Fitness Activity Recognition

Demo 1
4/29/2022

Brady Hong, Alqama Sams, Ahmed Ceif, Alex Lidiak, Devansh Sharma, Sarthak Gupta, Raghav Kachroo
headed by Mike Chung

Activity Recognition Overview

Human Activity Recognition (HAR) is a 3 step process[1]:

1. (**Detection**): Video frame segmentation and body identification
2. (**Tracking**): Action representation with respect to posture and motion
3. (**Classification**): Learning process that recognizes actions

In current setup of last cohort, MoveNet performs (1,2) by segmenting and generating skeleton keypoints and their coordinates for each frame in the video (recorded in Coordinates.csv). Step (3) is performed via feature engineering and subsequent XGBoost Classification with 5 frame Accuracy: 95% and 1 second accuracy: 88%.

Improvements to the algorithm can be made/tested by improving steps according to metrics such as:

Steps 1-2 (for Skeleton-based models)**:** Percentage of Detected Joints (PDJ), Percentage of Correct Keypoints (PCK)

Step 3: Confusion Matrix and metrics derived from it (AUC, Precision, etc)

[1] Beddiar, D. R., Nini, B., Sabokrou, M., & Hadid, A. (2020). Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, *79*(41), 30509-30555.
[2] Zheng, C., Wu, W., Yang, T., Zhu, S., Chen, C., Liu, R., ... & Shah, M. (2020). Deep learning-based human pose estimation: A survey. *arXiv preprint arXiv:2012.13392*.

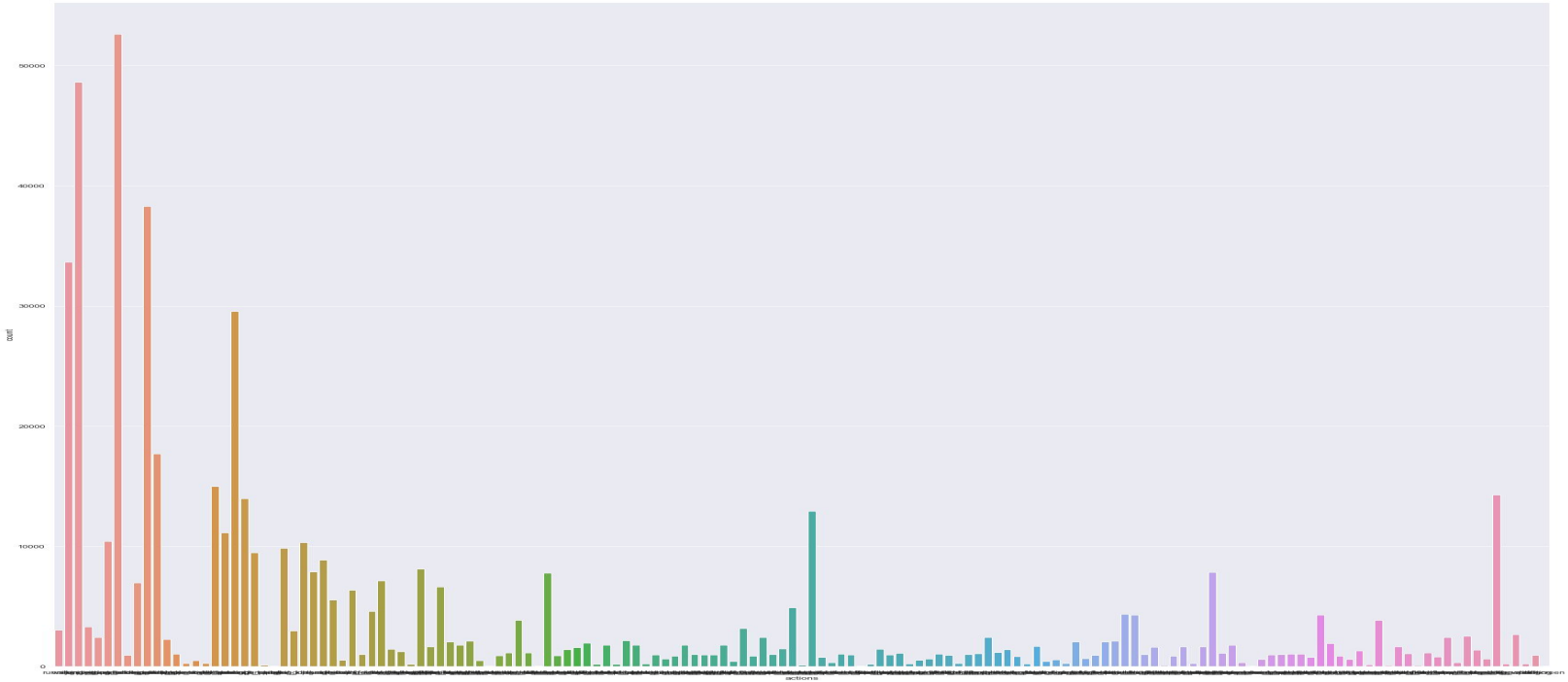
Dataset - coordinates.csv

	video_id	frame_count	fps	0	1	2	3	4	5	6	...	31	32	33	actions
0	279	1	29	0.194167	0.474421	0.177525	0.486390	0.177479	0.467132	0.190676	...	0.464268	0.773160	0.460820	running
1	279	2	29	0.194559	0.473831	0.179554	0.483210	0.179706	0.468016	0.191008	...	0.468995	0.771903	0.470143	running
2	279	3	29	0.196080	0.468627	0.181912	0.478633	0.180850	0.463631	0.193267	...	0.492166	0.774354	0.489078	running
3	279	4	29	0.196092	0.468655	0.181905	0.478662	0.180863	0.463647	0.193274	...	0.492249	0.774339	0.489007	running
4	279	5	29	0.197715	0.467405	0.184015	0.475692	0.181017	0.464770	0.194321	...	0.512477	0.772853	0.513822	running
5	279	6	29	0.193251	0.466999	0.180784	0.474405	0.178592	0.464480	0.190587	...	0.544553	0.773895	0.547832	running
6	279	7	29	0.177526	0.470639	0.162743	0.480819	0.162704	0.464099	0.172374	...	0.481188	0.599147	0.497471	running
7	279	8	29	0.171354	0.468604	0.158290	0.478144	0.157476	0.463318	0.169546	...	0.489421	0.673070	0.547264	running
8	279	9	29	0.169871	0.466745	0.156035	0.477650	0.155124	0.463267	0.167496	...	0.504811	0.721864	0.592635	running
9	279	10	29	0.169094	0.466090	0.155799	0.476993	0.154233	0.462399	0.167942	...	0.504984	0.722581	0.592423	running

- Each number (0-33) on the column represents 17 keypoint joints for the X and Y axis total of 34 for each frame
- Fps tells how many frames per second for the video
- 153 actions total

```
yoga,stretching                  52613
standing                         48603
talking,walking                  38290
walking                          33645
boxing                           29559
                                   ...
lay_down_one_knee_up                30
adjusting                           30
bear_hold_knee_taps                 30
torso_up_double_leg_extension       30
offscreen                           29
Name: actions, Length: 153, dtype: int64
```

EDA of Coordinate.csv



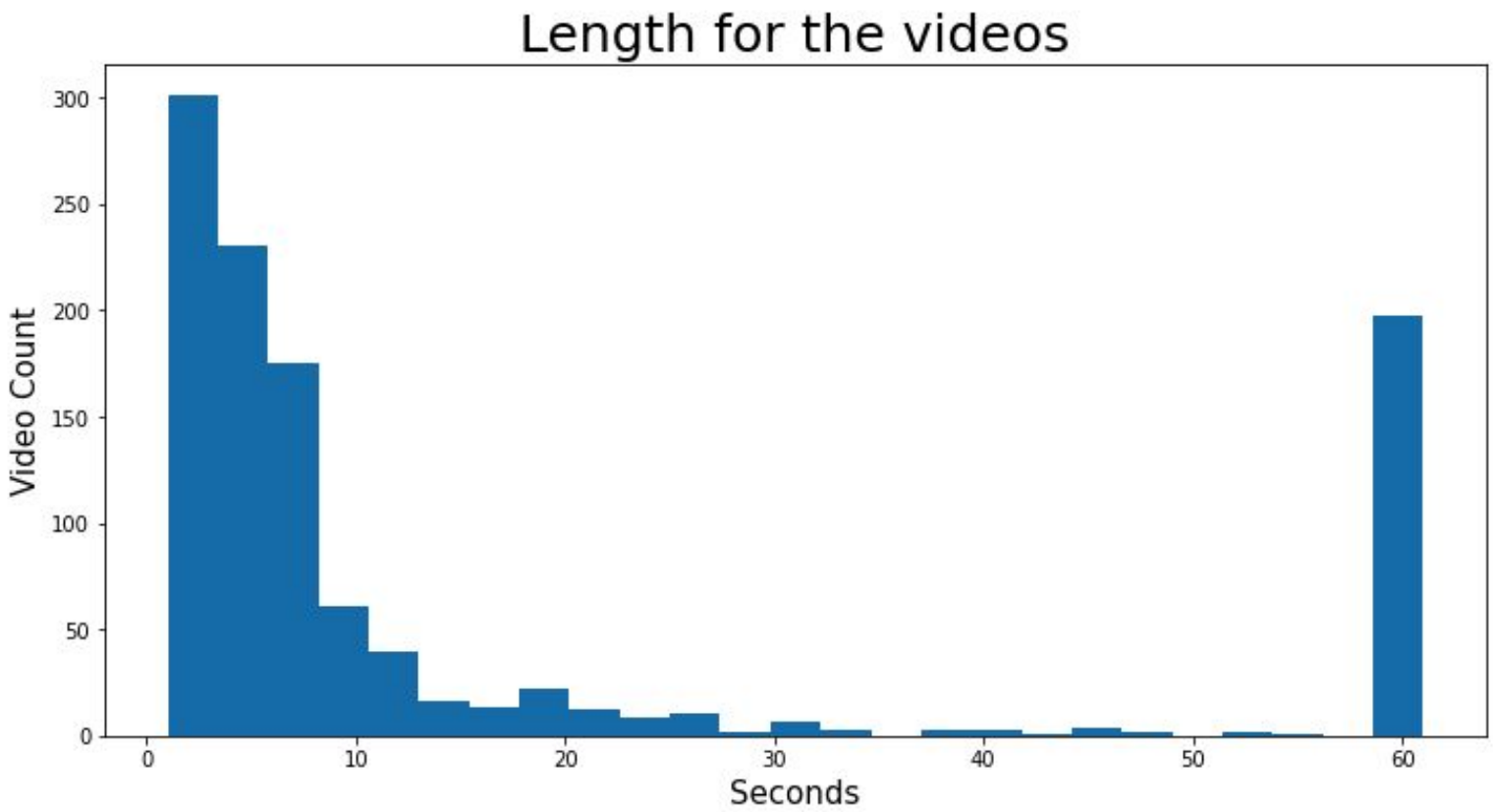
Dataset - annotated.csv

	name	video_id	Length	0	1	2	3	4	5	6	...
0	3869387-preview	279	30	running	running	running	running	running	running	running	...
1	6669347-preview	404	20	walking	walking	walking	walking	walking	walking	walking	...
2	6669362-preview	777	20	walking	walking	walking	walking	walking	walking	walking	...
3	25340660-preview	823	20	walking	walking	walking	walking	walking	walking	walking	...
4	25340660-preview	823	15	running	running	running	running	running	running	running	...
5	1083655534-preview	470	12	walking	walking	walking	walking	walking	walking	walking	...
6	1083832744-preview	576	12	running	running	running	running	running	running	running	...
7	Pexels Videos 2785536	150	25	standing	standing	standing	jump_rope	jump_rope	jump_rope	jump_rope	...
8	pexels-cottonbro-5836750	108	8	sitting	sitting	sitting	sitting	sitting	sitting	sitting	...
9	pexels-cottonbro-6551816	877	31	sitting	sitting	sitting	sitting	sitting	sitting	sitting	...

- Numbers (0-60) on the column represent the each second of the video
- Classified status for each second of the video
- Length = second

```
annotated.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1114 entries, 0 to 1113
Data columns (total 64 columns):
 #    Column      Non-Null Count  Dtype
---   ------      --------------  -----
 0    name        1114 non-null   object
 1    video_id    1114 non-null   int64
 2    Length      1114 non-null   int64
 3    0           1114 non-null   object
 4    1           1112 non-null   object
 5    2           1050 non-null   object
 6    3           813 non-null    object
 7    4           682 non-null    object
 8    5           583 non-null    object
 9    6           503 non-null    object
 10   7           443 non-null    object
 11   8           408 non-null    object
 12   9           372 non-null    object
 13   10          347 non-null    object
 14   11          325 non-null    object
 15   12          307 non-null    object
```


Length for the videos
Video Count
300
250
200
150
100
50
0
0
10
20
30
40
50
60
Seconds

FUTURE STEPS

Link 1 - Interesting approach to multi-person pose estimator. Code seems similar to ours, but apparent drawback seems threshold for joints, where in a situation a person may not show up as 34 (17*2) joint movements in case the threshold values are not set properly.

Link 2 - Pertaining to 3D pose estimation - "However, obtaining accurate manual 3D annotation requires either a lab setup or specialised hardware with depth sensors for 3D scans, which introduces additional challenges to preserve a good level of human and ecological diversity in the dataset." - This presents a higher possibility of options for us to explore, but of course performing annotation in 3D brings along with it an incredibly laborious annotation process.

Link 3 - "Ryan Eder, founder and CEO at IncludeHealth, said the MoveNet model enhanced the speed and accuracy in delivering prescriptive care. "While other models trade one for the other, this unique balance has unlocked the next-generations of care delivery," said Eder." - Depending on our constraints, we can look into slower but more accurate models, and perform some augmentations to speed those models up.

"Ensemble Sequence to Sequence model." This could perhaps be a path we take to tackle this situation.

https://www.gameassetdeals.com/asset/201269/movenet-3d-realtime-3d-pose-tracking

Actions feature in coordinated csv

It tells how many vids performing which type of task as there were so many attributes which made difficult for me to write which one is which

- But around 3k are running videos
- Around 6k are standing videos
- Around 7k are walking videos
- 6k videos are on yoga

Conclusion: most the the frames consists of walking action as we can see in the above graph

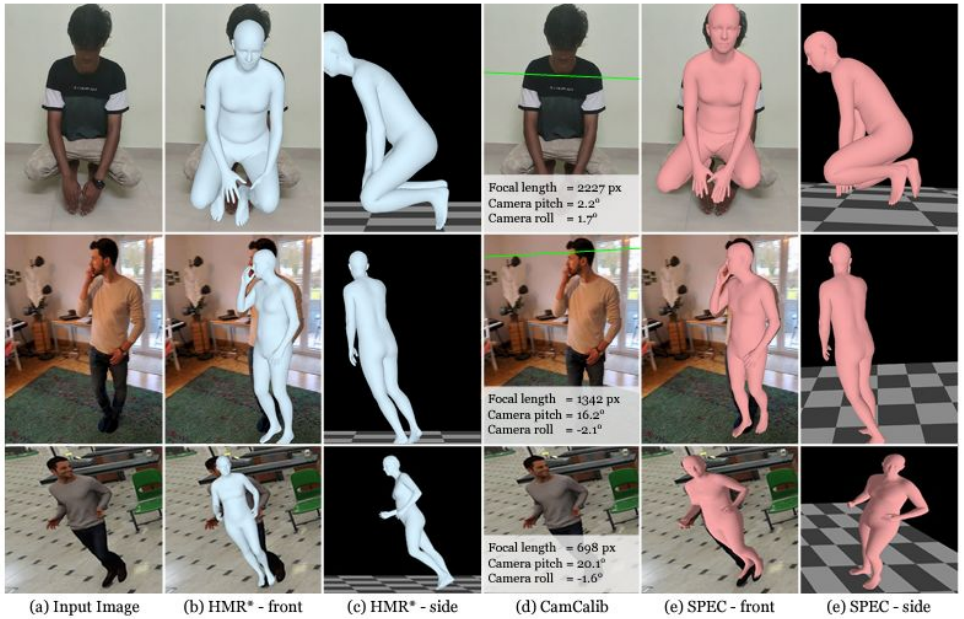
SPEC: Seeing People in the Wild with an Estimated Camera

What is it?

- Images are input
- Estimates the perspective camera
- Reconstruct 3-d bodies
- Front & Side

How it can help?

- Can allow us dive deeper into specific poses
- Allowing to make a more complex model



Paper Link: https://arxiv.org/abs/2110.00620

Ideas to look upon: Calculating the amount of period an action is repeated

1) Image processing + Signal processing:

- A naive method to analyze the positions of moving pixels using signal processing and count the peaks for activities like pushups.
- This method suffers from wrong counting when the user doesn't do the activity.
- The counting result is also affected by environmental factors such as other moving objects or changing light conditions.

2) Google RepNet:

- It is a state of the art method of general counter, where we can feed a video stream in and receive the counting.
- This approach can be used to count multiple activities with the network.
- Does not works well when the period of the activity is unstable.
- This architecture also requires a huge amount of computation, which is not suitable for running in realtime on weak desktop PCs or mobile devices.