electronics

MDPI

Review

A Survey of Video Surveillance Systems in Smart City

Yanjinlkham Myagmar-Ochir and Wooseong Kim *

Computer Engineering Department, Gachon University, Seongnam 1342, Republic of Korea; kyg971@gachon.ac.kr
* Correspondence: wooseong@gachon.ac.kr

Abstract: Smart cities are being developed worldwide with the use of technology to improve the quality of life of citizens and enhance their safety. Video surveillance is a key component of smart city infrastructure, as it involves the installation of cameras at strategic locations throughout the city for monitoring public spaces and providing real-time surveillance footage to law enforcement and other city representatives. Video surveillance systems have evolved rapidly in recent years, and are now integrated with advanced technologies like deep learning, blockchain, edge computing, and cloud computing. This study provides a comprehensive overview of video surveillance systems in smart cities, as well as the functions and challenges of those systems. The aim of this paper is to highlight the importance of video surveillance systems in smart cities and to provide insights into how they could be used to enhance safety, security, and the overall quality of life for citizens.

Keywords: video surveillance system; smart city; video analysis

Citation: Myagmar-Ochir, Y.; Kim, W. A Survey of Video Surveillance Systems in Smart City. *Electronics* **2023**, *12*, 3567. https://doi.org/10.3390/electronics12173567

Academic Editor: Juan-Carlos Cano

Received: 30 June 2023
Revised: 14 August 2023
Accepted: 21 August 2023
Published: 23 August 2023

1. Introduction

Cities the world over are becoming smart with the integration of advanced information technologies and data-driven solutions. A smart city, such as an urban area equipped with a well-developed infrastructure, improves the quality of life of its citizens, enhances sustainability, and optimizes urban services such as transportation, energy distribution, communications, and public safety. One of the key components of a smart city is a video surveillance system (VSS), which enables the detection and identification of various situations that are relevant to smart city applications, including public safety, crime prevention, traffic management, and environmental monitoring.

The modern city suffers from high population density, which causes various problems for urban living. For instance, roads that carry too many vehicles are always congested, and this may cause accidents between vehicles and pedestrians. At the same time, air and water pollution from vehicles and factories threatens the health of citizens. However, receiving healthcare is challenging as the number of clinics and hospitals is limited relative to the population. For healthcare issues, telemedicine has been adopted in many countries and the recent COVID-19 pandemic has accelerated its adoption. Furthermore, crimes and fires frequently occur and require a rapid response to prevent damage, which eventually increases the social cost required to maintain more police and fire stations. These modern city problems could be solved effectively by a VSS.

Recently, some of the published VSS survey papers focused on one specific area such as attacks and preventive measures on VSS [1], anomaly detection [2–5], crowd behavior analysis [6], and drone surveillance system [7]. Some survey papers highlight the benefits of using specific datasets in VSSs [8,9]. In contrast, our survey proposes a VSS architecture and investigates system components and recent technologies required for surveillance tasks in urban areas.

Table 1 describes possible use cases for the VSS to solve the aforementioned problems such as in the areas of healthcare, traffic management, public safety, environment monitoring, and crowd management. Efforts have previously been made to develop VSS solutions for these problems. Figure 1 shows the representation of topics in the literature

that are related to the categories in Table 1. Studies that are oriented toward public safety initiatives like crime prevention and detection, public safety analytics, and the surveillance of restricted areas represent the largest portion. Other studies related to environmental monitoring, healthcare, crowd management, and traffic management have also been carried out.

Table 1. Smart city video surveillance applications.

Healthcare	• Elderly care and telemedicine • Disaster and pandemic control
Traffic management	• Car accident and congestion detection • Parking lot management • Optimization of traffic flow
Public safety	• Public space and crime monitoring • Building and greenspace monitoring • Crowd monitoring
Environmental monitoring	• Air quality and weather monitoring • Fire and smoke detection

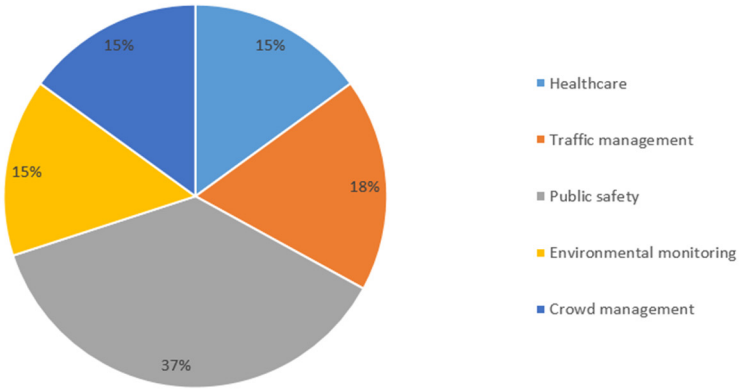

Figure 1. Literature related to video surveillance systems for smart cities.

Traditional VSSs relied on human supervision for monitoring and analyzing video feeds in real time, and often encountered problems in processing large amounts of data. Law enforcement and other authorized departments would install cameras and other visual equipment in areas of the city, such as public parks, transportation hubs, and high crime areas, to monitor and respond to unusual situations [10].

In contrast, recent VSSs have evolved with advanced technologies, called intelligent surveillance systems [11], which automatically record and save video data in a secure blockchain, and then analyze and interpret the video data using computer vision techniques based on deep learning performed by edge or cloud computing. The intelligent VSS aims to detect and track objects, recognize faces, identify anomalies, and predict potential incidents or emergencies. It also sends alarms for proactive monitoring, rapid response, and efficient resource allocation by city authorities [6]. Surveillance system applications for smart cities require functions such as object detection and classification, object tracking, human action recognition (HAR), anomaly detection, and video storage management:

- **Object Detection and Classification**: Object detection and classification techniques, that employ traditional computer vision and machine learning in edge or cloud computing, are utilized to identify various types of objects in a city scene. These objects may include people, vehicles, street plants, animals, and environmental factors. Different technologies and algorithms may be applicable, depending on the characteristics

of the object, such as shape, color, and movement, to accurately detect and classify objects in video frames.

- **Object Tracking:** Object tracking is the process of continuously following and monitoring objects as they move across consecutive frames of a video. Valuable information such as trajectory, speed, and interactions with other objects or individuals can be obtained by tracking moving objects; for example, a hit-and-run vehicle can be tracked using a video stream from roadside closed-circuit televisions (CCTVs). Deep learning methods are often employed for object tracking and use tracking based on a single point of the object, tracking based on shape changes, or kernel-based approaches.
- **HAR:** HAR plays a significant role in healthcare and crowd monitoring applications, and it focuses on identifying and responding to potentially harmful actions or emergencies. HAR systems analyze and understand human actions rather than focusing solely on characteristics like movement, body shape, or skin color. Therefore, deep learning has recently been used to extract meaningful features and recognize human actions from the visual information available in video sequences.
- **Anomaly Detection:** Anomaly detection is important for real-time monitoring and the identification of unusual or suspicious activities. It makes proactive measures possible and prompts responses to potential threats or incidents in a smart city. Anomaly detection methods can be applied to many tasks, such as road and traffic anomaly detection, concealed weapon detection, crowd surveillance, and the detection of suspicious activity. These methods often utilize machine learning or deep learning techniques to detect behavioral patterns that deviate significantly from the expected behavior.
- **Video Storage Management:** For a VSS, ensuring the integrity, security, and accessibility of video data is critical. Therefore, VSSs use the blockchain to ensure data integrity, security, and controlled access by authorized personnel while efficiently storing and managing large volumes of video data using distributed storage management.

In this survey, we present an overview of a general VSS architecture with key components for smart city applications, and we discuss their challenges based on studies published in 2018 or more recently. State-of-the-art techniques that are applicable to the VSSs are discussed in current research and development. Despite previous efforts, the VSS still faces several challenges that include system accuracy, scalability, real-time processing, variability, privacy, data storage, and retrieval problems that can impact their effectiveness and implementation. For instance, a high false alarm rate incurs a significant administrative workload, which requires a new approach like multi-modal features and unsupervised learning methods for better detection accuracy. Additionally, drones are being used as surveillance equipment, and they may be useful in future surveillance systems with better accessibility. However, further study on drones is needed, considering the drones' limited resources and privacy issues. Furthermore, scalability in video storage and management is critical, as a VSS generates a huge amount of video data in real time.

This paper is organized as follows: Sections 2 and 3 presents an overview and a description of the VSS for a smart city. Then, Section 4 describes the features of the VSS with applicable technologies from previous studies. In Section 5, we present the remaining challenges for future works, and we conclude our study in Section 6.

2. Video Surveillance Systems in Smart Cities

As the VSS using cameras to monitor activity in a specific urban area is incorporated with many smart city applications, the VSS contributes to improving public safety, healthcare, traffic management, urban management, and efficiency:

- **Healthcare:** Healthcare organizations use the VSS for emergency medical care, remote patient monitoring, and quarantine monitoring. Depending on a particular state of a patient detected by the camera and analyzed by deep learning algorithms such as walking, falling down, or being motionless, medical care may be given to the patient immediately. For this, video data are recorded and sent to a nearby edge node or to a cloud server. Deep learning methods like convolutional neural network (CNN)

or deep neural network (DNN) decide whether the patient status requires the help of a healthcare center. Additionally, the VSS is a valuable tool to ensure compliance with quarantine guidelines and to monitor potential risks to public health during an outbreak of an infectious disease like COVID-19. For instance, health authorities can realize public quarantine to prevent the spread of the disease using the VSS with cameras placed outside the houses of quarantined individuals.

- **Traffic management:** Traffic management involves monitoring traffic accidents and rule violations on the road, while calculating and analyzing traffic jams using the VSS. Typically, cameras placed on roads and major intersections monitor traffic conditions and provide real-time video feeds, which can pre-process the video data using background extraction and region of interest (ROI) algorithms for real-time traffic control. Various algorithms analyze the pre-processed video data for its purpose, such as computer vision and deep learning techniques. For example, supervised deep learning methods such as CNN, mask R-CNN (MRCNN), and deep CNN (DCNN) monitor common accidents and identify similar patterns during accident monitoring. Additionally, unsupervised deep learning methods like incremental spatiotemporal learner (ISTL) discover new types of accidents to broaden the scale of the system. Furthermore, motion-detection methods based on You Only Look Once (YOLO) and CNN can predict the future movement of cars in motion in particular scenarios.
- **Public safety:** Video cameras placed in various public areas, such as streets, parks, transportation hubs, and commercial districts, enable the continuous monitoring of citizen activities for public safety, which particularly identify criminal activities such as theft, vandalism, and public disturbances, as well as detect suspicious movements among crowds. For this, real-time processing of recorded footage occurs on a nearby edge node or cloud server, where motion-based methods, like frame differentiating, optical flow, and deep learning algorithms, are mostly applied for human action detection. In addition, deep learning methods such as long short-term memory (LSTM), CNN, recurrent neural network (RNN), and DNN can be employed for encoder and classifier tasks, enabling the identification and categorization of prohibited human movements in an environment. Controlling cold weapons in public environments is essential to public safety. For hand-held cold weapons, they look similar to mobile phones, wallets, and cards, so deep-learning-based fine-grained algorithms are recently promising. In addition, the deployment of green plants and buildings in the urban area is primarily developed by color-based, shape-based, and texture-based computer vision methods. For instance, color-based classification methods, such as support vector machine (SVM) and k-nearest neighbor (kNN), are used to monitor plant diseases, and texture-based methods, such as Gabor filtering and local binary patterns (LBP) histograms, are used for the analysis of buildings in the city.
- **Environmental monitoring:** Air pollution and weather condition monitoring in urban areas rely on color-based computer vision methods to analyze live video feeds and detect visual cues related to air quality and weather patterns. For example, the VSS can identify the presence of smog or haze, which may appear as a discolored or hazy layer in the atmosphere. By focusing on specific color ranges indicative of air pollutants, real-time alerts can be provided when pollution levels exceed certain thresholds. Additionally, the system can detect rain, snow, fog, or other weather phenomena, providing valuable data for weather monitoring and forecasting. For early fire detection, it is important to detect even low-level flames as quickly as possible. Surveillance cameras are strategically placed in and around potential fire-prone zones to continuously monitor for signs of fire. Then real-time image processing techniques are applied to analyze the video frames and identify potential fire-related patterns. A combination of color-based methods, such as YOLO, CNN, and SVM classifiers, along with shape-based methods, like generative adversarial network (GAN) discriminator and DNN, can be employed to distinguish fire areas and non-fire areas. By leveraging

both color and shape information, the VSS can minimize false alarms and improve the accuracy of fire detection.

3. Video Surveillance System

A VSS in a smart city is designed to monitor the urban environment using various devices such as public or private CCTVs, the dash cams of cars or unmanned aerial vehicles (UAVs), and even smartphone cameras. Figure 2 presents a VSS architecture, which consists of end camera devices, an edge computing system that first processes video data and then analyzes the data for a fast response to users, a cloud computing system with enough computing resources and storage to enable users to analyze the video data intensively using deep learning, and a blockchain system that provides secure storage and consensus between anonymous users on the video data.

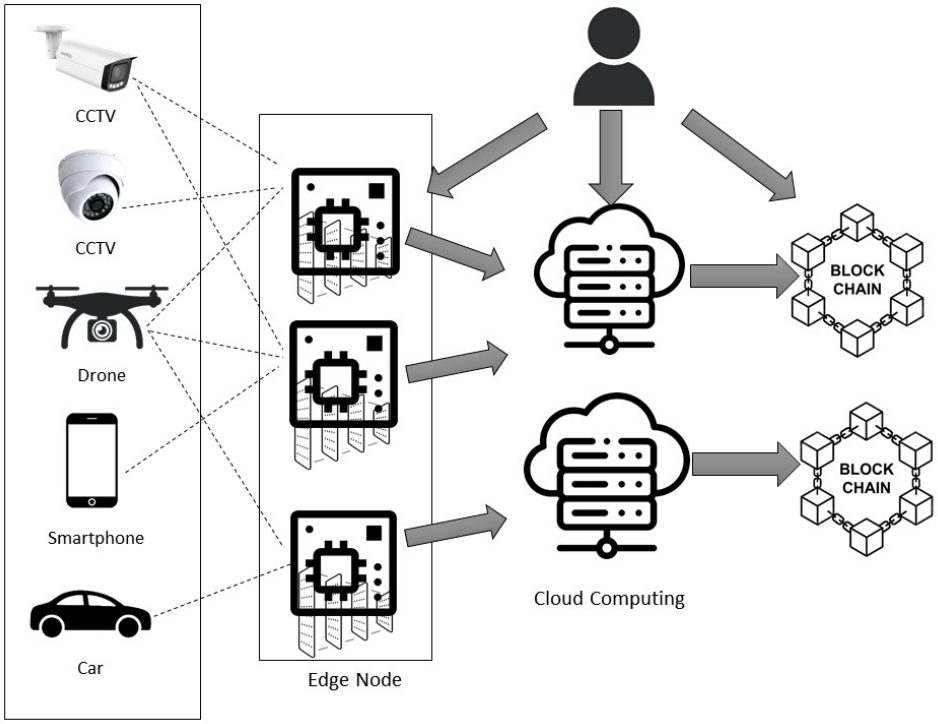

Figure 2. Overview of video surveillance system.

The technologies and devices that are related to the VSS vary according to its purpose and scale. For example, edge computing is widely used in scenarios where real-time transmission and immediate action are critical, such as in hospitals, and for environmental monitoring. In a traffic monitoring system, the video captured by roadside cameras is analyzed using deep learning techniques. Additionally, for early fire detection, a color-based deep learning classification method is utilized to determine if the frames of the incoming videos contain red color and to serve as an initial stage of fire detection.

3.1. Monitoring Device

As cameras have become common in many areas of cities, visual data can easily be collected from the surroundings for smart city applications. Monitoring devices are classified into two types: moving and fixed monitoring devices. Fixed monitoring devices are typically located inside and outside of buildings, on streets, and at road intersections to continuously monitor designated areas. In contrast, moving monitoring devices are designed to move freely and monitor areas that are not directly visible. Figure 3 categorizes monitoring devices for the smart city VSS with respect to their mobility; some cameras, like CCTVs, are static, whereas others, such as those on vehicles and mobile phones, are moving.

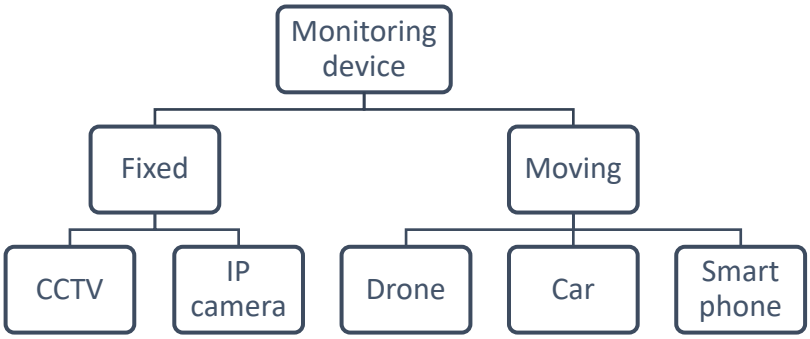

Figure 3. Monitoring device classification.

Some examples of fixed monitoring devices would include pan-tilt-zoom (PTZ) cameras, Internet Protocol (IP) cameras, dome CCTVs, wireless CCTVs, and bullet CCTVs [12]. IP cameras, in particular, are becoming popular because they not only capture video footage, but also transmit and receive data over a computer network or the Internet, which makes it easier to access footage from remote locations. Due to their excellent scalability, IP cameras can be used to cover wide urban areas [13].

Moving monitoring devices, like UAVs, mobile phones, and cars, incorporate location information from GPS or other tracking algorithms [14]. UAVs, which are mainly drones, can collect complete and consistent data, unlike the more limited fixed monitoring devices. Drones can move freely in the sky, cover large and hard-to-reach areas, and are easily deployed and retrieved with the click of a button. Drones provide an aerial view; their real-time data and mobility make them valuable tools for enhancing security and improving various aspects of urban management, which are used popularly for purposes such as object and people detection, data collection, general traffic and wildlife monitoring, radio surveillance, and search and rescue operations during disasters [15–18]. For object and people detection, drones equipped with high-resolution cameras and AI-based image recognition algorithms can detect and track objects, vehicles, and individuals in real time. Drones in the environment monitor air quality, pollution levels, and changes in the environment over time. Additionally, they are used for the quick and efficient delivery of emergency medical supplies, packages, or small goods in congested urban areas.

Crowd-sourcing based on smartphone cameras is not common, but it is a very innovative approach to monitoring the smart city. In [19], smartphones were used for a video data collector to record urban incidents such as traffic updates, accidents, thefts, and street disturbances using citizens' smartphone cameras. Smartphone users could store recorded video data on a cloud server via a dedicated application or web portal. Other users of the system could then access the shared information individually or receive an alarm about incidents from the cloud.

3.2. Edge Computing and Pre-Processing

Edge computing is gaining popularity in VSSs, as it provides an efficient and effective way to process video data. In traditional VSSs, video data is typically sent to a central server or a cloud for processing, analysis, and storage. Sending all the data to a central server can be slow and costly, as it requires a high-speed network connection, large storage capacity, and powerful computing resources. However, edge computing involves processing and analyzing video data closer to where the data are generated, such as on local camera devices, which may include smartphones and network entities like base stations and wireless access points [20]. Edge computing is particularly powerful for real-time data-driven applications; for example, deep learning is used on edge nodes to detect system failures, and to regulate traffic jams based on current road conditions. Therefore, data can be stored and processed locally, which eliminates the need to upload all the data to the central server. This reduction in network burden enhances the efficiency of network bandwidth utilization [21].

Edge computing offers several advantages that make it well-suited for use in public safety organizations such as hospitals and police departments. For instance, when a child goes missing, local edge devices can be used to quickly retrieve and examine data recorded by surveillance cameras within each region; this can be an effective strategy for law enforcement agencies [22]. In addition, during an ambulance ride, edge computing may involve using onboard medical devices and sensors to monitor a patient's vital signs, collect data about their current condition, and transmit that data to the nearest hospital in real time. This gives healthcare providers at the receiving facility access to critical information about the patient's condition before the patient arrives [23].

In smart city surveillance systems, edge nodes are typically positioned between the monitoring devices and the cloud server [24]. Edge computing analyzes data locally and sends only relevant or summarized data, thereby reducing the processing load on the cloud server [25]. Cloud computing offers scalability by providing virtually unlimited computational resources and robust storage capability. It is suitable for handling large volumes of data, computationally intensive tasks, or scenarios where edge computing resources may be limited [26]. Moreover, cloud computing allows for the centralized management and control of the entire smart city surveillance system. If an edge node experiences a failure or disruption, the cloud server can seamlessly take over the processing tasks, which ensures continuous operation and minimizes downtime [27].

Pre-processing is an essential step in video processing, as it optimizes video data before they are analyzed by computer vision algorithms. Due to the significant overhead of directly processing raw data using algorithms, it is often necessary to pre-process the data before the main processing step. The goal of pre-processing is mainly to remove noise and irrelevant information from the video stream and enhance the relevant information to improve the accuracy of the computer vision algorithms. Pre-processing involves various operations that have often been employed in previous studies, such as ROI segmentation, video compression, noise reduction, image resizing, and reformatting. Specifying an ROI retains the quality of the most pertinent information in a transmission and reduces the amount of data transmitted, which increases the efficiency and bandwidth of a network utility [28–32]. Video compression involves reducing the size of video data to make it easier to store and transmit without sacrificing too much quality, which optimizes the use of bandwidth and reduces the cost of data storage and the transmission time [29,33]. Image resizing and reformatting effectively improves the quality of a video image by adjusting the brightness, contrast, and color balance. This method is especially useful in low-light environments where video quality may be poor. Furthermore, noise caused by low illuminance or signal interference can be reduced by removing random fluctuations in the video signal [34,35].

3.3. Cloud Computing and Deep Learning

Cloud computing plays an important role in VSSs by providing a scalable, flexible, and cost-effective way to store, process, and analyze video data. The storage of video footage can be offloaded to cloud-based storage solutions; this is one way that cloud computing is used in video surveillance. This eliminates the need for an on-premises storage infrastructure and allows for easy access to footage from any location that has an Internet connection. Moreover, cloud-based storage provides high availability and redundancy, which ensures that the data are always accessible and protected from data loss [31]. Although cloud computing is beneficial for efficient data transfer, protection, processing, and storage, it still requires a large network bandwidth and may suffer from long response times due to network latency. Therefore, edge computing is essential for providing solutions that are feasible for real-time intelligent video surveillance. Edge computing processes data at the edge of a network in advance. The processing is closer to the data source, which reduces latency and workload on the central cloud for faster response times [36,37].

Deep learning has revolutionized video surveillance in recent years. With the help of advanced deep learning algorithms, it is now possible to analyze and understand large amounts of video data in real time, which enables improved safety and security measures [38]. Deep learning algorithms can accurately detect and track objects in video streams, even in complex and cluttered environments, and can detect and classify different types of objects, such as people, vehicles, and animals. Deep learning can be performed by various types of neural networks such as CNNs [39–42], DNNs [43,44], and RNNs [45,46], to recognize objects, track movements, and analyze behavior in video data [47]. To train on video data in a supervised learning task, it is necessary to pre-process and annotate the data. The annotated data are then fed into a neural network and the network parameters are adjusted to optimize the performance on the given task. Afterward, the trained model is deployed on the edge or in a central cloud for VSS [48].

The CNN is the network most widely used for VSSs. It is used to grade objects for classification, identify actions in video frames, perform similarity matching, and calculate image characteristics. A CNN consists of multiple layers of convolutions and pooling operations that enable it to learn and identify features in images, such as edges and shapes. By stacking these operations in multiple layers, CNNs can learn increasingly complex representations of images, which enables them to identify and classify objects within the images [28–30,32]. Meanwhile, DNNs are used for a wide range of tasks, including natural language processing (NLP), and image and speech recognition. The main purpose of DNNs is to understand intricate patterns in data by constructing deep hierarchical representations of the information. DNN models are trained using a backpropagation technique that adjusts the network weights based on the errors the network makes when making predictions. This iterative process allows the DNN to gradually refine its predictions and improve its overall performance [49,50]. RNNs are used for sequential data analysis, such as NLP and speech recognition.

3.4. Blockchain

Since Bitcoin was introduced in 2008, researchers have been developing blockchain technologies and demonstrating their usefulness for various industries. The blockchain is also used for video surveillance in smart cities in contexts where IoT sensors, actuators, and surveillance devices collect and process data. The collected data and the deep learning models are made available through servers to users/stakeholders [51]. Large amounts of sensitive data can cause bottlenecks, single points of failure, and cybersecurity issues on a network. Blockchain technology is often considered to solve these concerns due to its decentralized, secure, immutable, persistent, and fault-tolerant characteristics [52–54]. To prevent data tampering by unauthorized access and the loss of important records, permissioned or private blockchain systems are often preferred for smart cities. For the VSS, the blockchain can provide scalable storage and grant users restricted access to data without being influenced by external factors such as cybersecurity threats, data privacy concerns, and system downtime. To ensure data integrity, provide authentication/authorization, and enable cryptocurrency transactions [55], the key features of blockchains are:

- **Authorization and authentication:** This characteristic ensures that data are exchanged only between authorized devices within the VSS for the security, integrity, and reliability of blockchain networks. By verifying the identity of devices or users with the use of membership information, authentication prevents unauthorized access to sensitive data. This process typically involves verifying credentials or digital signatures to ensure the legitimacy of participants within the blockchain network [56–58]. Furthermore, authentication and authorization are also used in data sharing and system integration between separate VSSs. By registering on the same blockchain, users of distinct VSSs can communicate securely by using an authentication process that allows only authorized devices to access shared data, which further enhances the security and integrity of the video surveillance ecosystem [59].

- **Data integrity:** This characteristic ensures that the video surveillance data stored on the blockchain are accurate, complete, and unaltered. This can be carried out using various techniques, such as digital signatures, cryptographic hashing, data encryption, and tamper-evident seals. These methods detect unauthorized changes made to the data and prevent any malicious attempts to alter the video footage [60,61].
- **Distributed storage:** This characteristic is based on peer-to-peer (P2P) networking, and it is used for storage like the InterPlanetary File System (IPFS). Distributed storages need to be specified for the process of accessing and cross-referencing data from multiple sources and integrating it into a single system. This makes data easier to store and distribute and enables stakeholders to access and share it seamlessly [62]. In other words, video data can be shared directly between devices without a centralized server through P2P distributed storage. This improves efficiency and reduces the risk of data loss or tampering. P2P distributed storage is used in various applications, including file sharing, content distribution, and communication platforms [63].
- **Smart contracts:** The smart contract is a blockchain feature that has been utilized to automate secure operations without user intervention. It increases transparency, efficiency, and flexibility in various distributed applications. Smart contracts are self-executing programs that are executed by a third party; their result is inspected by other blockchain nodes and finally written into a block. This blockchain feature enables various secure and reliable operations without having a trusted party in the distributed system [64]. Smart contracts can be used to manage assets, budgets, and even traffic congestion. For instance, smart contracts can increase transparency and ensure the proper use of a city's budget, by automatically allocating funds for road maintenance, by taking traffic patterns and road usage as inputs in a function of the smart contract. Accordingly, city resources are scheduled efficiently for the city's infrastructure [65].

4. Features of Video Surveillance System in Smart Cities

This section discusses the key functionalities of a VSS for monitoring a city environment using cameras, edge/cloud computing, and deep learning algorithms. These functions mainly include surveillance video analysis such as HAR, object detection, anomaly detection, object classification, object tracking, and the secure video data management.

4.1. Object Classification and Recognition

Object classification plays a vital role in VSSs as it helps identify and localize specific objects within a video stream. After objects have been detected in a video stream, object classification is used to identify and categorize them into predefined classes that include people, vehicles, plants, animals, and environmental factors in a city area. By classifying objects into different categories, we gather the necessary information required for further analysis and processing [66].

The object classification process consists of several steps, such as image acquisition, image pre-processing, feature extraction, and object recognition. The image pre-processing step utilizes video compression, noise reduction, and image enhancement algorithms. Feature extraction is carried out to identify patterns in the image, and object recognition involves comparing the features that have been extracted to a database of known objects to identify the object of interest. If feature extraction is done before object recognition, the recognition can be realized by several factors, such as motion, shape, color, and texture, as described in the following sub-sections. A summary of sub-sections is described in Table 2.

4.1.1. Motion-Based Object Detection

Motion-based object detection is a technique commonly used by VSSs to detect and track objects such as pedestrians, vehicles, and animals based on their motion characteristics. Motion-based object detection involves analyzing the changes in motion within a video sequence to identify regions or objects of interest. Various motion detection algorithms can be used, such as frame differencing, background subtraction, or optical flow. These

algorithms are computationally less complex and suitable for a dynamically changing environment [67]. The background subtraction method is useful in scenarios where the background is relatively static but the foreground objects exhibit motion such as street scenes and indoor environments [68]. The frame differencing and optical flow algorithms both compare consecutive frames to detect changes in pixel values, but optical flow algorithms are more suitable for dynamically changing environments, which can process complex motion patterns [69–71]. In [72], the authors introduced the frame differencing and W4 algorithm for detecting moving pedestrians and vehicles in a noisy environment with a complex background. This algorithm calculates the difference between frames using the frame differencing and the W4 algorithm separately, and combines the outcome of each technique using a logical OR operation. Finally, morphological operations with connected component labeling are used to remove the noise and detect the final object in the combined outcome.

In the comparison of the proposed approach with the inter-frame and three-frame methods, it was evaluated across three datasets: one focused on pedestrians in a street (walking), another featuring a commercial street scenario (meetings), and a third centered around a four-road intersection scenario (traffic). The findings revealed that, in various environments, the proposed method exhibited an improvement in accuracy ranging from 2% to 4% when contrasted with the inter-frame and three-frame difference methods. Notably, the highest accuracy was achieved in tracking people during meetings, while the least accurate results were observed in the context of tracking traffic. These variations in accuracy were influenced by factors such as the distance between the camera and the object being tracked, as well as the speed of the object's movement.

Recently, deep-learning-based approaches like CNN and DNN have often been used for motion-based object detection. For example, the CNN finds the location of a moving car in a traffic system, predicts the forward path of the car, and optimizes the crossroad throughput [28,30]. In [73], YOLOv4 was used to detect multiple vehicles in a traffic system that employed a CSPDarknet53 classifier and spatial pyramid pooling to connect to a YOLOv3 head. In order to reduce accidents on the road, a YOLOv4 repository was used to build a custom object detector, which contains the details for all the parameters related to the road. It achieves a high detection accuracy with the precise positioning of a bounding box and fast computation. According to the results of the implementation, it was used to distinguish between the vehicles on the road, and the accuracy was 97% in the test on the road where the density of the car is high.

Some research studies have combined deep learning with other approaches to improve performance. Zahra et al. [29] proposed the ESSE (efficient shallow segmentation-based encoding) framework for video surveillance to identify suspicious people or vehicles and to detect traffic patterns and road conditions. ESSE uses a CNN and a modified high-efficiency video coding (HEVC) encoder together. First, the CNN segments salient regions in video frames captured by a camera. Then, a modified HEVC encoder is used to minimize the video file size while maintaining the high quality of the salient regions. Utilizing a modified HEVC encoder yielded a notable accomplishment, achieving a cross-road object pixel accuracy of 92.31%. In contrast, employing the default HEVC encoder led to a slightly lower object pixel accuracy of 86.78%. The enhancement of object pixels through these encoders offers the advantage of minimizing inaccuracies in object detection and contributing to an overall improvement in system accuracy.

Using the lightweight DNN with edge computing can reduce the computational cost of object detection [49]. Whereas the aforementioned methods usually use one camera to detect a moving object based on the difference between recorded frames, in [74], federated edges were used to position multiple cameras to detect an object from multiple angles. Federated edges in smart city surveillance allow multiple cameras to work together to detect objects from various angles. For example, federated edges can be used to monitor traffic and the license plates of vehicles entering restricted areas. By selecting the most

appropriate camera and prioritizing the highest-rated video feeds, the system saves time and resources, thereby improving efficiency.

Some studies have used a combination of traditional computer vision algorithms [75]. In [76], the adaptive motion estimation segmentation (AMES) and the proposed sequential outline separation (SOS) methods were used to detect multiple moving objects. AMES is a computer vision technique that analyzes the motion information between consecutive frames and then uses that information to separate moving objects from the background. Meanwhile, SOS iteratively analyzes the structure of the outline and separates it into distinct levels based on indentation or other hierarchical indicators. During the evaluation of the system's ability to detect multiple cars on the road, the SOS-based moving object recognition algorithm achieved a high classification accuracy of 97.45%, surpassing the results of traditional block-matching approaches. These conventional techniques are commonly employed in image and video processing for tasks like motion estimation and compression. They function by dividing an image or video frame into smaller blocks, subsequently analyzing and comparing these blocks between successive frames to gauge the extent of motion. In comparison, the accuracy of the block-matching methods in classifying cars on the road reached 96.61%.

In [34], a double-mode surveillance system was proposed. The system checks audio and video simultaneously using a laser Doppler vibrometer and a PTZ camera to detect remote human movements in an indoor environment. The laser Doppler vibrometer acquires remote audio by detecting the vibrations in the area, in which humans are in motion. The PTZ camera is capable of adjusting its position, which enables the dynamic monitoring and tracking of the specific area on which the laser Doppler vibrometer is focused.

4.1.2. Shape-Based Object Detection

Shape-based object detection in video surveillance uses computer vision algorithms to analyze the shapes and contours of objects that are captured in video footage, which enables the detection and classification of specific objects of interest. These objects may include people, vehicles, visually similar objects, and faces [77]. The DeepDC system [78] uses a CNN-based algorithm and a GAN discriminator to classify people and vehicles on a public street using a post-processing operation. In the system's initial stage, a CNN-based algorithm called DeepSphere is used for object segmentation. This means that it separates the different objects (such as people and cars) in the scene. After the objects are segmented, the GAN discriminator extracts deep features from the segmented objects and assigns them to their respective labels. Comparing the test results, we can observe that when utilizing only a CNN for object classification, the accuracy achieved is 89.12%. However, when incorporating a GAN discriminator after the CNN, the accuracy significantly improves to 93.82%. This demonstrates the effectiveness of adding a GAN discriminator in enhancing the classification performance of the model. To differentiate between people and cars on the street, the system considers where they entered and exited the street, as well as what activities they were engaged in while on the street.

Face recognition is one kind of shape-based classification that identifies and verifies individual faces captured by surveillance cameras in a city environment. The method proposed in [79] involves using LBPs to analyze facial features and match them against a database of known individuals or generating unique identifiers for unidentified faces on an edge node and then sending the detected faces to a CNN server for further processing and analysis. The LBPs are used to analyze the local texture patterns of the video frames input to the edge device itself, which helps in detecting the presence and location of faces in the captured data efficiently. By combining LBP-based feature extraction with CNN-based analysis on a server, it is possible to achieve accurate and robust face detection and recognition.

Improving the precision of the machine identification of small and intricate objects is a challenge. This task involves distinguishing subtle visual differences between objects that belong to the same broad category. A deep-learning-based binary classifier involves

distinguishing between two distinct classes or categories. It is particularly useful in detecting small objects that share similar visual features like cards, cashes, bills, pistols, and pocket knives [32]. By training a deep-learning-based binary classifier on labeled data, where one class represents the target object and the other class represents the background, it is possible to detect these small objects effectively. When employing binary classifier algorithms like one-versus-all and one-versus-one in isolation, the objects generated by the classifier are differentiated based on the algorithm's traces. According to the test results, after using the deep learning algorithm like the proposed method, the binary classifier algorithm reduced the number of false detections by 56.5%. This approach for detecting small objects in surveillance videos showed an accuracy of up to 88.5%. Additionally, CNN models like fractional-based CNNs and subtractive CNNs can be utilized to distinguish subordinate-level categories, such as bird species or dog breeds [80]. Fractional-based CNNs upsample or reconstruct feature maps, which is particularly beneficial for capturing intricate details that are crucial for distinguishing between subordinate-level categories. Subtractive CNNs employ a process of subtracting or canceling out common features across categories, which enables them to emphasize the unique characteristics of each category.

The object detection method in [81] uses a CNN with a probabilistic neural network (PNN). A CNN is used to extract high-level features from the images, which are then fed into multiple PNNs. Each PNN produces a binary decision for the image class. The outputs of these binary decisions are then combined to make a final decision. The use of multiple PNNs helps to enforce consistency between decisions and reduce the risk of misclassification due to noise or other sources of uncertainty. Additionally, the use of PNNs can help to reduce the risk of overfitting to the training data until 17%, which improves the generalization performance of the model until 89%.

Although the above-mentioned methods are usually based on single-view object classification, Deng et al. [82] presented an insightful study that utilized a multi-view fusion network and processed data from a three-dimensional perspective, rather than a two-dimensional perspective. Their research focused on the precise identification of animals in specific environments, such as at the beach or in the forest. To achieve higher classification accuracy, the researchers employed a multi-view fusion network that analyzed and integrated information from multiple viewpoints, which enabled a more comprehensive understanding of the data. By considering different perspectives simultaneously, the network was able to capture a broader range of features and characteristics associated with the animals. The suggested three-dimensional method requires more time for data integration compared to the two-dimensional approach. However, the detection accuracy has the potential to rise significantly, reaching up to 98% based on the dataset.

4.1.3. Texture-Based Object Detection

Texture-based object classification identifies and classifies objects based on their textural patterns in a video sequence. It involves analyzing the spatial distribution of textural features in images or video frames to distinguish between different object categories. Manik et al. [83] proposed a plant classification system based on gray level co-occurrence matrix (GLCM) extraction using a kNN classifier in 2019. The GLCM is a technique in digital image processing that is commonly used to extract texture features from images. It calculates the co-occurrence matrix of gray level values in an image, from which four features can be extracted: contrast, correlation, energy, and homogeneity. The features extracted from the GLCM are then used as inputs to a kNN classifier, which is a type of supervised learning algorithm used for classification tasks, that classifies a plant sample based on its similarity to the k nearest plant samples in the training dataset. The effectiveness of this system depends on the quality of the GLCM features and the training data used to train the kNN classifier. During the experiment, the classification of plant types in leaf images relied on GLCM characteristic extraction paired with a kNN classifier. When segmenting the data into clusters with values of k = 3, k = 5, and k = 7, the most noteworthy outcome was

achieved when k = 3, resulting in the highest accuracy of 83%. Conversely, with k = 5, the accuracy dropped to 60%, and for k = 7, it reached 62%.

Uzair et al. [84] introduced the hyperspectral image classification approach to distinguish between the facilities in city. This approach focuses on green facilities specifically. When it analyzes green facilities, it employs texture analysis to identify the types of leaves that are present, especially in vegetable gardens. Hyperspectral image classification is based on DCNN and Gabor filtering. The Gabor filtering is used to extract spatial features from hyperspectral images. These features are then fed as input to a DCNN for classification. The proposed method underwent testing across three distinct datasets, achieving an average accuracy of 95.8%. This surpasses the performance of alternative approaches such as 2D-CNN, machine feature learning, and deep feature fusion networks that utilize different CNN architectures.

4.1.4. Color-Based Object Detection

Color-based object detection analyzes color distribution in an image or video frame and identifies objects that match a predefined color model. The authors in [85] identified plant diseases in a city area using a kNN classifier with color image segmentation in 2019. The kNN classifier attempts to measure the similarity of a pixel with its closest neighboring pixels; it uses a distance matrix for the similarity calculation. The segmented regions are represented by three distinct colors: one color represents the leaf, the second color represents the disease, and a third color represents the background. During the experiment, the classifier parameters were set to utilize three nearest neighbors for the detection of five types of plant diseases. The proposed approach achieved a recognition accuracy of 96.76%. In comparison, systems based on SVM classifiers, which are commonly employed for plant disease detection, have achieved a maximum accuracy of 92.1%.

The presence of fire in an urban area can be detected by analyzing the color of the flames: images of the environment are first captured by a camera and then the flames in those images are analyzed. In [86], fire detection was achieved by analyzing the spatial-temporal features of flames in a captured video using an SVM classifier, which was then trained on these features to accurately distinguish between fire and non-fire events. The reported experiment results for the SVM classifier show an accuracy of 95.7%.

Table 2. Comparison between methods used in object classification applications.

Feature	Classification	Method	References	Description
Object classification	Motion-based object detection	Deep learning	[28–30,49,73]	Analyzes patterns of motion and changes in appearance over time, is suitable to real-time systems, and provides high accuracy
		Frame differencing	[68,69,72]	Computationally less complex and suitable for dynamically changing environment
		Optical flow	[70,71]	Requires more complex computational methods but is more accurate than frame Differencing
		Traditional computer vision algorithms	[34,74–76]	When algorithms are tailored to specific tasks and applications, they can achieve higher accuracy and efficiency in solving the problem at hand

Table 2. *Cont.*

Feature	Classification	Method	References	Description
	Shape-based object detection	GAN discriminator, DNN, multi-view, deep learning	[32,78,79,81,82,87]	Deep learning algorithms are trained to detect objects based on their shapes and other features
	Texture-based object detection	Gabor filtering, LBP, GLCM, LBP histogram	[31,83,84]	Feature extraction uses an algorithm based on the texture of the object, and objects are classified by deep learning algorithms
	Color-based object detection	SVM, kNN classifier	[85,86]	Analyzes the color distribution in a video frame and identifies objects that match a predefined color model

4.2. Object Tracking

Object tracking in video surveillance is a process of automatically detecting objects of interest in a video stream and following the objects over time. The goal of object tracking is to keep track of objects as they move through the frames of the video, even when the objects are partially or fully occluded. Object tracking in video surveillance can be realized by numerous techniques for a wide range of applications, including security, traffic monitoring, and gesture recognition [88]. For example, point-based, kernel-based, and silhouette-based approaches can be applied to object tracking. The selection of an appropriate method for a given task depends on the specific goals and requirements of the task [89]. A summary of these three methods is described in Table 3.

4.2.1. Point-Based Object Tracking

The point-based approach involves identifying and tracking a specific point on an object. Kalman filtering and particle filtering are frequently used in this approach. In [35], a multi-object detection and tracking (MOTD) method was proposed in 2019. This method uses Kalman filtering with a probability-based grasshopper algorithm (PGA) to track multiple objects on roads and sidewalks. Initially, morphological operations and region growing are used to extract objects from pre-processed frames. Then, Kalman filtering, and the PGA estimate the motion of the tracked object using optimized parameters. The Kalman filter improves the tracking rate until 86.78%, by utilizing previous state evaluations to assess the current state of the object. In the experimental phase, when utilizing the CAR dataset, the proposed system attained an accuracy of 69.22%. When contrasted with a Kalman-filtering-based system and an optimal partial filtering combined with a morphological operation-based system, the disparities in accuracy were minimal, with differences of 4.22% and 11.2%, respectively. Cob-Parro et al. [90] proposed a system for human movement tracking using a Kalman filter bank on a low-power embedded device with an accuracy of 87.82% in 2021.

Issam et al. [91] proposed a method for the detection and tracking of moving objects for street surveillance with an accuracy of 98%. Because it is not possible to calculate how many objects will be on a street in advance, a general filter is created when the system starts. When a new frame containing an object is encountered by the system, the initialized filter determines if the object is new or not; if the object is new, the filter generates a new particle filter for that object. This process repeats for each moving object, which ensures that a dedicated filter is created for tracking purposes. Subsequently, when an object either stops or exits the scene, the corresponding filter that is responsible for tracking that object is removed. This methodology enables the efficient detection and tracking of multiple objects in real-time video surveillance applications.

Due to the limited borders that the object was prohibited to reach or unexpected obstacles, it is necessary to pre-calculate potential motion boundaries. Zhu et al. [92] achieved this by utilizing a distance transform state and particle filtering to determine safe

boundaries for objects like a person walking on the road. The distance transform state calculates the appropriate distance between the object (dynamic state) and the environment (environment state), and the particle filtering uses particle sets to represent the probability distribution of the object's state. A moving object is represented by rectangular bounding boxes. The position and size of the bounding box is continuously updated according to the object's movement. As the object moves, the particles update to reflect the object's possible new positions. The distance transform state evaluates the safety of each particle's position. Another approach to object tracking in the presence of unexpected obstacles is to utilize a correlation-filtering-based algorithm. By employing a correlation tracking algorithm, the system can track the object based the characteristics of its appearance [93]. The basic idea of correlation-filter-based tracking is to train a filter using information about the appearance of the object in the initial frame, and then to use this filter to locate the object in subsequent frames.

4.2.2. Kernel-Based Object Tracking

Kernel-based object tracking is a popular approach for tracking objects in video sequences. This approach estimates the location and motion of an object of interest in consecutive frames of a video. In kernel-based object tracking, a kernel function measures the similarity between the object being tracked and regions in subsequent frames of the video sequence. By comparing the similarity scores of different candidate regions, the tracker can determine the most likely location of the object in the current frame [94,95]. Chen et al. [96] presented a model that used low-rank representation with contextual regularization to track moving objects in indoor and outdoor environments in 2017. This approach separates the background and foreground using a custom-designed cost function. As a result, the foreground mask can be precisely identified with a 96% accuracy rate in 46 sequences. Furthermore, they introduced sparse and low-rank representation with contextual regularization (SLRC) in 2019 [97]. SLRC utilizes a specially designed cost function to distinguish between the background and the foreground in multiple scenarios. In the foreground model, objects in motion are identified as connected segments that each have a relatively small size. At the same time, the background model adheres to the principles of low-rank and sparse representation in each scenario. The new model effectively breaks down complex video sequences into distinct background and foreground elements. This significantly boosts the accuracy of detecting moving objects in individual scenarios. Consequently, the system's overall performance enhances up to 95% accuracy in 60 scenarios, which enables the model to simultaneously detect multiple moving objects with exceptional precision, surpassing the capabilities of the previous system.

Object tracking algorithms that employ kernels have shown a remarkable real-time performance with they use bounding boxes. Jha et al. [98] proposed an N-YOLO approach with a correlation-based tracker to produce efficient bounding boxes that predict the movement of the object in 2021. To achieve object localization and classification in urban environments, N-YOLO divides an image into a grid of uniform size and extracts two candidate bounding boxes per grid. By examining the content of each candidate bounding box, the object contained within it can be recognized using the associated class identifier. Then, we merge the detection results of each sub-image using a correlation-based tracking algorithm. In the evaluation of YOLOv3 and N-YOLO using a road traffic dataset, the results revealed that N-YOLO achieved a 7% accuracy enhancement according to the balanced accuracy metric. Notably, the study underscored the significance of bounding box quality in optimizing correlation-based object tracking. The N-YOLO showed the most efficient bounding box among the various object detection algorithms examined. Furthermore, the object tracker served a dual role by functioning as a merging manager for detected objects, leading to improve linear scalability.

4.2.3. Silhouette-Based Object Tracking

When objects have complex shapes, such as shoulders, fingers, and hands that cannot be accurately described by simple geometric shapes, silhouette-based tracking can be used to define the precise shape of the object and to enable accurate tracking.

This method is capable of processing occlusion and object fragmentation and merging, as well as various complex object shapes [99]. Kanagamalliga et al. [100] used the contour tracking method with optical flow with an accuracy 94%. The counter tracking method extracts shape and optical flow features from a detected object. This approach obtains an accurate bounding silhouette that indicates the tracked object. Contour-based object tracking is then performed by locating the object region in each frame using an object model created from previous frames, which can be used to monitor someone running or moving in place. Because contour tracking is suitable for non-rigid object structures, the object shapes are considered to be boundary silhouettes, and the tracking results obtained are dynamically updated in the video frames.

Table 3. Comparison between methods used in object tracking applications.

Feature	Classification	Method	References	Description
Object tracking	Point-based	Particle filter, Kalman filter, correlation filter	[35,70,90–93]	Identifies specific points on an object's surface and monitors their movement over time to track the object's position and motion
	Kernel-based	YOLO, sparse low-rank Representation	[96–98]	Uses a probabilistic model to estimate the object's position and motion based on a set of kernel functions
	Silhouette-based	Contour tracking	[100]	Deals with objects having complex or irregular shapes

4.3. HAR

Human action detection is a function that is essential for maintaining public safety and essential for private healthcare. Suspicious or criminal behavior, accidents such as someone falling, and dangerous situations can be detected with this function. However, detecting human actions can be a challenging task because there are many variables to consider, such as body shape and gait, that vary according to the individual's psychological and physical state [101].

In recent years, deep learning has emerged as a popular approach to addressing the challenges of HAR. Deep learning techniques provide increased flexibility and effectiveness in analyzing and understanding the patterns of human motion [102]. HAR applications briefly shown in Table 4.

4.3.1. Abnormal Action Detection

Abnormal action detection in HAR refers to the task of identifying actions or behaviors that deviate from what is considered normal or expected. Abnormal action detection focuses on identifying actions that are uncommon, unusual, or potentially dangerous, which include falling, stumbling, abnormal body movements, sudden changes in speed or direction, and actions that are out of context in a given environment [76].

Yair et al. [103] designed a temporal CNN that uses spatiotemporal features to analyze and recognize human actions that require immediate analysis, like sudden falls, or loss of consciousness in a public area, using only a short video as input in 2022. Convolutional short-term memory is used in this type of CNN, because correlation with previous frames is not relevant as only one moment can be monitored at a time. Temporal CNN allows for the simultaneous analysis of multiple frames, which enables the detection of changes in displacement or object size over time. The accuracy of the proposed system was evaluated using the Microsoft Research (MSR) daily activity 3D dataset. In comparison the proposed system achieved an accuracy of 95.6%, compared to the lower accuracy, 90.8%, in [104].

In HAR, it is possible to monitor the human body with the help of a wearable device in addition to a camera [105]. Sensors are integrated into wearable devices, such as smart watches or fitness trackers, and data on the movement of the user's body are collected. The collected data are then processed and analyzed using machine learning algorithms to identify different human actions. The accuracy of the recognition depends on the quality and quantity of data collected and the complexity of the machine learning model that is used for analysis [106–109]. According to [106], wearable devices achieved about 96% accuracy which surpasses that of the camera-based system presented in [103] for HAR. In addition to wearable device monitoring, double monitoring with a camera will increase the accuracy of action recognition. In [110], data collected by wearable devices was transferred to a nearby edge using an UAV. After the data were transferred to the edge, they were processed using various algorithms and methods to extract insights and identify any potential issues or anomalies. If any issues were detected, appropriate actions could be taken by nearby medical institutions, such as alerting the user or sending notifications to healthcare providers.

4.3.2. Action Classification

Human-behavior-based action classification extracts meaningful features' video data input and employs deep learning algorithms to classify and identify specific actions or behaviors, including surveillance systems, activity recognition, video analysis in sports, healthcare, and public safety [111,112].

The CNN recognizes patterns in the input data that have a spatial structure [113], whereas LSTM is used specifically to capture patterns that change over time in sequential data. To combine the strengths of both models, a CNN-LSTM hybrid architecture was developed in [114] to monitor and recognize human actions in indoor environments in 2022 with an accuracy of 90.89% on 30 frames. In this hybrid model, the CNN is used to extract discriminative features from the input data, and then the LSTM is used to learn and model temporal dependencies between these features. The CNN-LSTM hybrid model dataset contains different physical activities, and thus can provide better monitoring of an individual's health. This architecture has proven to be effective in recognizing complex actions that involve both spatial and temporal patterns.

In [115], a spatiotemporal transfer-learning-based framework was proposed to recognize similar or overlapping actions in sports activities, such as skipping rope, jumping for football headshots, and skateboarding in 2022. Transfer learning involves the use of a pre-trained CNN to extract deep features from video frames, which are then compressed using a deep auto-encoder to reduce their dimensionality. This compressed representation is then fed into an RNN with an LSTM to capture long-term temporal information and learn the hidden patterns in the visual data stream. The RNN with LSTM model achieved an accuracy of 96.3%. Examining the hybrid systems outlined in [114,115], it becomes evident that the choice between RNNs and CNNs for human action classification hinges on the nature of the task. When the task necessitates a strong emphasis on capturing temporal dynamics—like discerning intricate actions or gestures that evolve over time—RNNs tend to be better-suited. On the other hand, if actions can be differentiated effectively by visual patterns and local features within individual frames, CNNs emerge as a favorable option.

In healthcare, HAR can be used to monitor and assess patient movement and activity levels. Monitored results can then be useful for elderly care, disease management, and telemedicine. Rajavel et al. [116] introduced a cloud-based object tracking and behavior identification system (COTBIS) for monitoring remote patients and elderly people. COTBIS has four layers: the sensor layer, edge/fog computing layer, cloud layer, and consumer layer. The sensor layer captures live video from surveillance cameras and records data, which are then transmitted to the edge layer. The edge layer filters out the non-sensitive data using its edge computing framework and sends only the necessary data to the cloud layer. In the cloud layer, a CNN is used to analyze the data and make decisions. The system then sends notifications of the remote patient's activities and triggers alarms to

alert the caretaker or ambulance service when necessary. Therefore, using cloud and edge computing increases accuracy and reduces computation time. In the experiment, authors monitored the movement of the patient at home using three different methods: a proposed CNN classifier, an SVM classifier, and a linear regression classifier. Among these, the proposed CNN method outperformed the others, achieving an accuracy of 94.5% in just 72.76 s. The SVM classifier also yielded good results, with an accuracy of 90.32% in 81.32 s. Meanwhile, the linear regression classifier had a decent performance, achieving an accuracy of 80.64% in 83.54 s.

Disease management benefits from the use of multiple cameras to monitor and collaborate with DNNs. Research has shown that using multi-camera setups can achieve up to 98% accuracy in tracking human movements [117]. However, using multiple cameras to monitor individual households on a regular basis may not be practical or cost-effective. During a pandemic, a feasible solution using a single camera or a drone to monitor multiple households in the same area emerges [118]. This approach could reduce costs while still providing valuable information for disease management efforts with false positive rate of 4%.

Table 4. Comparison between methods used in human action detection applications.

Feature	Classification	Method	References	Description
Human action detection	Abnormal action detection	Deep learning, Blockchain	[76,103,105,110]	To enhance security, it is important to identify abnormal actions among people in both outdoor and indoor environments
	Action classification behavior analysis	Deep learning	[113–117]	Identifying and classifying human actions and behavior systems help to automate the detection of suspicious behavior and improve security and safety

4.4. Anomaly Detection

Anomaly detection in video surveillance involves identifying unusual or unexpected events or behaviors in video footage captured by surveillance cameras. This detection is crucial for maintaining safety. Taking action based on the detection of such activity is an effective way to address the abnormal issues [2]. This section explores the latest advancements in detecting abnormal behavior, including the identification, and monitoring of anomalies in road traffic, the environment, and human activity as shown in Table 5.

4.4.1. Road and Traffic Anomalies

Anomalies occur frequently in traffic, so it is important to identify common problems on the road and to develop strategies to prevent them [119]. Because not all accidents are alike and the types of accidents can change over time, deep learning methods are utilized to improve feature extraction and to keep up with the evolving nature of accidents. To classify abnormal activities, it is common to identify the area of interest and to evaluate activities as normal or abnormal using a specific method [120–122].

Zhou et al. [120] proposed a system called AnomalyNet for monitoring avenues in 2019. Their system considered people walking as normal, and considered the presence of bicycles, electric cars, etc., entering the avenue as abnormal. AnomaylNet consists of a motion fusion block, feature transfer block, and coding block. These blocks work together with neural networks for feature learning, sparse representation, and dictionary learning to perform action classification. The motion fusion block compresses multiple video clips into a single image, which can then be used in a sequence of RGB frames to make a CNN-less complex. The feature transfer block extracts deep features from the original data, and the coding block optimizes the network to achieve fast and accurate results. When evaluating the AnomalyNet system on the avenue dataset, which comprised 47 abnormal events, it achieved an accuracy of 95.6%. In comparison, when applying the methods

proposed in [123,124] to the same dataset, the accuracy rates were as follows: [123] achieved 92.3%, [124] achieved 91.8%, and [125] achieved 95.2%. In [121], a CNN was initially used to extract features, and a MRCNN is employed for further classification with an average accuracy of 97.5% in 2020. MRCNN can be used in areas where many people and cars gather, because all the objects recorded in the video frame can be semantically segmented one by one. The MRCNN solves instant segmentation problems, which involve both object detection and pixel-level segmentation, where the goal is to identify and segment individual objects within an image. By incorporating factors such as the object class, bounding box, and mask, the MRCNN is able to achieve accurate results. Additionally, the advantage of this approach is that it can provide precise classification within a relatively short timeframe. Although, MRCNN is effective in well-lit environments, it may fail to detect objects or to produce accurate results in low-light situations. In such cases, employing a DCNN may be a better alternative [122]. DCNNs can classify images even under varying levels of brightness, and they utilize advanced feature descriptors to enhance their classification performance. The utilization of DCNN for tracking anomaly movements in low-light environments resulted in an impressive accuracy of 92.15%. Meanwhile, the accuracy rates of other methods such as KNN, SVM, NN, and CNN were 80.97%, 80.02%, 87.20%, and 90.65%, respectively.

Nawaratne et al. [126] presented a system that uses unsupervised deep learning called ISTL. By using unsupervised deep learning techniques, the ISTL system can learn and adapt to new instances without relying on pre-labeled training data. This ability to dynamically incorporate new objects enables the system to process a wider range of objects and adapt to evolving environments. For example, suppose a person on a bicycle and an electric cart appear on a sidewalk. In experimental analysis, authors compared the performance of the ISTL method against other techniques including Conv-AE, S-RBM, ConvLSTM-AE, and unmasking methods. Interestingly, the outcomes varied depending on the dataset being used. For instance, on the UCSD Ped2 dataset, the ISTL method exhibited the highest accuracy at 91.8%. On the other hand, when considering the UCSD Ped1 dataset, the Conv-AE approach achieved the highest accuracy of 81%. ISTL is capable of detecting anomalies in real-time video surveillance with an accuracy of 91.1%, but unlike the automated systems, it relies on human observation, which is one of its limitations.

4.4.2. Concealed Weapon Detection

Controlling and limiting the carrying of cold weapons in public is an important measure for ensuring public safety. Using a single camera to detect objects held in a person's hand can lead to false alarms, as the camera may not be able to distinguish between items such as a mobile phone, money, bills, cards, and a pistol or knife. To reduce the occurrence of false alarms, differentiating between these object is necessary. To address this issue, a two-level methodology based on deep learning was proposed in [32], which achieved 88.5% accuracy. In the first step, a CNN is used to select candidate regions in the input. In the second step, a binary-classifier-based binarization technology is employed to individually analyze all the objects in the frame, which results in high accuracy. Binarizaton, also known as thresholding, is the process of converting a grayscale or color image to a binary image, in which each pixel is classified as either black (foreground) or white (background) based on a specific threshold value.

To effectively detect cold weapons, a network of multi-view cameras may be needed, rather than a single camera to monitor the carriage of weapons in public. The approach presented in [127] involves fusing binocular images captured from multiple viewpoints to create a comprehensive and informative representation of a scene. This composite image provides three-dimensional information about objects within the scene, which can help to reduce the occurrence of false positives during classification. In implementation, as the number of images within the dataset increases, there is a corresponding rise in the accuracy rate. For instance, while with a dataset of 124 images, the accuracy rate stood

only at 80.62%; the accuracy rate notably improved, reaching 87.93% upon expanding the dataset to 332 images.

The probability of false positives that occur due to problems in illumination was discussed in [128]. This study proposed a new pre-processing technique called darkening and contrast at learning and test (DaCoLT) in 2019 that utilizes brightness guidance to overcome the negative effects of varying illumination. After it applied DaCoLT, a CNN-based detection model was trained and evaluated on low-quality videos, and it demonstrated high potential as an automatic alarm system by its satisfactory results. In low-brightness environment, DaCoLT detects knifes with an accuracy 87.74%. In low-brightness environments, DaCoLT's accuracy rate slightly lags behind that of system employing DCNN [122] for anomaly motion detection. Additionally, the frame processing time of DaCoLT might experience delays due to sudden knife movements.

In [129], a lightweight multi-class subclass detection CNN (MSD-CNN) was used to classify incoming video data into abnormal frames (dangerous events such as carrying guns or knives) and normal frames (events such as walking or office work) in 2022. This MSD-CNN model is designed to be lightweight, to allow for the creation of multiple instances of the model. By applying the model to each video sequence individually using dynamic programming, one can simultaneously detect abnormal objects in multi-view cameras without significantly increasing the computational overhead. The video sequence is transferred from the main memory to the global memory for the implementation of threading and the optimization of computational resources. The MSD-CNN achieves 90.7% accuracy with respect to parameters such as different types of guns and knives, real-time deployment, and multi-view cameras.

4.4.3. Fire and Environmental Monitoring

Ensuring environmental safety involves several measures, which include early fire detection as a crucial aspect of public safety. Datta et al. [37] proposed blockchain and edge/drone-based secure data delivery for forest fire surveillance (BESDDFFS) in 2021. The BESDDFFS system utilizes the YOLOv3 algorithm for the drones to predict areas that have potential for fire. If the calculated probability of a fire is greater than 50%, the data are sent to an edge node via the leader drone. The edge node then verifies the data using a Merkle tree-based validation algorithm. This way, the BESDDFFS system ensures secure data delivery with up to 66% lesser delay, 36% greater throughput, and 12% greater ratio of successfully delivered packets and accurate verification of potential fires than previously proposed system [130], which helps in the early detection and prevention of forest fires.

Detecting flames is crucial to fire detection because the extent of damage caused by a fire is determined by how quickly the flames are detected. Flames can be recognized by analyzing the color, shape changes, and movement that are captured by the camera [131]. In 2019, Mahdi et al. [86] developed an automated system for detecting flames using a combination of several techniques. Their system first utilizes the imperialist competitive algorithm (ICA) to extract color-based candidate regions. Second, it applies a motion intensity-aware motion detection technique to further refine the candidate regions. Finally, an SVM classifier is applied after the candidate regions are identified and refined. The SVM classifier is trained using supervised learning with labeled samples of real fire and non-fire regions to distinguish between these two classes, based on the extracted features. In their experiment, the researchers utilized various datasets, encompassing indoor fires, outdoor fires, non-fire scenarios, and moving objects that closely mimicked the fiery color. For instance, the outcomes of the Mivia dataset experiment achieved an accuracy of 95.32%, outperforming another CNN-based system [132] that achieved 94.39% accuracy.

In [133], a modified version of the YOLOv3 algorithm was developed to detect and classify regions affected by fire with an accuracy of 98.9% in 2021. The authors improved the accuracy of the recognition method by training the algorithm on images that specifically contained the red color that is typically associated with fire. By training the modified YOLOv3 algorithm on fire-related images, the model becomes more adept at accurately

calculating the probability of smoke appearing before any flames become visible. This training approach allows the algorithm to utilize a dataset that contains images of both flames and smoke, which enables it to effectively recognize and differentiate between these two features. In [134], a DCNN model approach inspired by the GoogleNet architecture was designed to detect both flames and smoke, and it achieved a 65% success rate in detecting smoke.

Table 5. Comparison between methods used in anomaly detection applications.

Feature	Classification	Method	References	Description
Anomaly detection	Road and traffic	Deep learning	[119–122,126]	Analyzes video by identifying and classifying unusual or anomalous events that occur on a road network, such as accidents, traffic congestion, or hazardous conditions
	Concealed weapon detection	Deep learning	[32,127–129]	Identifies potential threats and prevents accidents in public places by identifying cold objects such as knives, swords, and axes in real-time video through deep learning
	Fire and environmental monitoring	Deep learning, edge computing	[37,86,131,133,134]	Identifies potential fire hazards and takes appropriate actions to prevent the spread of fire and minimize property damage and human casualties

4.5. Secure Video Data Management

Data security in a VSS refers to protecting from unauthorized access, tampering, or loss the video footage, and other related data captured by surveillance cameras. Video surveillance data typically contain sensitive information, including private, criminal, sexual, national security, and financial information, which must be safeguarded to prevent data breaches and ensure privacy. Unauthorized alterations can compromise the authenticity and originality data media, which makes the safe management of data essential. However, videos are often leaked or viewed by unauthorized persons, and this poses a significant risk to the security of the videos. To address this problem, many organizations are turning to blockchain systems, which offer tamper-proof and secure cryptography, and fault-tolerant features to ensure system reliability and robustness [13]. Table 6 lists references from the literature that use blockchain for authentication, authorization, and data storage security.

4.5.1. Authentication and Authorization of Video Data

Once a user and a device have been authenticated, the system can then use authorization mechanisms to determine what actions the user is allowed to take. In order to establish a secure and authorized environment for a surveillance system, blockchain technology can be utilized. Different types of blockchain architecture are available, including public, private, and permissioned blockchains. In the context of video security, the use of private or permissioned blockchains instead of a public blockchain is recommended, as private blockchains use control mechanisms to regulate access to videos to ensure that the blockchain network is only accessible to authorized participants [135].

Before the collection of data begins, the system registers all devices that will be used and ensures that only registered devices can send data to the system [56]. Furthermore, a surveillance system may be needed to limit video recording to a certain portion of the camera's field of view. This can be relevant in a sensitive location where privacy concerns are high, such as a fitting room or a bathroom. The BlockSee system [57] sets camera settings including camera position, direction of view, and zoom level. The camera setting is guaranteed by camera manufacturers, certified installers, citizens, and court officials. Each CCTV device connects to the blockchain with a unique hashed key value, and the

client nodes must be authenticated by a membership service provider (MSP) [13]. The MSP is managed by a city-wide oversight authority. After a client node is authenticated, it gains access to the blockchain network. The authenticated client can then generate a task to verify the authenticity of CCTV images stored on the blockchain. This task compares the hash of the extracted image with the hash stored on the blockchain to verify if it has been tampered with or not. In [136], a blockchain-empowered surveillance architecture (BUMAR) system that employed UAVs to monitor fishing boats was introduced. The BUMAR system uses a two-phase authentication process to verify the identity of the fishing boats that are visible in the UAV monitored area. First, all valid boats are registered and stored in the blockchain. When a fishing boat enters the UAV monitored area, the UAV attempts to validate the boat using the two-phase authentication process. The result of this validation is then reported to an edge server, and then the edge control server takes further action based on the result. The authentication time of validating new UAV is faster than using hash map and sequential search method.

4.5.2. Video Data Integrity

Video data integrity can be easily verified by comparing the hash of video data and the hash written by an original publisher on a blockchain. To do this comparison, a video recording camera device is initially configured to ensure that the recorded data is secure and tamper-proof [137]. The basic concept is that if a camera is set up with secure settings, all subsequent recordings made by that camera will be considered tamper-proof and all recorded data will be saved in the blockchain. This makes it suitable for controlling a specific area, such as a small district of a city or a household.

Kerr et al. [138] combined blockchain with digital watermarking to provide a secure and reliable method for storing video evidence. This method is well-suited to the safeguarding of CCTV data against unauthorized distribution for repurposing. In this blockchain system with blocks created on the cameras, video segments are linked to these blocks using watermarks that are embedded in the video stream. This ensures that the video evidence is securely stored on the blockchain and cannot be tampered with or manipulated.

In Hao Li et al. [139], a video management system in city introduced to ensure the integrity and availability of video data. The system implemented to monitor the environment in the central part of the city. All the cameras installed for the system and the users who wish to view the video data are required to register with a trusted authority and to obtain their unique key. After the registration process, the users are permitted to access the video data. To enhance security, the recorded data from all registered and deployed cameras are encrypted before they are stored on the blockchain. Specifically, the system encrypts the ROI within the video and stores the video on a server that can only be accessed by authorized devices that possess the corresponding key. The video security decryption speed can reach 91.47 Mb/s on average. Lee et al. [140] developed a system that utilizes a Merkle-tree to guarantee secure synchronization in a blockchain environment. CCTVs transmit all recorded data to a cloud server, which encrypts the received data before sending it to all the blockchain nodes using a Merkle tree. A Merkle tree is a data structure that allows for efficient verification of data integrity and synchronization, and it accomplishes this by dividing a large amount of data into blocks of a specific size and placing these blocks in the corresponding leaf nodes of the tree. The synchronization between the cloud server and the CCTVs are ongoing, which means that any new data recorded by the CCTVs are transmitted to the cloud server and synchronized within the blockchain network. This system provides an economical solution for transmitting CCTV data within smart city public safety systems. It leverages the benefits of the Merkle tree data structure, blockchain technology, and cloud servers to achieve secure synchronization while optimizing data transmission efficiency.

4.5.3. Distributed Video Data Storage

A city's surveillance systems generate a significant amount of video data, which require generous storage space. Due to the substantial size of the data and the associated economic limitations, it is often infeasible to store all of the data on a blockchain. Therefore, many surveillance systems adopt the IPFS and off-chain storage to save all the data. By using the IPFS and off-chain storage, surveillance systems can reduce the costs associated with storing large amounts of data on the blockchain while still benefiting from the security and transparency that the blockchain provides [141]. In a typical blockchain system, blocks store hashed references and relevant access details as transactions, whereas other data are stored off of the blockchain. The off-chain storage is usually connected to the blockchain for data integrity and confidentiality. This mechanism may involve the use of blurred keys, which are cryptographic keys that have been obfuscated in a way that makes them difficult to reverse engineer or duplicate [142].

The IPFS can be used as a decentralized storage layer for blockchain applications. When a file is saved in the IPFS, the system generates a unique hash based on the file's contents. This hash can then be stored on the blockchain as a transaction, and this stored transaction creates a tamper-proof record of the file's existence. The hash can also be used to retrieve the file from any node on the IPFS network, which provides a reliable and decentralized storage solution for blockchain applications [143,144]. In [145], a proposed surveillance system used blockchain and the IPFS to store videos with restricted access. In this system, the blockchain technology allows for the creation and processing of licenses for the videos, and the IPFS serve as a distributed and decentralized file storage system. Access to the videos is restricted to authorized users through the use of a digital rights management (DRM)-enabled video player.

Tsai et al. [146] optimized the storage space and computational power while still retaining as much information as possible by using downsampling methods. To accomplish this, a downsampling decision maker evaluates the space available and assigns a quality level to each video. Then, a predictor determines the amount of space needed and the computational resources required to perform the information analysis and video clip downsampling at an appropriate time. Dave et al. [147] suggested a private blockchain for storing personal information, such as people's faces, that involved using chaotic masking at fog nodes to blur privacy-sensitive objects before storing the processed data securely on the blockchain.

Table 6. Comparison between methods used in video data management applications.

Feature	Classification	Method	References	Description
Data storage security	Authentication, authorization	Blockchain	[13,56–59,136]	Both authentication and authorization are crucial to maintaining the security and integrity of a VSS
	Data integrity	Blockchain	[137–140]	Helps to prevent or mitigate security breaches and reduce false alarms
	Distributed video data storage	Blockchain, off-chain, IPFS	[141–147]	Sensitive information contained data can be achieved through the use of encryption, access controls, and other security measures

5. Challenges and Future Work

Over the past 20 years, numerous research studies have focused on the development of automatic VSSs. These studies have addressed challenging issues related to video surveillance. As a result of these efforts, many approaches and algorithms have been proposed and implemented successfully and have led to feasible and effective outcomes. Although significant progress has been made further improvement in the effectiveness of

VSSs are needed. In this section, we introduce, as our future work, tasks that still require improvement in terms of robustness and accuracy.

5.1. Drone-Based Monitoring System

Drone-based surveillance systems are useful for recording video in areas that are unreachable by fixed cameras on the ground. By transmitting recorded data to a nearby server or edge node, video data can be analyzed in real time to detect objects or situations and to respond [16]. In particular, it is possible to track moving people and animals even when they disappear accidentally from the scene or exit the camera's field of view [12]. Also, drones are useful for weather and air pollution monitoring systems as environment monitoring, but monitoring error or data loss due to strong winds or heavy rain is problematic. Additionally, due to the possibility of access to the vulnerable personal data such as home address and the human face, robust privacy and personal data protection are necessary ensuring compliance with data protection laws, and transparently open the use and limitations of the technology to the public.

Furthermore, drones are increasingly being employed for search and rescue operations. For instance, leveraging algorithms like the Firefly algorithm can aid in rapidly estimating flood levels and spread rates [148]. In urban disaster situation, drones can also serve as transporters, facilitating the delivery of emergency medical supplies in addition to the surveillance [149]. As the drones can intentionally or unintentionally enter restricted flight areas, they can be potential threaten to national security by spying secretly, disturbing flight of passenger aircraft, destroying public infrastructure, etc., To prevent those misuses as in [150–154], counter-drone technology is widely under study, which for example, enables the configuration of guarded areas to either permit or block drones.

Despite the beneficial use of drones in smart cities, the inherent problems of drones are also significant challenges such as limited flight duration due to short battery lifetime, limited CPU resource and memory, and security concerns [7,155,156]. For instance, the battery limitation of the drones increases interest in solar energy operation for the sustainable VSS [11], which, however, have serious drawbacks like weather condition. Alternatively, a power line from a ground center to drones can be used for consistent power supply, but which limit surveillance region and is unstable for strong wind.

5.2. Unsupervised-Learning-Based Surveillance System

In object classification, supervised deep learning methods are commonly used to assign appropriate labels to objects. In supervised deep learning, a deep neural network is trained on a dataset in which each object is labeled with its corresponding class. During training, the algorithm learns to identify patterns and features that are unique to each class. After the algorithm is trained, it can be used to classify new objects based on the patterns and features it has learned. Indeed, as mentioned in [114,157], when new types of operations and objects are not labeled or known in advance, unsupervised deep learning techniques can be explored to improve object classification and recognition. Unsupervised deep learning is a type of machine learning that uses unlabeled data to train models that then identify patterns and relationships within the data.

The framework proposed in [97] uses an ISTL within unsupervised deep learning to classify abnormal actions. The unique aspect of this learner is that it dynamically adjusts and re-determines the anomaly value based on the characteristics of the data. Therefore, all new incoming actions can be classified using updated values. In the context of detecting anomalies in the marine domain [158], a challenge arises from the constrained availability of labeled data for supervised approaches, leading to potential oversight in detecting certain abnormalities. Employing an unsupervised guided background modeling approach can help enhance background updates and improve the accuracy of foreground detection [159]. For example, it can be used to determine whether a car is parked on the road accurately even in the circumstance of nearby tree shadows and poor lighting conditions.

Although unsupervised-deep-learning-based systems have worked effectively, there are several challenges like human intervention, scalability, interpretability, and domain adaptation that still need to be addressed. For instance, unsupervised deep learning requires human intervention to control unsupervised data, as accuracy decreases when classifying unlabeled data by its own metrics, and space is inefficiently added by unimportant objects as new categories.

5.3. False Alarm Reduction

Alarms that are based on analytics plays an important role in a VSS for public safety. VSSs can use advanced analytics algorithms to automatically analyze video footage in real time and to identify potential security threats. When a potential threat is detected, the system can trigger an alarm or notify security personnel of the potential threat, which enables the personnel to take appropriate action. However, it is important to note that not all alarms generated by the system are necessarily indicative of actual threats. In VSSs, false alarms may be triggered by a variety of factors, which are listed in Table 7. False alarms can be a major issue for security personnel, as they can waste time and resources and can also lead to a decrease in the credibility of the system if they occur frequently. Furthermore, employed deep learning can lead to numerous false alarms if there is a poor alignment between the environment and the algorithm in [160].

Table 7. Examples of false alarms.

System	Method	Reason for False Alarm
[126]	Deep learning	Alarms are recognized as abnormal when unknown or new normality appears
[121]	Neural network	What is considered an anomaly today may not be considered an anomaly tomorrow due to the lack of data
[161]	Deep learning	Post-incident alarm triggering occurs from training crowd safety analysis on only human movement
[116,118]		In order not to create false alarms in the healthcare system, it is necessary to use a human observer
[162]		Intruders may deliberately trigger false alarms by covering or tampering with cameras

Blockchain smart contracts, multi-factor authentication, and multi-class deep learning can be used to mitigate false alarms. Smart contracts deployed on a blockchain help reduce false alarms by providing a decentralized peer review and tamper-resistant framework for alarm management. Smart contracts may include rules and conditions for alarm triggers, to ensure that only valid and authorized events activate alarms [37]. Multi-step approaches in VSSs can indeed add extra layers of verification to check the alarm trigger conditions multiple times. This reduces false alarms and ensures that only genuine threats are detected.

The technology employed to minimize false alarms in VSSs is continuously advancing and improving. Regular updates are essential for keeping system up-to-date and for maintaining optimal performance. Therefore, ongoing research that focuses on action and object classification is crucial to enhancing public safety while effectively reducing false alarms.

5.4. Multi-Modal-Based System

In video surveillance, relying solely on camera sensors to monitor the environment may limit performance. However, by integrating additional sensors such as sound, image, and temperature sensors, a multimodal system may be created to enhance surveillance capability by obtaining a more comprehensive understanding of the environment. Traditionally, machine learning algorithms have mainly focused on processing unimodal data, such as text or images, in isolation. However, in many real-world scenarios, data from

multiple modalities need to be considered together to obtain an accurate and complete understanding of a situation.

Multimodal machine learning has been intensively studied to develop models and algorithms that effectively leverage information from different modalities [163]. In a VSS, a multimodal system captures a broader range of information by combining video data from the camera sensor with other data from additional sensors. For example, wearable sensors such as accelerometers, gyroscopes, and magnetometers capture detailed information about a person's movements and posture, whereas cameras capture visual information about the person and their surroundings [114,116,164,165]. By combining the data from these sensors, a system can recognize when a person is not in the camera's field of view. In [166], multi-sensors are used to distinguish falls and human daily motion from human motion. Feature extraction is performed with the help of machine learning, and after that, movement classification is performed with the help of logistic regression, and optimal classification is performed in a short time. In [167], data gathered from physical motion, ambient, and vision-based sensors undergoes individual pre-processing tailored to each type. These specific pre-processors optimize the data for their respective category. Subsequently, the outcomes from each pre-processor are merged, resulting in a reduction of errors stemming from the intricate aspects of motion-related challenges.

In addition to monitoring forest fire detection through a single camera, establishing a collaborative multi-model system that combines smoke sensors, temperature sensors, and drought condition meters yields a substantial reduction in the risk of false alarms [168]. All sensors report the sensed data to a base-station, that utilizes Neuro fuzzy algorithms to process the sensor data and a CNN to process the image data. The processed data are then evaluated to determine if there is a risk of fire. If a fire risk is detected, the base station generates an alarm that is promptly sent to the forest department for the necessary response.

While multimodal systems offer significant advantages, they also face several challenges including storage management, real-time processing, and format adjustments in handling data from multiple diverse sources. For instance, different sensors typically have different data formats, resolutions, and sizes, which requires standardized rule for data fusion. Moreover, environmental conditions need to be considered as sensors may respond differently to such as lighting, weather, and temperature changes. Required processing power to handle data from multiple sensors in real-time increases according to degree of modality.

5.5. System Resource Management

VSSs generate a huge amount of data that runs to millions of records. Analyzing and managing such a large amount of data can represent a significant challenge in terms of both cost and complexity. System resource management in video surveillance refers to the effective allocation and utilization of various resources, including computational power, storage, bandwidth, and network connectivity. Effective resource management ensures that the VSS operates smoothly, delivers real-time monitoring and analysis capabilities, and optimizes the use of the resources available. The following are some key aspects of resource management in video surveillance:

- Networking: Networking is critical to the operation of VSSs that deal-with real-time transmission of large amounts of data. By focusing on improving bandwidth utilization, reducing delay, enhancing scalability, and ensuring network security, robust and efficient networks can be established for the increasing demands of modern applications and to support seamless connectivity for users. Content filtering [22], compression [29], caching [169], and dynamic content delivery [50] techniques may be used to ensure efficient bandwidth utilization. Indeed, edge-computing [23] methods help to minimize delay and improve the performance of networked systems. In addition, software-defined networking is profitable for controlling network traffic and reducing the congestion of the network [11].

- Storage: Large amounts of video data must be stored for future reference and analysis. For the storage of video files in a surveillance system, the size of the system as well as data retention requirements, data accessibility, cost, and compliance regulations should be considered. There are several options available for storing videos, such as local storage, cloud storage, blockchain, and IPFS, and each option has particular advantages and use cases. Of these, the blockchain technology provides a decentralized, transparent, and immutable data storage system for enhancing video data security. Blockchain uses cryptographic hash to create a block on a decentralized network of nodes, which makes it difficult for anyone to manipulate surveillance video data stored in the blocks. Comparing to data storage in centralized systems vulnerable to single points of failure and attacks, the blockchain as a distributed system that operates on a network of nodes spread across different locations and, maintained by various participants, is robust to malfunction of a particular storage node.

6. Conclusions

Video surveillance is crucial to the advancement of smart cities. Its primary goal is enhancing safety and improving residents' quality of life. To ensure a secure environment, a combination of different activities is needed. However, from a surveillance perspective, cameras can be used to achieve various objectives, that include enhancing public security, managing traffic effectively, and preventing abnormal actions. By deploying surveillance cameras strategically, cities can monitor and respond to incidents promptly to build a safer and more convenient urban environment for residents. In this paper, we provide an in-depth review of VSSs and a presentation of related works that focus on state-of-the-art technologies from camera devices to video analysis algorithms. We also address the challenges that remain for the VSS as future research directions in order to inspire subsequent research.

Author Contributions: This manuscript was designed and written by W.K. and Y.M.-O. Y.M.-O. conducted the survey. W.K. supervised and contributed to the analysis and discussion. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Gachon University Research Fund (GCU-202106320001) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2022R1F1A1074767).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vennam, P.; T.C., P.; B.M., T.; Kim, Y.-G.; B.N., P.K. Attacks and Preventive Measures on Video Surveillance Systems: A Review. *Appl. Sci.* **2021**, *11*, 5571. [CrossRef]
2. Patrikar, D.R.; Parate, M.R. Anomaly detection using edge computing in video surveillance system: Review. *Int. J. Multimed. Inf. Retr.* **2022**, *11*, 85–110. [CrossRef] [PubMed]
3. Gawande, U.; Hajari, K.; Golhar, Y. Pedestrian detection and tracking in video surveillance system: Issues, comprehensive review, and challenges. In *Recent Trends in Computational Intelligence*; Intech Open: London, UK, 2020; pp. 1–24.
4. Rezaee, K.; Rezakhani, S.M.; Khosravi, M.R.; Moghimi, M.K. A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance. *Pers. Ubiquitous Comput.* **2021**, 1–17. [CrossRef]
5. Duong, H.-T.; Le, V.-T.; Hoang, V.T. Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey. *Sensors* **2023**, *23*, 5024. [CrossRef] [PubMed]
6. Sreenu, G.; Saleem Durai, M.A. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. *J. Big Data* **2019**, *6*, 48. [CrossRef]
7. Dilshad, N.; Hwang, J.; Song, J.; Sung, N. Applications and Challenges in Video Surveillance via Drone: A Brief Survey. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 21–23 October 2020.

8. Ezzat, M.A.; Abd El Ghany, M.A.; Almotairi, S.; Salem, M.A.M. Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends. *Sensors* **2021**, *21*, 3222. [CrossRef] [PubMed]
9. Shidik, G.F.; Noersasongko, E.; Nugraha, A.; Andono, P.N.; Jumanto, J.; Kusuma, E.J. A Systematic Review of Intelligence Video Surveillance: Trends, Techniques, Frameworks, and Datasets. *IEEE Access* **2019**, *7*, 170457–170473. [CrossRef]
10. Gavalas, D.; Nicopolitidis, P.; Kameas, A.; Goumopoulos, C.; Bellavista, P.; Lambrinos, L.; Guo, B. Smart Cities: Recent Trends, Methodologies, and Applications. *Wirel. Commun. Mob. Comput.* **2017**, *2017*, 7090963. [CrossRef]
11. Rego, A.; Canovas, A.; Jimenez, J.M.; Lloret, J. An Intelligent System for Video Surveillance in IoT Environments. *IEEE Access* **2018**, *6*, 31580–31598. [CrossRef]
12. Elharrouss, O.; Almaadeed, N.; Al-Maadeed, S. A review of video surveillance systems. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103116. [CrossRef]
13. Khan, P.; Byun, Y.-C.; Park, N. A Data Verification System for CCTV Surveillance Cameras Using Blockchain Technology in Smart Cities. *Electronics* **2020**, *9*, 484. [CrossRef]
14. Tsakanikas, V.; Dagiuklas, T. Video surveillance systems-current status and future trends. *Comput. Electr. Eng.* **2018**, *70*, 736–753. [CrossRef]
15. Jung, J.; Yoo, S.; La, W.; Lee, D.; Bae, M.; Kim, H. AVSS: Airborne Video Surveillance System. *Sensors* **2018**, *18*, 1939. [CrossRef] [PubMed]
16. Memos, V.A.; Psannis, K.E. UAV-Based Smart Surveillance System over a Wireless Sensor Network. *IEEE Commun. Stand. Mag.* **2021**, *5*, 68–73. [CrossRef]
17. Khan, M.A.; Alvi, B.A.; Safi, A.; Khan, I.U. Drones for good in smart cities: A review. In Proceedings of the 2018 International Conference on Electrical, Electronics, Computers, Communication, Mechanical and Computing (EECCMC), Chennai, India, 28–29 January 2018; pp. 1–6.
18. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. Drone-surveillance for search and rescue in natural disaster. *Comput. Commun.* **2020**, *156*, 1–10. [CrossRef]
19. Durga, S.; Surya, S.; Daniel, E. SmartMobiCam: Towards a New Paradigm for Leveraging Smartphone Cameras and IaaS Cloud for Smart City Video Surveillance. In Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–12 May 2018.
20. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
21. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An Overview on Edge Computing Research. *IEEE Access* **2020**, *8*, 85714–85728. [CrossRef]
22. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
23. Zhang, Q.; Sun, H.; Wu, X.; Zhong, H. Edge Video Analytics for Public Safety: A Review. *Proc. IEEE* **2019**, *107*, 1675–1696. [CrossRef]
24. Pan, J.; McElhannon, J. Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet Things J.* **2018**, *5*, 439–449. [CrossRef]
25. Ren, J.; Yu, G.; He, Y.; Li, G.Y. Collaborative Cloud and Edge Computing for Latency Minimization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5031–5044. [CrossRef]
26. Aslanpour, M.S.; Gill, S.S.; Toosi, A.N. Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. *Internet Things* **2020**, *12*, 100273. [CrossRef]
27. Kai, C.; Zhou, H.; Yi, Y.; Huang, W. Collaborative Cloud-Edge-End Task Offloading in Mobile-Edge Computing Networks with Limited Communication Capability. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 624–634. [CrossRef]
28. Fedorov, A.; Nikolskaia, K.; Ivanov, S.; Shepelev, V.; Minbaleev, A. Traffic flow estimation with data from a video surveillance camera. *J. Big Data* **2019**, *6*, 73. [CrossRef]
29. Zahra, A.; Ghafoor, M.; Munir, K.; Ullah, A.; Ul Abideen, Z. Application of region-based video surveillance in smart cities using deep learning. *Multimed. Tools Appl.* **2021**, 1–26. [CrossRef]
30. Nguyen, M.T.; Truong, L.H.; Tran, T.T.; Chien, C.F. Artificial intelligence based data processing algorithm for video surveillance to empower industry 3.5. *Comput. Ind. Eng.* **2020**, *148*, 106671. [CrossRef]
31. Yaseen, M.U.; Anjum, A.; Rana, O.; Hill, R. Cloud-based scalable object detection and classification in video streams. *Future Gener. Comput. Syst.* **2018**, *80*, 286–298. [CrossRef]
32. Pérez-Hernández, F.; Tabik, S.; Lamas, A.; Olmos, R.; Fujita, H.; Herrera, F. Object Detection Binary Classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. *Knowl.-Based Syst.* **2020**, *194*, 105590. [CrossRef]
33. Zhao, L.; Wang, S.; Wang, S.; Ye, Y.; Ma, S.; Gao, W. Enhanced Surveillance Video Compression with Dual Reference Frames Generation. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 1592–1606. [CrossRef]
34. Lv, T.; Zhang, H.Y.; Yan, C.H. Double mode surveillance system based on remote audio/video signals acquisition. *Appl. Acoust.* **2018**, *129*, 316–321. [CrossRef]
35. Elhoseny, M. Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems. *Circuits Syst. Signal Process.* **2020**, *39*, 611–630. [CrossRef]

36. Zhou, X.; Xu, X.; Liang, W.; Zeng, Z.; Yan, Z. Deep-Learning-Enhanced Multitarget Detection for End–Edge–Cloud Surveillance in Smart IoT. *IEEE Internet Things J.* **2021**, *8*, 12588–12596. [CrossRef]
37. Sinha, S.D.D. BESDDFFS: Blockchain and EdgeDrone Based Secured Data Delivery for Forest Fire Surveillance. *Peer-Peer Netw. Appl.* **2021**, *14*, 3688–3717.
38. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
39. Fernandez-Carrobles, M.M.; Deniz, O.; Maroto, F. Gun and knife detection based on faster R-CNN for video surveillance. In *Proceedings of the Pattern Recognition and Image Analysis: 9th Iberian Conference, IbPRIA 2019, Madrid, Spain, 1–4 July 2019*; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2019; pp. 441–452.
40. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access* **2017**, *6*, 1155–1166. [CrossRef]
41. Mumtaz, A.; Bux Sargano, A.; Habib, Z. Fast learning through deep multi-net CNN model for violence recognition in video surveillance. *Comput. J.* **2022**, *65*, 457–472. [CrossRef]
42. Leon, D.G.; Groli, J.; Yeduri, S.R.; Rossier, D.; Mosqueron, R.; Pandey, O.J.; Cenkeramaddi, L.R. Video Hand Gestures Recognition Using Depth Camera and Lightweight CNN. *IEEE Sens. J.* **2022**, *22*, 14610–14619. [CrossRef]
43. Song, W.; Yu, J.; Zhao, X.; Wang, A. Research on action recognition and content analysis in videos based on DNN and MLN. *Comput. Mater. Contin.* **2019**, *61*, 1189–1204. [CrossRef]
44. Williams, J.; Kleinegesse, S.; Comanescu, R.; Radu, O. Recognizing emotions in video using multimodal dnn feature fusion. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 11–19. [CrossRef]
45. Fan, Y.; Lu, X.; Li, D.; Liu, Y. Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, Tokyo, Japan, 12–16 November 2016; pp. 445–450.
46. Rouast, P.V.; Adam, M.T.; Chiong, R. Deep learning for human affect recognition: Insights and new developments. *IEEE Trans. Affect. Comput.* **2019**, *12*, 524–543. [CrossRef]
47. Sandhya Devi, M.R.S.; Vijay Kumar, V.R.; Sivakumar, P. A Review of image Classification and Object Detection on Machine learning and Deep Learning Techniques. In Proceedings of the 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2–4 December 2021.
48. Rehman, A.; Belhaouari, S.B. Deep learning for video classification: A review. *TechRxiv* **2021**. [CrossRef]
49. Zhao, Y.; Yin, Y.; Gui, G. Lightweight Deep Learning Based Intelligent Edge Surveillance Techniques. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 1146–1154. [CrossRef]
50. Xu, R.; Kumar, R.; Wang, P.; Bai, P.; Meghanath, G.; Chaterji, S.; Mitra, S.; Bagchi, S. ApproxNet: Content and Contention-Aware Video Object Classification System for Embedded Clients. *ACM Trans. Sens. Netw.* **2022**, *18*, 1–27. [CrossRef]
51. Ibba, S.; Pinna, A.; Seu, M.; Pani, F.E. *CitySense: Blockchain-Oriented Smart Cities*; ACM: New York, NY, USA, 2017.
52. Mora, O.B.; Rivera, R.; Larios, V.M.; Beltran-Ramirez, J.R.; Maciel, R.; Ochoa, A. A Use Case in Cybersecurity based in Blockchain to deal with the security and privacy of citizens and Smart Cities Cyberinfrastructures. In Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 16–19 September 2018.
53. Viriyasitavat, W.; Anuphaptrirong, T.; Hoonsopon, D. When blockchain meets Internet of Things: Characteristics, challenges, and business opportunities. *J. Ind. Inf. Integr.* **2019**, *15*, 21–28. [CrossRef]
54. Chattu, V.K.; Nanda, A.; Chattu, S.K.; Kadri, S.M.; Knight, A.W. The Emerging Role of Blockchain Technology Applications in Routine Disease Surveillance Systems to Strengthen Global Health Security. *Big Data Cogn. Comput.* **2019**, *3*, 25. [CrossRef]
55. Rejeb, A.; Rejeb, K.; Simske, S.J.; Keogh, J.G. Blockchain technology in the smart city: A bibliometric review. *Qual. Quant.* **2021**, *56*, 2875–2906. [CrossRef]
56. Yetis, R.; Sahingoz, O.K. Blockchain Based Secure Communication for IoT Devices in Smart Cities. In Proceedings of the 2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), Istanbul, Turkey, 25–26 April 2019; pp. 134–138.
57. Gallo, P.; Pongnumkul, S.; Quoc Nguyen, U. BlockSee: Blockchain for IoT Video Surveillance in Smart Cities. In Proceedings of the 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Palerno, Italy, 12–15 June 2018; pp. 1–6.
58. Botello, J.V.; Mesa, A.P.; Rodríguez, F.A.; Díaz-López, D.; Nespoli, P.; Mármol, F.G. BlockSIEM: Protecting Smart City Services through a Blockchain-based and Distributed SIEM. *Sensors* **2020**, *20*, 4636. [CrossRef]
59. Li, J.; Liu, X.; Zhao, J.; Liang, W.; Guo, L. Application Model of Video Surveillance System Interworking Based on Blockchain. In Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 18–20 June 2021; Volume 4, pp. 1874–1879.
60. Wei, P.; Wang, D.; Zhao, Y.; Tyagi, S.K.S.; Kumar, N. Blockchain data-based cloud data integrity protection mechanism. *Future Gener. Comput. Syst.* **2020**, *102*, 902–911. [CrossRef]
61. Zarour, M.; Alenezi, M.; Ansari, M.T.J.; Pandey, A.K.; Ahmad, M.; Agrawal, A.; Kumar, R.; Khan, R.A. Ensuring data integrity of healthcare information in the era of digital health. *Healthc. Technol. Lett.* **2021**, *8*, 66–77. [CrossRef]
62. Gedara, K.M.; Nguyen, M.; Yan, W.Q. *Visual Blockchain for Intelligent Surveillance in Smart Cities*; IGI: Antwerp, Belgium, 2018.
63. Atlam, H.F.; Azad, M.A.; Alzahrani, A.G.; Wills, G. A Review of Blockchain in Internet of Things and AI. *Big Data Cogn. Comput.* **2020**, *4*, 28. [CrossRef]

64. Nagothu, D.; Xu, R.; Nikouei, S.Y.; Chen, Y. A Microservice-enabled Architecture for Smart Surveillance using Blockchain Technology. In Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas, MO, USA, 16–19 September 2018; pp. 1–4.
65. Alam, T. IBchain: Internet of Things and Blockchain Integration Approach for Secure Communication in Smart Cities. *Informatica* **2021**, *45*, 477–486. [CrossRef]
66. Mishra, P.K.; Saroha, G.P. A Study on Classification for Static and Moving Object in Video Surveillance System. *Int. J. Image Graph. Signal Process.* **2016**, *8*, 76–82. [CrossRef]
67. Chen, K.-H.; Wang, J.-H.; Su, C.-W. An Energy-efficient and Accurate Object Detection Design for Mobile Applications. In Proceedings of the 2022 IEEE International Conference on Consumer Electronics, Taiwan, China, 6 July 2022.
68. Rakibe, R.S.; Patil, B.D. Background subtraction algorithm based human motion detection. *Int. J. Sci. Res. Publ.* **2013**, *3*, 2250–3153.
69. Susheel Kumar, K.; Prasad, S.; Saroj, P.K.; Tripathi, R.C. Multiple Cameras Using Real Time Object Tracking for Surveillance and Security System. In Proceedings of the 2010 3rd International Conference on Emerging Trends in Engineering and Technology, Goa, India, 19–21 November 2010.
70. Huang, H.; Xu, Y.; Huang, Y.; Yang, Q.; Zhou, Z. Pedestrian tracking by learning deep features. *J. Vis. Commun. Image Represent.* **2018**, *57*, 172–175. [CrossRef]
71. Joshi, R.C.; Joshi, M.; Singh, A.G.; Mathur, S. Object Detection, Classification and Tracking Methods for Video Surveillance: A Review. In Proceedings of the 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Nodia, India, 14–15 December 2018; pp. 1–7.
72. Sengar, S.S.; Mukhopadhyay, S. Moving object detection based on frame difference and W4. *Signal Image Video Process.* **2017**, *11*, 1357–1364. [CrossRef]
73. Naik, U.P.; Rajesh, V.; Kumar, R. Implementation of YOLOv4 Algorithm for Multiple Object Detection in Image and Video Dataset using Deep Learning and Artificial Intelligence for Urban Traffic Video Surveillance Application. In Proceedings of the 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 15–17 September 2021; pp. 1–6.
74. Martella, F.; Fazio, M.; Celesti, A.; Lukaj, V.; Quattrocchi, A.; Di Gangi, M.; Villari, M. Federated Edge for Tracking Mobile Targets on Video Surveillance Streams in Smart Cities. In Proceedings of the 2022 IEEE Symposium on Computers and Communications (ISCC), Rhodes Island, Greece, 30 June–3 July 2022; pp. 1–6.
75. Wang, Y.; Zhang, J.; Zhu, L.; Sun, Z.; Lu, J. A Moving Object Detection Scheme based on Video Surveillance for Smart Substation. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 12–16 August 2018; pp. 500–503.
76. Thenmozhi, T.; Kalpana, A.M. Adaptive motion estimation and sequential outline separation based moving object detection in video surveillance system. *Microprocess. Microsyst.* **2020**, *76*, 103084. [CrossRef]
77. Arikumar, K.S.; Deepak Kumar, A.; Gadekallu, T.R.; Prathiba, S.B.; Tamilarasi, K. Real-Time 3D Object Detection and Classification in Autonomous Driving Environment Using 3D LiDAR and Camera Sensors. *Electronics* **2022**, *11*, 4203. [CrossRef]
78. Ammar, S.; Bouwmans, T.; Zaghden, N.; Neji, M. Deep detector classifier (DeepDC) for moving objects segmentation and classification in video surveillance. *IET Image Process.* **2020**, *14*, 1490–1501. [CrossRef]
79. Kunpeng, Y.; Shan, H.; Sun, T.; Hu, R.; Wu, Y.; Yu, L.; Zhang, Z.; Quek, T.Q.S. Reinforcement Learning-based Mobile Edge Computing and Transmission Scheduling for Video Surveillance. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1142–1156. [CrossRef]
80. Zhao, B.; Feng, J.; Wu, X.; Yan, S. A survey on deep learning-based fine-grained object classification and semantic segmentation. *Int. J. Autom. Comput.* **2017**, *14*, 119–135. [CrossRef]
81. Dhiyanesh, B.; Rajkumar, S.; Radha, R. Improved Object Detection in Video Surveillance Using Deep Convolutional Neural Network Learning. In Proceedings of the 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2021), Palladam, India, 11–13 November 2021; pp. 1–8.
82. Deng, Y.; Chen, H.; Li, Y. MVF-Net: A Multi-view Fusion Network for Event-based Object Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 8275–8284. [CrossRef]
83. Manik, F.Y.; Saputra, S.K.; Ginting, D.S.B. Plant Classification Based on Extraction Feature Gray Level Co-Occurrence Matrix Using k-nearest Neighbour. *J. Phys. Conf. Ser.* **2019**, *1566*, 012107. [CrossRef]
84. Bhatti, U.A.; Yu, Z.; Chanussot, J.; Zeeshan, Z.; Yuan, L.; Luo, W.; Nawaz, S.A.; Bhatti, M.A.; Ain, Q.U.; Mehmood, A. Local Similarity-Based Spatial–Spectral Fusion Hyperspectral Image Classification with Deep CNN and Gabor Filtering. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
85. Hossain, E.; Hossain, M.F.; Rahaman, M.A. A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–6.
86. Hashemzadeh, M.; Zademehdi, A. Fire detection for video surveillance applications using ICA K-medoids-based color model and efficient spatio-temporal visual features. *Expert Syst. Appl.* **2019**, *130*, 60–78. [CrossRef]
87. Zhang, M.; Cao, J.; Sahni, Y.; Chen, Q.; Jiang, S.; Yang, L. Blockchain-Based Collaborative Edge Intelligence for Trustworthy and Real-Time Video Surveillance. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1623–1633. [CrossRef]
88. Mangawati, A.; Mohana; Leesan, M.; Aradhya, H.V.R. Object Tracking Algorithms for Video Surveillance Applications. In Proceedings of the International Conference on Communications and Signal Processing, Chennai, India, 3–5 April 2018; pp. 0667–0671.

89. Balaji, S.R.; Karthikeyan, S. A survey on moving object tracking using image processing. In Proceedings of the 2017 11th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 5–6 January 2017; pp. 469–474.
90. Cob-Parro, A.C.; Losada-Gutiérrez, C.; Marrón-Romera, M.; Gardel-Vicente, A.; Bravo-Muñoz, I. Smart Video Surveillance System Based on Edge Computing. *Sensors* **2021**, *21*, 2958. [CrossRef] [PubMed]
91. Elafi, I.; Jedra, M.; Zahid, N. Unsupervised detection and tracking of moving objects for video surveillance applications. *Pattern Recognit. Lett.* **2016**, *84*, 70–77. [CrossRef]
92. Zhu, J.; Lao, Y.; Zheng, Y.F. Object Tracking in Structured Environments for Video Surveillance Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 223–235. [CrossRef]
93. Liu, S.; Liu, D.; Srivastava, G.; Połap, D.; Woźniak, M. Overview and methods of correlation filter algorithms in object tracking. *Complex Intell. Syst.* **2020**, *7*, 1895–1917. [CrossRef]
94. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577. [CrossRef]
95. Xu, S.; Wang, J.; Shou, W.; Ngo, T.; Sadick, A.-M.; Wang, X. Computer vision techniques in construction: A critical review. *Arch. Comput. Methods Eng.* **2021**, *28*, 3383–3397. [CrossRef]
96. Chen, B.-H.; Shi, L.-F.; Ke, X. Low-Rank Representation with Contextual Regularization for Moving Object Detection in Big Surveillance Video Data. In Proceedings of the 2017 IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, CA, USA, 19–21 April 2017; pp. 134–141.
97. Chen, B.-H.; Shi, L.-F.; Ke, X. A Robust Moving Object Detection in Multi-Scenario Big Data for Video Surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 982–995. [CrossRef]
98. Jha, S.; Seo, C.; Yang, E.; Joshi, G.P. Real time object detection and trackingsystem for video surveillance system. *Multimed. Tools Appl.* **2021**, *80*, 3981–3996. [CrossRef]
99. Kothiya, S.V.; Mistree, K.B. A review on real time object tracking in video sequences. In Proceedings of the 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, India, 25 January 2015; pp. 1–4.
100. Kanagamalliga, S.; Vasuki, S. Contour-based object tracking in video scenes through optical flow and gabor features. *Optik* **2018**, *157*, 787–797.
101. Kong, Y.; Fu, Y. Human Action Recognition and Prediction: A Survey. *Int. J. Comput. Vis.* **2022**, *130*, 1366–1401. [CrossRef]
102. Tomar, A.; Kumar, S.; Pant, B. Crowd Analysis in Video Surveillance: A Review. In Proceedings of the 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 23–25 March 2022; pp. 162–168.
103. Andrade-Ambriz, Y.A.; Ledesma, S.; Ibarra-Manzano, M.A.; Oros-Flores, M.I.; Almanza-Ojeda, D.L. Human activity recognition using temporal convolutional neural network architecture. *Expert Syst. Appl.* **2022**, *191*, 116287. [CrossRef]
104. Li, Q.; Lin, W.; Li, J. Human activity recognition using dynamic representation and matching of skeleton feature sequences from RGB-D images. *Signal Process. Image Commun.* **2018**, *68*, 265–272. [CrossRef]
105. Bevilacqua, A.; Macdonald, K.; Rangarej, A.; Widjaya, V.; Caulfield, B.; Kechadi, T. Human Activity Recognition with Convolutional Neural Networks. In *Machine Learning and Knowledge Discovery in Databases*; Springer International Publishing: Wurzburg, Germany, 2019; pp. 541–552.
106. Zhang, S.; Li, Y.; Zhang, S.; Shahabi, F.; Xia, S.; Deng, Y.; Alshurafa, N. Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. *Sensors* **2022**, *22*, 1476. [CrossRef]
107. Zhou, X.; Liang, W.; Wang, K.I.-K.; Wang, H.; Yang, L.T.; Jin, Q. Deep-Learning-Enhanced Human Activity Recognition for Internet of Healthcare Things. *IEEE Internet Things J.* **2020**, *7*, 6429–6438. [CrossRef]
108. Wan, S.; Qi, L.; Xu, X.; Tong, C.; Gu, Z. Deep Learning Models for Real-time Human Activity Recognition with Smartphones. *Mob. Netw. Appl.* **2020**, *25*, 743–755. [CrossRef]
109. Xia, K.; Huang, J.; Wang, H. LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access* **2020**, *8*, 56855–56866. [CrossRef]
110. Islam, A.; Shin, S.Y. BHMUS: Blockchain Based Secure Outdoor Health Monitoring Scheme Using UAV in Smart City. In Proceedings of the 2019 7th International conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, 24–26 July 2019; pp. 1–6.
111. Ko, T. A survey on behavior analysis in video surveillance for homeland security applications. In Proceedings of the 2008 37th IEEE Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 15–18 October 2008; pp. 1–8.
112. Gowsikhaa, D.; Abirami, S.; Baskaran, R. Automated human behavior analysis from surveillance videos: A survey. *Artif. Intell. Rev.* **2014**, *42*, 747–765. [CrossRef]
113. Khalifa, O.O.; Roubleh, A.; Esgiar, A.; Abdelhaq, M.; Alsaqour, R.; Abdalla, A.; Ali, E.S.; Saeed, R. An IoT-Platform-Based Deep Learning System for Human Behavior Recognition in Smart City Monitoring Using the Berkeley MHAD Datasets. *Systems* **2022**, *10*, 177. [CrossRef]
114. Khan, I.U.; Afzal, S.; Lee, J.W. Human Activity Recognition via Hybrid Deep Learning Based Model. *Sensors* **2022**, *22*, 323. [CrossRef]
115. Bilal, M.; Maqsood, M.; Yasmin, S.; Hasan, N.U.; Rho, S. A transfer learning-based efficient spatiotemporal human action recognition framework for long and overlapping action classes. *J. Supercomput.* **2022**, *78*, 2873–2908. [CrossRef]

116. Rajavel, R.; Ravichandran, S.K.; Harimoorthy, K.; Nagappan, P.; Gobichettipalayam, K.R. IoT-based smart healthcare video surveillance system using edge computing. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 3195–3207. [CrossRef]
117. Khan, M.A.; Javed, K.; Khan, S.A.; Saba, T.; Habib, U.; Khan, J.A.; Abbasi, A.A. Human action recognition using fusion of multiview and deep features: An application to video surveillance. *Multimed. Tools Appl.* **2020**, 1–27. [CrossRef]
118. Dahmane, S.; Yagoubi, M.B.; Lorenz, P.; Barka, E.; Lakas, A.; Lagraa, N.; Kerrache, C.A. V2X-based COVID-19 Pandemic Severity Reduction in Smart Cities. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–6.
119. Pramanik, A.; Sarkar, S.; Maiti, J. A real-time video surveillance system for traffic pre-events detection. *Accid. Anal. Prev.* **2021**, *154*, 106019. [CrossRef]
120. Zhou, J.T.; Du, J.; Zhu, H.; Peng, X.; Liu, Y.; Goh, R.S.M. AnomalyNet: An Anomaly Detection Network for Video Surveillance. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2537–2550. [CrossRef]
121. Franklin, R.J.; Dabbagol, V. Anomaly Detection in Videos for Video Surveillance Applications using Neural Networks. In Proceedings of the 2020 Fourth International Conference on Inventive Systems and Control (ICISC), TamilNadu, India, 8–10 January 2020; pp. 632–637.
122. Gayal, B.S.; Patil, S.R. Detecting and localizing the anomalies in video surveillance using deep neuralnetwork with advanced feature descriptor. In Proceedings of the 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), TamilNadu, India, 28–29 January 2022; pp. 1–9.
123. Lu, C.; Shi, J.; Jia, J. Abnormal event detection at 150 fps in matlab. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2720–2727.
124. Hasan, M.; Choi, J.; Neumann, J.; Roy-Chowdhury, A.K.; Davis, L.S. Learning temporal regularity in video sequences. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, LasVegas, NV, USA, 27–30 June 2016; pp. 733–742.
125. Medel, J.R.; Savakis, A. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv* **2016**, arXiv:1612.00390.
126. Nawaratne, R.; Alahakoon, D.; De Silva, D.; Yu, X. Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance. *IEEE Trans. Ind. Inform.* **2020**, *16*, 393–402. [CrossRef]
127. Olmos, R.; Tabik, S.; Lamas, A.; Pérez-Hernández, F.; Herrera, F. A binocular image fusion approach for minimizing false positives in handgun detection with deep learning. *Inf. Fusion* **2019**, *49*, 271–280. [CrossRef]
128. Castillo, A.; Tabik, S.; Pérez, F.; Olmos, R.; Herrera, F. Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning. *Neurocomputing* **2019**, *330*, 151–161. [CrossRef]
129. Ingle, P.Y.; Kim, Y.-G. Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities. *Sensors* **2022**, *22*, 3862. [CrossRef]
130. Fan, X.; Huang, C.; Fu, B.; Wen, S.; Chen, X. UAV-assisted data dissemination in delay-constrained VANETs. *Mob. Inf. Syst.* **2018**, *2018*, 8548301. [CrossRef]
131. Foggia, P.; Saggese, A.; Vento, M. Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556. [CrossRef]
132. Muhammad, K.; Ahmad, J.; Baik, S.W. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* **2018**, *288*, 30–42. [CrossRef]
133. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*, 6519. [CrossRef]
134. Muhammad, K.; Ahmad, J.; Mehmood, I.; Rho, S.; Baik, S.W. Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access* **2018**, *6*, 18174–18183. [CrossRef]
135. Jayamohan, M.; Yuvaraj, S.; Vijayakumar, P. Review of Video Analytics Method for Video Surveillance. In Proceedings of the 2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), Jamshedpur, India, 11–12 February 2022; pp. 43–47.
136. Islam, A.; Sadia, K.; Masuduzzaman, M.; Shin, S.Y. BUMAR: A Blockchain-Empowered UAV-Assisted Smart Surveillance Architecture for Marine Areas. In Proceedings of the 2020 International Conference on Computing Advancements, Dhaka, Bangladesh, 10–12 January 2020; pp. 1–5.
137. Uda, R. Data Protection Method with Blockchain against Fabrication of Video by Surveillance Cameras. In Proceedings of the 2020 The 2nd International Conference on Blockchain Technology, Hilo HI, USA, 12–14 March 2020; pp. 29–33.
138. Kerr, M.; Han, F.; Schyndel, R.V. A Blockchain Implementation for the Cataloguing of CCTV Video Evidence. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zeland, 27–30 November 2018; pp. 1–6.
139. Li, H.; Xiezhang, T.; Yang, C.; Deng, L.; Yi, P. Secure Video Surveillance Framework in Smart City. *Sensors* **2021**, *21*, 4419. [CrossRef]
140. Lee, D.; Park, N. Blockchain based privacy preserving multimedia intelligent video surveillance using secure Merkle tree. *Multimed. Tools Appl.* **2021**, *80*, 34517–34534. [CrossRef]

141. Deepak, K.; Badiger, A.N.; Akshay, J.; Awomi, K.A.; Deepak, G.; Kumar, N.H. Blockchain-based Management of Video Surveillance Systems: A Survey. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), TamilNadu, India, 6–7 March 2020; pp. 1256–1258.
142. Fitwi, A.; Chen, Y.; Zhu, S. A Lightweight Blockchain-Based Privacy Protection for Smart Edge. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Seoul, Republic of Korea, 14–17 March 2019; pp. 552–555.
143. Raj, A.V.B.; Srikanth, B.; Thilagavathy, A.; Mathivanan, B. A Surveillance System Focused On Approved Blockchains and Computation of Edges. *J. Phys. Conf. Ser.* **2020**, *1964*, 042058.
144. Wang, R.; Tsai, W.-T.; He, J.; Liu, C.; Li, Q.; Deng, E. A Video Surveillance System Based on Permissioned Blockchains and Edge Computing. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; pp. 1–6.
145. Jeong, Y.; Hwang, D.; Kim, K.-H. Blockchain-Based Management of Video Surveillance Systems. In Proceedings of the 2019 International Conference on Information Networking (ICOIN), Kuala Lumpur, Malaysia, 9–11 January 2019; pp. 465–468.
146. Tsai, M.H.; Venkatasubramanian, N.; Hsu, C.H. Multi-level feature driven storage management of surveillance videos. *Pervasive Mob. Comput.* **2021**, *76*, 101441. [CrossRef]
147. Dave, M.; Rastogi, V.; Miglani, M.; Saharan, P.; Goyal, N. Smart Fog-Based Video Surveillance with Privacy Preservation based on Blockchain. *Wirel. Pers. Commun.* **2022**, *124*, 1677–1694. [CrossRef]
148. Li, X.; Savkin, A.V. Networked Unmanned Aerial Vehicles for Surveillance and Monitoring: A Survey. *Future Internet* **2021**, *13*, 174. [CrossRef]
149. Nikooghadam, M.; Amintoosi, H.; Islam, S.H.; Moghadam, M.F. A provably secure and lightweight authentication scheme for Internet of Drones for smart city surveillance. *J. Syst. Archit.* **2021**, *115*, 101955. [CrossRef]
150. Yue, X.; Liu, Y.; Wang, J.; Song, H.; Cao, H. Software Defined Radio and Wireless Acoustic Networking for Amateur Drone Surveillance. *IEEE Commun. Mag.* **2018**, *56*, 90–97. [CrossRef]
151. Lykou, G.; Moustakas, D.; Gritzalis, D. Defending Airports from UAS: A Survey on Cyber-Attacks and Counter-Drone Sensing Technologies. *Sensors* **2020**, *20*, 3537. [CrossRef]
152. Castrillo, V.U.; Manco, A.; Pascarella, D.; Gigante, G. A Review of Counter-UAS Technologies for Cooperative Defensive Teams of Drones. *Drones* **2022**, *6*, 65. [CrossRef]
153. Isaac-Medina, B.K.; Poyser, M.; Organisciak, D.; Willcocks, C.G.; Breckon, T.P.; Shum, H.P. Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1223–1232.
154. Souli, N.; Kolios, P.; Ellinas, G. An Autonomous Counter-Drone System with Jamming and Relative Positioning Capabilities. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 5110–5115.
155. Kumar, V.S.; Sakthivel, M.; Karras, D.A.; Gupta, S.K.; Gangadharan, S.M.P.; Haralayya, B. Drone Surveillance in Flood Affected Areas using Firefly Algorithm. In Proceedings of the 2022 International Conference on Knowledge Engineering and Communication Systems (ICKES), Chickballapur, India, 28–29 December 2022; pp. 1–5.
156. Kumar, A.; Sharma, K.; Singh, H.; Naugriya, S.G.; Gill, S.S.; Buyya, R. A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic. *Future Gener. Comput. Syst.* **2021**, *115*, 1–19. [CrossRef]
157. Chen, K.W.; Xie, M.R.; Chen, Y.M.; Chu, T.T.; Lin, Y.B. DroneTalk: An Internet-of-Things-Based Drone System for Last-Mile Drone Delivery. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15204–15217. [CrossRef]
158. Ruichek, Y. Attractive-and-repulsive center-symmetric local binary patterns for texture classification. *Eng. Appl. Artif. Intell.* **2019**, *78*, 158–172.
159. Shi, H.; Ghahremannezhad, H.; Liu, C. Unsupervised Anomaly Detection in Traffic Surveillance Based on Global Foreground Modeling. In Proceedings of the 2022 IEEE International Conference on Imaging Systems and Techniques (IST), New York, NY, USA, 21–23 June 2022; pp. 1–6.
160. Gamage, C.; Dinalankara, R.; Samarabandu, J.; Subasinghe, A. A comprehensive survey on the applications of machine learning techniques on maritime surveillance to detect abnormal maritime vessel behaviors. *WMU J. Marit. Aff.* **2023**, *22*. [CrossRef]
161. Olmos, R.; Tabik, S.; Perez-Hernandez, F.; Lamas, A.; Herrera, F. MULTICAST: MULTI Confirmation-level Alarm SysTem based on CNN and LSTM to mitigate false alarms for handgun detection in video-surveillance. *arXiv* **2021**, arXiv:2104.11653.
162. Allaoui, T.; Jeridi, M.H.; Ezzedine, T. False Alarm Reduction in WSN Surveillance Application through ML techniques. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023.
163. Zhang, X.; Yu, Q.; Yu, H. Physics Inspired Methods for Crowd Video Surveillance and Analysis: A Survey. *IEEE Access* **2018**, *6*, 66816–66830. [CrossRef]
164. Azam, Z.; Islam, M.M.; Huda, M.N. Comparative Analysis of Intrusion Detection Systems and Machine Learning Based Model Analysis through Decision Tree. *IEEE Access* **2023**, *11*, 80348–80391. [CrossRef]
165. Liang, P.P.; Zadeh, A.; Morency, L.P. Foundations and Recent Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions. *arXiv* **2022**, arXiv:2209.03430.
166. Hafeez, S.; Alotaibi, S.S.; Alazeb, A.; Al Mudawi, N.; Kim, W. Multi-sensor-based Action Monitoring and Recognition via Hybrid Descriptors and Logistic Regression. *IEEE Access* **2023**, *11*, 48145–48157. [CrossRef]

167. Javeed, M.; Mudawi, N.A.; Alabduallah, B.I.; Jalal, A.; Kim, W. A Multimodal IoT-Based Locomotion Classification System Using Features Engineering and Recursive Neural Network. *Sensors* **2023**, *23*, 4716. [CrossRef]
168. Vikram, R.; Sinha, D. A multimodal framework for Forest fire detection and monitoring. *Multimed. Tools Appl.* **2023**, *82*, 9819–9842. [CrossRef]
169. Alladi, T.; Chamola, V.; Sahu, N.; Guizani, M. Applications of blockchain in unmanned aerial vehicles: A review. *Veh. Commun.* **2020**, *23*, 100249. [CrossRef]