

# Channel State Information Auto-Labeling System (CALs) for Large-Scale Deep-Learning-Based Wi-Fi Sensing

Jung-Ik Jang, Jaehyuk Choi\*

School of Computing, Gachon University, Republic of Korea

\*Corresponding author: Jaehyuk Choi, jchoi@gachon.ac.kr

**Copyright:** © 2022 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

## Abstract

Wi-Fi sensing involves using Wi-Fi technology to sense the surrounding environments, and it has strong potential in a variety of sensing applications. Recently, several advanced deep-learning-based solutions using channel state information (CSI) data have shown great performance, but it is still difficult to use in practice without explicit data collection, which requires expensive adaptation efforts for model retraining. In this study, we propose a Channel State Information Automatic Labeling System (CALs) that automatically collects and labels training CSI data for deep-learning-based Wi-Fi sensing systems. The proposed system enhanced the efficiency of collecting labeled CSI data for supervised learning through computer vision technologies like object detection algorithms. We built a prototype of CALs to demonstrate its efficiency and collected data to train deep learning models for detecting the presence of a person in an indoor environment. Our results indicate an accuracy of over 90% using the auto-labeled datasets generated by CALs.

## Keywords

Wi-Fi sensing  
Channel state information  
Auto-labeling  
Computer vision  
Deep learning

## 1. Introduction

With the technological advancement of wireless networks and the increasing demand for them, Wi-Fi is now available everywhere. Based on this Wi-Fi infrastructure, there has been active research on utilizing channel state information (CSI), which is used for transmission control for smooth communication in

sensing technology. CSI is sensitive to the surrounding environment, i.e., its value depends on the multi-pass propagation and is influenced by interference, dispersion, refraction, and attenuation that occurs while the signal from the transmitter is transmitted to the receiver. These characteristics have been utilized in a variety of research and development applications, including human

identification <sup>[1,2]</sup>, activity recognition <sup>[3-5]</sup>, indoor presence detection <sup>[6,7]</sup>, and indoor localization <sup>[7-10]</sup>.

However, data-driven deep learning techniques that use CSI for training have been limited by the fact that they can only learn patterns that are domain-dependent, where the data is collected. This is because the propagation of Wi-Fi signals, which are in the form of radio frequency (RF), is affected by even small changes in the surrounding environment, leading to changes in the collected CSI values and their patterns. To address this issue, there has been a growing focus on researching deep learning technologies that remain unaffected by environmental changes <sup>[11,12]</sup>, and attempts have been made to learn data after generating new CSI data using generative adversarial networks GANs <sup>[5]</sup>. However, when trying to perform Wi-Fi-based sensing by changing the target space itself, performance degradation is inevitable with the current technology level, and new data collection is required. In particular, labeling the collected CSI data for supervised learning cumbersome and time-consuming task.

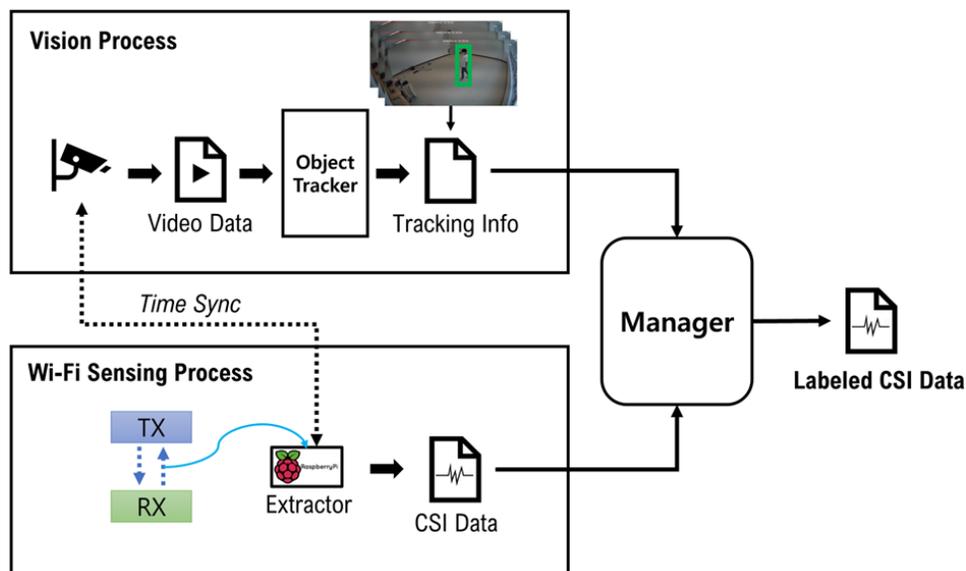
In this paper, we proposed a Channel State Information Auto-Labeling System (CALs) for large-scale deep-learning-based Wi-Fi sensing that utilizes object recognition technology of computer vision (CV)

to improve the efficiency of CSI data collection and labeling for training deep learning models based on supervised learning. CALs performs CSI measurements in parallel with the CV module, such as object recognition of images from cameras installed in the target domain when collecting Wi-Fi sensing data, and uses the recognition results of CV to label the collected CSI data. Based on this automated labeling function, CALs can efficiently collect large amounts of data, and various labels are possible depending on the CV model. It also has the advantage of easy model replacement. In addition, a deep learning model was trained and evaluated for real-time human presence detection in indoor environments using CSI data collected by CALs, and achieved a performance of over 90%.

## 2. CALS

### 2.1. System structure and processing process

**Figure 1** shows the structure of the proposed CALs. There are two processes involved in this system: CV process and Wi-Fi sensing process, which occur simultaneously. Prior to the system execution, time synchronization is achieved by both the camera and the CSI extractor through the same Network Time Protocol



**Figure 1.** Proposed CALs architecture

(NTP) server.

In this study, the object tracking technology is used in the CV process, and the ByteTrack<sup>[13]</sup> model, which is achieving state-of-the-art (SOTA) in the field of Multi-Object Tracking (MOT), is applied for the purpose of binary class labeling for training a deep learning model for human presence detection. When the images from the cameras installed in the CSI extraction space are input to the object tracking model, the object tracking model outputs the frame number, target ID, and bounding box coordinates for the frames in which a person is detected. Since time information is required for CSI labeling, the time value corresponding to the frame index is added to the output value based on the fps of the video.

During the Wi-Fi sensing process, data is extracted from Wi-Fi communication through a CSI extractor. For the subcarrier frequency transmitted from the transmitter to the receiver, the receiver generates CSI (amplitude and phase information) and retransmits it to the receiver. During the process, the CSI extractor collects data through User Datagram Protocol (UDP) packets. For data packets, CSI can be represented using the same matrix as Equation (1). The number of antennas in the transmitter and the number of antennas in the receiver determine the size of the matrix, with elements representing a set of subcarriers composed of complex values, as shown in Equation (2). For the second subcarrier frequency, represented by amplitude and phase information, it can be expressed as Equation (3).

$$CSI_p = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,r} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ H_{t,1} & H_{t,2} & \cdots & H_{t,r} \end{bmatrix} \quad (1)$$

$$H_{(i,j)} = [h_{(i,j,1)}, \dots, h_{(i,j,k)}] \quad (2)$$

$$h_k = |h_k| e^{j\angle h_k} \quad (3)$$

The CSI value as well as the receiver's MAC address and time value are extracted from the UDP packet and stored in a table format. The MAC address is used to determine the source of the collected CSI.

Finally, the target tracking information obtained from the CV process and the CSI data obtained from the Wi-Fi sensing process are labeled by the CALS manager by comparing the time information. First, in the target tracking information, the continuous time when the target is tracked is divided into  $T_i$ , which contains the start frame time and end frame time information, as shown in Equation (4). Equation (5) shows the time information divided into a total number. At this time, the criterion for separating consecutive time is determined by the frame threshold  $\theta_f$ , which is set to 10 in the 20 fps environment in this study. For all the temporal information belonging to  $[s_i, e_N]$ , the CSI time ( $t$ ) that overlaps with the corresponding time zone will be labeled as 1 because a person is present in the room, and the CSI data that does not fall anywhere in the range  $[s_i, e_N]$  will be labeled as 0. At this time, a specific time value,  $\gamma$ , is used to reduce the CSI labeling error, and it is labeled as -1 for  $S_i - \gamma \leq t < S_i$ ,  $e_i < t \leq e_i + \gamma$ . The operation flow of the described CALS can be seen in **Figure 2**.

$$T_i = (s_i, e_i) \quad (4)$$

$$T = \{T_1, T_2, \dots, T_N\} \quad (5)$$

$$CSI(t) = \begin{cases} 1 & \text{if } s_i \leq t \leq e_i, \\ 0 & \text{if } s_1 < t < e_N \text{ and } t \notin T, \\ -1 & \text{if } s_1 < t < e_N \text{ and} \\ & s_i - \gamma \leq t \leq s_i \text{ or} \\ & e_i < t < e_i + \gamma. \end{cases} \quad (6)$$

## 2.2. CALS manager module

In the proposed system, the manager is responsible not only for labeling CSI data by synthesizing the results of the two processes, but also for processing to improve the performance of deep learning models and provide data customized to the user's needs.

In training binary classification models, the balance of data classes can achieve more accurate prediction performance. Therefore, since the number of situations without people in the experimental space is larger, the number of CSI data labeled with that class is adjusted to balance the number of CSI data collected during the time when people are active in the space. Unnecessary data collection can also be avoided by ensuring that

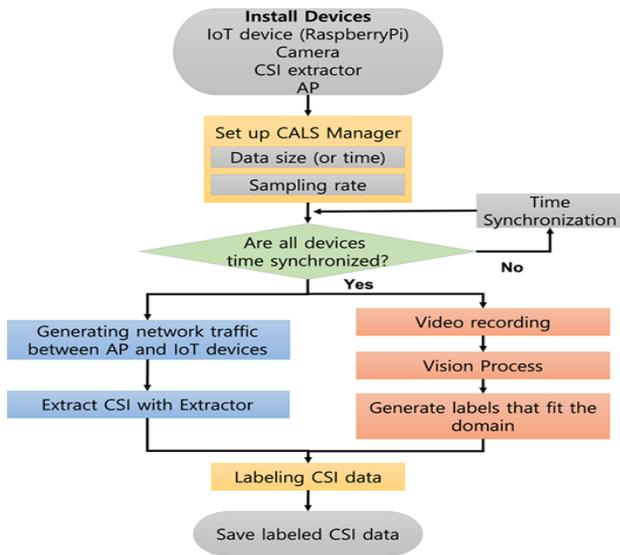


Fig. 2. Proposed CALS flowchart.

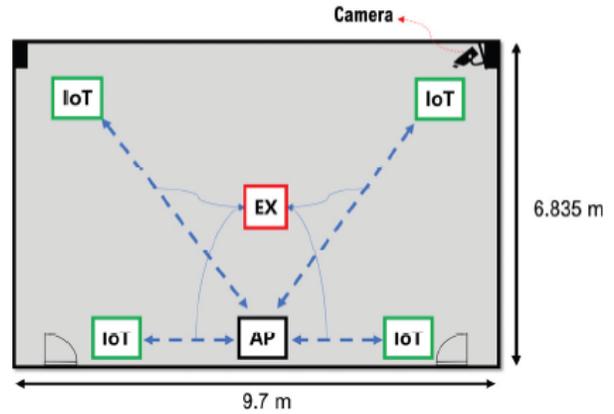


Figure 3. Human presence detection testbed

data is collected at the size required by the user, which can be accomplished by setting the collection time rather than the size. Additionally, the user can dynamically adjust the desired CSI data sampling rate per second by communicating with the CSI extractor.

### 3. Real-time human presence detection in indoor environments based on deep learning

Human presence detection in indoor environments is used for various purposes in various facilities such as smart homes and industrial facilities. Usually, camera-based monitoring or physical sensors are used, but if CSI is used, the privacy problem of camera-based monitoring system and the small amount of information of physical sensors can be solved<sup>[15]</sup>. In addition, CSI can be utilized without additional equipment due to the establishment of Wi-Fi infrastructure.

In **Section 3**, we utilized CSI data collected by CALS to train and evaluate a deep learning model for real-time human presence detection.

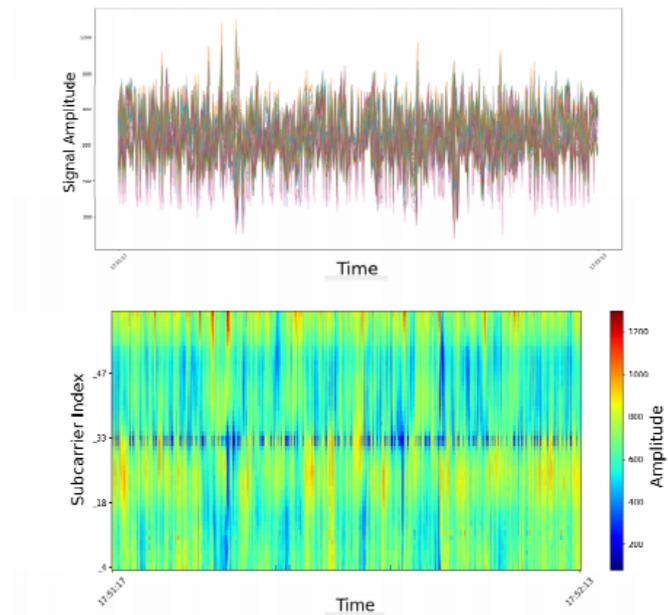
#### 3.1. Test environment

**Figure 3** shows the CSI collection environment for training the human presence detection model. A total of

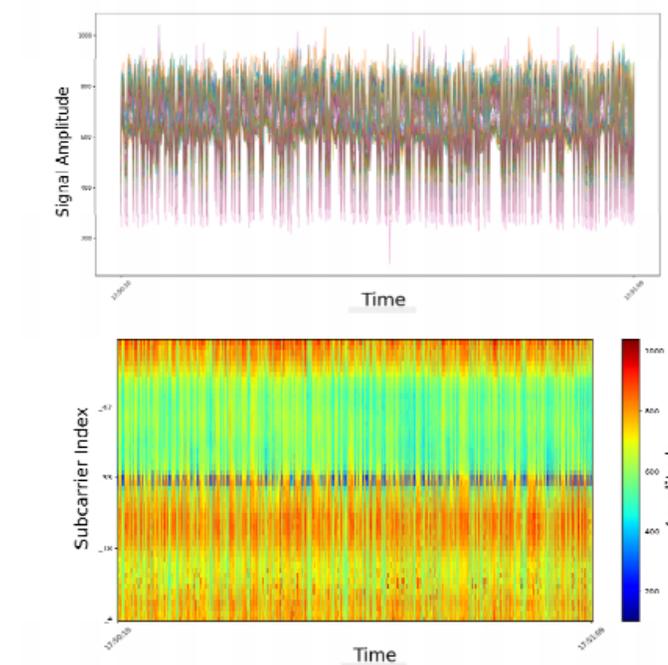
four Raspberry Pi 4 models were used as IoT devices, and a TP-LINK model was used as an access point (AP), with one antenna pair each to obtain 64 subcarrier information for each pair in the Wi-Fi 4 (IEEE 802.11n) 2.4 GHz band, 20 MHz bandwidth. For the CSI extractor, we installed the Nexmon CSI Extractor<sup>[14]</sup> provided by Nexmon on the Raspberry Pi 4 model. The extractor extracts CSI data generated during wireless network communication between IoT devices and APs regardless of their location. The experimental scenarios included repeated walking and stopping between IoT devices and APs, as well as collection in the absence of people. All experimental processes were recorded by installing a WiseNet SNK-B73047BW camera in a location that covers the entire space.

#### 3.2. Data visualization

The CSI data collected by CALS in this study consists of binary classes with a value of 1 for the presence of a person and 0 for the absence of a person. **Figures 4 and 5** illustrates the amplitude values of the subcarriers for each class. In the top graph, we can see that they have different signal patterns, and in the heat map, we can see that the amplitude variation of the signal is more pronounced in class 1. This can be attributed



**Figure 4.** Amplitude of 52 subcarriers when there is human activity indoors



**Figure 5.** Amplitude of 52 subcarriers when no one is in the room

to the effect of multipath propagation due to human activity between the receiver and transmitter. To obtain the graph, we used 52 subcarriers with 12 null, pilot subcarriers removed.

### 3.3. Data preprocessing

For pre-training and testing, we used CSI extracted

from about 415,000 packets, and the data size per class is balanced at about 200,000 packets each. For data preprocessing, we first removed unused attributes (MAC address, time), null subcarriers that serve to protect the band so that Wi-Fi can be used with other wireless technologies, and 12 pilot subcarriers used for Wi-Fi link control. We then divided the training and test data by 8:2.

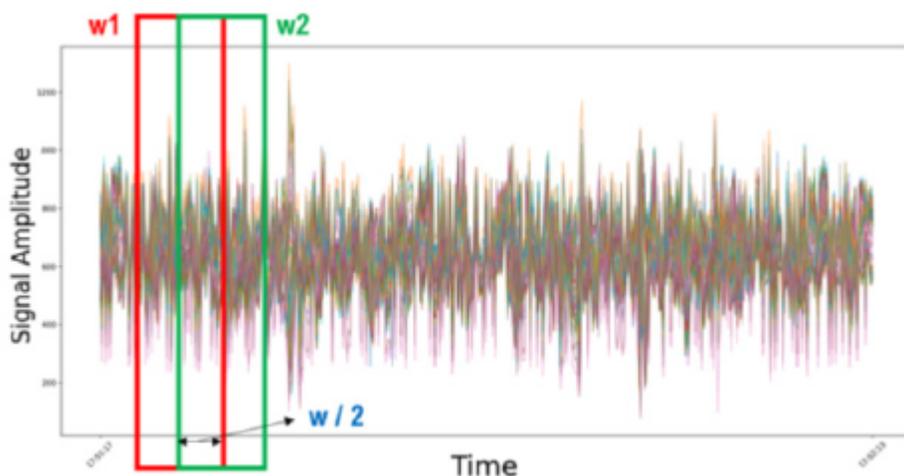
Since the data has 52 subcarriers with different scales of amplitudes, using it for training as is can lead to poor performance. Therefore, for the training data, we used StandardScaler provided by scikit-learn to standardize the data so that it has a distribution with a mean of 0 and a standard deviation of 1. In this case, the signal outliers were not removed and used for training because they often occur in the real world. The standardization information was stored separately and used for standardization of test data and real-time data.

Since CSI data was collected sequentially over a period of time (time series data), not only current information but also past information was used for training. Therefore, in this study, we set a constant window size and trained on a window-by-window basis. In this case, two consecutive windows were organized in such a way that half of the size (half of the time) was overlapped. **Figure 6** shows an example of window utilization in this study.

### 3.4. Real-time presence detection process and evaluation

For real-time human presence detection, we first trained and evaluated three models before selecting the most efficient one. The models used Random Forest, which is an ensemble learning method with good generalization performance, long short-term memory

network (LSTM), which is widely used for learning time series data, and 1D CNN. When we checked the correlation coefficient between each subcarrier for the data collected by CALS, we found that the three consecutive subcarriers were highly correlated. Therefore, we selected the three consecutive subcarriers for training, and the results of the evaluation of the test sets are shown in **Figure 7**. For the three models, we evaluated the window size in increments of 20 from 10 (1 second) to 70 (7 seconds) and found that the 1D CNN had the highest accuracy. All three models also achieved higher accuracy with larger window sizes, but we chose a window size of 2.5 seconds (half the window size), i.e., window 50, which is a reasonable time for real-time classification tasks. **Table 1** shows the training time for each model. As you can see, the random forest model trains the fastest. However, as we increased the number of hyperparameters or estimators and expanded the window size in an effort to enhance performance, we noticed that the improvement plateaued at 93%. In contrast, the other two time series models continued to improve and reached a performance level of 99% when the window size was increased to 70 or more. Consequently, we decided to prioritize the 1D CNN as our primary model due to its faster learning speed, even though it achieved similar performance to the other models.



**Figure 6.** Example of window utilization

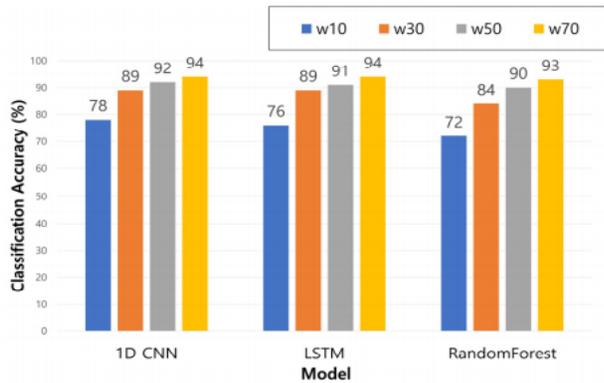


Figure 7. Evaluation results for datasets collected by CALs

Table 1. Learning time according to model

Model	Time (s)
1D CNN	1278.41
LSTM	1587.01
Random Forest	90.51

For training the 1D CNN, we used Binary Cross Entropy as the loss function for binary classification and Adam as the optimizer. We set the maximum number of training epochs to 50, the batch size to 32, and the learning rate to start with an initial learning rate of 0.01 and decrease by a factor of 10 every 10 training cycles. The configuration of the proposed 1D CNN model is shown in Table 2. The input was the set window size ( $w\_size$ ), and each CNN operation was padded to have the same size output. Rectified Linear unit (ReLU) was used as the activation function applied after the computation. The total number of parameters trained was 74,811.

Table 2. Configuration of the proposed 1D CNN

Type	Output size	Filter
Input	$w\_size \times 1$	-
Conv1D	$w\_size \times 128$	$5 \times 1$
MaxPool1D	$w\_size/2 \times 128$	$1 \times 2$
Conv1D	$w\_size/2 \times 128$	$3 \times 1$
GlobalMaxPool1D	$1 \times 128$	-
FC	$1 \times 128$	-
FC	$1 \times 64$	-
Sigmoid	1	-

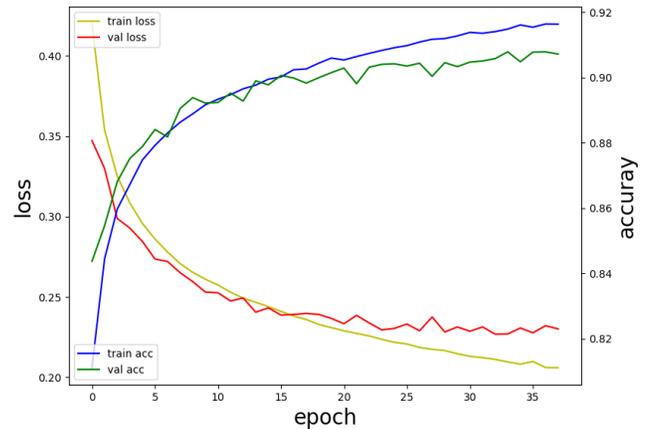


Figure 8. Loss and accuracy of 1D CNN according to epoch.

Figure 8 shows the classification accuracy as a function of the number of training epochs when the window size was set to 70, which was the highest performance among the test cases. The maximum number of training epochs was set to 50, but training ended at 38 epochs because the validation loss no longer showed a steady decline starting at 33 epochs. The loss and accuracy graphs for the training and validation datasets all show an ideal learning curve shape, and the accuracy of the validation dataset is about 94%.

Figure 9 shows the process of real-time human presence detection. First, the CSI obtained in real-time is converted to amplitude values, and unused features are removed and standardized with pre-stored information. Then, for each subcarrier  $K$  selected in the pre-training, the data is put into two queues of window size. Once the queues are filled, the data is combined and fed into the pre-trained 1D CNN model, which in turn determines the presence of a person. Then, for new CSI data, the queue performs a simultaneous pop and push. The classification process is performed every time half of the window size is updated by discarding the first data and pushing in new data.

For real-time human presence detection, three new indoor spaces were selected and evaluated first with the pre-trained model, then retrained on the CSI data collected in each space, and then evaluated again.

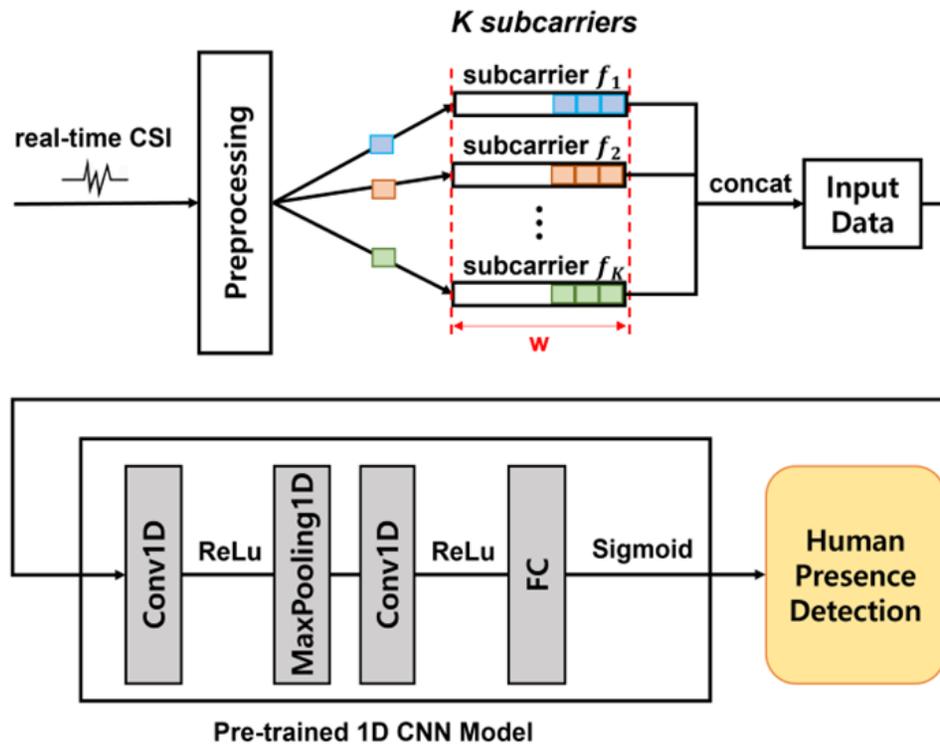


Figure 9. Overview of real-time human presence detection process

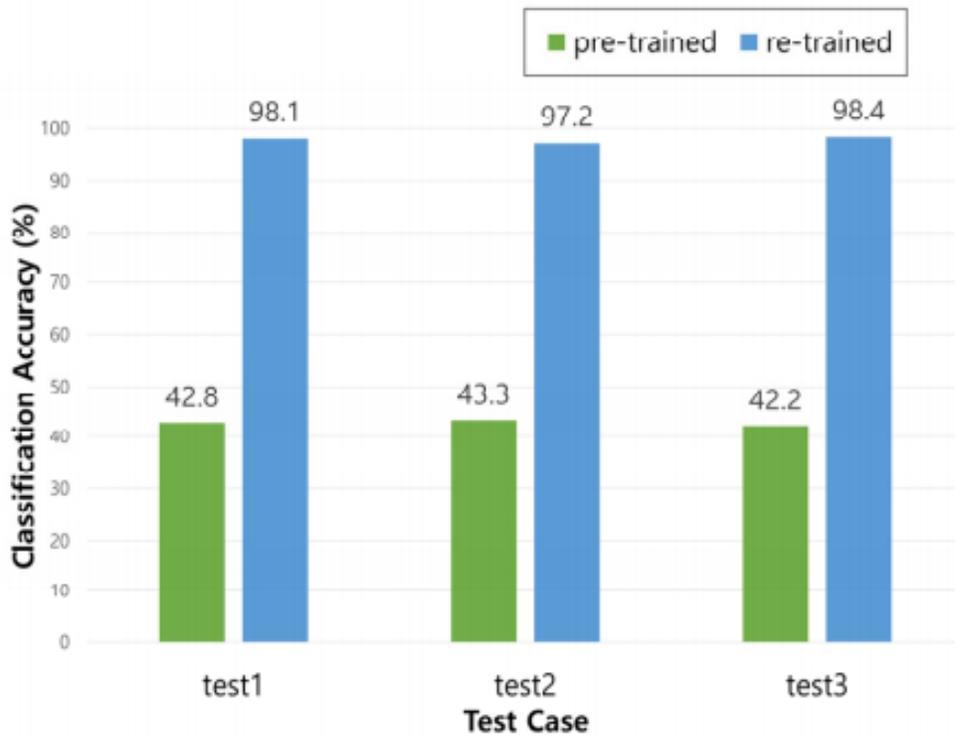


Figure 10. Accuracy results for 3 testbeds

Based on the graph in **Figure 10**, the pre-trained model showed low accuracy in the low 40% range for all test spaces, but the evaluation after retraining on newly collected data showed accuracy in the high 90% range. This demonstrates the environment-sensitive nature of CSI as described earlier.

### 3. Conclusion

In this paper, we proposed CALs, an automatic CSI labeling system, to improve the efficiency of CSI

labeling, and obtained an accuracy rate of over 90% by performing and evaluating human presence detection in indoor environments using deep learning on data collected by CALs. We also confirmed the sensitivity of CSI to environmental changes by conducting tests on new spaces using pre-trained models. In future work, we plan to study the verification of labeling errors that are currently lacking in CALs and how to utilize CSI in dynamic environments.

#### Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2020R1A2C1013308).

#### Disclosure statement

The authors declare no conflict of interest

### References

- [1] Wang J, Zhao Y, Fan X, et al., 2018, Device-Free Identification Using Intrinsic CSI Features. *IEEE Transactions on Vehicular Technology*, 67(9): 8571–8581, <https://www.doi.org/10.1109/TVT.2018.2853185>
- [2] Mo H, Kim S, 2021, A Deep Learning-Based Human Identification System with Wi-Fi CSI Data Augmentation. *IEEE Access*, 9: 91913–91920, 2021. <https://www.doi.org/10.1109/ACCESS.2021.3092435>
- [3] Wang W, Liu AX, Shahzad M, et al., 2017, Device-Free Human Activity Recognition Using Commercial WiFi Devices. *IEEE Journal on Selected Areas in Communications*, 35(5): 1118–1131. <https://www.doi.org/10.1109/JSAC.2017.2679658>
- [4] Chen Z, Zhang L, Jiang C, Z, et al., 2018, WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. *IEEE Transactions on Mobile Computing*, 18(11): 2714–2724. <https://www.doi.org/10.1109/TMC.2018.2878233>
- [5] Wang D, Yang J, Cui W, et al., 2021, Multimodal CSI-Based Human Activity Recognition Using GANs. *IEEE Internet of Things Journal*, 8(24): 17345–17355. <https://www.doi.org/10.1109/JIOT.2021.3080401>
- [6] Palipana S, Agrawal P, Pesch D, 2016, Channel State Information Based Human Presence Detection Using Non-Linear Techniques. *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, 177–186. <https://www.doi.org/10.1145/2993422.2993579>
- [7] Zhou R, Lu X, Zhao P, et al., 2017, Device Free Presence Detection and Localization with SVM and CSI Fingerprinting. *IEEE Sensors Journal*, 17(23): 7990–7999. <https://www.doi.org/10.1109/JSEN.2017.2762428>

- [8] Wu K, Xiao J, Yi Y, et al., 2012, CSI-Based Indoor Localization. *IEEE Transactions on Parallel and Distributed Systems*, 24(7): 1300–1309. <https://www.doi.org/10.1109/TPDS.2012.214>
- [9] Wang X, Gao L, Mao S, et al., 2016, CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach, *IEEE Transactions on Vehicular Technology*, 66(1): 763–776. <https://www.doi.org/10.1109/TVT.2016.2545523>
- [10] Li Q, Liao X, Liu M, et al., 2021, Indoor Localization Based on CSI Fingerprint by Siamese Convolution Neural Network. *IEEE Transactions on Vehicular Technology*, 70(11): 12168–12173. <https://www.doi.org/10.1109/TVT.2021.3107936>
- [11] Zheng Y, Zhang Y, Qian K, et al., 2019, Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 313–325. <https://www.doi.org/10.1145/3307334.3326081>
- [12] Zhang J, Tang Z, Li M, et al., 2018, CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 305–320. <https://www.doi.org/10.1145/3241539.3241570>
- [13] Zhang Y, Sun P, Jiang Y, et al., 2021, Bytetrack: Multi-Object Tracking by Associating Every Detection Box, arXiv.
- [14] Gringoli F, Schulz M, Link J, et al., 2019, Free your CSI: A Channel State Information Extraction Platform for Modern Wi-Fi Chipsets. *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 21–28. <https://www.doi.org/10.1145/3349623.3355477>
- [15] Ma Y, Zhou G, Wang S, 2019, Wi-Fi Sensing with Channel State Information: A Survey. *ACM Computing Surveys (CSUR)*, 52(3): 1–36. <https://www.doi.org/10.1145/3310194>

**Publisher's note**

*Art & Technology Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.*