Tian *et al. J Wireless Com Network* (2023) 2023:79
https://doi.org/10.1186/s13638-023-02289-6

EURASIP Journal on Wireless
Communications and Networking

RESEARCH **Open Access**

Texture features-based lightweight passive multi-state crowd counting algorithm

Yong Tian[1*], Ying Li[1], Jiaxin Li[1], Fangting Gao[1], Chuanzhen Zhuang[1] and Xuejun Ding[2*]

*Correspondence:
tianyong@lnnu.edu.cn;
dingxuejun@dufe.edu.cn

[1] School of Physics and Electronic Technology, Liaoning Normal University, Dalian 116029, China
[2] School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China

Abstract

Passive crowd counting using channel state information (CSI) is a promising technology for applications in fields such as smart cities and commerce. However, the most existing algorithms can only recognize the total number of people in the monitoring area and cannot simultaneously recognize the number and states of people and ignore the real-time performance of the algorithms. Therefore, they cannot be applied to the scenarios of multi-state crowd counting requiring high real-time performance. To address this issue, a lightweight passive multi-state crowd counting algorithm called TF-LPMCC is proposed. This algorithm constructs CSI amplitude data into amplitude and time–frequency images, extracts texture features using the gray-level co-occurrence matrix (GLCM) and gray-level difference statistic (GLDS) methods, and uses the linear discriminant analysis (LDA) algorithm to count the crowd in multi-states. Experiments show that the TF-LPMCC algorithm not only has low time complexity but also achieves an average recognition accuracy of 98.27% for crowd counting.

Keywords: Crowd counting, Multi-state, Channel state information, Texture feature

1 Introduction

With the advancement of science and technology, human–computer interactions, including indoor crowd counting, indoor localization, and activity recognition, have become a new trend in the development of intelligent society. Crowd counting is the process of determining the number of people in a specific environment. As the urban population grows, various problems arise, such as the unreasonable allocation of public resources and declining service quality [1]. Consequently, the application requirements for crowd counting are also increasing. By utilizing effective crowd counting schemes, relevant departments or enterprises can obtain real-time information on the number of people in a specific area, thereby allocating public resources more reasonably, reducing resource waste, and improving service quality [2, 3]. For example, by counting the number of people applying for different businesses, more staff can be allocated to the business departments with larger queues. By counting the number of people in different exhibition halls of museums or exhibition centers, the manager can provide more air conditioning for the exhibition halls with larger number of people. By counting the number of people near different shelves in a supermarket, managers can place products with higher attention levels closer to the entrance or more easily visible to customers [4]. Therefore, crowd counting research has become an important research field in human–computer interaction, and it is of great significance for achieving smart cities, architectures, commerce, and homes.

Currently, crowd counting algorithms are categorized into four main categories: based on the video, based on the special sensors, based on the received signal strength (RSS), and based on the channel state information (CSI). Video-based crowd counting algorithms acquire data through cameras, extract features, and perform crowd counting using machine learning algorithms. For instance, Wu et al. [5] proposed a video-based spatial–temporal graph network that fuses multi-scale features from both temporal and spatial perspectives to achieve efficient crowd counting in videos. The method is mature and has high accuracy. However, it is unsuitable for no-line-of-sight environments or environments with smoke, may breach privacy, and has high deployment costs. Special sensor-based crowd counting methods use RFID, infrared sensors, and other technologies to obtain data for crowd counting. For example, Ding et al. [6] proposed a system called R# that estimates the number of people using passive RFID tags. These methods have good environmental adaptability and high accuracy but are costly and not suitable for mass application. RSS-based crowd counting methods count the number of people using the obtained RSS. For example, Denis et al. [7] designed and tested their crowd estimation systems which are wireless sensor networks using RSS information to estimate visitors. These methods have high device popularity, non-line-of-sight, and protect privacy but are susceptible to environmental factors that cause unstable RSS. With the popularity of commercial WiFi, CSI-based crowd counting methods have become the research focus of scholars. CSI is a fine-grained physical layer information that enables passive sensing using amplitude and phase information [8]. The advantages of this method are that no additional equipment needs to be deployed, it is not affected by light and occlusion, the signals are stable, and it can protect privacy.

CSI-based wireless sensing technology has been developed for over a decade, and numerous CSI-based crowd counting algorithms have emerged [2–4, 9–16]. In 2014, Xi et al. [9] proposed the Electronic Frog Eye system, the first to use CSI information for crowd counting. The system utilized gray theory for crowd prediction and proposed the dilatation-based crowed profiling algorithm, which was based on the positive correlation between the change of CSI and the number of people. Since then, numerous CSI-based crowd counting research papers have been published. In 2018, Zou et al. [10] proposed an indoor crowd counting system that achieved 96% recognition accuracy using a feature selection scheme based on information theory. However, the system had a high learning cost. Liu et al. [11] proposed the WiCount system in 2017, the first to use a neural network for crowd counting with an accuracy of 82.3%. They designed an online learning mechanism [12] to determine whether someone enters/leaves the room by using an activity recognition model, fine-tuning the deep learning model with an average accuracy of 87% for up to 5 people. Ma et al. [13] proposed a device-free crowd density estimation system called Wisual, which predicted crowd density with 98% accuracy and accurately displayed the spectrum of mobile people based on CSI. Zhang et al. [2] proposed a queueing crowd counting system based on CSI and deep learning networks, called Quee-Fi system, which used a static model based on fully connected neural networks with convolutional long short-term memory for queueing crowd counting. Zhang et al. [3] proposed a WiFi-based cross-environment crowd counting system with the ability to estimate walking directions and perform crowd counting over only one link, called WiCrowd. Liu et al. [4] proposed a CSI-based device-free crowd counting scheme, which utilized the intuition that different numbers of people wandering in the environment would have different effects on WiFi signals. The scheme achieved an experimental accuracy of 87.2%. Guo et al. [14] proposed a wall-piercing crowd counting system using ambient WiFi signals, called TWCC, which took the phase difference data of channel state information (CSI) and fed it into a BP neural network after preprocessing, with an average recognition accuracy of 90%. Alizadeh et al. [15] proposed a HARC algorithm that simultaneously recognized human activity and counted the number of people at bus stops. The algorithm used a LSTM-RNN model as a classifier with 94% recognition accuracy. Choi et al. [16] proposed a simultaneous recognition system for headcount and localization using CSI and machine learning, achieving a counting error of 0.35 MAE (89.8% of 1-person internal error) and localization accuracy of 91.4%.

After analyzing the existing CSI-based crowd counting algorithms, we discovered that most previous studies focused on counting crowds with the same activity state, such as stationary or walking in sequence (known as single-state crowd counting in this paper). However, in real-world scenarios, crowds can exhibit different states, such as stationary, walking in sequence, raising one hand, or running. Some applications not only need to count the number of people in the monitoring area, but also need to recognize the activity states of the crowd. For example, in fitness venues, in order to count the number of people who are exercising and the items that people are training, the total number of people in the venues and the number of people for each training item need to be recognized, so that the managers can adjust fitness equipment and improve business strategies. In nursing homes or kindergartens, the managers need to master the number and activity state data of the elderly or children to understand their living habits and provide better services. This type of application requires simultaneous recognition of multiple activity states and a total number of people (referred to as multi-state crowd counting in this paper), which is an important problem faced by CSI-based crowd counting algorithms. Additionally, previous studies ignored the real-time performance of algorithms and often used algorithms with high time complexity to recognize the number of people, such as deep learning algorithms, which made them not suitable for applications requiring high real-time performance. This is another important problem currently faced by CSI-based crowd counting algorithms. To address the two problems, we propose a texture features-based lightweight passive multi-state crowd counting algorithm, referred to as TF-LPMCC, which can recognize both the number and activity state of volunteers at the same time by utilizing texture features of CSI images. The specific contributions of this paper are summarized as follows:

(1) The existing research results have shown that the temporal stability of CSI can ensure capturing abnormal entities and their activities that cause environmental changes, and the frequency diversity of CSI can reflect the multipath reflection of wireless signals [17]. Therefore, in order to ensure the high recognition accuracy of multi-state crowd counting algorithms, we construct CSI amplitude data into the

form of amplitude images (utilizing the temporal stability of CSI) and time–frequency images (utilizing the frequency diversity of CSI). Then, two texture analysis methods of digital images, the gray-level co-occurrence matrix (GLCM) method and the gray-level difference statistics (GLDS) method, are used to extract the features of two types of images, which can characterize the local changes and spatial distribution of image pixels. The extracted features form feature vectors for recognizing the states and quantity of crowds. This above novel method extracts the features of CSI amplitude changes caused by different numbers and states of people from both time-domain and frequency-domain aspects, thereby achieving high-precision and more complex multi-state crowd counting.

(2) To reduce the time complexity of the algorithm, we construct CSI amplitude data of multiple subcarriers into CSI amplitude images and time–frequency images with matrix form, and extract CSI data features using the faster matrix operations. In addition, the linear discriminant analysis (LDA) algorithm with lower time complexity is used to recognize the state and number of people.

(3) Numerous experiment results demonstrate that, compared with the other two state-of-the-art algorithms, the proposed TF-LPMCC algorithm achieved an average recognition accuracy of 98.27%, which increased by 4.04% and 4.42%, respectively. The running time was 0.068 s, which decreased by 46.88% and 65.48%, respectively.

The remaining sections of this paper are structured as follows: Sect. 2 details the TF-LPMCC algorithm; Sect. 3 gives the experimental results, and Sect. 4 discusses the limitations of the algorithm; Sect. 5 concludes the paper and presents future work.

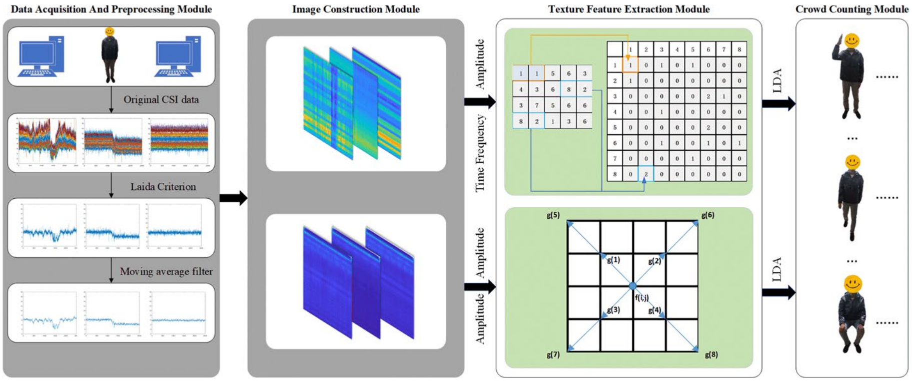
2 TF-LPMCC algorithm

2.1 Algorithm framework

The TF-LPMCC algorithm proposed in this paper consists of four primary modules: data acquisition and preprocessing, image construction, texture feature extraction, and crowd counting. In the data acquisition and preprocessing module, two computers equipped with Intel 5300 network cards are utilized as transceiver devices to collect and preprocess CSI data at the receiving end. The image construction module involves using the preprocessed CSI information to construct amplitude-subcarrier images (also known as amplitude images) and frequency-time images (also known as time–frequency images). In the texture feature extraction module, texture features are extracted from the amplitude images and time–frequency images using GLCM and GLDS methods. Finally, in the crowd counting module, the feature vectors obtained from the texture feature extraction module are input to the LDA algorithm for crowd counting. The framework of the TF-LPMCC algorithm proposed in this paper is shown in Fig. 1.

2.2 Data acquisition and preprocessing

The P_{t} and P_{r} antennas are connected to the wireless network card of the transmitting and the receiving device, respectively. This setup allows for simultaneous reception of data from P links ($P = P_{\mathrm{t}} \times P_{\mathrm{r}}$) at the receiving end. The measurement value matrix of CSI for the i-th data packet can be expressed as follows:



1	1	5	6	3
4	3	6	8	2
3	7	5	6	6
8	2	1	3	6

	1	2	3	4	5	6	7	8
1	1	0	1	0	1	0	0	0
2	1	0	0	0	0	0	0	0
3	0	0	0	0	0	2	1	0
4	0	0	1	0	0	0	0	0
5	0	0	0	0	0	2	0	0
6	0	0	1	0	0	1	0	1
7	0	0	0	0	1	0	0	0
8	0	2	0	0	0	0	0	0

Fig. 1 TF-LPMCC algorithm framework

$$h(i) = \begin{vmatrix} h_{1,1}(i) & \cdots & h_{1,q}(i) & \cdots & h_{1,Q}(i) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{p,1}(i) & \cdots & h_{p,q}(i) & \cdots & h_{p,Q}(i) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{P,1}(i) & \cdots & h_{P,q}(i) & \cdots & h_{P,Q}(i) \end{vmatrix}, \tag{1}$$

P and Q represent the total number of wireless links and subcarriers, respectively. The variable $h_{p,q}(i)$ is a complex number that includes amplitude and phase, representing a CSI value. Given a certain data packet sending rate, the CSI data of a sample can be expressed as follows:

$$\mathbf{H} = [h(1), \cdots, h(i), \cdots h(I)], \tag{2}$$

The matrix $\mathbf{H}$ is three-dimensional, with dimensions of $P \times Q \times I$. Here, I represents the total number of CSI data packets for a sample, which includes all the data received by the wireless network card within a fixed period.

To reduce the amount of data processed in this study and the time complexity of the algorithm, we employ a setup where one antenna transmits and one antenna receives, with the transmitting link containing 30 subcarriers. As a result, each experimental sample contains 30 sets of CSI data. The CSI phase information is typically affected by factors such as carrier frequency offset and sampling frequency offset, so it needs to be corrected before use, which increases the algorithm's time complexity. However, the CSI amplitude information is generally more stable [18] and is used for multi-state crowd counting in this paper. Due to the influence of transceiver hardware and the surrounding environment, each set of CSI data contains a significant amount of noise. Figure 2 illustrates a set of CSI amplitude data without any processing, showing that raw CSI amplitude data cannot be directly used for crowd counting. Therefore, we needed to preprocess the collected CSI amplitude data to maximize the accuracy and stability of multi-state crowd counting.

To handle outliers in the raw CSI amplitude data that deviate from the normal range, we use the PauTa Criterion to remove them [19]. This involves selecting a sliding window of data with a fixed length and calculating the mean and standard deviation of the data within the window. Then, we subtract the mean value from each data within the

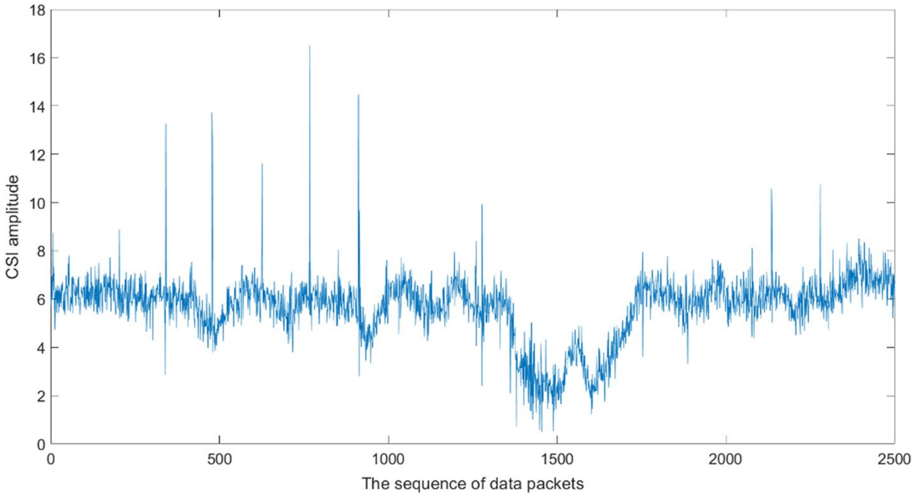


Fig. 2 A set of CSI amplitude data without any processing

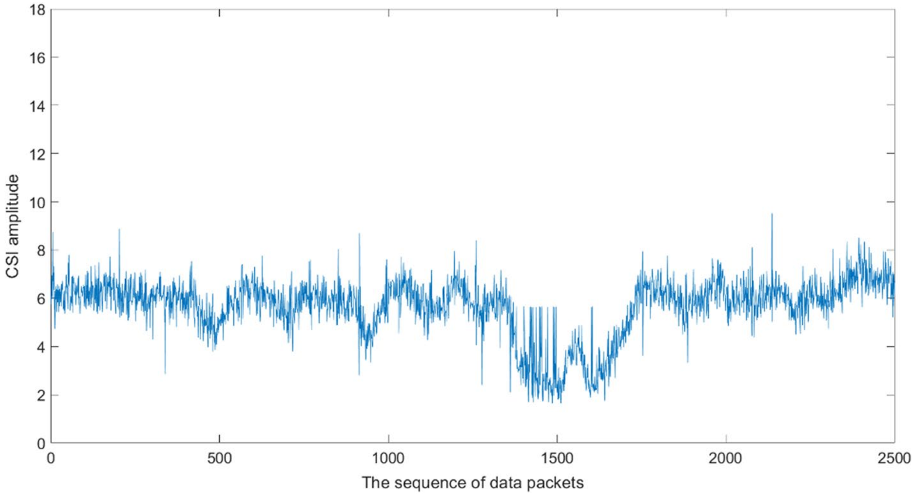


Fig. 3 The CSI amplitude data after removing outliers by using the PauTa Criterion

window. If the difference is greater than three times the standard deviation, we consider it an outlier and replace it with the mean value. This method was applied to the data in Fig. 2, and the results are shown in Fig. 3, which depicts the removal of outliers in Fig. 2.

The raw CSI amplitude data not only contains outliers affecting crowd counting, but also contains high-frequency noise caused by the surrounding environments and multipath effects. CSI amplitude changes caused by human bodies and their activities are mainly concentrated in the low-frequency part of CSI amplitude data, so CSI-based wireless sensing algorithms generally use low-pass filters to filter CSI amplitude data [20], such as moving average filter, Gaussian filter [8], and wavelet threshold method [21]. This paper focuses on reducing the time complexity of the algorithm, so we use the moving average filter with lower time complexity [22] to filter the high-frequency noise of CSI amplitude data. Moreover, this paper aims to use CSI amplitude for recognizing the number and states of people, which belongs to coarse-grained information recognition. Therefore, the TF-LPMCC algorithm using the moving average filter can already achieve high enough recognition accuracy for the coarse-grained crowd counting algorithm, which can be verified by the experimental results in Sect. 3.

The moving average filter is expressed as follows:

$$\overline{\left|h_{1,q}(i)\right|} = \frac{\left(\left|h_{1,q}(i)\right| + \left|h_{1,q}(i+1)\right| + \left|h_{1,q}(i+2)\right| + \cdots + \left|h_{1,q}(i+N-1)\right|\right)}{N}, \quad (3)$$

where i is the serial number of the data packet, N is the length of the sliding window and is set as 5 in the paper, and q is the serial number of the subcarrier. We applied the moving average filter to the CSI amplitude data shown in Fig. 3, and the resulting filtered data is presented in Fig. 4. As can be seen from the figure, the filter effectively reduced the amount of noise present in the data.

2.3 Image construction

In this paper, each link contains 30 subcarriers, and each subcarrier contains a few seconds of CSI data. The sampling frequency of CSI date is 1000 Hz. Therefore, one sample of raw CSI data includes a large amount of CSI amplitude data. If raw CSI amplitude data is directly used, the algorithm's running speed will be very slow. To reduce the time complexity of the algorithm, we construct raw CSI amplitude data for each sample into the matrix form of amplitude image. On the one hand, the optimized matrix operation can greatly reduce the time complexity of the algorithm, such as the matrix operation in MATLAB. On the other hand, mature digital image processing technology can be used to extract the features that characterize differences between amplitude images such as texture and color. After experimental verification in this paper, these features can ensure that the multi-state crowd counting algorithm achieves high accuracy.

To leverage the data correlation between CSI subcarrier and the temporal features of the CSI amplitude data, we represent the preprocessed CSI amplitude data as an amplitude image, using the following approach:

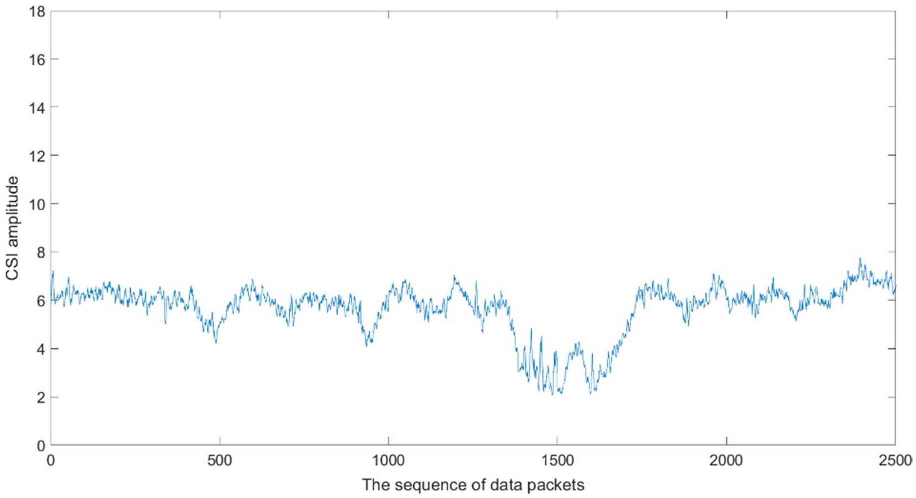


Fig. 4 The CSI amplitude data after denoising by using the moving average filter

$$\text{Am} = \begin{vmatrix} \overline{|h_{1,1}(1)|} & \cdots & \overline{|h_{1,1}(i)|} & \cdots & \overline{|h_{1,1}(I)|} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \overline{|h_{1,q}(1)|} & \cdots & \overline{|h_{1,q}(i)|} & \cdots & \overline{|h_{1,q}(I)|} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \overline{|h_{1,Q}(1)|} & \cdots & \overline{|h_{1,Q}(i)|} & \cdots & \overline{|h_{1,Q}(I)|} \end{vmatrix}, \tag{4}$$

where Am is a two-dimensional matrix representing the amplitude image, q is the subcarrier serial number ($q = 1, \ldots, Q$), i represents the serial number of data packets ($i = 1, \ldots, I$), and $\overline{|h_{1,q}(i)|}$ represents the amplitude of the ith data packet in the q-th subcarrier. According to the Am matrixes of the samples, i is set as the x-axis and q is set as the y-axis, and the examples of amplitude images are drawn as shown in Fig. 5. From Fig. 5, the texture features of CSI amplitude images corresponding to the different states of 2 people and 4 people are also different, which shows that it is rationality to recognize the number and states of people by using the texture features of images.

While the amplitude image provides information on the trend of CSI amplitude changes over time, it does not capture the frequency-domain variations caused by different human actions. To address this limitation and enable to recognize the number of people in multi-states, we apply the Morlet wavelet transform to the preprocessed CSI amplitude data. Thus, we can construct a time–frequency image of CSI amplitude, using the calculated wavelet coefficients as follows:

$$\text{TF} = \begin{vmatrix} x_{11} & \cdots & x_{1i} & \cdots & x_{1I} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{j1} & \cdots & x_{ji} & \cdots & x_{jI} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{J1} & \cdots & x_{Ji} & \cdots & x_{JI} \end{vmatrix}, \tag{5}$$

where i denotes the ith time component ($i = 1, \ldots, I$), j denotes the jth frequency component $\left(j = 1, \ldots, J\right)$, and x_{ji} denotes the wavelet coefficient. As the action frequencies

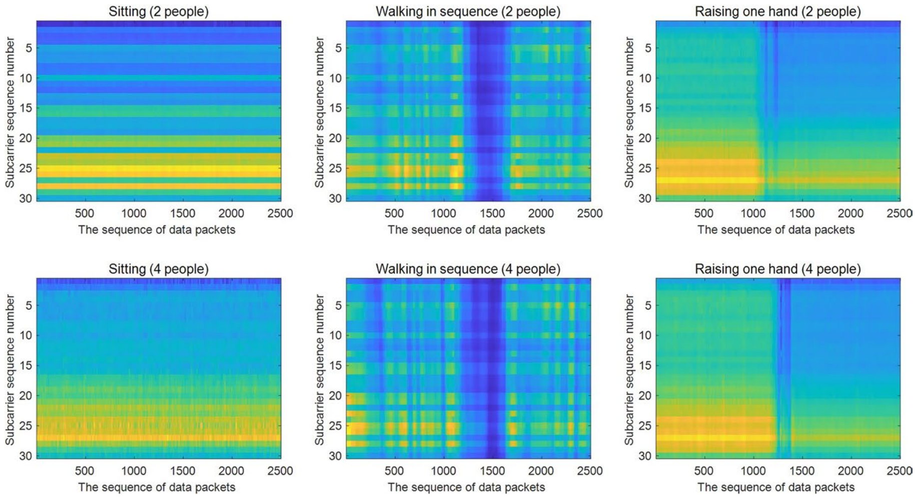


Fig. 5 The amplitude images of CSI

of humans are low, the number of frequency components J can be set to a fixed value. For example, in this paper, J is set to 60, which can fully represent the change frequency of human action. Using the TF matrixes of the samples, we set i as the x-axis and j as the y-axis, and draw the examples of time–frequency images as shown in Fig. 6. From Fig. 6, it can also be seen that the texture features of CSI time–frequency images affected by different number and states of people are also different. Therefore, using the texture features of CSI amplitude images and time–frequency images can more accurately count multi-state crowds.

2.4 Texture feature extraction

The TF-LPMCC algorithm aims to classify the texture features of the constructed CSI amplitude and time–frequency images to recognize the number of people in different states. The texture is a feature that reveals the local variations and spatial distribution of image pixels. In this paper, the GLCM method [23–25] is utilized to extract the texture features of the CSI amplitude and time–frequency images to achieve efficient crowd counting. GLCM can capture the amplitude change of images in different directions and neighboring intervals, and analyze the relevant features of spatial distribution and arrangement of image pixels. However, the CSI amplitude images also contain a significant number of local difference texture features. To improve the accuracy of crowd counting, the GLDS method is also used to extract texture features of the CSI amplitude images, and both sets of texture features are combined to form the feature vector for crowd counting.

2.4.1 GLCM

GLCM is a statistical method proposed by Haralick et al. [26] in 1973, which is used to represent the joint probability density between pixels of a certain distance and orientation. Its mathematical expression is shown below.

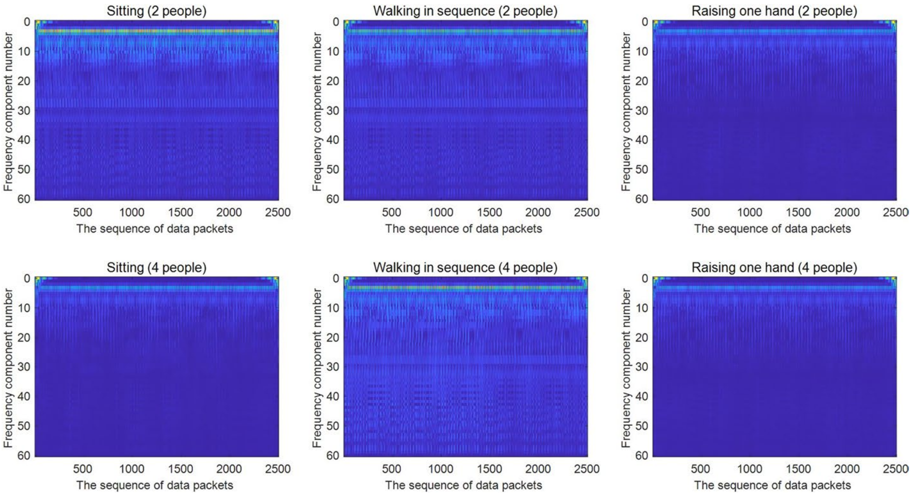


Fig. 6 The time–frequency images of CSI

$$P(a,b|c,d) = \frac{\#\left\{(x,y)|f(x,y) = a, f(x \pm c, y \pm d) = b\right\}}{\#\left\{(x,y)|f(x,y) = p1, f(x \pm c, y \pm d) = p2\right\}}, \tag{6}$$

where $\#\{X\}$ represents the number of elements in the set X, $a, b = 0, 1, 2, \cdots, K-1$, K is the total number of gray levels of the pixel values, $p1, p2$ are two arbitrary gray levels, x, y are the column and row numbers of the pixels in the image, c and d denote the relative distances of two pixels in the x and y directions, respectively, and $f(x, y)$ denotes the pixel value in the x-th column and y-th row.

To reduce the time complexity of the algorithm, we select four mutually uncorrelated statistics of energy, entropy, contrast, and correlation as the texture features of GLCM. Although Haralick et al. [26] proposed 14 kinds of statistics calculated according to GLCM, most of these statistics are correlated.

(1) The feature known as energy, or angular second-order moment, is used to measure the uniformity of image texture. It is calculated using the following equation:

$$\mathrm{ASM}(c,d)=\sum_a \sum_b P(a,b\big|c,d)^2, \tag{7}$$

where $\sum_a \cdot$ and $\sum_b \cdot$ denote $\sum\limits_{a=0}^{K-1} \cdot$ and $\sum\limits_{b=0}^{K-1} \cdot$, respectively.

(2) Entropy is a feature that characterizes the level of confusion, complexity, and randomness in an image. It is calculated using the following equation:

$$\mathrm{ENT}(c,d) = -\sum_a \sum_b P(a,b\big|c,d)\log(P(a,b\big|c,d)). \tag{8}$$

(3) Contrast is a feature that characterizes the sharpness and intensity of the transitions between neighboring pixel values in an image, indicating the presence of edges or boundaries, and is calculated as follows:

$$\mathrm{CON}(c,d) = \sum_a \sum_b (a-b)^2 P(a,b\big|c,d). \tag{9}$$

(4) Correlation is a feature that measures the degree of linear dependence between local pixels in an image and is calculated as follows:

$$\mathrm{COR}(c,d) = \frac{\sum_a \sum_b abP(a,b\big|c,d) - \mu_1(c,d)\mu_2(c,d)}{\sigma_1^2(c,d)\sigma_2^2(c,d)}, \tag{10}$$

where

$$\begin{cases} \mu_1(c,d) = \sum_a a \sum_b P(a,b\big|c,d) \\ \mu_2(c,d) = \sum_b b \sum_a P(a,b\big|c,d) \\ \sigma_1^2(c,d) = \sum_a (a-\mu_1(c,d))^2 \sum_b P(a,b\big|c,d) \\ \sigma_2^2(c,d) = \sum_b (b-\mu_2(c,d))^2 \sum_a P(a,b\big|c,d) \end{cases}. \tag{11}$$

The above four features are all functions of c and d. To comprehensively characterize the features of multiple directions of pixels and reduce the time complexity of the algorithm, in this paper, we set c and d as (1,0) and (1,1), respectively, and realize that the orientation angle θ of the pixel is 0° and 45°, respectively. Then, we calculate the mean and standard deviation of the above two texture features, respectively, which constitute the feature vector extracted according to GLCM as follows:

$$F_{\text{GLCM}} = [\mu_{\text{ASM}}, \sigma_{\text{ASM}}, \mu_{\text{ENT}}, \sigma_{\text{ENT}}, \mu_{\text{CON}}, \sigma_{\text{CON}}, \mu_{\text{COR}}, \sigma_{\text{COR}}], \tag{12}$$

where μ_{ASM}, σ_{ASM}, μ_{ENT}, σ_{ENT}, μ_{CON}, σ_{CON}, μ_{COR} and σ_{COR} represent the mean and standard deviation of $\text{ASM}(c,d)$, $\text{ENT}(c,d)$, $\text{CON}(c,d)$, and $\text{COR}(c,d)$ of the GLCM, respectively.

Since the GLCM of the CSI amplitude and time–frequency images are calculated separately and the texture features are extracted, the GLCM-based texture features can be represented as follows:

$$F_1 = [F_{\text{GLCM1}}, F_{\text{GLCM2}}], \tag{13}$$

where F_{GLCM1} and F_{GLCM2} denote the texture feature vectors extracted from the GLCM of the amplitude and time–frequency images, respectively.

2.4.2 GLDS

GLDS is a statistical technique that characterizes the variation of grayscale values among adjacent image pixels, allowing for the analysis of differences and fluctuations in localized regions of the image.

If the position of a pixel is $\left(x,y\right)$ and the position of a neighboring pixel is $\left(x+\Delta x, y+\Delta y\right)$, then the grayscale difference between the two pixels can be expressed as:

$$f_\Delta\left(x,y\right) = f\left(x,y\right) - f\left(x+\Delta x, y+\Delta y\right), \tag{14}$$

The grayscale difference, denoted as $f_\Delta\left(x,y\right)$, represents the variation between adjacent pixel values in an image. Typically, both Δx and Δy are small deviations, and for the purposes of this paper, both Δx and Δy have been fixed at a value of 1.

Given M possible levels for grayscale differences, a histogram of $f_\Delta\left(x,y\right)$ can be constructed to compute the probability, $P_\Delta(m)$, for each value of $f_\Delta\left(x,y\right)$ using the histogram, where $m = 1, 2, \cdots, M$. In this paper, we utilize the grayscale difference probability distribution, $P_\Delta(m)$, to extract four texture features from the amplitude images, namely contrast, angular second-order moment, entropy, and mean. The following equations are employed for their computation:

$$\mathrm{CON}_{\mathrm{GLDS}} = \sum_{m=0}^{M-1} m^2 P_\Delta(m), \tag{15}$$

$$\mathrm{ASM}_{\mathrm{GLDS}} = \sum_{m=0}^{M-1} P_\Delta^2(m), \tag{16}$$

$$\mathrm{ENT}_{\mathrm{GLDS}} = -\sum_{m=0}^{M-1} P_\Delta(m) \lg P_\Delta(m), \tag{17}$$

$$\mathrm{MEAN}_{\mathrm{GLDS}} = \frac{1}{M}\sum_{m=0}^{M-1} m P_\Delta(m). \tag{18}$$

The texture features of the amplitude image, which are based on the four aforementioned statistics (i.e., contrast, angular second-order moment, entropy, and mean), can be expressed using the following equations:

$$F_{\mathrm{GLDS}} = [\mathrm{CON}_{\mathrm{GLDS}}, \mathrm{ASM}_{\mathrm{GLDS}}, \mathrm{ENT}_{\mathrm{GLDS}}, \mathrm{MEAN}_{\mathrm{GLDS}}]. \tag{19}$$

The feature vector for multi-state crowd counting consists of the texture features extracted from amplitude and time–frequency images using both GLCM and GLDS methods.

$$F = [F_{\mathrm{GLCM1}}, F_{\mathrm{GLCM2}}, F_{\mathrm{GLDS}}]. \tag{20}$$

2.5 LDA algorithm

To increase the running speed of the algorithm, we utilize the LDA algorithm, which has low time complexity, to recognize the number of people. The LDA algorithm transforms the high-dimensional classification problem into a one-dimensional classification problem using the projection method. The specific algorithm is outlined below.

The intra-class dispersion matrix for the same category samples is calculated as follows:

$$S_{ic} = \sum_{u=1}^{U}\sum_{v=1}^{V} (F(u,v) - \mu_u)(F(u,v) - \mu_u)^T, \tag{21}$$

where u denotes the serial number of the category ($u = 1, 2, \cdots, U$, where U is the number of categories for multi-state crowd counting.), v denotes the serial number of the sample ($v = 1, 2, \cdots, V$, where V is the number of samples collected for each category.), $F(u,v)$ denotes the eigenvector of the v-th sample of the u-th category, μ_u is the mean of the V samples for the u-th category, and T denotes the transpose of the matrix.

The inter-class dispersion matrix is calculated as follows:

$$S_{bc} = \sum_{u=1}^{U} (\mu_u - \mu)(\mu_u - \mu)^T, \tag{22}$$

where μ denotes the mean value of all μ_u.

To achieve the minimum intra-class dispersion and the maximum inter-class dispersion, the following objective function must be optimized:

$$\underbrace{\arg\max}_{W} J(W) = \frac{\prod_{\text{diag}} W^T S_{\text{bc}} W}{\prod_{\text{diag}} W^T S_{ic} W}, \tag{23}$$

where $\prod\limits_{diag}$ denotes the product of the main diagonal elements of the matrix, $W = [w_1, w_2, \cdots, w_r]$ is a low-dimensional matrix in the projection direction with dimension $R \times r$, R denotes the length of the eigenvector F, and r denotes the total number of basis vectors in W. Generally, r is taken as the largest integer that is smaller than U. Equation (23) can become:

$$\underbrace{\arg\max}_{W} J(W) = \prod_{g=1}^{r} \frac{w_g^T S_{\text{bc}} w_g}{w_g^T S_{ic} w_g}. \tag{24}$$

The maximum value of the objective function is the product of the largest r eigenvalues of the matrix $S_{ic}^{-1} S_{bc}$. The eigenvectors corresponding to the largest r eigenvalues are $w_1, w_2, \cdots, w_r$.

3 Results

3.1 Experimental setup and data acquisition

In this study, experiments were conducted in a 3.5 m × 5 m laboratory containing tables, chairs, cabinets, and experimental equipments. The experimental scenario is depicted in Fig. 7. Two computers with Intel 5300 network cards and Ubuntu 12.04 operating systems were used. One computer was connected to one antenna as the transmitter, while the other computer was connected to three antennas as the receiver. The antennas were placed 0.5 m from the ground, and the distance between the transmitting and receiving devices was 3 m. The channel bandwidth of WiFi was set to 20 MHz, and the operating frequency was 2.4 GHz. Data was transmitted through three channels, each with 30 subcarriers, resulting in each data packet containing $1 \times 3 \times 30$ groups of CSI data. The experiment involved four volunteers, including two males and two females. Thirteen different experimental cases were conducted, as shown in Table 1. The crowd counting in these states has potential applications in smart education, such as calculating the number of students in different states in a classroom. The experimental setup required recognizing a total of $3 \times 4 + 1 = 13$ categories, including the case of no people. The device had a sending packet frequency of 1000 Hz, and 4 s of data were collected for each sample, consisting of 4000 data packets. Seventy samples were collected for each category, with 40 randomly selected for training and the remaining 30 for testing during algorithm simulation.

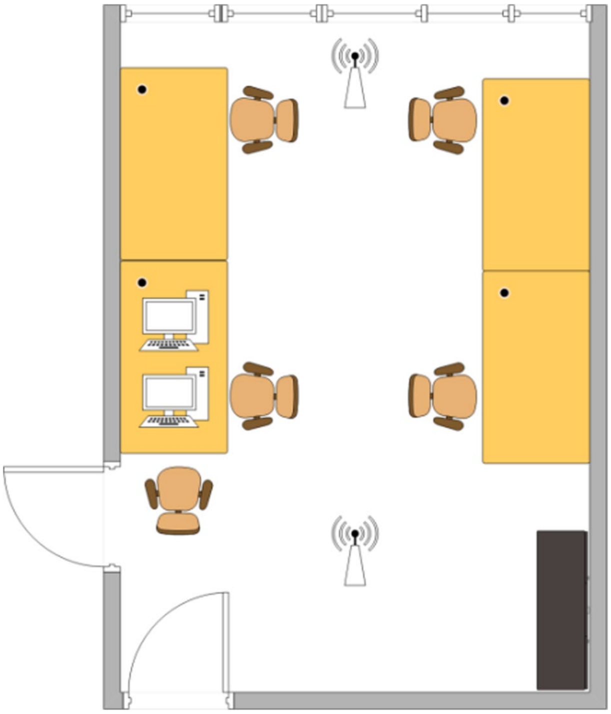


Fig. 7 The experimental scenario 1

Table 1 Setup of experimental scenes

Serial Number	Action	Number of people
1	Sitting	1
2	Sitting	2
3	Sitting	3
4	Sitting	4
5	Walking in sequence	1
6	Walking in sequence	2
7	Walking in sequence	3
8	Walking in sequence	4
9	Raising one hand	1
10	Raising one hand	2
11	Raising one hand	3
12	Raising one hand	4
13	Empty	0

3.2 Parameter analysis

3.2.1 Analysis of parameter J

The TF-LPMCC algorithm's resolution of frequency components varies with the number of frequency components J, which may affect the recognition accuracy of action and crowd counting. To evaluate the impact of J, we tested the algorithm's performance with J values of 20, 40, 60, 80, and 100, and compared the average recognition

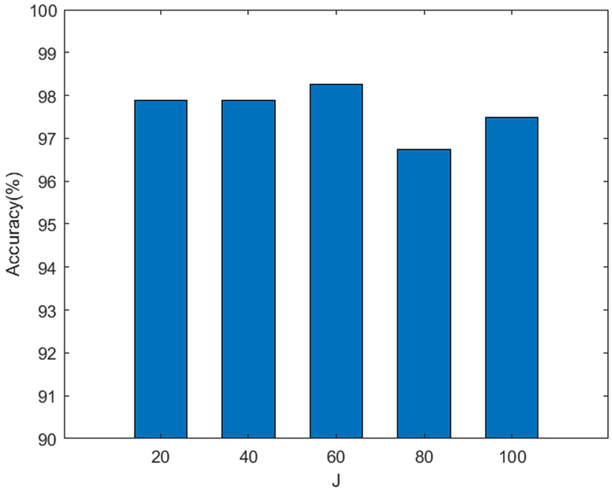


Fig. 8 Effect of parameter *J* on average recognition accuracy

Table 2 The values of parameters *c* and *d*

	The values of *c* and *d*	Orientation angle of the pixel θ
Cd1	(1,0), (1,1)	0°, 45°
Cd2	(0,1), (− 1,1)	90°, 135°
Cd3	(1,0), (1,1), (0,1), (− 1,1)	0°, 45°, 90°, 135°
Cd4	(1,0), (1,1), (2,0), (2,2)	0°, 45°, 0°, 45°
Cd5	(0,1), (− 1,1), (0,2), (− 2,2)	90°, 135°, 90°, 135°
Cd6	(1,0), (1,1), (0,1), (− 1,1), (2,0),(2,2),(0,2),(− 2,2)	0°, 45°, 90°, 135°, 0°, 45°, 90°, 135°

accuracy. The results are presented in Fig. 8, which shows that the algorithm achieves the highest average recognition accuracy of 98.27% when $J = 60$. However, for all other J values, the average recognition accuracy was above 96%, indicating that J has a minor impact on the algorithm's performance. This is because human action frequency is relatively slow, and all tested J values can adequately capture the changes in human action frequency. In conclusion, when applying the TF-LPMCC algorithm, the parameter J can be set to 20 for high operation speed or 60 for high average recognition accuracy.

3.2.2 Analysis of parameters c and d

The TL-LPMCC algorithm uses texture features calculated according to the GLCM, where the texture features are functions of parameters c and d. To analyze the effect of the values of parameters c and d on the performance of the algorithm, the authors set the parameters c and d according to Table 2 and compared the average recognition accuracy of the TL-LPMCC algorithm. The experimental results, as shown in Fig. 9, demonstrate that the TL-LPMCC algorithm achieves the highest average recognition accuracy when the parameters c and d are set to (1,0) and (1,1). As the parameters c and d vary from Cd1 to Cd6, the average recognition accuracy of the TL-LPMCC

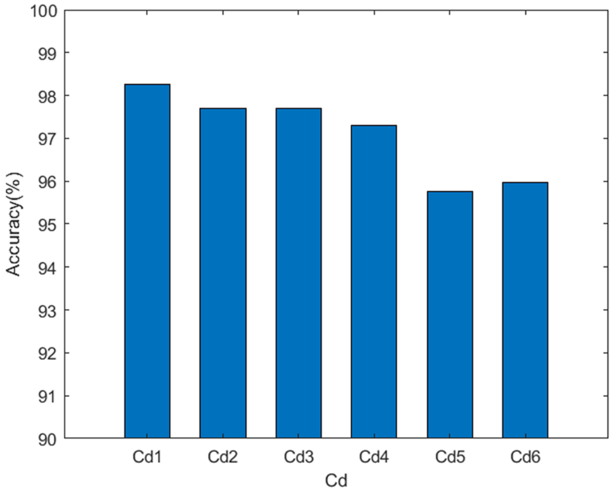


Fig. 9 Effect of parameters *c* and *d* on average recognition accuracy

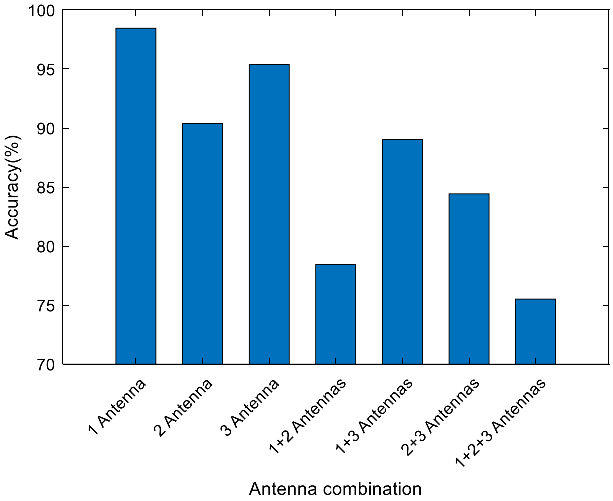


Fig. 10 Experimental results of transmitting and receiving antennas

algorithm gradually decreases. The authors note that the algorithm complexity is significantly lower when c and d are set to Cd1 and Cd2 than when other values are used. Therefore, the values of parameters c and d can be set to (1,0) and (1,1) when using the TL-LPMCC algorithm.

3.2.3 Analysis of transmitting and receiving antennas

In the experiment, the transmitter is connected to 1 antenna, and the receiver is connected to 3 antennas. To analyze the impact of the number of transmitting and receiving antennas on the accuracy of the TF-LPMCC algorithm, we number the receiving antennas as 1, 2, and 3. 1 Antenna, 2 Antenna, and 3 Antenna represent the experiments using CSI data received by antennas 1, 2, and 3 for multi-state crowd counting, respectively, and $1+2$ Antennas represent the experiments using CSI data received by antennas 1 and 2, and so on. Figure 10 shows the average recognition accuracy of the TF-LPMCC algorithm under different combinations of transmitting and receiving antennas. From Fig. 10, the average recognition accuracies obtained by using CSI data from different receiving antennas are also different. When only one antenna is used, the algorithm has the highest accuracy. The more antennas are used, the lower the accuracy of the algorithm, and the higher the time complexity of the algorithm. This is because the CSI data received by different antennas are affected differently by multipath interference effects. Therefore, we evaluate the performance of the TF-LPMCC algorithm using CSI data from the first receiving antenna in the paper.

3.2.4 Analysis of subcarriers and bandwidth

To analyze the impact of the number of subcarriers on the accuracy of the TF-LPMCC algorithm, we compare the average recognition accuracy of the TF-LPMCC algorithm when using CSI data of 1, 5, 10, 20, and 30 subcarriers from the first link. The experimental results are shown in Fig. 11. From Fig. 11, the average recognition accuracy of the TF-LPMCC algorithm shows an upward trend as the number of subcarriers increases. When using 30 subcarriers, the recognition accuracy of the algorithm is the highest. Therefore, we use 30 subcarriers in the subsequent experiments.

It is necessary to analyze the impact of channel bandwidth on the accuracy of the TF-LPMCC algorithm, as signals with different frequencies are subject to different environmental interference and different impact of multipath effects. We set the bandwidth to 10, 20, 30, and 40 MHz, respectively, and then evaluate the average recognition accuracy of the TF-LPMCC algorithm. The experimental results are shown in Fig. 12. From Fig. 12, the average recognition accuracy of the TF-LPMCC algorithm increases with the increase in bandwidth, because the greater the frequency difference between subcarriers, the less interference between them. However, when the bandwidth changes from 10 to 40 MHz, the accuracy of the algorithm does not change much.

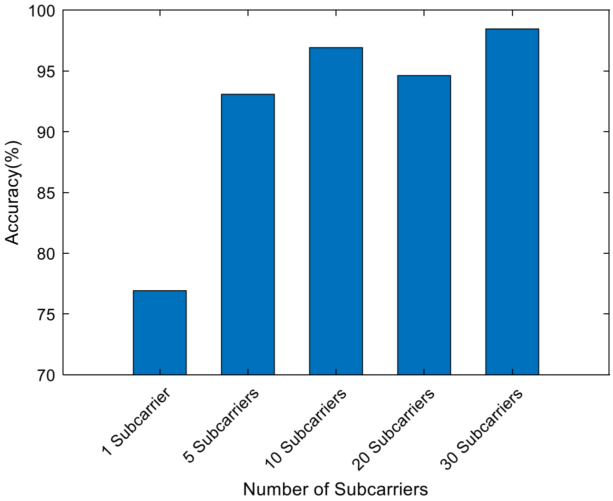


Fig. 11 Experimental results of subcarriers

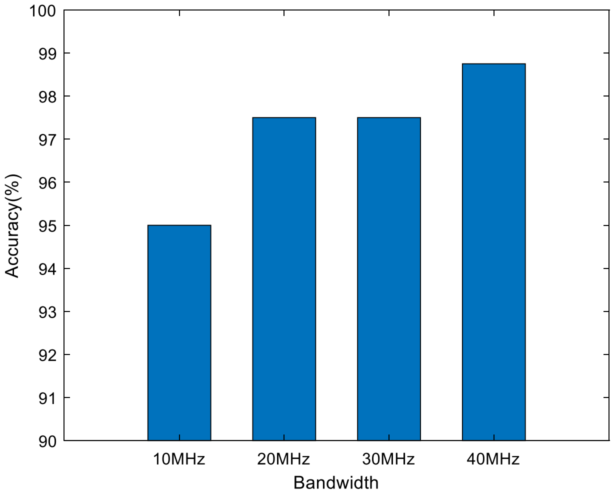


Fig. 12 Experimental results of bandwidths

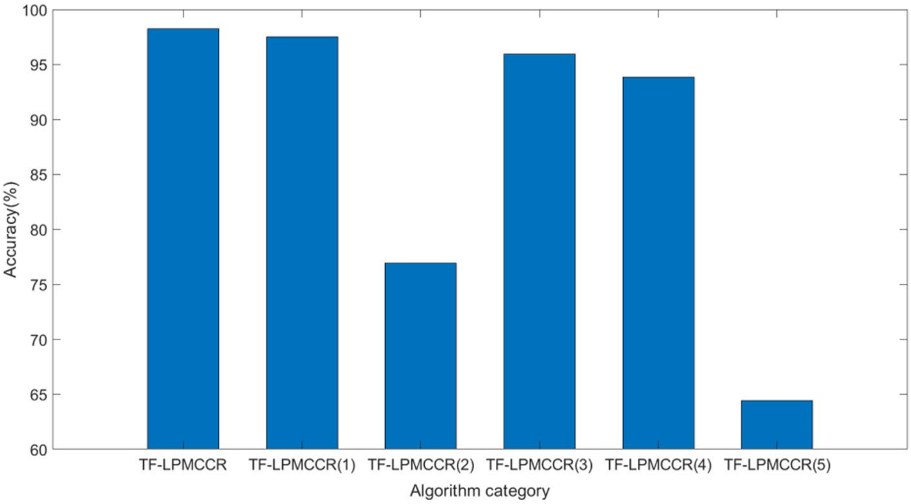


Fig. 13 The results of the ablation study

3.3 Performance of TF-LPMCC algorithm

3.3.1 Ablation study

In the TF-LPMCC algorithm, amplitude and time–frequency images of CSI are constructed and texture features are extracted using the GLCM method for both images. Additionally, texture features are extracted from the amplitude images using the GLDS method. The composed feature vectors are then input to the LDA algorithm for classification. To evaluate the contribution of each step in the TF-LPMCC algorithm, we compare its performance with five other algorithms: (i) TF-LPMCC(1) which uses only the GLCM method for texture feature extraction, (ii) TF-LPMCC(2) which uses only the GLDS method for texture feature extraction, (iii) TF-LPMCC(3) which uses only the CSI amplitude image, (iv) TF-LPMCC(4) which uses only the GLCM method for texture feature extraction of the CSI amplitude image without using the GLDS method or CSI time–frequency image, and (v) TF-LPMCCR(5) which extracts texture features only from the CSI time–frequency image without using the CSI amplitude image.

Figure 13 shows the simulation results of the TF-LPMCC algorithm and the five other algorithms. The TF-LPMCC algorithm achieves the highest average recognition accuracy of 98.27%. In contrast, the TF-LPMCC(2) and TF-LPMCC(5) algorithms have significantly lower average recognition accuracies, suggesting that using only the CSI time–frequency images or the GLDS methods results in poorer performance. The average recognition accuracies of the TF-LPMCCR(1), TF-LPMCCR(3), and TF-LPMCCR(4) algorithms are all above 93%, indicating that the CSI amplitude image and the GLCM method contribute much more to the recognition accuracy of the TF-LPMCC algorithm than the CSI time–frequency image and the GLDS method. However, the CSI time–frequency image and the GLDS method can further improve the average recognition accuracy. Therefore, the TF-LPMCC algorithm can adjust the composition of the algorithm based on the application's requirements. If high average recognition accuracy is required, the TF-LPMCC algorithm can be used. If less running time is required, the TF-LPMCC(1), TF-LPMCC(3), or TF-LPMCC(4) algorithms can be used.

3.3.2 Comparing different algorithms

Currently, there are fewer studies on CSI-based crowd counting compared to CSI-based activity recognition. Two existing works [14] and [10] achieved good crowd counting results using the SVM algorithm (referred to as PNR-SVM) and naive Bayesian classification algorithm (referred to as PNR-NB), respectively. In this paper, we compare the average recognition accuracy of TF-LPMCC, PNR-SVM, and PNR-NB algorithms, as shown in Fig. 14. The results demonstrate that the TF-LPMCC algorithm achieves the highest average recognition accuracy of 98.27%, which is 4.04% and 4.42% higher than that of PNR-SVM and PNR-NB algorithms, respectively. In TF-LPMCC algorithm, the amplitude image of CSI fully utilizes the temporal stability of CSI, and the time–frequency image of CSI fully utilizes the frequency diversity characteristics of CSI. The GLCM and GLDS methods extract the local changes and spatial distribution features

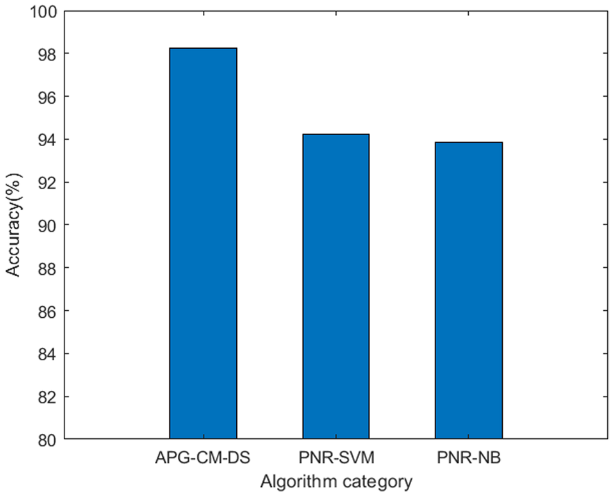


Fig. 14 Comparison results of different algorithms

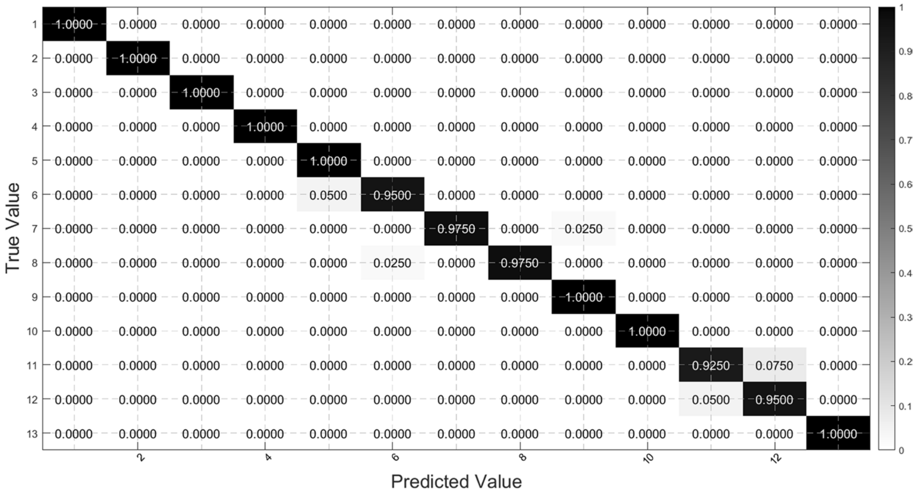


Fig. 15 Confusion matrix of TF-LPMCC algorithm

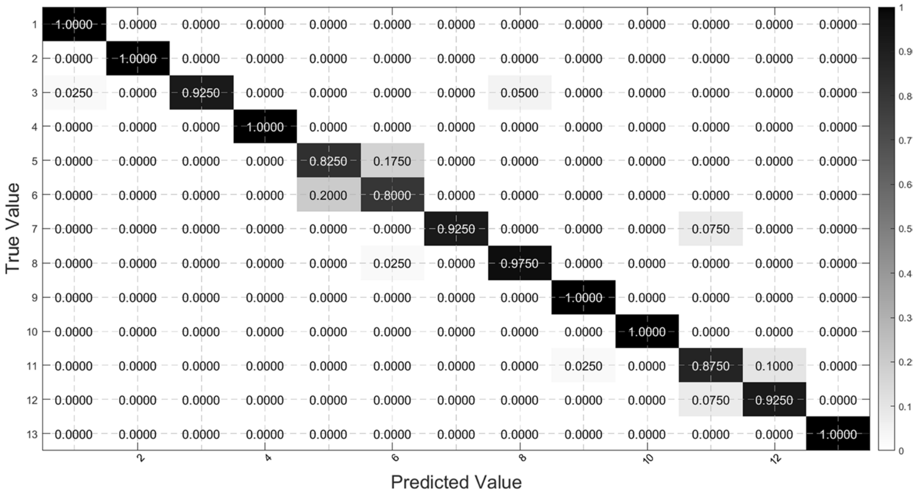


Fig. 16 Confusion matrix of PNR-SVM algorithm

of CSI data. Therefore, the superior performance of the TF-LPMCC algorithm is attributed to its ability to extract more fine-grained features of multi-state crowd information contained in the CSI amplitude. In addition, we compare the confusion matrices of the three algorithms, as shown in Figs. 15, 16 and 17, respectively, where the meanings of the serial numbers in the confusion matrix are shown in Table 1. The results indicate that the recognition accuracy of TF-LPMCC algorithm is above 90% for all categories, while the recognition accuracy of PNR-SVM algorithm is below 85% for categories 5 and 6, and the recognition accuracy of PNR-NB algorithm is below 80% for categories 5 and 11. This demonstrates that the TF-LPMCC algorithm not only has a high average recognition accuracy but also has a high recognition accuracy of all categories.

To evaluate the computational efficiency of the TF-LPMCC algorithm, we measured the running times of the three algorithms on a laptop computer equipped with an Intel I5-7200U 2.5 GHz CPU and 8 GB RAM. The TF-LPMCC, PNR-SVM, and

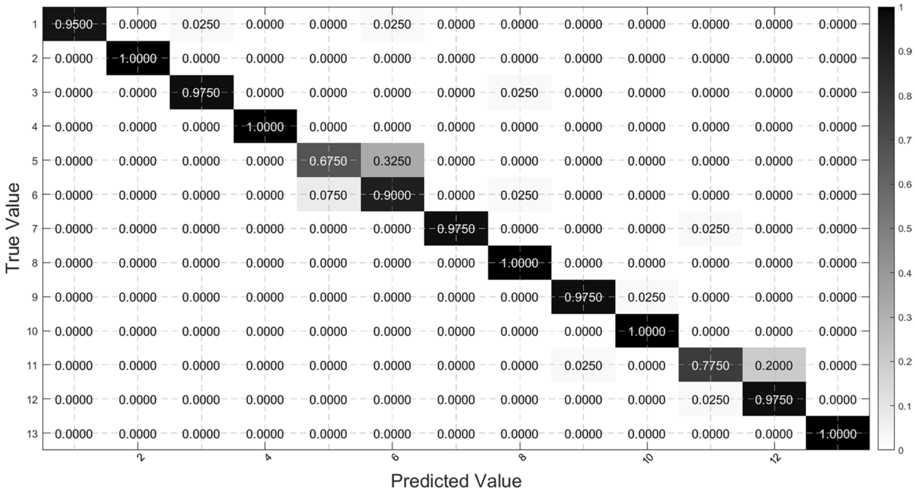


Fig. 17 Confusion matrix of PNR-NB algorithm

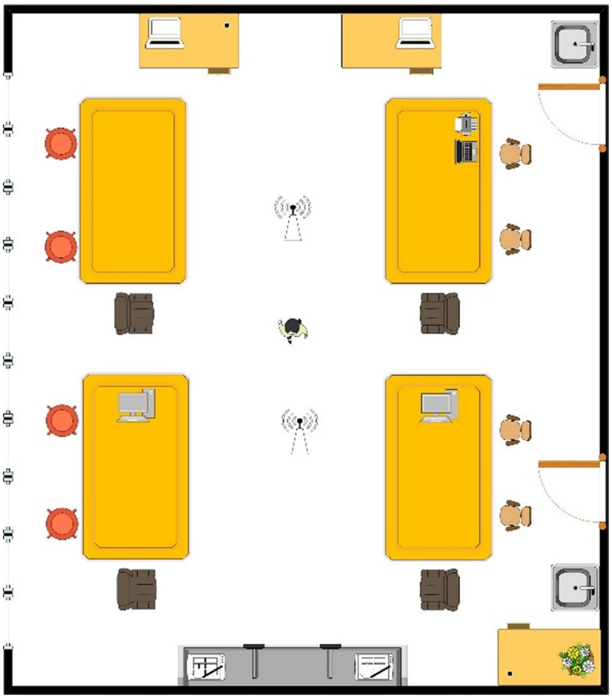


Fig. 18 The experimental scenario 2

PNR-NB algorithms took 0.068 s, 0.128 s, and 0.197 s, respectively, to recognize one sample. Notably, the TF-LPMCC algorithm had the shortest running time compared to the PNR-SVM and PNR-NB algorithms, with a reduction of 46.88% and 65.48%, respectively. These results demonstrate that the TF-LPMCC algorithm exhibits both high accuracy and low time complexity.

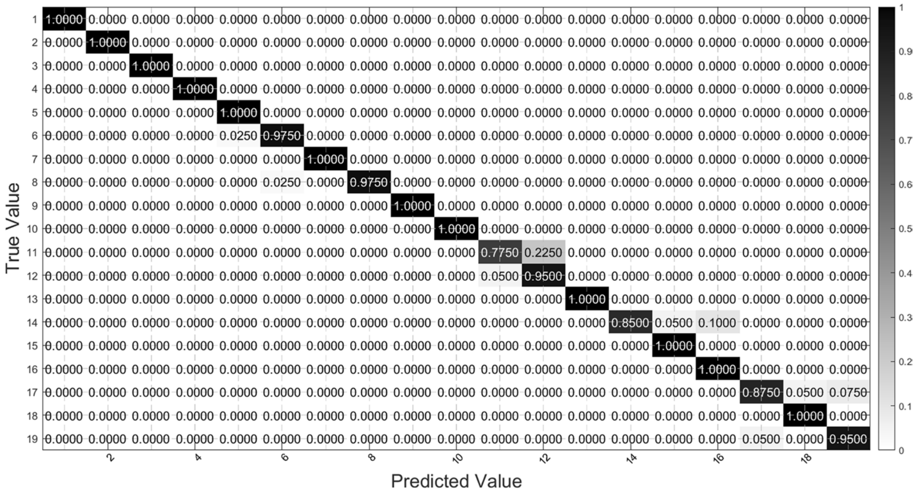


Fig. 19 Confusion matrix for 19 types of scenarios

Table 3 The experimental results in the new scenario

Activity/Number of people	1 people	2 people	3 people	4 people	Average
State 1	1	1	1	1	1
State 2	1	1	1	0.95	0.9875
State 3	1	1	0.95	0.95	0.975
State 4	1	0.90	0.90	0.80	0.90
State 5	1	0.95	1	0.95	0.975
State 6	1	1	1	1	1
Total average recognition accuracy					0.9729

3.3.3 Analysis of algorithm scalability

In the previous analysis and experiments, we have verified that the TF-LPMCC algorithm has low time complexity and high recognition accuracy in the experimental scenario described in Sect. 3.1. However, the scalability of this algorithm in other experimental scenarios and different states of volunteers still needs further analysis. For this purpose, we conducted an experiment in another laboratory that was different from the experimental scenario in Fig. 7, as shown in Fig. 18. In this experimental scenario, we still collect CSI data for 1–4 people, but the volunteers also add three states of chest-expanding exercise (State 4), bending and lifting both arms in front of the chest (State 5), and arm pulling from high to low (State 6), in addition to the three states of sitting (State 1), raising one hand (State 2), and walking in sequence (State 3). Therefore, the number and states of people are classified into 24 categories to recognize. The experimental setup is the same as Sect. 3.1. Fifty samples are collected for each category, where 30 samples are randomly selected for training and the remaining 20 samples are tested (Fig. 19). The experimental results are shown in Table 3.

Table 3 shows that in the new experimental scenario, the average recognition accuracy of the TF-LPMCC algorithm for State 1, 2, and 3 is 98.75%, which is consistent with the experimental results in the scenario shown in Fig. 7. After adding State 4, 5, and 6, the average recognition accuracy of the algorithm slightly decreased, but it can still reach as high as 97.29%. However, in the case that there are four people who are in State 4, the recognition accuracy of the algorithm is only 80%, which shows that the number of people and the complexity of the state have a certain impact on the recognition accuracy of the algorithm. From Table 3, the more the number of people and the more complex the state of people, the lower the recognition accuracy of the algorithm. In summary, the TF-LPMCC algorithm can still achieve high recognition accuracy of crowd size and state in different experimental scenarios and more states, which shows that this algorithm has good scalability. Although the number and states of people are limited, the average recognition accuracy of the algorithm can already meet the needs of most applications.

4 Discussion and limitation

Although the TF-LPMCC algorithm performs well, there are still some issues that require further discussion.

(1) In this paper, we conducted experiments on thirteen cases as shown in Table 1, collecting 70 samples for each case. Consequently, a total of 910 samples were collected, making the sample collection process labor-intensive. As the number and states of crowds to be recognized by the TF-LPMCC algorithm increase, so does the workload of collecting training samples. This hinders the applicability and scope of the TF-LPMCC algorithm.

(2) The TF-LPMCC algorithm is capable of accurately recognizing the crowd size when all individuals perform the same action. However, in real-world applications, people may perform different actions. For example, in a room with four people, two may be sitting while the other two are walking. To test the performance of the TF-LPMCC algorithm in such scenarios, we followed the same experimental setup as in Sect. 4.1 and conducted additional experiments in six new scenarios: 1 person walking with 1, 2, and 3 people sitting, respectively, 2 people walking with 1 and 2 people sitting, respectively, and 3 people walking with 1 person sitting. Therefore, the TF-LPMCC algorithm needed to classify a total of 19 scenes. The experimental results show that the algorithm's average recognition accuracy can still reach 96.58%, indicating that the TF-LPMCC algorithm can still perform well in counting crowds in arbitrary states. However, the confusion matrix shown in Fig. 15 reveals that the recognition accuracy of the algorithm decreased to less than 90% in the 11th, 14th, and 17th cases, indicating that while the algorithm's average recognition accuracy decreases less, the recognition accuracy of the algorithm decreases significantly for a few categories after adding six scenarios where the crowd is in arbitrary states.

(3) If a human and an object, such as a robot or a chair, enter the monitoring area at the same time, the TF-LPMCC algorithm cannot distinguish between the human and the object, and the object is also recognized as a human. Therefore, the TF-LPMCC algorithm is only applicable to the scenarios where only humans are dynamically changing in the monitoring area.

5 Conclusion

As artificial intelligence continues to advance, the demand for crowd counting applications is increasing. However, existing studies cannot still count crowds in different states, and the accuracy and time complexity of crowd counting algorithms need further improvement. In response to this need, we propose the TF-LPMCC algorithm, which constructs CSI data into amplitude images and time–frequency images, and extracts texture features from the two images using the GLCM method. To enhance the algorithm's recognition accuracy, we also extract texture features from the amplitude images using the GLDS method. The features extracted from both methods form the input feature vector of the LDA classification algorithm. We conducted extensive experiments to analyze the effects of the parameters and on the recognition accuracy of the TF-LPMCC algorithm. Through an ablation study, we illustrated the contribution of each method of the TF-LPMCC algorithm to recognition accuracy. Results compared with existing algorithms demonstrate that the TF-LPMCC algorithm not only achieves a higher average recognition accuracy of up to 98.27%, but also has a lower algorithm running time of 0.068 s.

Moving forward, we will focus on two aspects related to our work: (i) The TF-LPMCC algorithm currently requires a large and expensive workload for testing training samples to recognize the number of people in multi-states. To address this issue, we will explore algorithms that can achieve high recognition accuracy using smaller samples and also aim to enhance the cross-domain performance of the algorithm when adapting to new application environments. (ii) As the number of people increases, the stability of the TF-LPMCC algorithm decreases when counting crowds in arbitrary states. This not only increases the human and financial cost of collecting training samples but also reduces the algorithm's performance. We will work toward developing algorithms that can effectively recognize the number of people in arbitrary states, even when counting more people.

6 Methods/experimental

The existing crowd counting algorithms struggle with low counting accuracy and high algorithm complexity when counting humans in multiple states. For this problem, we construct CSI amplitude data into amplitude and time–frequency images, and then extract texture features using the gray-level co-occurrence matrix (GLCM) and gray-level difference statistic (GLDS) methods, and finally use the linear discriminant analysis (LDA) algorithm to count the crowd in three states. To verify the TF-LPMCC algorithm proposed in this paper, we conducted experiments in a laboratory. The layout of the laboratory, the devices and settings used in the experiments, the volunteers in the experiments, the activity design of the volunteers, and the data collection are all described in detail in Sect. 3.1. Using the collected data and a large number of simulations, we analyzed the performance of the proposed algorithm from many aspects and verified the accuracy and robustness of the proposed algorithm for multi-state crowd counting.

Abbreviations

CSI	Channel state information
TF-LPMCC	Texture features-based lightweight passive multi-state crowd counting

GLCM	Gray-level co-occurrence matrix
GLDS	Gray-level difference statistic
LDA	Linear discriminant analysis
RSS	Received signal strength
SVM	Support vector machine
NB	Naive Bayesian

Acknowledgements
The authors would like to acknowledge the editors and reviewers and all the participants for the paper.

Author contributions
YT was involved in methodology, supervision, writing—review, investigation, software, writing—original draft. JL helped in data collection, software, investigation. FG contributed to data preprocessing, software. CZ was involved in investigation, writing—original draft. XD helped in resources, writing—review & editing. All authors read and approved the final manuscript.

Funding
This work was supported in part by the National Natural Science Foundation of China under grant numbers 62076114, 71874025; the Applied Basic Research Program Project of Liaoning Province under grant number 2023JH2/101300189; the Humanities and Social Sciences Research Planning Foundation of the Ministry of Education of China under grant number 20YJA630058.

Availability of data and material
The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

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.

Received: 5 April 2023 Accepted: 7 August 2023
Published online: 17 August 2023

References

1. D. Khan, I. Wang-Hei Ho, CrossCount: Efficient device-free crowd counting by leveraging transfer learning. IEEE Internet Things J. **10**(5), 4049–4058 (2023)
2. H. Zhang, M. Zhou, H. Sun, G. Zhao, J. Qi, J. Wang, H. Esmaiel, Que-Fi: a Wi-Fi deep-learning-based queuing people counting. IEEE Syst. J. **15**(2), 2926–2937 (2021)
3. L. Zhang, Y. Zhang, B. Wang, X. Zheng, L. Yang, WiCrowd: counting the directional crowd with a single wireless link. IEEE Internet Things J. **8**(10), 8644–8656 (2021)
4. Z. Liu, R. Yuan, Y. Yuan, Y. Yang, X. Guan, A sensor-free crowd counting framework for indoor environments based on channel state information. IEEE Sens. J. **22**(6), 6062–6071 (2022)
5. Z. Wu, X. Zhang, G. Tian, Y. Wang, Q. Huang, Spatial-temporal graph network for video crowd counting. IEEE Trans. Circuits Syst. Video Technol. **33**(1), 228–241 (2023)
6. H. Ding, J. Han, A.X. Liu, W. Xi, J. Zhao, P. Yang, Z. Jiang, Counting human objects using backscattered radio frequency signals. IEEE Trans. Mob. Comput. **18**(5), 1054–1067 (2019)
7. S. Denis, B. Bellekens, M. Weyn, R. Berkvens, Sensing thousands of visitors using radio frequency. IEEE Syst. J. **15**(4), 5090–5093 (2021)
8. Y. Tian, C. Zhuang, J. Cui, R. Qiao, X. Ding, Gesture recognition method based on misalignment mean absolute deviation and KL divergence. EURASIP J. Wireless Commun. Netw. **96**, 1–21 (2022)
9. W. Xi, J. Zhao, X.Y. Li, K. Zhao, S. Tang, X. Liu, Z. Jiang, Electronic frog eye: Counting crowd using WiFi, in: 2014-IEEE Conference on Computer Communications (INFOCOM), 2014, pp. 361–369.
10. H. Zou, Y. Zhou, J. Yang, C.J. Spanos, Device-free occupancy detection and crowd counting in smart buildings with WiFi-enabled IoT. Energy Build. **174**, 309–322 (2018)
11. S. Liu, Y. Zhao, B. Chen, WiCount: A deep learning approach for crowd counting using WiFi signals, in: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), 2017, pp. 967–974.
12. Y. Zhao, S. Liu, F. Xue, B. Chen, X. Chen, DeepCount: crowd counting with Wi-Fi using deep learning. J. Commun. Inform. Netw. **4**(3), 38–52 (2019)
13. X. Ma, W. Xi, X. Zhao, Z. Chen, H. Zhang, J. Zhao, Wisual: Indoor crowd density estimation and distribution visualization using Wi-Fi. IEEE Internet Things J. **9**(12), 10077–10092 (2022)
14. Z. Guo, F. Xiao, B. Sheng, L. Sun, S. Yu, TWCC: A robust through-the-wall crowd counting system using ambient WiFi signals. IEEE Trans. Veh. Technol. **71**(4), 4198–4211 (2022)
15. R. Alizadeh, Y. Savaria, C. Nerguizian, Human activity recognition and people count for a SMART public transportation system, in: 2021 IEEE 4th 5G World Forum (5GWF), 2021, pp. 182–187.

16. H. Choi, T. Matsui, S. Misaki, A. Miyaji, M. Fujimoto, K. Yasumoto, Simultaneous crowd estimation in counting and localization using WiFi CSI. International Conference on Indoor Positioning and Indoor Navigation (IPIN) **2021**, 1–8 (2021)
17. J. Xiao, K. Wu, Y. Yi, L. Wang, L. M. Ni, Pilot: Passive device-free indoor localization using channel state information, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS), 2013, pp. 236–245.
18. Z. Chen, L. Zhang, C. Jiang, Z. Cao, W. Cui, WiFi CSI based passive human activity recognition using attention based BLSTM. IEEE Trans. Mob. Comput. **18**(11), 2714–2724 (2019)
19. J. Xia, J. Zhang, Y. Wang, L. Han, H. Yan, WC-KNNG-PC: Watershed clustering based on k-nearest-neighbor graph and Pauta Criterion. Pattern Recogn. **121**, 108177 (2022)
20. Z. Wang, Z. Huang, C. Zhang, W. Dou, Y. Guo, D. Chen, CSI-based human sensing using model-based approaches: a survey. J. Comput. Design Eng. **8**(2), 510–523 (2021)
21. X. Yang, J. Cheng, X. Tang, L. Xie, CSI-based human behavior segmentation and recognition using commodity Wi-Fi. EURASIP J. Wirel. Commun. Netw. **2023**(46), 1–25 (2023)
22. S.J. Kweon, S.H. Shin, S.H. Jo, H.J. Yoo, Reconfigurable high-order moving-average filter using inverter-based variable transconductance amplifiers. IEEE Trans. Circuits Syst. II Express Briefs **61**(12), 942–946 (2014)
23. Z. Xi, Y. Niu, J. Chen, X. Kan, H. Liu, Facial expression recognition of industrial internet of things by parallel neural networks combining texture features. IEEE Trans. Industr. Inf. **17**(4), 2784–2793 (2021)
24. X. Yang, Y. Ding, X. Zhang, L. Zhang, Spatial-temporal-circulated GLCM and physiological features for in-vehicle people sensing based on IR-UWB radar. IEEE Trans. Instrum. Meas. **71**, 1–13 (2022)
25. Y. Yuan, M.S. Islam, Y. Yuan, S. Wang, T. Baker, L.M. Kolbe, EcRD: Edge-cloud computing framework for smart road damage detection and warning. IEEE Internet Things J. **8**(16), 12734–12747 (2021)
26. R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification. IEEE Trans. Syst. Man Cybern. **SMC-3**(6), 610–621 (1973)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.