

THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN CỬA TỰ ĐỘNG SỬ DỤNG CÔNG NGHỆ XỬ LÝ GIỌNG NÓI VÀ CÔNG NGHỆ XỬ LÝ ẢNH

DESIGNING AN AUTOMATIC DOOR CONTROL SYSTEM APPLYING SPEECH PROCESSING AND IMAGE PROCESSING

Lê Tiến Dũng¹, Trịnh Ngọc Văn², Nguyễn Văn Kiên², Nguyễn Đức Tường²

¹Trường Đại học Bách khoa, Đại học Đà Nẵng; ltdung@dut.udn.vn

²Lớp 12DCLC, Trường Đại học Bách khoa, Đại học Đà Nẵng;

vantnbk91@gmail.com, vankienemk@yahoo.com, tuongnguyen9494@gmail.com

Tóm tắt - Ngày nay, với sự phát triển không ngừng của ngành tự động hóa, con người có xu hướng sử dụng các thiết bị tự động để nâng cao tính tiện nghi, an toàn cho không gian sống của họ. Trong bài báo này, chúng tôi đề xuất một giải pháp nhằm nâng cao tính an ninh và tiện lợi của hệ thống điều khiển cửa tự động. Hệ thống điều khiển này sử dụng các công nghệ bảo mật như mật khẩu, bảo mật bằng thẻ RFID, hay bằng các phương pháp sinh trắc học khác như nhận diện khuôn mặt hay giọng nói của người dùng. Ngoài ra hệ thống có thêm chức năng truyền hình ảnh từ camera đặt ở cửa tới các thiết bị trong mạng LAN, cũng như giao tiếp với các thiết bị trong mạng LAN thông qua giao thức MQTT. Đồng thời chúng tôi xây dựng một mô hình để kiểm nghiệm hoạt động thực tế các giải pháp đã đề ra.

Từ khóa - xử lý giọng nói; xử lý hình ảnh; hệ thống bảo mật; nhận dạng khuôn mặt; cửa thông minh.

Abstract - Nowadays, because of the consistent development of automation technology, there is a growing tendency among people to use automatic devices to improve the convenience and safety of their living space. In this paper, we propose a measure to enhance the convenience and security of the control system of automatic doors. Our control system uses security technology such as password, RFID or biometric methods like face recognition or speech recognition. Besides, our system have two extra functions: transmitting images captured by camera via Local Area Network (LAN) to other devices and saving video to review when it is necessary. The system adopt MQTT as the main communication protocol to exchange messages. Finally, we build up a working model to demo and test our solution.

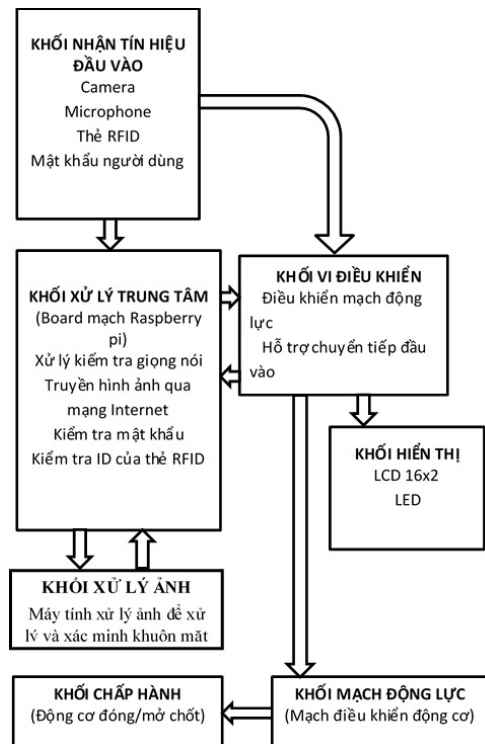
Key words - speech processing; image processing; security systems; face recognition; intelligent door.

1. Đặt vấn đề

Ngày nay, khi đời sống kinh tế - xã hội phát triển, vấn đề an ninh được quan tâm nhiều hơn. Ở nước ta, việc bảo vệ an ninh đang dần chuyển trọng tâm từ việc người ta vẫn đang sử dụng con người cho mục đích an ninh là chính như việc thuê các bảo vệ viên, hay lắp đặt các hệ thống camera thông thường mà một kẻ gian hiểu biết có thể qua mặt được.

Trên thế giới hiện đã có nhiều công trình nghiên cứu và áp dụng hệ thống điều khiển cửa thông minh. Trước hết, có thể kể đến hệ thống chuông cửa thông minh Chui [4] sử dụng kỹ thuật nhận diện khuôn mặt, thẻ từ và mật khẩu để nhận diện và quyết định mở khóa. Ngoài ra, hệ thống này còn truyền hình ảnh qua mạng LAN để người dùng biết được người đang đứng trước cửa là ai. Tuy nhiên, hệ thống đó chưa lưu lại video để người dùng dễ dàng xem lại khi cần. Một ứng dụng thương mại khác là hệ thống khóa HF6618 sử dụng kỹ thuật nhận diện khuôn mặt, thẻ từ, khóa cơ và mật khẩu hoặc kết hợp để mở khóa [8]. Nhược điểm của hệ thống này là chưa truyền hình ảnh qua mạng LAN cũng như lưu lại video để người sử dụng giám sát ngôi nhà của họ. Bên cạnh đó, có thể kể đến hệ thống cửa thông minh ở các sân bay Australia [2] đối chiếu khuôn mặt trong hộ chiếu điện tử và khuôn mặt thu được từ camera để mở cửa cho du khách nhập cảnh. Hệ thống này khá cồng kềnh và cũng chưa có chức năng lưu video hay truyền qua mạng LAN.

Trong bài báo này, chúng tôi sẽ xây dựng một mô hình hệ thống cửa tự động với cấp độ bảo mật cao và khắc phục các vấn đề còn tồn tại của các hệ thống cửa thông minh kể đến ở trên. Sơ đồ khối cấu trúc của toàn hệ thống được biểu diễn ở hình 1.



Hình 1. Sơ đồ khối của hệ thống

Hệ thống cửa này sử dụng camera để phân biệt, nhận diện các đối tượng được phép mở cửa làm phương thức bảo mật chính. Ngoài ra, hệ thống cửa còn chấp nhận các yêu cầu khác như việc nhập khẩu từ bàn phím có sẵn hay nhận dạng giọng nói người dùng, đồng thời lưu trữ hình ảnh từ camera để thuận tiện cho việc truy xuất hình ảnh khi cần.

Khi người dùng yêu cầu mở cửa, một cơ cấu chốt khóa sẽ tự động được kéo ra cho phép người dùng có thể mở cửa. Cơ cấu này cũng sẽ gài lại khi có yêu cầu từ người dùng. Hệ thống còn có một số tính năng bổ sung khác như truyền hình ảnh từ camera đến các thiết bị khác như điện thoại, máy tính, cũng như chế độ từ chối tiếp khách liên động với chuông cửa. Cái cốt lõi nhất mà bài báo này hướng đến đó là giải pháp tự động hóa mang tính cải tiến, nâng cao bảo mật và ứng dụng thực tế cao.

Phần còn lại của bài báo được bố cục như sau: Mục 2 trình bày thuật toán xử lý ảnh và nhận diện khuôn mặt; Mục 3 trình bày về phương án xử lý giọng nói cho hệ thống điều khiển của cửa tự động; Mục 4 trình bày về giải pháp truyền thông. Một mô hình thực nghiệm được xây dựng để kiểm chứng giải pháp đề xuất, được trình bày trong mục 5. Cuối cùng, mục 6 đưa ra những kết luận, đề xuất.

2. Thuật toán xử lý ảnh và nhận diện khuôn mặt

2.1. Thuật toán xử lý ảnh

Hệ thống xác minh khuôn mặt trong bài báo này thực hiện xử lý ảnh qua bốn bước như sau:

- Phát hiện khuôn mặt (detection): Tìm ra vị trí và kích thước của khuôn mặt trong bức ảnh.
- Căn chỉnh khuôn mặt (alignment): Tìm các điểm landmarks rồi dựa vào các điểm này để biến đổi khuôn mặt (xoay, tịnh tiến hay thu phóng) sao cho khuôn mặt trong bức ảnh ở góc nhìn chính diện nhất có thể.
- Biểu diễn (Representation): Biểu diễn khuôn mặt thành các số đo để dựa vào đó xác minh khuôn mặt.
- Xác minh (Verification): Dựa vào các số đo đã được tìm ra để xác minh khuôn mặt có phải là người dùng hay không.

Bức ảnh thu được từ camera có kích thước là 480x640x3, bức ảnh này sẽ được xử lý để phát hiện ra vị trí và kích thước của khuôn mặt; khuôn mặt này được cắt ra rồi được điều chỉnh về kích thước 96x96x3. Bức ảnh đã được điều chỉnh sẽ được xử lý tiếp qua các giai đoạn sau để xác minh.

Các khung hình được xử lý liên tiếp. Sau khi xử lý xong một khung hình, máy tính sẽ lấy tiếp một khung hình nữa rồi lặp lại các bước xử lý như trên; quá trình này được lặp lại vô hạn.

Trong ngôn ngữ lập trình Python, các bức ảnh được biểu diễn ở dạng numpy array. Ví dụ một mảng numpy array như sau:

$$I = [[1, 0, 0], [0, 1, 2]]$$

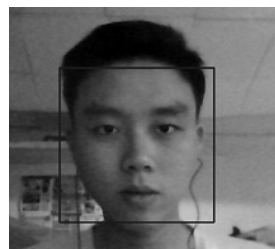
Trong mảng này, các số ở giữa mỗi cặp ngoặc vuông bên trong là các phần tử của một hàng; dấu phẩy giữa các cặp ngoặc vuông phân cách các hàng với nhau. Việc đánh số các phần tử trong hàng và cột được bắt đầu bằng số 0, ví dụ $I[0, 0]$ trả về giá trị 1. Để lấy tất cả các phần tử ở một chiều nào đó, ta dùng dấu ":" ở các chiều khác, ví dụ để lấy hàng đầu tiên, ta dùng lệnh $I[1,:]$ (lấy tất cả các cột ở hàng 1).

Các bức ảnh được biểu diễn bằng mảng numpy array 3 chiều theo thứ tự là chiều dọc, chiều ngang và chiều sâu. Sau đây, chi tiết về các bước xử lý ảnh và xác minh sẽ được trình bày.

2.2. Phát hiện khuôn mặt

Thuật toán được sử dụng để phát hiện khuôn mặt trong bài báo này là thuật toán Viola-Jones. Ưu điểm nổi bật của thuật toán này là tốc độ xử lý. Thực tế đây là thuật toán phát hiện khuôn mặt có tốc độ nhanh nhất và có độ chính xác tương đối cao. Bài báo này sử dụng thư viện xử lý ảnh mã nguồn mở OpenCV [3] để thực hiện việc phát hiện khuôn mặt.

Sau khi phát hiện khuôn mặt, chương trình trả về vị trí và kích thước của hình chữ nhật bao quanh khuôn mặt như hình 2. Từ đó, ta có thể tách khuôn mặt ra khỏi bức ảnh để thực hiện tiếp các bước nhận dạng.



Hình 2. Kết quả của đoạn chương trình phát hiện khuôn mặt

2.3. Căn chỉnh khuôn mặt

Để căn chỉnh khuôn mặt, trước hết, cần phải tìm các điểm landmarks, đây là các điểm đặc biệt có mặt trong mọi khuôn mặt, chẳng hạn như góc mắt, đỉnh mũi hay khõe miệng. Bài báo này sử dụng thư viện xử lý ảnh mã nguồn mở Dlib [6] để tìm 68 điểm landmarks trên khuôn mặt. Thuật toán được sử dụng là thuật toán Ensemble of Regression Trees; thuật toán này có tốc độ thực thi rất cao, nó có thể xử lý 1.000 bức ảnh trong vòng 1s.

Minh họa các điểm landmarks trên một khuôn mặt như hình 3.



Hình 3. Kết quả tìm điểm landmarks và căn chỉnh khuôn mặt

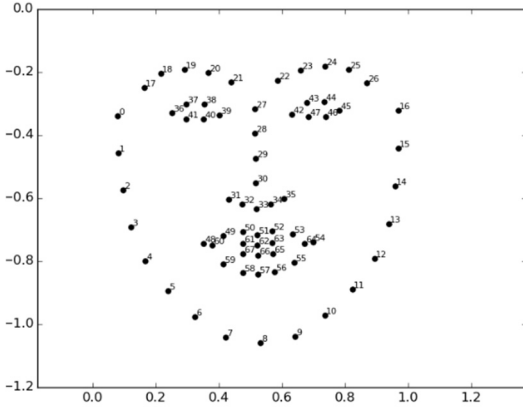
Sau khi phát hiện các điểm landmarks, ta sẽ dựa vào các điểm này để biến đổi khuôn mặt sao cho góc nhìn được chính diện nhất. Ở đây ta sẽ sử dụng phép biến đổi affine. Cụ thể là tìm một ma trận H , ma trận này biến đổi các điểm có tọa độ $(x_i, y_i, 1)$ trong bức ảnh gốc thành các điểm có tọa độ (x'_i, y'_i) [3] trong bức ảnh mới sao cho:

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (1)$$

Để tìm được H , ta cần dựa vào một tính chất của phép biến đổi affine như sau: Với hai hình tam giác bất kỳ, ta luôn tìm được một phép biến đổi affine để biến đổi tam giác này thành tam giác kia [5]. Dựa vào tính chất này, ta sẽ chọn ra ba điểm, tìm ma trận H sao cho với ma trận H , ba điểm landmarks được chọn trong bức ảnh gốc được biến đổi thành ba điểm landmarks chuẩn tương ứng. Hình 4 bên dưới minh họa 68 điểm landmarks chuẩn mà bài báo này sử dụng [1].

Gọi A, B, C là ba điểm landmarks được chọn trong bức ảnh gốc và A', B', C' là ba điểm tương ứng của landmarks chuẩn, từ phương trình (1) ta có:

$$\begin{bmatrix} x_{A'} \\ y_{A'} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix} \quad (2)$$



Hình 4. Các điểm landmarks chuẩn được sử dụng

$$\begin{bmatrix} x_{B'} \\ y_{B'} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x_{C'} \\ y_{C'} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix} \quad (4)$$

Giải hệ phương trình trên ta tìm ra H:

$$\begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \end{bmatrix} = \begin{bmatrix} x_A & y_A & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_A & y_A & 1 \\ x_B & y_B & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_B & y_B & 1 \\ x_C & y_C & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_C & y_C & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_{A'} \\ y_{A'} \\ x_{B'} \\ y_{B'} \\ x_{C'} \\ y_{C'} \end{bmatrix} \quad (5)$$

Ma trận biến đổi cuối cùng là:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix} \quad (6)$$

Trong bài báo này, các điểm landmarks số 39, 42 và 33 được sử dụng để biến đổi.

2.4. Biểu diễn khuôn mặt

Để so sánh hai khuôn mặt, điều cần thiết là phải biểu diễn bức ảnh thành các tính chất để làm giảm sự biến thiên đối với các bức ảnh của cùng một người, trong khi đó phải làm tăng sự biến thiên đối với các bức ảnh của hai người khác nhau. Đây là phần quan trọng nhất trong quá trình xác minh khuôn mặt vì quá trình xác minh hoàn toàn dựa vào các tính chất này.

Hệ thống của bài báo này sử dụng mạng neuron đã được luyện sẵn của thư viện mã nguồn mở OpenFace [1]. Mạng neuron có input là một bức ảnh màu RGB với kích thước 96x96x3 và output là một vector chứa 128 số đo, được gọi là embeddings. Khoảng cách Euclide giữa hai embeddings của hai bức ảnh sẽ lớn nếu chúng thuộc về hai người khác nhau và sẽ nhỏ nếu chúng thuộc về hai người khác nhau. Gọi x là một bức ảnh đầu vào và $f(x)$ là vector chứa các embeddings của x [7]. Mạng này được huấn luyện sao cho:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \quad (7)$$

Trong đó x_i^a là bức ảnh của một người để đối chiếu; x_i^p là một bức ảnh khác của cùng một người với x_i^a và x_i^n là ảnh của một người khác với x_i^a và x_i^p . Từ bất đẳng thức (7) ta thấy rằng khoảng cách của các embeddings hai khuôn mặt của hai người khác nhau luôn lớn hơn khoảng cách của các embeddings của hai khuôn mặt cùng một người ít nhất là α (với mạng neuron của OpenFace, $\alpha = 0.2$) [1].

2.5. Xác minh khuôn mặt

Sau khi biểu diễn khuôn mặt, việc xác minh là rất dễ dàng. Từ công thức (7) ta rút ra:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 - \alpha \quad (8)$$

Từ công thức (8) suy ra:

$$0 < \|f(x_i^a) - f(x_i^p)\|_2^2 - \alpha \quad (9)$$

Như vậy, khoảng cách Euclide của hai khuôn mặt của hai người khác nhau luôn lớn hơn α , trong bài báo này là 0,2. Như vậy, khi lập trình, chỉ cần tính khoảng cách giữa khuôn mặt đã được lưu trong máy tính với khuôn mặt thu được từ camera, nếu khoảng cách này nhỏ hơn 0,2 thì kết luận là hai khuôn mặt là của cùng một người. Ngược lại, nếu lớn hơn 0,2 thì kết luận là hai khuôn mặt khác nhau. Tuy nhiên, để đảm bảo an toàn, ta chọn ngưỡng thấp hơn, trong bài báo này chúng tôi chọn 0,18.

2.6. Đọc ảnh từ sever

Để đọc ảnh từ server trả về, trước hết, cần phải kết nối với server. Server trong bài báo này trả về ảnh dạng JPEG; đây là chuẩn nén ảnh phổ biến trên internet. Nếu server chỉ trả về một bức ảnh thì ta chỉ cần giải mã chuỗi dữ liệu nhận được trong biến stream là đã được một bức ảnh. Tuy nhiên, trong bài báo này, server liên tục trả ảnh về nên cần phải có cách để đọc từng chuỗi mã hóa ảnh trong stream ra. Để tìm được chuỗi mã hóa ảnh, ta cần tìm các chuỗi '\xff\xd8' và '\xff\xd9'; đây là các chuỗi mở đầu và kết thúc được quy ước trong định dạng ảnh JPEG. Sau khi đọc được chuỗi mã hóa, ta sẽ xóa các giá trị từ '\xd9' trở về trước. Quá trình đọc ảnh tiếp tục được lặp lại để đọc các ảnh tiếp theo.

2.7. Lưu ảnh thành video

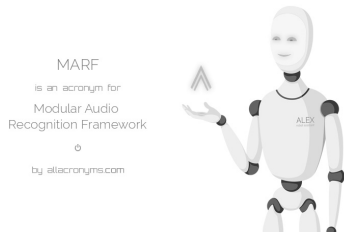
Để lưu ảnh thành video, ta dùng hàm cv2.imwrite() của thư viện opencv. Ta có ba thông số cần phải khai báo. Thông số đầu tiên là tên của video (ví dụ 'out.avi'), sau đó cần phải khai báo mã Fourcc và số khung hình trên một giây (frames per second). Cuối cùng là cờ isColor; nếu cờ này được set bằng True thì chương trình sẽ lưu thành video với các khung hình màu, nếu bằng False thì chương trình lưu lại thành video đen trắng.

Fourcc là một đoạn mã dùng để nhận biết mã ảnh động (video codec), định dạng nén và định dạng pixel hoặc màu được sử dụng trong video. Mỗi mã Fourcc được ký hiệu bằng bốn chữ cái trong bảng mã ASCII. Mã Fourcc phụ thuộc vào hệ điều hành đang được sử dụng cũng như định dạng video; nếu khai báo sai mã Fourcc thì việc lưu video sẽ thất bại. Mã Fourcc được sử dụng trong bài báo này là 'MJPG' với định dạng video là .avi. Nếu việc lưu video thất bại thì cần phải thử với các mã Fourcc khác trên trang web <https://www.fourcc.org/codecs.php>.

3. Phương pháp xử lý giọng nói

3.1. Thư viện âm thanh MARF

Trong phạm vi bài báo này, chúng tôi nhận thấy việc xác minh người nói (speaker identifier) có ý nghĩa quan trọng hơn trong việc bảo mật của hệ thống thay vì nhận diện giọng nói, vì thế chúng tôi quyết định chọn những phần mềm, thiết bị liên quan tới công việc này.



Hình 5. Giới thiệu về MARF

Giải pháp phần mềm được đưa ra trong bài báo này là thư viện MARF (Modular Audio Recognition Framework) đã được chỉnh sửa lại để có thể sử dụng với ngôn ngữ lập trình python và hoạt động được trên máy tính Raspberry Pi.

MARF là một nền tảng nghiên cứu mã nguồn mở và là một tập hợp của các thuật toán về xử lý giọng nói/ âm thanh/ ngôn ngữ được viết trên nền tảng java, được sắp xếp vào trong một mô-đun và một cái sườn có thể mở rộng được bằng việc thêm vào các thuật toán mới. MARF có thể chạy trên mạng (network) hoặc có thể hoạt động như một thư viện thêm vào các ứng dụng, hoặc có thể dùng như một nguồn để nghiên cứu và phát triển.

3.2. Cài đặt thư viện

Thư viện MARF được tùy biến thành thư viện piwho sử dụng cho ngôn ngữ lập trình python. Về bản chất, thư viện MARF được viết trên nền tảng java, vì vậy để sử dụng được thư viện chúng ta cần nền java tối thiểu là phiên bản 1.7. Đối với máy tính Raspberry pi cũng như nhiều dạng máy tính linux khác, trong nhân máy đã được cài sẵn java. Để kiểm tra phiên bản của nền tảng java, chúng ta sử dụng câu lệnh:

```
$ java -version
```

Để ghi âm phục vụ cho nhận diện người nói, chúng ta cần một thư viện âm thanh dành cho python là pyaudio:

```
$ pip install pyaudio
```

Cuối cùng là câu lệnh cài đặt thư viện:

```
$ pip install piwho
```

3.3. Huấn luyện máy nhận diện

Việc huấn luyện máy nhận diện không quá khó khăn, người dùng chỉ việc lặp lại việc nói một câu nào đó nhiều lần với các âm điệu khác nhau, thường là từ năm đến mười lần. Sau khi các đoạn âm thanh được ghi âm xong, máy tính sẽ bắt đầu huấn luyện trên cả thư mục chứa các đoạn âm thanh đã được ghi âm và lưu tập huấn luyện được vào một tập tin.

Yêu cầu của việc huấn luyện:

- Số lượng huấn luyện: Tối thiểu hai người.
- Định dạng tệp âm thanh huấn luyện: PCM signed (wav).

- Tốc độ mẫu âm thanh: 8.000Hz (8Khz).
- Độ phân giải mẫu: 16bit.
- Kênh thu âm: 1 (mono).

Nếu tập âm thanh không đúng định dạng trên thì nó sẽ được chuyển về đúng định dạng và ghi đè lên tập tin gốc. Sau khi huấn luyện lần đầu tiên thì một mô-đun có đuôi mở rộng là “.gzbin” và một tập tin “speaker.txt” được tạo ra tại thư mục làm việc hiện tại. Tập tin “speaker.txt” chứa thông tin tên người nói và số ID của người nói. Tập tin “*.gzbin” chứa các đặc tính được phân tích ra từ tập âm thanh huấn luyện. Mỗi lần huấn luyện, tập tin “*.gzbin” sẽ được cập nhật lại.

Để sử dụng thư viện piwho trong chương trình python chúng ta phải nhập thư viện:

```
from piwho import recognition
recog = recognition.SpeakerRecognizer()
```

Nếu tập âm thanh dùng để huấn luyện và nhận diện nằm ở một thư mục khác thì chúng ta phải chỉ rõ ra địa chỉ của chúng. Ví dụ các tập tin nằm ở địa chỉ ‘/home/pi/recordings’:

```
recognition.SpeakerRecognizer('/home/pi/recordings/')
```

Để huấn luyện một tập dữ liệu âm thanh mới sử dụng câu lệnh:

```
recog.train_new_data()
```

3.4. Nhận diện người nói

Sau khi thực hiện xong việc huấn luyện thì bước tiếp theo là thử nghiệm bằng việc chạy trình nhận diện người nói. Để thực hiện được điều này có một số yêu cầu như sau:

- Phải đảm bảo tập tin “*.gzbin” và “speaker.txt” nằm trong cùng thư mục, nơi chứa đoạn code python được thực thi.
- Kiểm tra “speaker.txt” không phải là file trống.

Thực hiện việc nhận diện bằng hàm identify_speakers() và kết quả trả về ‘name’ dưới đây là danh sách của hai người nói có điểm số tốt nhất trong tập huấn luyện. Người đầu tiên là người nói gần giống nhất, tức là có điểm số thấp nhất:

```
name = recog.identify_speaker()
```

4. Giải pháp truyền thông

4.1. Truyền hình ảnh qua mạng LAN

Mạng LAN (Local Area Network) là tên của một loại mạng cục bộ. Các thiết bị kết nối trong mạng này có thể chia sẻ thông tin, dữ liệu với nhau mà không bị can thiệp, tấn công từ các thiết bị ngoài mạng. Việc truyền hình ảnh qua mạng LAN giúp các thiết bị trong mạng có thể xem được hình ảnh từ camera ở cửa gửi về. Để truyền được hình ảnh qua mạng LAN trên máy tính Raspberry Pi (sử dụng hệ điều hành Debian) chúng tôi sử dụng một dịch vụ (service) đi kèm, đó là motion.

Để cài đặt dịch vụ, chúng ta nhập vào terminal câu lệnh: sudo apt-get install motion.

Để bắt đầu sử dụng dịch vụ chúng ta gõ vào terminal câu lệnh: sudo service motion start. Để dừng dịch vụ sử dụng câu lệnh: sudo service motion stop. Nếu việc truyền

hình ảnh không như ý muốn, chúng ta có thể thay đổi các thông số trong tập tin cấu hình của dịch vụ bằng câu lệnh: `sudo nano/etc/motion/motion.conf`, lúc đó tập cấu hình được mở ra và chúng ta chỉnh sửa lại dựa trên chú thích trước mỗi thông số, hoặc tìm hiểu để biết thêm cách chỉnh sửa các thông số cho phù hợp.

4.2. Giao tiếp với thiết bị khác qua giao thức MQTT

MQTT là một giao thức truyền tin hiệu quả, đáng tin cậy dựa trên nền tảng giao thức TCP/IP được sử dụng phổ biến hiện nay. MQTT thích hợp với việc truyền tải những dữ liệu nhỏ, yêu cầu tốc độ nhanh và sự đồng bộ dựa trên phương pháp đăng ký/xuất bản (subscribe/publish) lên một kênh bất kỳ.

MQTT có hỗ trợ ngôn ngữ python. Để cài đặt gói giao thức MQTT cho python, sử dụng câu lệnh `pip install paho-mqtt`. Để thêm thư viện MQTT vào chương trình python, sử dụng `import paho.mqtt.client as mqtt`. Để đăng nội dung lên một kênh (topic), sử dụng `client.publish("topic", data, qos=0, retain=False)`. Để đăng ký nội dung từ một kênh bất kỳ, sử dụng câu lệnh `client.subscribe("topic", qos = 0)`. Chú ý, yêu cầu kênh đó phải tồn tại, có nghĩa là chúng ta phải chỉ rõ địa chỉ máy chủ nơi mà kênh được thiết lập và tồn tại, sử dụng `client.connect(địa chỉ máy chủ, cổng, thời gian giữ tin)`. Ví dụ: `client.connect("iot.eclipse.org", 1883, 60)`.

5. Mô hình thực nghiệm

Để kiểm nghiệm những đề xuất, giải pháp của bài báo, chúng tôi đã xây dựng một mô hình thực tế và đã thử nghiệm các phương án nhiều lần. Chương trình xác minh khuôn mặt đã xác minh đúng 98,55% số lần thử. Với chế độ bảo mật bằng bàn phím, tỷ lệ thành công là 100%, chế độ bảo mật bằng giọng nói có xác suất thành công là 91%, và chế độ bảo mật bằng thẻ RFID đạt tỷ lệ thành công 100%. Với các chế độ kết hợp, RFID kết hợp xác minh khuôn mặt đạt tỷ lệ thành công là 96%; RFID kết hợp giọng nói đạt 89%, xác minh khuôn mặt kết hợp xác minh giọng nói đạt 88%. Như vậy, các chương trình cho kết quả với độ chính xác cao, chế độ RFID kết hợp với xác minh khuôn mặt là chế độ tốt nhất và nên được sử dụng.



Hình 6. Hình ảnh thiết kế mô hình và các thiết bị mặt trước



Hình 7. Mặt trước mô hình thực nghiệm thực tế



Hình 8. Mặt sau mô hình thực nghiệm thực tế

6. Kết luận

Bài báo đã trình bày phương án thiết kế hệ thống điều khiển cửa tự động sử dụng công nghệ xử lý giọng nói và công nghệ xử lý ảnh và các phương thức bảo mật kết hợp. Các thuật toán xử lý hình ảnh, nhận dạng giọng nói đã được trình bày chi tiết. Các thiết kế trình bày trong bài báo giúp làm chủ công nghệ, đưa ra giải pháp để chế tạo hệ thống điều khiển cửa thông minh có chất lượng tốt, nhiều ưu điểm so với các hệ thống đã được ứng dụng trong thực tiễn. Đồng thời, một mô hình thực tế đã được xây dựng để kiểm nghiệm các đề xuất của bài báo. Qua quá trình thử nghiệm thực tế, chúng tôi nhận thấy rằng mô hình hoạt động có hiệu quả, đáp ứng được yêu cầu đặt ra và có thể áp dụng được trong thực tiễn.

TÀI LIỆU THAM KHẢO

- [1] Amos, B., Ludwiczuk, B. & Satyanarayanan, M., 2016, *OpenFace: A general-purpose face recognition library with mobile application*, CMU-CS-16-118, CMU School of Computer Science.
- [2] Australian Border Force, n.d. The Department of Immigration and Border Protection, [Online] Available at: <https://www.border.gov.au> [Accessed 20 August 2016]
- [3] Bradski, G., 2000, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*.
- [4] Chui, n.d. *CHUI*, [Online] Available at: <https://www.getchui.com/> [Accessed 20 August 2016]
- [5] Gonzalez, R.C. & E., W.R., 2007, *Digital Image Processing*. 3rd ed. Pearson.
- [6] King, D.E., 2009, "Dlib-ml: A Machine Learning Toolkit", *Journal of Machine Learning Research*, 10, page 1755-58.
- [7] Schroff, F., Kalenichenko, D. & Philbin, J., 2015, "FaceNet: A Unified Embedding For Face Recognition And Clustering", In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015.*, 2015.
- [8] Xiamen Surmount Industry Co., Ltd., n.d, *Xiamen Surmount Industry Co., Ltd.*, [Online] Available at: <http://www.xmsurmount.com/en/> [Accessed 20 August 2016]

(BBT nhận bài: 17/12/2016, hoàn tất thủ tục phản biện: 03/02/2017)