

THIẾT KẾ HỆ THỐNG NHÚNG NHẬN DẠNG CHỮ VIẾT TAY

Trần Song Toàn *

Đặng Hữu Phúc **

Tóm tắt

Nhận dạng chữ viết tay là vấn đề đang được nghiên cứu và phát triển. Công việc xây dựng các hệ thống nhận dạng chữ viết tay được thực hiện khá nhiều trên các hệ thống máy tính cá nhân. Ngày nay, hệ thống nhúng đang được nghiên cứu và ứng dụng rộng rãi trong đời sống. Các hệ thống phân cứng được thiết kế ngày càng nhỏ gọn và có tính linh hoạt cao. Nội dung bài báo sẽ giới thiệu quá trình xây dựng hệ thống nhận dạng chữ viết tay off-line trên board nhúng Beagleboard xM (BBxM). Bài toán nhận dạng được thực hiện dưới sự hỗ trợ của thư viện xử lý ảnh OpenCV và phương pháp phân lớp SVM (Support vector machines). Ảnh các chữ viết tay sẽ được chụp từ camera. Hệ thống thực hiện công việc phát hiện và tách các ký tự thành các mẫu. Các mẫu ký tự sẽ được trích chọn đặc trưng theo phương pháp chu tuyến kết hợp với phương pháp xác định mật độ điểm ảnh. Quá trình xây dựng các ứng dụng trên hệ điều hành nhúng được thực hiện trên hệ điều hành Linux với sự hỗ trợ của phần mềm QtCreator và công cụ biên dịch chéo Qt-Everywhere. Kết quả thực nghiệm được thực hiện trên các mẫu chữ cái in hoa với tỉ lệ nhận dạng đúng đạt gần 95%.

Từ khóa: Beagleboard xM, hệ thống nhúng, nhận dạng chữ viết tay, phương pháp phân lớp Support Vector Machines.

Abstract

Handwriting recognition is an issue that is being studied and developed. The handwriting recognition systems are designed on personal computers. Nowadays, the embedded system is being studied and applied widely in life. The hardware systems are more compact and flexible. The content of the paper will introduce the process of building the offline handwriting recognition system on embedded board BeagleBoard xM (BBxM). The identification is carried out with the support of image processing library OpenCV and classification method SVM (Support vector machines). The handwriting letters will be captured by camera. The system will detect and separate the letter into samples. The features are extracted from these samples by the methods of circumference and pixel density. The process of building applications on embedded operating system is implemented on Linux operating system with the assistance of QtCreator software and Qt-Everywhere cross-compiler. The experimental result is performed on the Latin uppercase letters with the correct recognition rate up to approximately 95%.

Keywords: Beagleboard xM, Embedded system, Handwriting recognition, classification method Support Vector Machines.

1. Giới thiệu

Vấn đề nhận dạng chữ viết tay đang được quan tâm nghiên cứu và có vai trò rất quan trọng trong công nghiệp. Trọng tâm của vấn đề nằm ở khả năng thiết kế một giải thuật hiệu quả để có thể nhận dạng được các ký tự viết tay bởi người dùng qua tablet, scanner và các thiết bị số khác. Các ứng dụng của nhận dạng chữ viết tay đã và đang có nhiều đóng góp vào đời sống. Nhận dạng chữ viết tay có thể phục vụ cho các ứng dụng đọc và xử lý các chứng từ, hóa đơn, phiếu ghi, bản viết tay chương trình. Tuy nhiên, cho đến nay, việc nhận dạng chữ viết tay vẫn còn là thách thức lớn đối với các nhà nghiên cứu. Đã có nhiều công trình nghiên cứu về nhận dạng các mẫu chữ viết tay của

các hệ chữ cái Latin, Ả Rập, Trung Quốc,... nhưng kết quả vẫn còn khá hạn chế do chữ viết tay rất đa dạng về mẫu chữ cũng như các biến thể.

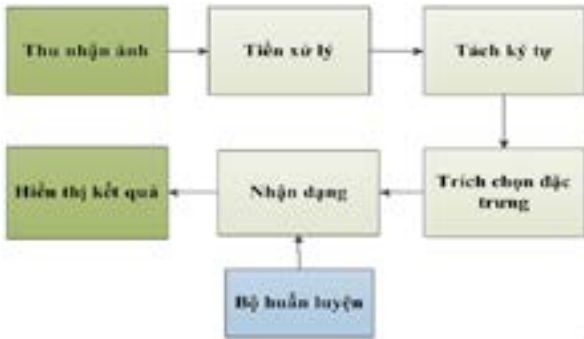
Nhận dạng chữ viết tay được thực hiện dựa trên hai giải thuật chính: memory base và learning base. Memory base lưu trữ ảnh các ký tự mẫu và nhận dạng một ký tự chưa biết bằng cách so sánh với các ký tự mẫu. Learning base là giải thuật cố gắng học các mẫu ký tự chưa biết và xây dựng hàm nhận dạng tương ứng. Các nghiên cứu về nhận dạng chữ số viết tay đã đạt được các kết quả trên 90%. Nhận dạng chữ viết tay thực hiện dựa trên sự kết hợp giữa các thuật toán xử lý ảnh và các phương pháp phân lớp nhận dạng như mạng Neural, Support Vector Machines,... Các nghiên cứu về bài toán nhận dạng

chữ viết tay đều thực hiện trên máy tính cá nhân.

Nội dung bài báo gồm các phần sau: phần 2 trình bày về hệ thống nhận dạng chữ viết tay, phần 3 trình bày các bước xây dựng ứng dụng trên board nhúng, phần 4 tổng hợp các kết quả thực hiện được và phần 5 trình bày kết luận và hướng nghiên cứu tiếp theo.

2. Hệ thống nhận dạng chữ viết tay

Hình 1 là sơ đồ khối của hệ thống nhận dạng chữ viết tay. Hệ thống gồm bốn khối chức năng chính: tiền xử lý, tách ký tự, trích chọn đặc trưng, nhận dạng. Bộ huấn luyện được kết nối với khối nhận dạng.

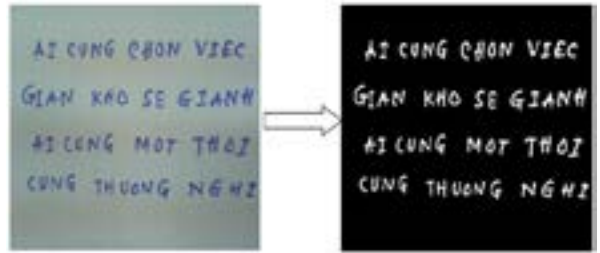


Hình 1. Sơ đồ hệ thống nhận dạng chữ viết tay

2.1. Tiền xử lý

Quá trình tiền xử lý thực hiện các phương pháp lọc nhiễu, hiệu chỉnh độ nghiêng, tìm ngưỡng để đưa ảnh về dạng nhị phân phục vụ cho các bước

tiếp theo. Ảnh nhị phân phục vụ cho quá trình nhận dạng là ảnh có nền đen và nét chữ trắng.



Hình 2. Quá trình tiền xử lý

2.2. Tách ký tự

Ảnh sau khi được nhị phân hóa sẽ được tách ra thành từng dòng chữ, từng chữ và cuối cùng là các ký tự rời rạc.

2.2.1. Tách dòng chữ

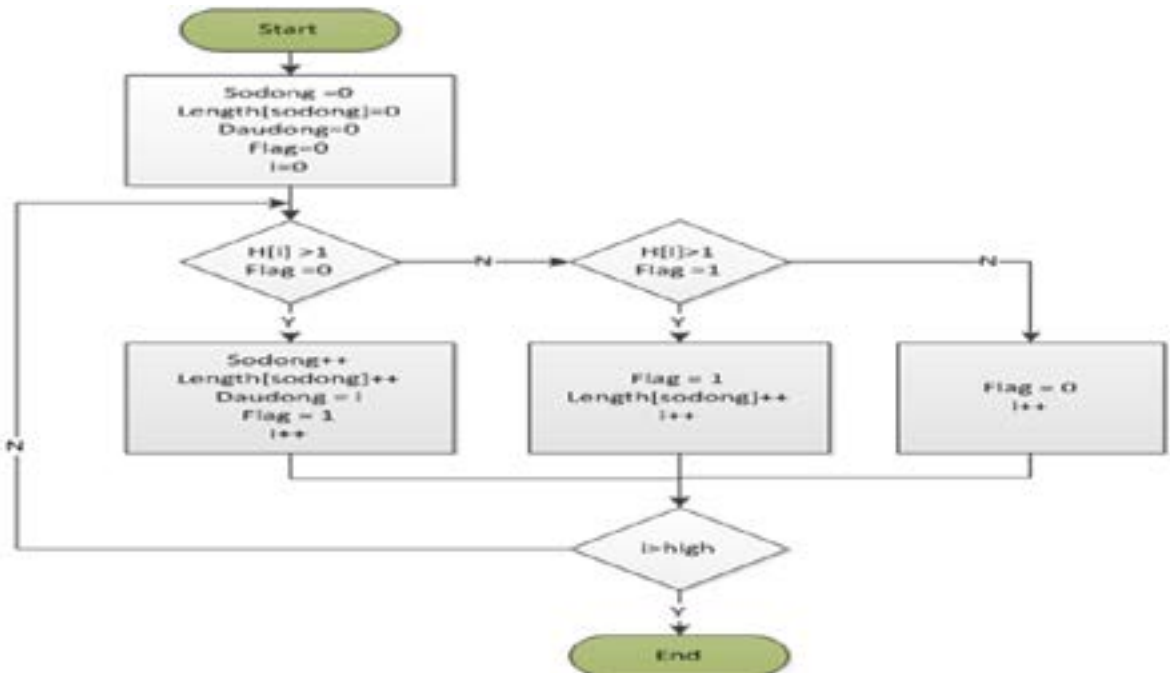
Quá trình tách dòng chữ được thực hiện theo các bước sau:

+ Xác định phân bố điểm ảnh theo dòng được thực hiện theo (1)

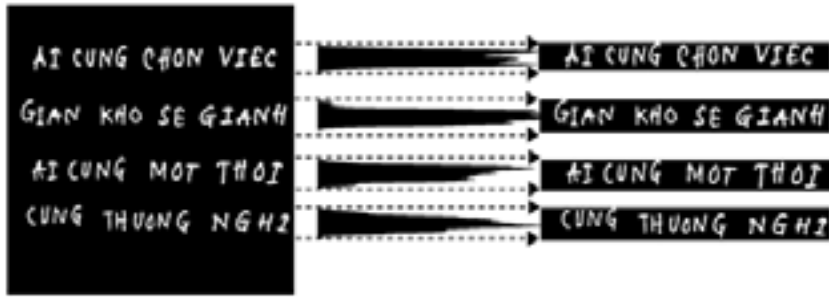
$$H[i] = \sum_{j=0}^{width-1} P(i,j) \text{div}(255) \quad (1)$$

Với $i = 0 \rightarrow high-1$ ($high$: chiều cao ảnh); $width$: độ rộng ảnh.

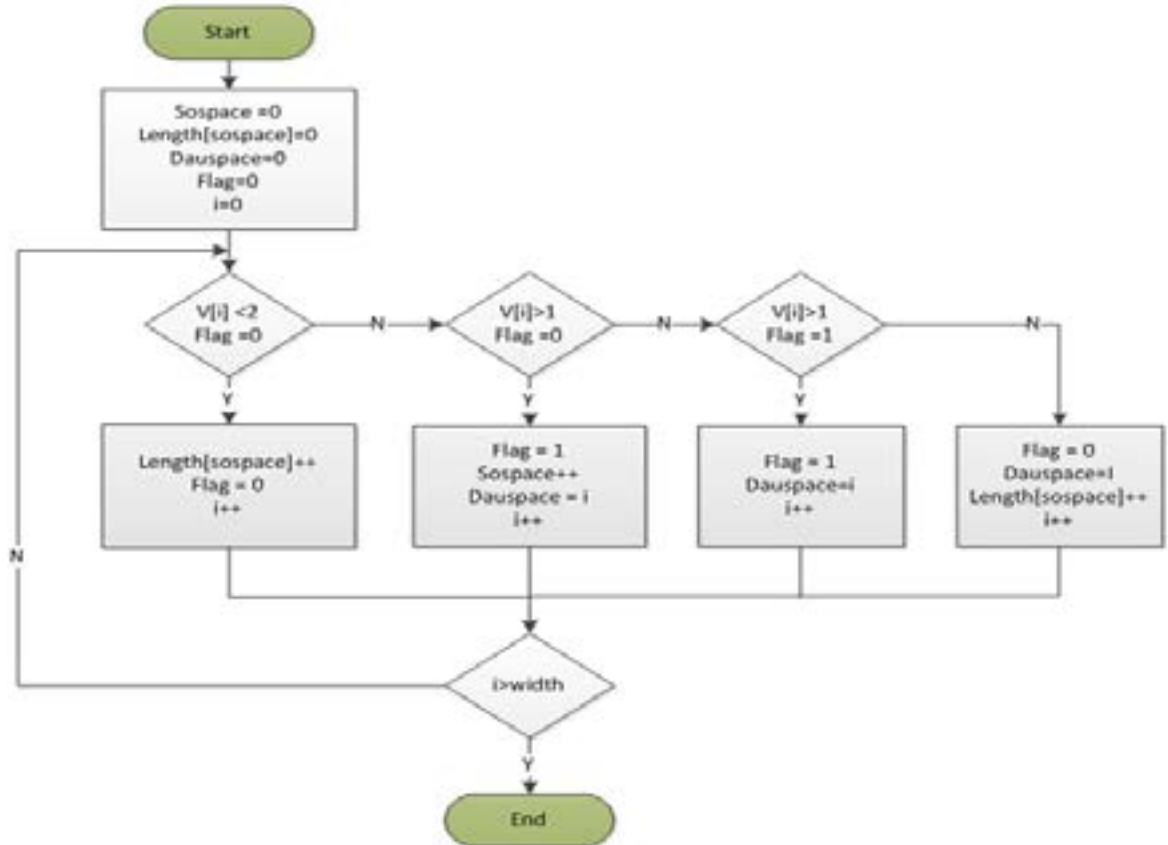
+ Xác định giá trị điểm bắt đầu dòng chữ và độ rộng dòng chữ. Giải thuật được thể hiện trong lưu đồ hình 3.



Hình 3. Giải thuật xác định dòng chữ



Hình 4. Kết quả quá trình tách dòng chữ



Hình 5. Giải thuật xác định các chữ trên dòng

+ Cắt ảnh dòng chữ từ ảnh gốc: Việc cắt ảnh các dòng được thực hiện bằng các hàm của thư viện OpenCV với vị trí độ lớn được xác định từ lưu đồ hình 3.

2.2.2. Tách chữ trên từng dòng chữ

Quá trình thực hiện việc tách chữ được thực hiện theo các bước:

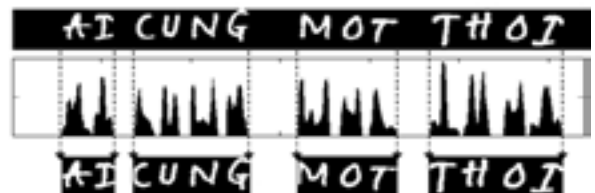
+ Xác định phân bố điểm ảnh theo cột được thực hiện theo (2)

$$V[j] = \sum_{i=0}^{high-1} P(i, j) \text{div}(255) \quad (2)$$

Với $i = 0 \rightarrow \text{width}-1$ (width: độ rộng ảnh); high: chiều cao ảnh.

+ Xác định điểm bắt đầu khoảng trắng, độ rộng khoảng trắng. Quá trình này được thực hiện theo lưu đồ Hình 5.

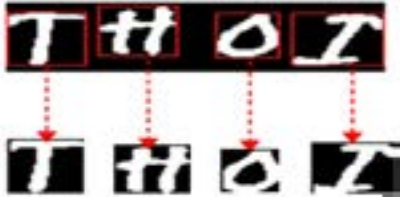
+ Cắt ảnh các chữ từ ảnh gốc thực hiện tương tự như cắt ảnh dòng chữ với các vị trí được xác định theo giải thuật trong hình 5.



Hình 6. Kết quả quá trình tách chữ

2.2.3. Tách ký tự

Quá trình tách ký tự được thực hiện theo giải thuật phát hiện hình bao của đối tượng và sau đó cắt các ký tự theo hình chữ nhật nhỏ nhất bao quanh ký tự.



Hình 7. Kết quả quá trình tách ký tự

2.3. Trích chọn đặc trưng

Quá trình trích chọn đặc trưng sẽ được thực hiện trên từng ký tự nhận được sau khi tách ký tự. Đặc trưng có ý nghĩa quan trọng trong việc quyết định tỷ lệ nhận dạng đúng ký tự. Để tránh những phức tạp của chữ viết tay cũng như tăng cường độ chính xác, ta cần phải biểu diễn thông tin chữ viết dưới những dạng đặc biệt hơn và cô đọng hơn, rút trích các đặc điểm riêng nhằm phân biệt các ký tự khác nhau. Đặc trưng của các ký tự có thể tìm ra bằng các phương pháp biến đổi, giải thuật toán học. Có nhiều hình thức trích đặc trưng thông dụng trong nhận dạng ký tự viết tay: đặc trưng nhị phân, đặc trưng vùng, đặc trưng theo chu tuyến, dùng biến đổi Haar Wavelet...

Phương pháp trích chọn đặc trưng được sử dụng trong bài báo là phương pháp trích đặc trưng theo chu tuyến kết hợp với phân bố điểm ảnh theo chiều ngang và chiều dọc. Ảnh ký tự được chuẩn hóa kích thước 22x30. Ảnh có nền màu đen nét chữ màu trắng. Với phương pháp trích đặc trưng trong bài báo thì chiều dài vector đặc trưng là 156. So với sử dụng phương pháp đặc trưng nhị phân (chiều dài vector đặc trưng là 660), phương pháp sử dụng trong bài báo sẽ giúp cho thời gian huấn luyện cũng như nhận dạng nhanh hơn gấp nhiều lần.

Vector đặc trưng F có được từ sự kết hợp của 6 giá trị.

$$F = \{L, R, U, D, V, H\} \tag{3}$$

Vector đặc trưng F có chiều dài là 156, trong đó L: đặc trưng lấy từ trái sang có 30 giá trị; R: đặc trưng lấy từ phải sang có 30 giá trị; U: đặc trưng lấy từ trên xuống có 22 giá trị; D: đặc trưng lấy từ dưới lên có 22 giá trị; V: đặc trưng phân bố dòng có 30 giá trị; H: đặc trưng phân bố cột có 22 giá trị.

2.3.1. Đặc trưng theo chiều từ trái sang

Việc lấy chu tuyến từ trái sang là xác định khoảng cách từ lề trái của mẫu đến vị trí điểm ảnh trắng đầu tiên theo từng dòng. Các giá trị đặc trưng L được xác định theo công thức (4).

$$L[i] = d[P(0,i),P_w] \tag{4}$$

Với $i = 0 \rightarrow 29$; $d[P(0,i),P_w]$ là khoảng cách từ điểm ảnh có tọa độ (0,i) đến điểm ảnh có giá trị 255 đầu tiên theo chiều từ trái sang.

2.3.2. Đặc trưng theo chiều từ phải sang

Chu tuyến từ phải sang xác định khoảng cách từ lề phải đến điểm trắng đầu tiên trên từng dòng. Các giá trị đặc trưng R được xác định theo công thức (5).

$$R[i] = d[P(21,i),P_w] \tag{5}$$

Với $i = 0 \rightarrow 29$; $d[P(21,i),P_w]$ là khoảng cách từ điểm ảnh có tọa độ (21,i) đến điểm ảnh có giá trị 255 đầu tiên theo chiều từ phải sang.

2.3.3. Đặc trưng theo chiều từ trên xuống

Xác định khoảng cách từ lề trên của mẫu đến điểm trắng đầu tiên theo từng cột. Các giá trị đặc trưng U được xác định theo công thức (6).

$$U[j] = d[P(j,0),P_w] \tag{6}$$

Với $j = 0 \rightarrow 21$; $d[P(j,0),P_w]$ là khoảng cách từ điểm ảnh có tọa độ (j,0) đến điểm ảnh có giá trị 255 đầu tiên theo chiều từ trên xuống.

2.3.4. Đặc trưng theo chiều từ dưới lên

Xác định khoảng cách từ lề dưới của mẫu đến điểm trắng đầu tiên theo từng cột. Các giá trị đặc trưng D được xác định theo công thức (7).

$$D[j] = d[P(j,29),P_w] \tag{7}$$

Với $j = 0 \rightarrow 21$; $d[P(j,29),P_w]$ là khoảng cách từ điểm ảnh có tọa độ (j,29) đến điểm ảnh có giá trị 255 đầu tiên theo chiều từ dưới lên.

2.3.5. Đặc trưng phân bố điểm ảnh theo dòng

Đặc trưng này dựa trên tổng số điểm trắng phân bố trên từng dòng ảnh. Các giá trị đặc trưng V được xác định theo công thức (8).

$$V[i] = \sum_{j=0}^{21} I(j) \text{ với } \begin{cases} I(j) = 1 \text{ nếu } P(i,j) = 255 \\ I(j) = 0 \text{ nếu } P(i,j) = 0 \end{cases} \tag{8}$$

Với $j = 0 \rightarrow 21$; $P(i,j)$ giá trị điểm ảnh (i,j)

2.3.6. Đặc trưng phân bố điểm ảnh theo cột

Xác định tổng số điểm trắng phân bố trên từng cột. Các giá trị đặc trưng H được xác định theo công thức (9).

$$H[j] = \sum_{i=0}^{29} J(i) \text{ với } \begin{cases} J(i) = 1 \text{ nếu } P(i,j) = 255 \\ J(i) = 0 \text{ nếu } P(i,j) = 0 \end{cases} \quad (9)$$

Với $j = 0 \rightarrow 21$; $P(i,j)$ giá trị điểm ảnh (i,j)

2.4. Nhận dạng

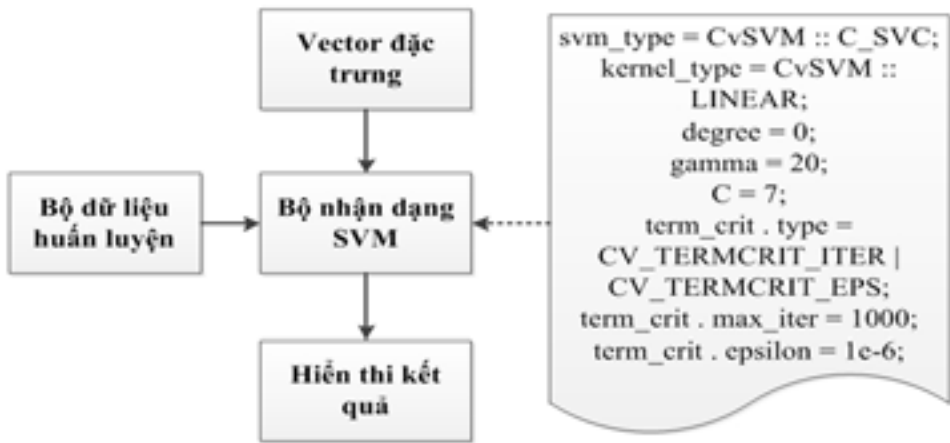
Mẫu ký tự được tách ra trong quá trình tách ký tự sẽ được đưa vào kết hợp với bộ huấn luyện sẽ cho kết quả nhận dạng. Kết quả sau khi nhận dạng với phương pháp phân lớp Support Vector machines (SVM) sẽ trả về một giá trị nhãn trong bộ huấn luyện SVM. Hàm nhân (kernel) được sử

dụng trong phương pháp SVM là hàm tuyến tính theo biểu thức (10), các tham số được thiết lập theo phương pháp C-SVM được thể hiện theo công thức (11) và các giá trị tham số được thể hiện trong hình 12.

$$K(x_i, x_j) = x_i^T x_j \quad (10)$$

$$\text{Minimize}_{\xi, w, b} = \left[\frac{1}{2} \langle w^T w \rangle + C \sum_{i=1}^l \xi_i \right] \quad (11)$$

Quá trình huấn luyện được thực hiện với sự hỗ trợ của thư viện OpenCV với các tham số được trình bày trong hình 8. Giá trị nhãn sẽ cho ta một ký tự tương ứng. Kết quả sẽ được ghi vào file đồng thời hiển thị trên giao diện của ứng dụng.



Hình 8. Quá trình nhận dạng mẫu

3. Xây dựng ứng dụng trên board nhúng BeagleBoard xM (BBxM)

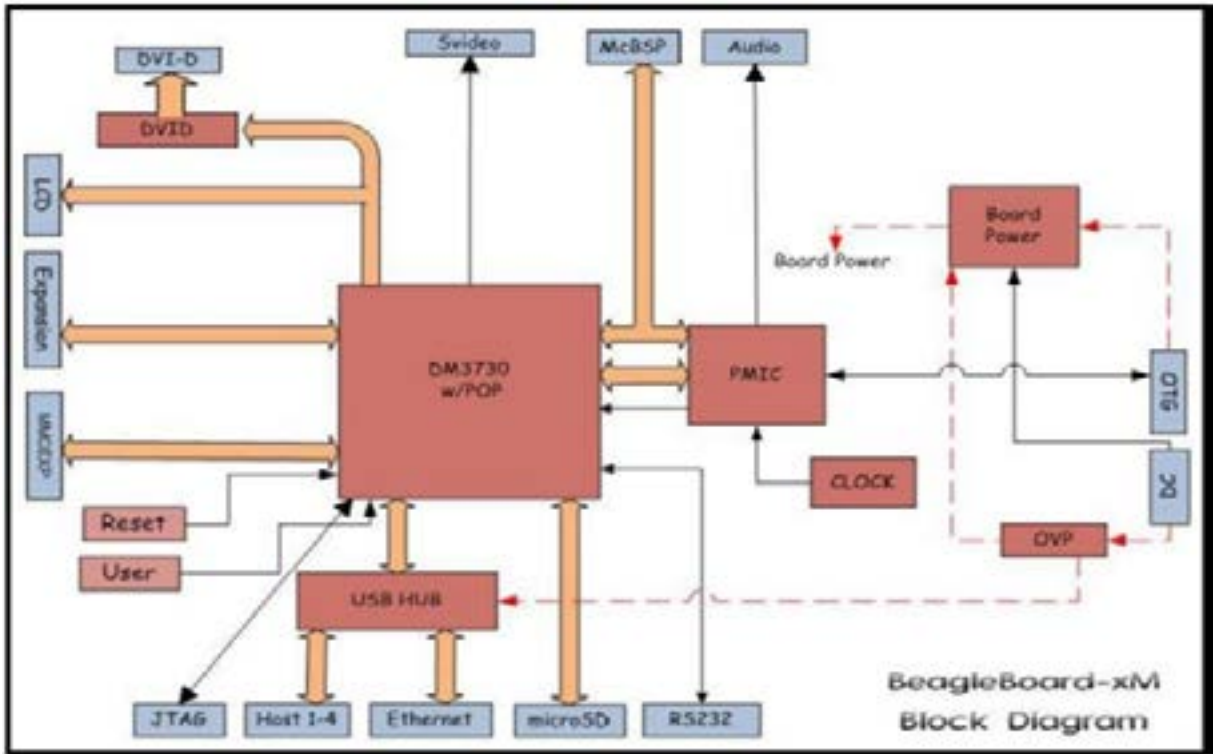
3.1. Hệ thống phần cứng

Sơ đồ hệ thống phần cứng của ứng dụng được thể hiện trong Hình 9. Ảnh các ký tự được thu qua camera đưa vào board để xử lý và kết quả của chương trình được hiển thị trên màn hình LCD.

BBxM sử dụng CPU ARM-Cortex A8 có thể hoạt động tối đa ở tốc độ 1Ghz cùng với bộ nhớ DDRAM công suất thấp 512MB. Kích thước của BBxM khá nhỏ gọn: 3.25” × 3.25”. BBxM cung cấp khá đầy đủ các kết nối của một máy tính mini: USB port, Audio input connector, Audio output connector, Svideo connector, DVI-D connector, LCD header...



Hình 9. Hệ thống phần cứng



Hình 10. Sơ đồ khối BBxM

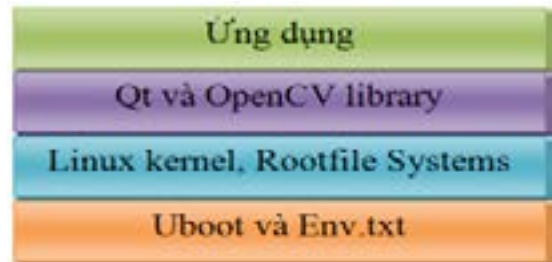
Hệ thống nhận dạng chữ viết tay được thực hiện và kiểm tra trên máy tính cá nhân. Sau khi kết quả đạt yêu cầu, nó sẽ được biên dịch thành ứng dụng có thể thực thi trên board nhúng BBxM. Quá trình này được trình bày trong phần 3.3.

3.2. Hệ thống phần mềm

Hệ thống hỗ trợ thiết kế phần mềm gồm có:

- + Hệ điều hành Ubuntu 10.10: xây dựng và thiết kế các giao tiếp với BBxM.
- + Angstrom-toolchain: công cụ biên dịch chéo các ứng dụng và các gói cho kiến trúc ARM-Cortex A8.
- + Qt Creator 2.0.1: công cụ thiết kế giao diện và ứng dụng.
- + Qt-Everywhere 4.8.2: Công cụ biên dịch Qt cho kiến trúc ARM.
- + OpenCV library: thư viện hỗ trợ cho công việc xử lý ảnh.

Quá trình cài đặt các ứng dụng cần quan tâm đến vấn đề cài đặt driver cho hệ thống. Hệ thống nhận dạng có sử dụng USB camera nên cần phải cài đặt trước khi sử dụng cho hệ thống.



Hình 11. Hệ thống phần mềm

3.3. Biên dịch ứng dụng cho BBxM

Ứng dụng được xây dựng với phần mềm Qt Creator được cài đặt trên hệ điều hành Ubuntu 10.10. Công việc xử lý ảnh được thực hiện với thư viện OpenCV. Để có thể thực thi ứng dụng trên BBxM các thư viện hỗ trợ của Qt và OpenCV phải chạy được trên nền kiến trúc ARM. Một toolchain cần được cài đặt để biên dịch code trên hệ thống Linux có thể thực thi trên vi xử lý ARM của BBxM. Các vi xử lý ARM khác nhau sẽ được biên dịch khác nhau. Toolchain hỗ trợ cho việc build ứng dụng trên BBxM với hệ điều hành Angstrom là Angstrom toolchain.

Để có thể sử dụng thư viện OpenCV trên BBxM cần phải build các thư viện này với angstrom-toolchain.



Hình 12. Build thư viện OpenCV cho BBxM

Chương trình được xây dựng trên phần mềm Qt nên cần phải build các ứng dụng này với Qt-Everywhere.



Hình 13. Build thư viện Qt cho BBxM

Hệ điều hành được sử dụng trên BBxM là Angstrom OS. Để các ứng dụng có thể chạy được trên BBxM, cần cài đặt hệ điều hành trên BBxM.

4. Kết quả

Ứng dụng được thiết kế trực quan với giao diện dễ dàng sử dụng và kiểm tra. Kết quả cho phép nhận dạng các chữ viết tay rời rạc trên các dòng chữ khác nhau. Hình 14a, hình 14b và 14c cho thấy kết quả nhận dạng trên các loại văn bản khác nhau.

Kết quả quá trình tách ký tự đối với các văn bản các ký tự không dính nét đạt xấp xỉ 95%. Do phương pháp tách dòng và tách ký tự được sử dụng trong bài báo là phương pháp hình chiếu nên đối với các ảnh văn bản có độ nghiêng lớn dễ gây nhầm lẫn trong quá trình tách dòng và tách chữ.

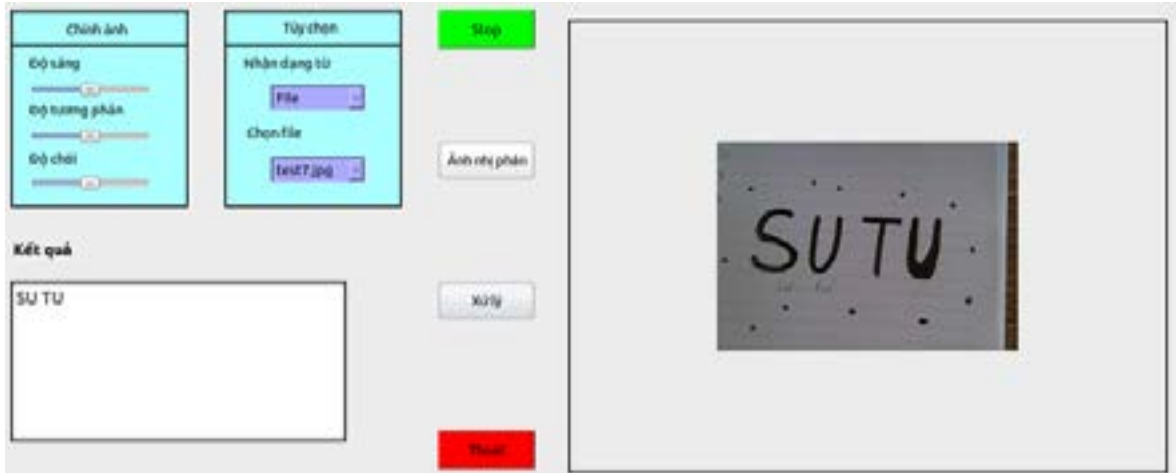
Bài báo sử dụng ba phương pháp trích đặc trưng: đặc trưng theo chu tuyến (CT), đặc trưng nhị phân (NP) và đặc trưng chu tuyến kết hợp với phân bố mật độ điểm ảnh (CT+PB). Số lượng mẫu được sử dụng trong bài báo bao gồm 4799 mẫu trong đó bao gồm 1504 mẫu được dùng để training và 3295 mẫu dùng để testing.



Hình 14a. Nhận dạng văn bản với nhiều dòng chữ



Hình 14b. Nhận dạng văn bản nghiêng



Hình 14c. Nhận dạng văn bản có nhiễu

Phương pháp trích đặc trưng được sử dụng trong bài báo đã cho thời gian huấn luyện cũng như nhận dạng ngắn hơn so với các phương pháp trích đặc trưng nhị phân, đặc trưng vùng hay biến đổi Haar Wavelet.

Thời gian training và testing (thực hiện trên máy tính cá nhân, CPU core i3, 2GB RAM) được thể hiện trong bảng 1.

Bảng 1: Thời gian training và testing

Đặc trưng	CT + PB	CT	NP
Training	~0.66s	~0.51s	~1.8
Testing	~1.17s	~0.88s	~3.78s

Thời gian nhận dạng được thể hiện trong Bảng 2. Thời gian này phụ thuộc vào kích thước bức ảnh, độ phức tạp và số lượng ký tự có trên bức ảnh.

Thiết bị	Kích thước ảnh	
	320x240	640x480
Máy tính cá nhân	~20-40ms	~100-130ms
BBxM	~0.5s	~1s

Bảng 2: Thời gian nhận dạng

Kết quả nhận dạng trên các mẫu ký tự được thể hiện trong bảng 3. Kết quả này được thực hiện trên các mẫu ký tự rời rạc, các mẫu này gồm 26 mẫu ký tự in hoa (A->Z). Kết quả thể hiện trong Bảng 3 là kết quả trung bình cộng của 26 kết quả đối với từng ký tự riêng lẻ.

Bảng 3: Kết quả nhận dạng

Số mẫu training	Số mẫu testing	Tỉ lệ nhận dạng đúng (%)		
		CT+PB	CT	NP
1504	3295	94.43	93.39	90.94

Từ kết quả trên, ta nhận thấy phương pháp trích đặc trưng theo chu tuyến kết hợp với mật độ điểm ảnh cho hiệu suất nhận dạng tốt nhất và thời gian xử lý ngắn. Phương pháp trích đặc trưng nhị phân thời gian xử lý nhiều nhưng tỉ lệ nhận dạng vẫn thấp hơn so với phương pháp chu tuyến.

5. Kết luận

Bài báo đã đưa ra giải thuật nhận dạng chữ viết tay và thực hiện trên board nhúng BBxM. Kết quả thực hiện trên BBxM có thể áp dụng được trên thực tế (thời gian nhận dạng khoảng 100ms với kết quả đạt được xấp xỉ 95%).

Kết quả của bài toán xử lý ảnh chỉ dừng lại ở mức độ tách các chữ viết rời rạc không bị dính nét. Hướng nghiên cứu và phát triển tiếp theo là xây dựng mô hình nhận dạng chữ viết để có thể nhận dạng được các chữ viết bị dính nét. Việc này đòi hỏi cần phải sử dụng kỹ thuật phân đoạn ảnh nâng cao để có thể tách được các ký tự dính nét. Phát triển mô hình nhận dạng chữ viết tay tiếng Việt. Xây dựng giải thuật hậu xử lý để nâng cao tỷ lệ nhận dạng. Kết quả sau khi nhận dạng sẽ được xem xét trong ngữ cảnh để có thể giải quyết một số trường hợp sai so với cấu trúc ngữ pháp tiếng Việt. Cải tiến giải thuật giúp cho hệ thống hoạt động nhanh hơn để có thể đáp ứng với hệ thống thời gian thực.

Tài liệu tham khảo

- BeagleBoard-xM Rev C System Reference Manual. April 4th, 2010.
- Bui Minh Thanh, Truong Quang Vinh, Hoang Trang. 2012. “*An Automatic Licence Plate Recognition System Based on Support Vector Machine*”. The Second Solid-State Systems Symposium – VLSIs and Semiconductor Related Technologies. pp 31-36.
- Jose Israel Pacheco. 2011. “*A comparative study for the handwritten digit recognition problem*”. Presented to the Department of Mathematics and Statistics California State University, Long Beach.
- Le Anh Cuong, Ngo Tien Dat, Nguyen Viet Ha. 2010. “*Isolated Handwritten Vietnamese Character Recognition with Feature Extraction and Classifier Combination*”. VNU Journal of Science, Mathematics - Physics 26. pp 123-139.
- Lê Trần Hùng Ân. 2012. “*Nhận dạng chữ viết tay*”. Luận văn tốt nghiệp đại học. Trường Đại học Bách Khoa, Đại học QG TP HCM.
- Merciadri Luca, Köen Kooi. 2010. *Ångström Manual*.
- OpenCV Reference Manual v2.1. 2010.
- Phạm Anh Phương. 2009. “*Một số phương pháp trích chọn đặc trưng hiệu quả cho bài toán nhận dạng chữ viết tay rời rạc*”. Tạp chí khoa học, Đại học Huế. số 53. pp. 73-79.
- Phạm Anh Phương, Ngô Quốc Tạo, Lương Chi Mai. 2008. “*Vietnamese Handwritten Character Recognition By Combining SVM and Haar Wavelet Features*”. Các công trình nghiên cứu khoa học, nghiên cứu triển khai CNTT - TT. Số 20. pp. 36-42.
- Y. LeCun and C. Cortes. 2012. “*The MNIST Database of Handwritten Digits.*” Internet: <http://yann.lecun.com/exdb/mnist/>