

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO
NHẬN DIỆN HÀNH ĐỘNG

Môn học: Nhập môn thị giác máy tính

Mã lớp học: CS231.M21.KHTN

Giảng viên hướng dẫn: Mai Tiến Dũng

THÀNH PHỐ HỒ CHÍ MINH – 2022

DANH SÁCH SINH VIÊN

STT	Họ và tên	Mã số sinh viên	Email
1	Nguyễn Tư Thành Nhân	20520079	20520079@gm.uit.edu.vn
2	Lê Nhật Minh	20520070	20520070@gm.uit.edu.vn
3	Lê Nhật Huy	20520056	20520056@gm.uit.edu.vn

LỜI MỞ ĐẦU

Nếu như chúng ta là người quản lý của một hoặc một số bãi đỗ xe. Ta đang muốn giảm gian lận và các mối đe dọa đến bãi đỗ xe của mình, như là: hành vi bạo lực, sự hiện diện có thể có của vũ khí và các vụ trộm cố gắng. Nhận diện hành động có thể được coi là một giải pháp có thể áp dụng vào thực tiễn được.

Thật vậy, bạn có thể trang bị cho camera giám sát của mình một hệ thống nhận diện hành động được đào tạo để phát hiện một trong những tình huống bất thường này. Các luồng video sẽ là đầu vào của bạn. Khi camera thông minh phát hiện tình huống bất thường, phần mềm sẽ đưa ra kết quả đầu ra. Đầu ra sẽ chỉ rõ, với một mức độ tin cậy nhất định, liệu có gian lận hoặc đe dọa hay không.

Nhận diện hành động (Action Recognition) trong Video là bài toán thuộc lớp Video Understanding, trong đó sử dụng khả năng của máy tính để thu thập, xử lý và phân tích dữ liệu đến từ các nguồn trực quan, tức là video. Nói cách khác, nó cho phép máy tính “xem” hàng nghìn luồng Video và “hiểu” thông tin mà nó nhận được từng khung hình. Từ đó, chúng ta có thể biết và cho biết hành động trong Video là gì?

Chúng ta sẽ tìm hiểu về công nghệ của nó. Nhận diện hành động, giống như các bài toán thị giác máy tính khác, dựa vào các thuật toán học sâu để đưa ra kết quả cần tìm. Ý tưởng của học máy là ánh xạ một số loại đầu vào thành đầu ra. Cụ thể hơn, chúng ta sẽ đặt một câu hỏi, khi đó đầu vào và thuật toán cung cấp cho chúng tôi câu trả lời hay đầu ra cần thiết. Mạng neural nhân tạo (Artificial Neural Networks) sẽ cung cấp câu trả lời đó.

Trong bài viết này, chúng tôi sẽ đề cập đến 3 phương pháp giải quyết bài toán sử dụng các kiến trúc mạng học sâu (Deep Learning) để giải quyết.

MỤC LỤC

DANH SÁCH SINH VIÊN	I
LỜI MỞ ĐẦU	II
PHẦN 1 - GIỚI THIỆU BÀI TOÁN	1
1.1. Nhận diện hành động là gì?	1
1.2. Các tập dữ liệu cho bài toán nhận diện hành động.....	1
1.3. Tập dữ liệu của chúng tôi	2
PHẦN 2 - CÁC NGHIÊN CỨU LIÊN QUAN.....	3
2.1. Spatiotemporal filtering.....	3
2.2. Optical flow for video recognition	3
2.3. Phương pháp của chúng tôi	3
PHẦN 3 - PHƯƠNG PHÁP ĐỀ XUẤT	4
3.1. Vấn đề trích xuất đặc trưng từ video	4
3.2.1. <i>ResNet (ResNet – 50)</i>	4
3.2.2. <i>DenseNet (DenseNet – 201)</i>	5
3.2.3. <i>SlowFast</i>	6
3.2. Mô hình phân loại.....	7
3.2.1. <i>SVM kết hợp với biểu quyết đa số</i>	7
3.2.2. <i>Các mô hình phân loại chuỗi (Fully Connected, LSTM, GRU, Conv1D)</i>	8
PHẦN 4 - KẾT QUẢ THỰC NGHIỆM.....	9
4.1. Hướng tiếp cận 1	9
4.2. Hướng tiếp cận 2	10
4.3. Hướng tiếp cận 3	11
4.4. Các thách thức gặp phải và hướng giải quyết:	11
KẾT LUẬN	13
TÀI LIỆU THAM KHẢO	14

PHẦN 1

GIỚI THIỆU BÀI TOÁN

1.1. Nhận diện hành động là gì?

Nhận diện hành động là quá trình sử dụng những cảnh quay trong video để nhận diện, phân loại các hành động khác nhau được thực hiện bởi đối tượng trong video đó. Trong đó, chúng ta phải nhận diện được hành động các đối tượng là loại gì, nhận diện được mục đích của hành động? Và chúng ta cần xử lý được đa dạng các loại đối tượng, bao gồm tương tác giữa người-người, nhiều người, một người và người-vật. Do đó, bài toán Tự động hóa nhận diện hành động cũng đối mặt với rất nhiều thách thức:

- Một người có thể làm nhiều việc cùng một lúc
- Các hành động xen kẽ nhau: đang nấu ăn phải nghe điện thoại
- Xác định những hành động giống nhau theo các cách khác nhau: mở cánh cửa và lau cánh cửa
- Có nhiều người làm nhiều việc khác nhau trong hình

Tuy nhiên, cũng không thể phủ nhận được tiềm năng rất lớn của ứng dụng nhận diện hành động, có thể kể đến như là trợ giúp người lớn tuổi, tương tác người-máy và xây dựng các hệ thống giám sát an ninh.

1.2. Các tập dữ liệu cho bài toán nhận diện hành động

Chúng ta sẽ cùng nhau đi tìm hiểu ba bộ dữ liệu thường được sử dụng trong bài toán nhận diện hành động là UCF101 [20], HMDB51 [21], Kinetics400 [22]

Tập dữ liệu UCF101 là một phần mở rộng của UCF50 và bao gồm 13.320 video clip, được phân lớp thành 101 lớp. 101 lớp này có thể được phân thành 5 lớp (Chuyển động của cơ thể, Tương tác giữa con người với con người, Tương tác giữa con người và vật thể, Chơi nhạc cụ và Thể thao). Tổng thời lượng của các video clip này là hơn 27 giờ. Tất cả các video được thu thập từ YouTube và có tốc độ khung hình cố định là 25 FPS với độ phân giải 320×240 .







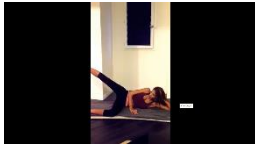



Tập dữ liệu HMDB51 là một bộ sưu tập lớn các video thực tế từ nhiều nguồn khác nhau, bao gồm phim và video trên web. Tập dữ liệu bao gồm 6.849 video clip từ 51 danh mục hành động (chẳng hạn như “nhảy”, “hôn” và “cười”), với mỗi danh mục chứa ít nhất 101 clip. Sơ đồ

đánh giá ban đầu sử dụng ba phần đào tạo/ thử nghiệm khác nhau. Trong mỗi phần, mỗi lớp hành động có 70 clip để huấn luyện và 30 clip để kiểm tra. Độ chính xác trung bình qua ba lần phân chia này được sử dụng để đo lường hiệu suất cuối cùng.

Tập dữ liệu Kinetics400 chứa 400 lớp hành động của con người, với ít nhất 400 video clip cho mỗi hành động. Mỗi clip kéo dài khoảng 10 giây và được lấy từ một video YouTube khác. Các hành động được tập trung vào con người và bao gồm một loạt các lớp bao gồm các tương tác giữa người và vật như chơi nhạc cụ, cũng như tương tác giữa con người với con người như bắt tay.

1.3. Tập dữ liệu của chúng tôi

Chúng tôi lấy ngẫu nhiên từ tập Kinetics400 10 lớp bao gồm bandaging, bowling, breakdancing, ironing, kissing, riding scooter, side kick, tap dancing, texting và washing hair và mỗi lớp được chọn bao gồm ít nhất 309 video. Và có tổng cộng tất cả là 4836 videos train, 401 videos test, 771 videos validation, được thống kê lại như sau

				
Bandaging	Bowling	Breakdancing	Ironing	Kissing
Train: 356 videos	Train: 793 videos	Train: 671 videos	Train: 300 videos	Train: 265 videos
Validation: 41 videos	Validation: 41 videos	Validation: 44 videos	Validation: 39 videos	Validation: 25 videos
Test: 83 videos	Test: 88 videos	Test: 82 videos	Test: 84 videos	Test: 25 videos
				
Riding scooter	Side kick	Tap dancing	Texting	Washing hair
Train: 414 videos	Train: 698 videos	Train: 699 videos	Train: 445 videos	Train: 195 videos
Validation: 39 videos	Validation: 47 videos	Validation: 40 videos	Validation: 43 videos	Validation: 42 videos
Test: 86 videos	Test: 89 videos	Test: 82 videos	Test: 80 videos	Test: 72 videos

(Ảnh 1. Thống kê số lượng dữ liệu của các lớp)

Nhìn chung tập dữ liệu của chúng tôi mất cân bằng, không đồng đều nhau.

Đầu vào của bài toán là: Một video có kích thước 480x360, 30fps, 10s

Đầu ra là: Lớp hành động của video

PHẦN 2

CÁC NGHIÊN CỨU LIÊN QUAN

2.1. Spatiotemporal filtering

Các hành động có thể được xây dựng dưới dạng các đối tượng không thời gian và được nhận dạng bằng các filters không thời gian, như HOG3D [1], 3DConvNets [2, 3, 4], mở rộng các mô hình hình ảnh 2D để xử lý cả hai chiều không thời gian tương tự nhau [5, 6, 7, 8]. Cũng có các phương pháp tập trung vào các filters và pooling sử dụng các stride trên chiều thời gian [9, 10, 11, 12], cũng như các phân rã thành 2D trên chiều không gian và 1D trên chiều thời gian [13, 14, 15, 16]

2.2. Optical flow for video recognition

Đây là một nhánh nghiên cứu cổ điển tập trung vào các đặc trưng con người tạo ra dựa trên optical flow, bao gồm histogram của flow [17], biểu đồ ranh giới chuyển động(motion boundary histogram) [18], đã cho thấy một độ chính xác khá cao trước sự phổ biến của học sâu. Trong thời kì của các mạng học sâu, các phương pháp 2 luồng [19] đã khai thác luồng quang học bằng cách xem như nó là một phương thức đầu vào, phương pháp này là nền tảng của nhiều kết quả cạnh tranh trong [13, 10, 11].

2.3. Phương pháp của chúng tôi

Trong đề án của chúng tôi, chúng tôi tập trung vào ba phương pháp kinh điển và đơn giản. Hai trong số chúng thuộc lớp các phương pháp Spatiotemporal filtering (phương pháp 2 và phương pháp 3) và phương pháp còn lại là một phương pháp thô sơ dựa trên phân loại hình ảnh. Trong đó, phương pháp 2 phân rã không thời gian thành hai trục không gian và thời gian, và xử lý chúng riêng biệt, và phương pháp 3 sử dụng hai luồng với đặc trưng từ hai chuỗi hình ảnh với hai tốc độ tốc độ khác nhau.

PHẦN 3

PHƯƠNG PHÁP ĐỀ XUẤT

3.1. Vấn đề trích xuất đặc trưng từ video

Từ những video trong bộ dữ liệu, chúng tôi lấy ra một số lượng khung hình nhất định mỗi video và thay đổi các khung hình về kích thước $224 \times 224 \times 3$.

Để trích xuất đặc trưng từ mỗi khung hình, chúng tôi đề xuất sử dụng hai mô hình là ResNet và DenseNet. Đây là hai mô hình nổi tiếng hiệu quả trong việc phát hiện và trích xuất đặc trưng của đối tượng trong ảnh, được ứng dụng trong nhiều cuộc thi, nghiên cứu trong thời gian gần đây và cho hiệu suất tốt.

Mỗi video trong bộ dữ liệu kéo dài 10 giây nên số lượng khung hình thích hợp là 64 khung hình. Tuy nhiên do giới hạn về thời gian nên chúng tôi chỉ tiến hành trích xuất đặc trưng của 16 hình mỗi video, cùng với phải sử dụng lại pre-trained model của ResNet-50 và DenseNet-201 trên tập ImageNet.

3.2.1. *ResNet (ResNet – 50)*

ResNet (Residual Network) được đề xuất lần đầu năm 2015 bởi nhóm tác giả Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun [23]. ResNet khiến cho việc huấn luyện hàng trăm thậm chí hàng nghìn lớp của mạng nơ ron trở nên khả thi và hiệu quả.

Nhờ khả năng biểu diễn mạnh mẽ của ResNet, không chỉ các ứng dụng phân loại hình ảnh mà hiệu suất của nhiều ứng dụng thị giác máy đều được tăng cường. Một số ví dụ có thể kể đến là các ứng dụng phát hiện đồ vật và nhận dạng khuôn mặt.

Mô hình ResNet-50 chúng tôi sử dụng:

- Input: 1 khung hình kích thước $224 \times 224 \times 3$
- Output: 1 vector đặc trưng kích thước 2048


```
[ ] resnet_model = Sequential()

pretrained_model= tf.keras.applications.ResNet50(include_top=False,
        input_shape=(IMAGE_HEIGHT, IMAGE_WIDTH, 3),
        pooling='avg',
        weights='imagenet')
for layer in pretrained_model.layers:
    layer.trainable=False

resnet_model.add(pretrained_model)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/94773248/94765736 [=====] - 1s 0us/step
94781440/94765736 [=====] - 1s 0us/step
```

```
[ ] resnet_model.summary()

Model: "sequential_1"

Layer (type)                Output Shape                Param #
=====
resnet50 (Functional)       (None, 2048)                23587712
=====
Total params: 23,587,712
Trainable params: 0
Non-trainable params: 23,587,712
=====
```

(Ảnh 2. Ảnh chụp màn hình trong quá trình thực nghiệm với mô hình ResNet50)

3.2.2. DenseNet (DenseNet – 201)

DenseNet (Dense connected convolutional network) được công bố trong CVPR2017 bởi nhóm tác giả Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger [24], là một trong những network mới nhất cho nhận diện đối tượng.

Densenet có cấu trúc gồm các dense block và các transition layers. Các lớp mạng trong một denseblock được kết nối dày đặc với nhau. Với CNN truyền thống nếu chúng ta có L layer thì sẽ có L connection, còn trong densenet sẽ có $L(L+1)/2$ connection.

Mô hình DenseNet-201 chúng tôi sử dụng:

- Input: 1 khung hình kích thước $224 \times 224 \times 3$
- Output: 1 vector đặc trưng kích thước 1920

```
[ ] densenet_model = Sequential()

pretrained_model= tf.keras.applications.DenseNet201(include_top=False,
            input_shape=(IMAGE_HEIGHT, IMAGE_WIDTH, 3),
            pooling='avg',
            weights='imagenet')
for layer in pretrained_model.layers:
    layer.trainable=False

densenet_model.add(pretrained_model)
```

Downloading data from <https://storage.googleapis.com/tensorflow/keras-applications/74842112/74836368> [=====] - 2s 0us/step
74850304/74836368 [=====] - 2s 0us/step

```
[ ] densenet_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
densenet201 (Functional)	(None, 1920)	18321984

=====

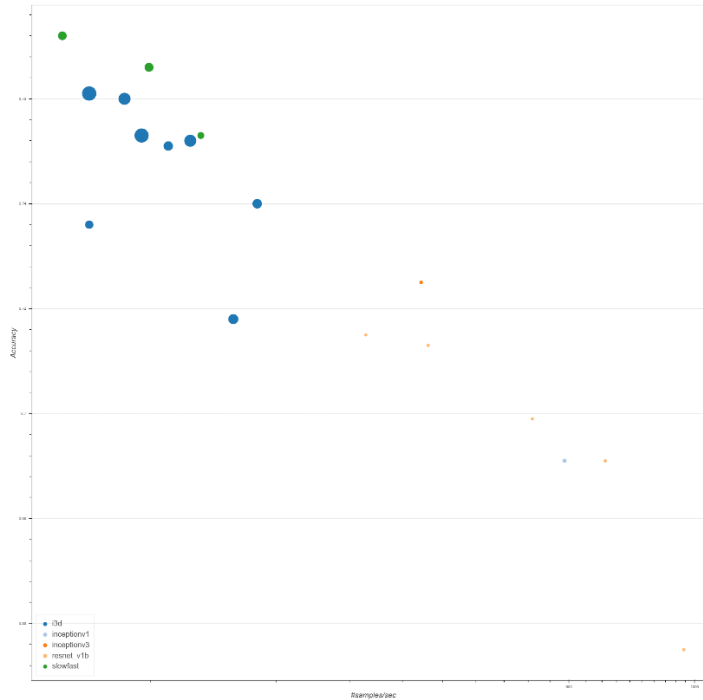
Total params: 18,321,984
Trainable params: 0
Non-trainable params: 18,321,984

(Ảnh 3. Ảnh chụp màn hình trong quá trình thực nghiệm với mô hình DenseNet201)

3.2.3. SlowFast

Với những phương pháp trên, đều sử dụng bộ trích xuất đặc trưng cho ảnh trên bộ dữ liệu Imagenet. Do đó, nó chưa thực sự tận dụng được đặc trưng cho một video trên bộ dữ liệu riêng nhận diện hành động. Vì vậy, nhóm chúng tôi đã tìm hiểu và lựa chọn SlowFast là phương pháp state of the art và có kết quả khá tốt so với các phương pháp được ra mắt cùng thời điểm

Chúng tôi giới thiệu mạng SlowFast [25] để nhận dạng video. Mô hình của chúng tôi bao gồm Slow pathway, hoạt động ở tốc độ khung hình thấp, để nắm bắt ngữ nghĩa không gian và Fast pathway, hoạt động ở tốc độ khung hình cao, để ghi lại chuyển động ở độ phân giải tạm thời tốt. Fast pathway có thể được tạo ra rất nhẹ bằng cách giảm dung lượng kênh của nó, nhưng vẫn có thể tìm hiểu thông tin hữu ích về thời gian để nhận dạng video.



(Ảnh 4. So sánh phương pháp SlowFast và một số phương pháp khác)

Mô hình SlowFast chúng tôi sử dụng:

- Input: Video có kích thước $224 \times 224 \times 3 \times 64$
- Output: Vector trích xuất đặc trưng 2304

3.2. Mô hình phân loại

Đối với kết quả trích xuất đặc trưng của SlowFast, chúng tôi chỉ thực hiện phân loại bằng Logistic Regression. Còn đối với kết quả trích xuất đặc trưng của ResNet và DenseNet, chúng tôi sẽ xử lý vector trích xuất đặc trưng như sau.

3.2.1. SVM kết hợp với biểu quyết đa số

Dựa vào quan sát trên tập dữ liệu, nhận thấy có thể dễ dàng phân loại một số video kiểu tương tác người vật thuộc lớp nào chỉ dựa trên vật. Từ ý tưởng này ta có thể đưa bài toán nhận diện hành động về bài toán phân loại hình ảnh quen thuộc. Vì còn nhiều hành động khác phức tạp hơn, nên mô hình phân loại nên sử dụng một loại phi tuyến tính. Ở đây chúng tôi sử dụng SVM kết hợp với biểu quyết đa số.

Quá trình thực hiện:

- Gán mỗi ảnh trong tập các video với nhãn là nhãn của video chứa ảnh
- Đưa các ảnh qua mô hình chiết xuất đặc trưng ảnh cho mô hình phân loại học

- Sử dụng mô hình phân loại để phân loại từng ảnh (hoặc một lượng ảnh nhất định) trong video, nhãn nào được mô hình chọn nhiều nhất sẽ là nhãn của video

3.2.2. Các mô hình phân loại chuỗi (*Fully Connected, LSTM, GRU, Conv1D*)

Việc sử dụng mô hình các mô hình phân loại hình ảnh trực tiếp thì không mô hình hóa được yếu tố thời gian, vì vậy chúng tôi có ý tưởng đề xuất sử dụng một mô hình thường được sử dụng trong quá trình xử lý chuỗi như: Fully Connected, LSTM, GRU, Conv1D để mô hình hóa yếu tố này.

Quá trình thực hiện:

- Tạo đặc trưng cho mỗi video từ đặc trưng của các khung hình đã trích xuất trong video đó.
- Thử thay thế quá trình đếm bằng một số mạng với các cấu trúc và tham số khác nhau
- Chọn ra các cấu trúc cho độ chính xác cao nhất trên tập test
- Chạy thử và lấy độ chính xác trên tập validation

PHẦN 4

KẾT QUẢ THỰC NGHIỆM

Kết quả tổng quan được thống kê lại như sau

Mô hình trích xuất	Mô hình phân loại	Augmented Data	Accuracy
DenseNet	SVM + Biểu quyết đa số	Không	78%
ResNet	Fully Connected	Không	29%
DenseNet	Fully Connected	Không	80%
ResNet	LSTM	Không	38%
DenseNet	LSTM	Không	82%
ResNet	GRU	Không	37%
DenseNet	GRU	Không	81%
ResNet	Conv1D	Không	30%
DenseNet	Conv1D	Không	80%
SlowFast	Logistics Regression	Có	92%

4.1. Hướng tiếp cận 1

Hướng tiếp cận sử dụng mô hình DenseNet nhằm trích xuất đặc trưng, dùng mô hình SVM để phân loại hình ảnh và xuất kết quả theo quá trình biểu quyết đa số đưa ra kết quả như sau:

Label	Accuracy
Bandaging	79%
Bowling	88%
Breakdancing	69%
Ironing	85%
Kissing	58%
Riding scooter	86%
Side kick	78%
Tap dancing	82%

Texting	83%
Washing hair	72%

Dựa vào kết quả, chúng tôi thấy được mô hình chỉ phân loại tốt ở các hành động dạng người-vật, tuy nhiên, các hành động tương tác người vật có một số khung hình thao tác có vẻ giống nhau, như cặp bandaging với ironing, có thể gây nhầm lẫn, nên accuracy của bandaging thấp hơn so với các hành động tương tác người-vật khác.

4.2. Hướng tiếp cận 2

Hướng tiếp cận sử dụng mô hình DenseNet nhằm trích xuất đặc trưng, và dùng các mô hình (Fully Connected, LSTM, GRU, Conv1D) để mô hình hóa chuỗi thời gian

Model Label	FC			LSTM + FC			GRU + FC			Conv1D + FC		
	Pre	Re	F1	Pre	Re	F1	Pre	Re	F1	Pre	Re	F1
Bandaging	85%	87%	86%	86%	84%	85%	85%	88%	86%	85%	81%	83%
Bowling	95%	88%	92%	95%	93%	94%	93%	93%	93%	97%	88%	93%
Breakdancing	68%	73%	71%	64%	74%	69%	60%	76%	67%	66%	77%	71%
Ironing	92%	81%	86%	91%	87%	89%	82%	89%	86%	88%	86%	87%
Kissing	51%	76%	61%	60%	84%	70%	76%	76%	76%	54%	80%	65%
Riding scooter	78%	88%	83%	80%	86%	83%	81%	87%	84%	82%	87%	84%
Side kick	77%	67%	72%	79%	70%	75%	80%	64%	71%	66%	66%	66%
Tap dancing	78%	68%	73%	79%	70%	74%	78%	71%	74%	77%	67%	72%
Texting	86%	81%	83%	87%	85%	86%	92%	81%	86%	93%	82%	88%
Washing hair	78%	88%	82%	85%	83%	84%	88%	85%	87%	81%	83%	82%

Có thể thấy được sự cải thiện rõ ràng về accuracy của hành động tương tác người-vật bandaging, texting so với hướng tiếp cận 1, và các một số hành động thuộc kiểu khác, như chuyển động cơ thể. Tuy nhiên, có thể do mô hình chỉ sử dụng tham số được huấn luyện sẵn trên tập ImageNet của DenseNet, nên accuracy của các hành động không thuộc kiểu tương tác người vật còn thấp.

Chúng tôi đã thí nghiệm tương tự với bộ chiết xuất đặc trưng ResNet, tuy nhiên, kết quả không được như kì vọng, độ chính xác là 39%. Chúng tôi đã đưa ra 2 giả thuyết sau :

Đặc trưng của mô hình phân loại tại lớp cuối cùng của mô hình ResNet không có khả năng ứng dụng cho bài toán này. Để ResNet có thể hoạt động được, chúng tôi cần phải tối ưu hóa lại các trọng số trên ResNet bằng cách huấn luyện lại nó trên tập Kinetics dựa trên trọng số của ImageNet. Hoặc là cần phải làm giàu cho bộ dữ liệu (augment data) bởi vì ResNet không hoạt động tốt đối với các thay đổi về kích thước.

Mô hình phân loại của chúng tôi không có khả năng sử dụng bộ trích xuất đặc trưng của ResNet.

4.3. Hướng tiếp cận 3

Hướng tiếp cận 3 sử dụng mô hình SlowFast để trích xuất đặc trưng.

Label	Precision	Recall	F1 Score
Bandaging	97%	92%	94%
Bowling	95%	95%	95%
Breakdancing	83%	91%	87%
Ironing	93%	92%	92%
Kissing	72%	84%	78%
Riding scooter	94%	98%	96%
Side kick	92%	85%	88%
Tap dancing	93%	93%	92%
Texting	94%	93%	93%
Washing hair	96%	93%	94%

Các phương pháp trên đều có kết quả không tốt hơn đối với lớp kissing, có thể giải thích rằng lớp này có tính phức tạp hơn, do khi hôn có thể là tương tác giữa người người hoặc là người vật, hay thậm chí có thể bản thân người ấy tự hôn vào tay.

4.4. Các thách thức gặp phải và hướng giải quyết:

Trong quá trình thực hiện đồ án, chúng tôi đã gặp các thách thức về vấn đề kỹ thuật như sau:

- Mô hình chiết xuất đặc trưng ảnh trả về tensor có kích thước quá lớn, không đủ RAM trên môi trường Colab

Cách giải quyết: Sử dụng một lớp GlobalAveragePooling2D để giảm kích thước tensor.

- Không đủ RAM và thời gian để huấn luyện mô hình SVM

Vì độ phức tạp không thời gian của SVM tăng theo hàm bậc ba đối với số lượng các điểm dữ liệu, chúng tôi đã thu nhỏ tập train và thực hiện quá trình ensemble các tập train con khác nhau để đưa ra kết quả.

- Không đủ thời gian để fine-tune lại bộ trích xuất đặc trưng

Chúng tôi đã lựa chọn giữ nguyên bộ trọng số của mô hình vì phải mất đến 30 epoches để mô hình phân loại hội tụ và mất 3 tiếng để duyệt qua toàn bộ điểm dữ liệu sử dụng mô hình trích xuất đặc trưng.

KẾT LUẬN

Trên đây là nội dung nghiên cứu và tìm hiểu của chúng tôi trong việc sử dụng một số mô hình trong việc trích xuất đặc trưng và phân loại cho bài toán nhận diện hành động - Action Recogniton. Tuy nhiên trong báo cáo của chúng tôi vẫn còn nhiều hạn chế. Chúng tôi đưa ra một số hướng cải tiến có thể tiến hành trong tương lai như sau:

- Phát triển mô hình để xử lý chuỗi khung hình không đồng đều trên trục thời gian
- Sử dụng các phương pháp Ensemble như XGB và Light GBM để tăng độ chính xác
- Fine-tuning lại mô hình trích xuất đặc trưng đối với ResNet, DenseNet và SlowFast
- Kết hợp với đặc trưng từ Pose Estimation hay Body Segmentation

Trong source code được cung cấp kèm theo, là 3 phương pháp mà nhóm chúng tôi đã chạy.

Chúng tôi xin cảm ơn thầy Mai Tiến Dũng đã hướng dẫn và đưa ra nhận xét để cải thiện báo cáo của chúng tôi. Chúng tôi cũng cảm ơn mọi người trong dự án Kinetics đã cung cấp dữ liệu cho thử nghiệm của chúng tôi.

TÀI LIỆU THAM KHẢO

- [1] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In Proc. BMVC., 2008.
- [2] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In Proc. ECCV, 2010.
- [3] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In Proc. ICCV, 2015.
- [4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In Proc. CVPR, 2017.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.
- [6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Proc. ICLR, 2015.
- [7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proc. CVPR, 2015.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. CVPR, 2016.
- [9] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. IEEE PAMI, 2018.
- [10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In Proc. CVPR, 2016.
- [11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In Proc. ECCV, 2016.
- [12] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In ECCV, 2018.
- [13] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In NIPS, 2016.
- [14] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In Proc. CVPR, 2018.

- [15] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. arXiv:1712.04851, 2017.
- [16] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In Proc. ICCV, 2017
- [17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In Proc. CVPR, 2008.
- [18] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In Proc. ECCV, 2006.
- [19] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In NIPS, 2014.
- [20] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. In Proc. arXiv:1212.0402v1, 2012
- [21] Limin Wang, Yu Qiao, and Xiaoou Tang. Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. In Proc. arXiv:1505.04868v1, 2015
- [22] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. In Proc. arXiv:1705.06950v1, 2017
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In Proc. arXiv:1512.03385v1, 2015
- [24] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks. In Proc. arXiv:1608.06993, 2016
- [25] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik and Kaiming He. SlowFast Networks for Video Recognition. In Proc. arXiv:1812.03982v3