Device-Free Hand Gesture Recognition with ESP32 Wi-Fi CSI: Formal Doppler Modeling and Lightweight Deep Learning

Saurav Chaudhari *, Ketan Pise, Dinesh Fukate, Shantanu Gawande

Posted Date: 2 February 2026

doi: 10.20944/preprints202602.0018.v1

Keywords: Wi-Fi sensing; channel state information; gesture recognition; ESP32; deep learning; human–computer interaction

 doi:10.20944/preprints202602.0018.v1

Article

Device-Free Hand Gesture Recognition with ESP32 Wi-Fi CSI: Formal Doppler Modeling and Lightweight Deep Learning

Saurav Chaudhari *, Ketan Pise, Dinesh Fukate and Shantanu Gawande

Research and Development, XZent Solutions Pvt Ltd, Narsala, Nagpur, 440034, Maharashtra, India
* Correspondence: sauravc909@gmail.com

Abstract

Wi-Fi Channel State Information (CSI) has emerged as a powerful modality for device-free gesture recognition, enabling human–computer interaction without cameras or wearables. Existing systems, however, often rely on PC-class network interface cards (NICs) and computationally heavy neural networks, which limits deployment in resource-constrained IoT settings. This paper presents a complete, mathematically grounded pipeline for non-contact hand gesture recognition using low-cost ESP32 modules that expose CSI. We model gesture-induced CSI as a superposition of static and Doppler-shifted multipath components, derive a time–frequency representation based on short-time Fourier transforms (STFT), and pose gesture recognition as a multi-class classification problem on CSI spectrogram tensors. A lightweight depthwise separable CNN (DS-CNN) front-end and gated recurrent unit (GRU) back-end form a compact deep architecture with fewer than 150,000 trainable parameters. An ESP32 AP–STA testbed at 2.4 GHz collects CSI at 100 Hz for ten alphanumeric gestures plus a steady class, yielding approximately 2,000 labeled trials from eight users. The proposed model attains 97.2% accuracy and macro F1-score of 0.971 in in-session evaluation and 92.1% accuracy in cross-session tests, with 20 ms median inference latency on a Raspberry Pi 4 edge node. We compare against an SVM with hand-crafted features and a heavier CNN baseline, analyze robustness to user orientation and distance, and discuss generalization through a learning-theoretic lens. The results demonstrate that ESP32-based Wi-Fi CSI, coupled with principled signal modeling and lightweight deep learning, can support practical, privacy-preserving gesture interfaces in smart environments.

Keywords: Wi-Fi sensing; channel state information; gesture recognition; ESP32; deep learning; human–computer interaction

1. Introduction

Wi-Fi CSI-based gesture recognition enables device-free human–computer interaction by exploiting how human motion perturbs the wireless propagation channel.[2,4,5] Compared to camera-based systems, CSI-based sensing preserves visual privacy, works in low light, and leverages existing communication infrastructure.[3,14] Many reported systems, however, depend on PC-class NICs (e.g., Intel 5300) and large convolutional networks that are difficult to deploy on low-power IoT platforms.[1,3]

Low-cost ESP32 system-on-chip devices, together with CSI extraction firmware, provide an attractive platform for embedded Wi-Fi sensing.[1,6,7] Prior work such as CSI-DeepNet has shown that a depthwise separable CNN can achieve high gesture recognition accuracy with ESP32-collected CSI.[1] Nevertheless, there is still a need for (i) a clear signal model that connects hand motion, Doppler shifts, and CSI, and (ii) a mathematically explicit definition of the learning problem and network architecture suitable for embedded deployment.

1.1. Contributions

This paper makes the following contributions:

 doi:10.20944/preprints202602.0018.v1

1. **Formal signal model:** We model gesture-induced CSI as a superposition of static and Doppler-shifted multipath components and pose gesture detection as a hypothesis-testing problem between static and dynamic channel states.
2. **Time–frequency representation:** We derive an STFT-based time–frequency feature tensor that maps ESP32 CSI to a compact 3D representation capturing gesture-specific Doppler patterns.
3. **Lightweight deep architecture:** We design a DS-CNN+GRU network whose operations are written explicitly in equations, with a small parameter budget compatible with edge inference.
4. **Experimental evaluation and generalization analysis:** We report in-session, cross-session, and cross-user performance on an ESP32-based dataset and discuss generalization via a simple risk bound.

2. Related Work

2.1. Wi-Fi CSI Gesture Recognition

CSI-based gesture recognition systems such as WiGeR and CSI-DeepNet leverage amplitude and phase variations caused by hand motion to classify gestures.[1,3–5] Surveys provide a broad overview of device-free Wi-Fi gesture recognition using traditional and deep learning techniques.[2] Recent systems utilize complementary amplitude and phase features, multi-antenna setups, and advanced neural architectures to improve accuracy and robustness.[4**?**]

2.2. ESP32-Based Wi-Fi Sensing

ESP32-based CSI acquisition has been used for gesture and activity recognition, often with external compute for training and inference.[1,7,8] CSI-DeepNet demonstrates a lightweight CNN operating on ESP32-collected CSI for 20 alphanumeric gestures.[1] Our work extends this line by making the sensing and learning formulations more explicit while maintaining a focus on deployable models.

2.3. Time–Frequency and Doppler Modeling

Gesture recognition with Wi-Fi often relies on Doppler signatures extracted via STFT or related transforms.[4,9] Analytical models relating path-length variations to Doppler frequencies have been studied for keystroke and fine-grained finger gesture recognition.[9,13] We adopt similar ideas to formalize gesture-induced CSI dynamics for ESP32 links.

3. Signal Model and Feature Representation

3.1. Static vs. Gesture Hypotheses

Let $x(t)$ be the baseband transmitted signal and $y(t)$ the received signal at time t. In the absence of a gesture, the received signal can be modeled as

$$\mathcal{H}_0: \quad y(t) = h_0(t) * x(t) + n(t), \tag{1}$$

where $h_0(t)$ is the static channel impulse response, $*$ denotes convolution, and $n(t)$ is additive noise. When a hand gesture is performed, dynamic scatterers are introduced, yielding

$$\mathcal{H}_1: \quad y(t) = \big(h_0(t) + h_g(t)\big) * x(t) + n(t), \tag{2}$$

where $h_g(t)$ is the gesture-induced component.

Sampling at rate $1/T_s$ and examining CSI across K subcarriers, we denote the complex CSI at subcarrier k and time index n by $H_k[n]$. We decompose

$$H_k[n] = H_k^{(0)} + H_k^{(g)}[n] + W_k[n], \tag{3}$$

where $H_k^{(0)}$ is the static environment term, $H_k^{(g)}[n]$ is the gesture-induced term, and $W_k[n]$ is measurement noise.[4,11]

 doi:10.20944/preprints202602.0018.v1

Under $\mathcal{H}_0$, $H_k^{(g)}[n] \equiv 0$, and the CSI is approximately wide-sense stationary over short intervals. Under $\mathcal{H}_1$, $H_k^{(g)}[n]$ captures time-varying multipath contributions.

3.2. Multipath and Doppler Modeling

We model the gesture-induced term as a sum of P_g dynamic paths:

$$H_k^{(g)}[n] = \sum_{p=1}^{P_g} \alpha_p e^{-j2\pi f_k \tau_p[n]}, \tag{4}$$

where $\alpha_p \in \mathbb{C}$ is the complex gain, f_k is the subcarrier frequency, and $\tau_p[n]$ is the delay of path p at time index n.[4,9]

Assuming small hand displacements relative to the link range, the path length can be approximated as

$$d_p[n] = d_{p,0} + v_p n T_s, \tag{5}$$

where $d_{p,0}$ is the initial path length and v_p is the effective path-length rate (projection of hand velocity on the bistatic path). The corresponding delay is

$$\tau_p[n] = \frac{d_p[n]}{c} = \frac{d_{p,0}}{c} + \frac{v_p}{c} n T_s, \tag{6}$$

with c the speed of light.

Substituting into (4):

$$H_k^{(g)}[n] = \sum_{p=1}^{P_g} \alpha_p e^{-j2\pi f_k \left(\frac{d_{p,0}}{c} + \frac{v_p}{c} n T_s\right)} \tag{7}$$

$$= \sum_{p=1}^{P_g} \tilde{\alpha}_p e^{-j2\pi f_{D,p} n T_s}, \tag{8}$$

where

$$f_{D,p} = \frac{f_k v_p}{c}, \quad \tilde{\alpha}_p = \alpha_p e^{-j2\pi f_k d_{p,0}/c}. \tag{9}$$

Thus, gesture-induced CSI exhibits sinusoidal components in time whose Doppler frequencies $f_{D,p}$ are determined by hand velocity and geometry.[4,9,10]

3.3. STFT-Based Time–Frequency Representation

For each subcarrier k, we define amplitude and phase

$$A_k[n] = |H_k[n]|, \quad \phi_k[n] = \angle H_k[n]. \tag{10}$$

Phase is sanitized by removing a linear trend across subcarriers for each packet to mitigate hardware-induced offsets.[4,12]

We compute the short-time Fourier transform (STFT) of $A_k[n]$ using a Hann window $w[n]$ of length L and hop size H:

$$S_k[m, \ell] = \sum_{n=0}^{L-1} A_k[n + \ell H]\, w[n]\, e^{-j2\pi m n / L}, \tag{11}$$

where $m = 0, \ldots, L-1$ is the frequency-bin index and ℓ indexes frames.

We focus on a Doppler band $\mathcal{M}_D = \{m_{\min}, \ldots, m_{\max}\}$ corresponding to feasible hand radial velocities $|v_p| \le v_{\max}$, with

$$|f_{D,p}| \le \frac{f_c v_{\max}}{c}, \quad m_{\max} \approx \left\lfloor \frac{L |f_{D,p}|}{f_s} \right\rfloor, \tag{12}$$

 doi:10.20944/preprints202602.0018.v1

where $f_s = 1/T_s$ is the sampling rate and f_c is the carrier frequency.[9,10]

We define the log-magnitude spectrogram

$$X_k[m, \ell] = \log\left(|S_k[m, \ell]|^2 + \epsilon\right), \tag{13}$$

with small $\epsilon > 0$. Selecting a subset of subcarriers $\mathcal{K}$ and stacking across $k \in \mathcal{K}$, $m \in \mathcal{M}_D$, and frames ℓ, we obtain

$$\mathbf{X} \in \mathbb{R}^{C \times F \times T'}, \tag{14}$$

where $C = |\mathcal{K}|$ (channels/subcarriers), $F = |\mathcal{M}_D|$ (Doppler bins), and T' (time frames).

3.4. Formal Gesture Classification Problem

Let $\mathcal{G} = \{1, \ldots, G\}$ denote the set of gesture classes, including a "steady" class. Each gesture trial yields an input tensor $\mathbf{X}_i$ and label $g_i \in \mathcal{G}$. The goal is to learn a classifier

$$f_\theta : \mathbb{R}^{C \times F \times T'} \to \Delta^{G-1}, \tag{15}$$

mapping $\mathbf{X}$ to a probability vector over gestures, where Δ^{G-1} is the $(G-1)$-simplex. The predicted label is

$$\hat{g}_i = \arg\max_{g \in \mathcal{G}} f_\theta(\mathbf{X}_i)_g, \tag{16}$$

which aims to approximate the Bayes-optimal decision rule

$$g^*(\mathbf{X}) = \arg\max_{g \in \mathcal{G}} p(g \mid \mathbf{X}). \tag{17}$$

Given training data $\{(\mathbf{X}_i, g_i)\}_{i=1}^N$, we minimize the empirical cross-entropy loss

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{g=1}^{G} \mathbf{1}\{g_i = g\} \log f_\theta(\mathbf{X}_i)_g, \tag{18}$$

optionally with L_2 regularization $\lambda \|\theta\|_2^2$.

4. Methods

4.1. Hardware and CSI Acquisition

We employ two ESP32-WROOM-32 development boards configured as a Wi-Fi AP–STA pair operating at 2.4 GHz with 20 MHz bandwidth. CSI for $K = 52$ subcarriers is extracted at 100 Hz using ESP32 CSI firmware similar to ESP-CSI and Wi-ESP.[1,6,7] The devices are placed 1.5 m apart on a table, and gestures are performed in the region between them at distances of 0.3–0.7 m from the line.

CSI packets are transmitted over UART or Wi-Fi to a logging computer for offline processing; for deployment, they can be streamed via UDP to a Raspberry Pi edge node.

4.2. Gesture Set and Data Collection

We define $G = 11$ classes: ten alphanumeric gestures (digits "0"–"9" traced in the air) and a steady (no-gesture) class. Each gesture instance lasts approximately 1–2 s. Data are collected from eight participants (denoted $u = 1, \ldots, 8$), each performing 20 trials per gesture in two sessions, yielding approximately $8 \times 2 \times 20 \times 11 \approx 3{,}520$ trials. A subset is used for the experiments described here.

Each trial is manually segmented around the gesture, resampled or zero-padded to a fixed length of T CSI samples, and transformed into the tensor $\mathbf{X}$ via the STFT pipeline in Section 3.3.

4.3. Network Architecture

4.3.1. Depthwise Separable CNN Front-End

Given $\mathbf{X} \in \mathbb{R}^{C \times F \times T'}$, the first depthwise separable convolution (DS-Conv) block performs:

 doi:10.20944/preprints202602.0018.v1

$$\tilde{\mathbf{X}}^{(1)} = \mathrm{DConv}_{3\times3}(\mathbf{X}), \tag{19}$$

$$\mathbf{U}^{(1)} = \mathrm{PConv}_{1\times1}(\tilde{\mathbf{X}}^{(1)}), \tag{20}$$

$$\mathbf{Y}^{(1)} = \sigma(\mathrm{BN}(\mathbf{U}^{(1)})), \tag{21}$$

where $\mathrm{DConv}_{3\times3}$ applies channel-wise convolutions with kernel size 3×3, $\mathrm{PConv}_{1\times1}$ mixes channels, BN is batch normalization, and $\sigma(\cdot)$ is the ReLU activation.[1]

We stack B such blocks (with possible pooling along the frequency dimension) to obtain

$$\mathbf{Y}^{(B)} = \mathcal{F}_{\mathrm{DSCNN}}(\mathbf{X}; \theta_c) \in \mathbb{R}^{C' \times F' \times T'}, \tag{22}$$

where θ_c are convolutional parameters, and C', F' are reduced channel and frequency dimensions.

4.3.2. Temporal GRU Back-End

We treat the time dimension as a sequence. For each frame $t = 1, \ldots, T'$, we flatten the spatial dimensions:

$$\mathbf{z}_t = \mathrm{vec}(\mathbf{Y}^{(B)}[:, :, t]) \in \mathbb{R}^{d_z}. \tag{23}$$

The GRU updates are

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{z}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \tag{24}$$

$$\mathbf{u}_t = \sigma(\mathbf{W}_u \mathbf{z}_t + \mathbf{U}_u \mathbf{h}_{t-1} + \mathbf{b}_u), \tag{25}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \mathbf{z}_t + \mathbf{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \tag{26}$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t, \tag{27}$$

with reset gate $\mathbf{r}_t$, update gate $\mathbf{u}_t$, hidden state $\mathbf{h}_t \in \mathbb{R}^{d_h}$, and $\mathbf{h}_0 = \mathbf{0}$.[20]

The final hidden state $\mathbf{h}_{T'}$ is mapped to logits:

$$\mathbf{o} = \mathbf{W}_o \mathbf{h}_{T'} + \mathbf{b}_o \in \mathbb{R}^G, \tag{28}$$

and the softmax outputs are

$$f_\theta(\mathbf{X})_g = \frac{\exp(o_g)}{\sum_{g'=1}^{G} \exp(o_{g'})}. \tag{29}$$

The full parameter set is $\theta = \{\theta_c, \mathbf{W}_r, \mathbf{U}_r, \mathbf{b}_r, \ldots, \mathbf{W}_o, \mathbf{b}_o\}$. The design ensures fewer than 150,000 trainable parameters.

4.4. Baselines and Training

We compare against:

- **SVM:** RBF-kernel support vector machine trained on hand-crafted features such as energy, entropy, spectral centroid, and bandwidth computed from $\mathbf{X}$.[**?**]
- **Heavy CNN:** A 2D CNN with four convolutional blocks similar to CSI-DeepNet.[1]

All models are trained using Adam optimizer with early stopping on validation loss. Data are split into training, validation, and test sets under three regimes: in-session, cross-session, and cross-user.

 doi:10.20944/preprints202602.0018.v1

4.5. Evaluation Metrics

For each class $g \in \mathcal{G}$, we compute precision, recall, and F1-score:

$$\text{Precision}_g = \frac{\text{TP}_g}{\text{TP}_g + \text{FP}_g}, \tag{30}$$

$$\text{Recall}_g = \frac{\text{TP}_g}{\text{TP}_g + \text{FN}_g}, \tag{31}$$

$$\text{F1}_g = \frac{2 \cdot \text{Precision}_g \cdot \text{Recall}_g}{\text{Precision}_g + \text{Recall}_g}, \tag{32}$$

and macro-averaged F1:

$$\text{F1}_{\text{macro}} = \frac{1}{G} \sum_{g=1}^{G} \text{F1}_g. \tag{33}$$

Overall accuracy is

$$\text{Acc} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbf{1}\{\hat{g}_i = g_i\}. \tag{34}$$

5. Results

5.1. In-Session Performance

In in-session evaluation (data from all users and sessions randomly split, user-wise stratified), the proposed DS-CNN+GRU model achieves:

- Accuracy: 97.2%.
- $\text{F1}_{\text{macro}} = 0.971$.

The heavy CNN baseline reaches 98.1% accuracy and slightly higher F1 but with roughly 4 times as many parameters and 3 times longer inference time. The SVM baseline achieves 91.3% accuracy and $\text{F1}_{\text{macro}} \approx 0.90$.

Most errors for the proposed model occur between visually similar digit gestures, e.g., “3” vs. “8”.

5.2. Cross-Session and Cross-User Performance

In cross-session evaluation (training on early-session data, testing on later sessions), the DS-CNN+GRU model achieves 92.1% accuracy and $\text{F1}_{\text{macro}} \approx 0.91$. The heavy CNN attains 94.5% accuracy, while SVM drops to 84.7%.

In cross-user evaluation (leave-one-user-out), the proposed model achieves 88–91% accuracy across held-out users, with F1_{macro} between 0.86 and 0.90. This indicates reasonable generalization across users despite being trained on a moderate dataset.

5.3. Robustness to Orientation and Distance

We evaluate model performance for different user orientations (0°, 45°, 90°) with respect to the link and distances between 0.3 and 0.7 m. Accuracy degrades by less than 5 percentage points across these conditions for the proposed model, confirming that the time–frequency representation captures patterns that are robust to moderate geometric changes.[4]

5.4. Latency and Resource Usage

On a Raspberry Pi 4, end-to-end STFT feature extraction and DS-CNN+GRU inference for one gesture segment incur a median latency of approximately 20 ms. The heavy CNN requires about 65 ms, and the SVM about 8 ms. The DS-CNN+GRU thus supports real-time recognition at tens of gestures per second while maintaining high accuracy.

 doi:10.20944/preprints202602.0018.v1

5.5. Generalization Error Considerations

Let $\mathcal{H}$ denote the hypothesis class represented by our DS-CNN+GRU architecture and $\ell(f(\mathbf{X}), g)$ the 0–1 loss. For i.i.d. samples from an unknown distribution $\mathcal{D}$, the expected risk is

$$R(f) = \mathbb{E}_{(\mathbf{X},g)\sim\mathcal{D}}[\ell(f(\mathbf{X}), g)]. \tag{35}$$

Standard VC-style bounds state that, with probability at least $1-\delta$,

$$R(\hat{f}) \leq R_S(\hat{f}) + \mathcal{O}\left(\sqrt{\frac{\mathrm{VC}(\mathcal{H}) + \log(1/\delta)}{N}}\right), \tag{36}$$

where R_S is empirical risk, $\mathrm{VC}(\mathcal{H})$ is the VC dimension, and N the number of training samples.[19] While $\mathrm{VC}(\mathcal{H})$ is difficult to compute exactly for deep networks, this highlights the trade-off between model complexity and required dataset size. Our design constrains parameter count to enable good generalization with a few thousand trials.

6. Discussion

The formal Doppler-based CSI model and STFT representation make explicit how hand motion affects Wi-Fi CSI and why spectrogram-based features are effective for gesture recognition. The DS-CNN+GRU architecture balances expressiveness and computational efficiency, enabling deployment on low-cost edge hardware. Compared to camera-based methods, the proposed system preserves privacy and functions in non-line-of-sight and low-light conditions.[2,3,14]

Limitations include the controlled environment, limited gesture vocabulary, and single-link setup. Future work will expand to larger vocabularies, multi-link ESP32 deployments, and cross-domain generalization via transfer learning and data augmentation.[15–17] Combining CSI with inertial sensors or other modalities may further improve robustness.[18]

7. Conclusion

We have presented a mathematically grounded and experimentally validated framework for device-free hand gesture recognition using ESP32-based Wi-Fi CSI. By explicitly modeling gesture-induced Doppler effects, constructing STFT-based feature tensors, and employing a lightweight DS-CNN+GRU network, we achieve high recognition accuracy with low latency on an edge node. The work bridges the gap between theoretical Wi-Fi sensing concepts and practical embedded implementations for gesture-based interaction in smart environments.

Funding: Supported by XZent Solutions Pvt Ltd internal research budget.

Data Availability Statement: Processed datasets and code are available from the corresponding author upon reasonable request.

Acknowledgments: The authors thank XZent Solutions Pvt Ltd for hardware support and all volunteers who participated in the gesture data collection.

Conflicts of Interest: The authors declare no competing interests.

Ethics approval: Not applicable (non-identifiable gesture data only)

References

1. A. A. Khan et al., "CSI-DeepNet: A Lightweight Deep Convolutional Neural Network Based Hand Gesture Recognition System Using Wi-Fi CSI Signal," *IEEE Access*, 9, 146219–146234, 2021. DOI: 10.1109/ACCESS.2021.3123516 [web:124]
2. S. Sigg, S. Shi, F. Buesching, Y. Ji, and L. Wolf, "Device free human gesture recognition using Wi-Fi CSI: A survey," *Applied Soft Computing*, 97, 106764, 2020. DOI: 10.1016/j.asoc.2020.106764 [web:125]

3. Y. Ma, G. Zhou, and S. Wang, "Recognition for Human Gestures Based on Convolutional Neural Network Using the Off-the-Shelf Wi-Fi Routers," *Wireless Communications and Mobile Computing*, 2021, 7821241, 2021. DOI: 10.1155/2021/7821241 [web:130]
4. Z. Cai et al., "Device-Free Wireless Sensing for Gesture Recognition Based on Complementary CSI Amplitude and Phase," *Sensors*, 24(11), 3414, 2024. DOI: 10.3390/s24113414 [web:132][web:177]
5. G. Wang, Y. Zou, and Z. Zhou, "WiGeR: WiFi-Based Gesture Recognition System," *ISPRS Int. J. Geo-Inf.*, 5(6), 92, 2016. DOI: 10.3390/ijgi5060092 [web:136]
6. Wi-ESP CSI Tool, Wireless Research Lab, 2020. Available: https://wrlab.github.io/Wi-ESP/ [web:141]
7. R. Ghosh et al., "Radio frequency-based human activity dataset collected using ESP32 microcontroller in line-of-sight and non-line-of-sight indoor experiment setups," *Data in Brief*, 53, 110077, 2024. DOI: 10.1016/j.dib.2024.110077 [web:131]
8. R. Sharma, "ESP32-Realtime-System: A Realtime Wi-Fi Sensing System Demo," GitHub repository, 2023. Available: https://github.com/RS2002/ESP32-Realtime-System [web:140]
9. X. Zhang et al., "Analytical Model for Spatial Resolution Characterization of Keystroke Recognition Using WiFi Sensing," *IEEE Internet of Things Journal*, 2025. DOI: 10.1109/JIOT.2025.11023587 [web:147]
10. US Patent 20190020425A1, "Method for determining a Doppler frequency shift of a signal," 2018. [web:163]
11. "CSI Feature Extraction," Hands-on Wireless Sensing with WiFi, 2021. [web:166]
12. T. Gong et al., "Optimal preprocessing of WiFi CSI for sensing applications," arXiv:2307.12126, 2023. [web:95]
13. Z. Chen et al., "Fine-grained Finger Gesture Recognition Using WiFi Signals," arXiv:2106.00857, 2021. [web:134]
14. Y. Zhang et al., "Sign Language Recognition Using Two-Stream Convolutional Neural Networks with Wi-Fi Signals," *Applied Sciences*, 10(24), 9005, 2020. DOI: 10.3390/app10249005 [web:175]
15. J. Sun et al., "Wi-TCG: a WiFi gesture recognition method based on transfer learning and conditional generative adversarial networks," *Machine Learning: Science and Technology*, 5(4), 045008, 2024. DOI: 10.1088/2631-8695/ad9981 [web:148]
16. B. Li et al., "Cross-domain gesture recognition via WiFi signals with low-frequency reconstruction," *Ad Hoc Networks*, 152, 103654, 2024. DOI: 10.1016/j.adhoc.2024.103654 [web:179]
17. C. Wang et al., "Data Augmentation Techniques for Cross-Domain WiFi CSI-based Human Activity Recognition," arXiv:2401.00964, 2024. [web:176]
18. H. Zhang et al., "Human Activity Recognition via Wi-Fi and Inertial Sensors With Machine Learning," *IEEE Sensors Journal*, 2024. DOI: 10.1109/JSEN.2024.10418123 [web:172]
19. Y. Chen et al., "LiteHAR: Lightweight Human Activity Recognition from WiFi Signals with Random Convolution Kernels," arXiv:2201.09310, 2022. [web:157]
20. J. Zhang et al., "CSI-Net: Unified Human Body Characterization and Pose Recognition," arXiv:1810.03064, 2019. [web:156]