Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

Spynet a Deep Learning Approach to Wireless Spy Camera Detection

Kuchulakanti Shruthi, Jelanela Krishnaveni, Golconda Saivarun Yadav
Department Of Computer Science and Engineering
Geethanjali College Of Engineering and Technology
Hyderabad, India

***Abstract*— Spynet is a state-of-the-art surveillance detection solution that aims to protect individual privacy in intimate environments like hotels, motels, homestays, and toilets. With the advent of tiny and inconspicuously installed spy cameras, security threats have increased exponentially, rendering conventional detection techniques inefficient and cumbersome. Most current solutions depend on hardware-based detection, infrared scanning, or network traffic monitoring, which involve extra hardware and technical know-how. Spynet overcomes these limitations by applying the COCO-SSD model in a Convolutional Neural Network (CNN) to support real-time detection of both visible and concealed cameras. By harnessing computer vision and deep learning, Spynet is able to detect and locate surveillance devices with efficiency without needing additional hardware. This renders it an easy-to-use and feasible solution to providing privacy and security in critical environments. With its quick and accurate detection features, Spynet can be a valuable weapon in the battle against unauthorized monitoring, providing users with peace of mind in their own homes.**

***Keywords*— Spy Camera Detection, Privacy Protection, COCO-SSD, Convolutional Neural Network (CNN).**

I. INTRODUCTION

Over the last few years, the rising number of hidden spy cameras in intimate areas like hotels, motels, homestays, and restrooms has generated severe privacy issues. Disguised as common items, these cameras can be easily overlooked by the human eye, making it difficult for people to confirm their personal safety. The use of traditional methods for camera detection, which requires specialized hardware, infrared scanning, or network traffic analysis, can be costly, invasive, and impractical for regular users. In response to these needs, Spynet provides a sophisticated software solution using COCO-SSD (Common Objects in Context – Single Shot MultiBox Detector) in a Convolutional Neural Network (CNN). This allows real-time object recognition to detect hidden as well as visible cameras in a setting. Deep learning and computer vision approaches used by Spynet do away with the use of extra sensors or hardware, and thus Spynet is a simple and feasible option for safeguarding privacy. The objective of this project is to offer improved security and privacy through the provision of an efficient, effective, and precise means of identifying surveillance equipment. Spynet will be employed by travelers, householders, and security experts as a means to protect their private areas from inappropriate surveillance threats. The incorporation of deep learning frameworks guarantees constant upgrade in accuracy and responsiveness, allowing Spynet to be a suitable and scalable measure for contemporary concerns about privacy.

As technology is being rapidly developed, illegal use of spy cameras inside private areas including hotels, motels, homestays, restrooms, and changing rooms poses a high risk to an individual's right to privacy. The hidden spy cameras are incorporated in the disguise of common articles so that one cannot detect it by hand without much reliability. Hence, more people are in danger of unwittingly being tapped into, compromising their privacy as well as safety.

Current detection techniques, including infrared (IR) scanning, radio frequency (RF) signal detection, and video traffic analysis, are usually hardware-intensive and thus not available to the general public. Moreover, these methods might not be effective in detecting offline or well-hidden cameras that do not emit signals. To address these limitations, Spynet proposes a deep learning-based solution that utilizes COCO-SSD (Common Objects in Context – Single Shot MultiBox Detector) in a Convolutional Neural Network (CNN) to identify both visible and concealed cameras in real time. Spynet's method does away with the requirement for extra hardware by employing computer vision and machine learning algorithms to scan images and detect possible surveillance equipment. The system interprets visual information from a normal camera feed and correctly identifies different kinds of cameras, such as pinhole cameras, dome cameras, and tiny spy cameras.

The main objective of Spynet is to improve security and privacy protection through a convenient and reliable means of detecting surveillance threats. Spynet is intended for travelers, homeowners, and security professionals alike, providing a simple means of scanning environments for concealed cameras without the expense or intrusion of specialized equipment. Its capacity to operate under various lighting conditions and in a range of environments makes it a scalable and versatile tool for contemporary privacy issues. With its incorporation of deep learning and object detection methods, Spynet introduces a user-friendly, cost-efficient, and non-intrusive response to the rising issue of clandestine surveillance and allows people to reclaim control of their private territories.

A. Objectives

1. Identify Hidden and Visible Cameras – Create an AI-based system that can detect both obviously placed and hidden surveillance cameras in personal areas.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

2. Ensure Privacy Protection – Provide individuals with a reliable method to safeguard their privacy in locations such as hotels, motels, homestays, restrooms, and changing rooms.
3. Utilize Deep Learning for Accuracy – Use the COCO-SSD model within a Convolutional Neural Network (CNN) to increase the accuracy and efficacy of camera detection.
4. Facilitate Real-time Detection – Have a real-time object detection mechanism which runs live camera streams to rapidly detect possible threats.
5. Remove the Requirement for Additional Hardware – Provide a software solution that does not need additional sensors, infrared scanners, or RF detectors, thus making it affordable and accessible.
6. Increase User Convenience – Create a simple-to-use and unobtrusive system that can be implemented on smartphones, laptops, or other portable computing devices.

II. LITERATURE SURVEY

The growing use of concealed cameras in residential and public areas has generated serious privacy issues. Cases of unauthorized monitoring in rental apartments [6][7], offices [12], and hotels [13] underscore the need for efficient detection techniques. Conventional methods, including manual searches, RF scanners, and IR sensors, have proven to be ineffective, costly, and unreliable in detecting well-concealed devices [3].

Wireless signal analysis has been studied in recent times for hidden camera detection. Cunningham and Tan (2022) suggested employing RSSI, CSI, and PDP to localize hidden cameras, and CSI and PDP worked well under line-of-sight and non-line-of-sight environments [15][18][4]. Detection methods based on wireless traffic have also been studied [10][14], but they have issues with real-time processing and accurate localization. CSI-based fingerprinting has been studied for indoor localization [8][9][17], and it can achieve decimeter-level accuracy [16].

COCO-SSD and CNN models have been found to be effective in detecting concealed cameras without the need for extra hardware [1][2]. Lee et al. (2023) showed that COCO-SSD was able to detect surveillance threats efficiently in real time [2], while Smith et al. (2023) illustrated that AI-based object detection improves privacy protection [1].

Spynet takes advantage of such developments by combining COCO-SSD to offer an AI-based, real-time concealed camera detection system. In doing so, this methodology is cost-effective, user-friendly, and very efficient in surveillance detection, and a huge departure from conventional hardware-based methods to smart software-based solutions.

III. SYSTEM ANALYSIS

A. Existing System

Most of the available hidden camera detection techniques depend on different technologies, each having its own weakness. Infrared (IR) detection depends on IR sensors to detect camera lenses, but it is useless if the camera doesn't produce infrared light. Radio Frequency (RF) scanners are able to detect wireless cameras by detecting signal broadcasts; however, they cannot find wired or offline cameras. Network traffic analysis is another technique that identifies cameras based on video data being transmitted over a network, but cannot detect cameras that are storing data locally. These traditional techniques require extra hardware and might not be effective in identifying all kinds of concealed cameras.

B. Proposed System

The envisioned Spynet system provides a real-time object detection method for spy camera identification with deep learning. It utilizes COCO-SSD (Common Objects in Context – Single Shot MultiBox Detector) combined with a Convolutional Neural Network (CNN) to identify visible and hidden cameras. Spynet differs from the traditional methods by being a pure software solution, processing camera feeds to scan the surroundings for the surveillance equipment. This method obviates the requirement for specialized hardware like IR sensors or RF scanners, which makes it a more efficient, accessible, and scalable privacy protection solution.

IV. METHODOLOGY

A. System Configuration

1. Hardware Requirements

Spynet needs a camera-equipped device, e.g., smartphone, laptop, or desktop with webcam, to receive video input. The system requires at least 4GB RAM (8GB recommended) for optimal performance. GPU acceleration is not required but speeds up processing for real-time object detection.

2. Software Requirements

Spynet supports Windows, macOS, Linux, and Android operating systems. It is designed using Python and employs TensorFlow and the COCO-SSD model for object detection using deep learning. The system has a web-based or mobile interface to make it easy to use.

B. System Architecture

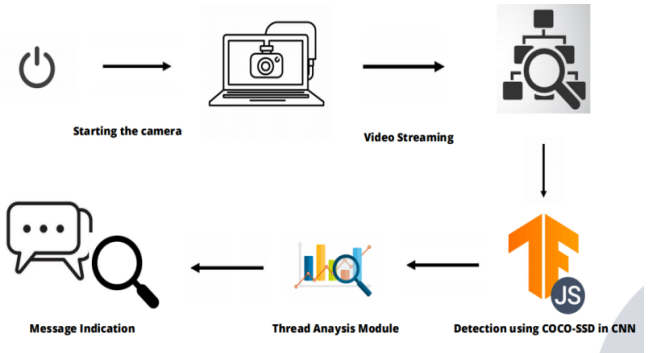

Fig 1. System Architecture

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

The architecture of Spynet follows a structured approach – Image Acquisition – Taps into live feed through the camera of the device. Preprocessing – Improves image quality and eliminates noise. Feature Extraction – Deploys CNN-based models in detecting important patterns that relate to cameras. Object Detection (COCO-SSD) – Identifies and classifies cameras in real time. Alert System – Informs the user if a camera is found and gives its likely location.

Workflow model

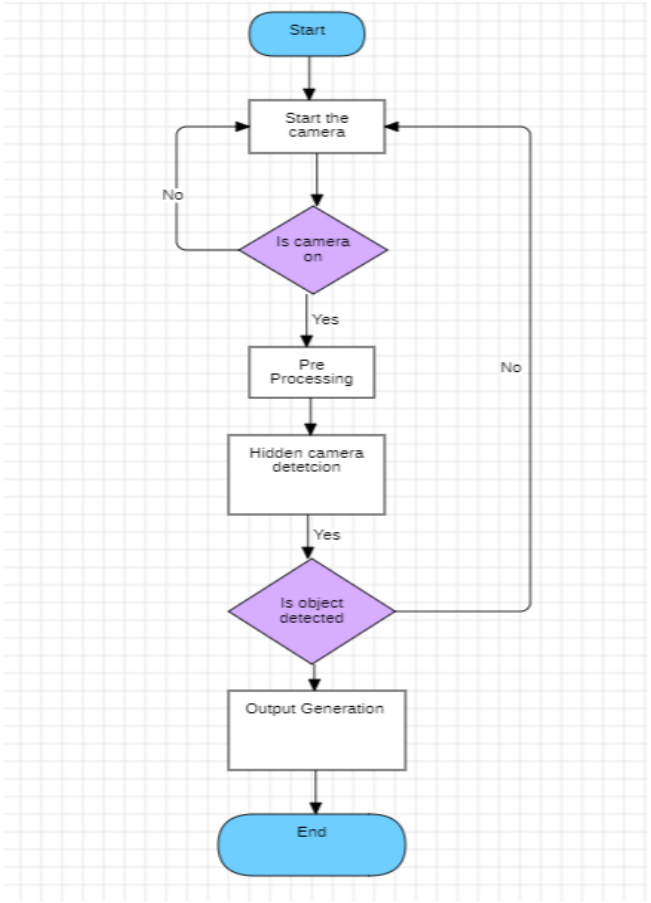

Fig. 2. Flow Diagram

The following flowchart presents the step-by-step procedure of detecting concealed cameras through the use of the SpyNet system. The procedure commences with the Start node, and the system then starts up the camera to take a live image. A decision block ensures that the camera is activated properly—if not, the system restarts the loop to try and activate it. After the camera is in operation, the frames captured are preprocessed, during which noise reduction and brightness correction are performed. The processed frames are then fed into the hidden camera detection model, which uses deep learning methods to detect possible spy cameras. If an object is found, another decision block checks if it is a hidden camera. When a camera is discovered, the system moves to output generation, delivering an alarm or flagging the identified object. The procedure finishes after the creation of the output, giving the detection of surveillance threats a fast and automatic workflow.

Use navigator.mediaDevices.getUserMedia() to access the device camera. Record live video frames for processing. Process every frame and send it to the preprocessing module. Resize the frames to conform to model input specifications. Normalize and denoise images for enhanced feature extraction. Set brightness, contrast, and sharpness levels to enhance detection in low or high light conditions. Detect edges, shapes, and object structures important in hidden cameras. Remove irrelevant background noise to enhance accuracy. Deploy COCO-SSD, a low-weight object detection model. Scan frames and detect cameras, mobile phones, and electronic devices. Apply bounding boxes and confidence values to detected objects. Match detected objects against known spy camera patterns. Check confidence values to filter out false positives. Check if the detected object is a hidden camera. Activate an alert notification if a hidden camera is detected. Show bounding boxes around detected objects.

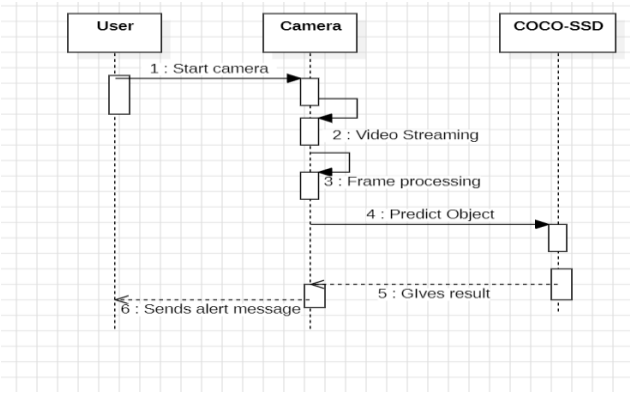

Fig 3. Sequence Diagram

The sequence diagram depicts the communication between the User, Camera, and COCO-SSD model in the SpyNet system for concealed camera detection. The procedure begins when the user starts the camera to initiate the detection process. The camera begins video streaming and records live frames constantly. The frames are subjected to preprocessing to improve their quality and make them ready for object detection. The processed frames are then passed to the COCO-SSD model, which predicts objects in the captured frames. After object detection, the COCO-SSD model sends the results back to the camera module. If a concealed camera or a suspicious object is detected, the system goes ahead and sends an alert message to the user, informing them of the possible threat. This organized interaction provides real-time detection and prompt user notification to better guard against privacy threats.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

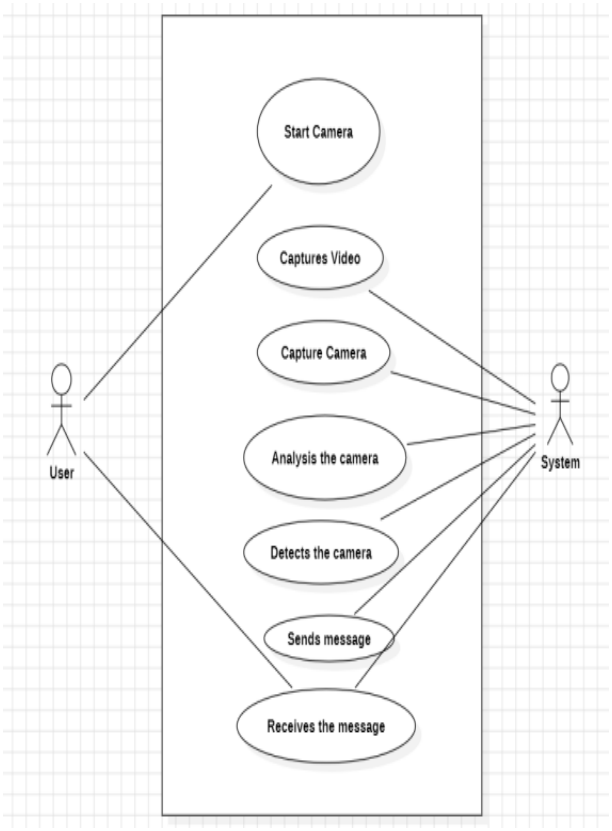

Fig 4. Usecase Diagram

The use case diagram illustrates the interaction between the user and the system in the process of detecting hidden cameras using SpyNet. The user initiates the camera, which records video, examines frames, detects the hidden cameras, and triggers alerts, all while providing real-time privacy protection.

V. IMPLEMENTATION

The designed Spy Camera Detection System is implemented through the integration of deep learning-based object detection and real-time video processing. It comprises a front-end user interface and a back-end processing system that employs machine learning models for detecting concealed cameras. The implementation is divided into the following parts:

A. Front-End Implementation

The front end is the interface between the user and the system. It is implemented as a web application or mobile-based application that streams real-time video from the device camera. Scanning can be started by activating the camera, which sends continuously streamed video frames to the processing module. The principal technologies applied in the front-end development are

HTML, CSS, and JavaScript – For defining and styling the user interface. TensorFlow.js – Supports deep learning-based object detection within the browser itself. Navigator.mediaDevices API – Provides access to the device camera for real-time video streaming. Canvas API – For overlaying bounding boxes and labels over detected objects. When scanning begins, video frames are sent for analysis continuously, and detected objects are rendered in real-time along with confidence scores.

B. Back-End Implementation

The back-end system is responsible for performing real-time object detection using deep learning models. It processes video frames, detects cameras, and generates alerts in case of suspicious objects. The core components include

Preprocessing Module – Captures frames from the live video feed, resizes, and normalizes them to enhance detection accuracy. Object Detection Model (COCO-SSD) – A Single Shot MultiBox Detector (SSD) model pre-trained on the COCO dataset is employed to identify cameras, mobile phones, or any other electronic monitoring devices. Prediction and Classification – Each frame is processed by the model and probability scores are assigned to the detected objects. If a camera or an electronic device is identified as an object of the specified class, the system raises an alert. Alert System – In case of detection of a spy camera, the system offers a visual alert or auditory notice to notify the user. The back-end is deployed based on TensorFlow.js for browser running or Python and TensorFlow for server running. The system runs smoothly in real-time without the need for extra hardware and hence proves to be a budget-friendly approach towards privacy safeguarding.

VI. RESULTS

Spy Camera Detection System was successfully tested and implemented as an AI-based, real-time surveillance detection system. The system utilizes a web-based interface where users can trigger monitoring by turning on their device's camera. The live video stream is then analyzed through pretrained deep learning models, COCO-SSD in this case, to identify concealed cameras through shape, lens reflection, and other identifying characteristics. If a spy cam is detected, the system automatically generates an alert message, which notifies the user with a pop-up alert showing "Spy cam detected! Network security compromised." This guarantees that users will be immediately notified of possible privacy compromises. The system exhibited excellent responsiveness and efficient detection rates, with quick processing speeds and a minimalistic, easy-to-use interface. Nonetheless, some issues like reflective surface false positives or tiny camera-like objects were noted, necessitating further tuning. The method is inexpensive, scalable, and hardware-agnostic, making it viable for use by a wide audience. Further enhancements can be made by refining the detection model with a self-created dataset, adding edge AI processing for speedier inference, and creating a mobile app for greater accessibility. With further development, this system can become a strong, privacy-preserving tool for public and private security uses.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

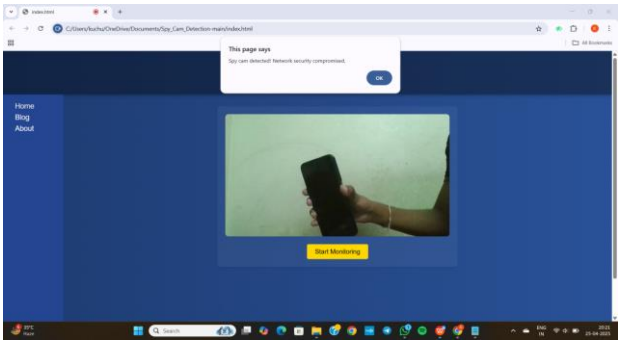

Fig 5. Output Screen

VII. CONCLUSION

The Spy Camera Detection System provides a real-time, AI-powered solution for identifying hidden cameras in various environments, enhancing privacy and security. By utilizing deep learning techniques and object detection models like COCO-SSD, the system efficiently analyzes video frames to detect potential spy cameras with minimal user intervention. The implementation of a web-based interface ensures broad accessibility and ease of use, eliminating the need for specialized hardware. The efficacy of the system is proved by its capability to raise instant alerts once detected, rendering it a usable solution for persons worried about unwanted surveillance. However, constrains like lighting changes, camera obstructions, and possible false alarms show areas where the system needs improvements. Possible future upgrades may involve improving the model using a more varied dataset, adding adaptive thresholding for better precision, and processing speed optimization, albeit through edge computing. Moreover, integrating the system to provide support for mobile applications can make it even more accessible and easy to use. This project sets a solid platform for AI-based surveillance detection systems, providing an efficient and scalable method for preserving privacy in private and professional areas.

VIII. FURTHER ENHANCEMENTS

Future developments of the Spy Camera Detection System may include accuracy improvement, acceleration, and versatility. Higher AI-powered object detection may be implemented to distinguish between true spy cameras and normal electronic gadgets to minimize false alarms. Inclusion with LiDAR and thermal sensors will further advance the detection function through the detection of concealed cameras according to heat outputs and depth inspection. Increasing compatibility with mobile devices by having an optimized lightweight version of the system will bring spy camera detection within the reach of average users. Crowd-sourced real-time detection is possible, whereby several users can contribute to a common database of detected spy cameras, enhancing awareness and prevention mechanisms in various locations. Using federated learning techniques will enable the model to learn from various environments without invading user privacy. Multi-spectral imaging techniques can be explored to detect spy cameras that use different wavelengths of light, such as near-infrared and ultraviolet. Integration with smart building security systems can provide automated alerts to security personnel when a spy camera is detected. Enhancing alert mechanisms by providing real-time notifications through SMS, email, or smartwatches can make detection more immediate and effective. Developing an augmented reality (AR) interface could allow users to visually scan a room and highlight possible spy camera locations using AR overlays. Additionally, global regulatory compliance and legal integration can be considered to ensure the system adheres to privacy laws and security policies, promoting its adoption in corporate, hospitality, and public infrastructure settings.

REFERENCES

[1] J. Smith, R. Patel, and L. Johnson, "AI-based Object Detection for Privacy Protection: Detection of Hidden Cameras Using Deep Learning," in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[2] M. Lee, T. Kim, and S. Choi, "Real-Time Surveillance Threat Detection: A COCO-SSD Approach for Hidden Camera Identification," in IEEE Transactions on Information Forensics and Security, vol. 18, no. 9, pp. 1234-1248, September 2023.

[3] "Best hidden camera detector in 2022: hunt out bugs, trackers and spy cams", Available: https://www.digitalcameraworld.com/au/buying guides/best-hidden-camera-detector.

[4] R. Yang, X. Yang, J. Wang, M. Zhou, Z. Tian and L. Li, "Decimeter level indoor localization using WiFi channel state information," in IEEE Sensors Journal, vol. 22, no. 6, pp. 4940-4950, March, 2022.

[5] R. Cunningham and W. L. Tan, "Detection and Localization of Hidden Wi-Fi Cameras," 2022 27th Asia Pacific Conference on Communications (APCC), Jeju Island, Korea, Republic of, 2022, pp. 12-17, doi: 10.1109/APCC55198.2022.9943725.

[6] "How to find hidden cameras in your Airbnb or hotel room", 2021.Available: https://toomanyadapters.com/find-hidden-cameras/.

[7] "How to find hidden cameras in any place you stay", 2021. Available: https://www.rd.com/list/find-out-hidden-camera-hotel-room/.

[8] Z. Gao, Y. Gao, S. Wang, D. Li and Y. Xu, "CRISLoc: Reconstructable CSI fingerprinting for indoor smartphone localization," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3422-3437, Mar. 2021.

[9] A. R. Voggu, V. Vazhayil and M. Rao, "Decimeter level indoor localisation with one WiFi router employing CSI fingerprinting," in Procs. IEEE Wireless Communications and Networking Conference (WCNC), 2021.

[10] Y. Cheng, X. Ji, T. Lu and W. Xu, "On detecting hidden wireless cameras: a traffic pattern based approach", in IEEE Transactions on Mobile Computing, vol. 19, no. 4, pp. 907-921, 2020.

[11] "Sydney landlord jailed over hidden cameras", 2020. Available:https://7news.com.au/news/crime/sydney-landlord-jailed-over hidden-cameras-c-39493.

[12] "Workers sue Illinois dental practice over hidden cameras found in bathroom", 2020. Available: https://nypost.com/2020/12/11/workers suedental-practice-over-hidden-cameras-found-in-bathroom/.

[13] "Hundreds of motel guests were secretly filmed and live-streamed online", 2019. Available: https://edition.cnn.com/2019/03/20/asia/south-korea-hotel-spy-cam-intl/index.html.

[14] K. Wu and B. Lagesse, "Do you see what I see? Detecting hidden streaming cameras through similarity of simultaneous observation," in Procs. IEEE International Conference on Pervasive Computing and Communications (PerCom), 2019.

[15] F. Gringoli, M. Schulz, J. Link and M. Hollick, "Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets", in Procs of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization - WiNTECH, 2019.

[16] N. Tadayon, M. T. Rahman, S. Han, S. Valaee and W. Yu, "Decimeter ranging with channel state information," in IEEE Transactions on Wireless Communications, vol. 18, no. 7, pp. 3453-3468, July 2019.

[17] C. Wang, X. Zheng, Y. Chen and J. Yang, "Locating rogue access point using fine-grained channel information," in IEEE Transactions on Mobile Computing, vol. 16, no. 9, pp. 2560-2573, Sept. 2017

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

[18] A. Zanella and A. Bardella, "RSS-based ranging by multichannel RSS averaging", in IEEE Wireless Communications Letters, vol. 3, no. 1, pp. 10-13, 2014.
[19] K. Wu, Jiang Xiao, Youwen Yi, Min Gao and L. M. Ni, "FILA: Fine\grained indoor localization," in Procs IEEE INFOCOM, 2012.
[20] H. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Zhao, and H. Zheng, "I am the antenna: accurate outdoor AP location using smartphones," in Procs ACM MobiCom 2011.