đầu vào.

### 2. Huấn luyện với mạng nơron

Sau khi trích chọn được vector đặc tính bước cuối cùng của thuật toán là xây dựng được mô hình nhận dạng để từ vector đặc tính cho ta kết quả dưới dạng chữ số. Nhóm tác giả đã tiến hành thử nghiệm trên các mô hình máy học thường dùng với bài toán nhận dạng kí tự OCR: SVM, phân lớp Bayes, mạng nơron... với các cấu trúc, thông số khác nhau và đã chọn cấu trúc mạng nơron truyền thẳng MLP

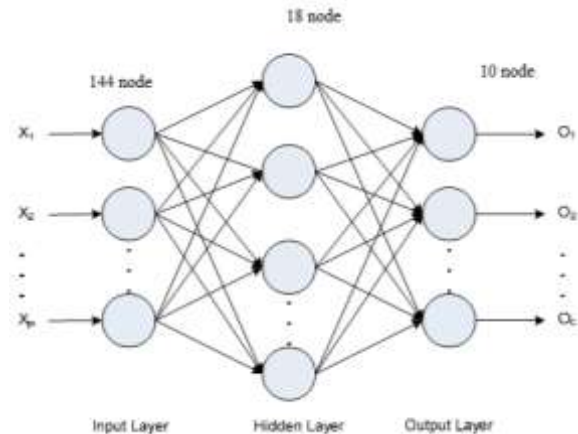
cho bài toán này. Đây là bài toán không quá phức tạp nên nhóm tác giả chọn mạng có 1 lớp ẩn.

Cấu trúc mạng nơron MLP, gồm có 3 lớp:

1. *Input Layer*: Số nơron là 144 tương ứng 144 đặc trưng đã chọn từ bước trên.
2. *Hidden Layer*: Thử bằng thực nghiệm rút ra được số nơron lớp này là 18.
3. *Output Layer*: Lớp này gồm 10 đầu ra tương ứng các số từ 0 đến 9.

Hàm kích hoạt nhóm tác giả sử dụng là hàm tansig [1]. Chương trình huấn luyện được viết trên nền tảng bộ thư viện mã nguồn mở OpenCV 2.4.9. Tập mẫu huấn luyện và tập kiểm tra, mỗi tập gồm 2000 mẫu. Mỗi chữ số ta lấy 200 mẫu. Quá trình huấn luyện mạng được thực hiện tối đa 10000 lần với hệ số học 0.01.

Để có thể tính toán số nơron lớp ẩn cho mạng nhóm tác giả đã thử chọn phương pháp thử tăng dần số nơron lớp ẩn [1] bắt đầu từ số nơron đầu ra và tăng dần lên cho đến khi kết quả test bắt đầu giảm xuống thì lấy giá trị trước đây là số nơron lớp ẩn cho cấu trúc mạng.



H. 14 Cấu trúc mạng nơron MLP sử dụng trong nhận dạng

## 4. Kết quả thực hiện

Để xây dựng mô hình nhận dạng bằng mạng nơron, đồ án đã thực hiện quá trình huấn luyện mạng nơron trên máy tính: sử dụng phần mềm Visual Studio 2012 và lập trình trên nền thư viện OpenCV 2.4.9.


Kết quả của quá trình huấn luyện được lưu trong file \*.xml. Để tìm được cấu trúc mạng nơron cũng như trích chọn được những đặc trưng phù hợp, nhóm tác giả đã phải thử trích chọn nhiều thuộc tính và huấn luyện trên nhiều cấu trúc mạng khác nhau và ghi nhận lại kết quả tối ưu.


Về độ chính xác của thuật toán xử lý ảnh, nhóm tác giả đã nghiên cứu và tìm ra những thuật toán hạn chế tối đa những điều kiện ảnh hưởng từ bên ngoài như: việc không sử dụng ngưỡng cố định cho việc phân ngưỡng, trích chọn những đặc tính ít nhạy nhiễu, đưa ảnh về một khung cố định... Cuối cùng chương trình đã được test thử trên 600 mẫu mới thu thập từ máy tính bảng ASUS MEMO HD7 chạy với hệ điều hành Android và thu được kết quả như sau:


Kết quả đúng	Xác định sai vùng chứa chỉ số công tơ	Không tách đủ chữ số trong vùng chứa chỉ số	Nhận dạng sai
568/600	1/600	8/600	23/600
94.67%	0.16%	1.33%	3.83%

Kết quả thuật toán nhận dạng đúng với tỉ lệ 94,67%. Trong số 568 mẫu được xử lý chính xác, chương trình đã nhận dạng đúng cả trong những trường hợp có nhiễu lớn hoặc có thay đổi lớn về góc chụp, ánh sáng khác nhau. Sau đây là một số ví dụ cho các trường hợp khó đã xử lý được:

1. Xử lý được những mẫu bị chụp không ngay ngắn, ảnh bị xoay lệch nhiều so với phương nằm ngang:

Ảnh vùng chụp 

Ảnh vùng cắt 

Các chữ số 

Kết quả số 00677

H. 15 Trường hợp ảnh bị nghiêng

2. Xử lý được những mẫu bị ngược sáng:

Ảnh vùng chụp 


Ảnh vùng cắt 


Các chữ số 


Kết quả số 00677

H. 16 Trường hợp ảnh bị chụp ngược sáng

3. Xử lý được những mẫu có các phần chữ số bị mất phần đầu hoặc cuối do đang dịch chuyển:

Ảnh vùng chụp 


Ảnh vùng cắt 


Các chữ số 


Kết quả số 00680

H. 17 Ảnh chụp các chữ số bị mất phần đầu

4. Xử lý được những mẫu bị bản:

Ảnh vùng chụp 


Ảnh vùng cắt 


Các chữ số 


Kết quả số 00680

H. 18 Mẫu nhận dạng bị bản

5. Xử lý được những mẫu chụp trong điều kiện ánh sáng tối:

Ảnh vùng chụp 

Ảnh vùng cắt 


Các chữ số 


Kết quả số 00680


H. 19 Ảnh chụp trong điều kiện cường độ ánh sáng thấp

Thuật toán còn nhận dạng lỗi trong một số trường hợp, ví dụ như sau:

6. Ánh sáng quá mạnh dẫn đến việc nhận dạng nhầm những chữ số giống nhau:

Ảnh vùng chụp 

Ảnh vùng cắt 

Các chữ số 

Kết quả số 00672

H. 20 Nhận dạng nhầm số 7 thành số 2 do xuất hiện nhiễu trong quá trình phân ngưỡng

## 5. Kết luận và hướng phát triển

Trong bài báo này, nhóm tác giả đã tìm hiểu, nghiên cứu và xây dựng thành công thuật toán hỗ trợ việc đọc ghi công tơ ứng dụng công nghệ xử lý ảnh. Sau khi hoàn thành thuật toán nhóm tác giả đã tiến hành triển khai hệ thống trên máy tính bảng Asus memo HD 7 với nền tảng Android, WebServer với máy chủ



Apache 2.4.10 và hệ quản trị cơ sở dữ liệu mySql. Hệ thống bước đầu đã hoạt động ổn định và cho những kết quả khả quan. Ứng dụng này thực sự có tính thực tiễn cao, nếu làm thành hệ thống hoàn chỉnh có thể áp dụng ngay để thay thế cho các phương pháp thủ công hiện có. Toàn bộ hệ thống nhóm tác giả xây dựng trên nền mã nguồn mở với cộng đồng người dùng lớn để giảm thiểu chi phí cũng như thuận tiện cho việc phát triển ứng dụng về sau. Trong tương lai nhóm tác giả sẽ tiếp tục cải thiện hệ thống cũng như nâng cao tính chính xác hệ thống bằng các giải pháp như:

- Tiếp tục thử nghiệm phát triển để thuật toán có thể thực hiện nhiều điều kiện ánh sáng khác nhau.
- Phát triển thuật toán nhận dạng kết hợp mô hình mạng nơron SOM và mạng MLP để nâng cao độ chính xác
- Phát triển trên bản OpenCV 3.x để có thể tận dụng bộ thư viện IPP trên đó nhằm tăng tốc độ thuật toán.
- Áp dụng thư viện Tesseract để cải thiện nhận dạng kí tự OCR chính xác.

#### Tài liệu tham khảo

- [1] Trần Hoài Linh, "Mạng nơron và ứng dụng trong xử lý tín hiệu", NXB Đại học Bách Khoa, 2014.
- [2] <http://docs.opencv.org/doc/tutorials/tutorials.html>
- [3] Kaushik Deb, Kang-Hyun Jo, "Segmenting the License Plate Region Using Color Model", Pattern Recognition, 2009.
- [4] Christopher M Bishop, "Pattern Recognition and Machine Learning", Springer, 2006
- [5] Rafael C.Gonzalez, Richard E.Woods, "Digital Image Processing", Pearson International Edition, 2000.
- [6] Mark S.Nixon, Alberto S.Aguado, "Feature Extraction and Image Processing", Newnes, 2002.
- [7] Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, 2010.
- [8] Paul Viola, Michael J.Jones, "Robust Real-Time Detection".
- [9] Saqib Rasheed, Asad Naeem, Omer Ishaq, "Automated Number Plate Recognition Using Hough Lines and Template Matching".
- [10] Kaushik Deb, Kang-Hyun Jo, "Segmenting the License Plate Region Using a Color Model".
- [11] Derek Bradley, Gerhard Roth, "Adaptive Thresholding Using Intergral Image"



**Trần Hoài Linh** sinh năm 1974, tốt nghiệp ĐHBK Vác-sa-va (Ba Lan) năm 1997 chuyên ngành Tin học ứng dụng, nhận bằng Tiến sỹ chuyên ngành Kỹ thuật điện năm 2000 (ĐHBK Vác-sa-va), bằng Tiến sỹ khoa học chuyên ngành Kỹ thuật điện và Trí tuệ nhân tạo năm 2005 (ĐHBK Vác-sa-va).

Năm 2007 được phong Phó Giáo sư. Hiện nay Trần Hoài Linh đang công tác tại Viện Điện, trường ĐHBK Hà Nội. Các nghiên cứu chính của ông là ứng dụng trí tuệ nhân tạo trong các giải pháp đo lường, điều khiển và tự động hóa, các thiết bị đo thông minh, hệ chuyên gia.



**Lê Ngọc Sơn** sinh năm 1992, tốt nghiệp ĐHBK Hà Nội năm 2015 chuyên ngành Đo lường và Tin học công nghiệp.

Hiện nay Lê Ngọc Sơn đang công tác tại công ty Advanced Technology Inc. Các lĩnh vực nghiên cứu chính là hệ thống nhúng, các giải pháp nhận dạng,

phân loại lỗi, đo lường bằng hình ảnh.