



HUMAN POSE ESTIMATION & HUMAN ACTION RECONIGTION

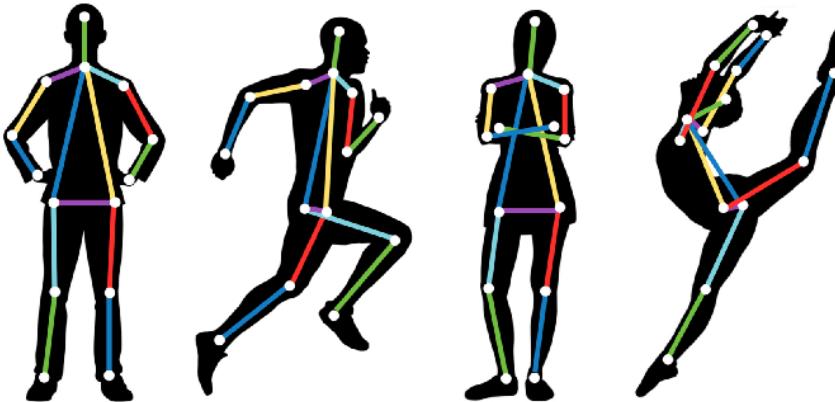
Mục lục

I. MOTIVATION:	3
1. Động lực nghiên cứu:.....	3
2. Ý nghĩa khoa học:	3
3. Ý nghĩa ứng dụng:	4
II. PROBLEM STATEMENT:.....	5
III. RELATED WORK:.....	7
1. Một số phương pháp xác định khung xương:.....	8
a) DeepPose: Human Pose Estimation via Deep Neural Networks ...	11
b) Human Pose Estimation with Iterative Error Feedback	15
c) Efficient Object Localization Using Convolutional Networks	17
d) High-Resolution Net (HRNet).....	19
e) OpenPose	21
f) Stacked Hourglass Networks for Human Pose Estimation	24
g) So sánh các phương pháp:	26
2. Một số phương pháp phân lớp hành vi	29
a) Semantics-Guided Neural Networks.....	32
b) Revisiting Skeleton-based Action Reconigition	34
c) Double-feature Double-motion Network	36
d) Spatial Temporal Graph Convolutional Networks	39
e) Spatio-temporal Tuples Transformer (STTFormer)	41
IV. METHOD	44
1. Sơ lược về mạng tích chập đồ thị (Graph convolution network)....	44
2. Phương pháp	46
V. APPLICATION	60
VI. REFERENCES:	62

I. MOTIVATION:

1. **Động lực nghiên cứu:**

- Nhận dạng hành vi người từ lâu đã là một vấn đề quan trọng của lĩnh vực Thị giác máy tính.



- Cùng với nhận dạng khuôn mặt, nhận dạng hành vi cho phép ta theo dõi những cử động dù là nhỏ nhất của người và từ đó phân tích hoạt động sinh học của người đó theo thời gian thực, mở ra nhiều ứng dụng lớn trong các lĩnh vực khác nhau.
- Hiện nay, có hai tiêu chí lớn trong việc nhận diện hành vi, thứ nhất là về mặt dữ liệu, thứ hai là về mặt số lượng.
 - + Về mặt dữ liệu: nhận dạng tư thế 2D và nhận dạng tư thế 3D.
 - + Về mặt số lượng: nhận dạng đơn đối tượng (Single-person pose estimation) và nhận dạng đa đối tượng (Multi-person pose estimation).

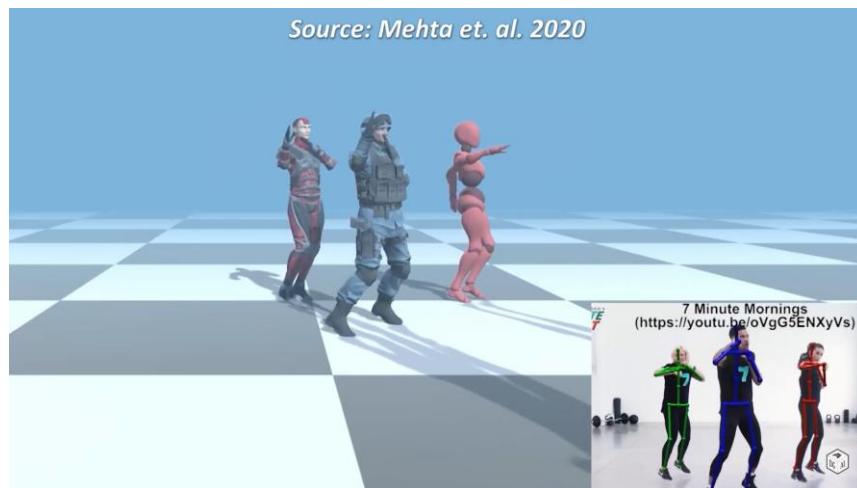
2. Ý nghĩa khoa học:

- Tìm được các điểm neo trên cơ thể người như là cổ tay, vai, đầu gối... từ đó nối chúng lại với nhau để tạo nên dáng người hoàn thiện.

- Việc tìm được điểm neo dựa trên các đặc trưng cục bộ bất biến trên cơ thể người sau đó ước lượng các vị trí bộ phận con người, các đặc trưng sẽ liên quan đến vân, dáng.

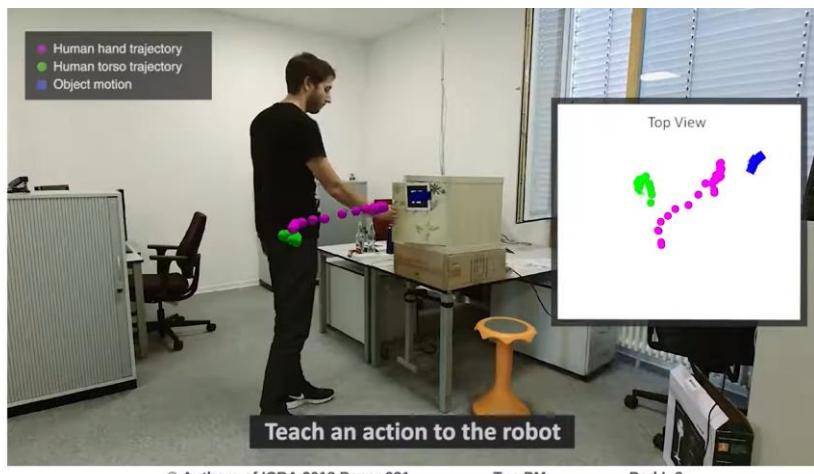
3. Ý nghĩa ứng dụng:

- Việc ước lượng khung xương người trên ảnh và nhận dạng hành động của con người rất có ý nghĩa về mặt ứng dụng trong cộng đồng có thể kể qua như sau:
 - Hiện nay việc đầu tư nguồn nhân lực rất nhiều vào việc giám sát con người ví dụ như: camera giám sát cần phải có người theo dõi, nhân viên y tế giám sát người bệnh.
 - Vì thế việc phát triển ra nhận dạng hành động con người có thể xem một giải pháp tuyệt vời có thể hỗ trợ hoặc thậm chí là thay thế con người trong việc giám sát các hành vi bất thường như đánh nhau, trộm cắp, việc té ngã của bệnh nhân, theo dõi hành động của bé trong nhà.
- Ngoài ra nếu có thể mở rộng theo hướng ứng dụng như:
 - + Tạo mô hình 3D hành động trong game mà không cần qua các thao tác edit phức tạp.



- + Các game thực tế ảo mà không cần thiết bị mà chỉ thông qua cử chỉ con người

- + Từ hành động 3D của con người giúp cho robot có thể tái tạo và thực hiện các hành động một cách mượt mà.

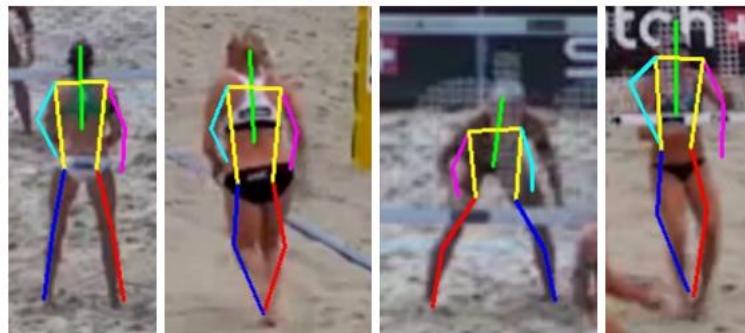


II. PROBLEM STATEMENT:

Bài toán Human Action Recognition có rất nhiều thách thức ví dụ như cùng một hành động do nhiều người cùng thực hiện, các đặc trưng rút trích cho các người khác nhau sẽ khác nhau hay là khác hoạt động nhưng lại cùng một người thì khi rút đặc trưng sẽ đưa về những đặc trưng tương tự nhau. Vì thế một hướng đi cho bài toán này là không xem xét về toàn bộ cơ thể người nữa mà chỉ xem xét khung xương người, để loại bỏ đi các thông tin không cần thiết.

Cách tiếp cận bài toán xác định các khớp của con người sẽ bao gồm:

- Single-Person Pose Estimation: việc xác định các khớp xương sẽ được đưa về bài toán hồi quy (keypoint regression) hay dựa vào detection-based phát hiện khung xương người dựa vào ảnh heatmap
- Multi-Person Pose Estimation: có thể chia ra làm 2 nhóm là bottom-up và top-down.
- **Bottom-up:** ban đầu sẽ ước lượng các khớp của người sau đó mới tiến hành group chúng lại để tạo nên các khớp hoàn chỉnh cho từng người.
- **Top-down:** đầu tiên sẽ phát hiện người và đóng bounding box sau đó sẽ rút trích khung xương cho từng người được xác định trong ảnh.
- Khó khăn: Human pose estimation là một bài toán khó khi trong thực tế phải ước lượng tư thế của người trong tình trạng
 - + Độ phân giải thấp, ảnh bị mờ:



- + Các bộ phận của người bị che khuất bởi các vật thể hay chính bởi các khớp lớn của người:



- + Sự thay đổi của thời tiết và điều kiện môi trường bên ngoài, đặt biệt các loại quần áo đa dạng khiến cho hình dáng người làm máy tính khó nhận dạng hơn cũng là một thử thách lớn.

Sau khi có được khung xương từ dãy ảnh ta sẽ tới bước tiếp theo là nhận dạng hành động dựa trên dãy khung xương, Human Action Recognition.

Ta có thể phát biểu bài toán thành:

- Bài toán xác định khung xương: từ dãy ảnh I ban đầu tạo ra một dãy khung xương S mang thông tin về dáng đứng của tất cả những người có trong mỗi ảnh. Mỗi khung xương sẽ bao gồm K khớp (đầu, cổ, vai, ...) của một người trong số N người có trong ảnh. Khung xương phải có tính chính xác, thể hiện đúng hành vi mà người trong ảnh đang thực hiện.
- Bài toán phân lớp hành vi của khung xương: liên kết dãy khung xương S rời rạc ban đầu thành một chuỗi hành động theo thời gian thông qua việc xem xét sự biến đổi của các khớp qua từng frame. Từ thông tin có được về sự biến đổi các khớp, tra cứu và so sánh với thông tin về những hành vi có trong cơ sở dữ liệu và dự đoán ra hành động mà khung xương đó đang thực hiện.

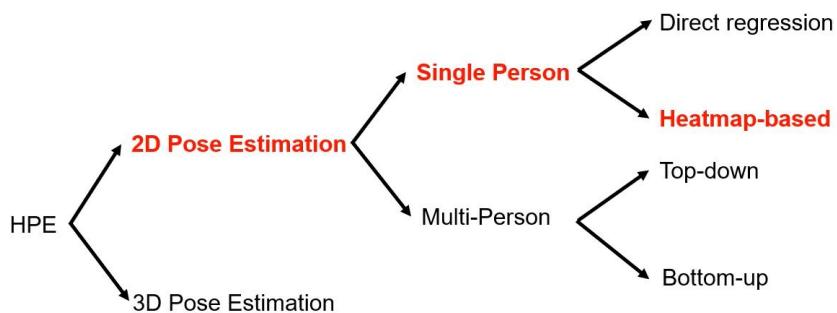
III. RELATED WORK:

Với mỗi phương pháp (khoảng 2-3 phương pháp cho từng phần khung xương và hành vi) cần nêu được những ý sau:

- **Đặc điểm nổi bật của phương pháp này so với những phương pháp khác.**
- **Phát biểu bài toán (input, output).**
- **Kiến trúc mạng.**
- **Hàm lỗi trong quá trình training.**
- **Phương pháp này còn những hạn chế gì.**

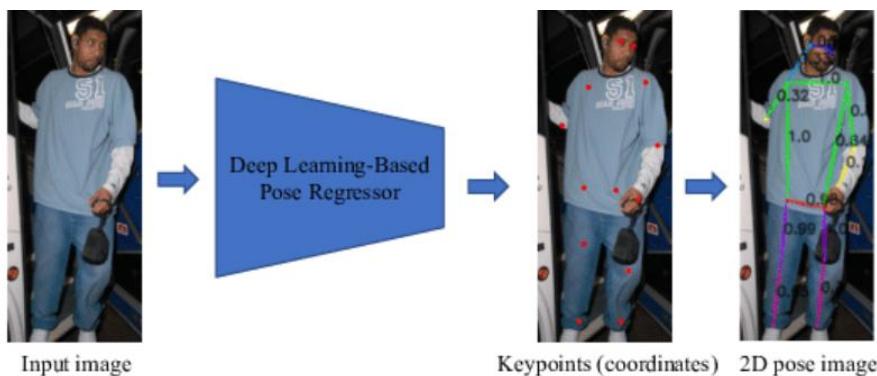
1. Một số phương pháp xác định khung xương:

Đối với phương pháp xác định khung xương được chia ra làm 2 trường phái Single-Person Pose Estimation và Multi-Person Pose Estimation.



Với Single-Person Pose Estimation sẽ đưa về bài toán hồi quy keypoint hoặc heatmap.

Keypoint regression: mạng sẽ học để xác định được chính xác khớp đang ở tọa độ nào, với output xác thực sẽ là tọa độ các điểm khớp được đánh nhãn.

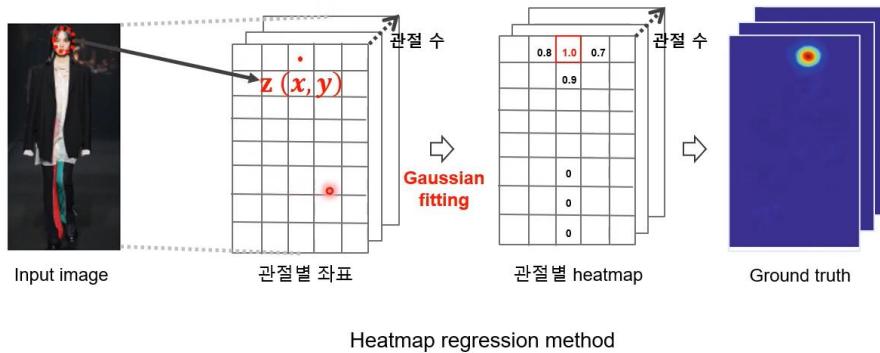


Heatmap regression: Với việc xác định chính xác vị trí của từng điểm khớp ở keypoint regression quả thật là một nhiệm vụ khó khăn, từ đó thay vì xác định một cách chính xác tuyệt đối, người ta đã chuyển qua sử dụng khả năng xuất hiện của khớp trên một vùng từ đó hiệu năng được cải tiến đáng kể. Tùy mỗi phương pháp các nhãn sẽ khác nhau ở phương sai.

Dưới đây là công thức để tạo ra ảnh heatmap.

$$h_i^g(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i^g)^\top \Sigma^{-1}(\mathbf{x} - \mathbf{x}_i^g)\right),$$

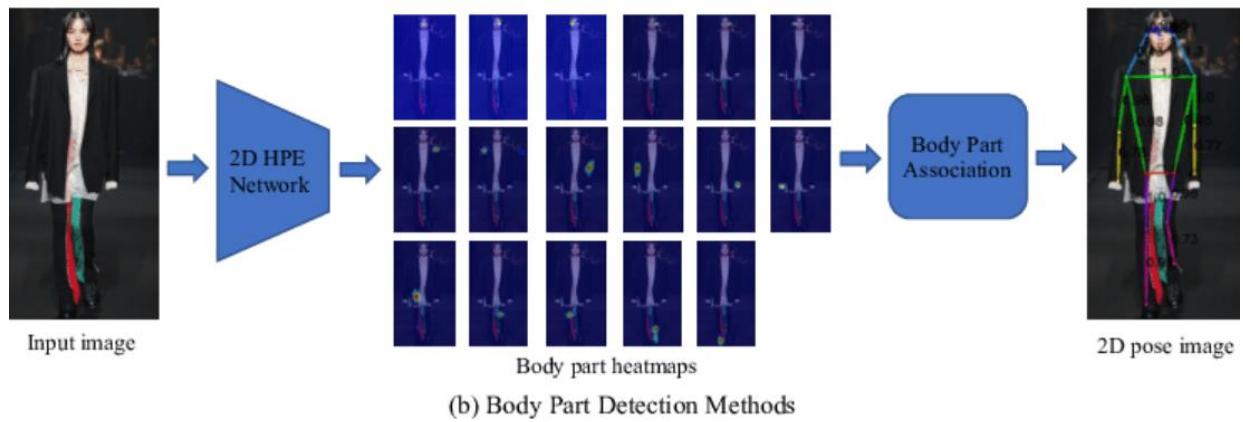
$$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}.$$



Các khớp thứ i sẽ được áp vào một hàm gauss với kỳ vọng tại chính điểm khớp thật với $\mathbf{x} \in \mathbb{R}^2$, sau đó tạo ra các ảnh heatmap cho mỗi khớp.

Có thể tinh ý nhận thấy keypoint regression cũng gần như là heat map regression trường hợp đặt biệt với $\sigma = 0$ được gọi là binary heatmap.

$$h_i^g(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}_i^g, \\ 0 & \text{otherwise.} \end{cases}$$

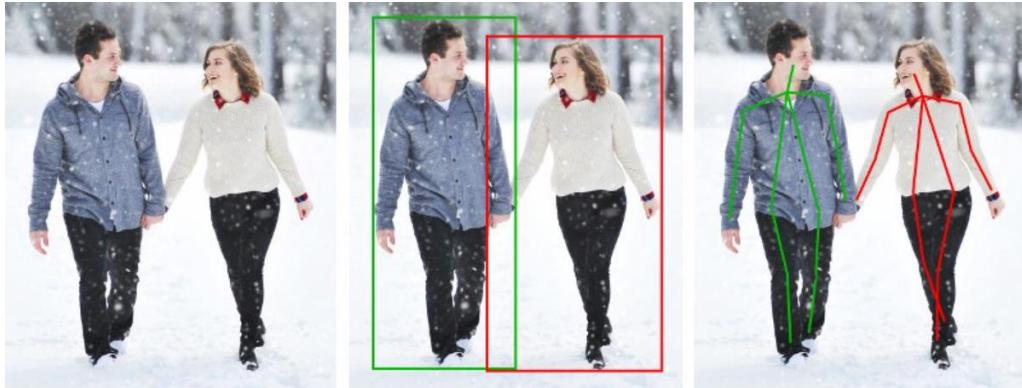


Và hàm lỗi dùng để training trong quá trình học đa số sẽ là Mean Squared Error giữa ảnh heatmap dự đoán và xác thực.

$$f_t = \left\| \text{Ground truth} - \text{Prediction} \right\|_2^2$$

Multi-Person Pose Estimation: sẽ được chia ra làm top-down và bottom-up:

Top-down: Đầu tiên sẽ phát hiện từng người trong ảnh và đóng bounding box, sau đó tiến hành xác định khung xương cho từng người.



Bottom-up: xác định khớp xương trước và sau đó liên kết chúng cho từng người để tạo khung xương.



* Keypoint regression:

a) DeepPose: Human Pose Estimation via Deep Neural Networks

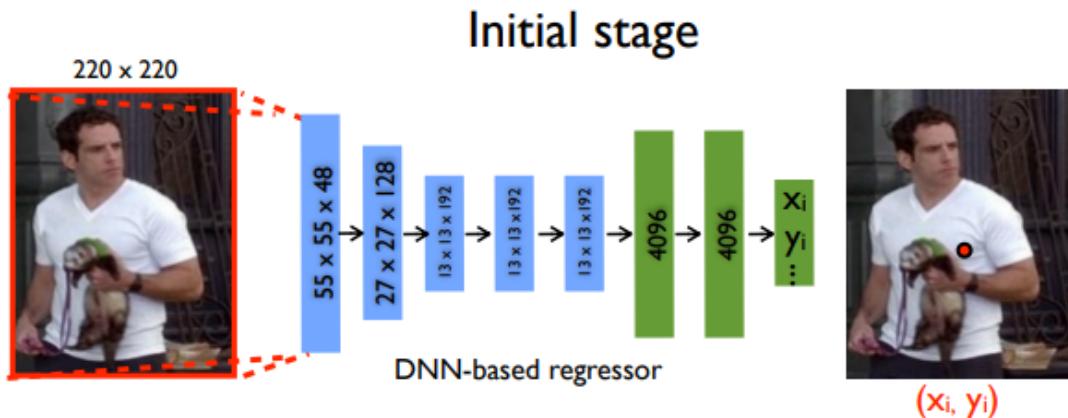
- DeepPose: một mô hình dựa trên kiến trúc mạng Alex net, là phương pháp đầu tiên áp dụng Deep Learning vào Human Pose Estimation
- Input: Ảnh chứa người
- Output: vector $\in \mathbb{R}^{2k}$ chứa tọa độ của từng keypoint trên vị trí của người đã được chuẩn hóa so với tâm được dùng để train.

$$\mathbf{y} = (\dots, \mathbf{y}_i^T, \dots)^T, i \in \{1, \dots, k\}$$

$$\psi(x; \theta) \in \mathbb{R}^{2k}$$

- Nếu muốn đưa về vị trí tuyệt đối trên ảnh thì ta sẽ chuẩn hóa ngược lại.

$$y^* = N^{-1}(\psi(N(x); \theta))$$



- Hàm lỗi được dùng là khoảng cách L2 (Euclidean)

$$\arg \min_{\theta} \sum_{(x,y) \in D_N} \sum_{i=1}^k \|\mathbf{y}_i - \psi_i(x; \theta)\|_2^2$$

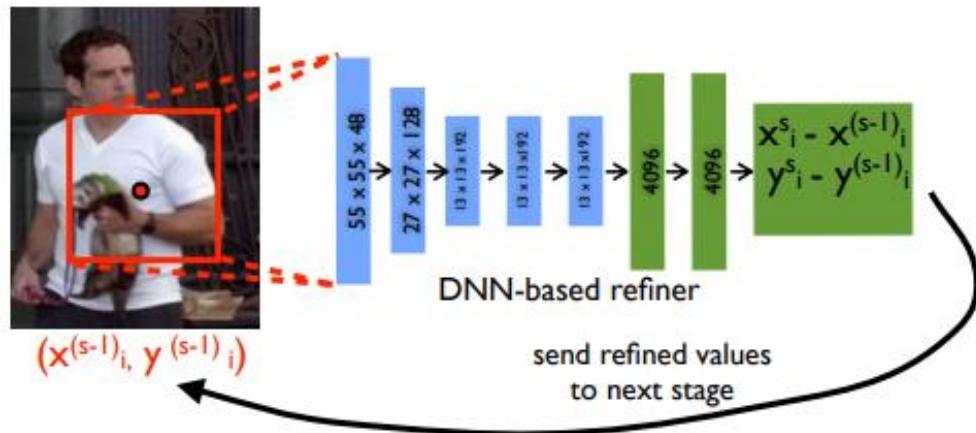
- Điểm đặc biệt ở mạng là hoàn thiện kết quả bằng hồi quy xếp tầng. Tư thế được ước lướt ở bước ban đầu sẽ được cải thiện dần qua S bước.

$$\text{Stage } s: \quad \mathbf{y}_i^s \leftarrow \mathbf{y}_i^{(s-1)} + N^{-1}(\psi_i(N(x; b); \theta_s); b) \quad (6)$$

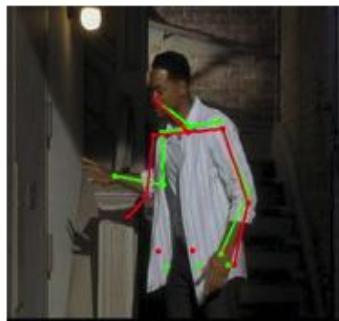
$$\text{for } b = b_i^{(s-1)}$$

$$b_i^s \leftarrow (\mathbf{y}_i^s, \sigma \text{diam}(\mathbf{y}^s), \sigma \text{diam}(\mathbf{y}^s)) \quad (7)$$

Stage s



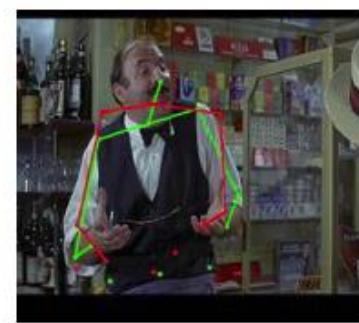
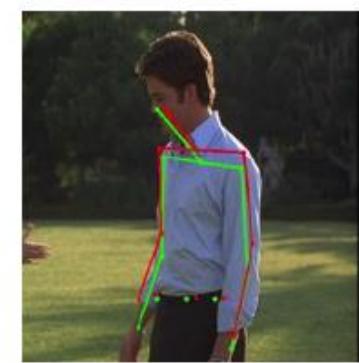
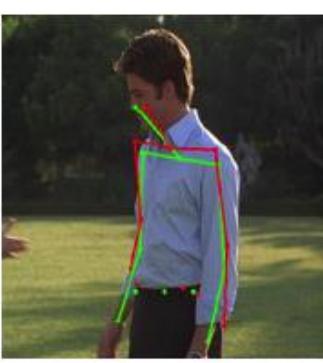
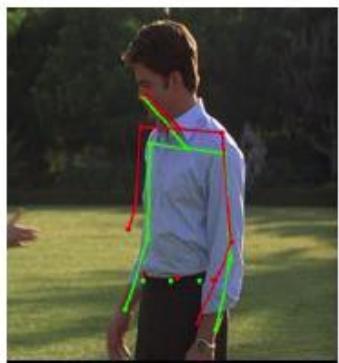
Initial stage 1



stage 2



stage 3



- Việc truy vấn dựa trên tọa độ XY là khó khăn, làm suy yếu sự khái quát hóa do đó có hiệu quả kém ở một số trường hợp nhất định.

Method	Arm		Leg		Ave.
	Upper	Lower	Upper	Lower	
DeepPose-st1	0.5	0.27	0.74	0.65	0.54
DeepPose-st2	0.56	0.36	0.78	0.70	0.60
DeepPose-st3	0.56	0.38	0.77	0.71	0.61
Dantone et al. [2]	0.45	0.25	0.65	0.61	0.49
Tian et al. [24]	0.52	0.33	0.70	0.60	0.56
Johnson et al. [13]	0.54	0.38	0.75	0.66	0.58
Wang et al. [25]	0.565	0.37	0.76	0.68	0.59
Pishchulin [17]	0.49	0.32	0.74	0.70	0.56

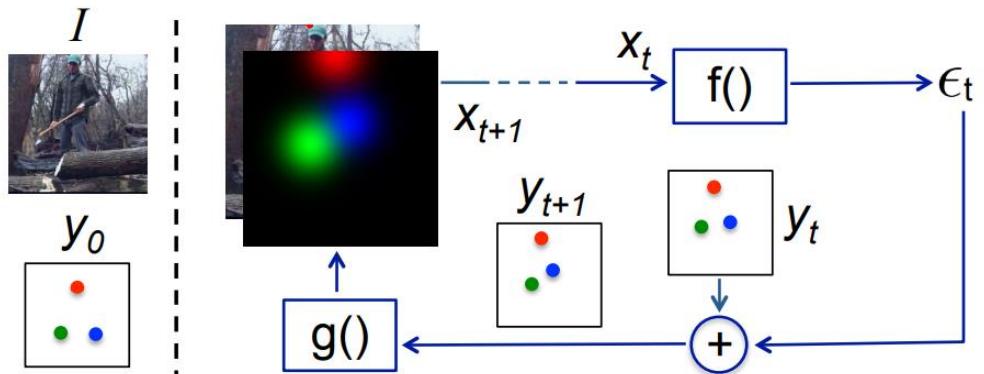
Table 1. Percentage of Correct Parts (PCP) at 0.5 on LSP for Deep-Pose as well as five state-of-art approaches.

- Heat map regression sẽ mang lại hiệu năng tốt hơn sẽ được trình bày bên dưới.

b) Human Pose Estimation with Iterative Error Feedback

- Là một phương pháp kết hợp giữa keypoint regression và có cả heatmap bên trong, tuy nhiên vẫn coi như là keypoint regression.
- Phương pháp có thể được tóm gọn như sau: dự đoán xem ước lượng tư thế người sai ở đâu và sau đó sửa lại, công đoạn sẽ được lặp lại nhiều lần. Thay vì dự đoán output trong một lần, họ sử dụng self-correcting model phát hiện lỗi và tự đưa ra giải pháp để sửa lỗi.
- Input: một ảnh I và output trước đó y_{t-1} (kết quả sẽ được cải thiện sau mỗi lần lặp)
- Output: tọa độ khung xương của người.
- Output xác thực: tọa độ khung xương người đã được đánh nhãn. $X_t = I \oplus g(y_{t-1})$ trong đó I là ảnh và y_{t-1} là output (tọa độ các khớp) trước đó (\oplus : là concat):
 - + $f(X_t) = \epsilon_t$ là một hàm tương trưng cho convolution network kết quả trả ra là các độ lệch của output được dự đoán trước đó, và sẽ được cộng vào y_t để tạo ra y_{t+1} .

- + $g(\cdot)$ là hàm có nhiệm vụ nhận vào tọa độ điểm các khớp và trả về ảnh gaussian heatmap dựa vào tọa độ của các khớp.

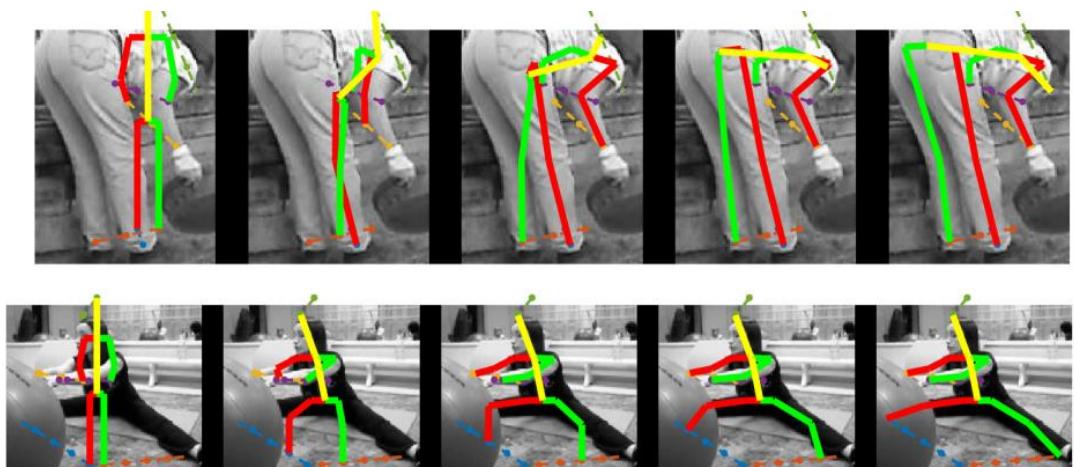


- $\epsilon_t = f(x_t)$
- $y_{t+1} = y_t + \epsilon_t$
- $x_{t+1} = I \oplus g(y_{t+1})$

- Tham số Θ_g và Θ_f sẽ học dựa trên hàm lỗi sau đây:

$$\min_{\Theta_f, \Theta_g} \sum_{t=1}^T h(\epsilon_t, e(y, y_t))$$

$$e(y^k, y_t^k) = \min(L, ||u||) \cdot \hat{u}$$



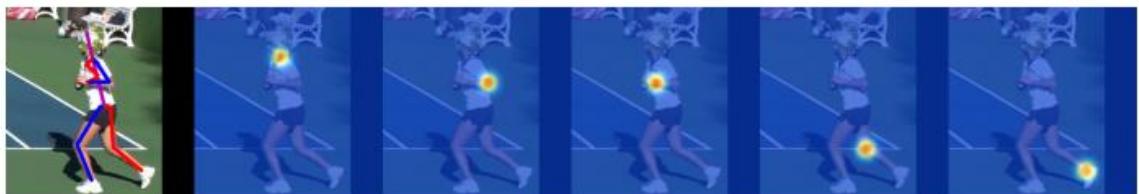
- Bạn có thể thấy khung xương sẽ cải thiện qua từng bước. Đây cũng là một cách mới lạ trong Human Pose Estimation.

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	UBody	FBody
Yang & Ramanan [48]	73.2	56.2	41.3	32.1	36.2	33.2	34.5	43.2	44.5
Pishchulin et al [29]	74.2	49.0	40.8	34.1	36.5	34.4	35.1	41.3	44.0
Tompson et al. [37]	96.1	91.9	83.9	77.8	80.9	72.3	64.8	84.5	82.0
IEF	95.7	91.6	81.5	72.4	82.7	73.1	66.4	82.0	81.3
Tompson et al. [37]	83.4	77.5	67.5	59.8	64.6	55.6	46.1	68.3	66.0
IEF	95.5	91.6	81.5	72.4	82.7	73.1	66.9	81.9	81.3

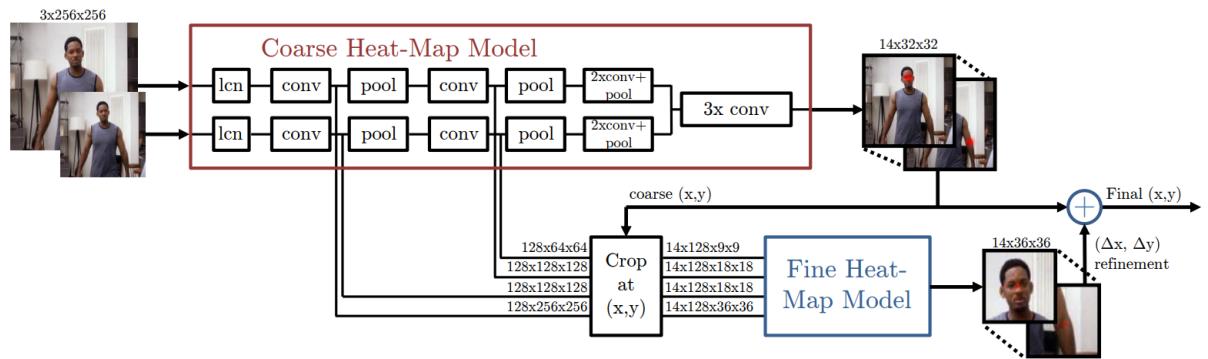
* Heatmap regression:

c) Efficient Object Localization Using Convolutional Networks

- Phương pháp này sẽ tạo ra một bản đồ nhiệt heatmap biểu thị xác suất xuất hiện điểm khớp cho từng khớp riêng lẻ.
- Input: Ảnh của một người
- Output: Dãy ảnh heatmap cho từng bộ phận trên cơ thể người
- Output xác thực: ở phương pháp này hàm gauss sẽ được áp lên từng khớp (x, y) ground truth với phương sai $\sigma = 1.5$ (pixel).



- Cho ảnh đi qua nhiều lớp tích chập song song nhau, để có được đặc trưng ở những tỉ lệ scale khác nhau.



- Hàm lỗi dùng để tối ưu ở giai đoạn thô:

$$E_1 = \frac{1}{N} \sum_{j=1}^N \sum_{xy} \|H'_j(x, y) - H_j(x, y)\|^2 \quad (1)$$

Where H'_j and H_j are the predicted and ground truth heat-maps respectively for the j th joint.

- Hàm lỗi dùng để tối ưu ở giai đoạn tinh:

$$E_2 = \frac{1}{N} \sum_{j=1}^N \sum_{x,y} \|G'_j(x, y) - G_j(x, y)\|^2 \quad (2)$$

Where G' and G are the set of predicted and ground truth heat-maps respectively for the fine heat-map model.

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Upper Body	Full Body
Gkioxari et al.	-	36.3	26.1	15.3	-	-	-	25.9	-
Sapp & Taskar	-	38.0	26.3	19.3	-	-	-	27.9	-
Yang & Ramanan	73.2	56.2	41.3	32.1	36.2	33.2	34.5	43.2	44.5
Pishchulin et al.	74.2	49.0	40.8	34.1	36.5	34.4	35.1	41.3	44.0
This work - scale normalized	96.1	91.9	83.9	77.8	80.9	72.3	64.8	84.5	82.0
This work - scale normalized (test only)	93.5	87.5	75.5	67.8	68.3	60.3	51.7	77.0	73.3
This work - unnormalized	83.4	77.5	67.5	59.8	64.6	55.6	46.1	68.3	66.0

Table 4: Comparison with prior-art: MPII (PCKh @ 0.5)

- Heatmap regression cho ra kết quả tốt hơn so với joint regression. Tuy nhiên phương pháp này vẫn thiếu về ngữ nghĩa của con người. Ví dụ như chưa xét về việc cơ thể người đối xứng, tính kết nối của khớp người... Các phương pháp sau sẽ cải thiện điều này, giúp nhận diện được các khớp đơn giản và có thể mở rộng nội suy ra phần bị che khuất.

* **Top-down:**

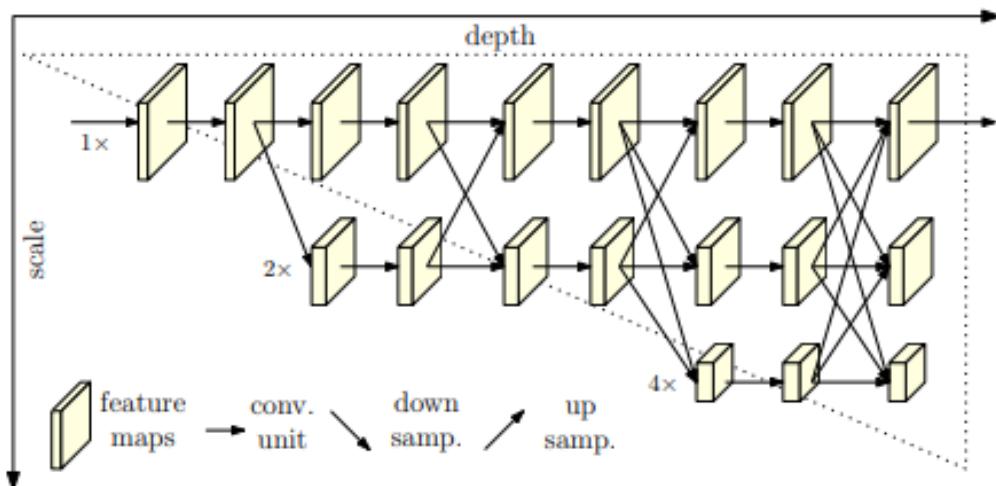
d) High-Resolution Net (HRNet)

- Ý tưởng của HRNet khác với phần lớn các hệ thống nhận diện khung xương top-down trước đó, chủ yếu áp dụng các mô hình high-to-low để tạo heatmap và xác định vị trí các key point thông qua heatmap đó. Sau đó sử dụng mô hình low-to-high để khôi phục lại hình ảnh gốc sau khi nhận diện được vị trí các key point. Vì phải trả qua quá trình khôi phục từ ảnh có độ phân giải thấp nên có thể xảy ra hiện tượng mất các chi tiết trong ảnh sau hồi phục.
- HRNet giữ được độ phân giải cao cho ảnh kết quả nhờ việc kết nối nhiều mạnh high-to-low song song với nhau thay vì chỉ kết nối liên tiếp nhau và duy trì mức độ phân giải cao liên tục trong toàn bộ quá trình thay vì việc chỉ khôi phục từ ảnh có độ phân giải thấp. Nhờ đó, kết quả của HRNet vẫn đạt được việc xác định key points đồng thời giữ được độ phân giải tốt như ban đầu.
- Input: là ảnh bounding box của một người

- Output: K ảnh heatmap $W \times H \{H_1, H_2, H_3, \dots, H_K\}$, mỗi heatmap mang giá trị dự đoán của key point thứ k



- Kiến trúc của HRNet bao gồm chuỗi mạng con high-to-low đa độ phân giải các chuỗi mạng con này có tính chất là song song nhau và các mạng con có độ phân giải khác nhau vẫn liên tục trao đổi thông tin với nhau.



- Hàm lỗi được dùng trong HRNet là Mean Squared Error, có công thức tổng quát như sau:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Trong đó: N là tổng số khớp tương ứng với các ảnh heatmap, y_i là heatmap của khớp thứ i, \hat{y}_i là heatmap xác thực của của khớp thứ i. Heatmap xác thực này được tạo ra bằng cách sử dụng bộ lọc Gaussian 2D lên tọa độ của các điểm khớp trong ảnh với độ lệch chuẩn là 1.

Table 2. Comparisons on the COCO test-dev set. #Params and FLOPs are calculated for the pose estimation network, and those for human detection and keypoint grouping are not included.

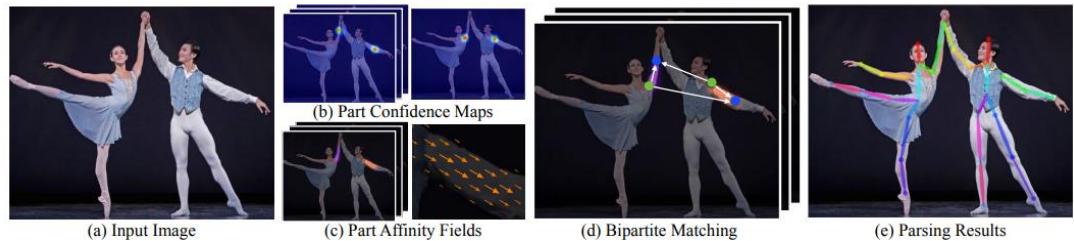
Method	Backbone	Input size	#Params	GFLOPs	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L	AR
Bottom-up: keypoint detection and grouping										
OpenPose [6]	–	–	–	–	61.8	84.9	67.5	57.1	68.2	66.5
Associative Embedding [39]	–	–	–	–	65.5	86.8	72.3	60.6	72.6	70.2
PersonLab [46]	–	–	–	–	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet [33]	–	–	–	–	69.6	86.3	76.6	65.0	76.3	73.5
Top-down: human detection and single-person keypoint detection										
Mask-RCNN [21]	ResNet-50-FPN	–	–	–	63.1	87.3	68.7	57.8	71.4	–
G-RMI [47]	ResNet-101	353 × 257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression [60]	ResNet-101	256 × 256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	–
G-RMI + extra data [47]	ResNet-101	353 × 257	42.6M	57.0	68.5	87.1	75.5	65.8	73.3	73.3
CPN [11]	ResNet-Inception	384 × 288	–	–	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [17]	PyraNet [77]	320 × 256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	–
CFN [25]	–	–	–	–	72.6	86.1	69.7	78.3	64.1	–
CPN (ensemble) [11]	ResNet-Inception	384 × 288	–	–	73.0	91.7	80.9	69.5	78.1	79.0
SimpleBaseline [72]	ResNet-152	384 × 288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNet-W32	HRNet-W32	384 × 288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNet-W48	HRNet-W48	384 × 288	63.6M	32.9	75.5	92.5	83.3	71.9	81.5	80.5
HRNet-W48 + extra data	HRNet-W48	384 × 288	63.6M	32.9	77.0	92.7	84.5	73.4	83.1	82.0

* Bottom-up:

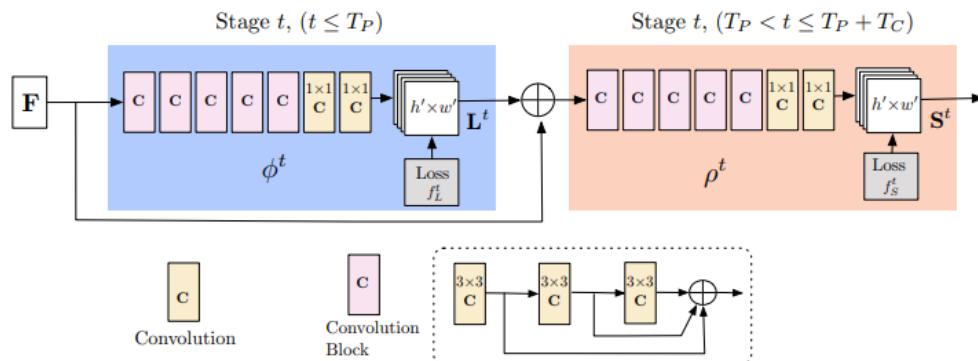
e) OpenPose

- OpenPose là một phương pháp nhận diện tư thế hoạt động theo nguyên tắc bottom-up, nghĩa là nó cũng gặp một vấn đề như những phương pháp bottom-up trước đó là tồn nhiều thời gian cho bước liên kết các key points thành các khung xương riêng biệt, lên đến vài phút cho một hình ảnh và không đáp ứng được yêu cầu nhận diện real-time.
- OpenPose giải quyết vấn đề này bằng cách sử dụng Part Affinity Fields (PAFs), một tập vector 2 chiều biểu diễn vị trí và hướng của các bộ phận (key points) trên ảnh. Nhờ việc sử dụng PAFs mà quá trình liên kết key points cuối cùng diễn ra nhanh hơn và hiệu quả hơn.
- Bài toán của OpenPose là xác định vị trí key point của tất cả mọi người trong ảnh cùng lúc. Đầu vào của OpenPose là bức ảnh có kích thước W^*H^*3 . Đầu tiên, một feedforward network sẽ dự đoán tập bản đồ độ tin cậy S và trường vector 2 chiều L. Trong đó, S bao gồm K bản đồ độ tin cậy $\{S_1, S_2, \dots, S_K\}$ với mỗi bản đồ thể hiện giá trị dự đoán 1 bộ phận, L gồm C trường vector $\{L_1, L_2, \dots, L_C\}$ với mỗi trường vector thể hiện mức độ

liên kết của 2 bộ phận kế nhau. Sau đó kết hợp các bản đồ độ tin cậy và trường vector lại để liên kết các điểm key points cho phù hợp từ đó tạo ra dáng người hoàn chỉnh.



- Kiến trúc của OpenPose gồm 2 phần lớn, phần màu xanh dự đoán mức độ liên kết giữa các bộ phận (PAFs), phần màu be dự đoán các confidence maps. Trong hình miêu tả giai đoạn thứ t, OpenPose gồm T giai đoạn lặp lại để tinh chỉnh kết quả của giai đoạn trước đó rồi mới đưa kết quả ra ngoài.



- Hàm loss tại giai đoạn t của confidence map và PAFs được định nghĩa như sau:

$$f_S^t = \sum_{j=1}^J \sum_P W(P) * \|S_j^t(p) - S_j^*(p)\|_2^2$$

$$f_L^t = \sum_{c=1}^C \sum_P W(P) * \|L_c^t(p) - L_c^*(p)\|_2^2$$

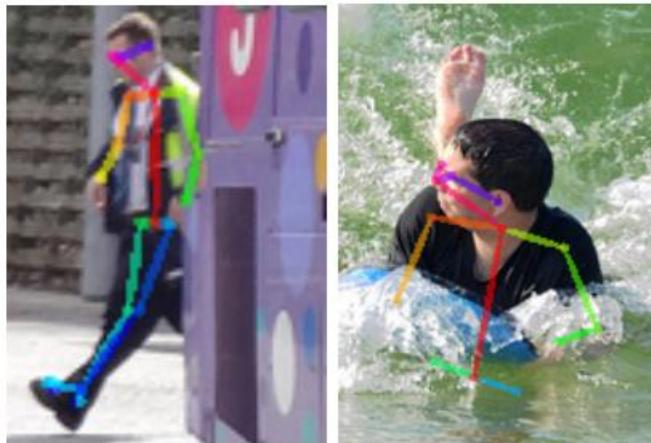
Trong đó: S_j^t là confidence map dự đoán được của khớp j ở giai đoạn t, S_j^* là confidence map xác thực của khớp j, L_c^t là giá trị PAF dự đoán được của liên kết c tại giai đoạn t, L_c^* là giá trị PAF xác thực của liên kết c, p là toạ độ pixel thứ p. W là

Trang 22

binary mask dùng để tránh trường hợp loại bỏ những dự đoán mang tính true positive trong quá trình training.

- Hạn chế của OpenPose:

- + Bộ phận cơ thể bị che khuất dẫn đến thiếu khớp và nhận diện sai.



- + Các bộ phận của hai người khác nhau bị chồng lên nhau dẫn đến hệ thống thành một bộ phận và đưa cho 1 người, người còn lại bị thiếu.



- + Nhận diện nhầm các bức tượng thành người.



Team	AP	AP ⁵⁰	AP ⁷⁵	AP ^M	AP ^L
Top-Down Approaches					
Megvii [43]	78.1	94.1	85.9	74.5	83.3
MRSA [44]	76.5	92.4	84.0	73.0	82.7
The Sea Monsters*	75.9	92.1	83.0	71.7	82.1
Alpha-Pose [6]	71.0	87.9	77.7	69.0	75.2
Mask R-CNN [5]	69.2	90.4	76.0	64.9	76.3
Bottom-Up Approaches					
METU [50]	70.5	87.7	77.2	66.1	77.3
TFMAN*	70.2	89.2	77.0	65.6	76.3
PersonLab [49]	68.7	89.0	75.4	64.1	75.5
Associative Emb. [48]	65.5	86.8	72.3	60.6	72.6
Ours	64.2	86.2	70.1	61.0	68.8
Ours [3]	61.8	84.9	67.5	57.1	68.2

TABLE 3: COCO test-dev leaderboard [73], “*” indicates that no citation was provided. Top: some of the highest top-down results. Bottom: highest bottom-up results.

Đánh giá mô hình dựa trên mAP với nhiều tỉ lệ thang đo PCKh

* Kết hợp cả top-down và bottom-up:

f) Stacked Hourglass Networks for Human Pose Estimation

- Là một bài báo mang tính bước ngoặc đánh bại tất cả các phương pháp trên. Nó gọi là stacked hourglass network vì cấu trúc mạng gồm pooling (giảm kích thước) và upsampling (tăng kích thước) nhìn giống như đồng hồ cát và chúng xếp chồng lên nhau. Với nhu cầu nắm bắt thông tin ở mọi tỉ lệ scale, vì kết quả cuối cùng sẽ là ngữ nghĩa toàn cục, nghĩa là những thông tin cần

thiết ở các tỉ lệ scale khác nhau sẽ đóng góp vào ảnh kết quả cuối cùng (ảnh heatmap).

- Với cấu trúc mạng đặc biệt có thể tiến hành cả top-down và bottom-up cùng một lúc.

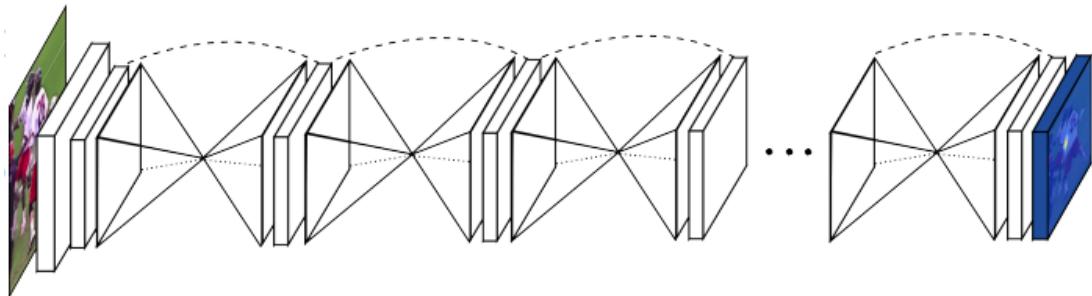


Fig. 1. Our network for pose estimation consists of multiple stacked hourglass modules which allow for repeated bottom-up, top-down inference.

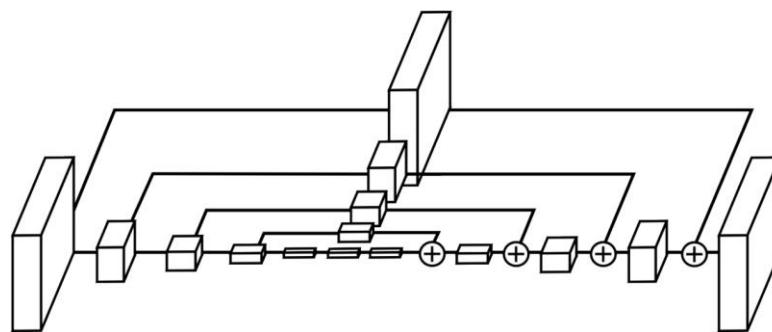
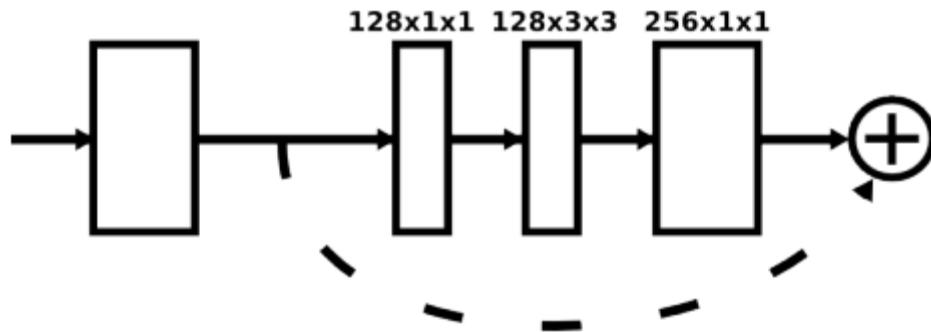
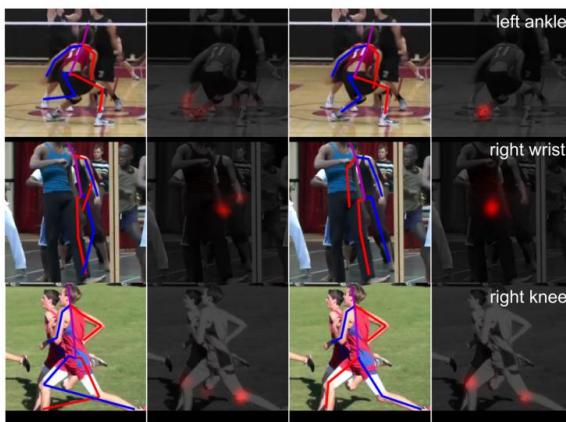


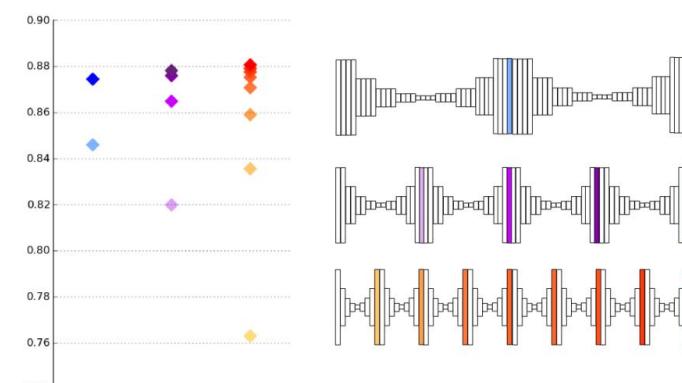
Fig. 3. An illustration of a single “hourglass” module. Each box in the figure corresponds to a residual module as seen in Figure 4. The number of features is consistent across the whole hourglass.



- Mạng có sử dụng các residual block kế thừa từ Resnet50.



Intermediate Prediction Accuracy (Validation, PCKh@0.5)



- Hàm lỗi để train mô hình là Mean Square Error: giữa ảnh heat map và ground truth heatmap.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- Khi xếp chồng các đồng hồ cát càng nhiều thì độ chính xác ngày càng cao ở layer cuối cùng.

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Tompson et al. [16], CVPR'15	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Carreira et al. [19], CVPR'16	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Pishchulin et al. [17], CVPR'16	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Hu et al. [27], CVPR'16	95.0	91.6	83.0	76.6	81.9	74.5	69.5	82.4
Wei et al. [18], CVPR'16	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Our model	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9

Table 2. Results on MPII Human Pose (PCKh@0.5)

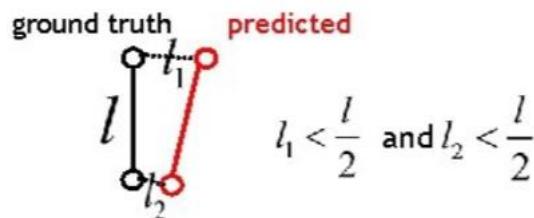
- Do lưu giữ lại thông tin trên các tỉ lệ scale khác nhau, đảm bảo đặc trưng toàn cục và địa phương được giữ lại để mạng có thể học được kết quả tốt.

g) So sánh các phương pháp:

Để đánh giá hiệu quả của Human Pose Estimation ta có các thang đo như sau:

Percentage of Correct Parts (PCP): đo lường tỉ lệ phát hiện của các chi, trong đó một chi được xem xét là được detect nếu khoảng cách

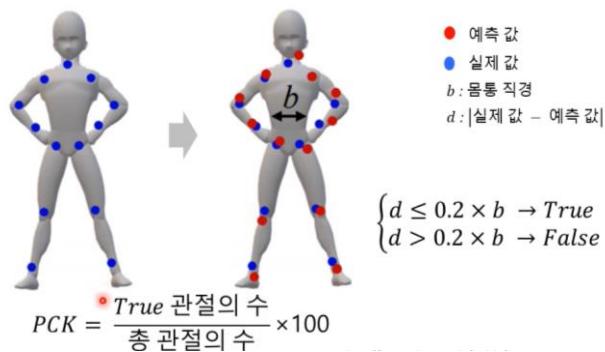
giữa vị trí khớp dự đoán và vị trí khớp thật bé hơn một nửa chiều dài của chi (PCP 0.5).



Percentage of Correct Key-points (PCK): một điểm khớp được xem xét là phát hiện nếu khoảng cách giữa điểm dự đoán và điểm ground truth trong một ngưỡng cho phép. Các ngưỡng sẽ là:

PCKh@0.5: ngưỡng sẽ là 50% chiều dài liên kết xương đầu

PCK@0.2: ngưỡng là 20% đường kính thân



Đôi khi ngưỡng cũng có thể là 150mm

Giúp giảm bớt vấn đề khớp ngắn, vì các chi nhỏ hơn sẽ có thân và liên kết đầu nhỏ hơn.

Percentage of Detected Joints (PJD): Khớp được cho là được detect nếu khoảng cách của khớp dự đoán và khớp thật vẫn nằm trong một phần đường kính của phần thân. Tỉ lệ này có thể khác nhau như (PJD 0.2 thì khoảng cách 2 khớp sẽ nhỏ hơn 0.2 lần đường kính thân người được xem xét)

Object Keypoint Similarity (OKS) based mAP:

$$\frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}$$

d_i là khoảng cách Euclidean của 2 khớp thứ i dự đoán và khớp thứ i ground truth.

s là tỉ lệ scale giữa bounding box và ảnh thật (cùng khoảng cách nhưng ở các tỉ lệ scale khác nhau mang ý nghĩa khác nhau)

k_i có thể xem như là một siêu tham số là hằng dương cho từng keypoint thứ i điều khiển độ rời của keypoint (chú ý số mũ âm nên số càng lớn càng tiến về 0 cũng như những keypoint quan trọng sẽ có siêu tham số nhỏ để tránh phạt lớn hơn những keypoint khác).

v_i là visible flag choàng khop có thể quan sát được ở ảnh thật vì việc tái tạo các khop bị khuất tầm nhìn có thể nói là không thể, nên tham số được dùng để tránh đưa những sai số cao vào đơn vị đo lường.

Metric **AP⁵⁰** là một người có khung xương được rút trích chính xác khi ngưỡng **OKS** = 0.5, tương tự với **AP⁷⁵**, AP là đo lường trung bình độ chính xác của 10 ví trí ngưỡng khác nhau **OKS** = 0.5, 0.55, ..., 0.9, 0.95, **AP^M** dành cho đối tượng trung bình, **AP^L** dành cho đối tượng lớn và **AR** (Average recall) của 10 ví trí ngưỡng khác nhau **OKS** = 0.5, 0.55, ..., 0.9, 0.95.

Metric	Description
AP	AP at OKS* = 0.50 : 0.05 : 0.95 (primary metric)
AP ^{0.5}	AP at OKS = 0.50
AP ^{0.75}	AP at OKS = 0.75
AP ^M	AP for medium objects: $32^2 < area < 96^2$
AP ^L	AP for large objects: $area > 96^2$

*OKS=Object Keypoint Similarity, same role as IoU

Trường phái	Phương pháp	Hàm lõi	Độ chính xác

Single-person pose estimation	Keypoint Regression	DeepPose	$\arg \min_{\theta} \sum_{(x,y) \in D_N} \sum_{i=1}^k \ \mathbf{y}_i - \psi_i(x; \theta)\ _2^2$	61% (PCP 0.5)
		HPE IEF	$\min_{\Theta_f, \Theta_g} \sum_{t=1}^T h(\epsilon_t, e(y, y_t))$	81.3% (PCKh@0.5)
	Heatmap Regression	EOLUC Networks	$E_1 = \frac{1}{N} \sum_{j=1}^N \sum_{x,y} \ H'_j(x, y) - H_j(x, y)\ ^2 \quad (1)$ <p>Where H'_j and H_j are the predicted and ground truth heat-maps respectively for the jth joint.</p> $E_2 = \frac{1}{N} \sum_{j=1}^N \sum_{x,y} \ G'_j(x, y) - G_j(x, y)\ ^2 \quad (2)$ <p>Where G' and G are the set of predicted and ground truth heat-maps respectively for the fine heat-map model.</p>	82% (PCKh@0.5)
Multi-person pose estimation	Top-down	HRNet	$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$	77% (AP) 92.7% (AP ⁵⁰) 84.5% (AP ⁷⁵) 82% (AR)
	Bottom-up	Open pose	$f_S^t = \sum_{j=1}^J \sum_P W(P) * \ S_j^t(p) - S_j^*(p)\ ^2$ $f_L^t = \sum_{c=1}^C \sum_P W(P) * \ L_c^t(p) - L_c^*(p)\ ^2$	61.8 (AP) 84.9% (AP ⁵⁰) 67.5% (AP ⁷⁵)
	Top-down và Bottom-up	Hourglass Networks	$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$	90.9% (PCKh@0.5)

2. Một số phương pháp phân lớp hành vi

Tên phương pháp	Mạng sử dụng	Input	Hàm lỗi	Đặc điểm nổi bật	Độ chính xác
-----------------	--------------	-------	---------	------------------	--------------

Semantics-Guided Neural Networks	Semantics-Guided Neural Network (SGN)	Chuỗi khung xương gốc từ giai đoạn nhận diện khung xương	Cross Entropy	Gồm 2 module: joint-level module (lấy thông tin về sự tương quan các khớp trong một khung xương) và frame-level module (lấy thông tin về sự tương quan giữa các frame) để xác định nhãn của một hành vi	94.5% (NTU60 RGB+D, Cross-View Setting) 81.5% (NTU120 RGB+D, Cross-View Setting)
Revisiting Skeleton-based Action Reconigtion	PoseConv3D	Dãy ảnh heatmap $K*T*H*W$ với K là số khớp, T là số frame, H và K là kích thước ảnh	Cross Entropy	Xử lý input đầu vào từ heatmap 2D tạo thành heatmap 3D và dùng mạng 3D-CNN thay vì GCN để phân lớp hành vi	94.1% (NTU60 RGB+D, Cross-Subject Setting)
Double-feature Double-motion Network	DD-Net	Dãy các JCD (Joint Collection Distance), là khoảng cách giữa các khớp trong một khung xương	Cross Entropy	Sử dụng một cấu trúc input mới (JCD) để xử lý vấn đề về toạ độ - góc nhìn. Kết hợp với việc xử lý dữ liệu thời gian ở cả hai	94.6% (SHREC, filters=64) 93.5% (SHREC, filters=32)

				điều kiện slow motion và fast motion để phân lớp hành vi chính xác hơn trong nhiều điều kiện khác nhau	91.8% (SHREC, filters=16) 77.2% (JHMDB, filters=64) 73.7% (JHMDB, filters=32) 65.7% (JHMDB, filters=16)
Spatial Temporal Graph Convolutional Networks	ST-GCN	Một đồ thị G gồm hai phần, phần thứ nhất mang thông tin về toạ độ các khớp, phần thứ hai mang thông tin về các cạnh bên trong 1 khung xương và giữa các khung xương với nhau	Cross Entropy	Sử dụng cấu trúc đồ thị để giải quyết hai vấn đề cùng một lúc, thứ nhất là vấn đề về không gian (toạ độ của khớp), thứ hai là thời gian (sự tương quan giữa các frame)	81.5% (NTU RGB+D, Cross-Subject Setting) 88.3% (NTU RGB+D, Cross-View Setting)
Spatio-temporal Tuples Transformer	STTFormer	Dãy “tuple”, mỗi tuple gồm nhiều frame hình liên tiếp nhau	Cross Entropy	Sử dụng mô hình mới (Transformer) bằng cách chia input đầu vào	89.9% (NTU RGB+D, Cross-

				thành nhiều chuỗi nhỏ để xử lý, giúp lấy được sự tương quan giữa các frame liên tiếp đồng thời mà không làm tăng chi phí tính toán	Subject Setting) 94.3% (NTU RGB+D, Cross-View Setting)
--	--	--	--	--	---

a) Semantics-Guided Neural Networks

- Phương pháp này chú trọng đến cả mặt ngữ nghĩa (semantics information) và thông tin tạm thời (temporal information) của thông tin khung xương đầu vào thay vì chỉ quan tâm đến toạ độ như hầu hết phương pháp trước đó. Ví dụ như xét một khớp có toạ độ ở trên đầu, nếu khớp này mang thông tin là tay, thì đó có thể là hành động vẫy tay, còn nếu khớp mang thông tin là chân, thì có thể là hành động đá cao, còn nếu xét hai hành động đứng lên và ngồi xuống, chuỗi khung xương của hai hành động này giống nhau nhưng chỉ khác thứ tự các frame.
- Việc sử dụng thêm 2 loại thông tin này giúp quá trình đối chiếu và so sánh trở nên hiệu quả hơn và giảm bớt số lượng thông tin phải xử lý.
- Input: chuỗi khung xương $S = \{X_t^k | t = 1,2,3, \dots, T; k = 1,2,3, \dots, J\}$. Với T là tổng số frame, J là tổng số loại khớp, X_t^k là khớp loại k tại frame thứ t.
- Output: dự đoán hành động của khung xương, bao gồm nhãn của hành động và giá trị dự đoán.
- Kiến trúc của SGN thể hiện qua sơ đồ sau

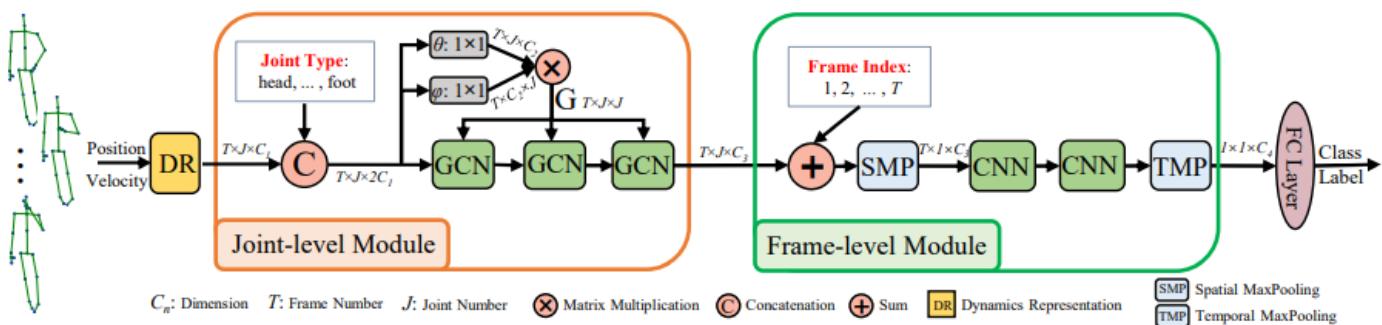


Figure 2: Framework of the proposed end-to-end Semantics-Guided Neural Network (SGN). It consists of a joint-level module and a frame-level module. In DR, we learn the dynamics representation of a joint by fusing the position and velocity information of a joint. Two types of semantics, *i.e.*, joint type and frame index, are incorporated into the joint-level module and the frame-level module, respectively. To model the dependencies of joints in the joint-level module, we use three GCN layers. To model the dependencies of frames, we use two CNN layers.

- Hàm lỗi được sử dụng là Cross Entropy, có công thức là:

$$L_{CE} = - \sum_{i=1}^n t_i * \log(p_i)$$

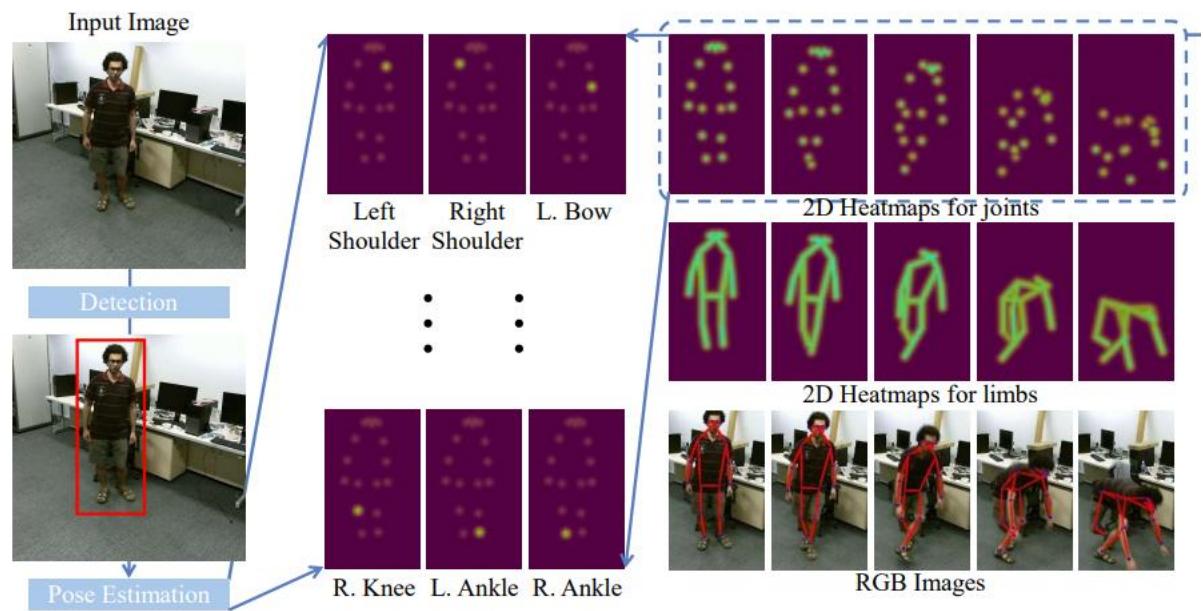
Trong đó, n là tổng số lớp, t_i là nhãn đúng của lớp thứ i và p_i là xác suất dự đoán của lớp đó.

Table 4: Performance comparisons on NTU60 with the CS and CV settings in terms of accuracy (%).

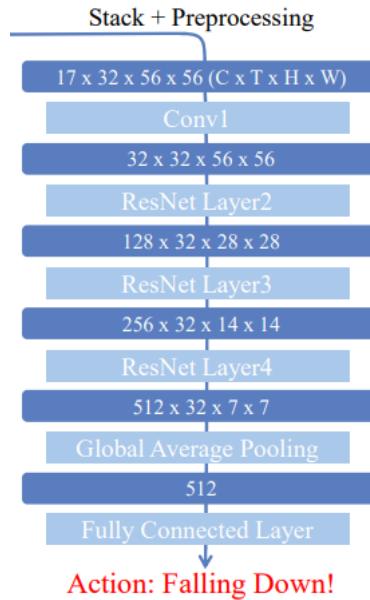
Method	Year	CS	CV
HBRNN-L [7]	2015	59.1	64.0
Part-Aware LSTM [36]	2016	62.9	70.3
ST-LSTM + Trust Gate [27]	2016	69.2	77.7
STA-LSTM [41]	2017	73.4	81.2
GCA-LSTM [29]	2017	74.4	82.8
Clips+CNN+MTLN [18]	2017	79.6	84.8
VA-LSTM [57]	2017	79.4	87.6
ElAtt-GRU[59]	2018	80.7	88.4
ST-GCN [54]	2018	81.5	88.3
DPRL+GCNN [44]	2018	83.5	89.8
SR-TSL [40]	2018	84.8	92.4
HCN [23]	2018	86.5	91.1
AGC-LSTM (joint) [39]	2019	87.5	93.5
AS-GCN [24]	2019	86.8	94.2
GR-GCN [8]	2019	87.5	94.3
2s-AGCN [37]	2019	88.5	95.1
VA-CNN [58]	2019	88.7	94.3
SGN w/o Sem.	-	86.9	92.8
SGN	-	89.0	94.5

b) Revisiting Skeleton-based Action Reconigton

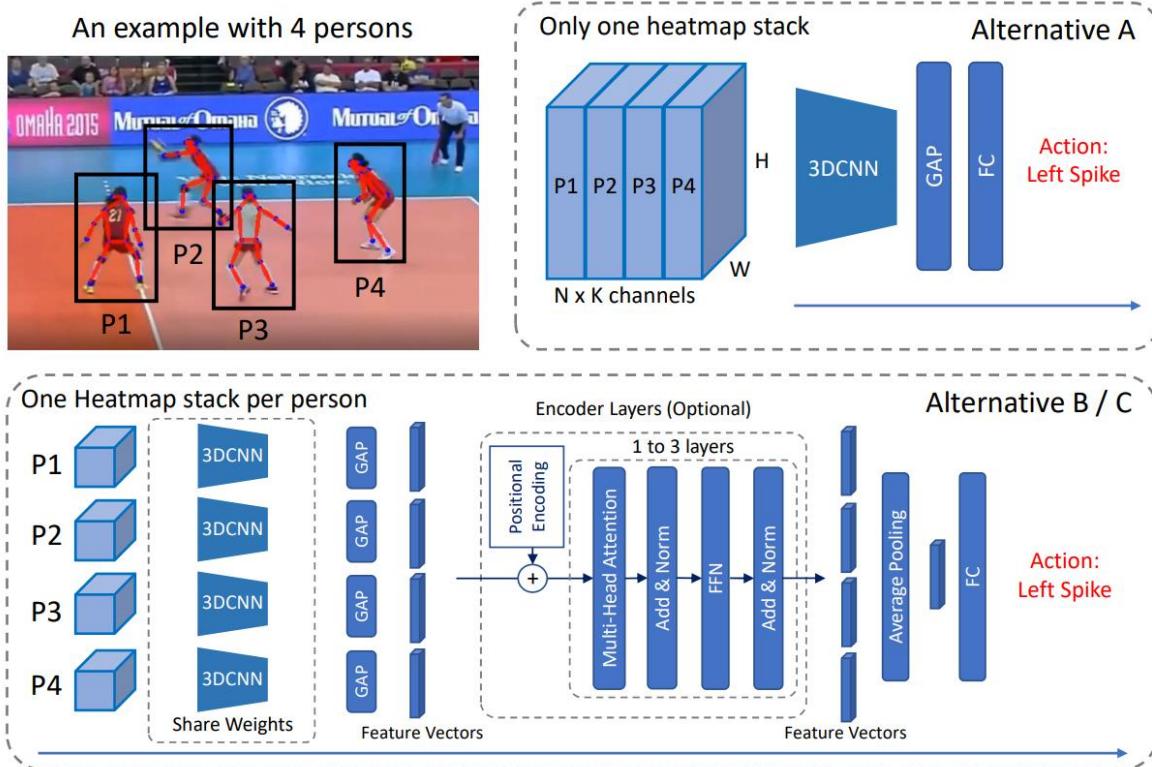
- Cũng là một phương pháp nhận dạng hành động dựa trên pose estimation, nhưng dùng Convolution 3D.
- Input: Dãy ảnh heatmap $K \times W \times H$ (K : số khớp) được dự đoán cho một người trong T frame liên tiếp $K \times T \times W \times H$ (T : số frame) gọi là 3D heatmap volume.



- Output: nhãn của hành động để dự đoán.



- Khi áp dụng cho một nhóm người input sẽ bao gồm tất cả ảnh heat map của tất cả mọi người trong T frame, $N*K*T*H*W$.



- Output: sẽ là nhãn hành động của một nhóm người.

Table 17. Uniform sampling also works for RGB-based action recognition. All results are for 10-clip testing, except the ‘uniform-16 (1c)’, which uses 1-clip testing.

(a) FineGYM.		(b) NTU-60 (X-Sub)	
Sampling	Mean-Top1	Sampling	Top1
16x2	87.9	16x2	94.9
16x4	88.7	16x4	95.1
uniform-16 (1c)	91.1	uniform-16 (1c)	95.7
uniform-16	91.6	uniform-16	96.1

c) Double-feature Double-motion Network

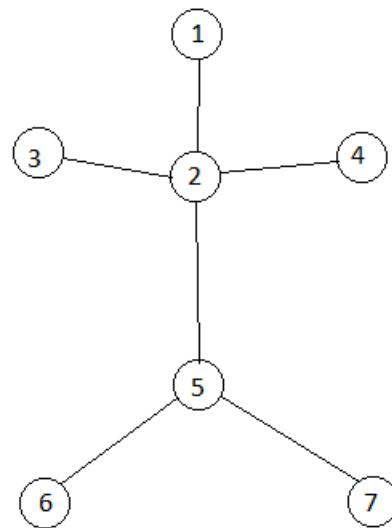
- Những phương pháp nhận diện hành động trước đây thường gặp những vấn đề như:

- + Vị trí – góc nhìn: tuỳ thuộc vào góc quay và góc toạ độ của hệ trực đang xét mà hai hành động hay hai điểm có thể được xem là khác nhau nhưng thực chất là một. Hệ thống phải nhận diện và cố định được các bộ phận thay vì phải nhận diện lại liên tục.
- + Tỉ lệ chuyển động: tốc độ của chuyển động là không cố định, có thể nhanh hoặc chậm, khi chậm thì sẽ có nhiều khung hình hơn và mỗi khung hình sẽ có sự thay đổi ít hơn, còn khi nhanh thì sẽ có ít khung hình và sự khác biệt giữa mỗi khung hình là rất lớn nên hệ thống phải mang tính bao quát, xử lí được nhiều loại tốc độ mà không làm mất đi thông tin của chuyển động.
- DDNet giải quyết các vấn đề trên bằng cách sử dụng cấu trúc dữ liệu mới là Joint Collection Distances (JCD) thay cho toạ độ thông thường và xem xét cả hai tỉ lệ chuyển động cùng lúc trong quá trình nhận dạng hành vi.
- Joint Collection Distances (JCD): là nửa dưới (không tính đường chéo) của ma trận đối xứng thể hiện khoảng cách Euclidean giữa các khớp. Bằng cách sử dụng JCD, DDNet đã xử lý được vấn đề vị trí – góc nhìn do JCD chỉ lưu khoảng cách giữa các khớp nên không phụ thuộc vào hệ toạ độ của ảnh hay góc nhìn của camera. Gọi tổng số khớp là N, khớp thứ i được có toạ độ trong không gian 2 chiều là J_i^k , ma trận JCD tại frame thứ k là:

$$JCD^k = \begin{bmatrix} \|\overrightarrow{J_2^k J_1^k}\| & & & \\ \|\overrightarrow{J_3^k J_1^k}\| & \|\overrightarrow{J_3^k J_2^k}\| & & \\ \dots & \dots & \dots & \\ \|\overrightarrow{J_N^k J_1^k}\| & \|\overrightarrow{J_N^k J_2^k}\| & \dots & \|\overrightarrow{J_N^k J_{N-1}^k}\| \end{bmatrix}$$

Trong đó: $\|\overrightarrow{J_i^k J_j^k}\|$ là khoảng ách Euclidean giữa J_i^k và J_j^k

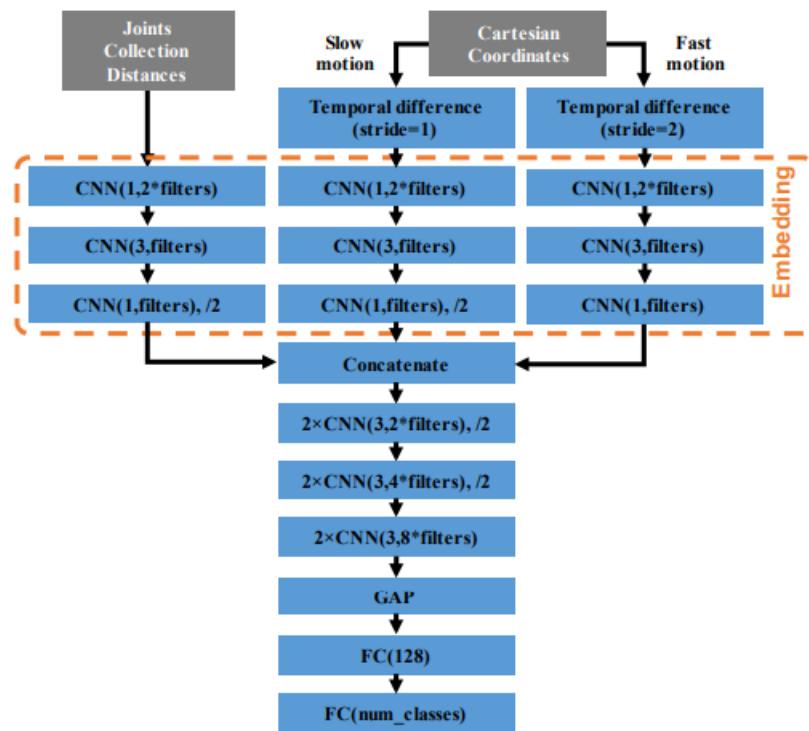
Ví dụ: có 7 khớp tạo nên 1 khung xương như sau:



Ta có khoảng cách giữa hai khớp i và j kí hiệu là D_{ij} và $D_{ij}=D_{ji}$, ta thu được ma trận khoảng cách giữa các khớp như sau, phần màu đỏ chính là JCD được sử dụng trong DDNet:

$$\begin{bmatrix} D_{11} & D_{12} & \dots & D_{16} & D_{17} \\ D_{21} & D_{22} & \dots & D_{26} & D_{27} \\ \dots & \dots & \dots & \dots & \dots \\ D_{61} & D_{62} & \dots & D_{66} & D_{67} \\ D_{71} & D_{72} & \dots & D_{76} & D_{77} \end{bmatrix}$$

- Kiến trúc của DDNet:



- Hàm lỗi được sử dụng là Cross Entropy, có công thức là:

$$L_{CE} = - \sum_{i=1}^n t_i * \log(p_i)$$

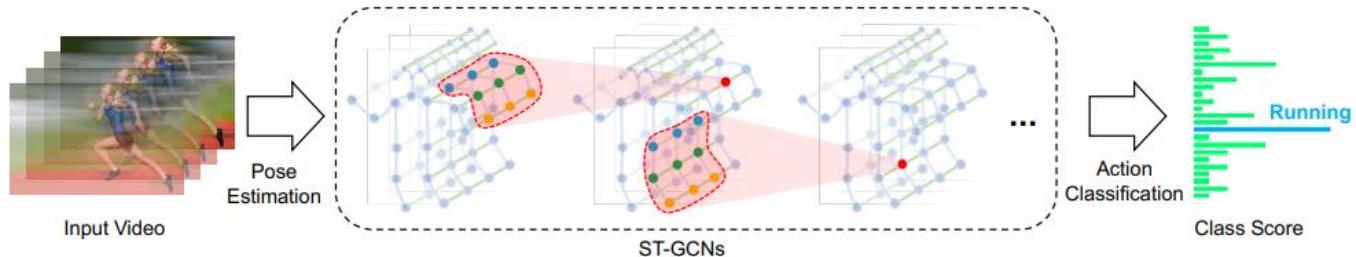
Trong đó, n là tổng số lớp, t_i là nhãn đúng của lớp thứ i và p_i là xác suất dự đoán của lớp đó.

d) Spatial Temporal Graph Convolutional Networks

- Phương pháp này là một trong những phương pháp đầu tiên áp dụng mạng nơ-ron sử dụng đồ thị để nhận diện hành vi thay vì sử dụng các toạ độ 2D hay 3D của các khớp xương như thông thường.
- Input: đồ thị không gian – thời gian $G = (V, E)$ dựa trên chuỗi khung xương ban đầu có N khớp và T frame. Trong đó, tập điểm $V = \{v_{ti} | t = 1, \dots, T; i = 1, \dots, N\}$ mang thông tin về tất cả các khớp xương trong chuỗi khung xương. Tập cạnh E gồm 2 thành phần, phần thứ nhất là tập các cạnh nối các khớp trong cùng 1 khung xương $E_S = \{v_{ti}v_{tj} | (i, j) \in H\}$ (H là khung

xuong tại frame t), phần thứ hai là tập các cạnh nối cùng một khớp qua các frame $E_F = \{v_{ti}v_{(t+1)i}\}$.

- Output: nhãn của hành động và giá trị dự đoán của nhãn đó.
- Sơ đồ pipeline:



- Kiến trúc của ST-GCN (Spatial-Temporal Graph Convolutional Networks) gồm 9 lớp toán tử tích chập đồ thị không gian – thời gian (ST-GCN units). Ba lớp đầu tiên có 64 channels output, ba lớp tiếp theo có 128 channels output, ba lớp cuối cùng có 256 channels output. Các lớp trên gồm 9 kernel thời gian (temporal kernel) có kích thước khác nhau. Cơ chế Resnet được sử dụng trên mỗi lớp. Để tránh hiện tượng overfitting thì sau mỗi lớp, mạng sẽ bỏ ngẫu nhiên một số features với tỉ lệ 0.5. Lớp thứ 4 và lớp thứ 7 được cài đặt thành các lớp pooling. Sau đó, một quá trình pooling toàn cục được tiến hành trên tensor kết quả để thu feature vector 256 chiều. Cuối cùng là đưa vector này vào bộ phân lớp Softmax.

- Hàm lỗi sử dụng là Cross Entropy, có công thức là:

$$L_{CE} = - \sum_{i=1}^n t_i * \log(p_i)$$

Trong đó, n là tổng số lớp, t_i là nhãn đúng của lớp thứ i và p_i là xác suất dự đoán của lớp đó.

	X-Sub	X-View
Lie Group (Veeriah, Zhuang, and Qi 2015)	50.1%	52.8%
H-RNN (Du, Wang, and Wang 2015)	59.1%	64.0%
Deep LSTM (Shahroudy et al. 2016)	60.7%	67.3%
PA-LSTM (Shahroudy et al. 2016)	62.9%	70.3%
ST-LSTM+TS (Liu et al. 2016)	69.2%	77.7%
Temporal Conv (Kim and Reiter 2017).	74.3%	83.1%
C-CNN + MTLN (Ke et al. 2017)	79.6%	84.8%
ST-GCN	81.5%	88.3%

Table 3: Skeleton based action recognition performance on NTU-RGB+D datasets. We report the accuracies on both the cross-subject (X-Sub) and cross-view (X-View) benchmarks.

e) Spatio-temporal Tuples Transformer (STTFormer)

- Phương pháp này chia chuỗi khung xương ban đầu thành nhiều phần gọi là “tuple”. Mỗi tuple gồm nhiều frame liên tiếp nhau, với mục đích nhận dạng hành động của người thông qua các hành động con.
- Input: chuỗi khung xương gồm V_0 khớp và có T_0 frame.
- Output: dự đoán hành động của khung xương, bao gồm nhãn của hành động và giá trị dự đoán.
- Cấu trúc của phương pháp này được thể hiện qua sơ đồ sau:

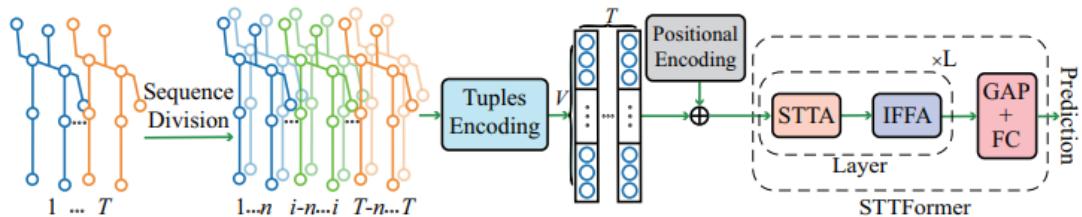


Fig. 2. Illustration of the overall architecture of the proposed model. which consists of two main modules: the spatio-temporal tuples encoding and spatio-temporal tuples Transformer.

- Kiến trúc của giai đoạn Tuples Encoding được miêu tả qua sơ đồ sau, với mục đích tạo ra các vector đặc trưng cho từng hành động con nắm giữ thông tin về hành động:

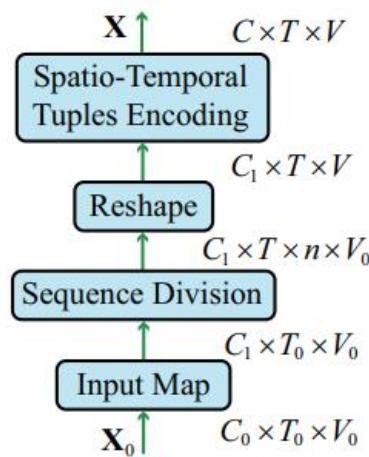


Fig. 3. Illustration of the proposed spatio-temporal tuples encoding module.

- Positional Encoding: sẽ được wise-element sum với đặc trưng sau khi qua Tuple Encoding, ở giai đoạn này các hành động con sẽ được cộng thêm thông tin về vị trí của hành động con (vd: hành động cuối lunge xuống có thể phân ra thành các hành động từ từ cuối xuống, nhưng nếu ta nhìn ngược các hành động này thì nó là hành động cuối lunge lên. Vì thế vị trí của hành động con cần được chú trọng)

$$PE(p, 2i) = \sin(p/10000^{2i/C_{in}})$$

$$PE(p, 2i + 1) = \cos(p/10000^{2i/C_{in}})$$

p là vị trí của khớp và i là số chiều của vector positional encoding.

- STTFormer gồm L lớp layer xếp chồng lên nhau, trong đó mỗi lớp layer gồm 2 phần chính là Spatio-Temporal Tuples Attention (phần cam) và Inter-Frame Feature Aggregation (phần tím), được nối với nhau bởi một mạng Feed Forward.

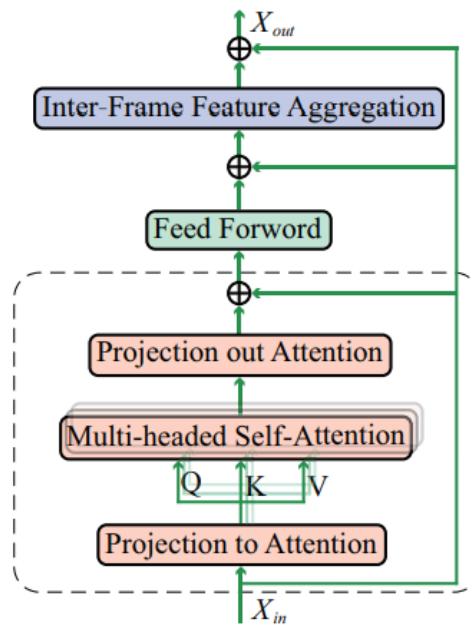


Fig.4. Illustration of the proposed spatio-temporal tuples Transformer layer, the complete STTFormer is stacked by L such layers.

- Hàm lỗi được sử dụng là Cross Entropy, có công thức là:

$$L_{CE} = - \sum_{i=1}^n t_i * \log(p_i)$$

Trong đó, n là tổng số lớp, t_i là nhãn đúng của lớp thứ i và p_i là xác suất dự đoán của lớp đó.

Methods	NTU RGB+D		NTU RGB+D 120	
	X-Sub (%)	X-View (%)	X-Sub (%)	X-Set (%)
MTCNN[10]	81.1	87.4	61.2	63.3
IndRNN[16]	81.8	88.0	-	-
HCN[13]	86.5	91.1	-	-
ST-GCN[38]	81.5	88.3	-	-
2s-AGCN[27]	88.5	95.1	82.9	84.9
DGNN[25]	89.9	96.1	-	-
Shift-GCN[5]	90.7	96.5	85.9	87.6
Dynamic-GCN[39]	91.5	96.0	85.9	87.6
MS-G3D[20]	91.5	96.2	86.9	88.4
MST-GCN[4]	91.5	96.6	87.5	88.8
ST-TR[23]	89.9	96.1	82.7	84.7
DSTA-Net[28]	91.5	96.4	86.6	89.0
STTFormer(Ours)	92.3	96.5	88.3	89.2

IV. METHOD

1. Sơ lược về mạng tích chập đồ thị (Graph convolution network)

Trình bày được dựa trên paper [GCN - Semi-Supervised Classification with Graph Convolutional Networks - 2016](#) hiện đang được cite nhiều nhất.

Trước khi đi vào graph convolution network. Sau đây là một số định nghĩa (đang xét đồ thị vô hướng):

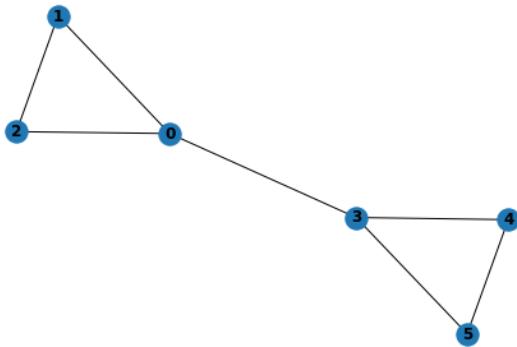
Ma trận đặc trưng: X (NxC) chứa thông tin đặc trưng của từng node, với C là số chiều không gian của đặc trưng.

Ma trận kè (adjacency matrix): A (NxN), chứa thông tin về các liên kết của N đỉnh trong đồ thị. Ví dụ đỉnh i và j có liên kết với nhau thì tại $A_{ij} = A_{ji} = 1$, i và j không có liên kết với nhau $A_{ij} = A_{ji} = 0$, và $A_{ii} = 0, i \leq N$

Ma trận đơn vị (identity matrix): I (NxN)

Ma trận bậc (degree matrix): Λ với $\Lambda_{ii} = \sum_j A_{ij} + I_{ii}$ dùng để chuẩn hóa hay cân bằng các đóng góp giữa các node có nhiều liên kết và ít liên kết sẽ trình bày ở phần sau.

Giả sử ta có 6 node trên đồ thị như sau:



Mỗi node mang thông tin đặc trưng 2 chiều, ta có được ma trận kề A(6x6) và ma trận X(6x2) chứa những thông tin về đặc trưng từng node.

Shape of A: (6, 6)

Shape of X: (6, 2)

Adjacency Matrix (A):

```
[[0. 1. 1. 1. 0. 0.]
 [1. 0. 1. 0. 0. 0.]
 [1. 1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 1. 1.]
 [0. 0. 0. 1. 0. 1.]
 [0. 0. 0. 1. 1. 0.]]
```

Node Features Matrix (X):

```
[[ 0.  0.]
 [ 1. -1.]
 [ 2. -2.]
 [ 3. -3.]
 [ 4. -4.]
 [ 5. -5.]]
```

Sau khi nhân AX thì ta có được tổng của các node láng giềng với từng node.

Dot product of A and X (AX):

```
[[ 6. -6.]
 [ 2. -2.]
 [ 1. -1.]
 [ 9. -9.]
 [ 8. -8.]
 [ 7. -7.]]
```

Trên đây là sơ lược về phép toán trên đồ thị tính toán, ta sẽ đi sâu

vào GCN. GCN định nghĩa layer tiếp theo như sau.

$$H^{(l+1)} = \sigma \left(\Lambda^{-\frac{1}{2}} (A + I) \Lambda^{-\frac{1}{2}} H^{(l)} W^{(l)} \right)$$

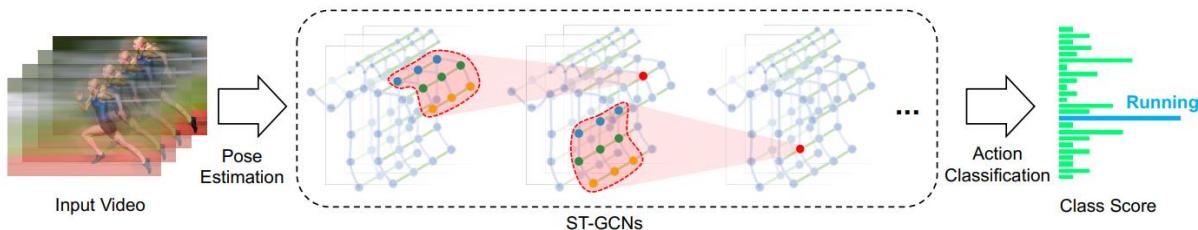
Với σ là hàm kích hoạt phi tuyến, $H^{(0)} = X$, $W^{(l)}$ là ma trận trọng số dùng để train, A là ma trận kè, $\Lambda^{-\frac{1}{2}}$ là ma trận bậc dùng để cân bằng đóng góp của các node nhiều liên kết và ít liên kết.

Giả sử $H^{(l)}$ là ma trận $(Nx C^{(l)})$ thì lần lượt kích thước của các ma trận là. Λ và A (NxN), W là ma trận $(C^{(l)} x C^{(l+1)})$.

2. Phương pháp

Sau khi tìm hiểu qua về một số phương pháp nhận dạng khung xương và phân lớp hành động phổ biến, chúng tôi lựa chọn sử dụng hai phương pháp là HRNet để nhận dạng khung xương và ST-GCN để phân lớp hành động. Một số lý do để lựa chọn hai phương pháp này là thứ nhất, HRNet có kiến trúc mới, khác biệt và được cải tiến nhiều hơn so với những kiến trúc trước đó nhưng vẫn không quá phức tạp để cài đặt, đây hiện được xem là một trong những thuật toán tiên tiến nhất trong lĩnh vực nhận dạng khung xương (the state-of-the art algorithm). Thứ hai, ST-GCN giải quyết tốt những vấn đề quan trọng của quá trình phân lớp hành vi, đó là thông tin về không gian và thời gian, thay vì xử lý những thông tin này một cách rời rạc như nhiều mô hình trước, ST-GCN xử lý chúng đồng thời bằng cách tạo ra một dữ liệu mới có thể mang thông tin cả về không gian và thời gian và xử lý song song nhau, tăng hiệu năng về độ chính xác của mức độ phân lớp và thời gian xử lý.

Đường ống mô hình như sau:



Đầu vào sẽ là một video, sau đó thông qua Pose Estimation (Ở đây sẽ sử dụng HRNet) để rút trích khung xương bằng cách phát hiện từng người rồi mới đưa vào mạng (Person detector: Faster RCNN). Sau khi có khung xương cho từng người trên từng frame ảnh. Sẽ bắt đầu tiến hành nhận dạng hành động (ST GCN).

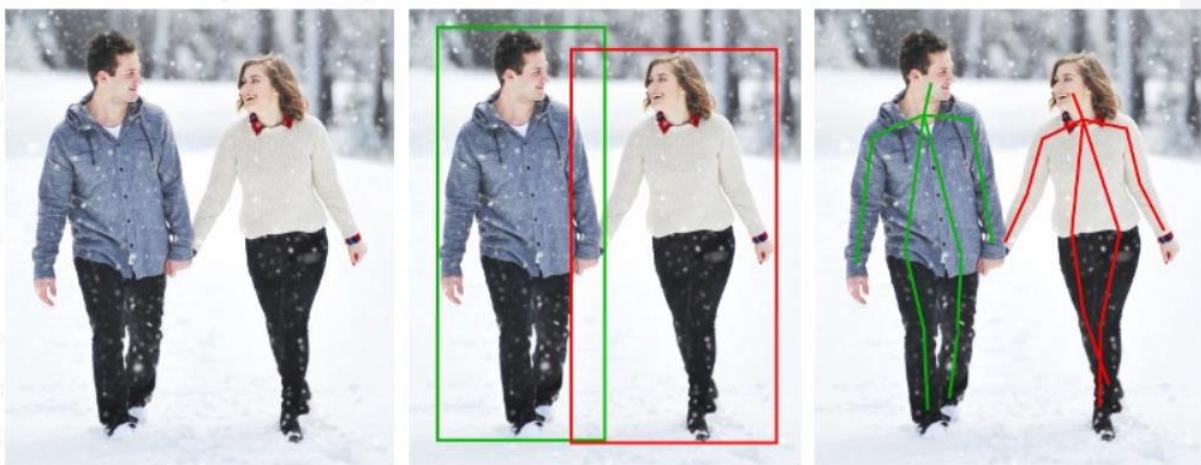
HRNet:

Ban đầu, video được chia thành nhiều frame ảnh, từng ảnh sẽ được đưa vào HRNet để xử lý, rút trích khung xương.

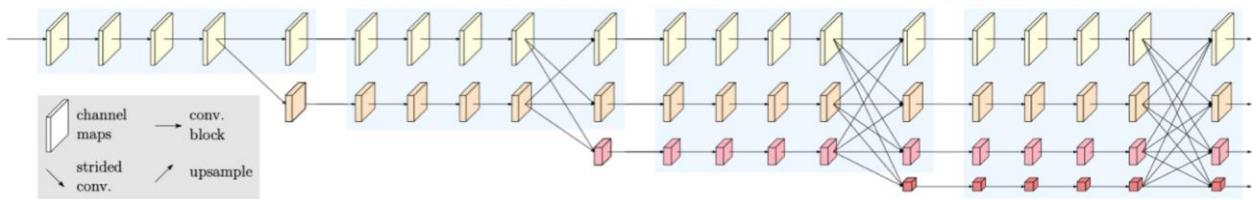
Input: tập ảnh I gồm T ảnh (T là số frame của video đầu vào), mỗi ảnh có kích thước $W \times H \times 3$, trong đó W là chiều rộng của ảnh, H là chiều cao của ảnh, 3 là số kênh màu, ở đây ở dụng ảnh RGB có 3 kênh màu.

Output: Tập heatmap H gồm $T \times K$ heatmap (T là số frame của video đầu vào), mỗi heatmap có kích thước $W' \times H'$. Một frame gồm K heatmap $\{H_{i1}, H_{i2}, \dots, H_{iK} | i = 1, 2, \dots, T\}$, heatmap thứ k mang thông tin confidence score của khớp thứ k, ở đây ta xét tổng cộng 18 khớp.

HRNet hoạt động theo cơ chế Top-down. Ảnh đầu vào trước hết sẽ được xử lý để nhận dạng các người có trong ảnh, ở đây chúng tôi sử dụng Faster RCNN để nhận dạng người, mỗi người sẽ được đóng một bounding box xung quanh. Sau đó, HRNet sẽ được áp dụng vào từng bounding box để nhận dạng các khớp xương của từng người và liên kết chúng lại để tạo ra một khung xương hoàn chỉnh.

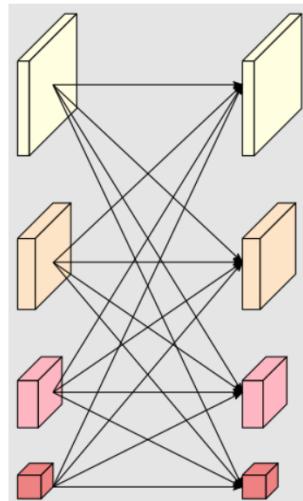


Sơ đồ kiến trúc của HRNet:

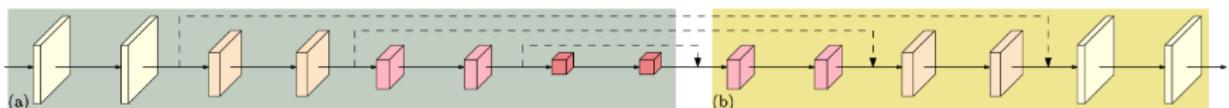


HRNet gồm nhiều khối tích chập (ở đây ta xét 4 khối), khối tích chập thứ n sẽ gồm n luồng song song ứng với n độ phân giải, độ phân giải sau sẽ nhỏ hơn độ phân giải trước. Ở mỗi khối, các luồng tích chập

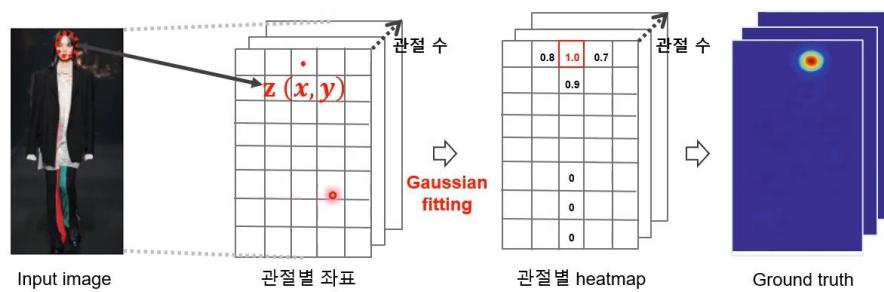
được thực hiện độc lập với nhau nhưng khi đến giai đoạn cuối một khối, mỗi luồng của khối trước sẽ tích chập với tất cả các luồng của khối sau, cơ chế này được gọi là tích chập đa độ phân giải.



Điều này nghĩa là HRNet sẽ kết nối các lớp tích chập của giai đoạn này với tất cả các lớp tích chập của giai đoạn tiếp theo, tạo ra sự khác biệt so với các phương pháp trước đó. Các phương pháp trước chủ yếu chỉ mở rộng mạng phân lớp bằng cách thêm những lớp tích chập để khôi phục độ phân giải, từ đó tái tạo lại ảnh gốc và sử dụng cơ chế phân lớp để nhận dạng khớp xương (phần a là mô hình mạng phân lớp, phần b là mô hình mạng khôi phục độ phân giải, hai phần được kết nối liên tiếp nhau).

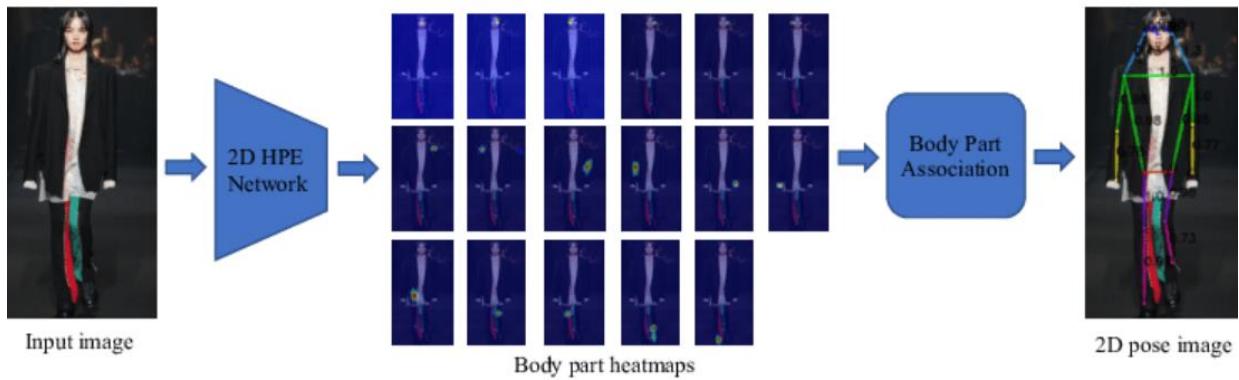


Sau khi xử lý qua các khối tích chập, heatmap được tạo ra bằng cách sử dụng bộ lọc Gaussian 2D với độ lệch chuẩn là 1-pixel vào tâm của điểm được dự đoán là có khớp xương trong ảnh kết quả. Áp dụng cho tất cả các điểm để thu được bản đồ heatmap cho từng khớp xương.



Heatmap regression method

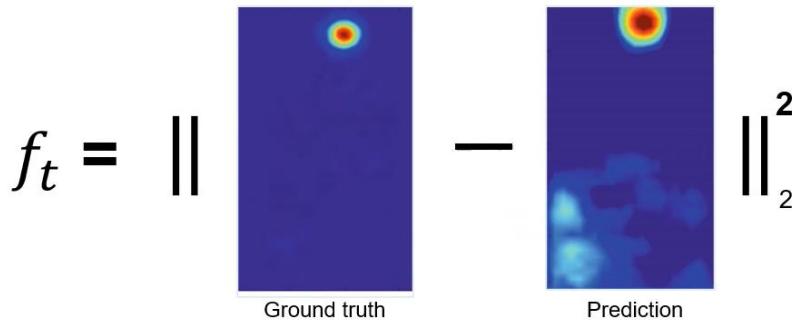
Nhờ cơ chế này mà HRNet có thể nhận dạng tốt hơn về mặt ngữ nghĩa do sử dụng các block gồm nhiều luồng có độ phân giải khác nhau, các luồng có độ phân giải thấp thì lượng thông tin rút ra được càng nhiều. Đồng thời HRNet vẫn giữ được độ chính xác về vị trí trên ảnh gốc do luôn duy trì ảnh có độ phân giải cao nhất mà không cần tái tạo lại từ ảnh có độ phân giải thấp. Điều này giúp HRNet đạt các kết quả tốt hơn nhiều so với những mạng trước đây như AlexNet, GoogleNet, VGGNet, ResNet, DenseNet, ...



(b) Body Part Detection Methods

Hàm lỗi được HRNet sử dụng là Mean Squared Error (MSE), là hàm lỗi đơn giản và vô cùng phổ biến trong lĩnh vực Machine Learning. MSE tính độ khác biệt của dự đoán và giá trị xác thực, sau đó bình phương kết quả lên để loại bỏ giá trị âm cũng như tăng sự chênh lệch của độ khác biệt và tính trung bình trên toàn bộ tập dữ liệu. Ở đây giá trị dự đoán và giá trị xác thực là các heatmap được tính toán từ HRNet và các heatmap được thực hiện thủ công. Công thức toán của MSE và cách nhìn trực quan của MSE trong HRNet được thể hiện qua ảnh sau.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$



Dưới đây là bảng kết quả chạy trên tập dữ liệu MPII, HRNet có kết quả (tỉ lệ nhận dạng đúng) cao trong đa số các trường hợp, hầu hết đều nằm trong top đầu so với nhiều phương pháp khác bao gồm cả SimpleBaseline.

Table 3. Performance comparisons on the MPII test set (PCKh@0.5).

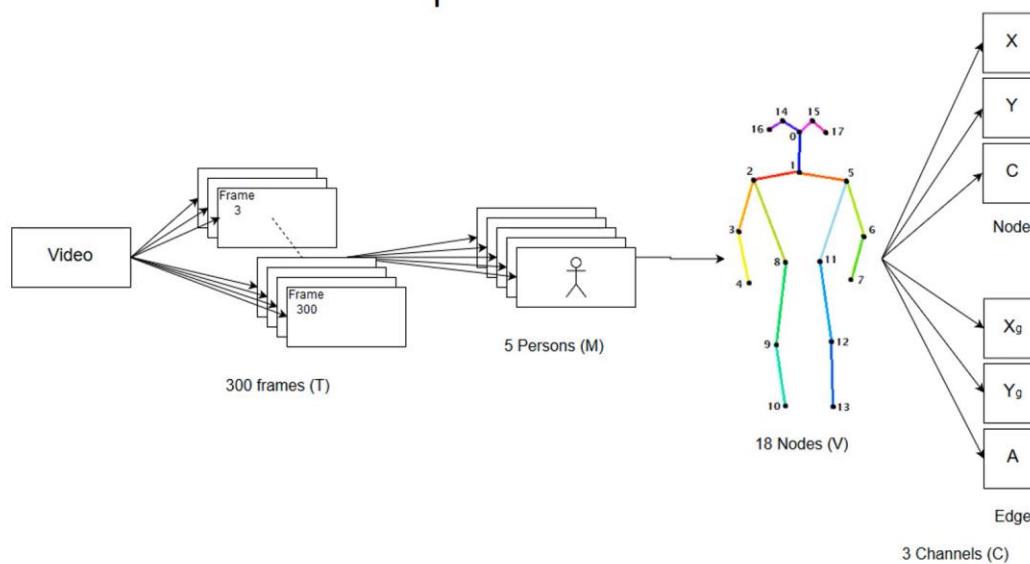
Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Total
Insafutdinov et al. [27]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Wei et al. [69]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat et al. [4]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Newell et al. [40]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Sun et al. [58]	98.1	96.2	91.2	87.2	89.8	87.4	84.1	91.0
Tang et al. [63]	97.4	96.4	92.1	87.7	90.2	87.7	84.3	91.2
Ning et al. [44]	98.1	96.3	92.2	87.8	90.6	87.6	82.7	91.2
Luvizon et al. [37]	98.1	96.6	92.0	87.5	90.6	88.0	82.7	91.2
Chu et al. [14]	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chou et al. [12]	98.2	96.8	92.2	88.0	91.3	89.1	84.9	91.8
Chen et al. [10]	98.1	96.5	92.5	88.5	90.2	89.6	86.0	91.9
Yang et al. [77]	98.5	96.7	92.5	88.7	91.1	88.6	86.0	92.0
Ke et al. [31]	98.5	96.8	92.7	88.4	90.6	89.3	86.3	92.1
Tang et al. [62]	98.4	96.9	92.6	88.7	91.8	89.4	86.2	92.3
SimpleBaseline [72]	98.5	96.6	91.9	87.6	91.1	88.1	84.1	91.5
HRNet-W32	98.6	96.9	92.8	89.0	91.5	89.0	85.7	92.3

ST-GCN:

ST-GCN (Spatial Temporal Graph Convolution Network): input đầu vào là khung xương của người, chính xác là 18 khớp. Mỗi khớp mang thông tin (c, X, Y) với X, Y mang thông tin tọa độ trên ảnh, c mang thông tin về confidence score từ ảnh heatmap. Và dãy khung xương được rút trích từ 300 frame ảnh liên tiếp trong video. Cụ thể đầu vào sẽ có số chiều tensor như sau: (C, T, V, M) với C là thông tin của node (c, X, Y), T là số frame (300), V là số khớp 18, M là số người trong ảnh. Giả sử video có 2 người thì tensor đầu vào sẽ là (3, 300, 18, 2).

Output: sẽ là nhãn hành động cho 300 frame input video.

ST-GCN : Model inputs

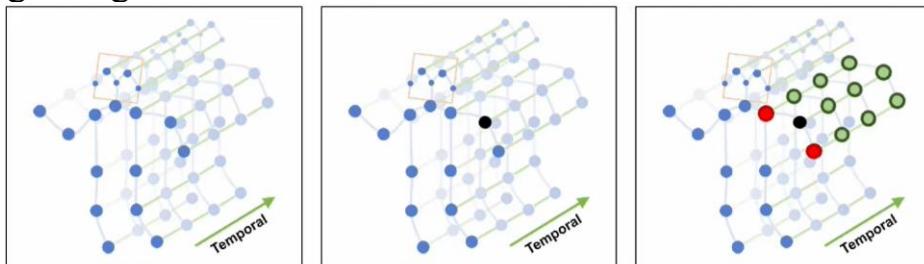


Đi sâu hơn vào phương pháp, ST GCN sẽ biến khung xương người thành đồ thị với các node chính là các khớp, các liên kết sẽ là các đường nối các khớp tự nhiên của con người, và cũng xét trên trục thời gian.

$$B(v_{ti}) = \{v_{qj} | d(v_{tj}, v_{ti}) \leq K, |q - t| \leq \lfloor \Gamma/2 \rfloor\}.$$

Trên đây là công thức để xác định tập hợp các lân cận của một node i tại thời điểm t . $d(v_{tj}, v_{ti})$ là khoảng cách ngắn nhất từ v_{tj} đến v_{ti} có với mỗi liên kết được xem có khoảng cách là 1. Γ là siêu tham số thể hiện

khoảng thời gian để chọn các node lân cận.



Ví dụ từ một node màu đen, các node v_{qj} nào thỏa $d(v_{tj}, v_{ti}) \leq 1$ và $|q - t| < \frac{\Gamma}{2}$ (lưu ý rằng không xét khoảng cách từ node xanh lá đến node đen mà xét node đen đến node đỏ sau đó xét nó có thỏa trong vùng thời gian cho phép không)

ST-GCN sẽ chia các lân cận ra làm các tập hợp riêng, và mỗi tập hợp riêng này sẽ có một ma trận kè và ma trận trọng số cho mỗi tập hợp, mỗi tập hợp sẽ có nhãn riêng và công thức đánh nhãn cho các tập hợp như sau:

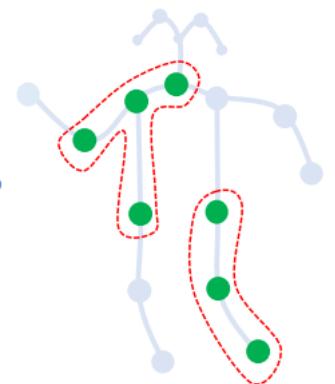
$$l_{ST}(v_{qj}) = l_{ti}(v_{tj}) + (q - t + \lfloor \Gamma/2 \rfloor) \times K.$$

Với v_{qj} là node thứ j tại thời gian q, $l_{ti}(v_{tj})$ sẽ xem xét nhãn tại thời gian t sau đó cộng thêm một đại lượng với K là số tập hợp con trong thời gian t.

Có 3 chiến lược để chia ra các tập hợp con (các chiến lược được minh họa dưới một frame, trong trường hợp multi frame thì đã có công thức phía trên để chia làm subsets).

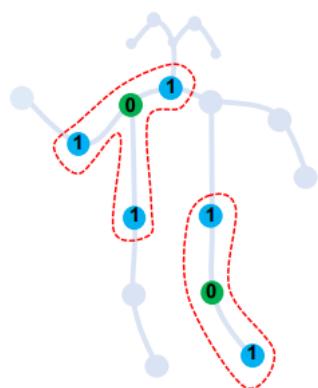
1. Uniform

Là cách đơn giản nhất các node lân cận và cả tại node đó sẽ được chia vào chung 1 tập hợp $K = 1$ và $l_{ti}(v_{ti}) = 0 \forall i, j \in V$. Tuy nhiên với chiến lược này, đồng nghĩa với chỉ tính trung bình cho từng node sau đó nhân với một vector trọng số, có thể những đặc trưng về các khớp cục bộ sẽ bị mất khi dùng chiến lược này.



2. Distance partitioning

Một cách chia tự nhiên khác là dựa vào khoảng cách với $K = 2$, những khoảng cách $d = 0$ hay chính là node gốc sẽ được chia riêng ra và $d = 1$, những node lân cận sẽ được chia riêng ra, và hệ quả là $l_{ti}(v_{tj}) = d(v_{tj}, v_{ti})$. Với chiến lược này, các đặc trưng tại 2 khớp sẽ được đánh trọng số khác nhau với qua quá trình học, giúp cho việc xem xét các đặc trưng cục bộ tốt hơn.

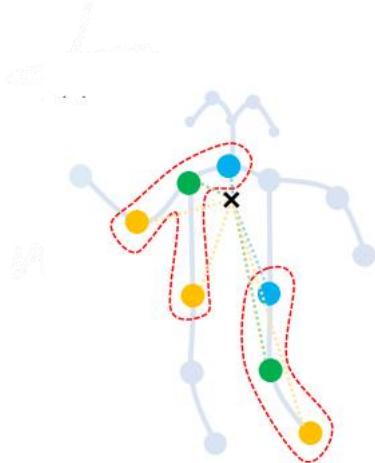


3. Spatial configuration partitioning

Tiếp theo là một cách chia các lân cận dựa vào cấu trúc cơ thể người. Cụ thể như sau: các lân cận sẽ được chia làm ba tập hợp, tập hợp đầu chính là node gốc, tập hợp thứ hai gọi là centripetal group những điểm sẽ gần với trọng tâm cơ thể hơn là điểm gốc, tập hợp thứ 3 là những điểm xa hơn gọi là centrifugal group.

$$l_{ti}(v_{tj}) = \begin{cases} 0 & \text{if } r_j = r_i \\ 1 & \text{if } r_j < r_i \\ 2 & \text{if } r_j > r_i \end{cases}$$

Với r là khoảng cách Euclidean đến trọng tâm cơ thể.



Việc chia thành 3 subsets với mong muốn nhận dạng được các hành động tốt hơn, mang lại hiệu năng cao hơn cho mô hình. (Ví dụ như hành động bắt tay phần xa trọng tâm cơ thể là tay sẽ có xu hướng cử động nhiều hơn, nên việc chia ra để đánh trọng số có vẻ hợp lí).

Sau khi đã hoàn chỉnh về việc chia ra thành các group, ta tiến hành vào quá trình học.

Từ định nghĩa: $H^{(l+1)} = \sigma(\Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}}H^{(l)}W^{(l)})$, ta sẽ biến đổi để phù hợp với mô hình có phân chia thành các group.

Tạm bỏ qua lớp kích hoạt ta có:

$$f_{out} = \Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}}f_{in}W$$

Giả sử ta theo phương pháp distance partitioning để chia nhóm, lúc này công thức trở thành:

$$\begin{aligned} f_{out} &= \sum_j \Lambda_j^{-\frac{1}{2}} A_j \Lambda_j^{-\frac{1}{2}} f_{in} W_j \\ &= \Lambda_0^{-\frac{1}{2}} A_0 \Lambda_0^{-\frac{1}{2}} f_{in} W_0 + \Lambda_1^{-\frac{1}{2}} A_1 \Lambda_1^{-\frac{1}{2}} f_{in} W_1 \end{aligned}$$

$$= If_{in}W_0 + \Lambda_1^{-\frac{1}{2}}A_1\Lambda_1^{-\frac{1}{2}}f_{in}W_1$$

Với $\Lambda_j^{ii} = \sum_k (A_j^{ik}) + \alpha$, $\alpha = 0.001$ để tránh trường hợp hàng rỗng. Có thể thấy rằng nếu áp dụng distance partitioning mỗi đặc trưng $A_0 = I$ vì lúc này sẽ chia ra 2 tập riêng biệt để áp các trọng số riêng vào, và $A_1 = A$. Tương tự với các trường hợp còn lại.

Learnable edge weight:

Việc đánh trọng số là 1 trên ma trận kè sẽ khiến cho một group sẽ gồm các khớp được xem là như nhau, tuy nhiên mỗi khớp sẽ mang cho mình một đóng góp khác nhau vào kết quả sẽ hợp lí hơn và mang tính khái quát hơn, nên tác giả đã đề xuất tạo ra ma trận M (NxN) để học về đánh trọng số vào các khớp trên cơ thể người. Việc này có thể giúp cải thiện và nâng cao hiệu suất học của mô hình.

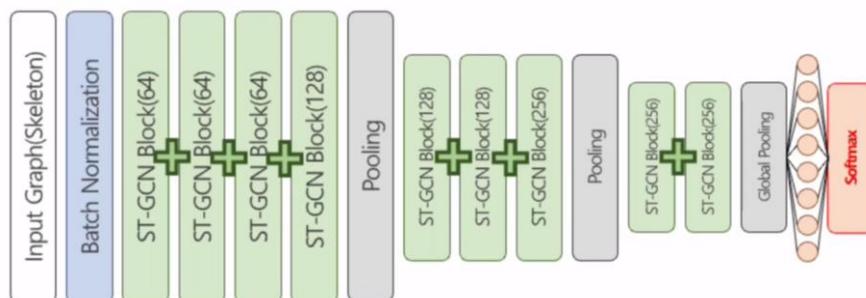
Ma trận M sẽ được khởi tạo với toàn bộ là 1 khi học, và khi áp dụng vào công thức GCN:

$$f_{out} = \sum_j \Lambda_j^{-\frac{1}{2}}(A_j \otimes M) \Lambda_j^{-\frac{1}{2}} f_{in} W_j$$

Với \otimes là toán tử nhân từng phần tử trong mảng (wise-element).

Network architecture:

Mô hình mạng được mô tả như sau:

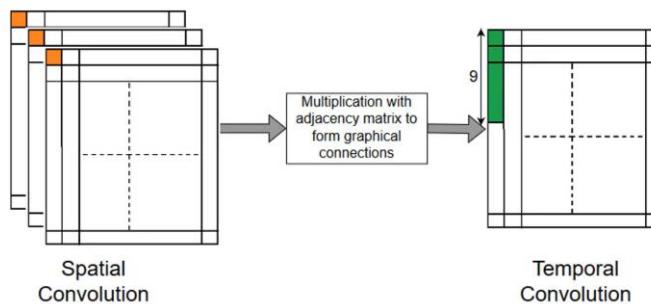


Với các khối ST-GCN giúp các rút trích các đặc trưng phù hợp nhất với mô hình do cơ chế tự học của mạng học, các khối pooling giúp giữ lại các đặc trưng quan trọng nhất để mang đi phân lớp nhãn hành động và sau cùng là một lớp softmax để phân lớp hành động.

Ngoài ra trong quá trình training để tránh tình trạng bị over fitting các frame ảnh được xoay, tịnh tiến, và scale để phù hợp với nhiều tình trạng hình ảnh. Thêm nữa là drop out với xác suất là 0.5 trên mỗi ST-GCN (là vô hiệu hóa đi một vài trọng số) điều này cũng có thể hiểu như là hành động bắt tay thì chỉ cần quan sát phần tay là có thể biết được, hoặc là ăn thì không cần quan sát phần chân, điều này cũng giúp tránh tình trạng overfitting.

Ở các khối ST-GCN:

ST-GCN : Convolution Operation



Number of input channels to temporal convolution is equal to the number of output channel from spatial convolution.

Như đã trình bày phía trên ma trận đặc trưng sẽ được xếp theo (C, T, V) số chiều đặc trưng, số frame và số khớp.

Các đặc trưng sẽ thực hiện phần $f_{in} W_j$ với các kernel $(1, 1)$, sau đó sẽ nhân với ma trận kè đã được chuẩn hóa $\Lambda_j^{-\frac{1}{2}} (A_j \otimes M) \Lambda_j^{-\frac{1}{2}}$. Sau đó dùng 1 kernel $(9, 1)$ để tích chập theo miền không gian.

```
x = self.conv(x)

n, kc, t, v = x.size()
x = x.view(n, self.kernel_size, kc//self.kernel_size, t, v)
x = torch.einsum('nkctv,kvw->nctw', (x, A))
```

Sau khi qua nhiều ST-GCN block đến cuối cùng sẽ qua softmax regression để phân lớp hành động cho chúng ta, với hàm lỗi mục tiêu là:

$$L_{CE} = - \sum_{i=1}^n t_i * \log(p_i)$$

Trong đó, n là tổng số lớp, t_i là nhãn đúng của lớp thứ i và p_i là xác suất dự đoán của lớp đó.

Dataset:

Kinetics: với 400 nhãn hành động của người



(a) headbanging



(b) stretching leg



(c) shaking hands



(e) robot dancing



(d) tickling



(f) salsa dancing

NTURGB+ D (2016): với 60 nhãn hành động cho người.



Result:

	Top-1	Top-5
Baseline TCN	20.3%	40.0%
Local Convolution	22.0%	43.2%
Uni-labeling	19.3%	37.4%
Distance partitioning*	23.9%	44.9%
Distance Partitioning	29.1%	51.3%
Spatial Configuration	29.9%	52.2%
ST-GCN + Imp.	30.7%	52.8%

Table 1: Ablation study on the Kinetics dataset. The “ST-GCN+Imp.” is used in comparison with other state-of-the-art methods. For meaning of each setting please refer to Sec.4.2.

Kết quả thực nghiệm cũng cho thấy khi áp dụng càng về sau thì độ chính xác càng tăng trên tập thử nghiệm Kinetics, và mô hình dùng để thực chiến và mang lại kết quả tốt nhất là ST-GCN + Imp: khi áp dụng chiến lược chia nhóm theo (spatail configuration partitioning) và áp dụng thêm learnable edge weight.

	X-Sub	X-View
Lie Group (Veeriah, Zhuang, and Qi 2015)	50.1%	52.8%
H-RNN (Du, Wang, and Wang 2015)	59.1%	64.0%
Deep LSTM (Shahroudy et al. 2016)	60.7%	67.3%
PA-LSTM (Shahroudy et al. 2016)	62.9%	70.3%
ST-LSTM+TS (Liu et al. 2016)	69.2%	77.7%
Temporal Conv (Kim and Reiter 2017).	74.3%	83.1%
C-CNN + MTLN (Ke et al. 2017)	79.6%	84.8%
ST-GCN	81.5%	88.3%

Kết quả trên tập dataset NTU-RGB+D cho thấy vượt bậc so với các mô hình khác.

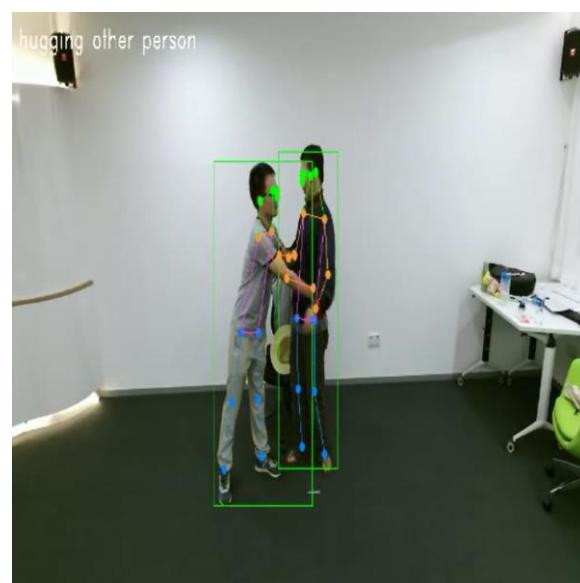
Đặt biệt hơn ở mô hình ST-GCN++ khi có một số thay đổi nhỏ là bỏ đi ma trận kè và thay bằng một ma trận có khả năng học, và từ single brach ở tích chập miền thời gian chuyển sang multi-brach đã mang đến

sự tiên bộ rõ rệt cho mô hình.

	NTURGB+D		NTURGB+D 120	
Model	XSub	XView	XSub	XSet
CTR-GCN	92.4	96.8	88.9	90.6
ST-GCN++	92.6	97.4	88.6	90.8

Demo:

Demo sẽ sử dụng bộ tham số đã train sẵn ở mô hình Faster RCNN để phát hiện người, HRnet để rút trích khung xương và ST-GCN để nhận dạng hành động.



Có thể thấy các phần tay bị ẩn đi có thể được phát hiện thông qua HRnet giúp cho việc nhận dạng hành động trở nên đơn giản.

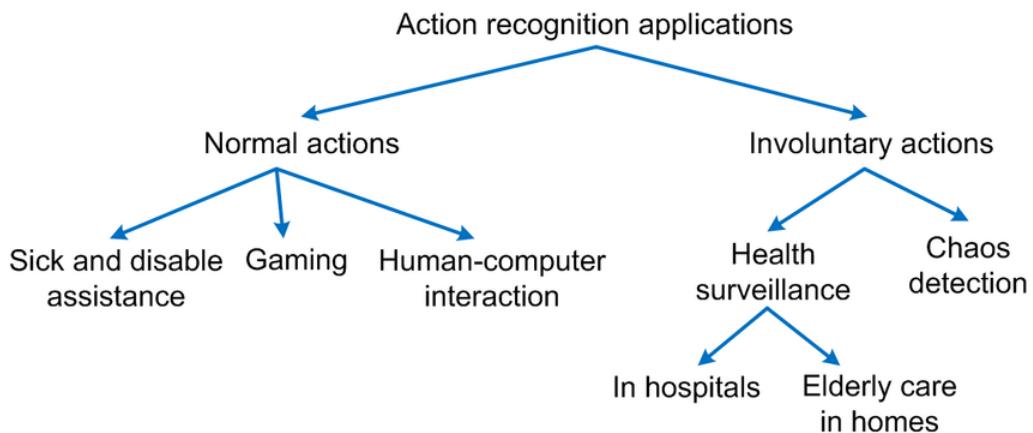
V. APPLICATION

- Trong lĩnh vực y tế và chăm sóc sức khoẻ: việc nhận dạng sớm những hành vi té ngã, tai nạn của bệnh nhân giúp người nhà và bác sĩ có thể phản ứng kịp thời hơn, tăng cơ hội cứu chữa cho những trường hợp khẩn cấp (đột quy, nhồi máu cơ tim, ...).
- Trong lĩnh vực an ninh: việc nhận dạng được những hành vi gây nguy hiểm như bắn súng, đánh nhau, ... giúp cơ quan an

ninh có phản ứng ngăn chặn kịp thời, hạn chế tối đa thiệt hại có thể xảy ra.



- Trong lĩnh vực tương tác người – máy: tăng mức độ hiểu biết của máy tính về con người, giúp con người có thể thực hiện nhiều hành vi giao tiếp trực tiếp với máy tính hơn như là điều khiển bằng cử chỉ thay vì chuột và bàn phím như hiện tại.
Ví dụ: [Điều khiển âm lượng bằng cử động tay – Youtube](#).
- Ngoài ra, nhận diện hành vi còn một số ứng dụng khác, thể hiện qua sơ đồ bên dưới.



VI. REFERENCES

- [HRNet – Paperswithcode.](#)
- [OpenPose – Paperswithcode.](#)
- [DD-Net – Paperwithcode.](#)
- [SGN-Net – Paperswithcode.](#)
- [STTFormer – Paperswithcode.](#)

Hết
