ĐỀ XUẤT HỆ THỐNG TRONG NHẬN DẠNG CỬ CHỈ, HÀNH ĐỘNG SỬ DỤNG TRÍ TUỆ NHÂN TẠO CHO CÁC ỨNG DỤNG NHÀ THÔNG MINH

Nguyễn Hữu Phát*, Nguyễn Thị Thu Hương†
*Bộ môn Mạch và Xử lý tín hiệu, Viện Điện tử viễn thông, Đại học Bách Khoa Hà Nội
† Viện Điện tử viễn thông, Đại học Bách Khoa Hà Nội

Tóm tắt: Bài báo nghiên cứu một hệ thống để nhận dạng cử chỉ, hành động trong nhà thông minh. Phương pháp mà chúng tôi đề xuất dựa trên các việc sử dụng mobilenetV2 trích xuất đặc trưng kết hợp với mạng SSD (Single Shot Detector). Chúng tôi sử dụng năm loại cử chỉ đứng lên, ngồi xuống, ngửa người về phía sau, đi giầy, và phẩy tay. Trong ứng dụng này nguồn cấp dữ liệu từ camera của thiết bị di động sau đó thực hiện chạy để phát hiện đối tượng. Kết quả đối tượng trên khung hình bằng hộp giới hạn. Mặc dù kết quả đạt yêu cầu đặt ra với độ chính xác trên 90 phần trăm. Tuy nhiên trong một số trường hợp độ chích xác còn phụ thuộc nhiều vào số lượng hình ảnh đào tạo và độ phân giải của chúng.

Từ khóa: MobilenetV2, SSD (Single Shot Detector), nhận dạng đối tượng, cử chỉ, hành động, dáng điệu.

I. ĐẶT VẤN ĐỀ

Ngày nay, nhờ có sự tiến bộ của khoa học kỹ thuật, máy tính dần trở thành công cụ được sử dụng rộng rãi trong công việc cũng như đời sống con người. Theo đó sự tương tác giữa con người và máy tính cũng càng đa dạng. Hiện nay, con người chủ yếu tương tác với máy tính qua bàn phím và chuột nhưng với sự phát triển nhanh chóng của khoa học kỹ máy tính các tương tác mới được tìm ra như sử dụng giọng nói, cử chỉ mang lại sự trực quan dễ dàng hơn cho người sử dụng. Theo đó các hệ thống tương tác giữa con người và máy tính được tập trung nghiên cứu.

Việc sử dụng cử chỉ, hành động người được xem là một ý tưởng hiệu quả để con người giao tiếp với nhau trong thế giới thực. Hành động của một sự kết hợp của nhiều bộ phận khác nhau trên cơ thể mang hàm ý tuyền đạt thông tin. Do đó trong bài báo này chúng tôi sẽ nghiên cứu phát triển hệ thống nhận dạng cử chỉ, hành động trong nhà thông minh. Đây là bước tiếp theo phát triển của bài báo [1] đã công bố trong hội thảo NICS.

Mục tiêu của bài báo là thực hiện tìm hiểu cách tương tác giữa con người và máy tính giúp điều khiển các thiết bị điện tử. Trong bài báo này chúng tôi sử dụng các hành động như đứng lên, ngồi xuống, ngửa người về phía sau, đi giầy, và phẩy tay để thực hiện việc tương tác giữa con người và máy tính. hệ thống chuyển sang định dạng tensorflow lite để có thể dễ dàng chạy trên một thiết bị thông minh như là điện thoại di động giúp giảm băng thông phía máy chủ, giảm độ trễ và cải thiện tốc độ phản hồi của trí tuệ nhân tạo (AI). Qua đó giảm chi phí lưu lượng truy cập di động cho người dùng vì không cần phải tải một lượng lớn dữ liệu thô trên máy tính.

Phần còn lại của bài báo được trình bày như sau. Trong phần II chúng tôi sẽ khảo sát qua về các hệ thống hiện có. Trong phần III và phần IV, chúng tôi lần lượt trình bày mô hình và đánh giá kết quả của mô hình để ra. Cuối cùng, chúng tôi kết luận bài báo trong phần V.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Nhận dạng hành động là một trong số ứng dụng trong việc kiểm soát các thiết bị kỹ thuật số trong tương lai. Đây là một công nghệ tiên tiến trong ứng dụng nhà thông minh. Hiện nay nhiều công ty và các phòng nghiên cứu đang tích cực nghiên cứu mô hình công nghệ cao cho phép điều khiển màn hình mà không cần chạm vào thiết bị bằng công nghệ AI và được quan tâm hơn cả là nhận dạng hành động.

Có nhiều nghiên cứu để nhận dạng hành động [2]÷[9]. Trong [2] tác giả thực hiện nhận dạng theo bộ xương 3D trên bộ dữ liệu NTU-RGB + D, Kinetic. Tác giả trong [3] nhận dạng dựa trên mạng noron và bản đồ quỹ đạo (JTM). Giải pháp thực hiện theo [4] đề xuất sự kết hợp tuần tự của Inception-ResNetv2 và mạng bộ nhớ ngắn hạn (LSTM) để tận dụng phương sai thời gian để cải thiện hiệu suất nhận dạng. Độ chính xác nhận dạng đạt được là 95,9 và 73,5 phần trăm trên UCF101 và HMDB51. Các thuật toán học máy như biểu đồ định hướng cục bộ, máy vectơ hỗ trợ (SVM) [9]. Nhờ khả năng học tập, mạng lưới thần kinh không cần thiết lập tính thủ công trong quá trình mô phỏng quá trình học tập của con người và có thể thực hiện đào tạo các mẫu cử chỉ, hành động để tạo thành bản đồ nhận dạng phân loại mạng. Các mô hình học tập sâu được lấy cảm hứng từ các mô hình xử lý thông tin và giao tiếp được phát triển từ các hệ thống thần kinh sinh học, bao gồm các mạng lưới thần kinh với nhiều hơn một lớp ẩn. Họ có thể có được các đặc điểm của đối tượng học tập một cách dễ dàng và chính xác dưới đối tượng phức tạp và thể hiện hiệu suất vượt trội trong thị giác máy tính và xử lý ngôn ngữ tự nhiên (NLP) [7], [8]. Các hệ thống phát

Tác giả liên hệ: Nguyễn Hữu Phát
Email: phat.nguyenhuu@hust.edu.vn
Đến tòa soạn: 4/2020, chỉnh sửa: 6/2020, chấp nhận đăng: 6/2020

hiện đối tượng hiện đại là các biến thể của Faster R-CNN [7]. Trong một bài báo theo [5] các tác giả đã khám phá ý tưởng sử dụng các LSTM trên các bản đồ tính năng được đào tạo riêng biệt để xem liệu nó có thể nắm bắt thông tin tạm thời từ các clip hay không. Họ kết luận rằng việc gộp các tính năng phức tạp theo thời gian tỏ ra hiệu quả hơn LSTM xếp chồng lên nhau sau các bản đồ tính năng được đào tạo. Trong bài báo hiện tại, các tác giả xây dựng trên cùng một ý tưởng sử dụng các khối LSTM (bộ giải mã) sau các khối tích chập (bộ mã hóa) nhưng sử dụng đào tạo từ đầu đến cuối của toàn bộ kiến trúc. Họ cũng so sánh RGB và dòng quang là lựa chọn đầu vào và thấy rằng việc chấm điểm dự đoán có trọng số dựa trên cả hai đầu vào là tốt nhất. Mạng lưới phân đoạn tạm thời: Hướng tới thực tiễn tốt để nhận biết hành động sâu sắc [6]. Mạng tích chập sâu đã đạt được thành công lớn cho nhận dạng hình ảnh trong ảnh tĩnh. Tuy nhiên, để nhận dạng hành động trong video, lợi thế so với các phương pháp truyền thống không quá rõ ràng.

Tuy nhiên, có một số thách thức đối với nhận dạng hành động như sau:

Phát triển mẫu đào tạo:

Nhận dạng bằng cách sử dụng máy học đòi hỏi bộ dữ liệu mẫu phù hợp do chúng ta phải mất nhiều thời gian để thu thập dữ liệu để tạo ra các mẫu tiêu chuẩn.

Thời gian xử lý:

Chúng ta cần xử lý một lượng lớn dữ liệu. Do đó, với một mạng phải xử lý quá nhiều tham số với máy tích có cấu hình yếu sẽ xử lý chậm ảnh hưởng đến kết quả trong thời gian thực

Độ chính xác của phương pháp:

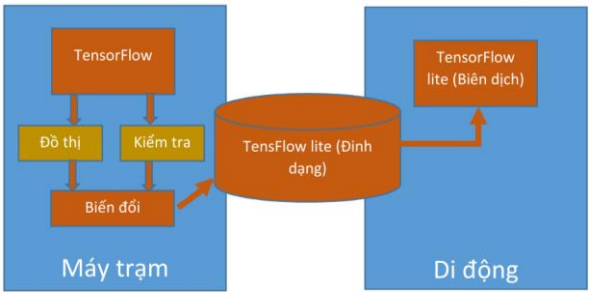
Đối với máy ảnh thông thường (webcam), độ chính xác bị ảnh hưởng bởi các điều kiện khác như ánh sáng, hình nền, tốc độ chuyển động của tay vì chúng tôi phải đưa ra một số giả định cho các ứng dụng.

Dựa trên kết quả phân tích ở trên, chúng tôi đề xuất một hệ thống nhận dạng hành động trên sự kết hợp giữa mạng mobilenetV2 kết hợp với mạng SSD để dễ dàng sử dụng trên các thiết bị nhúng có cấu hình yếu hơn.

III. GIẢI PHÁP THỰC HIỆN

A. *Tổng quan về hệ thống*

Hệ thống đề xuất được xây dựng dựa trên [10] để ứng dụng trong các mô hình nhà thông minh như trên hình 1.

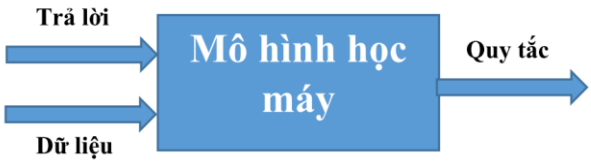


Hình 1. Mô hình tổng quan hệ thống thực hiện.

Mục tiêu của hệ thống này là xây dựng dữ liệu hành động đơn giản. Các cử chỉ được đề xuất bao gồm năm hành động, cụ thể là đứng lên, ngồi xuống, ngửa người về phía sau, đi giầy, và phẩy tay. Đầu tiên là trích xuất đặc trưng của dữ liệu đầu vào bằng mạng mobilenetV2 sau đó đưa vào mạng SSD để dự đoán kết quả. Kết quả thu được sau quá trình train được chuyển đổi sang định dạng tensorflow lite (.tflite) để dễ dàng chạy trên các thiết bị di động.

B. *Các bước thực hiện*

Tensorflow có thể được sử dụng cho việc tạo các mô hình, đào tạo, thao tác dữ liệu và thực hiện dự đoán như trên hình 2 dựa trên [11]. Vấn đề là, học máy, đặc biệt là học sâu, cần sức mạnh tính toán lớn. Có thể thực hiện đào tạo trong thiết bị di động và thiết bị nhúng, nhưng sẽ tốn rất nhiều thời gian. Vì vậy, sẽ sử dụng Tensorflow cho giai đoạn đào tạo và Tensorflow Lite có thể được sử dụng cho giai đoạn suy luận.



Hình 2. Mô hình nhận dạng cử chỉ hành động sử dụng tensorflow.

Phương pháp thực hiện quá trình huấn luyện gồm các bước sau:

Bước 1: Chuẩn bị dữ liệu của riêng bạn.

Bước 2: Gán nhãn cho dữ liệu.

Bước 3: Sử dụng mạng mobilemetV2 trích xuất đặc trưng.

Bước 4: Sử dụng đầu ra của mạng mobilenetV2 làm đầu vào của mạng SSD để phát hiện đối tượng.

Bước 5: Chuyển đổi sang định dạng Tensorflow Lite

Bước 6: Tạo app Android chạy mô hình Tensorflow Lite

Chi tiết các bước thực hiện được trình bày ở phần dưới đây.

Chuẩn bị dữ liệu của riêng bạn:

Trước hết chúng ta cần chuẩn bị dữ liệu từ nguồn trên mạng qua công cụ tìm kiếm của google và một phần bộ dữ liệu UCF101 [12] và BU203 [13] với các hành động đứng lên, ngồi xuống, ngửa người về phía sau, đi giầy, và phẩy tay như trên hình 3.

Hình 3. Chuẩn bị dữ liệu thực hiện [12],[13].

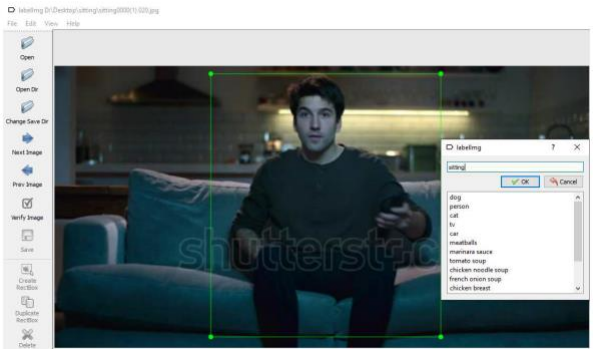
Số lượng các nhãn và các ảnh được thể hiện trên bảng I.

Bảng I. Số lượng ảnh và nhãn được chuẩn bị để thực hiện

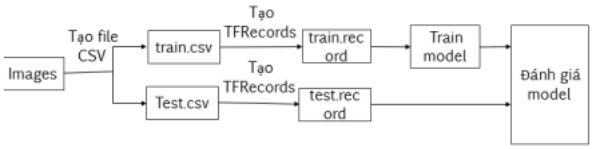
Nhãn	Số lượng ảnh train	Số lượng ảnh test
Đứng lên	400	100
Ngồi xuống	400	100
Ngửa tay	400	100
Vẫy tay	400	100
Đi giày		

Gán nhãn cho dữ liệu:

Trong bước này thực hiện xác định khối ROI của từng hành động dựa trên việc gán nhãn bằng tay. Trong bài báo này chúng tôi sử dụng một tool có sẵn là labeling. Quá trình này về cơ bản là vẽ các hộp xung quanh đối tượng trong ảnh. Trên hình 4 là một ví dụ sử dụng công cụ LabelImg tự động tạo một tệp XML mô tả vị trí đối tượng trong ảnh.



Hình 4. Một ví dụ về gán nhãn dữ liệu.



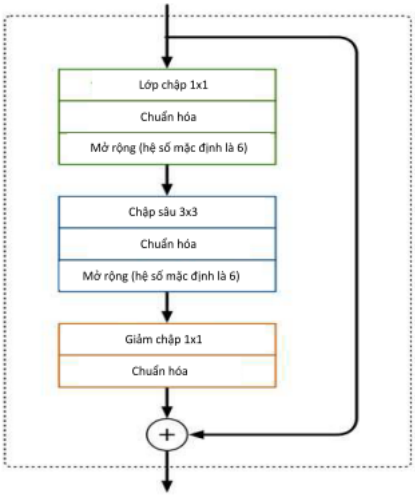
Hình 5. Mô hình chi tiết thực hiện gán nhãn.

Những giá trị thu được được thực hiện như trên hình 5 dựa trên [14]. Sau khi gán nhãn dữ liệu chia dữ liệu thành các tệp train/test. Chuyển đổi các tệp XML thành các tệp CSV và sau đó tạo TFRecords từ các tệp này. Tệp train TFRecords này được đưa để đào tạo mô hình. Cuối cùng các giá trị được đưa vào mô hình để đánh giá.

Trích xuất đặc trưng:

Ảnh đầu vào sau khi đã được gán sẽ được lưu dưới định dạng csv tiếp đến được chuyển thành định dạng record trong tensorflow. Ở đây sử dụng hai mạng MobilenetV2+SSD trong tensorlow để thực hiện việc nhận dạng hành động.

Trong phần trích xuất đặc trưng sẽ sử dụng mạng MobilenetV2 dựa trên [15], [16]. Mạng MobilenetV2 sử dụng các tích chập phân tách theo chiều sâu. Các khối được xây dựng giống như hình 6.

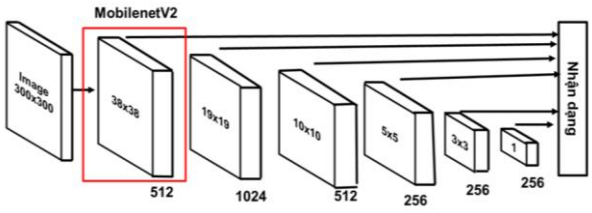


Hình 6. Mô hình mạng MobilenetV2.

Lớp chập đầu tiên là một tổ hợp 1×1 mục đích của nó mở rộng số lượng kênh trong dữ liệu trước khi đi vào tích chập sâu. Dữ liệu được mở rộng được đưa ra bởi hệ số mở rộng. Hệ số mở rộng mặc định là 6. Lớp chập theo độ sâu dùng để lọc đầu vào cuối cùng là lớp chập 1×1 làm cho số lượng kênh nhỏ hơn hay còn gọi là projection layer hoặc nút cổ chai. Nó đưa dữ liệu với số lượng kích thước (kênh) cao thành một thang đo với số lượng kích thước thấp hơn nhiều. Lớp này còn giảm dữ liệu chảy qua mạng.

Sử dụng mạng Single Shot Detector (SSD) để phát hiện:

Mạng SSD cơ sở là mạng VGG16, theo sau là các lớp multibox conv [17]÷[20]. SSD có hai thành phần: mô hình xương sống và đầu SSD. Mô hình xương sống thường là một mạng phân loại hình ảnh được đào tạo trước như là một trình trích xuất tính năng. Ở đây chúng tôi sử dụng mạng mobolenetV2. Đầu SSD chỉ là một hoặc nhiều lớp chập được thêm vào đường trục này. Các đầu ra được hiểu là các hộp giới hạn và các lớp đối tượng ở vị trí không gian của các kích hoạt lớp cuối cùng [21] như trên hình 7.



Hình 7. Model of SSD [17],[18].

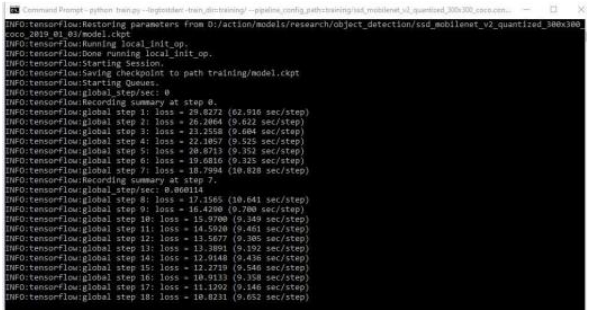
Thay vì sử dụng cửa sổ trượt, SSD chia hình ảnh bằng cách sử dụng lưới và mỗi ô lưới có trách nhiệm phát hiện các đối tượng trong vùng đó của hình ảnh. Các đối tượng phát hiện chỉ đơn giản là dự đoán lớp và vị trí của một đối tượng trong vùng đó. Nếu không có đối tượng nào hiện diện, chúng tôi coi nó là lớp nền và vị trí bị bỏ qua. Mỗi ô lưới có thể xuất vị trí và hình dạng của đối tượng mà nó chứa. Chi tiết xem thêm trong [21].

Chuyển đổi thành định dạng tensorflow lite (TSL):

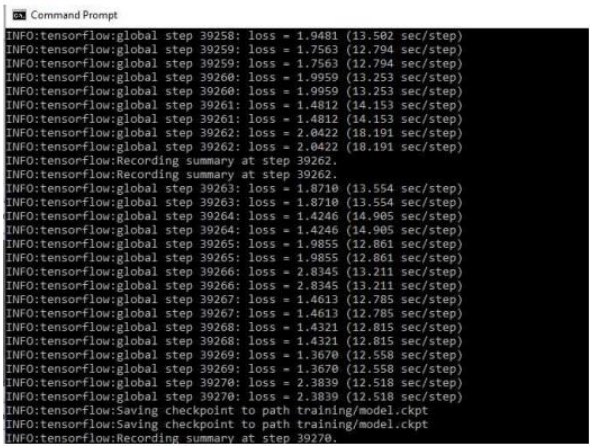
Tensorflow lite là giải pháp gọn nhẹ của tensorflow cho thiết bị di động và thiết bị nhúng. Nó cho phép chạy các mô hình học máy trên thiết bị di động. Quá trình thực hiện cho mô hình này thể hiện trên hình 8.

Hình 8. Mô hình tensorflow lite dựa trên [11].

IV. KẾT QUẢ ĐẠT ĐƯỢC

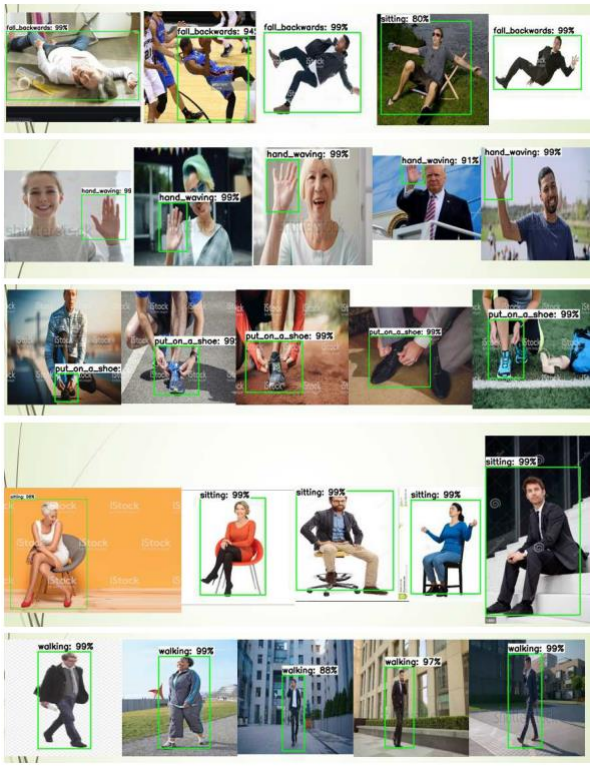


Hình 9. Bắt đầu chạy mô hình.



Hình 10. Kết quả huấn luyện với tập năm cử chỉ.

Hình 11. Kết quả huấn luyện với tập năm cử chỉ với tensorflow.



Hình 12. Kết quả huấn luyện với tập năm cử chỉ với tensorflow lite.

Bảng II. Đánh giá hiệu năng hai mô hình Tensorflow và Tensorflow lite

Mô hình	Độ chính xác (phần trăm)	Bộ nhớ sử dụng (MB)	CPU(phần trăm)
Tensorflow	82	317,9	76,7
Tensorflow Lite	98	121,8	30,1

Thực hiện huấn luyện với tập năm hành động nêu trên chúng ta có kết quả như trên hình 9 và 10.

Thực hiện nhận dạng tập năm hành động nêu trên với Tensorflow. Kết quả thực hiện hoạt động như trên hình 11.

Tiếp tục thực hiện nhận dạng tập năm hành động nêu trên với Tensorflow lite. Kết quả thực hiện hoạt động như trên hình 12.

Bên cạnh đó chúng tôi cũng thực hiện đánh giá hiệu năng và thời gian xử lý của hệ thống thông qua việc chạy video ở hai mô hình tensorflow và tensorlow lite trên máy tính với cấu hình core I5, RAM 8G. Kết quả thể hiện như trên bảng II.

Từ các kết quả trên chúng ta thấy hệ thống đạt được yêu cầu đặt ra với độ chính xác trên 90 phần trăm. Đặc biệt với việc sử dụng Tensorflow và Tensorflow lite hệ thống đạt độ chính xác lên đến 99 phần trăm với thời gian thực hiện là 14 giây. Đây là thời gian chấp nhận được cho hệ thống điều khiển trong nhà thông minh.

V. KẾT LUẬN

Bài báo tập trung vào nghiên cứu việc sử dụng các mạng nơ-ron trong việc nhận diện hành động của con người. Trong bài báo này chúng tôi đã nhận diện được các hành động với độ chính xác trên 90 phần trăm. Tuy nhiên hệ thống vẫn còn nhược điểm như kết quả nhận diện các hành động chưa cao và tốc độ khung hình trên giây còn thấp. Do đó hướng tiếp theo chúng tôi sẽ thực hiện các bước như tăng tốc độ khung hình trên giây, cải thiện độ chinh xác bằng cách tăng độ phân giải của ảnh đầu vào hoặc sử dụng phương pháp tiền xử lý đã thực hiện trong bài báo trước [22], [23], cũng như kết hợp mạng nơ-ron với các mạng khác để tăng hiệu quả tính toán và thực hiện với đối tượng bất kỳ.

LỜI CẢM ƠN

Nghiên cứu này được thực hiện trong khuôn khổ đề tài do Bộ Giáo dục và Đào tạo, Việt Nam tài trợ với tiêu đề "Nghiên cứu phát triển hệ thống nhận dạng cử chỉ, hành động ứng dụng trí tuệ nhân tạo trong nhà thông minh" theo đề tài cấp bộ mã số B2020-BKA-06. Cảm ơn Bộ KHCN đã tài trợ trong quá trình thực hiện bài báo này.

TÀI LIỆU THAM KHẢO

[1] P. N. Huu and H. N. T. Thu, “Proposal gesture recognition algorithm combining cnn for health monitoring,” in 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), 2019, pp. 209–213.

[2] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction,” pp. 1–19, 2019.

[3] P. Wang, W. Li, C. Li, and Y. Hou, “Action recognition based on joint trajectory maps with convolutional neural networks,” Knowledge-Based Systems, vol. 158, pp. 43 – 53, 2018.

[4] S. A. Khowaja and S.-L. Lee, “Semantic image networks for human action recognition,” International Journal of Computer Vision, vol. 128, no. 2, p. 393–419, Oct 2019.

[5] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” 06 2015, pp. 4694–4702.

[6] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, “Temporal segment networks: Towards good practices for deep action recognition,” 2016.

[7] J. Chen, Q. Ou, Z. Chi, and H. Fu, “Smile detection in the wild with deep convolutional neural networks,” Machine Vision and Applications, vol. 28, p. 173–183, 11 2016.

[8] P. Barros, G. I. Parisi, C. Weber, and S. Wermter, “Emotion-modulated attention improves expression recognition: A deep learning model,” Neurocomputing, vol. 253, pp. 104 – 114, 2017.

[9] C.-C. Hsieh and D.-H. Liou, “Novel haar features for real-time hand gesture recognition using svm,” Journal of Real-Time Image Processing, vol. 10, pp. 357–370, 2015.

[10] Brijesh, First time Tensorflow Lite and Android!, 2017 (accessed December 5, 2017). [Online]. Available: https://gist.github.com/rhezaharliman/

[11] Ehezaharliman, TensorFlow Lite, 2018 (accessed Dec. 24, 2018.). [Online]. Available: https://androidkt.com/tenserflow-lite/

[12] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” CoRR, vol. abs/1212.0402, 2012.

[13] S. Ma, S. A. Bargal, J. Zhang, L. Sigal, and S. Sclaroff, “Do less and achieve more: Training cnns for action recognition utilizing action images from the web,” Pattern Recognition, vol. 68, pp. 334 – 345, 2017.

[14] V. Sodha, TensorFlow Object Detection API tutorial-Training and Evaluating Custom Object Detector, 2018 (accessed March 26, 2018.). [Online]. Available: https://becominghuman.ai

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” 2018.

[16] M. Hollemans, Google’s MobileNets on the iPhone, 2017 (accessed 14 June 2017.). [Online]. Available: https://becominghuman.ai/

[17] J. Hui, SSD object detection: Single Shot MultiBox Detector for real-time processing, 2018 (accessed March 14, 2018.). [Online]. Available: https://becominghuman.ai/

[18] K. Duarte, Y. S. Rawat, and M. Shah, “Videocapsulenet: A simplified network for action detection,” 2018.

[19] M. Hollemans, MobileNet version 2, 2018 (accessed 22 April 2018.). [Online]. Available: https://machinethink.net/blog/mobilenet-v2/

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[22] N. H. Phat, T. Q. Vinh, and T. Miyoshi, “Video compression schemes using edge feature on wireless video sensor networks,” Journal of Electrical and Computer Engineering, vol. 2012, 10 2012.

[23] [23] P. N. Huu, V. Tran-Quang, and T. Miyoshi, “Image compression algorithm considering energy balance on wireless sensor networks,” in 8th IEEE Int’l Conf. Industrial Informatics (INDIN 2010), July 2010, pp. 1005–1010.

PROPOSING GESTURE ALGORITHM USING ARTIFICIAL INTELIGENCE FOR SMART HOME APPLICATIONS

Abstract: The paper studies a system for recognizing gestures and actions in smart homes. The proposed method is based on the use of mobilenetV2 to extract the feature associated with the SSD network (Single Shot Detector). We used five types of gestures of standing up, sitting down, leaning back, wearing shoes, and waving hands. In this application, the feed from the camera of the mobile device is used to detect the object. Objects on the frame are detected by bounding boxes. Results achieved with an accuracy of over 90 percent. However, the degree of sting will depend greatly on the number of training images and their resolution in some cases.

Keywords: MobilenetV2, SSD (Single Shot Detector), identify objects, gestures, actions, postures

Nguyen Huu Phat, nhận bằng kỹ sư 2003), thạc sỹ (2005) ngành Điện tử và Viễn thông tại Đại học Bách Khoa Hà Nội (HUST), Việt Nam và bằng tiến sĩ (2012) về Khoa học Máy tính tại Viện Công nghệ Shibaura, Nhật Bản. Hiện tại, đang là giảng viên tại Viện Điện tử Viễn thông, HUST, Việt Nam. Các nghiên cứu gồm xử lý hình ảnh và video, mạng không dây,

big data, hệ thống giao thông thông minh (ITS), và internet của vạn vật (IoT). Ông đã nhận được giải thưởng bài báo hội nghị tốt nhất trong SoftCOM (2011), giải thưởng tài trợ sinh viên tốt nhất trong APNOMS (2011), giải thưởng danh dự của Viện Công nghệ Shibaura (SIT).

Nguyen Thi Thu Huong, Hiện tại là sinh viên Viện Điện tử Viễn thông, Trường Đại Học Bách Khoa Hà Nội. Hướng nghiên cứu gồm xử lý hình ảnh và video kỹ thuật số và các ứng dụng nhà thông minh.