Journal of Asian Scientific Research
ISSN(e): 2223-1331
ISSN(p): 2226-5724
DOI: 10.55493/5003.v15i4.5751
Vol. 15, No. 4, 796-814.
© 2025 AESS Publications. All Rights Reserved.
URL: www.aessweb.com

AHCL: AI-based real-time hidden video stream identification platform

Shashidhar R[1]
Suraj V[2]
Vinayakumar Ravi[3+]
Varadaraj G[4]
Akshay Kumar[5]

[1,2,4,5]*Department of Electronics and Communication Engineering, JSS Science and Technology University, Mysuru-570006, India.*
[1]*Email: shashidhar.r@jssstuniv.in*
[2]*Email: surajvviswanathad@gmail.com*
[4]*Email: varadarajug51@gmail.com*
[5]*Email: akshayyk210@gmail.com*
[3]*Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia.*
[3]*Email: vravi@pmu.edu.sa*

(+ Corresponding author)

Article History
Received: 18 August 2025
Revised: 10 October 2025
Accepted: 4 November 2025
Published: 25 November 2025

Keywords
AI-Aided surveillance detection
Deep learning
Ensemble learning
Hidden camera detection
IoT network traffic analysis
Network traffic classification
Privacy protection
Support-vector machine.

ABSTRACT

The objective of the research is to address increasing privacy and safety issues related to hidden cameras in personal spaces such as bedrooms, bathrooms, or dressing areas, where such cameras can wirelessly transmit video signals clandestinely. The aim is to develop and evaluate an Artificial Intelligence (AI)-Enabled Hidden Camera Localization (AHCL) platform capable of identifying and locating hidden video streams through analysis of real-time network traffic. The methodology involves packet capturing, statistical analysis, and deep learning-based classifiers to detect anomalous streaming traffic in captured packets. The research generated a dataset comprising 60,412 packets, labeled as either 'normal' or video streaming, which was used to train and evaluate several models, including Support Vector Machines (SVM), Denoising Autoencoders, and ensemble deep learning models. The experimental results indicate that the ensemble model achieved the highest performance, with a detection accuracy of up to 98.27%, demonstrating good generalization and robustness across different network environments and over multiple days. The findings show that the AHCL platform is highly reliable in detecting hidden camera traffic from benign traffic. The practical contribution of this research is significant, providing users with an intelligent and affordable system for real-time privacy protection that can be deployed in residential or commercial settings, thereby enhancing trust and safety in a connected environment.

Contribution/ Originality: This work builds on established prior research in the area of hidden camera detection, utilizing both network traffic analysis and ensemble deep learning models. This study contributes to a new dataset containing labeled video and non-video packets. Our primary contribution is the accurate real-time localization of hidden cameras, minimizing false positives, and enhancing user privacy.

1. INTRODUCTION

As smart devices become more prevalent and inexpensive wireless camera equipment becomes more accessible, privacy concerns about hidden cameras are still prevalent in both private and public locations. Cameras of this nature are often camouflaged within everyday items smoke detectors, alarm clocks, USB charging ports, or even light bulbs rendering them difficult to detect through lay inspection. As a result, there is a need for an intelligent automated detection system that can assess privacy breach potential in multiple environments, such as hotel rooms, changing areas, and short-term rental accommodations. For many years, clandestine camera detection methods such as infrared scans, lens reflections, and visual inspections have been implemented. However, these methods can take time, rely on the human factor, and are not always reliable when controlling for modern technology [1, 2]. In comparison, hidden cameras can transmit a video feed to wireless networks and, therefore, produce unique digital footprint evidence that digital investigators can capture for future analysis. Once investigators capture packet data, they can identify statistical patterns to differentiate between "normal" bandwidth traffic and video streaming surveillance, where cameras/remotes could also be physically concealed during inspections [3, 4].

This principle is at the core of our AI-assisted Hidden Camera Localization (AHCL) system, which employs machine learning and deep learning algorithms to detect and localize hidden video cameras in real-time. Previous research in this domain has yielded valuable insights; however, they all exhibit certain limitations: a. many existing detection models fail to distinguish benign video streams (e.g., video conferencing or media streaming) from covert surveillance streams; b. most approaches lack resilience to network noise and tend to produce high false-positive or false-negative rates; and c. many methods only detect that a device is transmitting a video stream but do not localize the source of the stream Liu, et al. [5] and Heo, et al. [6]. These limitations underscore the need for an integrated detection and localization approach that accurately identifies hidden cameras, withstands network noise, and pinpoints the camera's location. The increasing frequency of documented incidents involving hidden surveillance in hotels, public restrooms, and shared accommodations clearly indicates a pressing need for accessible solutions that do not require extensive technical expertise. Current surveillance detection tools are either prohibitively expensive or overly complex for everyday users. To address this gap in performance and accessibility, we propose the AHCL, which offers a low-cost, efficient, and user-friendly method for self-adaptive detection of hidden surveillance based on traffic analysis and artificial intelligence (AI). The AHCL distinguishes itself from conventional strategies through its self-adaptability across various environmental and network conditions, ensuring practicality for real-world applications.

The main contributions of the anticipated work are:

1. Develop or extract labeled datasets via a Wi-Fi sniffing tool.
2. Develop a robust methodology for detecting hidden cameras.
3. Implement efficient algorithms for image and signal analysis.
4. Minimize false positives using advanced filtering techniques.
5. Create a cost-effective and user-friendly solution.
6. Ensure adaptability to various environmental conditions.
7. Evaluate and optimize the detection framework's performance.

With these contributions in mind, the research aims to (i) design and implement a method for finding hidden video cameras based on the analysis of network traffic, (ii) evaluate the method under realistic conditions, and (iii) determine its applicability in different network settings. The central thesis of the research is that ensemble learning methods with denoising will lead to greater accuracy and lower false-positive rates for detection than currently available single-model approaches to identify hidden cameras and improve reliability and accessibility to localizing hidden cameras. The rest of the paper is organized as follows: a review of related literature in Section II identifies limitations and gaps in traditional and AI-based approaches; Section III describes the proposed architecture and method of the AHCL system, including data collection, feature extraction, classification, denoising, and localizing; Section IV presents the experiment design, results, and the evaluation of classifiers' performance; and Section V concludes with a summary of the findings, including limitations and future research directions to build out the AHCL system and take it into the field.

2. LITERATURE SURVEY

The detection of hidden cameras is utilized in a number of ways, primarily through RF scanners, optical methods, and detection models based on artificial intelligence, especially CNN-LSTM. RF scanners and optical methods are more limited in the passive devices they can detect, and new ideas such as DeWiCam have been developed to utilize a smartphone's promiscuous mode and other analyses of wireless signals for additional opportunities. Commercial wireless hidden camera detectors have also faced challenges with new encrypted Wi-Fi hidden cameras, leading to the development of alternative solutions using thermal imaging and IoT. Hybrid detection models like DeepDeSpy are capable of using channel state information (CSI) data for data-driven detection in real-time [3]. Hidden camera detection methods were still only solved by Wi-Fi traffic analysis or laser-based retro-reflection, and are still limited by non-transmitting devices or just the raw cost of technology. One more unequivocally useful need was for depth-based sensing with reflective properties because time-of-flight (ToF) sensors, our sensors, could provide real-time detection analysis through depth. The worst detection rate was less than 30%, and even the best detection systems provided a detection rate of around 62.3%. The smartphone-based LAPD system reported using ToF sensors to detect hidden cameras at a rate of 88.9%, which was a significant development and warranted some attention [1].

Recent improvements in object detection and tracking systems, particularly useful for video surveillance of real-world contexts, have started to address fully occluded objects in complex backgrounds with occlusions and changes in illumination as part of the surveillance context. New methods have also been developed based on the use of symbolic reasoning to better augment computer vision techniques to facilitate detection. Although many statistical algorithms exist, the scale required to account for obstructions and occlusions has posed barriers when performing detection plans for this type of study. The field has also begun to utilize AI systems and knowledge-based reasoning to improve detection strategies for typical occlusions in complex areas [2]. The rapid growth in clandestine wireless cameras, specifically, hidden cameras in private areas like hotel rooms, raised serious privacy concerns. Traditional methodologies to detect cameras depend on expensive and impractical hardware that requires trained technicians and is not feasible for everyday users. LocCams created a phone-based approach to detect cameras based on packet transmission rates and channel state information, achieving an impressive 95.12% classification accuracy. LocCams reduced the barrier to entry for surveillance detection and raised the need for affordable and simple surveillance detection options [4].

Research suggested that hidden cameras were gaining popularity in sensitive settings because they could easily be concealed. There are mobile applications that allow users to detect magnetic emissions, but the human ability to detect hidden cameras remains difficult. New technologies, such as laser retroreflection and machine learning, have allowed for an even greater ability to identify hidden cameras. It was clear that there was a need for reliable detection technologies to address privacy concerns, and this would only improve as new technology and continued development were done [5]. The paper had a specific focus on the various vulnerabilities present in the operating systems, authentication, and insecure transmission of data, which led to the potential remote operation of hidden cameras that compromised a person's privacy. The paper also discussed how the manufacturers of the devices had minimal responsibility to provide any reader reliability for the levels of security that were part of the detection technology [7].

Detecting Wi-Fi-enabled spy cameras has traditionally proven to be too complicated for an average user because of data demands and encryption barriers. The new approach proposed here involves leveraging devices such as a Raspberry Pi to achieve real-time detection, classification, and localizing of a camera, all of which are not available with previous approaches. The research proposed using the Nilsimsa algorithm and RSSI values to assess the location of a camera with 30cm accuracy. The newly proposed approach demonstrated more efficacy and practicality than previous ML models [8]. This literature review of object detection models categorizes them into four distinct types of detections, which include both anchor-based and transformer-based methods. Models were compared to previous reviews, which criticized them for being too limited. The previous reviews failed to conduct a broad evaluation of object detection models, metrics, and datasets like MS-COCO. The literature review described key developments in detection accuracy and detection methodologies over time [9].

Detecting WLAN spy cameras was difficult because of uncertainty related to device behaviors, and, likely, the methods that use CSI, ML, passive RF, and Wi-Fi usage may hold promise. Most past methods of detection required poor data collection methods with complex hardware, and there is a direction for more scalable and efficient solutions [10]. Hidden cameras were an emergent privacy issue, particularly for women at community facilities. Even with good methods for discovering hidden cameras, it was always difficult to locate wireless cameras when they were small. Recently, new capabilities became available that defined work by identifying them based on detecting and jamming signals in the 0.1GHz-3GHz range. The intended outcome for these devices was to increase public safety and improve evasion for privacy reasons, in an immediate way in the real world [11]. In terms of the current detection methods, such as RF, optical, and thermal, they all have usability problems and require expertise to use. HeatDeCam used smartphone thermal imaging and a neural network to identify latent thermal images as evidence, easily and without cables or wires. In excess of 22,000 images, the researchers were able to achieve over 95% accuracy in reports. HeatDeCam was also an important milestone along the way toward real-world solutions for the detection of hidden cameras [12].

The progression of object detection showed how the field of detection had moved from modeling with algorithms employed in turn phase processing to convolutional neural networks (CNNs) using YOLO modeling. YOLO had increased the speed of inference and changed the universal definitions throughout different forms for speed and item counts, and accuracy had suffered somewhat as it used deeper, over-saturated information mean and mean pooling; the vision field in the computing aspect. The research being shown with the enhancement of YOLO architecture, parameters, detections, and included performance has been researched, which also included an exhaustive history of the deep learning era (The YOLO model era detections) [13].

This work was to integrate IoT and RFID to detect and jam hidden cameras. Operating with components like Arduino or Wi-Fi modules, the system was able to demonstrate accuracy and applicability in real-life scenarios. The framework improved privacy and security issues posed by surveillance situations. There were suggestions for further improving IoT security against unattended recording [14]. There were numerous detection systems; however, most are complex, hardware-intensive efforts or inefficient. Retro-reflection, magnetic sensors, and IR transmitters were just a few possible approaches. Object tracking and servo-driven disactivations were other possibilities. So many of the "practical" systems were attempted; however, there is a clear need for systems that are much, much simpler and better [15]. Traffic classification could not keep pace with changing networks. The study demonstrated that machine learning techniques, specifically the XGBoost model, can classify traffic with a high level of accuracy (99.97%) that could be used for Traffic Classification (TC) purposes. The authors used min-max performance indicators and scaling methods to improve results, which led them to determine that Machine Learning (ML) holds the potential to be helpful for optimal network management [16].

With complex threats and high traffic loads in university networks, a network monitoring system was essential. PRTG and Sophos Firewall were integrated applications providing real-time monitoring. Changing security systems from manual to automatic adaptive security systems. This study supports smarter, optimal network management decisions for university networks based on all aspects of your university network system [17]. LBP (Local Binary Patterns) was widely used in texture analysis and is suitable for practical applications and images. Although LBP was also good at distinguishing textures, it was unsuitable for noise. Other approaches, like LBCNN or LTCP, have shown better accuracy and robustness. The study proposed a method, LTBP, which takes advantage of spatial proximity, and where image features were extracted, unique to noise. This study has been particularly promising for use in several classification scenarios, such as biometric analysis and medical image classification [18]. This USA-based CNN-based spy camera detection system uses the lightweight ResNet-18 model to achieve a mobile-friendly outcome. Although initial classification accuracy studies investigated other models, such as MobileNets and Inception, ResNet was the only model considered for this investigation. The rationale was that ResNet-18 offered a balanced performance and effective feature extraction. This enables the surveillance device to accurately perform real-time detection of hidden cameras while operating within the constraints of mobile hardware devices [12].

Most spy camera detection systems relied on using the real, recognizable physical signals (i.e., RF, lens reflections) that hidden camera systems generated, or the distractions or amalgams of these signals, which current users could interpret. Similarly, most existing systems were fundamentally not accurate for a number of reasons, or were not tuned to an accurate enough level for user intervention. Also, hidden cameras could be affected by other electronic devices. There are major usability and clarity issues related to cueing feedback from detection [6]. Hidden Wi-Fi cameras transmitting video feeds likely generate unique packets that could be tracked while in use. Previous work has focused on detecting presence (Wi-Fi camera exists), and not necessarily locality or vicinity. This work aimed to extend previously done work in terms of camera detection and focus on only taking geographic locations of Wi-Fi cameras with evidence of network traffic data. Previous work would have focused solely on detection, while this work would have created tools for surveillance risk mitigation [19]. This paper provides a comprehensive survey of the problems and progression of the state-of-the-art detection of small objects from aerial images using deep learning. The methods are classified into five categories, existing datasets are described, and the results of the experiments are compared to illustrate the advantages and weaknesses of the existing algorithms. The paper offers future work suggestions to improve detection in a complex airborne imaging context [20].

In the paper "Spying on the Spy: Security Analysis of Hidden Cameras," researchers assessed the security of the spy camera that is commercially available and found significant, almost unresolvable vulnerabilities, including the unencrypted wireless transmission of data, firmware vulnerabilities, and the use of default password presets that could expose consumers to exploitation. The poorly regarded consumer surveillance devices posed a high security risk to consumers and suggested more thoughtful regulation around these types of devices and greater efforts to improve security practices [21]. They presented a detection system for hidden cameras through a laser reflection technique. These systems could be useful to protect user privacy in sensitive locations such as courtrooms and hotel rooms, where the surreptitious recording of video could legitimately violate privacy [22].

The article offers a comprehensive survey related to deep learning-based intrusion detection systems (IDS) on automotive networks. We organized the different deep learning schemes by topology and techniques, highlighting the characteristics of each scheme. Next, we broke down the evaluation of each scheme in terms of datasets, attack types, and metrics. Finally, we analyze the comparisons, evaluations, results, and pros & cons of the various deep learning architectures [23]. This article provides a comprehensive and current review of deep learning-based methods developed to detect violence within video surveillance data, specifically physical assaults. It organizes existing methods, datasets, and challenges, and outlines the current state of artificial intelligence-enabled violence detection, providing avenues for future research [24]. The researcher provided a comprehensive survey of deep learning-based rejection class approaches to perform anomaly detection in video surveillance. The survey highlighted the models that were developed to detect anomalies in video data, outlined the challenges and use cases, and evaluated metrics observed in anomaly detection research. The survey also outlined opportunities for future space for improving anomaly detection systems to be more comprehensive and complex, as seen in real-world surveillance [25].

The authors presented a case study on deep learning approaches, focusing particularly on convolutional neural networks (CNNs) to recognize digital cameras by identifying the unique photo-response non-uniformity (PRNU) noise specific to their images. The authors trained a modified AlexNet model and achieved a maximum identification classification accuracy between 80% and 90%. The assessment identified several weaknesses, including false identifications and the inability to distinguish cameras from the same manufacturer, which suggests that further refinements in deep learning could improve models for forensic applications [26].

This manuscript introduces LAPD, a platform that utilizes smartphone Time-of-Flight (ToF) sensors to locate hidden cameras via reflections of light from surfaces. In assessing the ToF sensor output, this system was able to classify materials that reflect light vulnerabilities associated with hidden cameras. This indicates that the prospective availability of detection provides inexpensive and highly accessible detection through the perspective of a smartphone [1].

This manuscript details a novel solution for finding wireless spy cameras through the monitoring of wireless electromagnetic fields emitted by wireless devices. The precise solution monitors the wireless response of the wireless device to a stimulus and reveals hidden cameras in real time, and once the wireless signals from a device are emitted, it can probe and locate devices that would not have been detected or visible through another medium [27]. As it pertains to the detection of hidden cameras, AI-powered methods such as CNNs, LSTMs, and Transformers enhance detection algorithms. These methods leverage real-time RF and improve detection by utilizing visual transformers to enhance visual detection; they also leverage Federated Learning to diminish privacy concerns [28].

The manuscript reviewed machine learning methods for network traffic classification with supervised learning, semi-supervised learning, and unsupervised learning methods. It provided a review of the methods used in reviewing network traffic [29].

Thermal imaging offers the ability to detect heat signatures, and Internet of Things (IoT) sensors provide potential low-cost options. Drones offer greater coverage, and edge computing might enable real-time processing [30]. The authors presented their application of deep learning methodologies to predict network traffic, covering convolutional neural networks (CNN) and long short-term memory (LSTM) networks. They provided insights into their ability to predict network traffic patterns and the possible impacts on future network performance [31]. This paper surveyed a number of techniques for the recognition of behavior based on the use of Wi-Fi Channel State Information (CSI). The paper discussed the potential use of CSI in terms of obtaining fine-grained motion data to detect or identify human activities, along with challenges and approaches applied in behavior recognition applications. It demonstrated the potential of CSI techniques for behavior monitoring in a non-intrusive way [32].

The paper explains CSI: DeSpy, which also provided passive sensing to identify spy cameras. When the WiFi Channel State Information (CSI) changed with user activity, the device was recording video and was likely to be a hidden camera. As the user moved, the bitrate changed, and potentially other parameters such as delays, jitter, RSSI, etc. This only required passive signals, did not require any active probing, and was an easier and still effective way to detect video surveillance in a seamless manner [33]. The paper described LocCams, a powerful system for detecting and localizing hidden wireless cameras using commodity devices. The system utilizes the wireless signal patterns associated with wireless security cameras and leverages the proprietary hardware already built into smart devices and other consumer technologies to classify hidden cameras and localize them. It is a virtually favorable and economical way to enhance privacy and security in a myriad of spaces [4].

The pre-training process utilized a significant amount of unlabeled data, which improved the model's ability to identify wireless cameras. Subsequently, a neural network - long short-term memory (NN-LSTM) classifier determined the presence of wireless cameras and indicated it may be a feasible option to detect unauthorized surveillance equipment quickly and efficiently in a real-time capacity [34]. This paper presents WiSOM, a self-adjusting system for occupancy monitoring in smart buildings using WiFi. The approach to detect and identify the presence of people involves analyzing the different strengths of WiFi signals. Thus, it efficiently utilizes WiFi readings to provide data for energy management and building management. The system can adjust or adapt to different levels of sensitivity for a range of environments and user behaviors, enhancing effectiveness and accuracy [35]. The basic premise of this approach was that the timing patterns of the data sent from hidden streaming cameras could be correlated with the timing patterns of a known camera recording in the same scene. The proposed approach included multiple similarity measures to demonstrate its high accuracy, with typical F1 scores greater than 0.95 for both indoor and outdoor scenarios [36].

The device used a risk assessment framework and identification of hidden cameras through scanning for radio frequency (RF) signals in the frequency range of 0.1 GHz to 3 GHz, which is within the operating frequency range of most wireless cameras. When the device identified a radio frequency signal, it would provide two responses to notify the user of the identified camera. First, the device would activate a visual alert by illuminating a red light indicator. Second, the device would send a jamming signal to deactivate the operation of the identified camera. The device designed by the researchers specifically identified and disabled devices that were captured in surveillance footage (electronic and visual), as a proactive measure to secure privacy in situations where unknown hidden cameras may have been present [10].

Lee, et al. [37] have developed an AI-aided Hidden Camera Locator system that detects hidden cameras from raw IoT network traffic. SVM classified video traffic, and a Denoising Autoencoder improved the quality of this data. After the data was enhanced, a neural-based classifier located the camera. In ensemble models of MLP, 1D-CNN, and Inception ResNet v2, the method was able to detect a camera with an accuracy of 97.65%. This method allows for real-time detection of cameras without requiring the reconstructed video to be fully stored [37].

Despite advancements in hidden camera detection approaches, there are important shortcomings in existing research. Many approaches are incapable of accurately classifying benign video traffic versus hidden video surveillance streams, leading to extremely high false-positive rates. Others depend on costly hardware, manual inspection, are sensitive to encryption, or rely on passive surveillance devices. Only a small number of systems attempt to localize the source of hidden cameras, thereby limiting their value in real-world applications for users. The fully automated AHCL system aims to eliminate such limitations with machine learning and deep learning models for network traffic classification and denoising, and additionally uses ensemble learning to detect local hidden cameras. Furthermore, the AHCL is designed to be cost-effective, easy to use, and adaptable to environmental conditions, making it a feasible and viable solution for everyday users concerned about their privacy.

3. METHODOLOGY

The AHCL system methodology provides a step-by-step procedure employed for the detection and localization of hidden cameras through network traffic analysis. The block diagram of the system is provided in this chapter, with a description of each component in mathematical terms, along with an explanation of the architecture in detail.

3.1. Block Diagram

The AHCL framework is composed of the principal components: the Monitoring Node and Edge Server. The Monitoring Node is tasked with capturing network traffic in real-time, and the Edge Server is responsible for performing complex packet classifications, denoising, and camera localization. The block diagram illustrates the process from raw capture to the prediction of the actual visible camera's location. Figure 1 shows the block diagram of the proposed methodology.

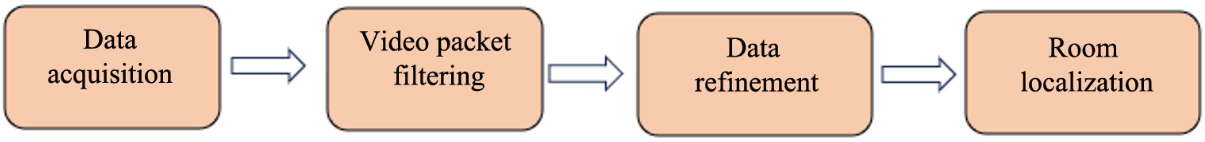


Figure 1. Block diagram of the proposed method.

3.1.1. Data Acquisition and Accessibility

In the first stage, a data set was created with the immediate goal of demarcating video streaming traffic from normal non-video traffic, a sample dataset as shown in Table 1. Real-time packet captures were performed using network capture applications such as Wireshark and tcpdump in a local area network (LAN) with a variety of devices. The data collection was varied as well across a range of network conditions, both high-bandwidth and low-bandwidth, as it was possible to make this data rich and representative. Traffic was recorded from video activities like streaming videos (YouTube, Netflix, IP camera streams) and non-video activities like browsing, file transfer, emails, and gaming. All the recorded network packets were hand-labelled either as "Video" (for streaming traffic) or "Non-Video" (regular traffic), and thus a dataset consisting of 60,412 samples 31,251 Video samples and 29,055 non-Video samples was created. In the data cleaning stage, important features included timestamp, destination and source IP addresses, the protocol type (TCP/UDP), the length of packets, and the content in the field information were identified. Duplicate and incomplete records were removed to make it more consistent. The cleaned data was then separated into a training data set (80% or 48,329 samples) and a test data set (20% or 12,083 samples), and the data was converted into CSV format to allow for model building.

In consideration of privacy and security, the dataset produced from this study will not be available to the public. However, the methods of data collection and the feature extraction process are provided in detail in order to allow for reproducibility by other researchers. Future work will investigate the possibility of the release of an anonymous dataset for general use.

Table 1. Sample dataset.

No.	Time	Source	Destination	Protocol	Length	Info	Label
47309	90.843	192.168.19.27	192.168.19.49	TCP	70	81 → 2555 [ACK] Seq=36611105 Ack=1 Win=5362 Len=16	Non-Video
48320	92.701	192.168.19.49	192.168.19.27	TCP	54	2555 → 81 [ACK] Seq=1 Ack=37390816 Win=64620 Len=0	Non-Video
55031	107.49	192.168.19.49	23.98.86.4	TCP	54	9465 → 8883 [ACK] Seq=63 Ack=63 Win=508 Len=0	Non-Video
11064	20.152	192.168.19.49	192.168.19.27	TCP	54	2555 → 81 [ACK] Seq=63 Ack=8438413 Win=64620 Len=0	Non-Video
46064	88.647	192.168.19.27	192.168.19.49	TCP	70	81 → 2555 [ACK] Seq=35658972 Ack=1 Win=5362 Len=1436	Video
31415	61.895	192.168.19.49	192.168.19.27	TCP	54	2555 → 81 [ACK] Seq=1 Ack=2425092 Win=64620 Len=0	Non-Video
11847	21.782	192.168.19.27	192.168.19.49	TCP	1490	81 → 2555 [ACK] Seq=9036229 Ack=1 Win=5362 Len=1436	Video
41222	80.014	192.168.19.27	192.168.19.49	TCP	719	81 → 2555 [ACK] Seq=31865780 Ack=1 Win=5362 Len=665	Video

3.1.2. Video Packet Filtering

After data acquisition, video packet filtering occurs. Feature engineering is then carried out to extract useful features from the packets (e.g., packet size, inter-transmission times, and ports). The features are combined into a feature vector, which can be represented mathematically as x [f1,f2,…, fn], where fi are the features that have been extracted. A Support Vector Machine (SVM) is used to classify packets as a video or non-video stream. The SVM uses a radial basis function, which has the kernel function definition K(x,x′) = exp(−γ‖x−x′‖^2), where γ is a hyperparameter that determines the smoothness of the kernel. The final classification decision is made using

$$f(x) = sign[\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b)] \qquad (1)$$

Packets classified as video traffic are then forwarded to the next stage of processing.

3.1.3. Data Refinement

A Denoising Autoencoder (DAE) is applied to improve the packets rated as video. The DAE is trained on pairs of clean and noisy packet representations to learn how to map noisy inputs to clean outputs. The DAE's mathematical formulation involves minimizing the Mean Squared Error (MSE) loss function.

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{N} ||x - \hat{x_i}||^2 \qquad (2)$$

Where xi is the original clean input and x^i is the denoised reconstruction. By denoising and removing spurious variations, the DAE provides localization models with only high-quality input features.

3.1.4. Room Localization

After applying denoising, feature vectors are passed to deep learning classifiers for room localization. For this task, three models are employed: Inception-ResNet-v2, a Multi-Layer Perceptron (MLP), and a 1D-Convolutional Neural Network (1D-CNN). These models learn the corresponding traffic patterns related to the rooms in which hidden cameras are located. The output of the neural network is computed through a Softmax layer, represented mathematically as

$$\hat{y} = \mathrm{Softmax}(Wx + b) \qquad (3)$$

$$\mathrm{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \qquad (4)$$

3.1.5. Ensemble Learning (Soft Voting)

To achieve the best possible accuracy in localization, ensemble learning is used through soft voting. This means that predictions from multiple models (Inception-ResNet-v2, MLP, and 1D-CNN) are summed up. The final decision is the class (room) with the maximum combined probability score from these models. The ensemble output is calculated as follows:

$$y_{\mathrm{final}} = \arg\max \sum_{i=1}^{m} p_{i,k} \qquad (5)$$

3.2. Procedure, Algorithm, and Architecture

The AHCL system follows a systematic methodology and algorithm to complete hidden camera detection and localization based solely on network traffic analysis. The process begins with Packet Capture, where raw network packets are captured in real-time from the monitored environment by utilizing tools such as Wireshark and tcpdump. Raw packets are captured, and Feature Extraction is performed, where relevant features, including packet size, packet transmission rate, and packet protocol type (TCP, UDP, etc.), are extracted to create a structured feature vector for each packet. After Feature Extraction, SVM Classification is performed. Each packet is classified as "video" traffic (a possible indication of hidden camera activity) or "non-video" traffic, using a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel. The decision boundary is determined mathematically using the SVM decision function. Figure 2 depicts the AHCL system architecture.

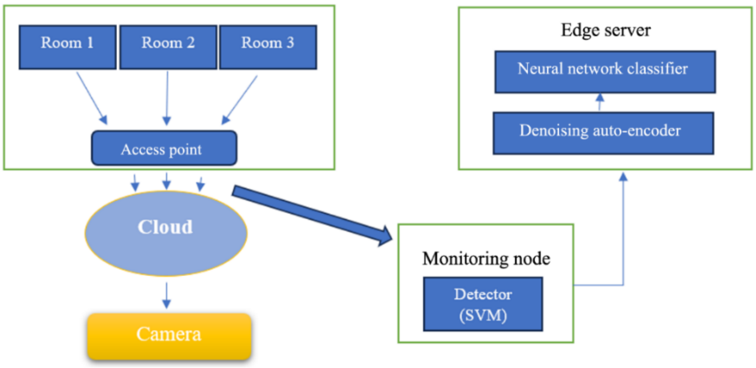


Figure 2. AHCL system architecture.

3.3. Algorithm: Hidden Camera Localization

3.3.1. Packet Capture

The first stage of the hidden-camera localization mechanism captures raw network packets in real-time. This is usually accomplished through the use of tools such as Wireshark or tcpdump. The raw packets are delivered over the network interface and can be stored in a packet capture (PCAP) file.

$$P \; = \; \{p1, p2, \dots, p_n\} \qquad (6)$$

This may be expressed mathematically as:

Where P is the set of captured packets, and each P_i is an individual packet. Each packet consists of the source IP address, the destination IP address, the timestamp, and the protocol type associated with the packet. This real-time acquisition becomes the raw data to perform further analysis.

3.3.2. Feature Extraction

The next step after capturing raw packets is to extract useful features that best characterize the traffic conditions. Important features include packet sizes, transmission rate r, and protocol type π. These features assist in discriminating video data from the rest of the traffic.

Packet size: Packet size refers to the size of a packet, usually denoted as

$$S_{avg} = \frac{1}{N}\sum_{i=1}^{N} s_i \qquad (7)$$

Transmission rate: This is the number of packets transmitted per unit of time. If T is the time window, the transmission rate r is:

$$r = \frac{N}{T} \qquad (8)$$

Protocol: π is the protocol, a one-hot feature indicating whether the packet is using TCP, UDP, or another protocol.

3.3.3. SVM Classification

After the feature extraction step, we can use a Support Vector Machine (SVM) classifier to classify video and non-video packets. The SVM will attempt to find a hyperplane that separates the packets of the two classes in feature space. The decision boundary based on the SVM classifier can be defined mathematically as:

$$f(x) = w^T x + b \qquad (9)$$

Where x is the feature vector (containing packet size, transmission rate, and protocol).

w is the weight vector and b is the bias term. The SVM classifier will learn the optimal values of w and b, which minimize the classification error.

Error when maximizing the margin between the two classes.

3.3.4. Denoising

After the classification step, there may still be noise within the video data. To further clean the video data, we utilize a Denoising Autoencoder (DAE), which can be trained to reconstruct a noise-free version of the input data. To this end, the DAE minimizes the reconstruction error:

$$\mathcal{L}(x, \hat{x}) = ||x - \hat{x}||^2 \qquad (10)$$

Where x is the noisy video input, $\hat{x}$ is the denoised output from the autoencoder. The DAE learns to map the noisy input to a lower-dimensional encoding, then reconstructs it back to the original video data, which in effect denoises the data.

3.3.5. Localization

Localization is performed by utilizing denoised video data to train deep learning classifiers to recognize the room or location where the hidden camera is situated. Localization is typically achieved using Convolutional Neural Networks (CNNs), which learn spatial hierarchies in image data; the output from the CNN can be treated as a classification problem, with classes corresponding to the locations within the building where the hidden camera is located.

Letting the output of the CNN be a distribution over all possible locations:

$$P(y|x) = \frac{e^{f(y,x)}}{\sum_k e^{f(k,x)}} \qquad (11)$$

3.3.6. Output

The final output is simply the predicted locations of the hidden camera, which is the highest probability P(y|x) produced by the deep learning model. This output can be shown on a graphical interface to display the predicted room or area where the hidden camera is located. The localization of the hidden camera (y) is simply the location with the highest probability.

$$y_{pred} = argmax\ P(y|x) \qquad (12)$$

This location (y) is the predicted location of the hidden camera by the algorithm based on analysis of network traffic. All of these steps packet capture, feature extraction, SVM classification, denoising, and localization with deep learning together will detect and localize hidden cameras with high accuracy from network traffic.

3.4. Experimental Setup

Figure 3 depicts the experimental workflow being carefully designed to simulate natural conditions closely for hidden camera detection and localization. Hidden cameras were made to stream video through Zoom in three different rooms, each representing different environmental conditions. While the video was being streamed, network packets were intercepted in real time to capture transmission characteristics related to hidden camera detection. The data of intercepted packets was then systematically collected, stored, and later used to train different AI models on feature extraction, classification, and localization.

3.4.1. Testing Models

To evaluate the models, the classifier was trained on packet data collected from all three rooms while allowing two different views per room in order to inject variability into the procedure, thereby modeling dynamic real-world conditions. This has also ensured that the models generalize with respect to different views and environmental parameters. Evaluation can also be carried out in several performance tests that include.

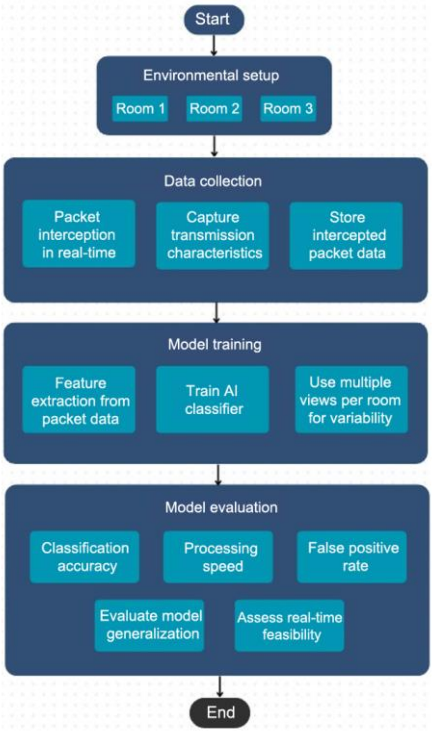

Start
Environmental setup
Room 1
Room 2
Room 3
Data collection
Packet interception in real-time
Capture transmission characteristics
Store intercepted packet data
Model training
Feature extraction from packet data
Train AI classifier
Use multiple views per room for variability
Model evaluation
Classification accuracy
Processing speed
False positive rate
Evaluate model generalization
Assess real-time feasibility
End

Figure 3. Experimental workflow.

3.4.2. Classification Accuracy

This pertains to the efficacy of the system in defining and localizing hidden cameras in diverse environments.

3.4.3. Processing Speed

This is the measurement of whether the system can perform in real time, thus enabling further consideration for deployment in immediate consumer and enterprise security solutions.

3.4.4. False Positive Rate (FPR)

This was kept monitored to ensure that misclassifications occurred as little as possible, so as not to generate unnecessary alarms and build up confidence in the predictions made by the system. This whole experimental setup proved the efficacy and accuracy of AHCL under realistic operational conditions, thus endorsing its scope for practical realization.

4. RESULTS AND DISCUSSION

This section provides results related to the application of the AI-based Hidden Camera Locator system (AHCL). It discusses the investigation of results based on accuracy, loss trends, validation graphs, and evaluation metrics. A detailed analysis of model performance using SVM and MLP classifiers is included.

4.1. Result using SVM

To understand the learning behavior of SVM, the accuracies and losses were plotted with respect to training iterations. In the top graph of Figure 4, the classification accuracy was progressively improved to over 97.55%, and at this point, it became stabilized. The stability of accuracy suggests a rich generalization ability and robustness of the model. The lower graph shows the loss plot, which dropped over time in a continuous, smooth manner during training. The nearly constant decline in loss indicates that the model could minimize more and more classification errors while tuning model parameters effectively. Therefore, all these observations prove that the SVM model has a good level of generalization over unseen instances of packet data, thereby making it very trustworthy for the real-time classification of packets between video and non-video streams. And Figure 5 depicts the validation loss being over 2.55%.

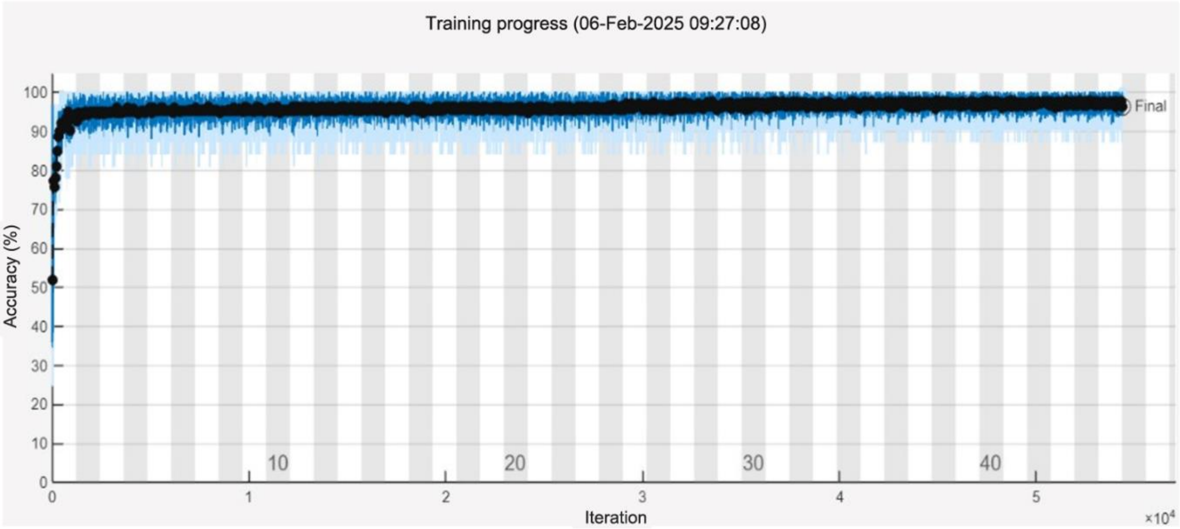


Figure 4. Validation accuracy curve.

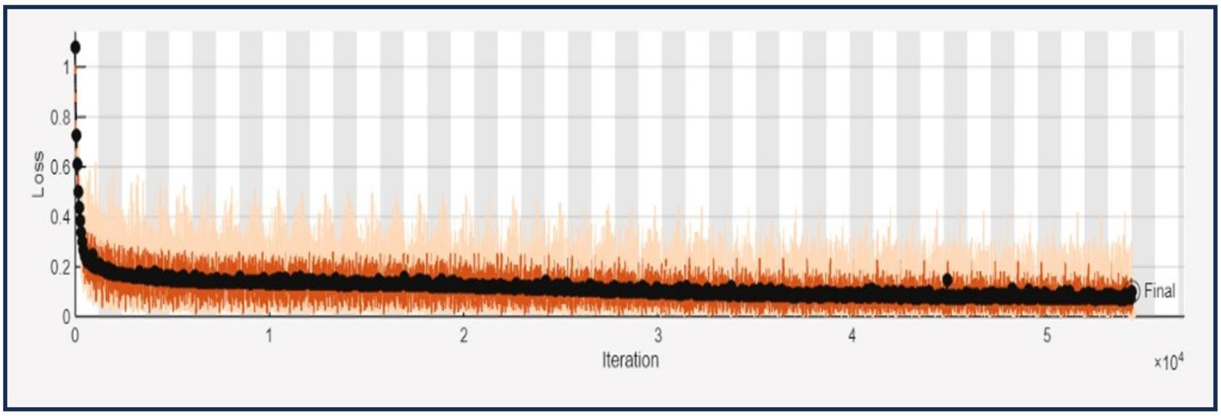
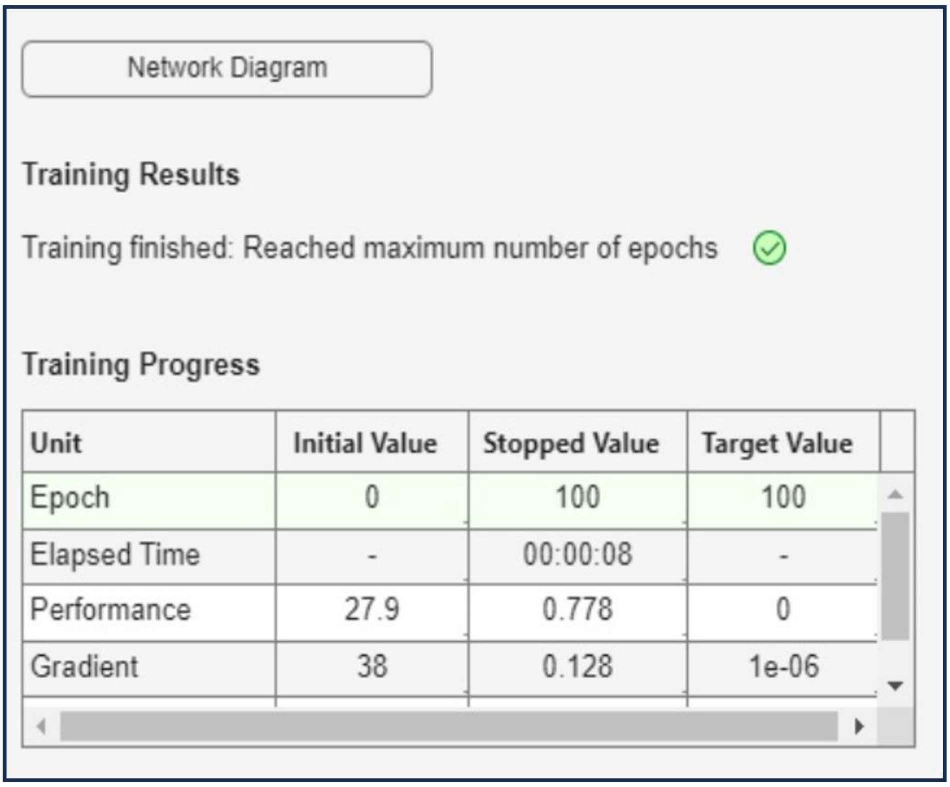


Figure 5. Validation loss curve.

4.1.1. Network Diagram

These epochs were optimized across multiple training stages for the SVM classifier. As training progressed, improvements in the model were noted, with a significant reduction in both gradient magnitudes and losses. Figure 6 shows the training phases through epochs, gradient updates, and the overall trend of the convergence. A complete training cycle was conducted for 100 epochs, after which the gradients stabilized, indicating that convergence was achieved and reliable learning behavior was observed.



Training Results

Training finished: Reached maximum number of epochs ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	100	100
Elapsed Time	-	00:00:08	-
Performance	27.9	0.778	0
Gradient	38	0.128	1e-06

Figure 6. Network diagram.

4.1.2. Scatter Plot Analysis of SVM

The SVM margins could be visualized in scatter plots presented in Figure 7. Herein, the set of parameters considered for this scatter plot includes transforming Packet Length into the x-axis and Packet Time into the y-axis, and demonstrating whether SVM is using the highest possible separation to decide whether it is a video or non-video packet. The complete separation visible in any plot is evidence of highly effective internal classification.

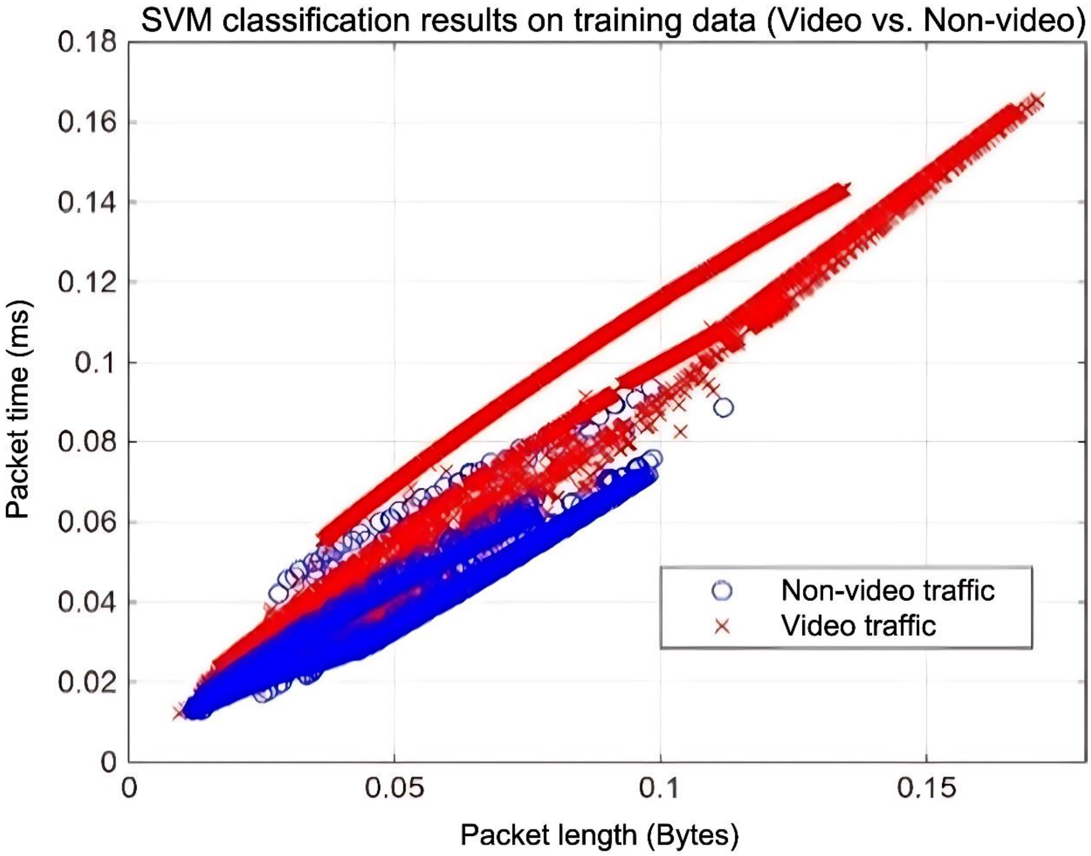


Figure 7. Scatter plot of SVM with packet length vs. packet time.

The performance of the SVM classifier on the task of distinguishing video packets from room classification is impressive. Support Vector Machines (SVM) proved to be very efficient, with an accuracy of 97.34 percent for detecting video packet transmissions across the network.

An ensemble neural network achieved an even higher accuracy of 99.5% in classifying the video streams against the room location. Further performance of the classification is validated using a confusion matrix. The diagonal entries indicate correct classification, while the off-diagonal entries correspond to misclassification. The higher presence of the diagonal elements in the matrix indicates high-precision classification by the model, thus supporting itself against the ground truth data.

4.1.3. Confusion Matrix

To illustrate the performance of the classification, Figure 8 plots the confusion matrix for the predictions by the ensemble model. Off-diagonal entries signify misclassification, and diagonal entries signify correctly classified items. From it, it is evident that the majority of correct classifications are in accord with the ground truth, verifying the correctness of the model.

The MLP model was further tested for video packet classification and room localization. The results are compared against validation patterns and a confusion matrix. Table 2 displays the classification report of the SVM model.

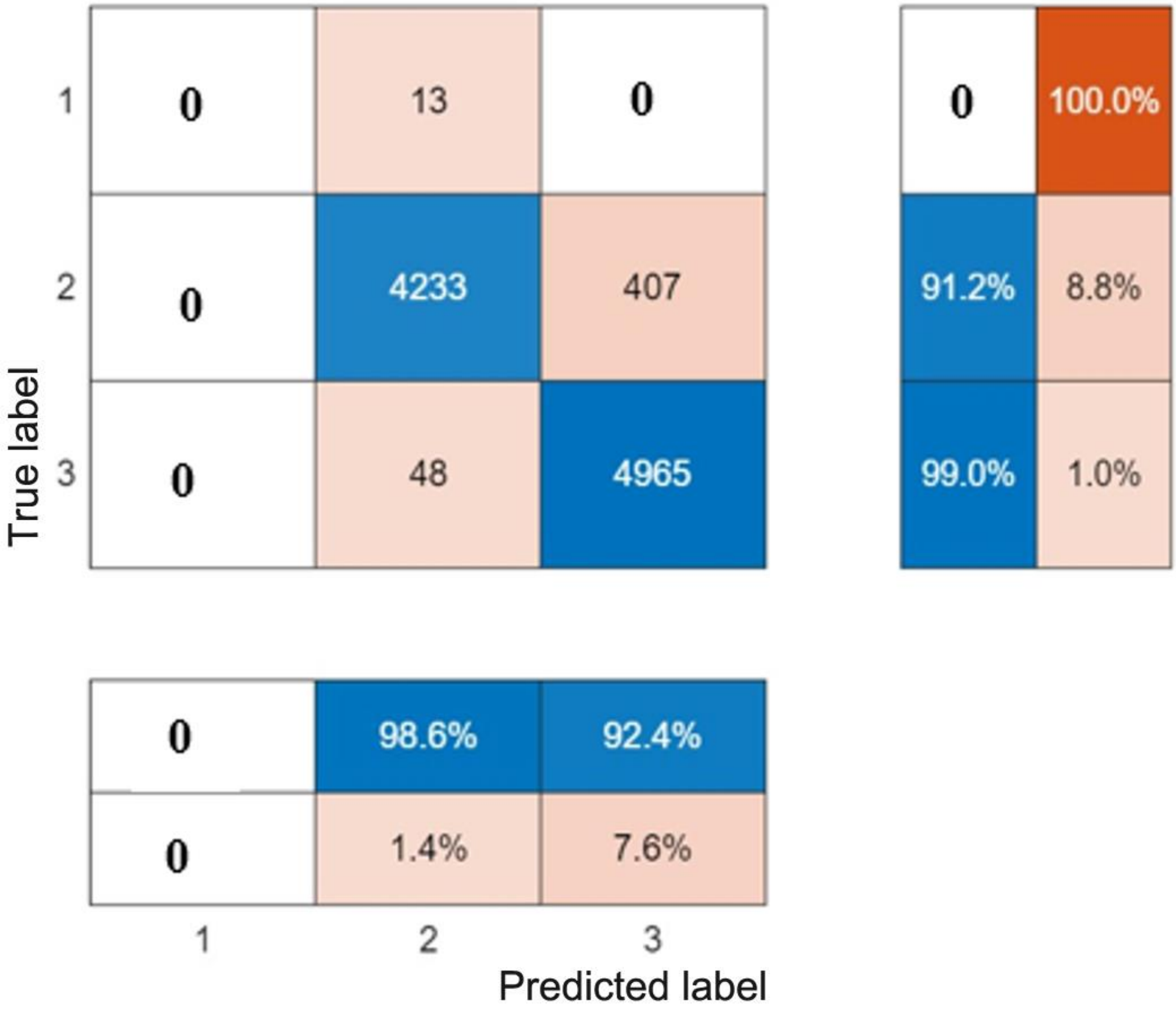


Figure 8. Confusion matrix.

Table 2. SVM Classification.

Class	Precision	Recall	F1-Score	Support
Non-Video	0.00	0.00	0.00	0.00
Video	0.98	0.98	0.98	18241

4.1.4. Results Using MLP

The Multi-Layer Perceptron (MLP) model was also evaluated for video packet classification and room localization. The results are analyzed based on validation trends and a confusion matrix.

4.1.5. Validation Graph

The validation accuracy and loss trends for MLP are shown in Figure 9. The model demonstrates steady learning, with accuracy stabilizing and loss minimizing over epochs. This confirms the network's ability to generalize well on unseen data. Figure 10 depicts the validation loss of the MLP, i.e., 1.73. Table 3 displays the training results of the MLP.

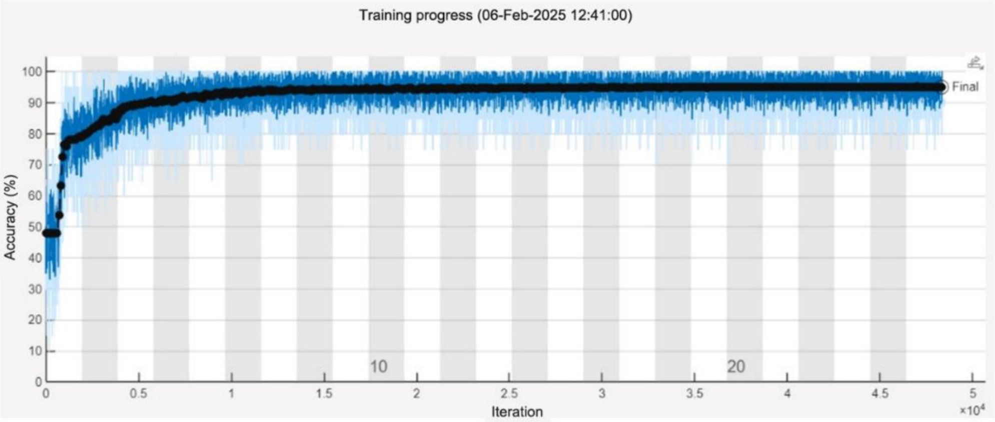


Figure 9. Validation accuracy of MLP.

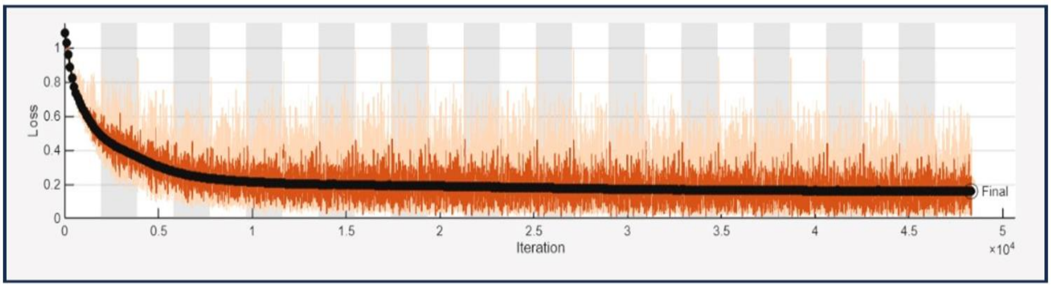


Figure 10. Validation loss of MLP.

Table 3. Training results of MLP.

Validation accuracy	Epochs	Iterations	Learning rate
98.27	25	48325	7.5e-0.5

4.1.6. Confusion Matrix

Figure 11 illustrates the confusion matrix of the MLP classifier. The well-classified samples are represented by the diagonal elements, and misclassifications are minimal, further supporting the high accuracy of the model. Table 4 displays a comparison of the performance of different models based on key evaluation metrics with existing results.

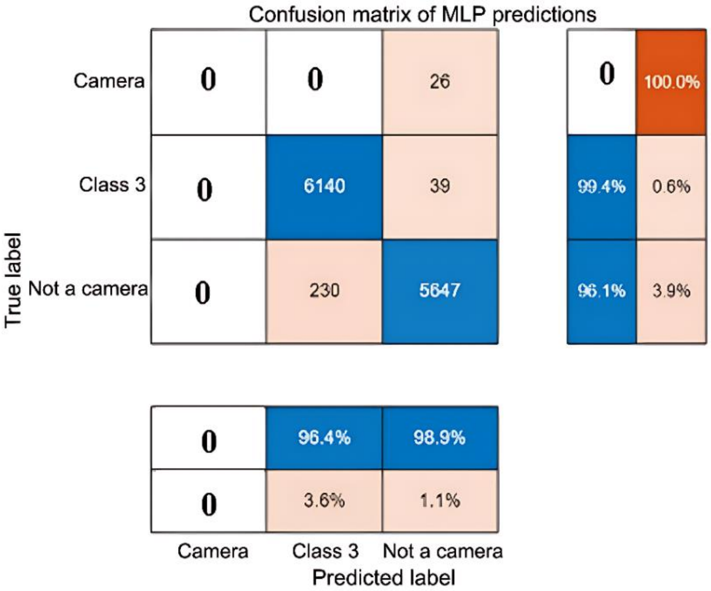


Figure 11. Confusion matrix of MLP.

Table 4. Comparison table.

Method	Dataset	Accuracy [%]	Loss [%]
SVM [37]	40,613	95.2	4.8
MLP [37]	40,613	97.65	2.35
SVM [proposed]	60,412	97.45	2.55
MLP [proposed]	60,412	98.27	1.73

The ensemble neural network outperformed the SVM classifier in all key metrics, demonstrating superior performance in identifying video packets and localizing hidden cameras. This confirms the robustness of the ensemble approach in handling network-based video stream classification.

5. CONCLUSION AND FUTURE SCOPE

The AHCL has successfully demonstrated its efficacy in detecting and locating hidden cameras through the systematic processing of raw IoT network traffic. The convolutional neural network (CNN)-based deep learning model adopted by the system has been found to be highly effective in detection accuracy, establishing such a tool as a capable privacy protection mechanism. The resilience of the model under varying signal-to-noise ratio (SNR) conditions further reinforces this model's potential for deployment in real-world environments. This ensures that the system maintains a consistently high level of performance across diverse network environments, making this solution a credible and viable approach to ensuring personal privacy and security against hidden surveillance threats.

The future development of the AHCL System includes a focus on the robustness of AI allocations, real-time operational capacity, adaptability of AI models, and accessibility for users. It is expected that there will be many enhancements, improvements, and expansions, with the following goals: first, we will expand sources of data by including a variety of video surveillance devices, environmental conditions, and different operational conditions to enhance generalizability; second, we will prioritize hidden camera detection in real-time to enable an immediate response in both consumer surveillance and enterprise security; third, we will aim to adapt AI models within the context of the rapidly changing landscape of IoTs in the surveillance area; and finally, we hope to integrate apps, which are more likely to be accessible and portable solutions for end-users in hidden camera detection in their everyday life. Even though our results appear encouraging, this study is still limited in several ways. First, the evaluation of the system used datasets sourced in controlled settings, and as such, these specific datasets may not adequately reflect the nuances of highly congested or encrypted networks in the real world. Second, the AHCL framework showed the ability to adapt to different variations in the described setup, but the system still needs to be tested in a large-scale deployment environment with heterogeneous IoT ecosystems. Lastly, deep learning models are computationally intensive, which may hinder deployment to low-end edge devices and therefore will require optimization. Collecting additional datasets, large-scale deployments in the field, and power-efficient models will be imperative to improve the practical applicability of the developed system while addressing the aforementioned limitations.

Funding: This study received no specific financial support.
Institutional Review Board Statement: Not applicable.
Transparency: The authors state that the manuscript is honest, truthful, and transparent, that no key aspects of the investigation have been omitted, and that any differences from the study as planned have been clarified. This study followed all writing ethics.
Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] S. Sami, S. R. X. Tan, B. Sun, and J. Han, "LAPD: Hidden spy camera detection using smartphone time-of-flight sensors," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, Coimbra, Portugal. https://doi.org/10.1145/3485730.3485941, 2021, pp. 288-301.

[2] J. I. Olszewska, "Tracking the invisible man: Hidden-object detection for complex visual scene understanding," in *Proceedings of the 8th International Conference on Agents and Artificial Intelligence - ICAART*, Rome, Italy. https://doi.org/10.5220/0005852302230229, 2016, vol. 2: SciTePress, pp. 223-229.

[3] D. Dao, M. Salman, and Y. Noh, "DeepDeSpy: A deep learning-based wireless spy camera detection system," *IEEE Access*, vol. 9, pp. 145486-145497, 2021. https://doi.org/10.1109/ACCESS.2021.3121254

[4] Y. Gu, J. Chen, C. Wu, K. He, Z. Zhao, and R. Du, "LocCams: An efficient and robust approach for detecting and localizing hidden wireless cameras via commodity devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 7, no. 4, pp. 1-24, 2024. https://doi.org/10.1145/3631432

[5] T. Liu, Z. Liu, J. Huang, R. Tan, and Z. Tan, "Detecting wireless spy cameras via stimulating and probing," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, Munich, Germany. https://doi.org/10.1145/3210240.3210332, 2018, pp. 243-255.

[6] J. Heo *et al.*, "Are there wireless hidden cameras spying on me?," in *Proceedings of the 38th Annual Computer Security Applications Conference*, Austin, TX, USA. https://doi.org/10.1145/3564625.3564632, 2022, pp. 714-726.

[7] S. Herodotou and F. Hao, "Spying on the spy: Security analysis of hidden cameras," presented at the International Conference on Network and System Security. https://doi.org/10.1007/978-3-031-39828-5_19, 2023, pp 345–362.

[8] H. An, W. Park, and S. Park, "Real-time sensing and on-site spotting scheme of multi-type WLAN spycams," *IEEE Access*, vol. 12, pp. 153965-153979, 2024. https://doi.org/10.1109/ACCESS.2024.3482429

[9] H. An, W. Park, and S. Park, "Wireless spy camera spotter system with real-time traffic similarity analysis and WiFi signal tracing," *IEEE Access*, vol. 12, pp. 4459-4470, 2024. https://doi.org/10.1109/ACCESS.2024.3350175

[10] S. K. R. Subhashini, S. Gowri, and J. S. Vimali, "A women's safety portable hidden camera detector and jammer," in *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India. https://doi.org/10.1109/CESYS.2018.8724066, 2018: IEEE, pp. 1187-1189.

[11] A. B. Amjoud and M. Amrouch, "Object detection using deep learning, CNNs and vision transformers: A review," *IEEE Access*, vol. 11, pp. 35479-35516, 2023. https://doi.org/10.1109/ACCESS.2023.3266093

[12] Z. Yu, Z. Li, Y. Chang, S. Fong, J. Liu, and N. Zhang, "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. https://doi.org/10.1145/3548606.3560669, 2022, pp. 3107-3120.

[13] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243-9275, 2023. https://doi.org/10.1007/s11042-022-13644-y

[14] A. Pravin, T. P. Jacob, K. M. Prasad, T. Judgi, and N. Srinivasan, "Efficient framework for hidden camera detection & jamming using IoT," presented at the 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India. https://doi.org/10.1109/ICOEI53556.2022.9776943, 2022, pp. 634-637.

[15] P. B. L. Meijer, C. Leistner, and A. Martiniere, "Multiple view camera calibration for localization," presented at the 2007 First ACM/IEEE International Conference on Distributed Smart Cameras, Vienna, Austria. https://doi.org/10.1109/ICDSC.2007.4357528, 2007, pp. 228-234.

[16] R. H. Serag, M. S. Abdalzaher, H. A. E. A. Elsayed, and M. Sobh, "Software-defined network traffic classification for QoS optimization using machine learning," *Journal of Network and Systems Management*, vol. 33, no. 2, p. 41, 2025. https://doi.org/10.1007/s10922-025-09911-6

[17] A. Fathima and G. S. Devi, "Enhancing university network management and security: A real-time monitoring, visualization & cyber attack detection approach using Paessler PRTG and Sophos Firewall," *International Journal of System Assurance Engineering and Management*, pp. 1-17, 2024. https://doi.org/10.1007/s13198-024-02448-y

[18] A. Tekade, T. Vijayan, B. Karthik, and A. Mahajan, "Person identification using novel local triangular binary pattern-based texture descriptor," *EURASIP Journal on Advances in Signal Processing*, vol. 2025, no. 1, p. 7, 2025. https://doi.org/10.1186/s13634-025-01213-y

[19] R. Cunningham and W. L. Tan, "Detection and localization of hidden Wi-Fi cameras," presented at the 2022 27th Asia Pacific Conference on Communications (APCC), 2022.

[20] W. Hua and Q. Chen, "A survey of small object detection based on deep learning in aerial images," *Artificial Intelligence Review*, vol. 58, no. 6, p. 162, 2025. https://doi.org/10.1007/s10462-025-11150-9

[21] W. Cai, J. Wang, Y. Li, H. Zhang, and X. Luo, "Spying on the spy: Security analysis of hidden cameras," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023, pp. 3241–3255.

[22] T. Shinde, R. Bhombe, N. Jadhav, H. Tiwar, and N. Inamdar, "Anti-spy tool: Detection of hidden cameras," *International Journal of Progressive Research in Engineering Management and Science*, vol. 2, no. 5, pp. 1–5, 2022.

[23] B. Lampe and W. Meng, "A survey of deep learning-based intrusion detection in automotive applications," *Expert Systems with Applications*, vol. 221, p. 119771, 2023. https://doi.org/10.1016/j.eswa.2023.119771

[24] P. Negre, R. S. Alonso, A. González-Briones, J. Prieto, and S. Rodríguez-González, "Literature review of deep-learning-based detection of violence in video surveillance," *Sensors*, vol. 24, no. 12, p. 4016, 2024. https://doi.org/10.3390/s24124016

[25] H.-T. Duong, V.-T. Le, and V. T. Hoang, "Deep learning-based anomaly detection in video surveillance: A survey," *Sensors*, vol. 23, no. 11, p. 5024, 2023. https://doi.org/10.3390/s23115024

[26] K. Qian, Y. Bi, Y. Zhao, J. Hou, and X. Liu, "Camera model recognition with deep learning," *Frontiers in Neuroscience*, vol. 12, p. 400, 2018.

[27] R. Chin, W. Chen, and J. Liu, "Detecting wireless spy cameras via stimulating and probing " in *Proceedings of the 2017 ACM Conference on Computer and Communications Security (CCS), November 2017*, 2017, pp. 385–396.

[28] B. Zhou, Y. Liu, and L. Chen, "CNN-LSTM for RF-based camera detection," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1567–1580, 2019.

[29] N. Sharma and B. Arora, "Review of machine learning techniques for network traffic classification," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*. https://doi.org/10.2139/ssrn.3747605, 2021.

[30] J. Smith, R. Brown, and K. Davis, "Thermal imaging for hidden camera localization," *IEEE Sensors Journal*, vol. 18, no. 20, pp. 8456–8468, 2018.

[31] X. Wang and Y. Zhang, "Deep learning on network traffic prediction: Rea cent advances and future directions," in *Proceedings of the 2023 ACM Conference on Networking and Communications (INFOCOM)*, 2023, pp. 1–13.

[32] N. Zhao, B. Liang, and X. He, "A survey on behaviour recognition using WiFi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98-104, 2017.

[33] M. Salman, N. Dao, U. Lee, and Y. Noh, "CSI: Despy: Enabling effortless spy camera detection via passive sensing of user activities and bitrate variations," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*. https://doi.org/10.1145/3534593, 2022, vol. 6, no. 2, pp. 1-27.

[34] Z. Shi, H. Wu, J. Zhang, M. Zhang, and W. Huang, "FindSpy: A wireless camera detection system based on pre-trained transformers," in *2023 IEEE Symposium on Computers and Communications (ISCC)*, Gammarth, Tunisia. https://doi.org/10.1109/ISCC58397.2023.10218088, 2023, pp. 816-822.

[35] Y. Liu, H. Zhang, and J. Li, "WiSOM: WiFi-enabled self-adaptive system for monitoring the occupancy in smart buildings," *Energy*, vol. 294, no. 5, p. 130420, 2024.

[36] K. Wu and B. Lagesse, "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)*, Kyoto, Japan. https://doi.org/10.1109/PERCOM.2019.8767411, 2019, pp. 1-10.

[37] J. Lee, S. Seo, T. Yang, and S. Park, "AI-aided hidden camera detection and localization based on raw IoT network traffic," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, Edmonton, AB, Canada. https://doi.org/10.1109/LCN53696.2022.9843203, 2022, pp. 315-318.

Views and opinions expressed in this article are the views and opinions of the author(s), Journal of Asian Scientific Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.