Moving 3D Pose Estimation with Portable Wi-Fi

Name1 Surname[1,2☯], Name2 Surname[2☯], Name3 Surname[2,3¤], Name4 Surname[2], Name5 Surname[2‡], Name6 Surname[2‡], Name7 Surname[1,2,3*], with the Lorem Ipsum Consortium[¶]

1 Affiliation Dept/Program/Center, Institution Name, City, State, Country
2 Affiliation Dept/Program/Center, Institution Name, City, State, Country
3 Affiliation Dept/Program/Center, Institution Name, City, State, Country

☯These authors contributed equally to this work.
‡These authors also contributed equally to this work.
¤Current Address: Dept/Program/Center, Institution Name, City, State, Country
†Deceased
¶Membership list can be found in the Acknowledgments section.
* correspondingauthor@institute.edu

Abstract

Doraemon.

Author summary

Doraemon.

Introduction

Wi-Fi is one of the most common network mediums nowadays. Pervasively, it is used for establishing a wireless network to connect to the internet. But, there are still many more functions Wi-Fi is good at. Wi-Fi can also be applied in fields beside connecting to the internet according to its stability being upgraded continuously. Decent Wi-Fi connectivity can extract more data other than the data to be transmitted like concentration, speed, obstacle between the transmission. Those can be composed to be many useful applications like Localization, Activity classification and etc.

Camera is a very good tool for monitoring things and is being used as a very effective data collector for mapping to the ground truth to create many popular machine-learning-based usable models like Pose estimation, Text segmentation, Object detection and many more. However, camera may be unavoidably judged as a serious privacy infringement since the data obtained like photos or videos are too clear and possess too much information that might be used in a bad way.

There are many works tried to extract those extractable features like videos does from Wi-Fi. But, they are mostly working with very specific tools and Network Interface Card (NIC) connected to a laptop running Linux that is currently one of the ways allowing to obtain fine-grain Channel State Information (CSI), the descriptive data of the Wi-Fi propagating in that environment. Those limitations significantly decrease simplicity of implementation. It is hard for public demonstration and intregration with many updated tools in operating system like Windows or OSX.

Actually, there are other existing ways for obtaining the CSI. One is from a ubiquitously used microprocessor, ESP32. which is still not much explored in Wi-Fi exploiting field. It is simple to implement and can be easily integrated with other tools in many platforms due to its massively produced external tools.

Human pose estimation is one of the most popular topics in machine-learning-based field. It can be used for visualization directly and also applied for further applications like Game-controlling, Activity classification, Violence detection and many more. Most of the models extract human pose from videos which are taken from camera which conduct privacy issue like statement above. We are expecting to extract human pose from Wi-Fi CSI to solve the issue. So, this paper proposes a machine-learning-based model to create a mapping rule from Wi-Fi CSI to 3D moving human pose estimation by using ESP32.

Background

Pose Estimation

There are many machine-learning-based human body pose estimation tools proposed and available online. Those can be found in both 2D and 3D. In our paper, we choose a Light-weight 3D pose estimation to create annotations due to its simplicity and the hypothesis that 3D should suit the most in our work. The project can be found at [github-lw3d] and is based on [4] and [5]. Its job is to simply create 3D human pose annotation from videos. Then, feed to our works training process as annotations.

Wi-Fi

Wi-Fi is a well-known connectivity with no wire needed (wireless). It has been used as a medium for connecting to the internet for over 10 years. However, the Wi-Fi is the name covering IEEE 802.11 n/g/ac protocols. It delivers data through 2.4/5GHz frequency with multiple channels. The bandwidth in each channel is 22MHz. the data are to be transmitted parallelly with multiplexing technique named orthogonal frequency division multiplexing (OFDM). Each carrier may propagate to a receiver with encountering many obstacles. The effect of that situation is the Doppler Effect. So, Channel State Information (CSI) is represented as physical layer indicator that can be used to investigate how each channel propagate to the receiver or back to the transmitter.

If a sender sends data to a receiver through Wi-Fi, the data will be surely not transmitted without any loss.

CSI data

As mentioned in Wi-Fi that data propagating to the receiver while touching surrounding environment, the CSI is a variation of the data. The CSI can be found at both sender and receiver since receiver may transmit data back. In this paper, we consider to mainly use CSI at the transmitter. Let the sender use the modulation method of 16-quadrature amplitude modulation (16-QAM) which one carrier can carry 4 bits. When the sender needs to send a '1111', the modulation returns $x = 1 + 1i$. Then, transmit to the receiver. At the receiver, let the obtained data is $y = 0.8 + 0.9i$. So, the CSI can be computed by the variation $h = y/x = 0.2 + 3.4i$.

Human body is literally water which reflect radio wave like Wi-Fi. [6], [7] and [8] have proven that human body can affect the CSI.

ESP32

ESP32 is a very popular single-board computer (SBC). With its affordable price and many available additional tools, ESP32 is commonly used in Internet of Things field. Moreover, it can be applied in research field. Quantitative CSI can be obtained from Wi-Fi in ESP according to [26]. The number of available subcarriers in ESP32 is 64.

According to the detail about Wi-Fi mentioned in Wi-Fi, the Wi-Fi in ESP32 has some limitation. It supports only 2.4GHz frequency and can be set only one channel over a connection. The bandwidth of each channel is 22MHz. The CSI can be both obtain from Access point (AP) and Station (STA) as shown in Fig 1. The frequency of each channel is as 802.11 standard.

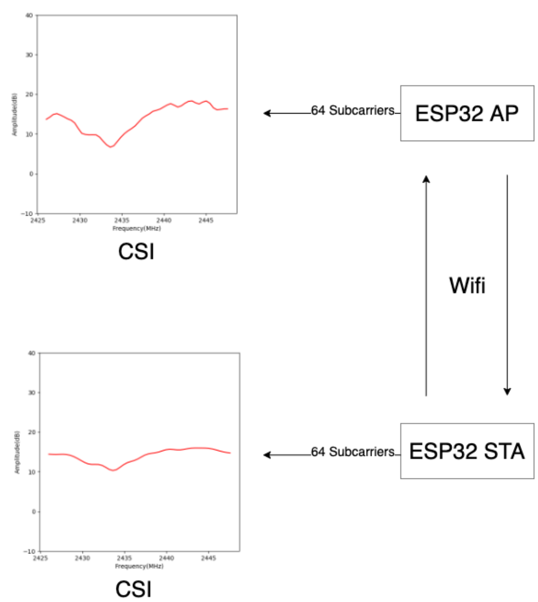

Fig 1. CSI from ESP32s with channel 6.

Materials and methods

Concept

Other famous proposed works like [6] [7] and [25] focus on line-of-sight (LOS) in between AP and STA while our work uses 2 directional Wi-Fi antennae and focus on reflection from human body as shown in Fig 2 on the left.

The reason we name “Moving Pose Estimation” instead of “Pose Estimation” is that CSI is not only affected by human body but mainly by overall environment. This means that 2 corresponding human poses can result obviously different CSIs if the environment around are not exactly the same as shown in Fig 2.

So, the definite detection of human standing still in every environment is nearly impossible since the CSI of that situation may be found exactly matched to a CSI of the environment that a big bottle of water placed in front of ESP32. In short, if it does not move, we do not know if it is human.

Meanwhile, the moving pose is totally different because we focus on its change instead. The example of mapping CSI’s change to Activity Classification can be found

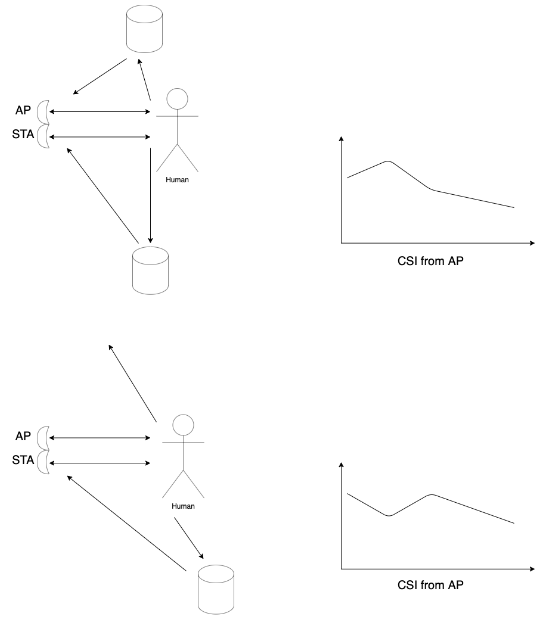

Fig 2. 2 different CSIs resulted from corresponding human poses.

in [8] and [27]. Our work does likewise but focusing on Pose Estimation instead of Activity Classification.

In different environment, the CSIs are different. But, the corresponding moving pose may affect to the same changing pattern of CSI. This hypothesis is investigated in the upcoming parts.

All steps of training method is shown in Fig 3.

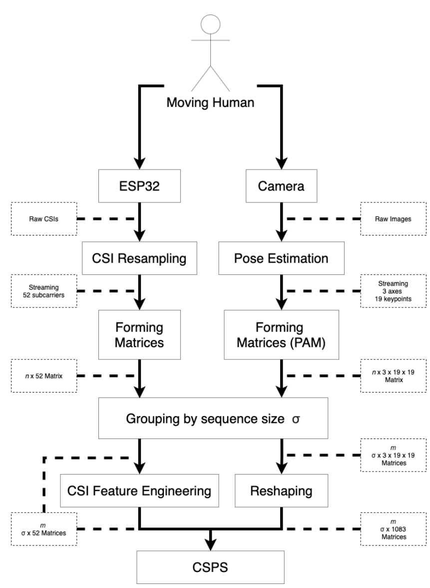

Fig 3. All steps of training method.

Pre-processing

CSI Resampling

As mentioned in ESP32, there are 64 subcarriers in CSI data from ESP32 but there are only 52 those are usable while the rest are null. So, we can construst a tensor of 1×52 to represent each CSI. We are to map CSI from the ESP32 to 3D human pose annotation from a camera. The sampling rate of the camera are set to 30Hz. So, we have 30 human pose annotations for one second. For the ESP32, the sampling rate is originally unpredictable and not constant but it is running around 120Hz. So we do a process called "Resampling" to obtain CSI at rate 30Hz in order to synchronize timestamps to each human pose annotation.

An example of CSI Resampling is shown in Fig 4. The top graph shows that the the original CSI is logged unstably. The bottom one is to pick a timestamp at rate 30Hz and calculated each with the closet data from the original with a simple mathamatical weight equation as in Eq. 1 in order to predict CSI at timestamp corresponding to each human pose annotation.

$$\begin{aligned} CSI_{now} &= CSI_{before} \\ &+ \left(\frac{ts_{now} - ts_{before}}{ts_{after} - ts_{before}} \times (CSI_{after} - CSI_{before})\right), \end{aligned} \tag{1}$$

where ts_{now}, ts_{before}, ts_{after} are desired timestamp, timestamp at the closest CSI before the desired timestamp and timestamp at the closest CSI after the desired timestamp respectively. And, CSI_{now}, CSI_{before}, CSI_{after} are CSI at the desired timestamp, CSI before the desired timestamp and CSI after the desired timestamp respectively.

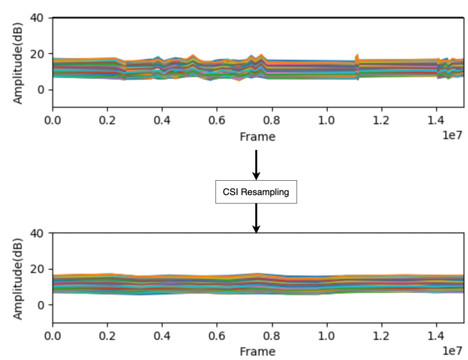

Fig 4. An example of CSI resampling.

Thus, we are able to map each CSI sample to one 3D human pose annotation at the corresponding timestamp.

Pose Estimation and PAM formation

We use [github-lw3d] to estimate 3D human pose from videos as stated above. the estimation gives us a 19×3 matrix for each image. The 19×3 matrix are used as an annotation where 19 is for keypoints in human body and 3 is for 3 axes coordination as shown in Fig 5. But, the annotation can still possess too much independency. Some alignment of keypoints may lead to some impossible pose e.g. body keypoint found very far from neck keypoint or head keypoint attached to hip keypoint. We assume those poses are not possible for normal human pose. To preserve those constraints, we form a

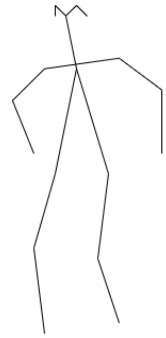

Fig 5. 19 keypoints of human pose.

pose adjacent matrix (PAM) from an original 19×3 matrix. the PAM is applied for all x, y and z axes. Each are to be form their 19×19 matrix by the following equations.

$$x'_{i,j} = \begin{cases} x_i - x_j & i \neq j \\ x_i & i = j \end{cases} \tag{2}$$

,

$$y'_{i,j} = \begin{cases} y_i - y_j & i \neq j \\ y_i & i = j \end{cases} \tag{3}$$

and

$$z'_{i,j} = \begin{cases} z_i - x_j & i \neq j \\ z_i & i = j \end{cases} \tag{4}$$

.

The PAM is finally a $3 \times 19 \times 19$ matrix created from 3 matrices of x',y' and z' stacked. Apparently, one PAM represents one human pose.

Conclusively, we are making a model by mapping a sequence of 1×52 matrix from CSI to each sequence of $19 \times 3 \times 3$ PAM from moving human pose annotation with the corresponding timestamp as shown in Fig 6.

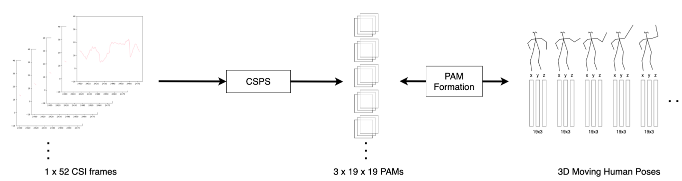

Fig 6. A mapping rule from CSI frames to 3D human poses.

Processing

Mapping CSI and PAM

Let D be a set of synchronized pose and CSI data package. Each pair has corresponding timestamp.

$$D = (C_t, P_t), t \in [1, n], \tag{5}$$

where n is a number of pairs, C is for CSI data from ESP32, P is for PAM annotation as the ground truth, t is the timestamp index when those 2 were collected.

Grouping By Sequence Size

Let σ be an adjustable window size. σ size of D are fed to the model solely. So, $\sigma \leq n$. And, Let m be the number of feeding iteration, $m = \lfloor \frac{n}{\sigma} \rfloor$.

Forming Network Layer

The network swallow m training data as an input, where each is a sequencial set of (C_t, P_t) at a corresponding timestamp with size σ. Let Γ be a representation of each sequencial set.

$$\Gamma = (C_u, P_u), u \in [1, \sigma], \tag{6}$$

where u is the timestamp index when those 2 were collected. Apparently, the number of Γ is m.

As the Γ is a sequencial set with size σ, we assume that Long-short term memory (LSTM) [28] is suitable for this type of data.

CSI Feature Engineering As mentioned in Concept, CSI (C) value implies environment where signal propagating. So, we need to ignore it and focus only on its change in order to universalize the circumstance. It means CSI variousity makes the model not to be applicable in every environment. To solve it, we simply sequentially substract C_u in each Γ backward to preserve only how CSI changes over each sequence by the following equation,

$$C_u = \begin{cases} \text{all } 0 & u = 1 \\ C_u - C_{u-1} & u > 1 \end{cases}, C_u \in \Gamma. \tag{7}$$

CSI Sequence to Pose Sequence (CSPS) The summation of layers is shown in Table 1. It is designed to shape a CSI Sequence ($\sigma \times 52$ tensor) to a predicted Pose Sequence ($\sigma \times 3 \times 19 \times 19$ tensor).

Table 1. Layers in CSPS.

Layer (type)	Output Shape	Param #
Bidirectional LSTM	(None, 200)	122400
RepeatVector	(None, σ, 200)	0
Bidirectional LSTM	(None, σ, 200)	240800
TimeDistributed over Dense	(None, σ, 1083)	217683

Table notes Phasellus venenatis, tortor nec vestibulum mattis, massa tortor interdum felis, nec pellentesque metus tortor nec nisl. Ut ornare mauris tellus, vel dapibus arcu suscipit sed.

We first use Bidirectional LSTM as an encoder layer with the input size of $(\sigma, 52)$ to satisfy dimension of sequential C_u in Γ. Then, the repeat vector is added with time σ to make the model treat with correct time-step. Afterward, we place a decoder layer with another Bidirectional LSTM.

Next, we dense the decoder to be 1083 where can be reshaped into $3 \times 19 \times 19$ later.

Lastly, the TimeDistributed layer is used in order to make the model treat the output for each time-step individually.

Results

Data Collection

dance in ma room ((We collected data under an approval of Carnegie Mellon University IRB 4 . We recruited 8 volunteers, and asked them to do casual daily actions in two rooms of the campus, one laboratory room and one class room. Floor plans and data collection positions are illustrated in Fig. 8. During the actions, we run the system in Fig. 3 to record CSI samples and videos, simultaneously. For each volunteer, data of his first 80test the networks. The data size of training and testing are 79496 and 19931, respectively))

Error Metric

Percentage of Correct Keypoint (PCK) is widely used to evaluate the performance of human pose estimation according to [6].

$$PCK_i@a = \frac{1}{N}\sum_{i=1}^{N} I\left(\frac{\|\, pd_i \text{ - } gt_i \,\|_2^2}{\sqrt{rh^2 + lh^2}} \leq a\right), \tag{8}$$

where $I(\cdot)$ is a binary indicator that outputs 1 while true and 0 while false, N is the number of frames, i is the index of keypoints that $i \in 1, 2, ..., 19$. The rh and lh are for the positions of the right shoulder and the left hip from the ground truth, respectively. So, the $\sqrt{rh^2 + lh^2}$ is considered as the length of the upper limb from the ground truth, which is used to normalize the prediction error length $\|\, pd_i$ - $gt_i \,\|_2^2$, and pd_i and gt_i are coordinates of prediction and ground-truth at the keypoint i respectively.

Experimental Result

All steps of testing method is shown in Fig 7.

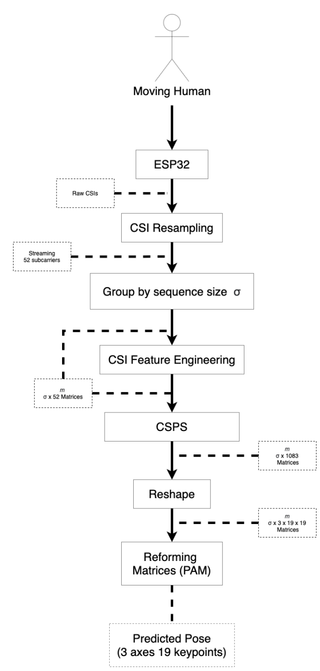

Fig 7. All steps of testing method.

An evaluation is achieved with by the algorithm written in Python 3.8. The code is available on *github*. An environment under the evaluation is MacBook Pro 2016, 2.6 GHz i7 processor, 16 GB of RAM.

As parameters stated in the previous sections, we demostrate the prediction by the followings.

Table 2 and Table 2 shows the estimation performance of 19 body keypoints in PCK when $\sigma = 15$ and $\sigma = 30$ respectively.

Github[1].

Table 2. Table of PCK.

Order	Keypoint	PCK@5	PCK@10	PCK@20	PCK@30	PCK@40	PCK@50
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
Avg							

Table notes Phasellus venenatis, tortor nec vestibulum mattis, massa tortor interdum felis, nec pellentesque metus tortor nec nisl. Ut ornare mauris tellus, vel dapibus arcu suscipit sed.

Discussion

Doraemon.

Table 3. Table caption Nulla mi mi, venenatis sed ipsum varius, volutpat euismod diam.

Heading1				**Heading2**			
$cell1row1$	cell2 row 1	cell3 row 1	cell4 row 1	cell5 row 1	cell6 row 1	cell7 row 1	cell8 row 1
$cell1row2$	cell2 row 2	cell3 row 2	cell4 row 2	cell5 row 2	cell6 row 2	cell7 row 2	cell8 row 2
$cell1row3$	cell2 row 3	cell3 row 3	cell4 row 3	cell5 row 3	cell6 row 3	cell7 row 3	cell8 row 3

Table notes Phasellus venenatis, tortor nec vestibulum mattis, massa tortor interdum felis, nec pellentesque metus tortor nec nisl. Ut ornare mauris tellus, vel dapibus arcu suscipit sed.

[1] https://github.com/rtmtree/CSPS

Conclusion

Doraemon.

Supporting information

S1 Fig. Bold the title sentence. Add descriptive text after the title of the item (optional).

S2 Fig. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 File. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Video. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Appendix. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Table. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

Acknowledgments

We thank to Sirindhorn International Institute of Technology for providing technical environment and supportive information.

References

1. Conant GC, Wolfe KH. Turning a hobby into a job: how duplicated genes find new functions. Nat Rev Genet. 2008 Dec;9(12):938–950.

2. Ohno S. Evolution by gene duplication. London: George Alien & Unwin Ltd. Berlin, Heidelberg and New York: Springer-Verlag.; 1970.

3. Magwire MM, Bayer F, Webster CL, Cao C, Jiggins FM. Successive increases in the resistance of Drosophila to viral infection through a transposon insertion followed by a Duplication. PLoS Genet. 2011 Oct;7(10):e1002337.

4. Osokin D. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. arXiv:1811.12004 [cs.CV]. 2018 Nov;18.

5. Mehta D, Sotnychenko O, Mueller F, Xu W, Sridhar S, Pons-Moll G, Theobalt C. Single-Shot Multi-Person 3D Pose Estimation From Monocular RGB. arXiv:1712.03453 [cs.CV]. 2018 Aug;28.

6. Wang F, Panev S, Ziyi D, Han J, Huang D. Can WiFi Estimate Person Pose?. arXiv:1904.00277 [cs.CV]. 2019 Apr;2.
7. Liu J, Liu H, Chen Y, Wang Y, Wang C. Wireless Sensing for Human Activity: A Survey. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 22, NO. 3, THIRD QUARTER 2020.
8. Chowdhury TZ, Leung C, Miao CY. WiHACS: Leveraging WiFi for Human Activity Classification using OFDM Subcarriers' correlation. IEEE, GlobalSIP 2017.
9. Guo L, Wang L, Liu J, Zhou W, Lu B. HuAc: Human Activity Recognition Using Crowdsourced WiFi Signals and Skeleton Data. Wireless Communications and Mobile Computing Volume 2018, Article ID 6163475.
10. Wang F, Feng J, Zhao Y, Xiaobin Zhang, Zhang S. Joint Activity Recognition and Indoor Localization with WiFi Fingerprints. arXiv:1904.04964 [cs.CV]. 2019 Jul;18.
11. Al-qaness MAA, Li F, Ma X, Zhang Y, Liu G. Device-Free Indoor Activity Recognition System. Appl. Sci. 2016, 6, 329; doi:10.3390.
12. Wang W, Liu AX, Shahzad M, Ling K, Lu S. Device-free Human Activity Recognition Using Commercial WiFi Devices. IEEE Journal on Selected Areas in Communications. DOI 10.1109/JSAC.2017.2679658.
13. Zhao T, Li F, Tian P. A Deep-Learning Method for Device Activity Detection in mMTC Under Imperfect CSI Based on Variational-Autoencoder. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 69, NO. 7, JULY 2020.
14. Liu J, Teng G, Hong F. Human Activity Sensing with Wireless Signals: A Survey. Sensors 2020, 20, 1210; doi:10.3390/s20041210.
15. Yousefi S, Narui H, Dayal S, Ermon S, Valaee S. A Survey on Behaviour Recognition Using WiFi Channel State Information. arXiv:1708.07129 [cs.AI]. 2017 Aug;23.
16. Chen Z, Zhang L, Jiang C, Cao Z, Cui W. WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 18, NO. 11, 2019 Nov.
17. Li B, Cui W, Wang W, Zhang L, Chen Z, Wu M. Two-Stream Convolution Augmented Transformer for Human Activity Recognition. Association for the Advancement of Artificial Intelligence, 2021.
18. Luo Y, Ren J, Wang Z, Sun W, Pan J, Liu J, Pan J, Lin L. LSTM Pose Machines. arXiv:1712.06316 [cs.CV].
19. Lee K, Lee I, Lee S. Propagating LSTM: 3D Pose Estimation based on Joint Interdependency. Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11211. Springer, Cham.
20. Du X, Vasudevan R, Johnson-Roberson M. Bio-LSTM: A Biomechanically Inspired Recurrent Neural Network for 3D Pedestrian Pose and Gait Prediction. arXiv:1809.03705 [cs.RO]. 2019 Sep;13.

21. Hossain MRI, Little JJ. Exploiting temporal information for 3D human pose estimation. arXiv:1711.08585 [cs.CV]. 2018 Sep;12.

22. Pavllo D, Feichtenhofer C, Grangier D, Auli M. 3D human pose estimation in video with temporal convolutions and semi-supervised training. arXiv:1811.11742 [cs.CV]. 2019 Mar;29.

23. Chen T, Fang C, Shen X, Zhu Y, Chen Z, Luo J. Anatomy-aware 3D Human Pose Estimation with Bone-based Pose Decomposition. arXiv:2002.10322 [cs.CV]. 2021 Jan;26.

24. Ruiz AH, Porzi L, Bulo SR, Moreno-Noguer F. 3D CNNs on Distance Matrices for Human Action Recognition. MM '17, Mountain View, CA, USA. 2017 Oct;23–27.

25. Hernandez SM, Bulut E. Lightweight and Standalone IoT based WiFi Sensing for Active Repositioning and Mobility. IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020.

26. Atif M, Muralidharan S, Ko H, Yoo B. Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS). Journal of Computational Design and Engineering, 2020, 7(5), 644–656.

27. Zou H, Zhou Y, Yang J, Jiang H, Xie L, Spanos CJ. DeepSense: Device-free Human Activity Recognition via Autoencoder Long-term Recurrent Convolutional Network. 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018, pp. 1-6, doi: 10.1109/ICC.2018.8422895.

28. Hochreiter S, Schmidhuber J. LONG SHORT-TERM MEMORY. Neural Computation 9(8):1735-1780, 1997.