Software Impacts 13 (2022) 100305

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original Software Publication

PyHAPT: A Python-based Human Activity Pose Tracking data processing framework

Hao Quan *, Andrea Bonarini

Department of Electronics, Information and Bioengineering, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano 20133, Italy

ARTICLE INFO

Keywords:
Data processing
Data visualization
Human activity recognition
Deep learning
Robotics

ABSTRACT

We propose a novel Python-based Human Activity Pose Tracking data processing framework (PyHAPT). It provides the functionality to efficiently process annotated human pose tracking raw video data collected in unconstrained environments. Besides, PyHAPT provides the functionalities of interpolation to recover the missing joints data and data visualization that gives insights into the spatial–temporal skeletal information. The processed data could be readily used for developing new human activity recognition deep learning models, which could be deployed on mobile service robots.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-40
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/1086569/tree/v1
Legal Code License	GPL-3.0-or-later
Code versioning system used	Git
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	OS: MS Windows or Linux; Python libraries: Sklearn, Pandas, Numpy, Matplotlib
If available Link to developer documentation/manual	https://github.com/AIRLab-POLIMI/PyHAPT/blob/main/README.md
Support email for questions	hao.quan@polimi.it, andrea.bonarini@polimi.it

1. Introduction

With its rapid development, Human Activity Recognition (HAR) has gained much attraction in the construction of mobile robotic application systems. Human activity recognition involves skeleton representations of human bodies instead of raw RGB videos. Due to its strong adaptability and highly abstract characteristics, many significant models were developed based on skeletal data [1–7]. Compared to the RGB video representation, the greatest benefits of the skeletal data are that they are free of dynamic environment noise and robust against complicated backgrounds (lighting conditions, color of clothing, object obstruction, etc.). It is important for service robots to recognize the actions of people in the real world to further enhance their capabilities to offer services.

We introduce PyHAPT, a Python-based Human Activity Pose Tracking data processing framework. It provides the functionality to automatically process annotated human pose tracking raw video data collected in unconstrained environments such that the pre-processed data could be directly used for developing new human activity recognition deep learning models, which could be deployed on the mobile service robots. Since PyHAPT is devoted to datasets collected in unconstrained environments, it also offers the functionalities of recovery of missing data and dynamic pose visualization to facilitate researchers to analyze the data deeply.

2. Related works

Some datasets collected from constrained environments have the skeleton data ready to use (e.g., NTU-RGB+D [8,9], PKU-MMD [10],

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: hao.quan@polimi.it (H. Quan), andrea.bonarini@polimi.it (A. Bonarini).

https://doi.org/10.1016/j.simpa.2022.100305
Received 30 March 2022; Received in revised form 25 April 2022; Accepted 28 April 2022

Table 1
Datasets used for recent human recognition models.

Model	Publisher	NTU 60 [8]	NTU 120 [9]	Kinetics-Skeleton [13,27]	NUCLA [24]	SYSU [25]
Efficient GCN [28]	TPAMI 22	✓	✓			
CTR-GCN [3]	ICCV 21	✓	✓		✓	
MST-GCN [29]	AAAI 21	✓	✓	✓		
SGN [5]	CVPR 20	✓	✓			✓
MSG3D [3]	CVPR 20	✓	✓	✓		
4S-Shift-GCN [30]	CVPR 20	✓	✓		✓	
Dynamic-GCN [31]	ACMMM 20	✓	✓	✓		
PA-ResGCN-B19 [32]	ACMMM 20	✓	✓			
DC-GCN+ADG [33]	ECCV 20	✓	✓		✓	
NAS-GCN [4]	AAAI 20	✓		✓		
PL-GCN [34]	AAAI 20	✓				✓
2S-AGCN [1]	CVPR 19	✓		✓		
DGNN [35]	CVPR 19	✓		✓		
AGC-LSTM [36]	CVPR 19	✓			✓	
AS-GCN [37]	CVPR 19	✓		✓		
ST-GCN [26]	AAAI 18	✓		✓		
Datasets used times		16	9	8	4	2

ETRI [11]), since they adopted depth cameras like the *Microsoft Kinect* to collect the data. However, the other datasets (e.g., ActivityNet [12], Kinetics [13], AVA [14], Charades [15], FineGym [16]) collected by crowd-sourcing methods from unconstrained environments do not directly provide the skeletal data. It requires researchers to employ other pose estimation methods (e.g., OpenPose [17], OpenPifPaf [18], MMPose [19], VIBE [20]) to extract and pre-process the skeletal representation such that the pre-processed skeletal data could be ready for training and evaluating the deep learning recognition models. Gupta et al. [21] made an effort to organize a couple of skeletal datasets obtained from other public datasets [22,23] that were still collected from constrained environments and crowd-sourcing methods rather than real public spaces (*In The Wild–ITW*), except that they do not provide the software to elaborate these data.

We analyzed some relevant skeleton-based HAR models since 2018 to check how public datasets were used to train and evaluate models in the community. As shown in Table 1, the most commonly used datasets are (in descending order): NTU RGB+D 60 [8], NTU RGB+D 120 [9], Kinetics [13], Northwestern-UCLA Multiview Action 3D [24], SYSU 3D Human-Object Interaction [25] datasets. Among those, only Kinetics was not collected from a constrained environment but from online streaming resources by using the crowd-sourcing method instead, while all the other datasets were collected in the respective laboratories.

Yan et al. [26] firstly extracted and pre-processed the skeletal data of the unique unconstrained Kinetics dataset using Python by OpenPose [17], namely Skeleton-Kinetics [27]. Then, most of the robust HAR models used Skeleton-Kinetics to evaluate the performance of their models on Skeleton-Kinetics [1,3,7]. Nevertheless, as the authors discussed, Skeleton-Kinetics takes only two persons into account in a clip at the most, ignoring other persons in the backgrounds with relatively lower average confidence score(s) provided by OpenPose. Meanwhile, they had to spend a lot of time to identify and track human pose data based on the skeletal data generated by OpenPose. We argue that not only two persons with the highest confident score, but also other persons in the background of a clip should be included in the pre-processed data. Moreover, a more efficient method to track and pre-process raw skeletal data is needed.

3. PyHAPT positive impacts

From Table 1, we could also see that the state-of-the-art models were not evaluated sufficiently on other datasets collected from unconstrained environments, i.e., datasets collected by crowd-sourcing methods and in public spaces. Although models developed based on crowd-sourcing collected datasets are much more reliable than those based on datasets collected in constrained environments, we believe that the *ITW* datasets could further improve the reliability of the models.

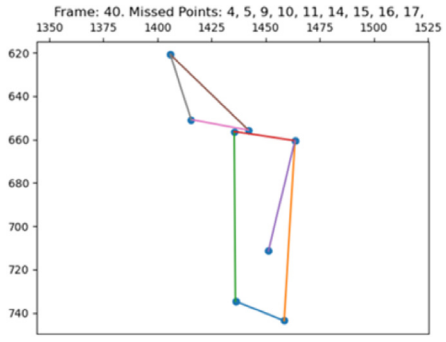

Fig. 1. An example of an invalid pose with eight joints (the threshold of valid joints set as nine).

We suppose that some reasons hinder the community from using datasets collected in unconstrained environments. Firstly, the publicly available datasets collected by crowd-sourcing methods in unconstrained environments usually do not provide human pose tracking data. Secondly, people's actions are continuous and sequential in real life, lasting at least a few seconds instead of single frames. Multiple persons or crowded scenes in a frame are often present in public spaces. The frame rate of RGB cameras in the current market is usually about 15 fps ∼ 30 fps. An enormous workload would be needed if researchers want to extract and pre-process the skeletal data from those datasets so that these data could be used for experiments. Thirdly, there is a lack of an efficient software framework for large-scale human activity recognition video datasets to track human pose automatically and pre-process raw, tracked skeletal data, making the pre-processed data ready for experiments.

Our proposed novel software framework PyHAPT could fill this gap. Researchers do not need to track the human pose, the raw skeletal data extracted from videos already include the tracking information. Then, the software could pre-process the tracked skeletal data automatically and make it available for training and evaluating deep learning models.

The framework of PyHAPT is divided into three parts. The first part uses OpenPifPaf [18] to detect and track the pose of people during the whole clip and store the data in JSON format to files. The second part is to use HAVPTAT [38][1] to associate action labels to people in videos based on the raw pose track data generated by OpenPifPaf and store back the annotated data to the file system. The last part employs the Python script of PyHAVPT to elaborate the annotated JSON format automatically. The elaborated pose tracking data is stored

[1] https://github.com/AIRLab-POLIMI/HAVPTAT_annotation_tool.

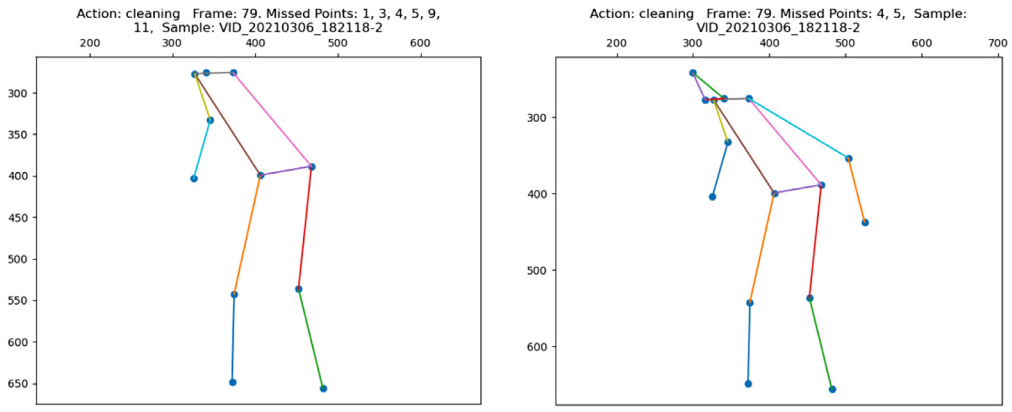

Fig. 2. Original pose (left); Pose reconstructed by "interpolation" (right).

in NPY format, and the corresponding action labels are stored in PKL format on file. Both data formats are widely used in the community and compatible with the major part of the state-of-the-art human activity recognition models.

The user could personalize the input parameters of the program: the path of the folder of the annotated JSON format raw data to be elaborated; the path of the file of the user-defined labels corresponding with the labels defined in the annotated JSON format raw data; the number of joints of a human body; the path of the folder where the generated data will be stored.

OpenPifPaf provides 2D coordinates (X, Y) in the pixel coordinate system for the 17 human joints following the definition of COCO body 17 keypoints [39,40]. We thus represent each joint with a tuple of pairs (X, Y) so that a skeleton frame is recorded as an array of 17 tuples. For the multi-person cases, we take all the detected persons in each clip into account instead of selecting only two people compared to Skeleton-Kinetics. By this way, we could treat each person's tracking data as an independent tuple which overcomes the limitation of considering at most two people in a frame. We consider each person performing the same action in a single video clip as a valid action sequence. If the same person performs multiple actions in a single video clip, we consider them as different action sequences performed by the same person. We consider an action performed by a person in a clip with tensor of (2, T, 17, 1) dimensions. For the whole dataset, the script gets the tensor of (N, 2, T, 17, 1) dimensions by concatenating the single action sequences of persons with N action samples. We summarize the meaning of each element in the tuple: the script generates N samples in total; two dimensions (X, Y) skeletal data; an action sequence lasts T frames; 17 keypoints of a human body; 1 person data in each tuple. All action skeleton sequences are padded to $T = 300$ frames by replaying the actions as done also by other skeletal datasets. The training set and test set split ratio is 70% and 30%. The number of padded frames and the training-test set split ratio can be both customized by users.

The skeletons of the data collected from the real world are often incomplete. A frame containing few joints could lead to ambiguity for the learning model. To reduce such a type of learning error, PyHAPT considers the pose as valid only when it has more than a given number of joints (nine) and offers the functionality to temporal-linearly interpolate the missing joints by the Python's Pandas library [41]. The threshold was fixed to nine to ignore only partial bodies could be captured by a camera in some cases.

PyHAPT provides the spatial–temporal data visualization script. We use PyHAPT to generate some pictures based on the samples of *POLIMI-ITW-S* datasets [42] to make some examples.

As shown in Fig. 1, when the camera is close to a person, only the upper part of the body is present, but the keypoints of the lower part of the body (14 left knee, 15 right knee, 16 left ankle, and 17 right ankle) miss. Given the lake of the joints of the lower part of the body, it is difficult for a model to predict the action among standing, walking, sitting or running. The setting of the threshold aims to reduce such a type of ambiguities which may decrease the accuracy of the recognition models. Users could also personalize the threshold to meet their needs.

Fig. 2 illustrates an example of the functionality of the interpolation offered by PyHAPT. The left picture is the original pose extracted by OpenPifPaf [18]. We regard it as a valid pose based since it has more than nine joints. We can see that Nose (1), Right eye (3), Right elbow (9), Right wrist (11) were not detected by OpenPifPaf. After having been processed by the interpolation operation provided by the Python's Pandas library [41], the three missing keypoints were reconstructed on the right picture.

4. Easy to use & lightweight

PyHAPT is based on Python. Python has become the most popular tool to use in the deep learning field. The framework could be executed both on Linux and MS Windows operation systems. Hence, we think our framework is easy to use for researchers. Because the script is lightweight, it could process and generate data efficiently. The formats of the generated data are also common used. The framework could be useful to accelerate the data processing work and advance the development of HAR models.

5. Use case

We have used PyHAPT to process a large-scale *POLIMI-ITW-S* video dataset for human activity recognition [42].

6. Future work

We would like to integrate some more data augmentation functionalities (e.g., horizontal flipping, scaling and translation) and train a reliable human activity recognition deep learning model to make PyHAPT framework a multi-functional deep learning framework.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by a research grant from the China Scholarship Council.

References

[1] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12026–12035.

[2] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng, View adaptive neural networks for high performance skeleton-based human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1963–1978.

[3] Z. Liu, H. Zhang, Z. Chen, Z. Wang, W. Ouyang, Disentangling and unifying graph convolutions for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 143–152.

[4] W. Peng, X. Hong, H. Chen, G. Zhao, Learning graph convolutional network for skeleton-based human action recognition by neural searching, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 2669–2676.

[5] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, N. Zheng, Semantics-guided neural networks for efficient skeleton-based human action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1112–1121.

[6] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[7] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, W. Hu, Channel-wise topology refinement graph convolution for skeleton-based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13359–13368.

[8] A. Shahroudy, J. Liu, T. Ng, G. Wang, NTU RGB+D: a large scale dataset for 3D human activity analysis, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 1010–1019, http://dx.doi.org/10.1109/CVPR.2016.115.

[9] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, A.C. Kot, NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding, IEEE Trans. Pattern Anal. Mach. Intell. (2019) http://dx.doi.org/10.1109/TPAMI.2019.2916873.

[10] C. Liu, Y. Hu, Y. Li, S. Song, J. Liu, PKU-MMD: A large scale benchmark for skeleton-based human action understanding, in: Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities, in: VSCC '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1–8, http://dx.doi.org/10.1145/3132734.3132739.

[11] J. Jang, D. Kim, C. Park, M. Jang, J. Lee, J. Kim, ETRI-activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 10990–10997.

[12] F. Caba Heilbron, V. Escorcia, B. Ghanem, J. Carlos Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in: Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition, 2015, pp. 961–970.

[13] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., The kinetics human action video dataset, 2017, arXiv preprint arXiv:1705.06950.

[14] C. Gu, C. Sun, D. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, J. Malik, AVA: A video dataset of spatio-temporally localized atomic visual actions, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6047–6056, http://dx.doi.org/10.1109/CVPR.2018.00633.

[15] G.A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, A. Gupta, Hollywood in homes: Crowdsourcing data collection for activity understanding, in: European Conference on Computer Vision, Springer, 2016, pp. 510–526.

[16] D. Shao, Y. Zhao, B. Dai, D. Lin, Finegym: A hierarchical video dataset for fine-grained action understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2616–2625.

[17] Z. Cao, T. Simon, S. Wei, Y. Sheikh, Realtime multi-person 2D pose estimation using part affinity fields, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 1302–1310, http://dx.doi.org/10.1109/CVPR.2017.143.

[18] S. Kreiss, L. Bertoni, A. Alahi, OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association, 2021, arXiv preprint arXiv:2103.02440.

[19] M. Contributors, OpenMMLab pose estimation toolbox and benchmark, 2020, https://github.com/open-mmlab/mmpose.

[20] M. Kocabas, N. Athanasiou, M.J. Black, Vibe: Video inference for human body pose and shape estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5253–5263.

[21] P. Gupta, A. Thatipelli, A. Aggarwal, S. Maheshwari, N. Trivedi, S. Das, R.K. Sarvadevabhatla, Quo vadis, skeleton action recognition? Int. J. Comput. Vis. 129 (7) (2021) 2097–2112.

[22] J. Carreira, E. Noland, C. Hillier, A. Zisserman, A short note on the kinetics-700 human action dataset, 2019, arXiv preprint arXiv:1907.06987.

[23] P. Weinzaepfel, G. Rogez, Mimetics: Towards understanding human actions out of context, Int. J. Comput. Vis. 129 (5) (2021) 1675–1690.

[24] J. Wang, X. Nie, Y. Xia, Y. Wu, S. Zhu, Cross-view action modeling, learning, and recognition, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2649–2656, http://dx.doi.org/10.1109/CVPR.2014.339.

[25] J. Hu, W. Zheng, J. Lai, J. Zhang, Jointly learning heterogeneous features for RGB-D activity recognition, IEEE Trans. Pattern Anal. Mach. Intell. 39 (11) (2017) 2186–2200.

[26] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: S.A. McIlraith, K.Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 7444–7452.

[27] S. Yan, Y. Xiong, D. Lin, Skeleton-kinetics, 2018, Available online at: https://github.com/yysijie/st-gcn/blob/master/OLD_README.md#kinetics-skeleton.

[28] Y.-F. Song, Z. Zhang, C. Shan, L. Wang, Constructing stronger and faster baselines for skeleton-based action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2022).

[29] Z. Chen, S. Li, B. Yang, Q. Li, H. Liu, Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 1113–1122.

[30] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, H. Lu, Skeleton-based action recognition with shift graph convolutional network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 183–192.

[31] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, H. Tang, Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 55–63.

[32] Y.-F. Song, Z. Zhang, C. Shan, L. Wang, Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1625–1633.

[33] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, H. Lu, Decoupling gcn with dropgraph module for skeleton-based action recognition, in: European Conference on Computer Vision, Springer, 2020, pp. 536–553.

[34] L. Huang, Y. Huang, W. Ouyang, L. Wang, Part-level graph convolutional network for skeleton-based action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11045–11052.

[35] L. Shi, Y. Zhang, J. Cheng, H. Lu, Skeleton-based action recognition with directed graph neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7912–7921.

[36] C. Si, W. Chen, W. Wang, L. Wang, T. Tan, An attention enhanced graph convolutional lstm network for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1227–1236.

[37] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian, Actional-structural graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3595–3603.

[38] H. Quan, A. Bonarini, HAVPTAT: A human activity video pose tracking annotation tool, Softw. Impacts 12 (2022) 100278, http://dx.doi.org/10.1016/j.simpa.2022.100278, URL https://www.sciencedirect.com/science/article/pii/S2665963822000318.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[40] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, P. Luo, Whole-body human pose estimation in the wild, in: European Conference on Computer Vision, Springer, 2020, pp. 196–214.

[41] W. McKinney, et al., Pandas: a foundational python library for data analysis and statistics, Python High Perform. Scient. Comput. 14 (9) (2011) 1–9.

[42] H. Quan, H. Yu, B. Bonarini, POLIMI-ITW-S: A shopping mall dataset in-the-wild, 2022, Available online at: https://airlab.deib.polimi.it/polimi-itw-s-a-shopping-mall-dataset-in-the-wild/.