Computer Systems Science & Engineering
DOI:10.32604/csse.2022.020590
Article

Tech Science Press

Adaptive Scheme for Crowd Counting Using off-the-Shelf Wireless Routers

Wei Zhuang[1,2], Yixian Shen[1], Chunming Gao[3], Lu Li[1], Haoran Sang[4] and Fei Qian[5,*]

[1]School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[2]Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[3]School of Engineering & Technology, University of Washington, Tacoma, WA 98402, USA
[4]China General Nuclear Power Group, Nanjing, 210028, China
[5]Jiangsu Province Hospital of Chinese Medicine, Affiliated Hospital of Nanjing University of Chinese Medicine, Nanjing, 210029, China
*Corresponding Author: Fei Qian. Email: seujaguar@163.com
Received: 29 May 2021; Accepted: 30 June 2021

Abstract: Since the outbreak of the world-wide novel coronavirus pandemic, crowd counting in public areas, such as in shopping centers and in commercial streets, has gained popularity among public health administrations for preventing the crowds from gathering. In this paper, we propose a novel adaptive method for crowd counting based on Wi-Fi channel state information (CSI) by using common commercial wireless routers. Compared with previous researches on device-free crowd counting, our proposed method is more adaptive to the change of environment and can achieve high accuracy of crowd count estimation. Because the distance between access point (AP) and monitor point (MP) is typically non-fixed in real-world applications, the strength of received signals varies and makes the traditional amplitude-related models to perform poorly in different environments. In order to achieve adaptivity of the crowd count estimation model, we used convolutional neural network (ConvNet) to extract features from correlation coefficient matrix of subcarriers which are insensitive to the change of received signal strength. We conducted experiments in university classroom settings and our model achieved an overall accuracy of 97.79% in estimating a variable number of participants.

Keywords: CSI; device-free; deep learning; crowd counting; Wi-Fi; wireless sensing

1 Introduction

Wi-Fi has gained an increasing interest in research due to the implementation of orthogonal frequency-division multiplexing (OFDM) and multiple-input multiple-output (MIMO) technology. In telecommunication with high throughput and multiantenna, the channel state information (CSI) can make the transmissions adapt to current channel condition, which is of great significance. CSI characterizes how wireless signals propagate from the transmitter to the receiver at certain carrier frequency of certain communication link. Each CSI entry represents the channel frequency response (CFR), which is shown in Eq. (1).

$$H(f;t) = \sum_{i}^{N} a_i(t) e^{-j2\pi f \tau_i(t)} \quad (1)$$

where, $a_i(t)$ is the amplitude attenuation factor, $\tau_i(t)$ is the propagation delay, and f is the frequency of carrier. For each subcarrier of one link, the channel can be modeled by $y = Hx + n$, where, y is the received signal, x is the transmitted signal, H is the CSI matrix, and n is the environment noise. In this paper, the CSI matrix H is estimated at the receiver side by evaluating the difference between the pre-defined transmitted signal x and received signal y after OFDM demodulation using the Atheros CSI Tool [1].

Wireless sensing based on Wi-Fi signals has caught tremendous attentions due to its ubiquity and privacy-preserving features [2–8]. Many researchers have paid much attention on human crowd counting based on the widely deployed wireless routers in public areas. Human crowd count estimation has also attracted increasing attention in many potential applications, such as intelligent surveillance, crowd management, urban security and business decision-making *etc*. For example, the accurate human population distribution information of one city can bring benefit for the government management personnel to make population-related decisions more efficiently. Since the outbreak of the world-wide novel coronavirus pandemic, crowd counting in public areas, such as in shopping centers and in commercial streets, has gained popularity among public health administrations for preventing the crowds from gathering. Traditionally, image-based methods are most often used to estimate the human crowd count, but they are limited to the illumination intensity of environment, line-of-sight propagation property of light, and the public consideration of privacy [9–18]. In this paper, we introduce an adaptive model for human crowd count estimation by exploiting rich CSI data embedded in 802.11n Wi-Fi networks. To test the robustness of the proposed model, we evaluated its performance in four different scenarios, which are shown in Tab. 1.

Table 1: Four different classes of room situations and the description

Class	Description
1	Empty room
2	With 1 person walking with normal speed
3	With 5 people walking with normal speed
4	With 10 people walking with normal speed

The CSI data is collected from the AR9344 NIC which is embedded in TP-LINK WDR4310 wireless router based on the Atheros CSI Tool.

After collecting the raw CSI data, Kalman filter with Mahalanobis Distance is used to detect abnormality and smooth out the signal [19–20]. Then, the correlation coefficient matrix of subcarriers is calculated for each data link to generate images. In order to extract fine features of the images, convolutional neural network (ConvNet) is used and the trained classification model achieves a satisfying result on the evaluation dataset in the four scenarios [21].

The remainder of the paper is structured as follows. The Section 2 presents the background and related works of crowd counting and Wi-Fi based wireless sensing. The Section 3 presents the system procedure of human crowd counting system, including data collection and analysis, data preprocessing, feature extraction, and construction of classification model. The Section 4 presents the implementation and evaluation of crowd counting system. The Section 5 presents the conclusion.

2 Background and Related Works

In 2015, Gong et al. [22] designed a Wi-Fi-based real-time calibration-free passive human motion detection system based on the physical layer information using two schemes: short-term averaged variance ration (SVR) and long-term averaged variance ration (LVR). According to the experiment result, a high detection rate and low false positive rate are achieved. In 2016, Domenico et al. [23] proposed one trained-once device-free crowd counting and occupancy estimation using Wi-Fi based on a Doppler spectrum approach in WiMob. The proposed approach analyzes the linear correlation relationship between the shape of the Doppler spectrum and the received signal. In 2017, Zhu et al. [24] proposed an abnormal activity detection system NotiFi which achieved satisfactory performance in accuracy, robustness, and stability. It is based on the fact that the amplitude and phase information of CSI change sensitively whenever the human body occludes the wireless signal from the access point (AP) to the monitor point (MP). Yen-Kai et al. extends crowd counting technique to people-centric Internet of Things (IoT) applications, e.g., security monitoring and energy management for smart homes based on fine-grained physical-layer wireless signatures. They achieved an average correct classification rate of 88% in estimating the exact number of the crowd of size up to nine people in general indoor scenarios. In 2014, Xi et al. [25] proposed the Percentage of nonzero Elements (PEM), in the dilated CSI Matrix, and then the monotonic relationship was explicitly formulated by the Grey Verhulst Model. In 2019, Ibrahim et al. [26] proposed CROSS-COUNT, which uses a single Wi-Fi link to estimate the human crowd count based on the temporal link-blockage pattern and achieves a high accuracy with non-labor-intensive data.

3 System Procedure of Human Crowd Count Estimation

3.1 Data Collection and Analysis

Each CSI measurement contains several fields, which are shown in Tab. 2.

Table 2: CSI field and value

Field	Physical meaning
Time stamp	Time stamp of CSI (microsecond)
Csi_len	Length of CSI (Byte)
Channel	Frequency of communication (MHz)
Err_info	Indicator for collection error, Boolean type
Noise_floor	Channel noise
Rate	Rate of communication
Csi	Channel State Information
Num_tones	Number of subcarriers
Nr	Number of the receiver's antennas
Nc	Number of the transmitter's antennas
Rssi	Received Signal Strength Indicator

Each CSI measurement is a $Nr \times Nc \times Numtones$ three-dimensional tensor, where Nr denotes number of antennas of the receiver, Nc denotes number of antennas of the sender, and $Numtones$ denotes number of subcarriers in the frequency band used for communication in the experiment. In this experiment, the

configuration $Nr = 2,\ Nc = 2,\ \ and\ Numtones = 56$. The sampling frequency is 30 Hz, and the sampling duration of each group is 60 s. The experiment was conducted in four different room situations, with the room being empty, with 1 person walking at normal speed, with 5 people walking at normal speed, and with 10 people walking at normal speed. A total of four groups of CSI data were collected and each group contains 1,800 CSI measurements. Fig. 1 shows the amplitude change of 56 subcarriers of 300 CSI packets in empty room situation. Fig. 2 shows the amplitude change of four different communication links between AP and MP of a single subcarrier in empty room situation.

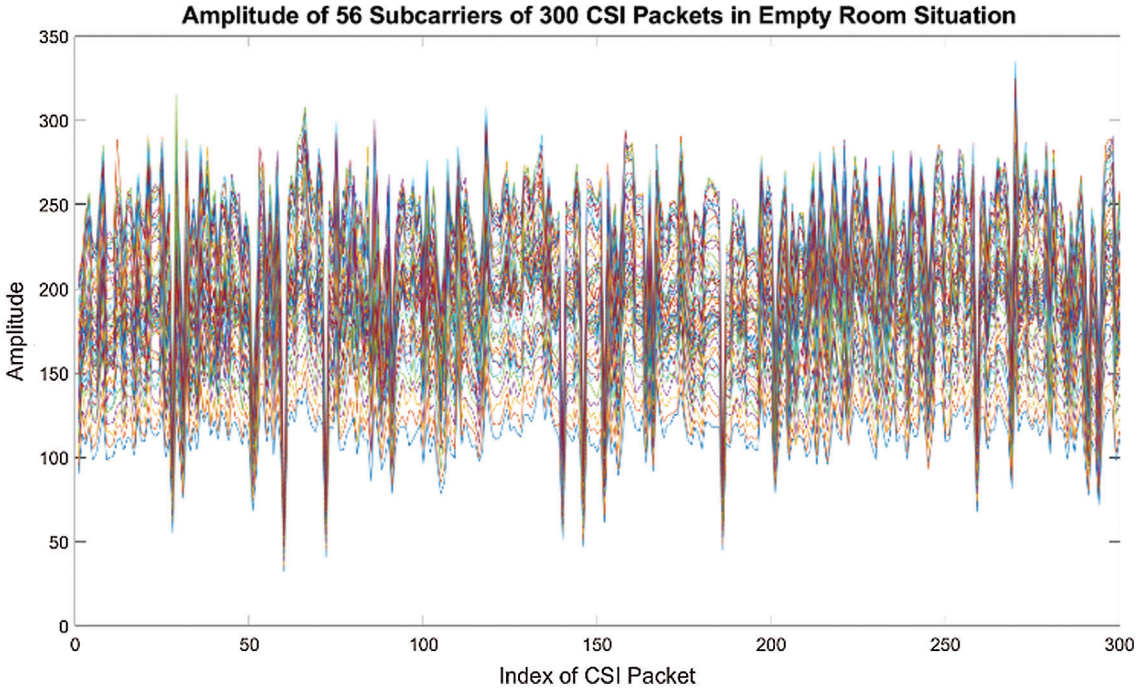

Figure 1: Amplitude of 56 subcarriers of 300 CSI packets in empty room

3.2 Data Preprocessing

Generally, the collected CSI is an estimate of the wireless channel and contains random noise and other inaccuracies. In order to have a better estimate of the wireless channel based on the collected CSI, in this paper, Kalman Filter is used to filter noise and remove outliers. It can be seen in Eqs. (2) and (3).

$$\mathrm{x(t)} = \mathrm{Ax(t-1)} + \mathrm{B(t)u(t)} + \mathrm{w(t)} \tag{2}$$

$$\mathrm{y(t)} = \mathrm{Cx(t)} + \mathrm{v(t)} \tag{3}$$

where, A is one-dimensional state transition matrix and A = [[1.0]] is implemented in our case. $B(t)$ is the influence of the control action at time t, and $u(t)$ is the control vector at time t. In our case, $B(t)$ and $u(t)$ are not implemented. $w(t)$ is the process noise at time t, C is the observation matrix which maps the true state space into the measured space, $v(t)$ is the measurement noise at time t, $x(t)$ is the estimated system state at time t derived from the state at time $t-1$, and $y(t)$ is the measurement at time t. $w(t)$ *and* $v(t)$ are assumed to be drawn from zero mean normal distribution $N(0, R_{ww})$ and $N(0, R_{vv})$ respectively, where R_{ww} denotes covariance of process noise and R_{vv} denotes covariance of measurement noise.

The Kalman filter can be divided into two procedures: “Prediction” and “Update”.

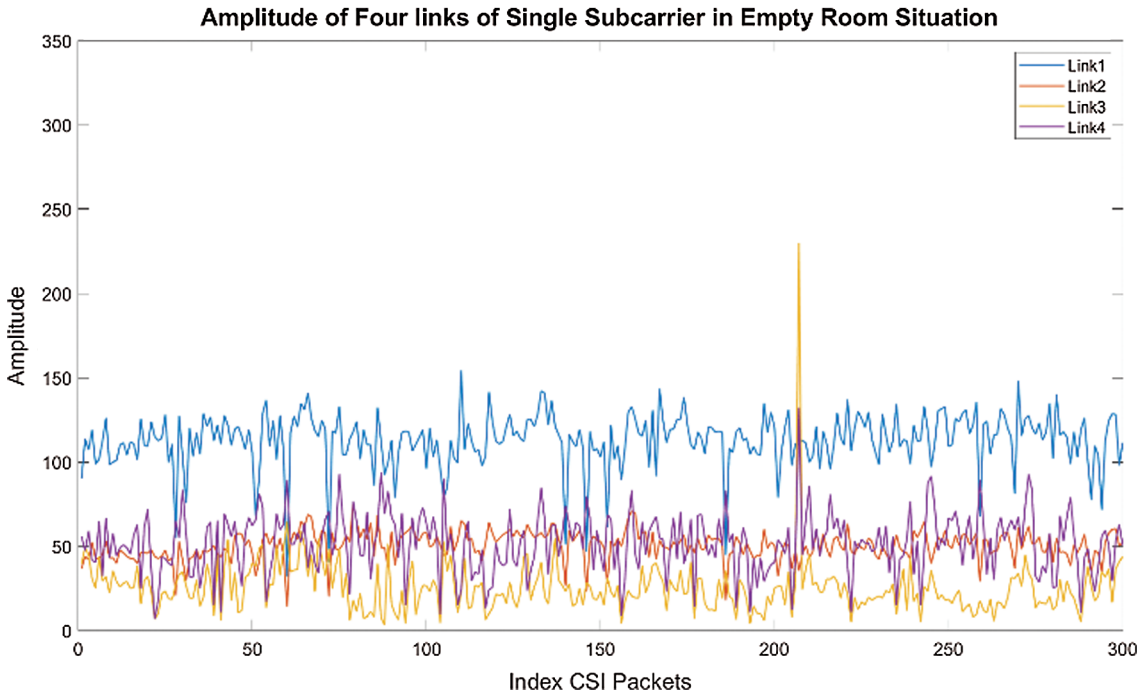

Figure 2: Amplitude of four links of single subcarrier in empty room

Prediction procedure using Eqs. (4) and (5):

$$\hat{\mathrm{x}}(\mathrm{t}|\mathrm{t}-1) = \mathrm{A}(\mathrm{t}-1)\hat{\mathrm{x}}(\mathrm{t}-1|\mathrm{t}-1) + \mathrm{B}(\mathrm{t})\mathrm{u}(\mathrm{t}) \tag{4}$$

$$\hat{\mathrm{P}}(\mathrm{t}|\mathrm{t}-1) = \mathrm{A}(\mathrm{t}-1)\hat{\mathrm{P}}(\mathrm{t}-1|\mathrm{t}-1)\mathrm{A}(\mathrm{t}-1)^{\mathrm{T}} + \mathrm{R}_{\mathrm{ww}}(\mathrm{t}) \tag{5}$$

where, $\hat{x}(t|t-1)$ is the priori estimated state of system at time t given measurement at time $t-1$, $A(t-1)$ is the state transition model at time $t-1$ applied to the previous posteriori estimated state $\hat{x}(t-1|t-1)$, $\hat{P}(t|t-1)$ is the priori estimated covariance, and $R_{ww}(t)$ is the covariance of process noise at time t. The priori state of current time is estimated using the posteriori estimated state from the previous time in the prediction procedure.

Update procedure Eqs. (6)–(10):

$$\mathrm{e}(\mathrm{t}) = \mathrm{y}(\mathrm{t}) - \mathrm{C}(\mathrm{t})\hat{\mathrm{x}}(\mathrm{t}|\mathrm{t}-1) \tag{6}$$

$$\mathrm{R}_{\mathrm{ee}}(\mathrm{t}) = \mathrm{C}(\mathrm{t})\hat{\mathrm{P}}(\mathrm{t}|\mathrm{t}-1)\mathrm{C}(\mathrm{t})^{\mathrm{T}} + \mathrm{R}_{\mathrm{vv}}(\mathrm{t}) \tag{7}$$

$$\mathrm{K}(\mathrm{t}) = \hat{\mathrm{P}}(\mathrm{t}|\mathrm{t}-1)\mathrm{C}(\mathrm{t})^{\mathrm{T}}\mathrm{R}_{\mathrm{ee}}(\mathrm{t})^{-1} \tag{8}$$

$$\hat{\mathrm{x}}(\mathrm{t}|\mathrm{t}) = \hat{\mathrm{x}}(\mathrm{t}|\mathrm{t}-1) + \mathrm{K}(\mathrm{t})\mathrm{e}(\mathrm{t}) \tag{9}$$

$$\hat{\mathrm{P}}(\mathrm{t}|\mathrm{t}) = (\mathrm{I} - \mathrm{K}(\mathrm{t})\mathrm{C}(\mathrm{t}))\hat{\mathrm{P}}(\mathrm{t}|\mathrm{t}-1) \tag{10}$$

where, $e(t)$ denotes the innovation, $R_{ee}(t)$ denotes the innovation covariance, $K(t)$ denotes the optimal Kalman gain, $\hat{x}(t|t)$ denotes the posteriori updated state, and $\hat{P}(t|t)$ denotes the posteriori updated estimate covariance.

Since only the current measurement and the estimated state from the previous time are required to compute the estimate for the current state, Kalman filter is a computationally efficient algorithm for real-time and light-weight applications.

In order to detect and remove outliers, Weighted Mahalanobis Distance $MD(t)$ of a given measurement $y(t)$ and a predicted value $\hat{x}(t|t-1)$ are used in this paper. As shown in Eq. (11):

$$\mathrm{MD(t)} = \sqrt{(\mathrm{y(t)} - \hat{\mathrm{x}}(\mathrm{t|t-1}))^{\mathrm{T}} \mathrm{R_{ee}}^{-1} (\mathrm{y(t)} - \hat{\mathrm{x}}(\mathrm{t|t-1}))} \tag{11}$$

$$\mathrm{R_{vv}} = \frac{\Delta}{1 + \mathrm{e}^{-\mathrm{MD(t)}+\xi}} \tag{12}$$

where, Δ and ξ are constants, which can be determined by analyzing the statistical feature of the signal.

The R_{vv} can be considered as how much the system can trust on the measurement. The bigger the R_{vv} value is, the less trust the system will have on the measurement. The value of R_{vv} can be adaptively updated based on the amount of noise suffered according to Eq. (12) above.

The amplitudes before and after Kalman filtering of the first subcarrier of link 1 in the empty room situation are shown in Fig. 3.

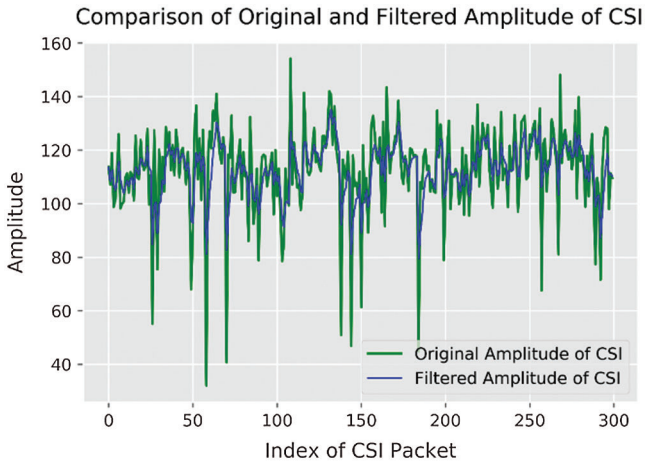

Figure 3: Comparison of original and filtered amplitude of CSI

3.3 Feature Extraction

The correlation coefficient matrix is calculated using Eq. (13).

$$\mathrm{M} = \begin{bmatrix} 1 & \mathrm{Cof}(1,2) & \cdots & \mathrm{Cof}(1,\mathrm{n}) \\ \mathrm{Cof}(2,1) & 1 & \cdots & \mathrm{Cof}(2,\mathrm{n}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cof}(\mathrm{n},1) & \mathrm{Cof}(\mathrm{n},1) & \cdots & 1 \end{bmatrix} \tag{13}$$

where, $Cof(i,j)$ means the Pearson Correlation Coefficient of *ith* subcarrier and *jth* subcarrier, which is calculated using Eq. (14):

$$\rho_{\mathrm{X,Y}} = \frac{\mathrm{cov(X,Y)}}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}} \tag{14}$$

where, cov is the covariance, σ_X is the standard deviation of X, and σ_Y is the standard deviation of Y. The Pearson Correlation Coefficient measures linear combination between two variables X and Y which has a value between -1 and $+1$. A value of -1 means totally negative linear correlation, 0 means no linear correlation between X and Y, and $+1$ means total positive linear correlation.

Considering the tasks of recognizing the number of people in a room, the window size $W = 10s$ and step size $S = 0.5s$ were selected when using the sliding window size method to produce the samples of each scenario for classification. The total number of CSI measures of one scenario is $N = 1800$ and a total number of $(N - W)/S + 1 = 108$ windows can be generated from the collected data of one scenario.

In this paper, only the amplitude information of CSI is used, as the amplitude correlation of subcarriers is sensitive to the number change of people in a closed room based on the experiment. The data shape of single window is $Nr \times Nc \times Numtones \times W$, which is $2 \times 2 \times 56 \times 300$ in this case. For simplicity, select the first antenna of receiver and the first antenna of sender in the beginning and apply the same method to the other three links later. Calculate the Person Correlation Coefficient of any two subcarriers in one window according to Eq. (14).

Generate gray level image with 56×56 pixels from correlation coefficient matrix M. The gray level image of four different classes is shown in Fig. 4. Since there are 2×2 links, the total number of images generated from the collected data is $2 \times 2 \times 108 \times 4 = 1728$.

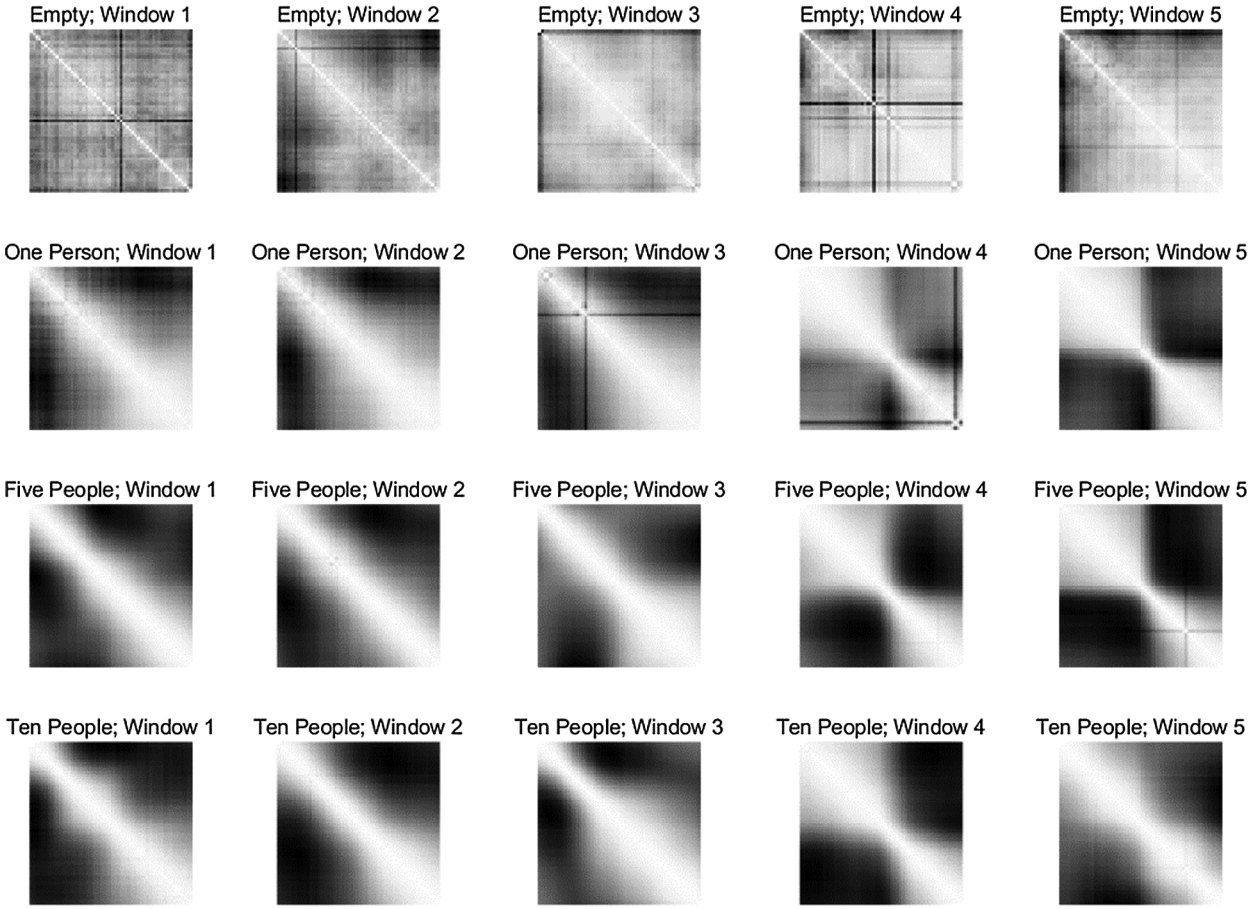

Figure 4: Image of correlation coefficient matrix of four different classes

3.4 Construction of ConvNet Classification Model

The structure of ConvNet constructed in this paper is shown in Tab. 3.

Table 3: Structure of the convolutional neural network

Layer Name	Description
Imageinput	56 × 56 × 1 images with 'zerocenter' normalization
Conv_1	8 3 × 3 × 1 convolutions with stride [1 1] and padding 'same'
batchnorm_1	Batch normalization with 8 channels
Relu_1	ReLU
Maxpool_1	2 × 2 max pooling with stride [2 2] and padding [0 0 0 0]
Conv_2	16 3 × 3 × 8 convolutions with stride [1 1] and padding 'same'
Batchnorm_2	Batch normalization with 16 channels
Relu_2	ReLU
Maxpool_2	2 × 2 max pooling with stride [2 2] and padding [0 0 0 0]
Conv_3	32 3 × 3 × 16 convolutions with stride [1 1] and padding 'same'
Batchnorm_3	Batch normalization with 32 channels
Relu_3	ReLU
Fc	4 fully connected layer
Softmax	Softmax activation function
Classoutput	Cross entropy with 't1' and 3 other classes

3.5 Description of Convnet's Layers and Parameters

3.5.1 The Convolutional Layer

The input is a tensor with shape $(N_I \times H_I \times W_I \times D_I)$, where N_I is the number of images, H_I is the height of the image, W_I is the width of the image, and D_I is the depth of the image. After passing through a convolutional layer, the tensor becomes abstracted to a feature map with shape $(N_I \times FH_I \times FW_I \times FC_I)$, where FH_I is the feature map height, FW_I is the feature map width, and FC_I is the feature map channels. The shape of convolutional kernel is 3×3 for all three convolutional layers and the number of input channels and output channels are $(1, 8)$, $(8, 16)$, $(16, 32)$ for conv_1, conv_2, and conv_3 respectively.

3.5.2 The Polling Layer

Pooling is a form of non-linear down-sampling, which partitions the input image into a set of non-overlapping sub-regions. The max pooling unit uses the function $f = max(A(1,1), A(1,2), \ldots, A(m,n))$, where A denotes the matrix of the sub-region with shape m by n, to generate single value from the partitioned sub-region. Pooling layer can decrease the spatial size of image and reduce the number of parameters significantly. Commonly, the filter with size 2×2 and a stride of 2 along both width and height is selected, and 75% of the activations will be discarded.

3.5.3 The Relu Layer

The rectifier is an activation function defined as Eq. (15).

$$\mathrm{f}(\mathrm{x}) = \max(0, \mathrm{x}) \tag{15}$$

It maps negative values to zero and keeps the non-negative values unchanged. The rectified linear unit increases the nonlinear properties of the decision function.

3.5.4 The Learning Rate

Learning rate is a hyperparameter in an optimization algorithm, which determines the step size at each iteration while moving towards a minimum of the cost function. A high learning rate will probably make the learning jump over the minima. On the opposite, a low learning rate generally takes too much time to converge and even makes the learning progress stuck in the local minimum. Therefore, there should be a trade-off when selecting the learning rate for a specific problem. In this paper, a common value 0.01 of learning rate was selected when training the ConvNet.

3.5.5 Batch Normalization

Batch normalization is a method which uses re-centering and re-scaling to accelerate the training progress and make the neural network more stable. The batch normalization improves the performance by smoothing the objective function.

Batch normalization fixes the means and variances of the inputs of each layer. $\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$, $\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$, where B denotes the mini-batch of size m of the entire training set, μ_B denotes the mean of mini-batch B, and σ_B^2 denotes the variance of mini-batch B. For a ConvNet, whose input layer has the shape $(N_I \times H_I \times W_I \times D_I)$, the batch normalization procedure is shown in Eqs. (16) and (17), and each element in the matrix x should be normalized separately.

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}(1,1) & \mathrm{x}(1,2) & \cdots & \mathrm{x}(1,\mathrm{W_I}) \\ \mathrm{x}(2,1) & \mathrm{x}(2,2) & \cdots & \mathrm{x}(2,\mathrm{W_I}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}(\mathrm{H_I},1) & \mathrm{x}(\mathrm{H_I},2) & \cdots & \mathrm{x}(\mathrm{H_I},\mathrm{W_I}) \end{bmatrix} \tag{16}$$

$$\hat{\mathrm{x}}_\mathrm{i}(\mathrm{j},\mathrm{k}) = \frac{\mathrm{x_i}(\mathrm{j},\mathrm{k}) - \mu_\mathrm{B}(\mathrm{j},\mathrm{k})}{\sqrt{\sigma_\mathrm{B}^2(\mathrm{j},\mathrm{k}) + \varepsilon}} \tag{17}$$

where $j \in [1, W_I]$, $k \in [1, H_I]$ *and* $i \in [1, m]$; $\mu_B(j,k)$ and $\sigma_B^2(j,k)$ are the mean and variance of each element in the matrix x respectively; ε is an arbitrarily small constant added for numerical stability. In the end, the $\hat{x}_i(j,k)$ will have zero mean and unit variance.

3.5.6 Softmax Function

SoftMax function is a generalized multiple dimensions version of logistic function which is a common S-shape curve. The equation of logistic function is $f(x) = \frac{L}{1+e^{-k(x-x_0)}}$, where x_0 is the value of the midpoint, L is the curve's maximum value and k is the logistic steepness of the curve. When $x_0 = 0,\ L = 1,\ k = 1, f(x)$ is the standard logistic function. Similarly, SoftMax function takes as input of a vector v and normalizes it into a probability distribution. After the normalization, each component in v will be in range $(0, 1)$ and all components will sum up to 1. Typically, the value of component in v can be interpreted as probability and the larger value corresponds to higher probability. The SoftMax function $\sigma : R^K \to R^K$ can be

defined as follows: $\sigma(v)_i = \frac{e^{v_i}}{\sum_{j=1}^{K} e^{v_j}}$, *for* $i = 1, 2, \ldots, K$ *and* $v = (v_1, v_2, \ldots, v_K) \in R^K$, where K is the dimension of input vector v.

4 Implementation and Evaluation

4.1 Layout of Experiment Classroom

The experiment was conducted in a university classroom and the layout is shown in Fig. 5. The MP was set in the front of the classroom and the AP was set in the back. The distance between AP and MP is 10 m. Students of certain number walked with normal speed in the aisle. The AP is controlled remotely from outside of the classroom to collect CSI data.

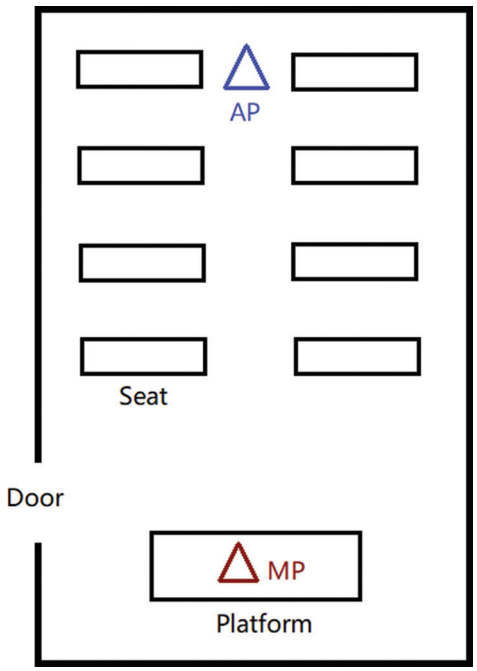

Figure 5: Layout of the experiment environment

4.2 Specification of the Experiment Device

In this experiment, one TL-WDR4310 wireless router flashed with customized OpenWRT firmware was used to collect CSI data. Tab. 4 displays the specifications of the experiment device.

4.3 Atheros-CSI-Tool

The CSI data was collected using the Atheros-CSI-Tool which is an open source 802.11n measurement and experimentation tool. Based on this tool, detailed PHY wireless communication information was extracted from the Atheros Wi-Fi NICs, including CSI, data rate, the received packet payload, RSSI, etc. All functionalities of Atheros-CSI-Tool are implemented in software without any modification of the firmware. In this experiment, Atheros-CSI-Tool was implemented in the Wi-Fi router with customized OpenWRT firmware.

Table 4: Specifications of the experiment device TL-WDR4310

Brand and model	TP-Link TL-WDR4310 v1.0
Soc	Atheros AR9344
CPU frequency (MHz)	560
Flash size (MB)	8
RAM size (MB)	128
WLAN hardware	Atheros AR9344, Atheros AR9580
Supported mode of WLAN 2.4	b/g/n
Supported mode of WLAN 5.0	a/n
Architecture of processor	MIPS 74Kc
Wireless no 1	SoC-integrated: Atheros AR9340 2.4GHz
Wireless no 2	Separate Chip: Atheros AR9580 5GHz

4.4 Training the ConvNet Classification Model

The ConvNet is implemented using MATLAB Deep Learning Toolbox. Fig. 6 is the graph of training progress.

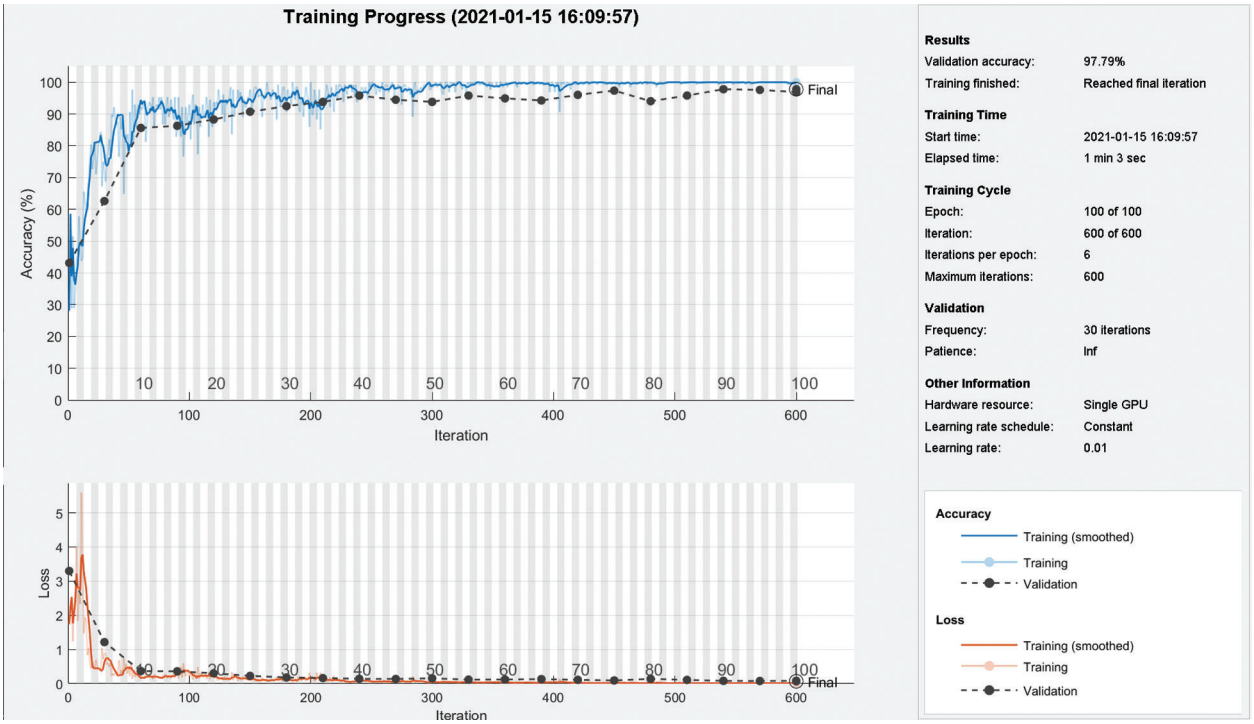

Figure 6: Training progress of the ConvNet with 100 epochs

4.5 Evaluation

Fig. 7 is the algorithm for estimating crowd count.

Algorithm 1: Main Procedure of Evaluating Human Density Using the Trained Model

Input: Collected CSI: $\boldsymbol{D}$; Trained Model: $\boldsymbol{M}$;
Output: The Class Label of Each Window of the Input CSI: $\boldsymbol{C}$

$\boldsymbol{Initialize}$: $\boldsymbol{D_L}$: The Unprocessed CSI and $\boldsymbol{D_L} = \boldsymbol{D}$; $\boldsymbol{D_W}$: The Selected CSI of One Window; $\boldsymbol{CofM}$: The Coefficient Matrix of the Subcarriers of Each Window $\boldsymbol{D_W}$;
while $\boldsymbol{D_L} \neq \varnothing$ **do**
  Select $\boldsymbol{D_W}$ from $\boldsymbol{D_L}$;
  Update $\boldsymbol{D_L}$ *to* $\boldsymbol{D_L} - \boldsymbol{D_W}$;
  Calculate the Correlation Matrix $\boldsymbol{CofM}$ of 56 Subcarriers from $\boldsymbol{D_W}$;
  Evaluate the Class Label $\boldsymbol{C}$ of the $\boldsymbol{CofM}$ Using the Trained Model $\boldsymbol{M}$;
  Output the Class Label $\boldsymbol{C}$;

Figure 7: Algorithm: main procedure of evaluating crowd count using the trained model

The confusion matrix of the evaluation is shown in Fig. 8. It can be seen that the model shows a perfect accuracy when recognizing in Classes 1 and 2 and makes minimal mistakes when distinguishing Class 3 with Class 4. The overall accuracy in all four classes is 97.8%, while the accuracy of recognizing in Classes 1 and 2 is 100% and the accuracy of recognizing in Classes 3 and 4 is 94.5% and 97.7% respectively.

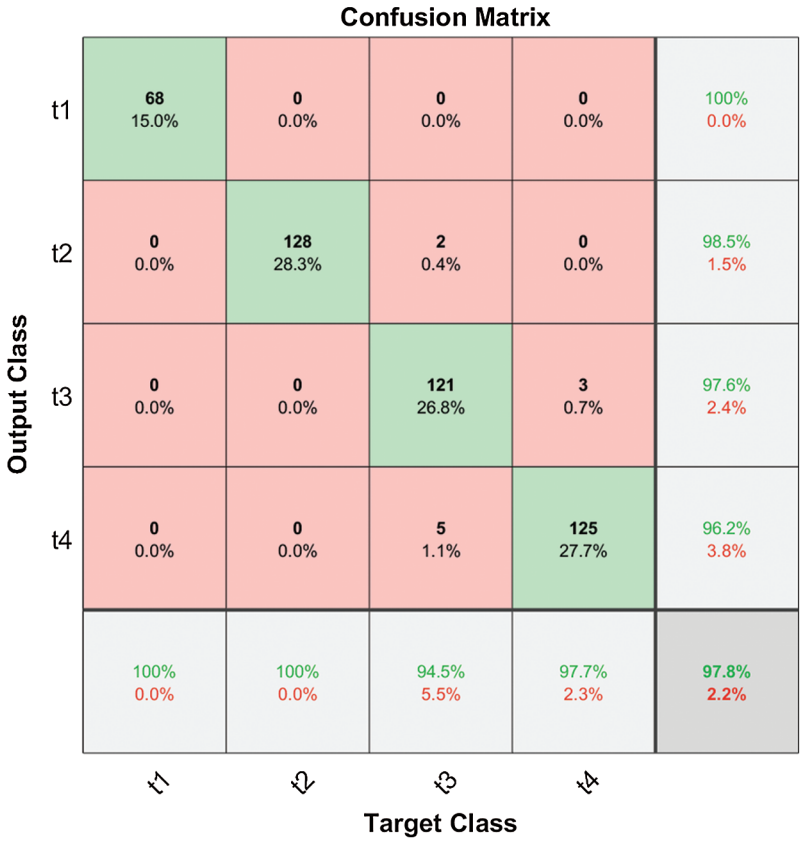

Figure 8: Confusion matrix of the prediction accuracy

Two different methods are compared with our proposed method. The comparison bar graph of overall accuracy is shown in Fig. 9. The Threshold-based methods utilize statistical property of the amplitude of CSI, such as variance and mean to recognize the number of people. The Eigenvalue-based methods extract the first several maximum eigenvalues of the correlation matrix of subcarriers. Support Vector Machine implemented with LIBSVM is used to train and evaluate the two methods above [27].

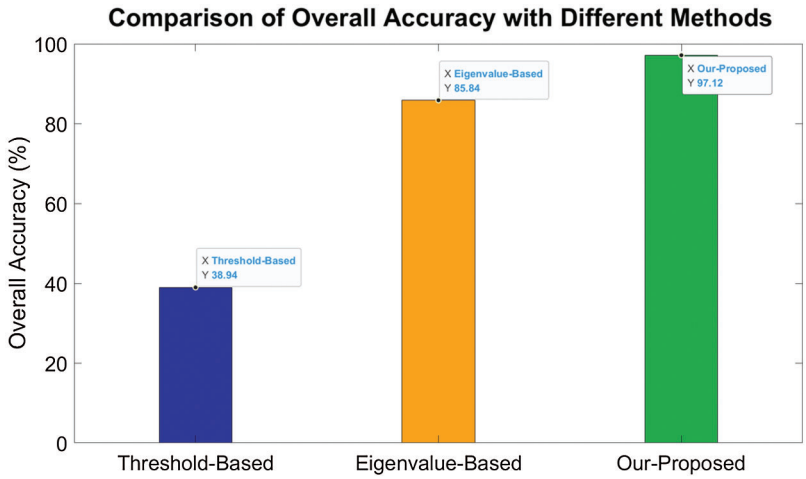

Figure 9: Comparison of overall accuracy with different methods

Fig. 10 shows the accuracy of recognizing each class with different methods. It is observed that Threshold-based method almost fails when deployed into different environments except for Scenario 3. The Eigenvalue-based method still shows relatively high performance but the accuracy of recognizing each scenario is lower than our proposed method.

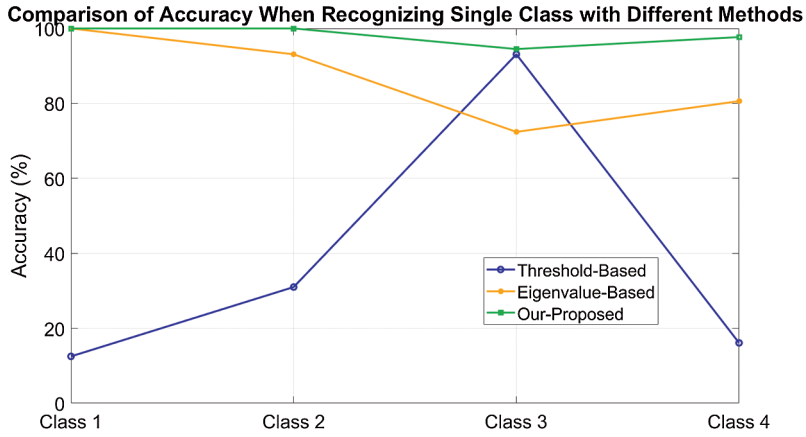

Figure 10: Comparison of accuracy when recognizing single class with different methods

5 Conclusion

In this paper, we presented the design, implementation, and evaluation of a novel lightweight and adaptive passive crowd counting method based on ConvNet. The system addresses the challenges found in the literature such as lack of robustness, low generalization ability, and high computational cost. The main idea is to generate images with fairly low resolution from the correlation coefficient matrix and

classify the small images with a relative shallow ConvNet. With only one pair of AP and MP deployed, an overall accuracy of 97.79% is achieved when experimenting with the number of people into four levels.

Currently, we are extending the method to estimate the number of people up to 20 with multiple APs and MPs deployed in public areas.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 61802196, url: http://www.nsfc.gov.cn/); Jiangsu Provincial Government Scholarship for Studying Abroad; The Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD); NUIST Students' Platform for Innovation and Entrepreneurship Training Program (Grant No. 202010300080Y, url: http://sjjx.nuist.edu.cn:81/CXCY/NUIST/).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Xie, Z. Li and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. of the 21st Annual Int. Conf. on Mobile Computing and Networking*, New York, NY, USA, pp. 53–64, 2015.

[2] Y. Cheng and R. Y. Chang, "Device-free indoor people counting using Wi-Fi channel state information for internet of things," in *2017 IEEE Global Communications Conf.*, Marina Bay Sands, MBS, Singapore, pp. 1–6, 2017.

[3] P. Wang, B. Guo, T. Xin, Z. Wang and Z. Yu, "TinySense: Multi-user respiration detection using Wi-Fi CSI signals," in *2017 IEEE 19th Int. Conf. on e-Health Networking, Applications and Services*, Dalian, China, pp. 1–6, 2017.

[4] L. Gong, W. Yang, Z. Zhou, D. Man, H. Cai *et al.,* "An adaptive wireless passive human detection via fine-grained physical layer information," *Ad Hoc Networks*, vol. 38, no. 8, pp. 38–50, 2016.

[5] S. Palipana, P. Agrawal and D. Pesch, "Channel state information based human presence detection using non-linear techniques," in *Proc. of the 3rd ACM Int. Conf. on Systems for Energy-Efficient Built Environments*, New York, NY, USA, pp. 177–186, 2016.

[6] S. Palipana, D. Rojas, P. Agrawal and D. Pesch, "FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–25, 2018.

[7] K. Qian, C. Wu, Z. Yang, Y. Liu and Z. Zhou, "PADS: Passive detection of moving targets with dynamic speed using PHY layer information," in *2014 20th IEEE Int. Conf. on Parallel and Distributed Systems*, Hsinchu, Taiwan, pp. 1–8, 2014.

[8] X. Wang, C. Yang and S. Mao, "ResBeat: Resilient breathing beats monitoring with realtime bimodal CSI data," in *2017 IEEE Global Communications Conf.*, Marina Bay Sands, Singapore, pp. 1–6, 2017.

[9] R. Chen, L. Pan, C. Li, Y. Zhou, A. Chen *et al.,* "An improved deep fusion CNN for image recognition," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1691–1706, 2020.

[10] D. T. Nguyen, W. Li and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition*, vol. 51, no. 3, pp. 148–175, 2016.

[11] S. Li, J. Xue and Y. Han, "No-reference stereoscopic image quality assessment based on local to global feature regression," in *2019 IEEE Int. Conf. on Multimedia and Expo*, Shanghai, China, pp. 448–453, 2019.

[12] H. Kyu Shin, Y. Han Ahn, S. Hyo Lee and H. Young Kim, "Digital vision based concrete compressive strength evaluating model using deep convolutional neural network," *Computers, Materials & Continua*, vol. 61, no. 3, pp. 911–928, 2019.

[13] V. Sheng and J. Zhang, "Machine learning with crowdsourcing: A brief summary of the past research and future directions," in *Proc. of the AAAI Conf. on Artificial Intelligence*, San Francisco, USA, vol. 33, pp. 9837–9843, 2019.

[14] Z. Xiao, B. Yang and D. Tjahjadi, "An efficient crossing-line crowd counting algorithm with two-stage detection," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 1141–1154, 2019.

[15] R. Chen, G. Zeng, K. Wang, L. Luo and Z. Cai, "A real time vision-based smoking detection framework on DDGE," *Journal on Internet of Things*, vol. 2, no. 2, pp. 55–64, 2020.

[16] Z. Pan, X. Yi, Y. Zhang, B. Jeon and S. Kwong, "Efficient in-loop filtering based on enhanced deep convolutional neural networks for HEVC," *IEEE Transactions on Image Processing*, vol. 29, pp. 5352–5366, 2020.

[17] Z. Pan, X. Yi, Y. Zhang, H. Yuan, F. L. Wang *et al.,* "Frame-level bit allocation optimization based on video content characteristics for HEVC," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 1–20, 2020.

[18] L. Shen, X. Chen, Z. Pan, K. Fan, F. Li *et al.,* "No-reference stereoscopic image quality assessment based on global and local content characteristics," *Neurocomputing*, vol. 424, no. 5, pp. 132–142, 2021.

[19] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[20] R. Kalman, "On the general theory of control systems," *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110, 1959.

[21] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 Int. Conf. on Engineering and Technology*, Antalya, Turkey, pp. 1–6, 2017.

[22] L. Gong, W. Yang, D. Man, G. Dong, M. Yu *et al.,* "WiFi-based real-time calibration-free passive human motion detection," *Sensors (Basel)*, vol. 15, no. 12, pp. 32213–32229, 2015.

[23] S. D. Domenico, G. Pecoraro, E. Cianca and M. D. Sanctis, "Trained-once device-free crowd counting and occupancy estimation using WiFi: A doppler spectrum based approach," in *2016 IEEE 12th Int. Conf. on Wireless and Mobile Computing, Networking and Communications*, New York, NY, USA, pp. 1–8, 2016.

[24] H. Zhu, F. Xiao, L. Sun, R. Wang and P. Yang, "R-TTWD: Robust device-free through-the-wall detection of moving human with WiFi," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1090–1103, 2017.

[25] W. Xi, J. Zhao, X. Li, K. Zhao, S. Tang *et al.,* "Electronic frog eye: Counting crowd using WiFi," in *2014 IEEE INFOCOM*, Toronto, Canada, pp. 361–369, 2014.

[26] O. T. Ibrahim, W. Gomaa and M. Youssef, "CrossCount: A deep learning system for device-free human counting using WiFi," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9921–9928, 2019.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.