hệ thống Camera Phát hiện té Ngã

Camera-based Fall Detection System

SVTH: Phùng Hữu Tài, Trần Văn Vũ, Đào Việt Hoàng

Lớp 16PFIEV3, Khoa FAST, Trường Đại Học Bách Khoa Đà Nẵng; Email: [phunghuutai421998@gmail.com](mailto:phunghuutai421998@gmail.com), [tranvu1998@outlook.com](mailto:tranvu1998@outlook.com), dvhoang2106@gmail.com

GVHD:TS Bùi Thị Thanh Thanh

Khoa CNTT, Trường Đại Học Bách Khoa Đà Nẵng; Email: bttthanh@dut.udn.vn

**Tóm tắt -** Hệ thống Camera phát hiện té ngã sử dụng 1 camera để quay video, Raspberry Pi 3 làm bộ xử lý trung tâm để xử lý video, để phát hiện ngã và gửi trạng thái, bình thường hoặc ngã , đến một máy chủ, nền tảng Firebase làm máy chủ, dùng ổ đĩa đám mây Dropbox để lưu trữ lịch sử video để kiểm tra trong trường hợp cần thiết và ứng dụng Android để gửi đi thông báo.

**Từ khóa -** Phát hiện té ngã; Raspberry Pi 3; stream thời gian thực; Optical Flow; SVM.

**Abstract -** The fall detection camera system uses 1 camera to record video, Raspberry Pi 3 is the central processor for video processing, to detect falls and send status, normal or falling, to a server, platform Firebase serves as a server, using Dropbox cloud drive to store video history to check in case of need and Android app to send notifications.

**Key words -** Fall-detection; Raspberry Pi 3; realtime stream; Optical Flow; SVM.

# Đặt vấn đề

Té ngã là sự mất thăng bằng ngoài ý muốn khiến cho cơ thể bất ngờ rơi xuống mặt đất, sàn nhà. Đặc biệt với người cao tuổi, té ngã là chuyện thường gặp. Theo nghiên cứu của Hoa Kỳ, cứ 3 người trên 65 tuổi thì có 01 người bị té ngã ít nhất 01 lần/năm và hơn 1 nửa trong số đó té ngã hơn 02 lần/năm [1]. Mặc dù hầu hết các té ngã không gây thương tích nhưng hậu quả để lại có thể nghiêm trọng. Nếu bạn bị gãy xương, nó có thể dẫn đến tàn tật lâu dài. Xương gãy không phải lúc nào cũng lành hoàn toàn khi bạn già đi và một tổn thương nặng có thể đồng nghĩa với việc bạn cần phải được hỗ trợ liên tục để duy trì cuộc sống.

Phát hiện té ngã sớm là điều rất cần thiết ở những người cao tuổi và những người có nguy cơ đột quỵ để giảm thiểu đi rủi ro mà té ngã mang tới. Ở trong bài báo này, nhóm tác giả đề xuất một hệ thống nhận dạng tự động sử dụng các thuật toán Optical Flow và Support Vector Machine cho kết quả tốt trên tập dữ liệu Le2i.

# Cơ sở lý thuyết

## Tổng quan

Video gốc được thu từ camera

Xử lý và trích xuất đặc trưng

Chương trình nhận dạng té ngã

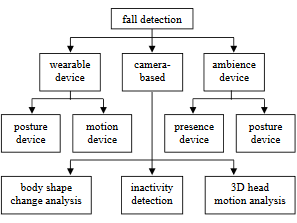
Cập nhật trạng thái nhận dạng lên Firebase

Stream hình ảnh về app trên Android

Lưu trữ video trên Dropbox

App Android thông báo khi trạng thái là “NGÃ”, xem video được stream và được lưu trữ từ hệ thống

Nhận dạng té ngã đang là đề tài rất được quan tâm trong giới nghiên cứu vì tính thực tiễn của nó. Có nhiều phương pháp đã được đưa ra để nhận diện té ngã. Theo [2], các phương pháp nhận dạng được phân loại như sau:



Mỗi phương pháp đều có ưu và nhược điểm riêng, trong đó, các phương pháp dựa trên việc phân tích hình ảnh từ camera cho thấy khả năng áp dụng thực tiễn cao, khi có thể nhận biết được các tư thế ở khoảng cách xa và tận dụng trong cả hạ tầng camera hiện thời.

Ngoài ra, thời gian từ lúc té ngã đến lúc người gặp tai nạn trở về trạng thái bình thường rất quan trọng. Yếu tố này phản ảnh tình trạng nghiêm trọng mà người bị ngã nghiêm trọng như thế nào. Nếu một người bị té ngã và có thể đứng dậy ngay sau đó thì có nghĩa tình trạng của người tai nạn không quá nghiêm trọng, không phải trong tình trạng cần cấp cứu, có thể cần phải đến bác sĩ để kiểm tra lại cho chắc chắn. Ngược lại, nếu như người bị ngã phải chật vật đứng dậy hoặc không thể ngồi hay đứng dậy thì có nghĩa người bệnh đang cần cấp cứu. Vì vậy cần có cơ chế thông báo đến người có trách nhiệm với người tai nạn để có thể xử lý nhanh nhất có thể kèm theo thông tin hình ảnh về tình trạng của người bị ngã.

## Mô hình đề xuất

Dựa vào những điều đã nêu trong phần Tổng quan, bài báo đề xuất mô hình Hệ thống nhận dạng té ngã dùng camera:

Tín hiệu hình ảnh được thu từ module camera của Raspberry Pi 3 model B+. Raspberry có nhiệm vụ chuyển tín hiệu hình ảnh lên website dể ứng dụng điện thoại có thể theo dõi và phát hiện các sự cố từ xa. Đồng thời hệ thống còn cho phép lưu trữ video trên dropbox mỗi một giờ để thuận tiện cho việc xem lại và kiểm tra các sự cố đã xảy ra. Trong quá trình nhận diện té ngã, các frame sẽ được vào xử lí bằng các thao tác xử lí ảnh cơ bản, trừ nền để tách đối tượng ra khỏi nền. Thuật toán Optical Flow để phân biệt các đối tượng chuyển động và đối tượng trong tĩnh trong bối cảnh. Sau khi đã tách đối tưởng ra khỏi bối cảnh, các thông tin về đối tượng sẽ được sử dụng làm đặc trưng trong giai đoạn phân loại, phương pháp phân loại được sử dụng trong bài báo là Support Vector Machine vì tính hiệu quả trong phân loại hai lớp với chi phí thấp phù hợp với hệ thống thời gian thực.

## Optical Flow

Thuật toán Optical Flow được cài đặt dựa trên ý tưởng khai triển đa thức của Gunnar Farneback được mô tả trong [3]. Trong xử lí ảnh, giá trị tọa độ của mỗi điểm ảnh được ánh xạ lên một không gian màu xác định. Để ước lượng độ dịch chuyển của điểm ảnh, trước tiên ánh xạ trên được xấp xỉ bằng khai triển đa thức:

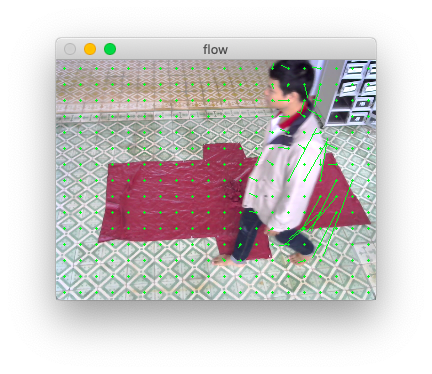
(1)

(2)

Do , nên từ (1) và (2) ta suy ra được độ dịch chuyển của điểm ảnh **d**.

(3)

Giá trị của vector dịch chuyển **d** cho điểm được tốc độ và hướng dịch chuyển của điểm ảnh. Dựa trên các vector này, vùng dịch chuyển của đối tượng sẽ được xác định dựa trên việc thiết lập các giá trị ngưỡng về cường độ.



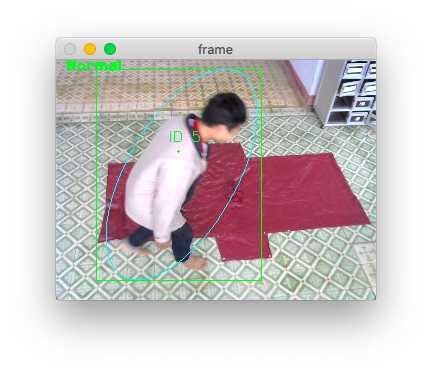
**Hình 1:** Optical Flow

Hình 1 thể hiện các vector dịch chuyển được tính toán dựa vào Optical Flow, có thể thấy được vector dịch chuyển ở phần thân dưới và chân của đối tượng có độ lớn lớn hơn các vector ở khu vực khác, điều này là hợp lí vì khi di chuyển thì chân là khu vực có tốc độ dịch chuyển cao hơn các khu vực còn lại.

Với việc sử dụng Optical Flow, hiệu quả nhận dạng được khá tốt nhưng trong một vài trường hợp như độ sáng môi trường thay đổi nhiều, vật thể có bóng thì phương pháp này có phần thể hiện không tốt, để khắc phục nhóm tác giả đã sử dụng một vài kĩ thuật xử lí ảnh trước khi đưa vào thuật toán phát hiện chuyển động như phép mở ảnh và phép đóng ảnh.

## Nhận dạng té ngã

Sau khi đã xác định được đối tượng chuyển động, thông tin về vùng chuyển động sẽ được trích xuất làm đặc trưng cho các phương pháp phân loại. Các đặc trưng có thể kể đến là đường bao bên ngoài đối tượng [4], ảnh lịch sử chuyển động (Motion History Image) [5]. Tuy nhiên, các đặc trưng có kích thước khá lớn dẫn đến chi phí để xử lí và tính toán là đáng kể. Trong hệ thống đề xuất, để tăng tốc độ nhận dạng té ngã trên nền tảng phần cứng hạn chế, các đặc trưng được sử dụng bao gồm có tỉ lệ kích thước vùng chuyển động, tốc độ thay đổi tỉ lệ kích thước và độ nghiêng của đối tượng so với mặt đất nằm ngang.



**Hình 2:** Đặc trưng phân loại

Các thông tin về đặc trưng trên được trích xuất từ việc xác định được các hình chữ nhật và ellipse bao quanh vùng chuyển động. Tỉ lệ kích thước và tốc độ thay đổi tỉ lệ này dựa trên tỉ số dựa chiều ngang và chiều dọc của hình chữ nhật, góc nghiêng có được nhờ xác định góc giữa trục chính ellipse và trục hoành. Các đặc trưng trên đảm bảo được việc phân loại được hành động thông thường với sự cố té ngã, ví dụ như tốc độ thay đổi tỉ lệ kích thước vùng chuyển động có ý nghĩa lớn trong phân loại hành vi ngồi và việc ngã của đối tượng. Các đặc trưng trên tạo nên một vector đặc trưng có độ dài 3. Ứng với mỗi khung hình, một vector đặc trưng sẽ được trích xuất.

Support Vector Machine (SVM) được sử dụng làm phương phân loại cho hệ thống vì được biết đến như một phương pháp phân loại hiệu quả trong trường hợp phân loại hai lớp với chi phí thấp. Ý tưởng chính của SVM là tìm một mặt phẳng giúp phân chia không gian đặc trưng thành hai phần. Một phần không gian dành cho các dáng đi bình thường, phần còn lại dành cho các dáng đi bất thường. Mặt phẳng phân lớp định nghĩa:

(4)

Trong đó, là vector trọng số, là một vector trong không gian đặc trưng, là hệ số lệch, là hàm có tập đích là không gian đặc trưng.

Bằng cách định nghĩa nhãn của mẫu nghiên cứu là:

(5)

Bằng kĩ thuật kernel, SVM có thể phân loại trong trường hợp không gian đặc trưng dược phân loại không tuyến tính.

# Kết quả nghiên cứu và khảo sát

## Cơ sở dữ liệu

Cơ sở dữ liệu sử dụng để đánh giá hệ thống được đề xuất trong báo cáo được trích từ tập dữ liệu được xây dựng bởi Trung tâm Nghiên cứu Khoa học Quốc gia Pháp (CNRS) [6]. Một tập hợp gồm 29 video được chọn để đánh giá cho khả năng của hệ thống bao gồm độ chính xác trong phân loại và tốc độ thực thi. Mỗi video được quay có tốc độ 25 khung hình / giây và độ phân giải là 320x240.

Các khung hình được dán nhãn positive đại hiện cho hành vi ngã của đối tượng. Ngược lại, các khung hình negative được chọn để thể hiện các hành vi thông thường trong cuộc sống hàng ngày như ngồi, quỳ, cúi xuống.

Trong giai đoạn huấn luyện có tổng cộng 2000 khung hình negative, 400 khung hình positive. 400 khung hình negative và 100 khung hình positive được sử dụng để đánh giá độ chính xác trong phân loại của hệ thống.

Các độ đo được áp dụng là Accuracy (độ chính xác), Precision và Recall. Accuracy là tỉ số giữa các mẫu phân loại đúng trên kích thước mẫu. Precision được định nghĩa là số các mẫu dán nhãn dương được phân loại đúng trên toàn bộ số các mẫu được phân loại dương. Tương tự Recall là tỉ lệ giữa số mẫu dán nhãn dương được phân loại đúng trên số các mẫu được dán nhãn dương. Precision và Recall giúp đánh giá tính hiệu quả trong phân loại trong trường hợp tỉ lệ số mẫu giữa các lớp không cân bằng.

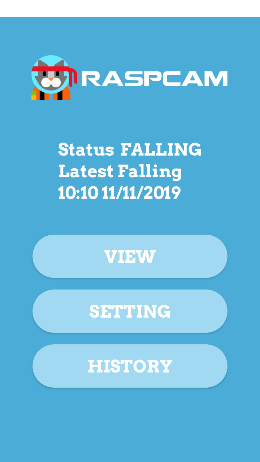
## Kết quả

|  |  |  |  |
| --- | --- | --- | --- |
| **Database** | **Accuracy** | **Precision** | **Recall** |
| Le2i | 0.954 | 0.913 | 0.537 |

**Bảng 1**: *Kết quả thực nghiệm*

Kết quả thực nghiệm cho thấy trên toàn bộ tập dữ liệu, số các khung hình được phân loại đúng là khá tốt. Độ đo Precision cao cho thấy xác suất hệ thống phân dự đoán sai các hành vi té ngã là thấp tuy Recall không tốt thể hiện nhiều khả năng ra các hành vi té ngã sẽ bị bỏ lỡ.

Sau khi hệ thống phân loại thành công các hành vi té ngã, tín hiệu về nhãn các hành vi sẽ được gửi đi nhờ vào Firebase đến thiết bị có cài ứng dụng phát hiện té ngã.



**Hình 3:** Màn hình ứng dụng thế hiện rằng hiện tại đang có một sự cố té ngã diễn ra.

# Hướng phát triển

Tuy đã xây dựng một mô hình tương đối hoàn thiện với các tính năng thiết thực với người dùng, hệ thống đề xuất vẫn không tránh được những hạn chế. Dễ nhận thấy nhất số lượng các mẫu dán nhãn negative bị nhầm lẫn là khá lớn do đó cần có xây dựng một tập dữ liệu tốt và áp dụng các phương pháp phân loại khác nhau để tìm ra được phương pháp tối ưu nhất.

Việc phát hiện chuyển động của hình ảnh vẫn còn bị ảnh hưởng đáng kể bởi yếu tố môi trường như ánh sáng, các đối tượng bị che khuất một phần không thể phân loại dựa trên hệ thống đề xuất. Với hướng đi này, các thuật toán về chuỗi thời gian có thể giải quyết bằng cách suy diễn ra thông tin ứng với mỗi khung hình.

Việt lắp đặt và đóng gói camera cần được cải thiện nhiều để sẵn sàng áp dụng vào thực tế.

# Kết luận

Nhóm tác giả đã đề xuất một hệ thống hoàn chỉnh khả năng theo dõi và nhận diện hành vi té ngã với độ chính xác có thể chấp nhận. Tuy còn nhiều hạn chế, nhưng hướng phát triển cho cách tiếp cận bài toán phân loại hành vi té ngã này hứa hẹn sẽ có nhiều đột phá trong thời gian không xa.

Tài liệu tham khảo

Tinetti ME, Kumar C. The patient who falls: “It’s always a trade-off.” JAMA. 2010;303(3):258–66

Lingmei Ren, Yanjun Peng. Research of Fall Detection and Fall Prevention Technologies: A Systematic Review, IEEE Access (Volume: 7), 77702 – 77722

Gunnar Farneback, Two-Frame Motion Estimation Based on Polynomial Expansion,13th Scandinavian Conference, SCIA, 2003

Subhash Chand Agrawal, Rajesh Kumar Tripathi, Anand Singh Jala. Human-fall detection from an indoor video surveillance, 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)

Shaikh Abdul Waheed, P. Sheik Abdul Khader. A Novel Approach for Smart and Cost Effective IoT Based Elderly Fall Detection System Using Pi Camera, 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)

<http://le2i.cnrs.fr/Fall-detection-Dataset>