LocCams: An Efficient and Robust Approach for Detecting and Localizing Hidden Wireless Cameras via Commodity Devices

YANGYANG GU, Wuhan University, China
JING CHEN, Wuhan University, China
CONG WU, Nanyang Technological University, Singapore
KUN HE, Wuhan University, China
ZIMING ZHAO, University at Buffalo, United States
RUIYING DU, Wuhan University, China

Unlawful wireless cameras are often hidden to secretly monitor private activities. However, existing methods to detect and localize these cameras are interactively complex or require expensive specialized hardware. In this paper, we present LocCams, an efficient and robust approach for hidden camera detection and localization using only a commodity device (e.g., a smartphone). By analyzing data packets in the wireless local area network, LocCams passively detects hidden cameras based on the packet transmission rate. Camera localization is achieved by identifying whether the physical channel between our detector and the hidden camera is a Line-of-Sight (LOS) propagation path based on the distribution of channel state information subcarriers, and utilizing a feature extraction approach based on a Convolutional Neural Network (CNN) model for reliable localization. Our extensive experiments, involving various subjects, cameras, distances, user positions, and room configurations, demonstrate LocCams' effectiveness. Additionally, to evaluate the performance of the method in real life, we use subjects, cameras, and rooms that do not appear in the training set to evaluate the transferability of the model. With an overall accuracy of 95.12% within 30 seconds of detection, LocCams provides robust detection and localization of hidden cameras.

CCS Concepts: • **Security and privacy** → *Privacy protections.*

Additional Key Words and Phrases: Hidden Camera Detection and Localization, Channel State Information, Deep Learning

ACM Reference Format:
Yangyang Gu, Jing Chen, Cong Wu, Kun He, Ziming Zhao, and Ruiying Du. 2023. LocCams: An Efficient and Robust Approach for Detecting and Localizing Hidden Wireless Cameras via Commodity Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 160 (December 2023), 24 pages. https://doi.org/10.1145/3631432

Authors' addresses: Yangyang Gu, guyangyang@whu.edu.cn, Wuhan University, Wuhan, Hubei, China; Jing Chen, chenjing@whu.edu.cn, Wuhan University, Wuhan, Hubei, China; Cong Wu, cnacwu@whu.edu.cn, Nanyang Technological University, Singapore, Singapore; Kun He, hekun@whu.edu.cn, Wuhan University, Wuhan, Hubei, China; Ziming Zhao, zimingzh@buffalo.edu, University at Buffalo, Buffalo, United States; Ruiying Du, duraying@whu.edu.cn, Wuhan University, Wuhan, Hubei, China.

2474-9567/2023/12-ART160 $15.00
https://doi.org/10.1145/3631432

1 INTRODUCTION

The widespread availability and affordability of wireless cameras have led to an increase in the deployment of unlawful hidden cameras in private spaces, such as hotel rooms, compromising individuals' privacy [1, 10, 21–23, 29, 44, 46]. According to a survey conducted in 2019 among 2,000 American travelers, it was found that 11% of the respondents had encountered hidden cameras in rental accommodations in the past [18]. Furthermore, reports have revealed the existence of a sophisticated cybercrime industry associated with hidden camera surveillance, with Chinese law enforcement uncovering 30,000 hidden cameras in just four months in 2022 [2, 35]. Consequently, there is an urgent need to develop effective and efficient approaches for detecting and localizing hidden cameras. In this paper, the term *detect* refers to the knowledge that a WiFi camera is monitoring the room, while *localize* entails identifying the specific area where the camera is located (e.g., on the ceiling), similar to the terms used in SnoopDog [42].

However, existing hidden camera detection and localization methods suffer from limitations such as complexity, high costs of dedicated hardware, and significant user efforts [4, 25–27, 30, 36, 37, 40, 42, 45, 51, 52, 57]. Some methods can only detect the presence of hidden cameras without providing localization [4, 5, 8, 25, 26, 30, 31, 36, 54]. Additionally, existing camera localization methods either require specialized probing equipment or are limited to two-dimensional plane localization [16, 17, 40, 42, 57]. For example, SnoopDog [42] plays specific videos using a laptop in different directions for camera localization. Lumos [40] and SCamF [17] require the user with the detector to walk along the edge of the room to achieve two-dimensional camera localization but ignore the case of the camera on the ceiling, where they may be ineffective since the variation of the indicator (e.g., video frame sizes) for localization is tiny in this walking mode. Moreover, some methods [27, 37, 45, 52] require meticulous close scanning of the entire surroundings for suspicious optical reflections. This scanning process demands users to possess expert knowledge in detecting hidden cameras, thereby posing a challenge in uncovering cameras that are concealed in high places (e.g., ceilings). These constraints limit the usability of existing methods and highlight the need for more cost-effective and user-friendly solutions for detecting and localizing hidden cameras.

To address these challenges and enable efficient, low-cost detection and localization of hidden cameras, two practical considerations must be taken into account. Firstly, users are more likely to have commodity devices rather than expensive specialized probing equipment. Secondly, since users typically have limited knowledge of detecting and locating hidden cameras, the detection process should be straightforward, providing direct indications of the suspected area within a room. Therefore, the localization technology must be affordable and easily deployable on readily available commodity devices.

In this paper, we present LocCams, an efficient and robust approach for hidden camera detection and localization through wireless data-packet analysis via commodity devices. LocCams leverages traffic analysis and Line-of-Sight (LOS) propagation identification to directly assist users in localizing hidden cameras. Note that the definition of LOS propagation is narrow and refers to the case where the user holds the detector directly in front of the camera, while Non-LOS (NLOS) refers to the case where the user turns their back to the camera [6]. The user, equipped with a commodity device as the detector, performs four stationary-to-turn actions where the user performs a right turn after standing stationary for approximately five seconds, enabling simultaneous hidden camera detection and localization. The turning actions leverage the changes in the camera's Packet Transmission Rate (PTR) [40] to determine if a hidden camera is monitoring the user, as cameras require additional packets to transmit pixel changes resulting from turning actions [4, 42]. Additionally, the turning actions play a crucial role in determining whether there is LOS propagation between the hidden camera and the detector, which can be characterized by Channel State Information (CSI) [11]. However, existing methods for LOS identification based on CSI either require an active connection to the transmitter or are specifically designed to identify large obstacles such as walls [6, 62], which limit their use in our scenario. Therefore, we propose a new lightweight Convolutional Neural Network (CNN) model for learning-based LOS identification. This model extracts representative features

Table 1. Acronyms along with Brief Explanations.

Acronym	Description	Utility
PTR	Packet Transmission Rate	It is associated with human motion and is used to indicate whether the camera is monitoring the user.
LOS	Line of Sight	A description of the camera packet transmission path where the human body does not block the transmission between the camera and the detector.
CSI	Channel State Information	It can be used to identify the LOS path between transceivers since it can reflect the state of the physical channel.
WPA	WiFi Protected Access	It is a security certification program to secure wireless networks and can prevent users from accessing the network without knowing the password.
VBR	Variable Bit Rate	Different bit rate coding is provided depending on the complexity of the multimedia content, which is an important reason for the PTR variation.
CFR	Channel Frequency Response	A description of the multiple path propagation of a signal.
OFDM	Orthogonal Frequency Division Multiplexing	A technology for transmitting wireless signals. CSI is estimated from the preamble of the OFDM packet.
PDF	Probability Density Function	It is used to describe the different distributions of CSI measurements under LOS conditions and NLOS conditions.
MAC	Media Access Control	A unique identifier is assigned to a network interface controller for use as a network address in communications within a network segment. It can be used to group collected WiFi packets as well as to distinguish between different cameras detected.
OUI	Organizationally Unique Identifier	The first half of the MAC address uniquely identifies a vendor, manufacturer, or other organization. It can be used to assist in recognizing cameras based on their MAC addresses.

from the distribution of CSI subcarrier measurements. By analyzing the results of LOS identification, we can effectively localize the hidden camera. To evaluate the performance of the method in real life, we use subjects, cameras, and rooms that do not appear in the training set to evaluate the transferability of the model. Evaluation results demonstrate that a new user can directly use the already trained model to localize a new camera in a new room. Some acronyms along with brief explanations are summarized in Table 1 to improve the comprehensibility of this paper. Our contributions are as follows.

- We present LocCams, an efficient and robust method for detecting and localizing hidden wireless cameras using commodity devices. It does not rely on specialized hardware and does not require users to execute complex interactions in the detection process.
- We develop an efficient adaptive packet-rate identification algorithm to determine whether a device is a camera actively monitoring the user.

- We design a learning-based method for localizing hidden cameras by identifying the LOS propagation between the hidden camera and the detector. It utilizes a lightweight CNN model to extract the distribution features of subcarriers from each CSI measurement.
- We evaluate the performance of LocCams using nine wireless cameras in eight different rooms under various conditions, including different users, distances, initial positions, and camera heights. The results demonstrate that LocCams achieves an overall accuracy of 95.12% for localizing different cameras and can localize a hidden camera within a room within 30 seconds of detection, even when the camera and room data are not part of the training dataset.

2 BACKGROUND

In this section, we discuss the related work of LocCams as well as background knowledge on the packet transmission rate of WiFi cameras, the wireless signal propagation model, and the distribution of CSI subcarriers under LOS and NLOS conditions.

2.1 Hidden Camera Detection and Localization

To address the privacy threat of hidden cameras, researchers have explored various methods to detect and localize them.

Methods based on lens reflection: Some efforts [27, 37, 45, 52] utilize smartphones or light-emitting devices to scan objects in the environment at close range, aiming to detect unusual specular reflections caused by hidden cameras. However, this method requires expertise and is ineffective for cameras placed in high locations, such as ceilings.

Methods based on electromagnetic/thermal emission: CamRadar [26] employs electromagnetic detection devices to capture electromagnetic radiation emitted by hidden cameras at close range. Similarly, some methods [57, 64] use specialized external infrared camera modules to detect anomalous infrared radiation emitted by objects, enabling the detection and localization of hidden cameras. These methods require expensive specialized equipment and significant user effort during the detection process.

Methods based on CSI: DeepDeSpy [8] develops a deep learning-based model to detect hidden cameras by analyzing the change in CSI resulting from human activity. DeSpy [36] detects the existence of a spy camera by exploiting the correlation between video bitrate changes and CSI changes. However, these methods lack the capability to precisely localize hidden cameras.

Methods based on wireless traffic analysis: Traffic analysis is commonly used for camera detection and localization due to distinctive characteristics in the encoding of video and audio frames [4]. Some methods, such as [4, 5, 25, 30, 31, 54], can detect hidden cameras using only statistics of encrypted traffic packets. Others design specific traffic stimuli, such as human activity or light changes, to detect and localize hidden cameras [16, 17, 40, 42]. However, these methods have limitations. For instance, SnoopDog [42] requires the user to play a specific video for 30 seconds on a laptop facing different directions to detect cameras in the area, which is time-consuming and burdensome. MotionCompass [16] locates cameras with motion sensors but requires the user to execute specific routes several times, making it difficult for users. Lumos [40] and SCamF [17] assume cameras are installed on walls or corners and use the Received Signal Strength Indicator (RSSI) and video frame sizes to localize cameras in a two-dimensional plane. However, their effectiveness may be limited when cameras are placed on ceilings, where the change in distance is small. Furthermore, if the device actively manipulates its traffic statistics by performing operations like traffic padding, this method simply becomes ineffective.

Table 2. Existing Schemes for Hidden Camera Localization vs. LocCams.

Scheme	Usability[1]	Non-specialized device	Reliability[2]	Localization dimension	Time[3]
HeatDeCam [57]	✓	×	✓	3-dimensional	N/A
Lumos [40]	✓	×	×	2-dimensional	30 min.
SnoopDog [42]	×	×	✓	2-dimensional	70 sec.
MotionCompass [16]	×	✓	✓	3-dimensional	150 sec.
LAPD [37]	×	✓	✓	3-dimensional	60 sec.
LocCams	✓	✓	✓	**3-dimensional**	**30 sec.**

[1] Usability: whether it eliminates the user to execute labor-intensive operations.
[2] Reliability: whether it is reliable under different settings in the wild.
[3] Time: the time required to complete a single hidden camera detection and localization process.

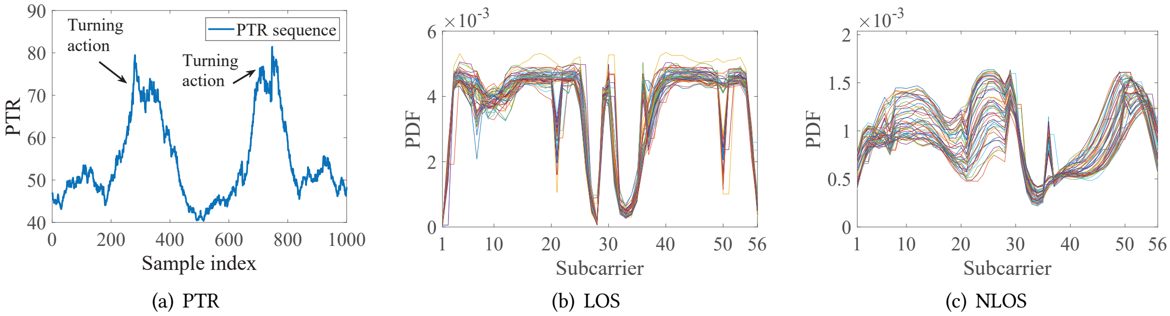
2.2 LOS Path Identification

LOS path identification plays a critical role in improving the performance of various wireless applications [33]. Several studies have been devoted to identifying the propagation path between the transmitter and the receiver. Some methods, such as [55, 62], primarily focus on identifying large obstacles like walls. Meanwhile, other techniques, such as [6, 9, 13, 24, 39], rely on specialized devices or have demanding requirements. For example, methods like [6, 9, 60] require the receiver to be actively connected to the Access Point (AP), while the method in [24] necessitates obtaining RSSI values from both 2.4GHz and 5GHz bands of the same transmitter. The method in [39] is based on the power-delay profile, which has limitations in resolution at narrow bandwidth and significant offsets with small phase errors [56]. However, such requirements are impractical in passive indoor scenarios, where wireless cameras typically operate in low bandwidth 2.4GHz networks, and our receiver cannot actively connect to the camera.

In contrast to the aforementioned methods, as illustrated in Table 2, we have developed a lightweight, cost-effective, and minimally interactive method for detecting and localizing hidden cameras. Our CSI-based localization approach remains effective even for devices with altered traffic statistics characteristics, such as traffic padding. With the help of LocCams, users can quickly determine the presence or absence of the wireless camera monitoring in their vicinity and localize the camera in a 3-dimensional area using only a commercial smartphone, requiring approximately 30 seconds of interaction.

2.3 Packet Transmission Rate of WiFi Camera

Traffic analysis has emerged as a prevalent approach for detecting hidden wireless cameras. Despite the encryption of wireless network traffic through protocols like WiFi Protected Access (WPA), certain meta-information about the traffic remains accessible to all devices in the environment. One commonly used meta-information is the PTR [40], which refers to the number of packets transmitted by the camera per second. PTR is particularly sensitive to motion activities due to the Variable Bit Rate (VBR) [48] encoding typically employed by most WiFi cameras. Frames with significant pixel changes are assigned a higher PTR to ensure smoother video transmission within limited bandwidth. Conversely, frames with minimal pixel changes are assigned a lower PTR. To validate the sensitivity, we collect CSI data packets from a camera when it monitors two turning actions. We calculate the PTR from timestamps of the CSI data packets using a sliding window with a window size of 100 packets and a step size of 1 packet. As shown in Fig. 1(a), when a user performs a turning action, it leads to a higher PTR, while a stationary body results in a lower PTR. The results indicate that PTR can be used to detect hidden cameras.



(a) PTR (b) LOS (c) NLOS

Fig. 1. (a) PTR sequence resulted from two turning actions where PTRs are calculated with timestamps of every 100 CSI data packets. (b) (c) Distribution of CSI subcarriers for LOS (b) and NLOS (c) conditions caused by the human body. (b) (c) show 50 CSI measurements for both LOS and NLOS conditions respectively, with each line representing a measurement.

2.4 Wireless Signal Propagation Model

During signal propagation, wireless signals traverse multiple paths with varying attenuation and delays, influenced by environmental factors such as reflection, scattering, and diffraction. The Channel Frequency Response (CFR) is a representation of multiple-path propagations [47]. Specifically, if a signal with frequency f reaches the receiver through M different paths, the CFR can be denoted as:

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

where $a_i(t)$ is the complex-valued representation of the overall attenuation due to the antenna pattern of transceivers, the nature of the reflector, and the length of the propagation path of the i_{th} path, and $\tau_i(t)$ is the time-varying delay of the i_{th} path.

In the IEEE 802.11.x protocol family, Orthogonal Frequency Division Multiplexing (OFDM) technology is employed to transmit wireless signals using orthogonal subcarriers with different central frequencies. By leveraging the preamble of the OFDM packet, which contains detailed CSI values at different subcarriers, the received CSI measurements can be utilized to estimate the CFR in time and frequency domains [14, 32]. Concretely, a CSI measurement $H(t)$ comprises CFR values of N carrier frequencies at time t:

$$H(t) = (H(t;f_1), H(t;f_2), \cdots, H(t;f_N)) \tag{2}$$

where $H(t;f_N)$ can be denoted as $H(t;f_N) = |H(t;f_N)|e^{i\angle H(t;f_N)}$ with the amplitude $|H(t;f_N)|$ and the phase $\angle H(t;f_N)$. Since we just utilize CSI amplitude in this paper, the term $H(t)$ in the rest of this paper denotes the CSI amplitude.

2.5 Distribution of CSI Subcarriers under LOS and NLOS Conditions

CSI is known for its sensitivity to the displacements and movements of transmitters, receivers, surrounding objects, and humans [28]. As a result, CSI has found wide applications in wireless sensing, including human activity identification [12, 41, 50, 53, 58, 59, 61], human localization [3, 33, 34, 63], and LOS identification [6, 49, 62]. The LOS path refers to the direct path between a transmitter and a receiver, traveling the shortest distance without additional losses due to reflection, diffraction, or scattering. By observing whether the LOS path between a hidden camera and the detector is obstructed, it becomes possible to determine the direction of the hidden camera.

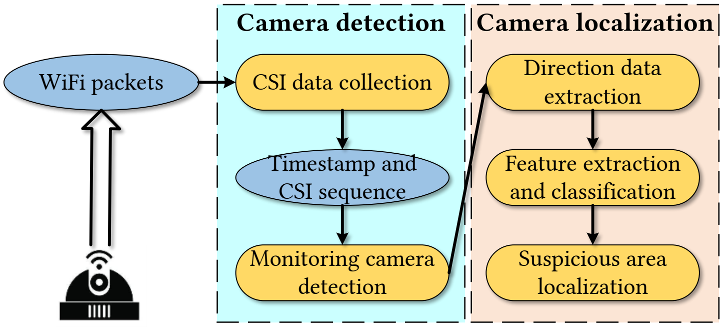

Camera detection
Camera localization
WiFi packets
CSI data collection
Timestamp and CSI sequence
Monitoring camera detection
Direction data extraction
Feature extraction and classification
Suspicious area localization

Fig. 2. Workflow of LocCams.

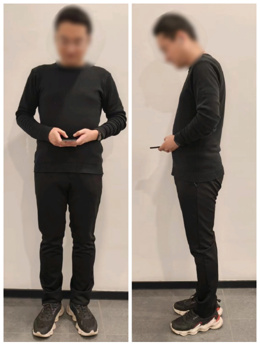


Fig. 3. Schematic diagram of the turning action.

To validate this concept, we collect CSI data under both LOS and NLOS conditions. In the LOS scenario, a subject holding the detector directly faces the wireless camera, while in the NLOS scenario, the subject turns his back to the camera. The wireless camera is connected to a 2.4GHz WiFi network with a 20MHz bandwidth, and a total of 56 subcarriers are available for data process [7, 11]. Fig. 1(b) and (c) depict the Probability Density Functions (PDFs) of 56 subcarriers for 50 CSI measurements under the two conditions. The significant differences between the LOS and NLOS distributions are evident, while the LOS/NLOS distributions remain consistent. Motivated by this observation, we investigate the distribution of CSI subcarriers to determine the direction of the hidden camera relative to the human body through LOS identification without the need for specialized hardware or aimless scanning, setting our approach apart from existing methods.

3 ATTACK MODEL AND SYSTEM OVERVIEW

In this section, we present the attack model and system overview.

3.1 Attack Model

The goal of the attacker is to install a hidden camera in a private space (e.g., a hotel/rental room) to surreptitiously capture the user's private information. The attacker may be the owner of the room or a previous customer, so he has plenty of time to set up the room and equipment in advance. The camera is connected to the Internet via a WiFi network. Specifically, we assume that the attacker's capabilities and constraints are as follows: (i) The attacker can change the layout and furnishings of the room to hide the camera appropriately. For example, the camera can be hidden behind a planter. (ii) The attacker can set up a separate WiFi network for the hidden camera that is not visible to the user. (iii) Similar to prior work [4, 25, 30, 31, 40, 43, 57], the attacker can modify the software configurations (e.g., resolution) and shape of the hidden camera. For example, the camera can be disguised as a socket [37]. However, the attacker is not capable of hacking the camera firmware to change the wireless transmission behavior.

3.2 System Overview

LocCams is designed to assist users in detecting the presence of hidden wireless cameras and accurately localizing them. It consists of two stages: camera detection and camera localization, as illustrated in Fig. 2.

In the camera detection stage, users are first instructed to perform four turns, collecting CSI data packets from active devices in the environment. By analyzing the characteristics of PTR calculated from packet timestamps, LocCams determines the presence of malicious hidden cameras.

In the camera localization stage, LocCams first extracts four sequences corresponding to the four turns in different directions from the raw CSI sequence. A lightweight learning-based feature extractor is employed to extract the features from these four sequences. Next, using a voting algorithm, LocCams determines which sequence is collected under LOS conditions based on the extracted CSI features, thereby identifying the area of the hidden camera localization.

4 CAMERA DETECTION

In this section, we detail how this system quickly discovers camera traffic based on the PTR. Specifically, we contribute a new method to identify the wireless camera based on the PTR.

4.1 CSI Data Collection

During the process of collecting CSI data packets, the user needs to perform continuous turning actions simultaneously. Here's a detailed procedure:

- The user selects an initial position in the room and holds the detector flat on the chest, as shown in Fig. 3.
- The data collection is initiated, and after approximately 5 seconds of stillness, the user starts turning to the right while following the instructions on the screen.
- The user performs four stationary-to-turn actions to complete a data collection and returns to the initial orientation. The data collection is automatically stopped at this point.
- The collected packets are then grouped based on their source MAC addresses to detect and localize the possible presence of multiple hidden cameras.

LocCams extracts two sequences, $R_T = (t_1, t_2, \cdots, t_K)$ and $R_{CSI} = (H(t_1), H(t_2), \cdots, H(t_K))$, from one group of data packets. Here, K is the number of collected packets, R_T represents the timestamp sequence, and R_{CSI} represents the raw CSI amplitude sequence of data packets.

4.2 Monitoring Camera Detection

To detect the presence of a hidden camera that is actively monitoring the user, we propose an adaptive packet-rate identification algorithm using the timestamp sequence R_T as input. The algorithm, outlined in Algorithm 1, performs the following steps:

Filtering based on mean PTR. The algorithm filters out the device with small traffic whose mean PTR is less than a threshold value v, since cameras typically require a certain PTR rate for real-time transmission of audio and video (lines 1-3). Specifically, we consider the mean PTR of a camera should satisfy the following condition:

$$K/t_K >= v \tag{3}$$

where K is the number of data packets collected, t_K is the last timestamp of CSI data packets collected, and v is the minimum PTR of a camera operating at low resolution.

PTR calculation. For each timestamp sequence R_T, the algorithm calculates the PTR every m timestamps and obtains the PTR sequence V_T (line 4). This step can be denoted as:

$$V_T = CalRate(R_T, m) \tag{4}$$

where $CalRate$ represents the function to calculate the PTR every m timestamps.

Division into turning sequences. The sequence V_T is divided into four PTR sequences, each corresponding to the four turning actions performed by the user (line 5). Each turning action results in a unique change in the camera's PTR. As shown in Fig. 4, PTR values are calculated from timestamps of every 100 CSI data packets collected from four different cameras (i.e., Y3, C2HC, H6C, and V380) listed in Table 3, while the camera is monitoring the subject. The dashed boxes in the figure indicate the unique PTR changes resulting from the subject's turning

Algorithm 1 Identify the monitoring state using PTR

```
Input: Timestamp sequence $R_T = (t_1, t_2, \cdots, t_K)$, threshold of mean PTR $v$, step $m$ for calculating PTR, threshold
    of adaptive ratio $\eta$
Output: Presence of camera monitoring (True) or absence of camera monitoring (False)
1:  if $K/t_K < v$ then
2:      return $False$
3:  end if
4:  $V_T \leftarrow CalRate(R_T, m)$    // $CalRate(R_T, m)$ is to calculate PTR from the timestamp sequence $R_T$ in steps
    of $m$ and outputs the PTR sequence $V_T$.
5:  $P_1, P_2, P_3, P_4 \leftarrow V_T$    // $P_i$ represents the quarter of $V_T$.
6:  for $i \leftarrow 1$ to 4 do
7:      $max_i \leftarrow max(P_i)$    // $max_i$ is the maximum PTR of the sequence $P_i$.
8:      $rem_i \leftarrow mean(P_i')$    // $P_i'$ is the PTR sequence after $P_i$ removes $max_i$ and $rem_i$ is the average PTR of
        $P_i'$.
9:  end for
10: $a_m \leftarrow mean(max_1, max_2, max_3, max_4)$    // $a_m$ is the average value of four maximum PTRs
    $max_1, max_2, max_3$, and $max_4$.
11: $a_r \leftarrow mean(rem_1, rem_2, rem_3, rem_4)$    // $a_r$ is the average value of four average PTRs
    $rem_1, rem_2, rem_3$, and $rem_4$.
12: if $(a_m - a_r)/a_r <= \eta$ then
13:     return $False$
14: else
15:     return $True$
16: end if
```

actions. Since the time intervals for all turning actions are the same, we can divide the PTR sequence V_T into four equal parts, denoted as P_1, P_2, P_3, and P_4.

Ratio calculation. In this step, we calculate the ratio between the difference of the mean rate a_m of maximum rates in four equal parts and the mean rate a_r of the remaining rates in four equal parts, divided by a_r (lines 6-12). As illustrated in Fig. 4, the maximum rate in each equal part exhibits significant differences compared to the other rates, which is a characteristic of the turning action. Furthermore, for the calculation of the ratio, we specifically choose a_m and a_r instead of the mean rate of the first four maximum rates in V_T and the mean rate of the remaining rates. This choice ensures that the ratio does not become too small, as this might occur when the first four maximum rates come from only three or fewer turning actions. The comparison can be denoted as:

$$(a_m - a_r)/a_r > \eta \tag{5}$$

where η represents the predefined threshold. The observed traffic associated with the R_T is considered to belong to a camera actively monitoring the user if its calculated ratio satisfies Eq. 5 (lines 12-16).

By analyzing the variations in the packet rates during the turning actions, the algorithm identifies cameras that are actively monitoring the user. The corresponding CSI sequence R_{CSI} associated with R_T is then passed to the camera localization phase.

5 CAMERA LOCALIZATION

In this section, we address the problem of indoor camera localization by formulating it as a classification task, where the goal is to identify the LOS propagation path between the camera and the detector. Specifically, LocCams

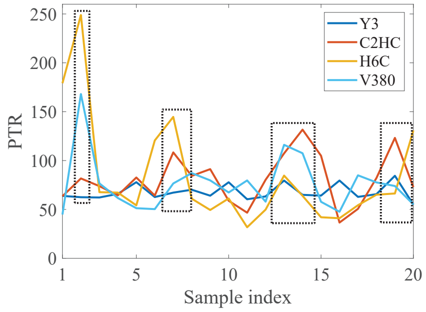


Fig. 4. PTRs calculated with timestamps of every 100 CSI data packets transmitted from different cameras when a subject performs four turning actions. Y3, C2HC, H6C, and V380 represent four cameras (c.f., Table 3). The dashed box indicates the unique change of PTR resulting from the turning action.

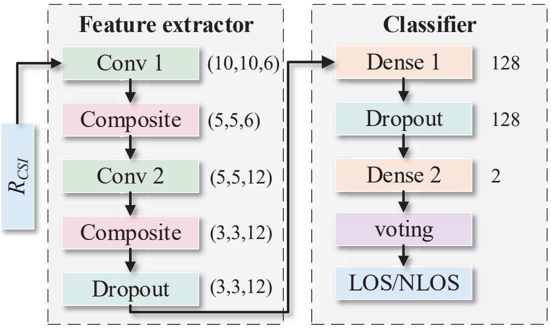


Fig. 5. Architecture of the CNN model. *Conv 1* and *Composite* represent the first 2-dimension convolution layer and the first composite layer, respectively. A composite layer contains a *BatchNormalization* layer, a *relu* activation layer, and a *MaxPool* layer. (10,10,6) represents the output shape of this layer.

determines the relative orientation by identifying the presence or absence of LOS propagation between the camera and the detector. Unlike existing methods for LOS identification [6, 9, 62], LocCams introduces a novel learning-based method that exhibits strong robustness in camera localization. Moreover, if LocCams detects multiple suspected cameras in the camera detection module, LocCams can simultaneously and independently analyze each CSI data stream and localize them based on the source MAC addresses.

5.1 Direction Data Extraction

As mentioned in Section 2.4, the distribution of CSI subcarrier measurements can provide valuable information about LOS propagation. Therefore, LocCams employs a segmentation algorithm (Algorithm 2) to extract four sequences when the user is stationary from raw CSI amplitude sequence, based on the similarity among CSI measurements [15]. This is because the CSI subcarriers exhibit a minimal change in the time domain when the device and the surrounding environment are stationary.

The segmentation algorithm utilizes the correlation ($CSIcorr$) between adjacent CSI measurements (lines 1-3), calculated using the Pearson correlation coefficient:

$$CSIcorr = corr(H(t_i), H(t_{i+1})) \tag{6}$$

where $corr$ represents the function for calculating the correlation coefficient. σ is the threshold for determining the motion state based on the correlation of the CSI measurements. For example, the experiment in [15] indicated that the detector is moving when $CSIcorr$ is less than 0.9. The algorithm identifies the starting point of a stationary sequence as the first point where the correlation coefficient $C(PointIndex)$ is not less than σ, and the end point as the first point where the coefficient is less than σ and the distance len from the last starting point is greater than k (lines 5-8).

By applying this process four times, four temporary sequences when the user is stationary are obtained. For each temporary sequence, a sliding window of size k is used to select a subsequence with the largest sum of correlation coefficients. This subsequence is then chosen as the input for feature extraction (lines 9-11). These four subsequences are denoted as *S1*, *S2*, *S3*, and *S4*.

Algorithm 2 Extract CSI sequences

```
Input: Raw CSI amplitude sequence $R_{CSI} = (H(t_1), H(t_2), \cdots, H(t_K))$, the threshold of correlation $\sigma$, length of a sequence $k$
Output: $S_1, S_2, S_3, S_4$    // The sequences are collected when the human body is stationary.
for $i \leftarrow 2$ to $K$ do
    $C[i-1] \leftarrow corr(H(t_{i-1}), H(t_i))$    // $corr(H(t_{i-1}), H(t_i))$ is to calculate the correlation of two adjacent CSI amplitude measurement.
end for
// Look for four sequences of start and end points.
for $i \leftarrow 1$ to 4 do
    $P[i_{start}] \leftarrow PointIndex\ IF\ C[PointIndex] >= \sigma$    // Start point.
    $P[i_{end}] \leftarrow PointIndex\ IF\ C[PointIndex] < \sigma$ and $i_{end} - i_{start} > k$    // End point.
end for
// Extract four sequences from raw CSI amplitude sequence $R_{CSI}$ according to the start and end point and the sequence length $k$.
for $i \leftarrow 1$ to 4 do
    $j_{max} \leftarrow max(C[P[j] : P[j+k-1]]), P[i_{start}] <= j <= P[i_{end} - k + 1]$    // Find the start point $j_{max}$ of the most correlated sequence.
    $S_i \leftarrow R[P[j_{max}] : P[j_{max} + k - 1], :]$    // Extract the sequence from $R_{CSI}$ according to $j_{max}$ and $k$.
end for
return $S_1, S_2, S_3, S_4$
```

5.2 Feature Extraction and Classification

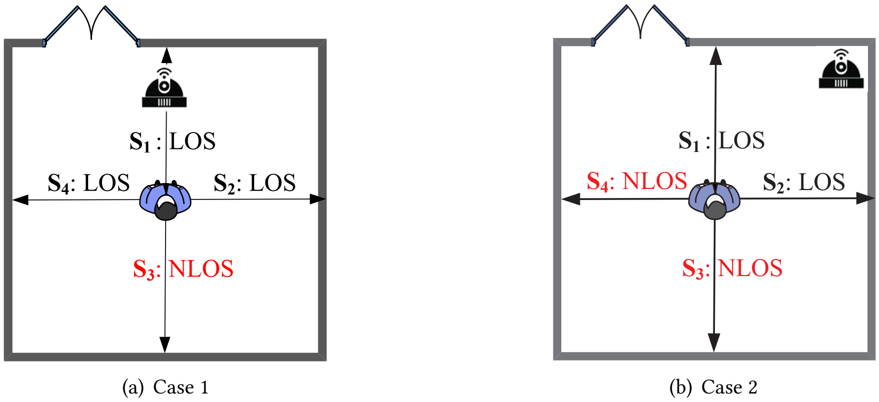
Since a CSI measurement consists of multiple subcarriers, which represent the aggregation of multipath components (as discussed in Section 2.4), we can consider a CSI measurement with multiple subcarriers as analogous to an image with multiple RGB pixel points. Therefore, we employ a CNN model, widely used in image processing, for LOS identification. The model, as shown in Fig. 5, consists of a feature extractor module and a classifier module.

Feature extractor module. Initially, LocCams performs subcarrier selection to filter out subcarriers with significant noise, based on our observation and guidance from the CSI extraction tool [38]. After subcarrier selection, a CSI measurement contains n subcarriers. To enhance the richness of input information, we concatenate two identical measurements into one vector, which can be expressed as:

$$x_i = (H(f_i, t_1), \cdots, H(f_i, t_n), H(f_i, t_1), \cdots, H(f_i, t_n)). \tag{7}$$

As depicted in Fig. 5, the architecture consists of two convolutional layers with a kernel size of 5×5 and a padding type of *SAME*. The first and second convolutional layers have 6 and 12 filters, respectively. Each convolutional layer is followed by a batch normalization operation, which aids in accelerating deep network training [19]. Subsequently, a downsampling layer follows each normalization operation, with a pool size of 2×2, a padding type of *SAME*, and strides of 2. The final downsampling layer is succeeded by a dropout layer with a rate of 0.6 to mitigate overfitting. The feature extractor generates a set of feature representations for each input CSI measurement.

Classifier module. As shown in Fig. 5, the classifier comprises two fully-connected layers responsible for generating the predicted result based on the feature representations. The output dimensions of these layers are 128 and 2, respectively. To facilitate this process, a flattening layer is inserted before the first fully-connected layer to transform the feature representation into a one-dimensional form. Considering that different devices and



(a) Case 1

(b) Case 2

Fig. 6. (a) Three consecutive directions are identified as LOS paths. (b) Two consecutive directions are identified as LOS paths.

environments may exhibit varying levels of noise, which can influence the distribution of the CSI measurement, the classifier incorporates a dropout layer between the two fully-connected layers to address the issue of overfitting. During the testing phase, we feed the four k-dimensional stationary sequences into the model separately. The raw output of each sequence is a k-dimensional vector consisting of only 0 and 1, where 0 represents NLOS path identification and 1 represents LOS path identification. To make a final classification decision, the classifier employs a majority voting algorithm on the output vector. If the value is not less than half of k, the sequence is classified as being collected under the LOS path. Otherwise, the sequence is classified as being collected under the NLOS path.

5.3 Suspicious Area Localization

Based on the identification results of the four sequences, we can localize hidden cameras. Out of the possible combinations of results when the user's body blocks the LOS propagation, only four combinations are considered valid, while the rest are deemed invalid, prompting the user to perform another interaction to collect CSI data if needed. Two typical cases are illustrated in Fig. 6.

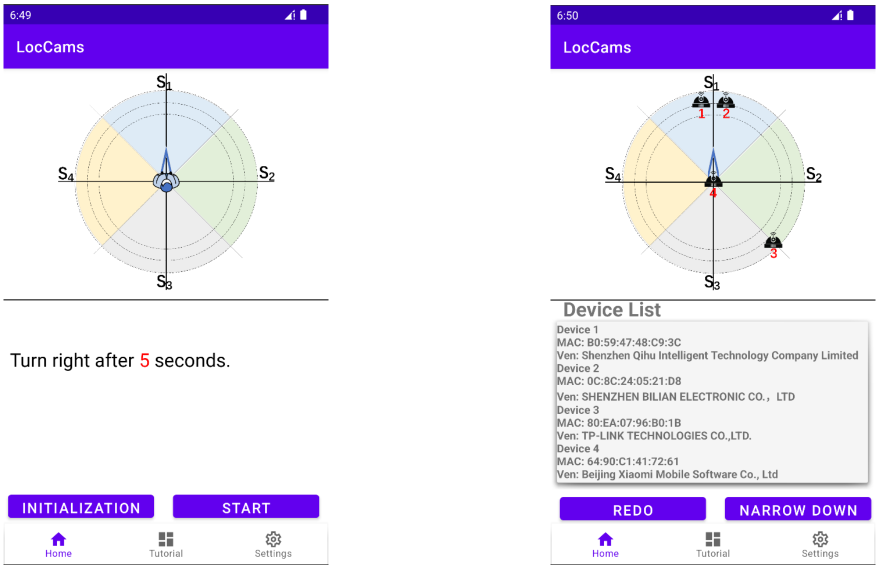
Case 1: In this case, one of the four propagation paths is classified as the NLOS path, as shown in Fig. 6(a). This result indicates that the camera is localized in the area of 90 degrees with S_1 direction as the angle bisector.

Case 2: In this case, two consecutive paths out of the four propagation paths (e.g., S_1 and S_2) are considered LOS paths, as depicted in Fig. 6(b). This result suggests that the camera is localized in the area between the two directions S_1 and S_2 and is most likely positioned along the angle bisector.

Case 3: If all four propagation paths are classified as LOS paths, it indicates that the camera is directly above the user (e.g., on the ceiling).

Case 4: If all four propagation paths are considered NLOS paths, it implies that there is an obstacle completely blocking the camera's propagation path to the user (e.g., the camera is localized in another room).

A single localizing process of about 30 seconds can narrow down the camera's suspicious position to a quarter of the room area, assuming the user's initial position is in the center of the room. In the case of a smaller room, this narrowed-down area is sufficient for the user to find the hidden camera. If the user desires to further narrow down the suspicious area, he can perform the localization process again by changing the initial direction (e.g., offset by 45 degrees) under the guidance of the LocCams Application. The smaller area is highly manageable for finding the hidden camera.



Turn right after 5 seconds.

Device List

Device 1
MAC: B0:59:47:48:C9:3C
Ven: Shenzhen Qihu Intelligent Technology Company Limited
Device 2
MAC: 0C:8C:24:05:21:D8
Ven: SHENZHEN BILIAN ELECTRONIC CO., LTD
Device 3
MAC: 80:EA:07:96:B0:1B
Ven: TP-LINK TECHNOLOGIES CO.,LTD.
Device 4
MAC: 64:90:C1:41:72:61
Ven: Beijing Xiaomi Mobile Software Co., Ltd

(a)

(b)

Fig. 7. An example UI of hidden camera detection and localization in LocCams Application. (a) The initialization and start of hidden camera detection and localization. (b) The result of hidden camera detection and localization.

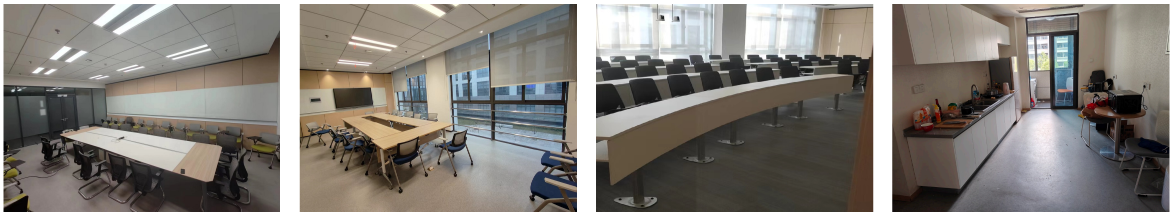
6 EVALUATION

In this section, we present a detailed evaluation of LocCams, including the environmental setup and various performance metrics. We first introduce the room layout, devices, subjects, and evaluation metrics in the environmental setup section. Next, we present the evaluation results for the parameter study, the overall performance of hidden camera detection and localization, and the impact of user's initial distances and positions, the camera position height, and the relative angle between the camera position and the user position on camera localization.

6.1 Environmental Setup

Room layout. Our experiments involve a total of eight rooms including the meeting room (i.e., Room 1, Room 2, Room 5, and Room 6), the discussion room (i.e., Room 3), the classroom (i.e., Room 4), and the domestic room (i.e., Room 7 and Room 8). The specific size and layout of some rooms are shown in Fig. 8.

LocCams Application. We implement LocCams on a Nexus 5 smartphone. Using the nexmon_csi tool [11], we set the network card on the phone to the monitoring mode, allowing it to passively sniff WiFi packets in the environment. The User Interface (UI) is displayed in Fig. 7. When a user wants to detect and localize possible hidden cameras in a room, he can press the *INITIALIZATION* button after selecting the initial position to calibrate the orientation to turn on the monitoring mode of the network card. Next, the user presses the *START* button and follows the on-screen timing prompts to perform the stationary-to-turn action as shown in Fig. 7(a). After completing a detection and localization process with four stationary-to-turn actions, LocCams displays the MAC addresses of the detected hidden cameras and their positions relative to the user on the screen. As shown in Fig. 7(b), camera 1 and camera 2 are localized in the area of 90 degrees with S_1 direction as the angle bisector (Case 1). Camera 3 is localized in the area between the two directions S_2 and S_3 and is most likely positioned along the angle bisector (Case 2). Camera 4 is localized on the ceiling overhead (Case 3). At this point, the user has the option *REDO* to perform the detection and localization process all over again or the option *NARROW*



(a) Room 1. (b) Room 2. (c) Room 3. (d) Room 4.

Fig. 8. Four rooms used in our experiments. The sizes of the four rooms are 8.9m×5.9m, 8.8m×4.5m, 14.5m×9m, and 5m×3.5m. The size and layout of Room 5 and Room 6 is the same as Room 1. The size and layout of Room 7 is the same as Room 2. Room 8 is a 3.6m×2.5m bedroom.

Table 3. The Specific Cameras Used in Our Experiments.

Camera	Abbreviation	Mean Packet Per Second	Cost
XiaoMi Cloud Camera	MI	54	$ 28.9
XiaoYi Smart Camera Y3	Y3	71	$ 23.1
EZVIZ C2HC	C2HC	63	$ 28.9
EZVIZ H6C	H6C	72	$ 30.4
TP-Link TL-IPC43AN-4	C43	34	$ 23.1
360 Cloud Camera 6C	6C	83	$ 24.6
360 D806 Cloud Camera	D806	56	$ 24.6
V380 Camera	V380	38	$ 4.4
Camera 83do	83do	34	$ 21.7

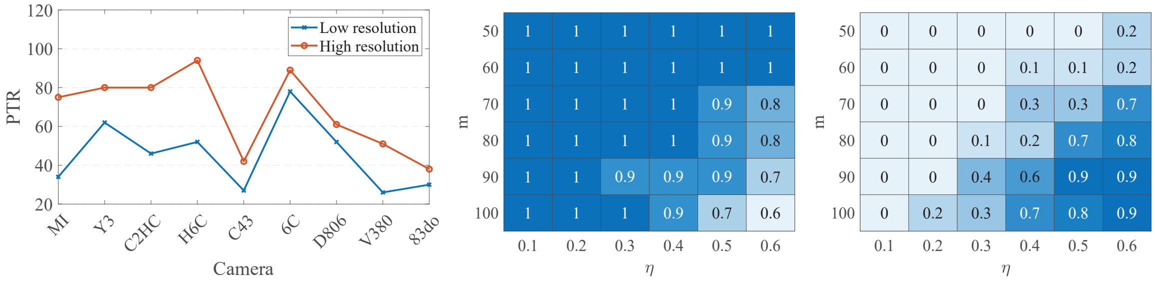
DOWN to further narrow down the localization area. The interpretation of the camera's position is explained in detail in the *tutorial* menu of the LocCams Application, similar to Section 5.3.

Device. We select 9 different popular cameras available on the shopping platforms such as Jingdong and Taobao, as shown in Table 3. During data collection, we only adjust the cameras to the appropriate mode without modifying any hardware settings. The camera is connected to a hidden 2.4GHz WiFi network, and we collect the CSI data emitted by these cameras multiple times over seven months.

Subjects. To validate the effectiveness of LocCams in real-life scenarios, we invite 10 volunteers to use LocCams to localize hidden cameras within the room. The volunteers include 5 men and 5 women aged between 20 and 28. They vary in height and weight and have no expertise in camera localization.

Evaluation metrics. We use the True Positive Rate (TPR) and True Negative Rate (TNR) to evaluate the camera detection. TPR is defined as the correct detection ratio of positive samples collected when a camera is working, i.e., $\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, where TP is the number of positive samples that are correctly detected. FN is the number of positive samples that are falsely detected as negative; TNR is defined as the correct detection ratio of negative samples collected when the camera is not monitoring the user, i.e., $\mathrm{FPR} = \frac{\mathrm{TN}}{\mathrm{TN+FP}}$, where TN is the number of negative samples that are correctly detected. FP is the number of negative samples that are falsely detected as positive.

Localizing camera performance can be divided into three categories. The *first category* is three-dimensional spatial point localization. For example, HeatDeCam [57] can directly display the position (e.g., in a picture frame) of the camera on the screen. This type of scheme requires specialized equipment or significant user effort in the detection process [16, 37, 57]. The *second category* is two-dimensional spatial point localization. These



(a) PTR

(b) TPR

(c) TNR

Fig. 9. (a) PTRs calculated with timestamps of CSI data packets transmitted from different cameras under high and low resolution when a subject performs the turning action. (b) (c) TPR and TNR under different values of m and η. The data is from camera MI.

schemes [17, 40] assume that the camera is only on the surrounding walls and do not take into account the fact that the camera is mounted on the ceiling and use the flat distance between the presumed position and the actual position as an evaluation metric. However, this distance varies greatly depending on the user's walking style, the size of the room, and the type of camera [40]. Therefore, even if the user knows the location, it is still necessary to search in a certain area and may mislead the user by not taking into account that the camera is mounted on the ceiling. The *third category* is three-dimensional spatial area localization. This type of scheme (e.g., [42]) instructs the user that the camera is located in a three-dimensional area (e.g., the area in front of them). It does not mislead the user to ignore the ceiling area and the suspicious area can be narrowed down to a sufficiently small area with several relocalization operations. The localization in this paper is of the third category. Specifically, we use *Area Detection Rate* (*ADR*) to evaluate the camera localization. *ADR* is the probability of correctly judging the area where the camera is localized in M independent camera detection and localization trials, i.e., ADR = $\frac{D}{M}$, where D is the number of trials correctly judging the target area.

6.2 Parameter Study

In this section, we investigate the impact of the parameters of v, m, η, and σ on the performance of the camera detection and localization, as discussed in Sections 4 and 5. The key system parameters are listed in Table 4.

Parameter v. The threshold v is used to filter out devices with small traffic (e.g., smart doorbells) to reduce computational complexity without missing cameras. To determine an appropriate value for v, we collect two samples under high and low resolutions of different cameras when the camera is monitoring a subject performing the stationary-to-turn action. The mean PTR of these samples across different cameras is shown in Fig. 9(a). For the low-resolution samples, the PTRs of cameras in Table 3 are 34, 62, 46, 52, 27, 78, 52, 26, and 30, while for the high-resolution, the PTRs are 75, 80, 80, 94, 42, 89, 61, 51, and 38. Considering that missing hidden cameras is not acceptable, we set v to 20 to ensure sensitivity to low rates of low-resolution cameras.

Parameters η and m. We evaluate the impact of different values of the parameters η and m on camera detection. We ask a subject to perform 10 trials with/without a camera monitoring the subject, respectively. Then, we vary η from 0.1 to 0.6 in step 0.1 and m ranges from 50 to 100 in step 10. The evaluation results of detecting the presence/absence of camera monitoring for camera MI are shown in Fig. 9(b) and 9(c). As the values of η and m increase, TPR tends to decrease, but it can still reach 0.9 when the values of η and m are 0.5 and 90 as shown in Fig. 9(b), On the other hand, TNR tends to increase and can reach 0.9 when the values of η and m are 0.6 and 100 as shown in Fig. 9(c). Since it is more important to detect the camera monitoring the user than the camera not

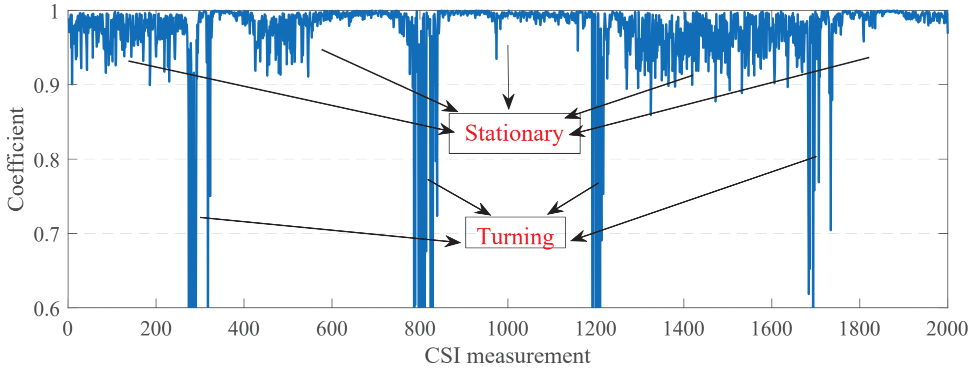


Fig. 10. Correlation coefficient between CSI measurements collected from a whole camera detection and localization process including four stationary-to-turn actions.

Table 4. Parameters Used in Our Experiments.

Parameters	Value	Parameters	Value
Threshold of mean PTR v	20	Threshold of correlation σ	0.9
Size of PTR calculation m	80	Minimum length k of a subsequence S_i	100
Threshold of identifying cameras η	0.5	Number n of CSI subcarriers as input	50

monitoring the user, we finally set the values of η and m as 0.5 and 80 considering the performance of detecting other cameras.

Parameter σ. The threshold σ indicates the motion state of the detector [15]. To determine an appropriate value for σ, a subject performs a whole camera detection and localization process including four stationary-to-turn actions. We calculate the correlation coefficients of collected CSI measurements using Eq. 6, as shown in Fig. 10. When the subject holding the detector is stationary, most values of the correlation coefficient are greater than 0.9. Therefore, We set the value of σ as 0.9 in this paper.

6.3 The Performance of Detecting the Hidden Camera

To evaluate the performance of the wireless camera detection method, we first create a query table of Organizationally Unique Identifier (OUI) information using a survey method, similar to prior work [16, 42]. Specifically, we count the brands of wireless cameras on major shopping platforms (e.g., Amazon, Jingdong, and Taobao), and the brand information is used to statistically record OUI information at MAC address query sites [20]. Additionally, we add the OUIs of some WiFi chip manufacturers to the table, as some cameras (e.g., V380) use MAC addresses that correspond to WiFi chips.

A subject performs 10 trials for each camera in Room 1 while being monitored by the camera. To verify whether our algorithm can detect cameras that are not monitoring the user, the subject also performs 10 trials for each camera in Room 1 when not being monitored by the camera. The success of the former is when the camera is detected and reported as monitoring, and the success of the latter is when the camera is detected and reported as not monitoring. The evaluation results are shown in Fig. 11. Our method achieves the TPR of 90%, 10%, 90%, 100%, 70%, 80%, 70%, 100%, and 0, while the TNR of 70%, 80%, 70%, 90%, 60%, 70%, 70%, 90%, and 100% for cameras in Table 3, respectively. The TPR of some cameras (e.g., Y3 and 83do) is very low, since these cameras are designed to send packets at a relatively stable rate and a simple turning action cannot cause a significant impact on their

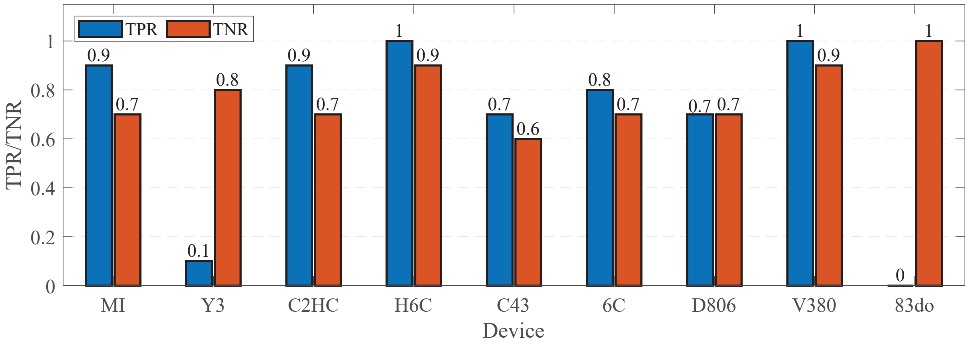
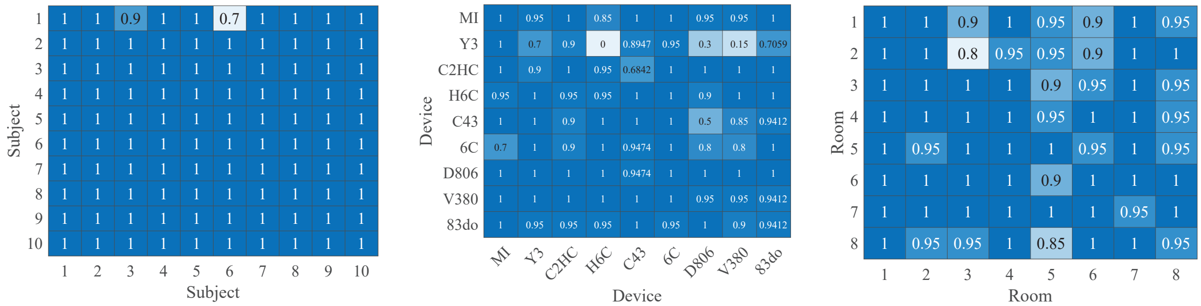


Fig. 11. TPR and TNR of detecting different cameras.



(a) (b) (c)

Fig. 12. (a) ADR values for training and testing with different subject data. (b) ADR values for training and testing with different device data. (c) ADR values for training and testing with different room data.

PTR. TNR of some cameras is relatively low, since some cameras send cover traffic [63] when the monitoring scene is stationary, to prevent attackers from inferring user presence by simply calculating PTR. However, even if LocCams mistakenly regards some cameras that are not monitoring the user, LocCams still reports their presence due to their OUIs and can localize them in the camera localization phase. Overall, the camera detection method demonstrates good performance in detecting hidden cameras in the environment.

6.4 Transferability Performance of Localizing the Hidden Camera

To evaluate the transferability of LocCams in localizing hidden cameras, we conduct experiments in various settings, including different subjects, devices, and rooms.

Different subjects. To evaluate our model's transferability to different users, we invite ten subjects to participate in the evaluation using Y3 in Room 1. Each subject performs ten camera detection and localization trails under Case 1, respectively. We take each subject's data as the training data to train a model to evaluate other subjects' data, respectively. The evaluation results are presented in Fig. 12(a). The diagonal test results are those tested with the training dataset of the model. Despite differences in height, weight, and body shape among these subjects, the ADR of the model almost remains at 100%. This result indicates that LocCams is subject-independent and effective for unseen subjects in the training dataset.

Different devices. We test nine different types of cameras listed in Table 3 in Room 1. For each type of camera, a subject performs ten trials under Case 1 and Case 2. We select the data of one device as the training data to train a

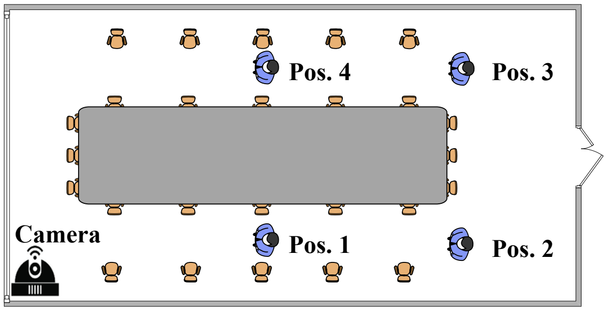


Fig. 13. Different positions in Room 1.

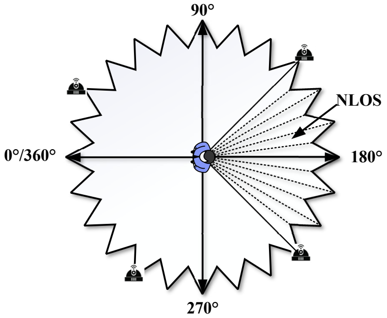


Fig. 14. Schematic diagram of data collection for different angles.

Table 5. ADRs for Different Distances

Distance	Success	Trial	ADR
1m	8	10	80%
2m	10	10	100%
3m	10	10	100%
4m	10	10	100%
5m	10	10	100%
6m	10	10	100%
7m	10	10	100%

Table 6. ADRs for Different Initial Positions

Position	Success	Trial	ADR
Pos. 1	10	10	100%
Pos. 2	10	10	100%
Pos. 3	10	10	100%
Pos. 4	10	10	100%

model, respectively, and the data of other devices as the test data. As shown in Fig. 12(b), ten models trained with data from different devices have a good performance, with an average ADR of 95.12%. The diagonal test results are those tested with the training dataset of the model. Some models do not test well with individual cameras, for example, the ADR is 50% for D806 testing the model trained with the data from C43. However, this result does not affect the user experience, since LocCams can achieve good performance as long as we choose a device with good performance for model training. In conclusion, this result demonstrates that LocCams is suitable for different WiFi cameras and robust to unseen cameras in the training dataset.

Different rooms. We randomly select a subject and a camera (e.g., Y3) to perform twenty trials in different rooms. Data collected from one room is used as the training data, while the remaining data is used as test data. The confusion matrix of the test results is shown in Fig. 12(c). Most of the ADRs are able to reach 100%. This result indicates that the model trained in one environment can be directly applied to a new environment. This system is robust to rooms that have not been seen in the training set.

The three evaluations presented above demonstrate the transferability of our CNN model to unseen users, devices, and rooms in the training dataset. Although individual sequences consisting of 100 measurements are only correctly identified at a rate of approximately 60%, the majority voting algorithm ensures that the sequences are classified correctly. This confirms the effectiveness and reliability of LocCams in localizing hidden wireless cameras.

Table 7. ADRs for Different Heights

Height	Success	Trial	ADR
1m	10	10	100%
2m	9	10	90%
3m	10	10	100%
ceiling	9	10	90%

Table 8. ADRs for Different Angles

Angle	Success	Trial	ADR
105°	0	10	0
120°	0	10	0
135°	10	10	100%
150°	10	10	100%
165°	10	10	100%

6.5 Impact of User's Initial Distances and Positions

To evaluate the performance of different distances and initial positions, we invite a subject to perform the camera localization trials in Room 1 as depicted in Fig. 13. The distances of Pos. 1 and Pos. 2 away from the camera are set to 4m and 6m, respectively. In addition, we select other distances, i.e., 1m, 2m, 3m, 5m, and 7m, along the straight line where Pos. 1 and Pos. 2 are situated. We collect ten trials of test data from each position and distance.

The test model used in the evaluation of different devices is directly used in this evaluation. ADRs for different initial distances and positions are presented in Table 5 and Table 6, respectively. For different distances, the ADRs are all 100% except for the rate 80% for the distance of 1m. This suggests that the model performs consistently well for most distances, but may have a slight drop in performance when the initial distance is too close. However, the overall performance remains satisfactory. For different positions, all ADRs are 100%. This indicates that the model is robust to different initial positions, and the features characterizing the LOS path do not change significantly, regardless of the starting point.

In conclusion, the evaluation results demonstrate the applicability and robustness of our model for different distances and initial positions. LocCams can effectively detect and localize hidden wireless cameras, even when the user starts at varying distances and positions from the camera.

6.6 Impact of the Camera Position Height

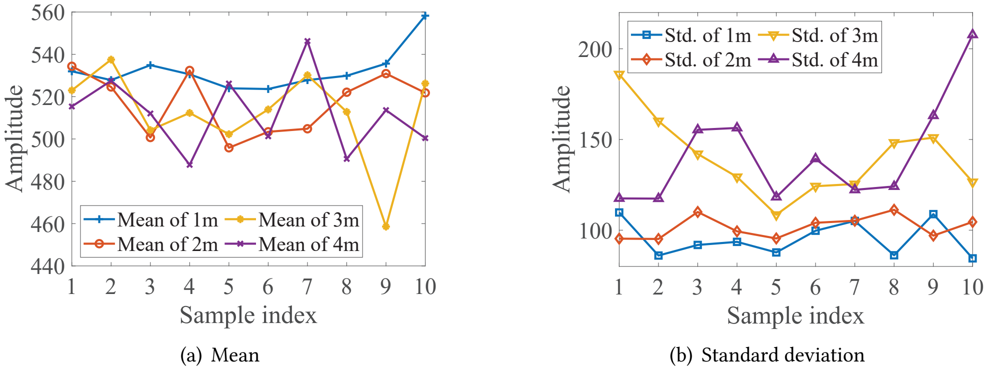
To evaluate the impact of the camera position height on LocCams's performance, we place the camera in the corner of Room 1 at three different heights: 1m, 2m, and 3m from the ground. Additionally, we install the camera on the ceiling to verify the effectiveness of LocCams for Case 3.

We collect data from 10 trials for each condition and evaluate the results. As shown in Table 7, the ADRs for different camera heights are 100%, 90%, and 100%. When the horizontal distance between the user and the camera is too short, the phone may be directly exposed to the LOS propagation path, which is classified as Case 3 in Section 5.3. As shown in Table 7, the ADR is 90% in this case. Therefore, the evaluation results demonstrate that our model is still robust to different camera heights.

6.7 Impact of the Relative Angle between the Camera Position and the User Position

Accurately identifying the LOS propagation is a crucial foundation of LocCams. To evaluate the angle resolution of our designed action, we collect data with ten trials from various angles. Specifically, we collect data for five angles from 90° to 180° in 15° steps, as we have previously evaluated four critical angles in Case 1. Each angle's data comprises ten trials with 500 CSI measurements per trial. Since the user remains stationary while collecting data, we directly divide each trial's data into five equal-interval samples.

We label all samples from each angle as under NLOS. The evaluation results, shown in Table 8, indicate that all samples from 105° and 120° are identified as the data under LOS, resulting in an ADR of 0. Therefore, when the camera is located in a 90° sector behind the user, as shown in Fig. 14, the data transmission between the hidden



(a) Mean

(b) Standard deviation

Fig. 15. Mean values (a) and standard deviations (b) of 10 CSI samples collected from distances of 1m, 2m, 3m, and 4m during a subject's turning action, respectively.

camera and the detector is considered under NLOS. That is, the NLOS resolution of LocCams is 90°. This means that LocCams is highly sensitive to detect hidden cameras positioned within a 90° sector behind the user.

6.8 Distance Estimation based on CSI

The distance between the human body and the wireless camera has a notable impact on CSI measurements [50]. This impact becomes more pronounced when the human body is in proximity to the transmitter [63]. To explore the feasibility of using CSI to estimate the distance between our detector and the wireless camera, we conduct a toy experiment.

In the experiment, a subject holds the detector and performs a turning action while 10 CSI samples are collected at different distances. For each distance, we calculate the mean standard deviations and means of the amplitudes based on 50 CSI measurements. As depicted in Fig. 15(a) and 15(b), the comparisons of means and standard deviations between short distances and longer distances reveal no significant differences. Additionally, it can be observed that the means and standard deviations for the same distances exhibit instability, indicating variability in the collected data. Therefore, based on the observed CSI variations caused by human motion, it is not feasible to accurately estimate the distance between the camera and the detector.

This may be attributed to the fact that the detector, equipped with only an antenna, is heavily influenced by the proximity of the user in our scenario. In future research, we plan to further explore distance estimation based on CSI, aiming to develop more advanced techniques that can overcome the limitations and challenges associated with estimating distances in this context.

7 DISCUSSION

In this section, we discuss the potential future direction and limitations of our work.

7.1 Narrow the Target Area

In theory, LocCams has the potential to narrow down the target area to smaller regions than just a quarter of a room by using the LOS identification method. Users can always choose to perform the turning action again within the target area if needed. Compared to SCamF [17] employing 2-dimensional spatial point localization, LocCams is more generalizable to different rooms and addresses the effect of traffic padding on localization. In addition, the 3-dimensional area localization used in this paper can be easily converted to 2D spatial localization and give the error distance, e.g., by taking the midpoint of the wall in the area as the reference point. Our evaluations have shown that our model is effective for the proposed cases, but we acknowledge that one-time localization to smaller target areas could be further pursued in future studies.

7.2 Limitations

Root authority. LocCams relies on the root authority to obtain information, such as CSI. However, granting root authority may pose a security risk to the phone. To address this limitation, one solution is to integrate the CSI retrieval function into the mobile system itself. This way, the application can directly access this function without requiring root permission. Another solution is to use a smartphone to connect to an embedded device (e.g., Raspberry Pi) that can capture CSI, via a wireless protocol such as Bluetooth, or just develop an embedded device specifically for our system serving as a dedicated tool for users. This is part of our future work.

Non-WiFi cameras. Since our monitoring device is not specialized for sensing, LocCams is not suitable for detecting cameras based on local storage, Cellular traffic, and Ethernet. Some existing works propose methods based on detecting infrared heat sources for these types of cameras [57, 64]. However, these methods require specialized infrared cameras, which can be expensive. Moreover, infrared rays have limited penetrating ability and can be easily interfered with by heat sources in the environment. Therefore, the detection and localization of non-WiFi cameras still present an open challenge.

VBR algorithm. The camera detection module in LocCams relies on the VBR [48] algorithm, which is commonly used in most surveillance devices. However, if a camera is specifically designed to encode video/audio information at a constant bit rate, LocCams may only be able to roughly detect its presence using the OUI table. However, LocCams still localize them by LOS propagation identification.

User effort. LocCams requires that the user holding a smartphone spend about 30 seconds performing the stationary-to-turn action four times during a camera detection and localization process. This interaction may cause discomfort to the user. However, the initial position and orientation can be randomly selected by the user. And the user can stand with ease when asked to be stationary. The ten subjects who participated in the experiment also indicated the ease of this interaction.

8 CONCLUSION

In this paper, we present LocCams, a method for detecting and localizing the hidden wireless camera using a commodity smartphone. LocCams employs a filtering algorithm based on PTR to automatically filter out non-camera devices and cameras that are not in the user's room or do not monitor the user. We model the camera localization problem as a LOS propagation identification problem between the hidden camera and the detector. LocCams utilizes a lightweight CNN architecture to extract features of the distribution of CSI subcarriers and applies a majority voting algorithm to identify the LOS propagation. Evaluation results demonstrate the method's robustness and effectiveness across different rooms and camera types, achieving a high probability of localizing hidden cameras within a short time. LocCams provides an accessible and practical solution for users to protect their privacy in private spaces by detecting and localizing potential hidden cameras. While there are some limitations and potential areas for improvement, the overall performance of LocCams is promising and empowers users in safeguarding their personal spaces from potential surveillance risks.

ACKNOWLEDGMENTS

This research was supported in part by the National Key R&D Program of China under grant No. 2021YFB2700200 and by the National Natural Science Foundation of China under grants NO. U1836202, 61772383, 62076187, 61802214, 62172303. The corresponding author is Jing Chen.

REFERENCES

[1] Jeffrey P. Bigham. 2019. A Camera is Watching You in Your AirBnB: And, you consented to it. https://jeffreybigham.com/blog/2019/who-is-watching-you-in-your-airbnb.html.

[2] Minyi Chai. 2021. CCTV reveals the black industrial chain of hotel sneak shots: there are hidden cameras in rooms and bathroom hooks. https://www.thepaper.cn/newsDetail_forward_12482186.

[3] Luan Chen, Iness Ahriz, and Didier Le Ruyet. 2020. AoA-Aware Probabilistic Indoor Location Fingerprinting Using Channel State Information. *IEEE Internet of Things Journal* 7, 11 (2020), 10868–10883. https://doi.org/10.1109/JIOT.2020.2990314

[4] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. 2018. DeWiCam: Detecting Hidden Wireless Cameras via Smartphones. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 1–13.

[5] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. 2020. On Detecting Hidden Wireless Cameras: A Traffic Pattern-based Approach. *IEEE Transactions on Mobile Computing* 19, 4 (2020), 907–921. https://doi.org/10.1109/TMC.2019.2900919

[6] Jeong-Sik Choi, Woong-Hee Lee, Jae-Hyun Lee, Jong-Ho Lee, and Seong-Cheol Kim. 2018. Deep Learning Based NLOS Identification With Commodity WLAN Devices. *IEEE Transactions on Vehicular Technology* 67, 4 (2018), 3295–3303. https://doi.org/10.1109/TVT.2017.2780121

[7] David Coleman. 2022. *Wi-Fi 6 & 6E for Dummies*. John Wiley & Sons, Inc.

[8] Dinhnguyen Dao, Muhammad Salman, and Youngtae Noh. 2021. DeepDeSpy: A Deep Learning-Based Wireless Spy Camera Detection System. *IEEE Access* 9 (2021), 145486–145497. https://doi.org/10.1109/ACCESS.2021.3121254

[9] Yinhuan Dong, Tughrul Arslan, and Yunjie Yang. 2022. Real-Time NLOS/LOS Identification for Smartphone-Based Indoor Positioning Systems Using WiFi RTT and RSS. *IEEE Sensors Journal* 22, 6 (2022), 5199–5209. https://doi.org/10.1109/JSEN.2021.3119234

[10] Sidney Fussell. 2019. Airbnb Has a Hidden-Camera Problem. https://www.theatlantic.com/technology/archive/2019/03/what-happens-when-you-find-cameras-your-airbnb/585007/.

[11] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. 2019. Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. 21–28.

[12] Yu Gu, Yantong Wang, Meng Wang, Zulie Pan, Zhihao Hu, Zhi Liu, Fan Shi, and Mianxiong Dong. 2021. Secure user authentication leveraging keystroke dynamics via wi-fi sensing. *IEEE Transactions on Industrial Informatics* 18, 4 (2021), 2784–2795.

[13] Donald L Hall, Matthew J Brandsema, and Ram M Narayanan. 2022. Derivation of K-Factor Detection Statistics to Discriminate Between LOS and NLOS Scenarios. *IEEE Transactions on Wireless Communications* 21, 4 (2022), 2668–2679. https://doi.org/10.1109/TWC.2021.3114614

[14] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *ACM SIGCOMM CCR* 41, 1 (Jan. 2011), 53.

[15] Daniel Chaim Halperin. 2013. Simplifying the Configuration of 802.11 Wireless Networks with Effective SNR. *CoRR* abs/1301.6644 (2013). arXiv:1301.6644 http://arxiv.org/abs/1301.6644

[16] Yan He, Qiuye He, Song Fang, and Yao Liu. 2021. MotionCompass: pinpointing wireless camera via motion-activated traffic. In *MobiSys '21: The 19th Annual International Conference on Mobile Systems, Applications, and Services*. 215–227. https://doi.org/10.1145/3458864.3467683

[17] Jeongyoon Heo, Sangwon Gil, Youngman Jung, Jinmok Kim, Donguk Kim, Woojin Park, Yongdae Kim, Kang G. Shin, and Choong-Hoon Lee. 2022. Are There Wireless Hidden Cameras Spying on Me?. In *Proceedings of the 38th Annual Computer Security Applications Conference* (Austin, TX, USA) *(ACSAC '22)*. Association for Computing Machinery, New York, NY, USA, 714–726. https://doi.org/10.1145/3564625.3564632

[18] JIM DALRYMPLE II. 2019. More than 1 in 10 Airbnb guests have found hidden cameras: Survey. https://www.inman.com/2019/06/07/more-than-1-in-10-airbnb-guest-have-found-cameras-in-rentals-survey/.

[19] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*. 448–456.

[20] ipchecktool. 2021. MAC manufacture search. https://www.ipchecktool.com/tool/macfinder.

[21] IPX1031. 2019. Survey: Do Airbnb Guests Trust Their Hosts? https://www.ipx1031.com/airbnb-guests-trust-hosts/.

[22] David Janssen. 2023. Many Airbnbs have cameras installed, especially in the US, Canada and Singapore. https://vpnoverview.com/news/camera-presence-airbnb-accommodations/.

[23] Sophie Jeong and James Griffiths. 2019. Hundreds of motel guests were secretly filmed and live-streamed online. https://edition.cnn.com/2019/03/20/asia/south-korea-hotel-spy-cam-intl/.

[24] Hyeon Jeong Jo and Seungku Kim. 2018. Indoor Smartphone Localization Based on LOS and NLOS Identification. *Sensors* 18, 11 (2018). https://doi.org/10.3390/s18113987

[25] Tian Liu, Ziyu Liu, Jun Huang, Rui Tan, and Zhen Tan. 2018. Detecting wireless spy cameras via stimulating and probing. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 243–255.

[26] Ziwei Liu, Feng Lin, Chao Wang, Yijie Shen, Zhongjie Ba, Li Lu, Wenyao Xu, and Kui Ren. 2023. CamRadar: Hidden Camera Detection Leveraging Amplitude-Modulated Sensor Images Embedded in Electromagnetic Emanations. *Proc. ACM Interact. Mob. Wearable*

Ubiquitous Technol. 6, 4, Article 173 (Jan. 2023), 25 pages. https://doi.org/10.1145/3569505

[27] LLC LSC. 2023. Hidden camera detector. https://apps.apple.com/us/app/hidden-camera-detector/id532882360.

[28] Yongsen Ma, Gang Zhou, and Shuangquan Wang. 2019. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.* 52, 3, Article 46 (June 2019), 36 pages. https://doi.org/10.1145/3310194

[29] Aanron Mak. 2019. How to Scan Your Airbnb for Hidden Cameras. https://slate.com/technology/2019/04/how-to-scan-airbnb-hidden-camera-apps.html.

[30] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2017. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2177–2184. https://doi.org/10.1109/ICDCS.2017.283

[31] Jorge Ortiz, Catherine Crawford, and Franck Le. 2019. *DeviceMien: Network Device Behavior Modeling for Identifying Unknown IoT Devices.* Association for Computing Machinery, New York, NY, USA. 106–117 pages. https://doi.org/10.1145/3302505.3310073

[32] Sameera Palipana, Piyush Agrawal, and Dirk Pesch. 2016. Channel state information based human presence detection using non-linear techniques. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. 177–186.

[33] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. 2017. Widar: Decimeter-Level Passive Tracking via Velocity Monitoring with Commodity Wi-Fi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 6:1–6:10. https://doi.org/10.1145/3084041.3084067

[34] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. 2018. Widar2. 0: Passive human tracking with a single Wi-Fi link. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 350–361.

[35] Qiluyidian. 2022. 1 man Controls 180,000 cameras: Why crime cameras are So Hard to Stop? https://baijiahao.baidu.com/s?id=1731865882372619380&wfr=spider&for=pc.

[36] Muhammad Salman, Nguyen Dao, Uichin Lee, and Youngtae Noh. 2022. CSI:DeSpy: Enabling Effortless Spy Camera Detection via Passive Sensing of User Activities and Bitrate Variations. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 72 (July 2022), 27 pages. https://doi.org/10.1145/3534593

[37] Sriram Sami, Sean Rui Xiang Tan, Bangjie Sun, and Jun Han. 2021. LAPD: Hidden Spy Camera Detection using Smartphone Time-of-Flight Sensors. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 288–301.

[38] seemoo lab. 2019. Nexmon_csi. https://github.com/seemoo-lab/nexmon_csi.

[39] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. 2012. SpinLoc: Spin Once to Know Your Location. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems and Applications*. New York, NY, USA, Article 12, 6 pages. https://doi.org/10.1145/2162081.2162099

[40] Rahul Anand Sharma, Elahe Soltanaghaei, Anthony Rowe, and Vyas Sekar. 2022. Lumos: Identifying and Localizing Diverse Hidden IoT Devices in an Unfamiliar Environment. In *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA, 1095–1112. https://www.usenix.org/conference/usenixsecurity22/presentation/sharma-rahul

[41] Biyun Sheng, Yuanrun Fang, Fu Xiao, and Lijuan Sun. 2020. An Accurate Device-Free Action Recognition System Using Two-Stream Network. *IEEE Transactions on Vehicular Technology* 69, 7 (2020), 7930–7939. https://doi.org/10.1109/TVT.2020.2993901

[42] Akash Deep Singh, Luis Garcia, Joseph Noor, and Mani B Srivastava. 2021. I Always Feel Like Somebody's Sensing Me! A Framework to Detect, Identify, and Localize Clandestine Wireless Sensors. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*. 1829–1846. https://www.usenix.org/conference/usenixsecurity21/presentation/singh

[43] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2018. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing* 18, 8 (2018), 1745–1759.

[44] SpyGuy. 2022. Airbnb Hidden Cameras: Here's How to Find Them. https://www.spyguy.com/a/blog/airbnb-hidden-cameras-four-ways-to-find-them.

[45] Jakobi Teknik. 2023. Spy hidden camera Detector. https://apps.apple.com/us/app/spy-hidden-cameradetector/id925967783?mt=8.

[46] Monica Torres. 2019. 4 Ways To Tell If There Are Hidden Cameras In Your Airbnb. https://www.huffpost.com/entry/airbnb-hidden-cameras-how-to-find_l_5cad177de4b01bf96007085c?guccounter=1.

[47] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication.* Cambridge university press.

[48] Geert Van der Auwera, Prasanth T. David, and Martin Reisslein. 2008. Traffic characteristics of H.264/AVC variable bit rate video. *IEEE Communications Magazine* 46, 11 (2008), 164–174. https://doi.org/10.1109/MCOM.2008.4689260

[49] Chen Wang, Xiuyuan Zheng, Yingying Chen, and Jie Yang. 2017. Locating Rogue Access Point Using Fine-Grained Channel Information. *IEEE Transactions on Mobile Computing* 16, 9 (2017), 2560–2573. https://doi.org/10.1109/TMC.2016.2629473

[50] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of WiFi signal based human activity recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 65–76.

[51] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. 617–628.

[52] Workshop512. 2023. Glint finder - camera detector. https://play.google.com/store/apps/details?id=com.workshop512.glintfinder.

[53] Chenshu Wu, Zheng Yang, Zimu Zhou, Xuefeng Liu, Yunhao Liu, and Jiannong Cao. 2015. Non-invasive detection of moving and stationary human with WiFi. *IEEE Journal on Selected Areas in Communications* 33, 11 (2015), 2329–2342.

[54] Kevin Wu and Brent Lagesse. 2019. Do You See What I See?Detecting Hidden Streaming Cameras Through Similarity of Simultaneous Observation. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom.* 1–10. https://doi.org/10.1109/PERCOM.2019.8767411

[55] Fu Xiao, Zhengxin Guo, Hai Zhu, Xiaohui Xie, and Ruchuan Wang. 2017. AmpN: Real-time LOS/NLOS identification with WiFi. In *IEEE International Conference on Communications, ICC 2017*. 1–7. https://doi.org/10.1109/ICC.2017.7997068

[56] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise Power Delay Profiling with Commodity WiFi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. New York, NY, USA, 53–64. https://doi.org/10.1145/2789168.2790124

[57] Zhiyuan Yu, Zhuohang Li, Yuanhaur Chang, Skylar Fong, Jian Liu, and Ning Zhang. 2022. HeatDeCam: Detecting Hidden Spy Cameras via Thermal Emissions. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. New York, NY, USA, 3107–3120. https://doi.org/10.1145/3548606.3560669

[58] Dongheng Zhang, Yang Hu, Yan Chen, and Bing Zeng. 2019. BreathTrack: Tracking indoor human breath status via commodity WiFi. *IEEE Internet of Things Journal* 6, 2 (2019), 3899–3911.

[59] Jin Zhang, Bo Wei, Fuxiang Wu, Limeng Dong, Wen Hu, Salil S Kanhere, Chengwen Luo, Shui Yu, and Jun Cheng. 2021. Gate-ID: WiFi-based human identification irrespective of walking directions in smart home. *IEEE Internet of Things Journal* 8, 9 (2021), 7610–7624. https://doi.org/10.1109/JIOT.2020.3040782

[60] Xianan Zhang, Lieke Chen, Mingjie Feng, and Tao Jiang. 2022. Toward Reliable Non-Line-of-Sight Localization Using Multipath Reflections. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 36 (March 2022), 25 pages. https://doi.org/10.1145/3517244

[61] Shuang Zhou, Lingchao Guo, Zhaoming Lu, Xiangming Wen, and Zijun Han. 2023. Wi-Monitor: Daily Activity Monitoring Using Commodity Wi-Fi. *IEEE Internet of Things Journal* 10, 2 (2023), 1588–1604. https://doi.org/10.1109/JIOT.2022.3210378

[62] Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan, Haibin Cai, Yunhao Liu, and Lionel M Ni. 2015. WiFi-Based Indoor Line-of-Sight Identification. *IEEE Transactions on Wireless Communication* 14, 11 (2015), 6125–6136. https://doi.org/10.1109/TWC.2015.2448540

[63] Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y. Zhao, and Heather Zheng. 2020. Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. https://www.ndss-symposium.org/ndss-paper/et-tu-alexa-when-commodity-wifi-devices-turn-into-adversarial-motion-sensors/

[64] Agustin Zuniga, Naser Hossein Motlagh, Mohammad A Hoque, Sasu Tarkoma, Huber Flores, and Petteri Nurmi. 2022. See No Evil: Discovering Covert Surveillance Devices Using Thermal Imaging. *IEEE Pervasive Computing* 21, 4 (2022), 33–42. https://doi.org/10.1109/MPRV.2022.3187464