Nowhere to Hide: Detecting Live Video Forgery via Vision-WiFi Silhouette Correspondence

Xinyue Fang[1,2], Jianwei Liu[1,2], Yike Chen[1,2], Jinsong Han[✉1,3], Kui Ren[1,2], and Gang Chen[4]

[1]School of Cyber Science and Technology, Zhejiang University, China
[2]ZJU-Hangzhou Global Scientific and Technological Innovation Center, China
[3]Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, China
[4]College of Computer Science and Technology, Zhejiang University, China
{xinyuefang, jianweiliu, chenyike, hanjinsong, kuiren, cg}@zju.edu.cn

***Abstract*—For safety guard and crime prevention, video surveillance systems have been pervasively deployed in many security-critical scenarios, such as the residence, retail stores, and banks. However, these systems could be infiltrated by the adversary and the video streams would be modified or replaced, i.e., under the video forgery attack. The prevalence of Internet of Things (IoT) devices and the emergence of Deepfake-like techniques severely emphasize the vulnerability of video surveillance systems under such attacks. To secure existing surveillance systems, in this paper we propose a vision-WiFi cross-modal video forgery detection system, namely *WiSil*. Leveraging a theoretical model based on the principle of signal propagation, *WiSil* constructs wave front information of the object in the monitoring area from WiFi signals. With a well-designed deep learning network, *WiSil* further recovers silhouettes from the wave front information. Based on a Siamese network-based semantic feature extractor, *WiSil* can eventually determine whether a frame is manipulated by comparing the semantic feature vectors extracted from the video's silhouette with those extracted from the WiFi's silhouette. Extensive experiments show that *WiSil* can achieve 95% accuracy in detecting tampered frames. Moreover, *WiSil* is robust against environment and person changes.**

***Index Terms*—WiFi Sensing, Video Forgery Detection, Deep Learning**

I. Introduction

Video/camera-based surveillance systems have been applied in a wide spectrum of applications, such as retails, banks, and logistics [1]. They can either prevent criminals or preserve visual evidence. With the development of IoT, these surveillance systems become more distributed, low-cost and autonomous, yet raising security risks. In particular, poor security management, e.g., using default passwords or weak remote access control, exacerbates the vulnerability towards adversaries. In this paper, we focus on the forgery attack [2], in which the attacker could hijack the camera [3] or the camera's connection Ethernet cable [4] to manipulate the live surveillance video streams. As a result, authentic frames could be replaced by fake ones. Even worse, the emergence of Deepfake techniques [5], [6] leads forge videos to be indistinguishable from real ones. Thus, the forgery attack severely threatens the authenticity and trustworthiness of existing video surveillance systems.

Xinyue Fang and Jianwei Liu are co-first authors.

The above threat reveals an urgent need to guarantee the trustworthiness of videos for surveillance systems. To this end, a large number of approaches have been proposed to detect forgery frames in suspicious videos. However, these schemes have their respective shortcomings. For example, watermark-based approaches need to adopt extra dedicated modules in cameras, which is hard to satisfy for mainstream cameras [2]. Video forensics approaches [7], [8], [9], [10], [11] usually extract temporal-spatial features from continuous frames to achieve fine-grained forgery detection. Nevertheless, such feature extraction would impose a penalty of high computational overhead and latency, which hinders their usage in live videos [2].

Given the fact that WiFi infrastructures have been deployed ubiquitously, along with the fine-grained sensing ability of WiFi signals [12], it is promising to treat WiFi channel as another trusted factor to help detect forged videos in a real-time and fine-grained manner. Pioneering approaches, e.g., [13] and [2], have demonstrated the feasibility of video looping detection and frame forgery detection using WiFi signals. However, existing approaches are either unable to realize fine-grained forged frame localization or can only detect the presence of human bodies. Considering that committing crimes via robots or trained animals has become more common [14], it is necessary to achieve real-time, fine-grained and environment-independent video forgery detection systems via ubiquitous WiFi signals, which can detect different kinds of intrusions, including humans, robots, animals, etc.

Achieving such a system, however, is non-trivial due to the following challenges. (1) To establish the correspondence between the video and WiFi, we need to find an 'intermediate' that satisfies the following two requirements. First, it can be extracted from both the video and WiFi. Second, no matter from video or WiFi, it can reflect the same kind of information about the object in the monitoring area. However, it is difficult to construct such an intermediate for two multi-modal signals, not to mention that WiFi signals are extremely complicated and unstructured [15]. (2) To further obtain robust and accurate correspondence of the intermediates between video and WiFi signals, the intermediate should record the same information about the object in the monitoring area, i.e., the perspective

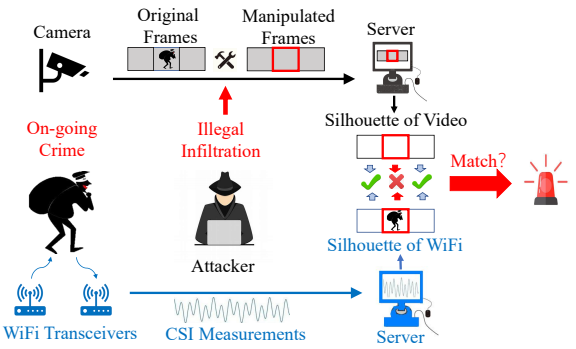


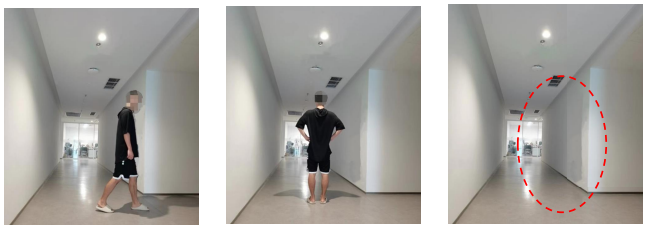
Fig. 1. Illustration of *WiSil*. By comparing the silhouette extracted from a frame with that extracted from corresponding WiFi signals, *WiSil* can determine whether the frame is manipulated or not.

of both the camera and WiFi channel should be as close to each other as possible. Nevertheless, it is impractical for them to owe the same perspective, because the WiFi antenna is omnidirectional and WiFi sensing is weakly dependent on the perspective. Thus, there are significant perspective differences between these two sensing mediums, which would impair the correspondence and degrade the forgery detection performance.

By addressing the above challenges, we propose a WiFi-assisted video forgery detection system, namely *WiSil*. As shown in Fig. 1, by establishing the correspondence between the video and trusted WiFi channel, *WiSil* is able to detect and locate falsified frames in the live video stream. On one hand, *WiSil* can be used to timely detect illegal activities erased by the attacker in the video surveillance systems. On the other hand, *WiSil* enables the WiFi measurements to be preserved as a verification factor for the video forensics systems.

Specifically, to tackle the first challenge, we propose a new intermediate, named *silhouette*, which meets the aforementioned two requirements. We noticed that the outline information, i.e., silhouette, of the object in the monitoring area can be derived from video frames via vision-based techniques, while the wave front [16] of WiFi signals can also reveal such information. We thereby dig the WiFi signal propagation and build a theoretical model to construct the wave front of objects from the channel state information (CSI) [17] of WiFi signals. Then, with a well-designed learning model, we can recover silhouette from the wave front and hence establish the correspondence between the video and WiFi. Since the wave front is only related to the dynamics of objects, i.e., being environment-independent, our silhouette construction approach is robust against environment variations. To deal with the second challenge, instead of directly comparing the differences between the two silhouettes, we design a Siamese network-based [18] feature extractor to mine semantic features that are robust against perspective differences from silhouette, i.e., enabling a perspective-insensitive matching. By calculating the matching degree of the two semantic feature vectors, we can determine if the two corresponding silhouettes are similar, i.e., whether the video frame is manipulated or not.

We build a prototype of *WiSil* with commercial off-the-shelf (COTS) devices and perform extensive experiments. The experiment results show that *WiSil* can achieve over 95% accuracy in detecting faked frames. Meanwhile, it is able to detect intrusions from various kinds of objects, including humans and robots. Moreover, *WiSil* is robust against environment and person variations. In a summary our contributions are as follows:



(a) Original frame. (b) Replaced frame. (c) Modified frame.

Fig. 2. Examples of frame forgery attack: (b) frame-replacement attack and (c) frame-modification attack.

- We propose a real-time, fine-grained, and environment-independent video forgery detection system, *WiSil*. It is capable of localizing tampered frames and allows the WiFi channel to be preserved as the additional evidence in video surveillance and forensics systems.
- We build the correspondence between video and WiFi by extracting the same outline information, i.e., silhouettes, from both of them. We propose a learning-based approach to recover the silhouette of the object from WiFi signals.
- We prototype *WiSil* with COTS devices and perform extensive experiments. The experiment results show that *WiSil* can achieve high faked forgery detection accuracy, Meanwhile, it is robust against environment and person changes.

II. Background and Threat Model

We focus on enhancing the security of the video surveillance systems that have been pervasively deployed in common scenarios, e.g., retail stores, banks, and logistics centers. In these systems, cameras are employed in a fixed way to monitor what is happening in the areas of interest. The target objects are the ones that intrude the monitoring areas, including humans and robots. Generally, the cameras have built-in Web servers and provide interfaces, e.g., Ethernet ports, so that any authorized client from network can remotely access, record, and retrieve the generated video data.

However, such surveillance systems could be compromised through illegal infiltration. An attacker can scan the relevant protocols and ports, browse the device management page, or even leverage the system vulnerabilities to intrude the system [1]. After that, the attacker can trigger video forgery attacks, i.e., frame-replacement attacks and frame-modification attacks. In the frame-replacement attack, the attacker replaces a number of original video frames with fake frames, as shown in Fig. 2. In the frame-modification attack, the attacker modifies the video by removing the content about some events that really happen, or adding certain events that actually

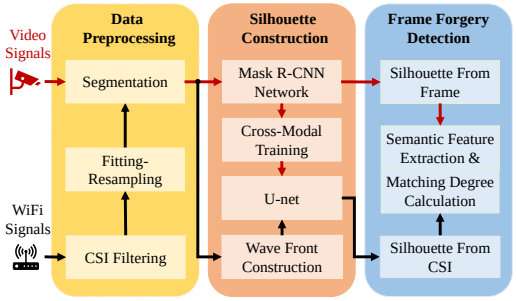


Fig. 3. Workflow of *WiSil*.

unhappen in the monitoring area. The photoshop [19] and AI-assisted technologies (DeepFake [5], [6]) can be applied in frame-modification attacks, making manipulated videos indistinguishable from normal ones.

We assume that there are WiFi facilities in the surveillance area. It is easy to meet because WiFi infrastructures have been deployed nearly everywhere. We also assume that the WiFi signals are authentic, because it is nearly impossible to manipulate complicated and unstructured WiFi measurements [2].

III. System Overview

To secure video-based surveillance and forensics systems, *WiSil* recovers the silhouette of the object in the monitoring area from reliable WiFi signals. By comparing the silhouette extracted from the frame with that extracted from the WiFi signal, *WiSil* is able to determine if a frame is manipulated. Taking as input the video signals frame by frame, *WiSil* can be used to timely localize falsified frames in live video streams. This not only can help security personnel find out ongoing illegal intrusions in real time, but also enables the WiFi channel to be preserved as an additional security guarantee for the video forensics system.

As shown in Fig. 3, *WiSil* is primarily composed of three modules: data preprocessing, silhouette construction, and frame forgery detection. The data preprocessing module takes both the video frames and WiFi signals as inputs to perform denoising and segmentation. The processed signals are then fed into the second module. In the second module, *WiSil* first transforms the frame into the silhouette via a vision-based technique, i.e., Mask R-CNN [20]. Then, the wave front is extracted from WiFi signals based on a theoretical model. With a pre-trained U-Net [21], *WiSil* reconstructs the silhouette from the wave front information. Lastly, in the third module, *WiSil* leverages a Siamese network-based structure to match the silhouette of the frame against that of WiFi. If it does not match, *WiSil* would alarm the user that the frame is faked.

■ **Data preprocessing.** This module plays the essential role of providing clean WiFi samples and synchronized video frames for the next module. The main processes include removing the noise from WiFi signals through filtering and synchronizing the video frames and WiFi samples based on their sampling rates. We will elaborate on this module in Sec. IV.

■ **Silhouette construction.** With the clean WiFi signals extracted from the reflection on the surface of the target (i.e., the object in the monitoring area), we first set an image plane with a coordinate system to compute the wave front of the reflector on the target. Then, we apply a pre-trained deep learning model to transform the wave front into the silhouette. In this process, we also extract the silhouette from the frame via a Mask R-CNN. The silhouette construction approach is detailed in Sec. V.

■ **Frame forgery detection.** After obtaining the silhouettes from both the frame and WiFi, *WiSil* tries to match between them. With a well-designed Siamese network-based architecture, *WiSil* quantifies the matching degree of the two silhouettes. If the matching degree of a frame is smaller than an acceptance threshold we set in advance, this frame is considered to be manipulated. Otherwise, this frame is authentic. The details of this process are introduced in Sec. VI.

IV. Data Preprocessing

Raw WiFi CSI contains environment and hardware noise that is irrelevant to the wave front of the moving object. Therefore, in the data preprocessing module, we utilize a series of denoising methods to obtain clean CSI measurements. In addition, to establish an accurate correspondence between the frame and CSI, the video should be synchronized with WiFi. To this end, we propose a fitting-resampling and segmentation method to assign a synchronized CSI sample to each frame. Note that in the following of this section, all the operations towards WiFi signals will be performed on each subcarrier.

A. Signal Denoising

Generally, the WiFi signals which traverse through the line-of-sight (LOS) path always incur low frequency noise, while burst noises caused by low-cost COTS devices would bring high frequency [22], [23]. Both kinds of noise exhibit extremely high/low values of CSI measurements. Therefore, we first apply a median filter to remove those noises. Then, we utilize a mean filter to suppress slight oscillations in the CSI sequences filtered by the median filter. It is worth noting that when objects move between WiFi transceivers, the Doppler effect would lead to continuous changes in the frequency observation. Meanwhile, the relative motion always has a low frequency. So, a low-pass Butterworth filter [24] with 20Hz cut-off frequency is applied to remove high-frequency noise for a much smoother CSI sequence.

B. Multi-Modal Signal Synchronization

■ **Fitting-resampling.** During the WiFi transmission, the WiFi packets could be lost due to the occlusion of the object and hardware imperfection. In this case, *WiSil* sometimes cannot obtain the same number of CSI measurements in a fixed-length period (e.g., a period used to sample a frame), resulting in an issue that we cannot unify the dimensionality of the CSI sample assigned to each frame. To solve this problem, we propose a fitting-resampling method based on our observation that the variation trace of the CSI in a short period looks like a cubic function of time t and measurement y. Specifically, we first perform cubic polynomial fitting on the

CSI of each period t_p. We then resample on the fitted cubic polynomial with equal and short time interval Δt. In this way, *WiSil* can obtain $\frac{t_p}{\Delta t}$ CSI measurements in every fixed-length period.

■ **Segmentation.** Next, we need to synchronize the video and WiFi through segmentation. The goal is to guarantee that each frame is assigned with a CSI sample and they are captured in the same time period. Since the sampling rate of WiFi packets is generally larger than that of video frames, each frame should be associated with multiple WiFi packets. Specifically, when the camera and WiFi transmitter are started at the same time t_{start}, the first frame should be assigned with the CSI sample of $\frac{SR_w}{SR_v}$ successive packets after t_{start}, where SR_w and SR_v are the sampling rates of the packet and frame. respectively. Owing to that we have performed fitting-resampling on the CSI sequences, each following frame can be assigned with a CSI sample with unified dimensionality $(\frac{SR_w}{SR_v}, N_f)$, where N_f is the number of subcarriers.

V. Silhouette Construction

WiSil achieves video forgery detection by comparing the silhouette of the frame with that of the CSI. To this end, an intuitive solution is to directly extract the silhouette from the frame using the vision-based technique (i.e., Mask R-CNN). However, it is impractical to do so with WiFi signals because they are unstructured without any human eye-perceptible information. Thus we cannot extract the outline information from WiFi signals via vision-based technologies like Mask R-CNN. To address this challenge, in this section, we propose a learning-based method to recover the silhouette from WiFi CSI. We observe that the wave front of WiFi signals can reveal the outline information of the object. We thereby build a theoretical model based on the real signal propagation to extract the wave front from the WiFi CSI. Then, we treat the silhouettes of the frames as the annotations of the CSI samples to train a U-Net. With the well-trained U-Net, we are able to recover silhouettes from WiFi signals.

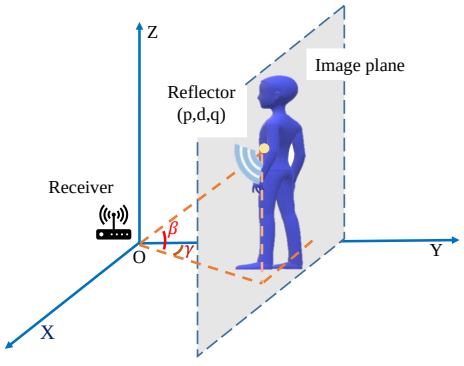


Fig. 4. Coordinate system for wave front extraction.

A. Wave Front Extraction

With the clean WiFi signals extracted from the reflection on the surface of the target, we can construct the wave front at the target side. We illustrate the scene in Fig. 4, where the target is in front of the receiver antennas.

As shown in Fig. 4, we first define a spatial coordinate system, and the receiver antennas are placed at the origin (0, 0, 0). According to [16], we can also define an image plane where the wave front is constructed on. The image plane is vertical to the ground. The target consists of many reflectors and each of them can respond to a wave front. We take one of the reflectors as the example to study the way to construct the wave front. We assume that a reflector of the target is denoted as $w_{p,q}$, and its coordinate vector is $\vec{L}_{p,q} = (x_{p,q}, y_{p,q}, z_{p,q}) = (p, d_{p,q}, q)$, with the azimuth angle $\gamma_p = arctan(p/d_{p,q})$ and the elevation angle $\beta_q = arctan(q/\sqrt{p^2 + d_{p,q}^2})$. The value of $d_{p,q}$ on the Y-axis is the depth information of the reflector, whose coordinate on the image plane is (p, q). We can get a more specific coordinate vector of the reflector as follows:

$$\vec{L}_{p,q} = (d_{p,q}tan(\gamma_p), d_{p,q}, \frac{d_{p,q}tan(\beta_q)}{cos(\gamma_p)}). \tag{1}$$

To recover the wave front, we assume that there is a virtual transmitter's antenna at the position of the reflector, indicated as $s_{p,q}$. This virtual transmitter produces signals with the same intensity and phase as the reflector. Then, we can derive the phase shift along the path between the reflector and the receiver's antennas, where the length of path is $|\vec{L}_{p,q}|$. After propagation, the signal received by the receiver antennas is known as $s_{p,q}^{R_x}$. We can describe the relationship between the wave front $s_{p,q}$ and the received signal $s_{p,q}^{R_x}$ as follows:

$$s_{p,q}^{R_x} = \alpha s_{p,q} exp(-j2\pi \frac{|\vec{L}_{p,q}|}{\lambda}), \tag{2}$$

where α is the amplitude attenuation factor and λ is the wavelength of WiFi. Besides, we define a function $K_{p,q}$ to represent the phase shift of each reflector $w_{p,q}$:

$$K_{p,q} = \exp(-j2\pi \frac{|\vec{L}_{p,q}|}{\lambda}). \tag{3}$$

We aim at recovering the wave fronts $\mathbb{S} \triangleq [s_{p,q}]_{P,Q}$ of all reflectors, i.e. all points $\forall p \in [1, P]$ and $\forall q \in [1, Q]$ on the image plane. To achieve it, the CSI measurement $S(t, f)$, with packet t and subcarrier frequency f, is mapped to the function $K_{p,q}$ as follows:

$$S(t, f) = \sum_{p=1}^{P} \sum_{q=1}^{Q} \alpha s_{p,q} K_{p,q}. \tag{4}$$

Now the CSI measurement $S(t, f)$ is known. We observe that Eq. 4 is similar with 2D Inverse Fast Fourier Transformation (2D IFFT). Therefore, we convert the goal of recovering wave fronts into a 2D IFFT problem via the relative motion between the transceiver and the target. We denote the velocity of the relative motion along Y-axis as v_y. Then, in the period Δt, the instantaneous displacement can be calculated as $\vec{d}_{p,q} = (0, v_y \Delta t, 0)$ with the direction $\frac{\vec{L}_{p,q}}{|\vec{L}_{p,q}|}$, while the initial position of the reflector is $(p, d_{p,q}, q)$ with the distance $|\vec{L}_{p,q}|_{t_0}$. Thus, we can use function $K_{p,q}$ to represent the phase

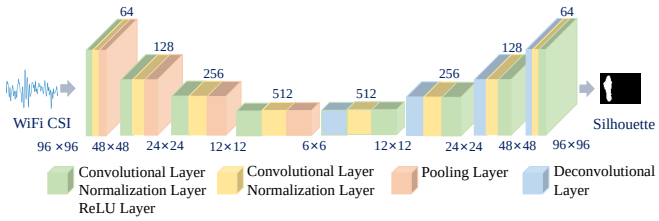


Fig. 5. Architecture of the U-Net used to recover silhouette from WiFi,

shift varying with time as follows:

$$K_{p,q} = \exp\left(-j2\pi(|\vec{L}_{p,q}|_{t_0} + \Delta\vec{d}_{p,q} \cdot \frac{\vec{L}_{p,q}}{|\vec{L}_{p,q}|})/\lambda\right) = \exp\left(-j2\pi\frac{|\vec{L}_{p,q}|_{t_0}}{\lambda}\right) \cdot \exp\left(-j2\pi v_y \frac{(cos\gamma cos\beta)t}{\lambda}\right). \tag{5}$$

Here, We only need to consider the change of variable t. After we replace λ in Eq. 5 with the wavelength λ_c of the central frequency f_c, Eq. 4 can be represented as follows:

$$S(t,f) = \sum_{p=1}^{P}\sum_{q=1}^{Q} \alpha s_{p,q} \exp\left(-j2\pi(\frac{f_c}{c}|\vec{L}_{p,q}|_{t_0} + \frac{v_y(cos\gamma cos\beta)t}{\lambda_c})\right). \tag{6}$$

Since α is a constant which can be unified, Eq. 6 is a standard 2D Fast Fourier Transformation (2D FFT) with the input signal $g(p,q)$ and the output signal $G(a,b)$. The packet t and the subcarrier frequency f is corresponding to the variable a and the variable b. Then, we replace $\frac{|\vec{L}_{p,q}|_{t_0}}{c}$ by p, and $\frac{v_y(cos\gamma cos\beta)}{\lambda_c}$ by q, Eq. 6 can be converted as follows:

$$G(a,b) = \sum_{p=1}^{P}\sum_{q=1}^{Q} g(p,q) \exp\left(-j2\pi(a \cdot \frac{p}{P} + b \cdot \frac{q}{Q})\right). \tag{7}$$

Note that $S(t,f)$ is known from the CSI measurement, i.e. $G(a,b)$ in Eq. 7. Thus we can apply 2D IFFT to compute $g(p,q)$, i.e. $s_{p,q}$. After that, we can recover the wave front $\mathbb{S} \triangleq [s_{p,q}]_{P,Q}$.

B. Recovering Silhouette From Wave Front

After obtaining the wave front of each reflector on the image plane, we apply a U-Net to transform it into the silhouette.

■ **Data annotation and model input.** We transfer the wave front of each reflector to the 3D tensors with dimensionality of $(N_f \frac{SR_w}{SR_v}, N_{tra}, N_{rec})$ as the input of our deep learning model, where N_f, N_{tra}, and N_{rec} are the number of the frequency, the transmit antennas, and the receive antennas, respectively. The WiFi transceiver and the camera work simultaneously while collecting the training set. In the training set, each wave front tensor is labeled by the silhouette extracted from its corresponding frame. The dimensionality of each silhouette is (N_{row}, N_{col}), where N_{row} and N_{col} are the number of the pixels in a row and a column, respectively.

■ **Network architecture.** The main part of our deep learning network is a semantic segmentation extractor based on U-Net. It can extract the outline feature of the target and then generate its silhouette. The network architecture is shown in Fig. 5. We begin with downsampling the input data to increase the Receptive Field (RF), so that the convolution operation can learn to extract more features. We then upsample the data for turning the low-resolution feature maps into high resolution feature maps. The reason why the input data can be mapped to the silhouette is that the 3D input data contains temporal information, wave front, and transceiver pairs among antennas. Due to the different distances and angles between the antenna pairs, multiple different descriptions of the same silhouette can be derived. In the convolutional operation, these descriptions can reconstruct 2D information of the silhouette by reorganizing and reweighing.

(a) Original frame. (b) Silhouette extracted from frame. (c) Silhouette recovered from CSI.

Fig. 6. Silhouette construction results.

■ **Loss function.** We encode the U-Net in our system to represent the silhouette. To force the network to focus on the target rather than the background, We define an efficient loss as flows :

$$\mathcal{L}_{(i,j,k)} = weight_{(i,j,k)} \cdot ||\widetilde{y}_{(i,j,k)} - y_{(i,j,k)}||^2, \tag{8}$$

where $weight_{(i,j,k)}$ is the element-wise weight at index (i,j,k). We use Matthew Weight [25] to achieve the optimizing attention mechanism on the wave front data:

$$weight_{(i,j,k)} = \begin{cases} k \cdot y_{(i,j,k)} + b & y_{(i,j,k)} \geqslant 0 \\ k \cdot y_{(i,j,k)} & y_{(i,j,k)} < 0. \end{cases} \tag{9}$$

Figure. 6(a), (b), and (c) show an original frame, the silhouette extracted from the frame, and the silhouette recovered from CSI, respectively. It can be seen that the outline information of the person in Fig. 6(a) is accurately described by Fig. 6(c).

VI. Frame Forgery Detection

So far, we have recovered the silhouette from WiFi signals. The recovered silhouette can reveal the real activities of the objects in the area under surveillance. To determine if a frame recorded by the camera is manipulated, we need to match the silhouette extracted from the frame against that recovered from the corresponding CSI sample. If the matching fails, the tested frame is extremely likely to be falsified; otherwise, we regarded the tested frame as an authentic one.

To do so, a straightforward method is to directly calculate the differences between pixels in these two silhouettes, and then count none-zero elements in the differences [2]. However, such a 'brute-forth' like method is susceptible to the noise, i.e., small outline offset of the object in the silhouettes would result in false positives. Considering that the perspectives of the camera and WiFi are impossible to be completely consistent, slight outline differences of the moving object between the two silhouettes may exist (e.g., the differences between Fig. 6(b) and (c)). In this case, a wise matching strategy is to compare

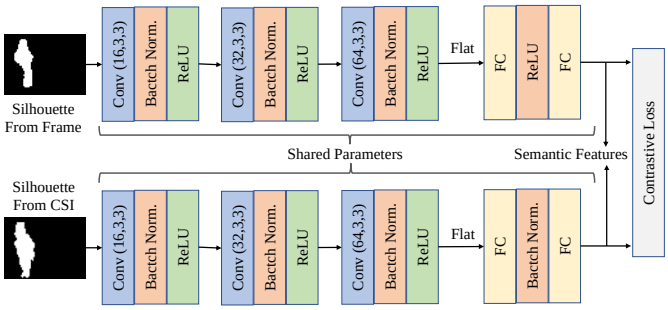


Fig. 7. Structure of our similarity-related semantic feature extractor.

the semantic similarity between the two silhouettes, rather than the pixel similarity.

It is widely known that Siamese network is significantly effective in comparing the similarity between two signature-like inputs [18]. Generally, a normal person cannot write two identical signatures, but Siamese network can accurately judge whether two signatures are from the same person by extracting deep semantic features. Analogically, we are inspired to design a Siamese network-based feature extractor to extract similarity-related semantic features from the two different kinds of silhouettes. With the semantic features extracted from the frame-associated and WiFi-associated silhouettes, we can quantify their matching degree by calculating the distance between them.

■ **Structure of semantic feature extractor.** As shown in Fig. 7, our feature extractor has two branches that share the same structure and parameters. It takes as inputs two silhouettes (one from the frame and the other from WiFi). Each silhouette is fed into an individual branch. In each branch, the feature extractor uses three two-dimensional convolutional layers to mine deep similarity-related semantic features from the silhouettes. The size of the convolutional kernel and the sliding stride are respectively set to (3, 3) and (2, 2). Each convolutional layer is followed by a batch normalization function (BN), a rectified linear unit (ReLU), and a dropout layer. The BN is utilized to prevent the offset of data distribution. The ReLU is used to decrease the dependence among neurons to improve the generalization ability of the feature extractor. As for the dropout layer, it is responsible for reducing the probability of being overfitting, which also enhances the feature extractor's generalization ability. After the last dropout layer, we add two fully-connected layers to project high-dimensional deep features into low-dimensional semantic feature vectors. A ReLU activation function is added behind the first fully-connected layer to increase the non-linearity, thus improve the feature extractor's ability of processing complex tasks. Finally, each branch would output a semantic feature vector with dimensionality of (1, N_{fea}), where N_{fea} is set to 64 empirically.

■ **Training strategy and loss functions.** To make the feature extractor possess the ability of mining similarity-related semantic features, we should train it on both positive pairs and negative pairs. The former contains the silhouette from a manipulated frame and that of its corresponding CSI sample, which are dissimilar to each other. In contrast, the latter is composed of the silhouette from a normal frame and that of its corresponding CSI sample, which are similar to each other. The similarity labels of the positive pair and the negative pair are set to 1 and 0, respectively. We use contrastive loss [15] to optimize the feature extractor, which can be formulated as:

$$\mathcal{L}_{con} = (1-Y)\frac{{D_w}^2}{2} + Y\frac{max\{0, m-D_w\}^2}{2}, \tag{10}$$

where Y is the similarity label and m is a hyper-parameter set to 2 empirically. D_w is the Euclidean distance between the two feature vectors $X_1 = [x_1^1, x_1^2, \cdots, x_1^n]$ and $X_2 = [x_2^1, x_2^2, \cdots, x_2^n]$, which can be calculated by:

$$D_w = \sqrt{(x_1^1 - x_2^1)^2 + (x_1^2 - x_2^2)^2 + \cdots + (x_1^n - x_2^n)^2}. \tag{11}$$

With the contrastive loss, we can update the parameters of the feature extractor through back propagation [26].

■ **Matching degree quantification.** After obtaining the similarity-related semantic feature vectors of the two silhouettes, we need to determine if they can match each other. In particular, we still leverage Euclidean distance to quantify the matching degree between them. The matching degree can be calculated by:

$$MD = \frac{1}{D_w(X_1, X_2)}. \tag{12}$$

Next, we set an acceptance threshold (empirically set to 0.83) to determine the final matching result. If MD is larger than or equals to the threshold, the tested frame is considered to be authentic; otherwise, we regarded it as a faked frame. In live video inspection, *WiSil* localizes the fake frame by taking as input the video stream in a frame-by-frame manner.

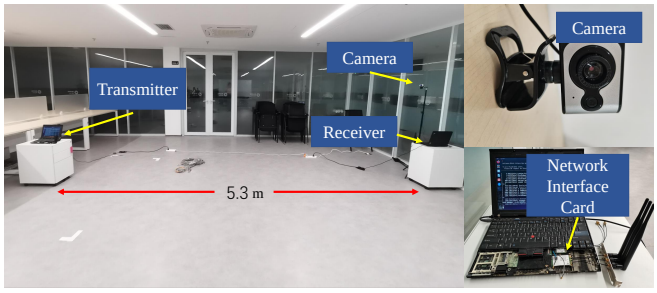


Fig. 8. Experiment setup.

VII. Evaluation

This section first describes the implementation of *WiSil*, and then details its quantitative performance.

■ **Experiment setup.** As shown in Fig. 8, we implement *WiSil* on three COTS devices: a Nuoxin camera and two Lenovo ThinkPad laptops equipped with Intel 5300 Network Interface Cards (NICs). One NIC is used as the WiFi transmitter and the other as the receiver. Each NIC is equipped with three antennas. The camera and the WiFi transceivers are placed 1.5 m and 60 cm off the ground, respectively. The transmitter is 5.3 m away from the receiver. In the default setting, we collect data in the laboratory environment shown in Fig. 8.

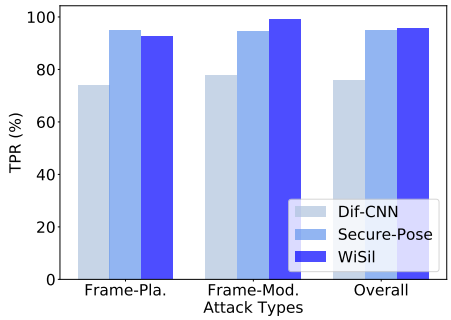


Fig. 9. TPR of *WiSil*.

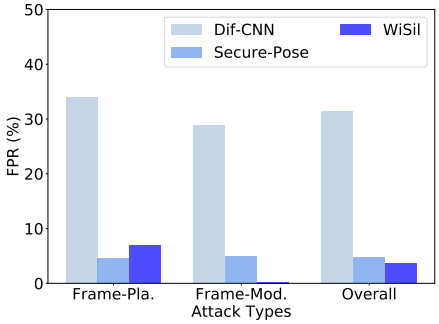


Fig. 10. FPR of *WiSil*.

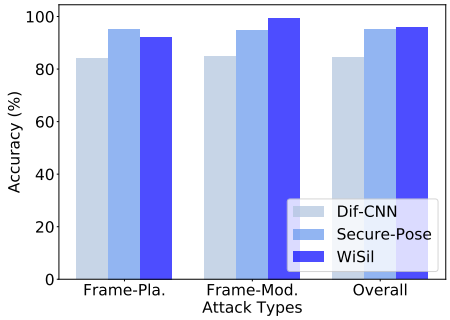


Fig. 11. Accuracy of *WiSil*.

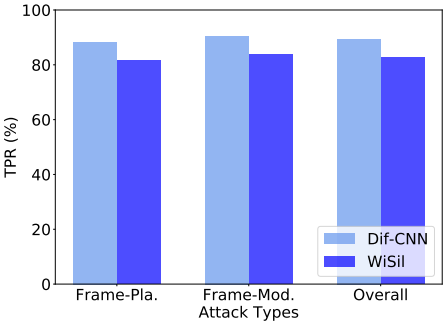


Fig. 12. TPR of cross-environment experiments.

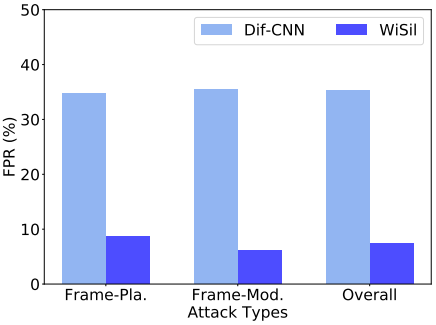


Fig. 13. FPR of cross-environment experiments.

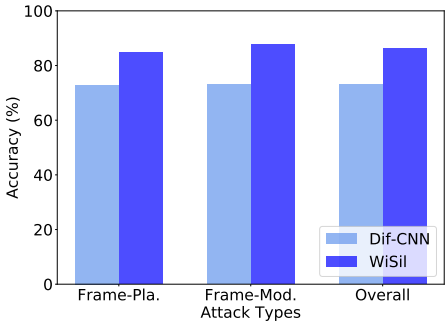


Fig. 14. Accuracy of cross-environment experiments.

The sampling rate of the video is 10 frames per second, while that of the WiFi channel is 100 packets per second. Besides, We use a Dell personal computer with 3GHz i7-9700 CPU to perform data processing. CSI is extracted from WiFi packets via PicoScenes measurement tools [27]. All the experiments are conducted by adhering to the approval of our university's Institutional Review Board (IRB).

■ **Data collection.** We invite 11 volunteers, including four females and seven males (with heights ranging from 156 cm to 184 cm, and aged from 22 to 30), to participate in our experiments. We ask the participants (1/2/3/4 persons) to move in the monitoring area for at least ten minutes. We also use a remote control car (with 20 cm length, 8.5 cm width, and 4.5 cm height) to act as an illegal robot. We totally collect 103000 frame-CSI pairs (half of them are positive). Therein, 36500 pairs (30000 for persons and 6500 for the robot) are used to evaluate the overall performance. The remaining 66500 pairs are used in the subsequent robustness assessment.

■ **Metrics.** We define three metrics to quantify the performance of *WiSil*: true positive rate (TPR), false positive rate (FPR), and accuracy. TPR describes the probability that *WiSil* successfully detects a manipulated frame. It can be calculated by:

$$TPR = 100\% \times \frac{N_{mani}^{corr}}{N_{mani}}, \tag{13}$$

where N_{mani}^{corr} and N_{mani} are the number of successfully detected manipulated frames and the number of all tested manipulated frames, respectively. FPR represents the probability that *WiSil* falsely recognizes a normal frame as a manipulated one. It can be formulated as:

$$FPR = 100\% \times \frac{N_{norm}^{inco}}{N_{norm}}, \tag{14}$$

where N_{norm}^{inco} and N_{norm} are the number of the normal frames that are wrongly detected as manipulated ones and the number of all tested normal frames, respectively. Accuracy is the probability that *WiSil* can correctly judge if a frame is manipulated. It can be calculated as:

$$accuracy = 100\% \times \frac{N^{corr}}{N^{all}}, \tag{15}$$

where N^{corr} and N^{all} are the number of correctly identified frames and the number of all tested frames, respectively. The higher the TPR and the accuracy are, the better *WiSil*'s forgery detection capability is. The lower the FPR is, the better *WiSil*'s usability is.

A. Overall Performance

We first assess the TPR, FPR, and accuracy of *WiSil* in terms of the human detection. For the dataset, we make a [80%, 20%] random split for training and testing. We also compare *WiSil* with two baselines: Dif-CNN and the state-of-the-art WiFi-assisted video forgery detection system named Secure-Pose [2]. The Dif-CNN is a cross-modal frame-CSI comparison algorithm proposed by [2], which leverages a CNN to determine if the silhouettes from the frame and CSI are similar to each other by taking the silhouette differences as inputs. The TPR, FPR, and accuracy are shown in Fig. 9, 10, and 11, respectively. It can be observed that, no matter for frame-placement attack or frame-modification attack, the TPR of *WiSil* is similar to that of Secure-Pose. The overall TPRs of *WiSil* and Secure-Pose are 95.9% and 94.9%, respectively. Thus, *WiSil* outperforms Secure-Pose. Meanwhile, it can be found that the TPRs of Dif-CNN for all attack types are lower than 80%. This demonstrates that our semantic feature extractor can effectively solve the problem of perspective

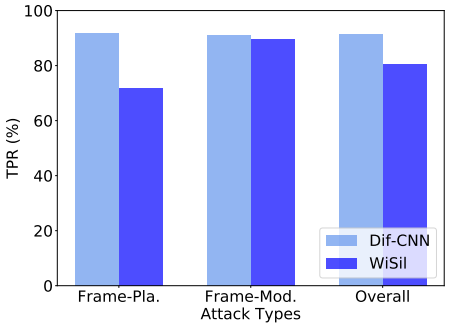


Fig. 15. TPR of cross-person experiments.

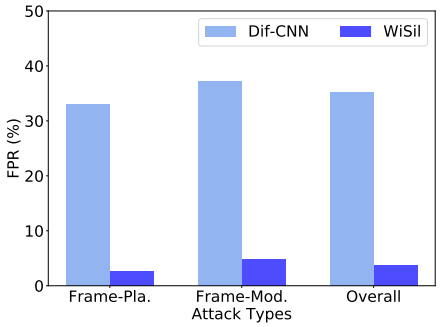


Fig. 16. FPR of cross-person experiments.

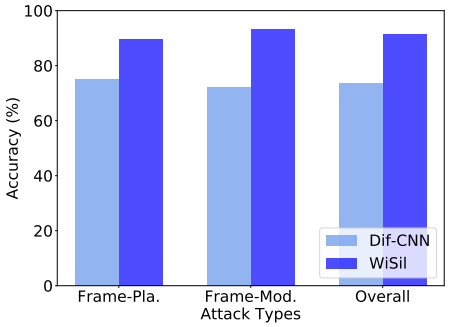


Fig. 17. Accuracy of cross-person experiments.

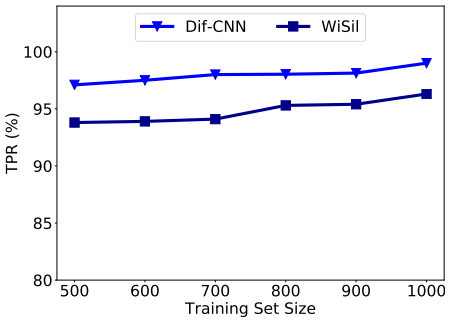


Fig. 18. Effect of the training set size on TPR.

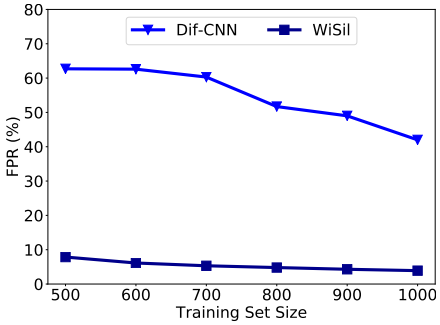


Fig. 19. Effect of the training set size on FPR.

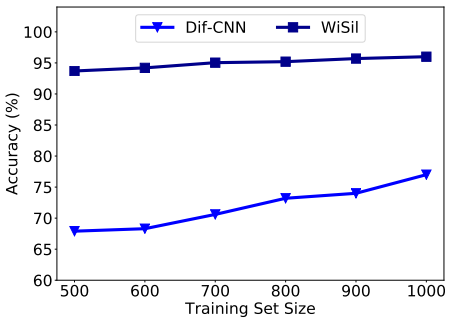


Fig. 20. Effect of the training set size on accuracy.

difference, while Dif-CNN cannot. From Fig. 10 we can find that the FPRs of Dif-CNN are always high, even higher than 25%. However, for all attack types, the FPRs of *WiSil* and Secure-Pose are less than 7%. The overall FPRs of them are 3.6% and 4.7%, respectively. Hence, compared with Dif-CNN and Secure-Pose, *WiSil* is more usable because it would produce less false alarms. Likewise, the results in Fig. 11 show that the overall accuracy of *WiSil* (95.8%) is better than that of Dif-CNN and Secure-Pose.

Next, we evaluate the performance of *WiSil* in robot detection. After adopting the same experiment method used in human detection, we find that the accuracy of frame-placement attack and frame-modification attack is 81.0% and 84.6%, respectively. Apparently, this accuracy is not as high as that of human detection. This is reasonable because the size of the robot is far smaller than that of humans. However, such accuracy, over 80%, is still high. More importantly, in terms of robot detection, *WiSil* outperforms the state of the art (i.e., Secure-Pose), because Secure-Pose cannot detect the objects other than humans.

B. Cross-domain Test

In practice, *WiSil* may be trained in Domain A (the domain includes environments and persons) but used in Domain B. In this case, the data of Domain B is not included in the training set. Thus, it is necessary to evaluate *WiSil*'s cross-domain performance.

■ **Cross-environment test.** This experiment is conducted by training *WiSil* on the data collected in the laboratory and testing on that in the office. The TPR, FPR, and accuracy are shown in Fig. 12, 13, and 14, respectively. It can be found that the overall TPR of *WiSil* is as high as 82.7%. The TPR of Dif-CNN is also high, i.e., 89.4%. It seems that Dif-CNN outperforms *WiSil*. However, from Fig. 13 we can find that the overall FPR of Dif-CNN is 35.2%, which is unacceptable to a forgery detection system. The overall FPR of *WiSil* is only 7.4%. Besides, the results in Fig. 14 show that the overall accuracy of *WiSil* (86.3%) is higher than that of Dif-CNN (73.0%). Thus, *WiSil* outperforms Dif-CNN. Meanwhile, it is reasonable that *WiSil* has good cross-environment performance, because the wave front information extracted by *WiSil* is environment-irrelevant. The wave front is only related to the dynamics of the object in the monitoring area. Besides, one possible reason why the cross-environment accuracy is not as high as 95.8% is that we cannot remove the noise from WiFi signals perfectly. However, since the original WiFi signals used by the state of the art (Secure-Pose) are environment-relevant, the robustness against environment variation of *WiSil* is stronger than that of Secure-Pose.

■ **Cross-person test.** In this experiment, we train *WiSil* on the data of six persons and test on that of other persons. The TPR, FPR, and accuracy are shown in Fig. 15, 16. and 17, respectively. From these results, we draw the same conclusions as the cross-environment experiments. It can be observed that, although the overall TPR of Dif-CNN (91.4%) is better than that of *WiSil* (80.6%), the overall FPR of Dif-CNN (35.2%) is far worse than that of *WiSil* (3.8%). Meanwhile, the overall accuracy of *WiSil* (91.5%) is far higher than that of Dif-CNN (73.7%). Apparently, *WiSil* performs better than Dif-CNN in cross-person tests. The reason why *WiSil* achieves high cross-person accuracy is also that *WiSil* recovers silhouette from the wave front which is only related to the object's dynamics.

C. Effect of Training Set Size

The requirement for the training set size is related to the user-friendliness of *WiSil*. Hence, in this experiment, we explore the effect of the training set size on *WiSil*'s performance. Specifically, we vary the training set sizes (i.e., the number of frame-CSI pairs) of both the U-Net and Siamese network-based feature extractor from 500 to 1000 in step of 100. The overall TPR, FPR, and accuracy are shown in Fig. 18, 19, and 20, respectively. It can be found that all the TPR and accuracy increase as the training set size, while the FPR decreases with the increase of the training set size. When the training set size is 500, *WiSil* already can achieve 93%+ accuracy. However, the accuracy of Dif-CNN is only 67.9%. When the training set size reaches 1000, the accuracy can achieve the maximum, 96.0%. Collecting 1000 pairs of frame-CSI only consumes 100 seconds. Thus, users only need to collect the data in a few minutes for training *WiSil*, which means that *WiSil* is significantly user-friendly.

D. Latency

Since the latency is directly related to the real-time performance, we assess the time cost used to identify one frame in this part. The time cost for CSI processing mainly comes from two components: (1) the silhouette extraction from CSI via U-Net and (2) the semantic features extraction from silhouettes of both frame and CSI via the Siamese network-based feature extractor. With an NVIDIA GeForce GTX 1060 Graphic Processing Unit (GPU), the average time costs of (1) and (2) are about 0.007 and 0.02 seconds, respectively. Therefore, with a good GPU, *WiSil* can achieve forgery detection for one frame within 0.1 seconds. If a server with a better GPU is adopted, such time cost could be lower. It means that *WiSil* has outstanding real-time performance and can be used in live video scenarios.

VIII. Related Work

This work is mainly related to two kinds of techniques: video forgery detection and WiFi-based sensing.

■ **Video forgery detection.** With the rapid development of IoT, video surveillance systems have been widely deployed in many security-critical areas like banks to detect and record illegitimate intrusions. However, poorly managed video surveillance systems are reported vulnerable to frame manipulation. An attacker could infiltrate into the surveillance system and edit the frames to hide authentic activities (possibly illegal). According to [2], traditional approaches to detecting video forgery can be divided into two categories: watermarker-based methods [2] and video forensics methods [7], [8], [9], [10], [11]. However, the former requires the cameras to have advanced modules, which cannot be achieved by many camera manufactures. The latter can judge whether a video is manipulated or not, but it cannot work in real time due to its computation-rich feature extraction process. To realize real-time counterfeit frame localization, Huang *et al.* [2] propose to recover human poses from secure WiFi signals to perform matching against that of frames. Nevertheless, their system is able to detect human motions only, while other illegal activities, e.g., completed by robots or trained animals, would be overlooked. In this work, we propose *WiSil*, a comprehensive video forgery detection system based on WiFi channel. It is capable of recovering all kinds of intentionally hidden clues demonstrated by humans, robots, animals, etc.

■ **WiFi-based sensing.** Thanks to the ubiquity of WiFi infrastructures, WiFi signals are exploited to enable a wide array of applications [28], [29], [30], [31], [32], [33], [34] in the recent decades. The pioneer [35] in this field uses an expensive universal software radio platform (USRP) to collect WiFi signals. Its sensing granularity is relatively coarse, i.e., it can only detect the motion of the sensing target. Later on, as WiFi sensing tools [17], i.e., CSI extraction tools, on commercial NICs are developed, it has become a main trend of WiFi-based sensing to utilize cheap standard WiFi devices equipped with NICs as the signal transceivers. For example, WiGest [36] is a gesture recognition system implemented on standard WiFi devices. WiFall [37] employs COTS NICs to capture standard WiFi signals to distinguish the signals of 'fall' from the other three kinds of activities. In addition to the overhead reduction on devices, machine learning techniques are also introduced to improve the sensing granularity [38], [39], [40], [41]. For instance, SignFi [38] leverages convolutional neural network and takes as input the CSI to recognize small-scale hand sign language. WiPose [42] utilizes convolutional and recurrent neural networks to reconstruct 3D human poses from WiFi sequences. Different from previous work, *WiSil* is a fine-grained video-WiFi cross-modal forgery detection system that can precisely outline the sensing target in surveillance.

IX. Conclusion

To secure video-based surveillance and forensics systems, we propose a WiFi-assisted video forgery detection system named *WiSil*. We noticed that the outline information of the object in the monitoring area can be revealed by the wave front of the WiFi signals. Therefore, *WiSil* first extracts the wave front information from the CSI based on a theoretical model of signal propagation. Then, *WiSil* leverages a pre-trained deep network to recover the silhouette from the wave front. With a Siamese network-based semantic feature extractor, *WiSil* can calculate the matching degree between the silhouettes from the frame and that of the CSI, achieving manipulated frame identification. Extensive experiments show that *WiSil* can achieve 95%+ tampered frame detection accuracy. Meanwhile, *WiSil* is also able to detect objects other than humans. Besides, *WiSil* is robust against environment and person variations.

Acknowledgement

This paper is partially supported by the National Key R&D Program of China (2021QY0703), National Natural Science Foundation of China under grant U21A20462 and 62032021, Research Institute of Cyberspace Governance in Zhejiang University, and Leading Innovative and Entrepreneur Team Introduction Program of Zhejiang (Grant No. 2018R01005).

REFERENCES

[1] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: A survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1222–1233, 2013.

[2] Y. Huang, X. Li, W. Wang, T. Jiang, and Q. Zhang, "Towards cross-modal forgery detection and localization on live surveillance videos," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2021.

[3] B. H. 2013, "Exploiting network surveillance cameras like a hollywood hacker," https://www.youtube.com/watch?v=B8DjTcANBx0, 2013.

[4] D. C. 23, "Looping surveillance cameras through live editing," https://www.youtube.com/watch?v=RoOqznZUClI, 2015.

[5] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] T. Ma, B. Peng, W. Wang, and J. Dong, "MUST-GAN: multi-level statistics transfer for self-driven person image generation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[7] M. A. Fayyaz, A. Anjum, S. Ziauddin, A. Khan, and A. Sarfaraz, "An improved surveillance video forgery detection technique using sensor pattern noise and correlation of noise residues," *Multimedia Tools and Applications*, vol. 79, no. 9-10, pp. 5767–5788, 2020.

[8] J. Yang, T. Huang, and L. Su, "Using similarity analysis to detect frame duplication forgery in videos," *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1793–1811, 2016.

[9] S. Tan, S. Chen, and B. Li, "GOP based automatic detection of object-based forgery in advanced video," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2015.

[10] G. Ulutas, B. Ustubioglu, M. Ulutas, and V. V. Nabiyev, "Frame duplication/mirroring detection method with binary features," *IET Image Processing*, vol. 11, no. 5, pp. 333–342, 2017.

[11] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *Proceedings of the ACM workshop on Multimedia & Security (MM&Sec)*, 2007.

[12] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Computing Surveys*, vol. 52, no. 3, pp. 46:1–46:36, 2019.

[13] N. Lakshmanan, I. Bang, M. S. Kang, J. Han, and J. T. Lee, "Surfi: detecting surveillance camera looping attacks with wi-fi channel state information," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019.

[14] K. Wolfe, "25 animals involved in crimes," https://www.mentalfloss.com/article/649702/animals-involved-crimes, 2022.

[15] J. Liu, C. Xiao, K. Cui, J. Han, X. Xu, and K. Ren, "Behavior privacy preserving in rf sensing," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2022.

[16] C. Li, Z. Liu, Y. Yao, Z. Cao, M. Zhang, and Y. Liu, "Wi-fi see it all: generative adversarial network-augmented versatile wi-fi imaging," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2020.

[17] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11n traces with channel state information," *Computer Communication Review*, vol. 41, no. 1, p. 53, 2011.

[18] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using A "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 669–688, 1993.

[19] "Adobe photoshop," https://www.adobe.com/products/photoshop.html, 2022.

[20] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention - MICCAI - International Conference*, 2015.

[22] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity wi-fi," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2017.

[23] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2.0: Passive human tracking with a single wi-fi link," in *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2018.

[24] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[25] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang, "Person-in-wifi: Fine-grained person perception using wifi," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[26] Saurabh, "Backpropagation – algorithm for training a neural network," https://www.edureka.co/blog/backpropagation/, 2022.

[27] Z. Jiang, T. H. Luan, X. Ren, D. Lv, H. Hao, J. Wang, K. Zhao, W. Xi, Y. Xu, and R. Li, "Eliminating the barriers: Demystifying wi-fi baseband design and introducing the picoscenes wi-fi sensing platform," *IEEE Internet of Things Journal (IOTJ)*, vol. 9, no. 6, pp. 4476–4496, 2022.

[28] Y. Ren, Z. Wang, S. Tan, Y. Chen, and J. Yang, "Winect: 3d human pose tracking for free-form activity using commodity wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 5, no. 4, pp. 176:1–176:29, 2021.

[29] R. Gao, W. Li, Y. Xie, E. Yi, L. Wang, D. Wu, and D. Zhang, "Towards robust gesture recognition by characterizing the sensing quality of wifi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 6, no. 1, pp. 11:1–11:26, 2022.

[30] J. Wang, J. Luo, S. J. Pan, and A. Sun, "Learning-based outdoor localization exploiting crowd-labeled wifi hotspots," *IEEE Transactions on Mobile Computing (TMC)*, vol. 18, no. 4, pp. 896–909, 2019.

[31] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2015.

[32] Y. Zhao, R. Gao, S. Liu, L. Xie, J. Wu, H. Tu, and B. Chen, "Device-free secure interaction with hand gestures in wifi-enabled iot environment," *IEEE Internet Things Journal (IOTJ)*, vol. 8, no. 7, pp. 5619–5631, 2021.

[33] S. Tan, J. Yang, and Y. Chen, "Enabling fine-grained finger gesture recognition on commodity wifi devices," *IEEE Transactions on Mobile Computing (TMC)*, vol. 21, no. 8, pp. 2789–2802, 2022.

[34] Z. Lin, Y. Xie, X. Guo, Y. Ren, Y. Chen, and C. Wang, "Wieat: Fine-grained device-free eating monitoring leveraging wi-fi signals," in *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2020.

[35] F. Adib and D. Katabi, "See through walls with wifi!" in *Proceedings of the ACM SIGCOMM Conference*, 2013.

[36] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[37] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 16, no. 2, pp. 581–594, 2017.

[38] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 2, no. 1, pp. 23:1–23:21, 2018.

[39] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2019.

[40] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[41] T. Zheng, Z. Chen, S. Ding, and J. Luo, "Enhancing RF sensing with deep learning: A layered approach," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 70–76, 2021.

[42] W. Jiang, H. Xue, C. Miao, S. Wang, S. Lin, C. Tian, S. Murali, H. Hu, Z. Sun, and L. Su, "Towards 3d human pose construction using wifi," in *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*, 2020.