IoT

MDPI

Review

Internet-of-Things Edge Computing Systems for Streaming Video Analytics: Trails Behind and the Paths Ahead

Arun A. Ravindran

Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223, USA; arun.ravindran@charlotte.edu

Abstract: The falling cost of IoT cameras, the advancement of AI-based computer vision algorithms, and powerful hardware accelerators for deep learning have enabled the widespread deployment of surveillance cameras with the ability to automatically analyze streaming video feeds to detect events of interest. While streaming video analytics is currently largely performed in the cloud, edge computing has emerged as a pivotal component due to its advantages of low latency, reduced bandwidth, and enhanced privacy. However, a distinct gap persists between state-of-the-art computer vision algorithms and the successful practical implementation of edge-based streaming video analytics systems. This paper presents a comprehensive review of more than 30 research papers published over the last 6 years on IoT edge streaming video analytics (IE-SVA) systems. The papers are analyzed across 17 distinct dimensions. Unlike prior reviews, we examine each system holistically, identifying their strengths and weaknesses in diverse implementations. Our findings suggest that certain critical topics necessary for the practical realization of IE-SVA systems are not sufficiently addressed in current research. Based on these observations, we propose research trajectories across short-, medium-, and long-term horizons. Additionally, we explore trending topics in other computing areas that can significantly impact the evolution of IE-SVA systems.

Keywords: video analytics; edge computing; streaming video; systems; deep learning; AI; latency; bandwidth; privacy

Citation: Ravindran, A.A. Internet-of-Things Edge Computing Systems for Streaming Video Analytics: Trails Behind and the Paths Ahead. *IoT* **2023**, *4*, 486–513. https://doi.org/10.3390/iot4040021

Academic Editor: Amiya Nayak

Received: 1 August 2023
Revised: 18 October 2023
Accepted: 20 October 2023
Published: 24 October 2023

1. Introduction

The falling cost of IoT video cameras [1], and the increasing capability of deep-learning-based computer vision algorithms [2,3] makes it possible to use streaming video analytics (SVA) to visually sense the environment. SVA refers to the real-time or near real-time processing of video streams to determine events in the environment as they happen. Figure 1 shows an example application of SVA in the use of traffic surveillance video cameras to detect and alert pedestrians and drivers of dangerous situations as they occur [4]. In contrast, in batch video analytics, the processing of stored videos may happen at a much later time. For example, traffic engineers may analyze stored traffic video streams to identify congestion patterns for the purpose of roadway planning. From a computing perspective, video analytics is challenging due to the large sizes of the data involved and the computation-intensive algorithms needed to process it [5]. A number of other use cases of streaming video analytics exist in multiple areas including healthcare, manufacturing, environmental monitoring, and national security [6].

1.1. Need for Edge Computing

The computation could potentially be realized on the cloud by taking advantage of powerful and scalable computation resources that are available on-demand. However, bandwidth, latency, and privacy necessitate the use of the edge computing paradigm for streaming video analytics [7,8]. To put this into context, a single H.265 1080p IP camera with a high video quality operating at 15 fps requires a 2.3 Mbps uplink and generates over

25 GB of data per day [9]. Half a dozen cameras in a home or at a traffic intersection could easily saturate the local uplink capacity (typically 10 Mbps). Regarding latency, the camera-to-cloud communication over the Internet is of the order of hundreds of milliseconds. For the pedestrian safety use case, a vehicle driving at a speed of 45 mph (72 kph) covers a distance of 60 ft (20 m) in a second. Detecting whether such a moving vehicle poses a threat to a pedestrian requires detecting events with tens of milliseconds of latency including computation and communication overheads. Regarding privacy, video streams are a rich source of information. Transmitting videos to a distant cloud data center could violate user expectations of privacy, and legal requirements such as GDPR [10] and the guiding principles of the government use of surveillance technologies [11]. Furthermore, if the video stream is accessed by unauthorized third parties, unintended information (for example, personal identities for the pedestrian safety use case) could be revealed. By performing video analytics at the edge close to the video cameras, communication is limited to local area networks, thus reducing the network latency. Furthermore, the aggregate bandwidth requirements are reduced due to the distributed processing at the edge enabling the scaling of video cameras. Moreover, sensitive video data can be confined to the privacy perimeter expected by the user (for example, a home or legal jurisdiction).

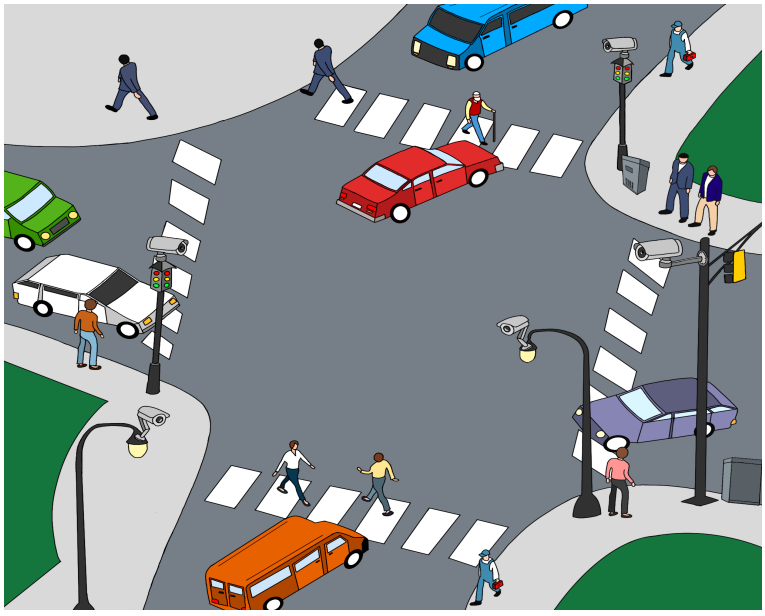

Figure 1. A smart-city traffic intersection equipped with surveillance cameras. Many city departments could consume the camera feeds for multiple applications including traffic monitoring, pedestrian safety, detecting traffic law violations, public safety, and environmental monitoring. Each application may require a separate video analytic pipeline (VAP), with the possibility of sharing VAP components between the applications. The processing of video streams is implemented on edge nodes including the cameras, and on nearby edge servers such as in the traffic box.

1.2. Key Contributions

In recent years, there has been a notable surge in research focused on SVA at the edge. Despite the widespread deployment of cameras in cities (for example, London, UK, has more than half a million IoT surveillance cameras) and private establishments, as well as the significant advancements in deep learning and AI-powered computer vision, a substantial gap still persists in effectively utilizing AI to analyze these camera streams in real-time to derive actionable insights. The research question that this paper seeks to answer is: what is the state-of-the-art in IoT edge streaming video analytics systems (IE-SVA)? The goal of this paper is to thoroughly analyze IE-SVA systems as reported in the research literature, so as to provide clarity to researchers and industry practitioners on the progress to date,

the techniques employed, and the additional research and development that needs to be performed to further the field.

To achieve these goals, we begin by outlining the characteristics of an ideal IE-SVA system. By using this ideal system as a framework, we analyzed existing research literature on video analytics along 17 unique dimensions. The analysis is presented in tabular format with 37 reported works listed in chronological order (2015–2023) to help the reader readily understand the research progress made by these systems along different dimensions. These systems are also classified according to their primary research focus, allowing readers to easily access works that delve into specific techniques aligned with their interests. Based on this analysis, we propose research directions for the short, medium, and long term.

In the short term (3 years), our proposed research aims to build upon existing work by incorporating different techniques reported in the literature in a straightforward manner so as to advance the state-of-the-art on end-to-end edge systems for video analytics. The medium-term research proposals (5 years) outline new areas of study that the community needs to undertake in order to address system-level challenges identified in the review. Looking ahead, the long-term research plan (10 years) envisions the future evolution of streaming video analytics at the edge and proposes exploratory research directions.

While experienced researchers may benefit from our comprehensive treatment of IE-SVA systems, this review specifically aims to provide new researchers with a broad background, and a timeline-based overview of how IE-SVA systems have developed over the years. In particular, Section 6 describes multiple techniques covering nine specific issues for implementing IoT edge streaming video analytics systems with research gaps identified that can be pursued in the short term.

1.3. Paper Organization

This paper is organized as follows. Section 2 describes related work, emphasizing how this review contributes a distinct perspective to the few available reviews on this topic. Section 3 lays out the necessary background on topics such as streaming video analytics, components of IoT edge systems, and the challenges of using edge computing for streaming video analytics. In Section 4, we outline the optimal system requirements for edge computing in the context of streaming video analytics. Section 5 offers a critical analysis of previously documented IE-SVA systems. Section 6 then drills down on the specific techniques with case studies from the reported work. Section 7 then outlines our research vision, breaking it down into short-term, medium-term, and long-term objectives. Section 8 briefly examines how other advancements in computing might impact systems designed for video analytics at the edge. Finally, in Section 9, we wrap up with a summary of the paper.

2. Related Work

The studies closest to ours are the surveys on deep-learning-driven edge video analytics by Xu et al. [12] and edge-based video analytics by Hu et al. [13]. These surveys catalog various edge video analytics systems from the literature, categorizing them based on a range of criteria including edge architectures, profiling, algorithms, scheduling, orchestration framework, and aspects of privacy and security. As a result, they often refer to the same system multiple times under different criteria. While this approach offers the advantage of providing an extensive perspective on the techniques used, its disadvantage is that it can make it challenging to understand how these techniques are integrated into an end-to-end system.

Other related reviews include Goudarzi et al.'s [14] survey which analyzes scheduling IoT applications in edge and fog computing environments. They study factors such as the application structure, environmental architecture, optimization characteristics, scheduling framework, and performance evaluation to identify research gaps. However, their focus is limited to scheduling. Zhang et al.'s [6] survey focuses on edge video analytics specifically for public safety. While some overlap exists with our work, they mainly examine algorithms

and systems from a public safety perspective. Other surveys related to edge computing or video analytics are limited in scope to their respective areas [15–17].

In contrast, our work adopts a more systems-oriented perspective with an emphasis on issues regarding the practical deployment of IE-SVA systems. We provide a comprehensive overview of 37 recently reported systems for edge video analytics, analyzing each system along 17 different dimensions. The information is presented in a tabular format allowing readers to readily grasp the evolution of edge video analytics systems, the mix of techniques employed in a particular system, and the areas that have received limited attention from the research community to date. Additionally, we focus on specific techniques, categorizing them by the problems they address. For each technique, we briefly dive into reported works, which in our opinion, provide good illustrations of the highlighted techniques. Furthermore, we briefly review research developments in other areas of computing that may impact edge video analytics. Based on this comprehensive analysis, we make concrete proposals for short-term, medium-term, and long-term research on edge video analytics systems. As a limitation, our review does not comprehensively cover all the reported work on IE-SVA systems. Instead, we selectively chose the works to review in order to make the content more manageable for readers.

3. Background

In this section, we provide a brief background of streaming video analytics, hardware and software system components that are needed to realize video analytics at the edge, and the challenges in implementing these systems. References to the more detailed tutorial-like treatment of these topics is provided.

3.1. Streaming Video Analytics

Streaming video analytics involves the real-time processing of video frames to extract valuable information [18,19]. For instance, IoT surveillance camera streams at traffic intersections can be utilized to detect and alert pedestrians about potential hazards. It is worth noting that streaming video analytics operates within specific time constraints. In contrast, in batch video analytics, video frames are stored and queried later for useful information. For example, traffic engineers can analyze months' worth of stored traffic video streams to understand traffic patterns. In addition to generating real-time actionable insights, streaming video analytics also offers the advantage of reducing data storage requirements. By storing only the object type and position instead of the entire video, significant storage savings can be achieved. Moreover, videos contain abundant information that can potentially compromise privacy and confidentiality. Through streaming video analytics, only the relevant information needed for the specific application is extracted, allowing the videos themselves to be discarded.

Video analytics typically involves a series of operations organized as a directed acyclic graph (DAG). These operations typically include video decompression, frame pre-processing, object detection, object classification, image segmentation, object tracking, pose estimation, and action classification [20] . Algorithm 1 [21] describes an example of a video analytics pipeline (VAP) for recognizing activity detection. The different stages utilize computationally intensive deep learning algorithms. Additionally, multiple deep learning algorithms with varying performance–accuracy trade-offs exist for each operation. For instance, object detection can be accomplished using a more accurate but slower two-stage detector like Fast-RCNN or Mask-RCNN, or a faster but less accurate single-stage detector such as YOLO or CenterNet. The Tensorflow model zoo has over 40 models for object detection [22]. Moreover, each implementation offers adjustable parameters like bit-rate, frame rate, and resolution. Consequently, a single VAP can have numerous implementations with different performance–resource trade-offs. Furthermore, the performance of these operations is heavily influenced by the content of the video scene. Techniques aimed at enhancing computational efficiency, such as adjusting the decoding bit rate, dropping frames,

or filtering specific parts of video frames, impact both the application's requirements and computational resources [23,24].

Algorithm 1: High-level Activity Detection Pipeline Pseudocode [21]

```
Data: videoStream
Result: activityClasses
procedure MAINPIPELINE
    Call PROPOPOSALGENERATION
    Call SPATIOTEMPORALCLASSFICATION
    Call POSTPROCESS
end procedure
procedure PROPOPOSALGENERATION
    proposals ← objectDetection_objectTracking_proposalGeneration()
    return proposals
end procedure
procedure SPATIOTEMPORALCLASSFICATION
    activityBoundaries ←
featureExtraction_ActivityClassification_SceneDetection(proposals)
    return activityBoundaries
end procedure
procedure POSTPROCESS
    activityClasses ← filter_fusion(activityBoundaries)
    return output
end procedure
```

3.2. Application Use Cases

This section provides a brief overview of the various applications of streaming video analytics across diverse industries. It is worth noting that each of these application areas poses domain-specific constraints that influence the system design.

Transportation: Autonomous vehicles leverage multiple video cameras for environmental perception. Given the stringent latency requirements, these video streams are processed in real-time within the vehicle itself [25]. Intelligent roadways use streaming video analysis to alert both the drivers and pedestrians of potentially hazardous situations [26,27].

Public safety: The public safety sector benefits from the automatic analysis of surveillance camera feeds to detect incidents, such as crime and assault [6]. Additional applications include crowd monitoring to avoid dangerous overcrowding [28] and the early detection of fire incidents [29]. However, this application poses concerns about potential biases against certain demographic groups [30] and potential privacy infringements [31].

Healthcare: The healthcare sector employs video analytics for continuous patient monitoring in healthcare and homecare facilities, enabling the immediate notification of healthcare personnel in the event of incidents like falls [32].

Environmental monitoring: Environmental monitoring uses video analytics for tracking events such as wildfires, flash floods, illegal dumping, and wildlife movement [33]. As climate change challenges intensify, visual environmental sensing to drive intelligent interventions will become increasingly crucial.

Industrial applications: On factory floors, video analytics serve to monitor worker and site safety [34] and assembly line efficiency [35].

Retail: In retail settings, video analytics are used to monitor customer behavior in stores, gauge customer interest, and effectively deploy store staff for customer assistance. Other applications include automated checkouts and inventory management [36].

Search and rescue: The ability to deploy drones to locate individuals needing rescue during adverse events like floods, earthquakes, and wildfires is greatly enhanced by performing streaming analytics on drone footage [37,38].

National security: National security applications encompass drone-based battlefield surveillance, perimeter monitoring, and personnel search and rescue operations [39].

Augmented and virtual reality (AR/VR): The demanding latency requirements for AR/VR applications necessitate the processing of camera streams from headsets at the edge, with results relayed back to the headset for visual display [40].

Robotics: Robots, in their operation, employ multiple cameras to perform onboard streaming video analytics, thereby enriching their perception of the environment [41].

3.3. Edge System Components

3.3.1. System Architecture

In a broad sense, edge computing refers to any computing performed on the edge of the network. As shown in Figure 2, the IoT edge streaming video analytics (IE-SVA) hardware hierarchy forms a tree structure with IoT cameras at the leaf nodes; single board computers on a local area network (LAN) are shared with the IoT cameras at the next higher level; workstations on the LAN at a level higher, micro data centers on a wide-area network (WAN) at a level further up; and finally public/private cloud at the root node. Multiple such edge hierarchies could be geo-distributed to cover a large area. Within this hierarchical structure, computing could be organized vertically with control flowing from top to bottom, and data flowing from bottom to top of the edge tree. Alternatively, computing could be organized in a hybrid fashion, where computing is organized vertically within a subtree, and horizontally (peer-to-peer) across subtrees. Furthermore, the individual nodes in the tree could be stationary or could be mobile (for example, a drone-based camera). Also, not all levels may not be present in a particular implementation. In others, additional levels of computation may be added to the tree structure as the system scales. End users connect to the system via web-based or mobile application frontends.

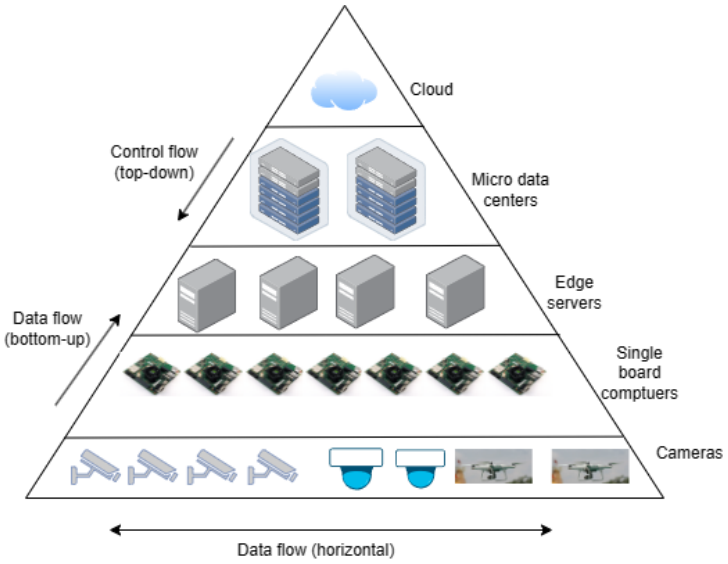

Figure 2. IE-SVA systems hierarchy. Aside from the cameras, not all levels need be present in an implementation. The computation nodes may be organized as clusters for availability and fault tolerance. Data flow vertically up the hierarchy starting from the cameras, while the control flows down the hierarchy. In some implementations, data may flow horizontally as well.

3.3.2. Hardware

Hardware components comprising IE-SVA systems include wired and wireless video cameras (equipped with or without onboard processing), a low-power edge computing cluster consisting of single-board computers (SBCs) incorporating embedded GPUs and TPUs, workstations equipped with consumer grade GPUs, and microdata centers housing server-class machines, powerful GPUs, FPGAs, TPUs, and a public cloud backend with

virtually limitless computing resources. Storage options range from SD cards on SBCs, to SSDs on workstations, to networked storage in the microdata center, and storage-as-a-service (for example, AWS S3 object store) in the cloud. Networking options range from wireless networks (WiFi, 4G, 5G) to wired networks (Ethernet, optical).

Cloud computing typically involves large data centers typically equipped with racks of homogeneous servers connected by high-speed networks (25–100 Gbps). In contrast, edge computing hardware is highly heterogeneous, connected by less reliable networks at lower speeds (1 Mbps–1 Gbps). Additionally, unlike the dedicated cloud data centers, edge resources are housed in a variety of locations from weather proof cases in the proximity of outdoor cameras to small server rooms in office buildings.

3.3.3. System Software Stack

Cloud computing entails a vast distributed system consisting of numerous servers organized in data centers. Over the past 15 years, an extensively developed open source software stack has aimed to simplify the programming and management of these large-scale distributed systems. However, it is important to note that the cloud software stack is specifically designed for a homogeneous distributed system. It assumes that there are high-speed, reliable networks connecting the system and a controlled environment within a data center. As large-scale edge computing is still in its early stages, initial attempts have involved using cloud system software at the edge. Hence, we will provide a brief overview of the cloud software stack. A comprehensive treatment of cloud system software is provided by [42–44].

Cloud computing has widely adopted the infrastructure-as-a-service (IaaS) paradigm, involving the virtualization of the physical computation, storage, and network. Users are able to provision these virtualized resources in a pay-as-you-go fashion, with the ability to scale the resources up and down as needed. Cloud applications typically adopt a microservice architecture, where different services primarily communicate via a REST or RPC API. The loose coupling of services enables rapid code iteration and scalability in deployments. Microservices are packaged within lightweight OS-level virtual machines known as containers, with Docker [45] being a widely used container implementation. Orchestrating these containers is Kubernetes [46], which currently serves as the de facto cloud operating system. Asynchronous communication between services is achieved through distributed broker-based messaging systems, with Kafka [47], NATS [48], and RabbitMQ [49] being prominent open source solutions. For data storage, open source options include both SQL (for example, Postgres, MySQL) and NoSQL (for example, MongoDB, Cassandra) solutions, each offering various architectures for distributed storage [50]. In recent years, serverless architectures have gained popularity, where application programmers focus solely on the business logic while the cloud provider handles server deployment [44]. Examples of the serverless paradigm include function-as-a-service and backend-as-a-service. Many of the computation, storage, and service offerings by cloud vendors today adhere to this paradigm. For instance, Amazon AWS S3 object storage seamlessly scales its capacity to accommodate any amount of data without users needing to worry about infrastructure provisioning. A comprehensive review by Schleier-Smith et al. [44] explores the state-of-the-art in serverless computing and argues that this paradigm will dominate cloud computing in the next decade. While Kubernetes has been adapted for edge computing through projects like K3s [51] supported by commercial vendors, much of the other system software native to the edge remain confined to research publications [52–55].

3.4. Edge Computing Challenges

Edge computing offers several advantages over cloud computing, including reduced latency, lower bandwidth requirements, and the ability to keep data within privacy boundaries. However, along with these benefits come a unique set of challenges. One of the primary challenges is the limited resources available at the edge.

Cloud companies have large-scale data centers equipped with thousands of servers, massive storage capacities, high-speed networks, and dedicated operational and maintenance teams. In contrast, edge computing takes place in various settings, ranging from server rooms in office buildings or cellular base stations to more constrained environments such as a traffic box at a roadway intersection. The hardware used at the edge is also diverse, ranging from comprehensive cloud-in-a-box solutions like AWS Outposts [56] to smaller clusters of low-cost single-board computers like Nvidia Jetson and Raspberry Pi. Moreover, as mentioned earlier, unlike the high-speed low-latency networks on the cloud, the edge employs a variety of networks including wired local/wide/metropolitan area networks (LANs, WANs, MANs), and wireless networks including WiFi, 4G/LTE, and 5G.

These resource limitations pose obstacles to implementing streaming video analytics applications at the edge. For instance, supporting multiple complex deep-learning-based computer vision pipelines may simply not be feasible due to resource constraints. Furthermore, from a system software perspective, these limitations make it challenging to employ the traditional cloud strategy of ensuring reliability through replication. Additionally, the hardware heterogeneity across different edge platforms makes it difficult to develop a single solution that can be applied universally.

In the context of streaming video analytics, a resource-intensive deep learning model that can efficiently execute on a cloud-in-a-box setup may experience high latency or fail to execute altogether on a single-board computer. Another challenge at the edge is the issue of shared ownership. In a smart-city scenario, for example, an IE-SVA platform, including IoT cameras, may be owned by state transportation departments. Various city departments such as traffic management, law enforcement, and environmental monitoring share their utilization of the cameras for their specific applications. Each application may have significantly different performance requirements, and some may even need to actively control the cameras (e.g., pan, zoom, and tilt) [5] for accurate event detection.

Securing the system from external attackers and malicious users is challenging at the edge as compared to the cloud because of the diversity of hardware and software systems, the limited physical security of the computation equipment, and the limited engineering experience of operational teams associated with end-users such as cities, community organizations, and small businesses. From a budgetary point of view, unlike the cloud infrastructure maintained by companies with considerable resources, community-owned and non-profit edge infrastructures are often subject to tight budgetary constraints. Finally, from an environmental perspective, minimizing the energy use, the incorporation of sustainable energy sources, and maximizing as well as extending the life of equipment are important considerations.

4. Ideal System Requirements for IoT Edge Streaming Video Analytics

In this section, we present our view of the ideal characteristics of IoT edge streaming video analytics IE-SVA systems designed for video analytics. These are grouped by hardware, application support, operational ease, human factors, and sustainability.

4.1. Resource Heterogeneity

Ideally, the edge system supports different types of IoT surveillance cameras, each with its own unique specifications such as resolution, connection capabilities, and processing capability. The system allows for the number of deployed cameras to grow as needed. Moreover, the system possesses the ability to virtualize these cameras, exposing logical camera streams to applications and thus allowing for flexibility and scalability in their deployment. Additionally, the system is designed to take advantage of a wide variety of distributed computation resources, including single board computers, edge servers, micro data centers, private and public clouds. These computation resources are heterogeneous, supporting multiple processor, accelerator, network, and storage architectures.

4.2. Application Support

The system is ideally designed for multiple organizations to deploy numerous video analytics applications, using shared resources such as camera feeds, deep learning models, and intermediate processing results. This approach supports the camera-as-a-service concept [57], separating camera users from owners to ensure the maximum utilization of the deployed hardware. It also fosters a rich ecosystem of applications.

Moreover, the system is ideally built to keep up with advancements in video analytics algorithms. It offers a mechanism to upgrade all or certain parts of existing applications with newer models, catering to the evolving demands of video analytics. For applications utilizing multiple cameras, the system facilitates localized cooperation between the cameras to enhance performance. The deployment framework provided by the system is both forward and backward compatible with video analytics applications. This feature further enriches its usability, making it a robust and adaptable platform for future advancements.

4.3. Operational Ease

The efficient operation of the IE-SVA system is critical to its success. The system is designed to be easily managed without the need for a sophisticated operational team. It incorporates built-in redundancy and fault tolerance mechanisms, enabling certain parts of the system to operate autonomously in case of failures. Moreover, the system possesses the ability for operators to perform root cause analysis in the event of a failure, allowing for the swift identification and resolution of issues. It is also built with a secure-by-design approach, promptly reporting security events and facilitating the isolation of compromised parts of the system. Furthermore, the modular nature of the system encourages innovation at every level of the stack, fostering continuous improvement and adaptability to dynamic changes in resource availability, and application requirements.

4.4. User Friendliness for End Users

The system prioritizes ease of use, ensuring that non-technical users can easily interact with the system, particularly when submitting queries or accessing relevant information. Additionally, the system provides multiple logical abstractions allowing application developers to choose the abstractions appropriate for their use case.

4.5. Sustainability

The sustainability of widely deployed edge systems for video analytics is both an economic and societal concern. This revolves around optimizing the system's utilization, minimizing power consumption, and tapping into renewable and possibly intermittent energy sources. It also includes maximizing the use of sunk costs, which allows for the operation of deployed hardware for many years. Fully utilizing the system's capabilities ensures that resources are efficiently employed, thus reducing e-waste and unnecessary expenditure.

5. Reported Systems for IoT Edge Streaming Video Analytics

In this section, we analyze research reported in the literature for realizing IoT edge streaming video analytics (IE-SVA) systems. Importantly, we exclude works solely dedicated to video analytics algorithms, or those involving cloud-only video analytics. Cloud video analytics has a longer history [19], and early works such as Gabriel [58] and Vigil [59] involved the use of video processing at the edge. However, the 2017 IEEE Computer article titled "Real-Time Video Analytics: The Killer App for Edge Computing" by Ananthanarayanan et al. [5] from Microsoft Research was influential in popularizing research on IE-SVA systems. In their article, the authors make a case of why edge computing is essential for streaming video analytics, and sketch out the core technologies needed to realize such a system. Their application emphasis is on live traffic analytics. Since their initial paper, the Microsoft Research group has been active in publishing research articles on this topic, including the release of the open source Microsoft Rocket for live video analytics [60]. Rocket is, however, tied to Microsoft products such as the Azure cloud and IoT platforms.

Tables 1 and 2 show research projects reported in the literature on IE-SVA systems from 2015 to 2023. Thirty-seven papers were analyzed in chronological order along 17 different criteria. The papers were selected for their capacity to illuminate various facets of IE-SVA. Our goal is to provide the reader with a sense of the historical evolution of IE-SVA systems, the multiple system design aspects considered in these works, and to highlight topics that have not yet received sufficient attention from the research community. A more detailed analysis of these works is presented Section 6.

Table 1. Literature review covering criteria 1–6.

Project (Year)	Focus	Perf. Obj.	Cross Cam.	VAP Compon.	Profile	Arch.
Vigil (2015) [59]	ECB	MaxBW	Yes	FDR	No	EC
Glimpse (2015) [61]	MEB	MaxAcc	No	FDR	No	ME
VideoStorm (2017) [62]	PS	MaxAcc-MinLat	No	BS, OD, OT	OFF	DE, EC
OpenFace (2017) [63]	SP	MaxPriv	No	FDR	No	DE
Lavea (2017) [64]	PS	MinLat	No	CR	OFF	DE, HE
VideoEdge (2018) [65]	CE, PS	MaxAcc-ResCon	No	OC, CR	OFF	DE, EC
AWStream (2018) [66]	ECB	MaxAcc-ResCon	No	OD, OC	OFF	EC
Chameleon (2018) [67]	CE	MaxAcc-ResCon	Yes	OD, OC	ON	SE
Wang et al. (2018) [68]	ECB	MinBW-MaxAcc	No	OD, OC	No	HE
EdgeEye (2018) [69]	SS	MaxTh	No	OD	No	SE
VideoPipe (2019) [70]	SS	MinLat	No	PD, AR	No	DE
FilterForward (2019) [71]	ECB	MinBW-MaxAcc	No	OC	ON	EC
DeepQuery (2019) [72]	CE	MinLat	No	OD, OT, SD, VD	ON	ME
Couper (2019) [73]	SS	UD	No	OC	UD	EC
HeteroEdge (2019) [74]	SS	MinLat	No	3DR	OFF	DE
Liu et al. (2019) [75]	ECB	MaxAcc-MinLat	No	OD	No	ME
Marlin (2019) [76]	EE	MaxAcc-MinPow	No	OD, OT	No	ME
DiStream (2020) [77]	PS	MaxTh	No	BS, OD	OFF	DE, HE
VU (2020) [78]	FT	MinBW-MaxAcc	No	LIP	OFF	EC
Clownfish (2020) [79]	ECB	MinBW-MaxAcc	No	AR	ON	EC
Spatula (2020) [80]	MC	MinBW-MaxAcc	No	OD, RID	OFF	HE
REVAMP^{2}T [81]	EE	MinPow-MaxAcc	No	PD, RID, OT	No	HE
Anveshak (2021) [82]	MC	MinBW-PerfCon	Yes	OD, RID, OT	No	HE, DE, EC
Jang et al. (2021) [83]	SS	MinLat-MaxAcc	No	OD, RID, OT	No	DE
OneEdge (2021) [84]	SS	MinLat	No	OD, OT	OFF	HE, DE, EC
Yoda (2021) [85]	CE	MaxAcc	No	OD	OFF	N/A
PECAM (2021) [86]	SP	MaxPriv-MaxAcc	Yes	GS	N/A	N/A
DeepRT (2021) [87]	CE	MaxTh-ResCon	No	OD	Yes	SE
CASVA (2022) [88]	ECB	MaxAcc-MinLat	No	OD, SS	OFF	HE
MicroEdge (2022) [89]	CE	MaxTh	No	OD, OT, PS	OFF	HE, DE
EdgeDuet (2022) [90]	ECB	MaxAcc-MinLat	No	OD	ON	EC
Ekya (2022) [91]	CE	MaxAcc	No	OC, OD	OFF	SE
Gemel (2023) [92]	CE	MinMem-MacAcc	No	M-DNN	OFF	EC
RECL (2023) [93]	CE	MaxAcc	No	OC, OD	ON	HE
Runespoor (2023) [41]	ECB	MaxAcc-MinBW	No	SS, OD	OFF	EC
REACT (2023) [94]	ECB	MaxAcc	No	OD	No	EC
RL-CamSleep (2023) [95]	EE	MinPow	No	OD	No	EC

Focus —ECB: edge cloud bandwidth; MEB: mobile edge bandwidth ; CE: computational efficiency; PS: placement and scheduling; SS: system software; MC: multi-camera; FT: fault tolerance; SP: security and privacy; EE: energy efficiency. Performance objectives—MaxAcc: maximize accuracy; MinLat: minimize latency; MinBW: minimize bandwidth; MaxTh: maximize throughput; MinPow: minimize power; MaxPriv: maximize privacy; ResCon: resource constraint; PerfCon: performance constraint; MinMem: minimize memory. VAP components—FDR: face detection and re-identification; OD: object detection; OT: object tracking; OC: object classifier; PD: pose detection; AR: activity recognition; BS: background subtraction; CR: character recognition; SD: scene detection; VD: video description; 3DR: 3D reconstruction; LIP: lightweight image processing; RID: re-identification; GS: GAN-based steganography; SS: semantic segmentation; MDNN: multiple-DNN backbone. VAP components—DE: distributed edge; EC: edge cloud; HE: hierarchical edge; SE: single-edge; ME: mobile edge.

Table 2. Literature review covering criteria 7–17.

Project (Year)	Sched.	Run-Time Adapt.	Ctrl. Plane	Data Plane	UI	Security	Privacy	Fault Tol.	Obsv.	Sust.	Testbed
Vigil (2015) [59]	HP	No	No	No	No	No	No	No	No	No	EXP, SIM
Glimpse (2015) [61]	No	No	No	No	No	No	No	No	No	No	EXP
VideoStorm (2017) [62]	HP	Yes	No	No	No	No	No	No	No	No	EMU
OpenFace (2017) [63]	No	No	No	No	No	No	Yes	No	No	No	EXP
Lavea (2017) [64]	MILP	Yes	No	Yes	No	No	No	No	No	No	EXP
VideoEdge (2018) [65]	BILP	Yes	No	No	No	No	No	No	No	No	EMU
AWStream (2018) [66]	HP	Yes	No	No	No	No	No	No	No	No	EMU
Chameleon (2018) [67]	N/A	Yes	No	No	No	No	No	No	No	No	EXP
Wang et al. (2018) [68]	No	Yes	No	No	No	No	No	No	No	No	EMU
EdgeEye (2018) [69]	N/A	No	Yes	Yes	Yes	No	No	No	No	No	EXP
VideoPipe (2019) [70]	No	No	No	Yes	No	No	No	No	No	No	EXP
FilterForward (2019) [71]	No	Yes	No	No	No	No	No	No	No	No	EXP
DeepQuery (2019) [72]	HP	Yes	Yes	Yes	No	No	No	No	No	No	EXP
Couper (2019) [73]	UD	No	Yes	Yes	No	No	No	No	Yes	No	EMP
HeteroEdge (2019) [74]	HP	Yes	Yes	Yes	No	No	No	No	Yes	No	EXP
Liu et al. (2019) [75]	No	Yes	No	No	No	No	No	No	No	No	EXP
Marlin (2019) [76]	No	No	No	No	No	No	No	N/A	N/A	Yes	EXP
DiStream (2020) [77]	NP	Yes	Yes	Yes	No	No	No	No	No	No	EXP
VU (2020) [78]	N/A	Yes	No	Yes	No	No	No	Yes	No	No	EMU
Clownfish (2020) [79]	No	Yes	No	No	No	No	No	No	No	No	EMU
Spatula (2020) [80]	No	Yes	No	No	No	No	No	No	No	No	EXP
REVAMP^{2}T [81]	No	No	No	Yes	No	No	Yes	No	No	Yes	EXP
Anveshak (2021) [82]	RR	Yes	Yes	Yes	No	No	No	No	No	No	EMU
Jang et al. (2021) [83]	No	No	Yes	Yes	No	No	No	No	No	No	EXP
OneEdge (2021) [84]	RR	Yes	Yes	Yes	No	No	No	Yes	Yes	No	EMU
Yoda (2021) [85]	N/A	Yes	N/A	N/A	No	No	No	No	No	No	EMU
PECAM (2021) [86]	N/A	Yes	No	No	No	No	Yes	No	No	No	EXP
DeepRT (2021) [87]	HP	Yes	No	No	No	No	No	No	No	No	EXP
CASVA (2022) [88]	No	Yes	No	No	No	No	No	No	No	No	SIM
MicroEdge (2022) [89]	BP	No	Yes	Yes	No	No	No	No	No	No	EXP
EdgeDuet (2022) [90]	HP	Yes	No	No	No	No	No	No	No	No	EXP, SIM
Ekya (2022) [91]	HP	Yes	No	No	No	No	No	No	No	No	EMU, SIM
Gemel (2023) [92]	HP	Yes	No	No	No	No	No	No	No	No	EXP
RECL (2023) [93]	HP	Yes	Yes	Yes	Yes	No	No	No	Yes	No	EXP
Runespoor (2023) [41]	No	Yes	No	Yes	No	No	No	No	No	No	EMU
REACT (2023) [94]	No	No	Yes	No	No	No	No	No	No	No	EXP
RL-CamSleep (2023) [95]	No	Yes	No	No	No	No	No	No	No	Yes	SIM

Scheduling Algorithm—HP: heuristic programming; MILP: mixed-integer linear program; BILP: binary integer linear program; NP: nonlinear program; RR: round robin; BP: bin packing. Testbed—EMU: emulation; EXP: experiment; SIM: simulator.

Analysis Criteria

We provide a brief description of the criteria by which the research works are analyzed in Tables 1 and 2. The criteria are derived from the requirements of an ideal IE-SVA system described in Section 4.

1. Project (year): Project name and year of publication. If the project is not named by the authors, then the last name of the first author is listed.
2. Focus: The primary design goal of the paper.
3. Cross-camera inference: "Yes" indicates that the video analytics pipelines jointly consider the output of two or more cameras. "No" indicates that the analytics of each camera are independent.
4. VAP components: Describes the distinct operations implemented by the video analytics pipelines described in the work. It should be noted that, while core components such as object detection and tracking involve computation-intensive deep learning algorithms, others such as video decoding and background subtraction use classical signal and image processing techniques.
5. Performance objectives: These include both application performance objectives and system performance objectives. Application latency is the end-to-end latency from the point of capturing the video stream until the delivery of detected events to the end user. Application accuracy is typically expressed with metrics such as F1 score. System performance objectives revolve around computation, memory, bandwidth, power, and cost constraints.

6. Profiling method: Profiling involves measuring the performance and resources associated with a video analytic pipeline using benchmark videos. Profiling could be performed either offline or online.
7. Architecture: Figure 2 shows a generic edge architecture for video analytics. Within this general framework, specific edge architectures include edge-cloud, distributed edge, and multi-tiered hierarchical edge depending on the layers involved, and the communication patterns. Furthermore, an implementation could involve a combination of these architectures. For example, a scalable system without a public cloud could be composed of clusters of distributed edge nodes, with a geo-distribution-based hierarchy (indicated as DE and HE in Table 1).
8. Scheduling: Describes algorithms reported for placing VAP components on the edge nodes such that performance and resource constraints are met.
9. Runtime adaptation: Indicates whether a run-time performance adaptation technique was employed.
10. Control plane: Indicates whether the work describes the design of a control plane. The control plane consists of the system software that controls the edge infrastructure.
11. Data plane: Indicates whether the work describes the design of a data plane. The data plane consists of the system software that facilitates the flow of data between the analytics components.
12. Human interface: Indicates whether the work reports aspects of the human user interface. Users, developers, and operators are the different types of people that interact with edge video analytic systems. The human interface design seeks to make this interaction easy and intuitive. A good UI/UX is key in ensuring that the systems constructed are used to their full potential by users.
13. Security: Indicates whether the work considers the cybersecurity aspects of the system. Securing the system from malicious use is of the utmost importance, especially considering the sensitive nature of video data.
14. Fault tolerance: indicates whether the work describes fault tolerance aspects of the system. Faults include both hardware and software failures.
15. Observability: indicates whether the work considers observability aspects of the system. The ability to measure and analyze system operational information and application logs are critical to understanding the operational status of large-scale IE-SVA systems, as well as troubleshooting, locating, and repairing failures.
16. Evaluation: Describes the type of evaluation testbeds used in the work. Approaches include the emulation of edge nodes using virtual machines, video workloads from standard datasets, the use of simulators, and edge hardware to build experimental testbeds.

6. Discussion

The previous section offered an overview of IoT edge streaming video analytics (IE-SVA) systems as discussed in the existing literature. In this section, we focus on specific techniques, categorizing them by the problems they address. For each technique, we briefly dive into a few of the papers listed in Tables 1 and 2, which in our opinion, provide good illustrations of the highlighted techniques. We also list the other papers wherein a similar technique is employed. Additionally, we highlight gaps in the research with suggestions for future work. It should be noted that the discussions are meant to provide a high level understanding of the technique, and are specifically focused on IE-SVA systems. The readers are encouraged to examine the cited work for implementation details.

6.1. Network Bandwidth

The "big data" nature of video analytics processing places a high demand on network bandwidth both between the edge and the cloud, and between the IoT camera nodes and the edge. As a result, many papers have focused on tackling the associated bandwidth challenges.

6.1.1. Technique 1: Trade-Offs in Application Accuracy vs. Bandwidth

The general idea is to exploit the ability of video analytics applications to operate at reduced accuracy to achieve bandwidth savings by using data reduction techniques such as dropping frames, or encoding frames at a reduced bit rate. The papers on AWSStream [66] and CASVA [88] provide two different approaches to the application of this technique.

AWStream employs a hybrid approach of offline and online training to construct a precise model that correlates an application's accuracy with its bandwidth usage via a set of tuning knobs. These knobs include resolution, framerate, and quantization. It autonomously identifies a Pareto-optimal policy to govern the timing and manner of leveraging these knobs. In real-time operation, AWStream's system continuously monitors network conditions and adjusts the data streaming rate to align with the available bandwidth. It maintains high-accuracy levels by utilizing the pre-learned Pareto-optimal configurations. During network congestion, the system employs a state machine-based adaptation algorithm to lower the accuracy, thereby reducing the data rate and preventing the buildup of a persistent queue.

In contrast to AWStream, CASVA aims to optimize server-side DNN inference accuracy while dynamically adjusting to fluctuating network bandwidth. It does this by manipulating configuration parameters such as resolution and frame rate. Unlike AWStream, CASVA forgoes the use of profiling-based methods to correlate configuration settings with performance. This decision is made to eliminate the computational overhead of profiling and its limitations in capturing video content variations. Instead, CASVA employs an actor–critic architecture based on deep reinforcement learning (DRL) to determine the optimal configuration for individual video segments. This approach enables the system to adapt to both granular changes in network bandwidth and variations in video content. CASVA's training process occurs offline and leverages a trace-driven simulator that accounts for diverse network conditions and video content. After completing the training, CASVA's configuration controller utilizes the learned policy to determine the optimal settings for each video segment during live streaming sessions.

Examples of other works reported in the literature that employ this technique include, Vigil [59], Wang et al. [68], and Runespoor [41].

6.1.2. Technique 2: Hybrid Computation between Edge and Cloud

It should be noted that, unlike the approach of exclusively scheduling the computation, either in the edge or the cloud, here the computation is performed jointly between the edge and the cloud. EdgeDuet [90] and Clownfish [79] projects employ this technique.

EdgeDuet conducts object detection through a hybrid edge-cloud architecture. Large objects are locally identified on the edge device using lightweight deep neural network (DNN) models. Conversely, small objects are detected using computationally intensive DNN models located in the cloud. To optimize the cloud-based detection of small objects, EdgeDuet employs two key techniques: region-of-interest (RoI) frame encoding and content-prioritized tile offloading. RoI frame encoding is specifically used to reduce the network bandwidth consumption. In this approach, only pixel blocks that are likely to contain small objects are transmitted at a high resolution, while the remaining portions of the frame are compressed and transmitted at low quality.

Clownfish employs a two-tiered deep learning (DL) architecture to optimize both the response speed and analytical accuracy. At the edge, it deploys a lightweight, optimized DL model for rapid data processing. In the cloud, a more comprehensive DL model ensures high-accuracy analytics. Clownfish takes advantage of the temporal correlation present in the video content to selectively transmit only a subset of video frames to the cloud, thereby conserving network bandwidth. The system then improves the quality of analytics by merging results obtained from both the edge and cloud models.

Examples of other works that employ this technique include FilterForward [71] and Runespoor [41].

6.1.3. Research Gaps

While offline and online profiling has received a fair amount of attention in the literature, the use of model-free methods has received considerably less attention. Only the CASVA [88] project explores the use of deep reinforcement learning (DRL) in IE-SVA systems. However, in our opinion, much more work needs to be performed in the application of DRL for exploiting the accuracy–bandwidth trade-offs. Offline learning strategies require the use of trace-driven simulators with real-world traces. The traces used in CASVA relies on a fixed broadband dataset provided by FCC which may not be applicable to a particular operating environment of interest. An alternative is to use online policy learning strategies, and hybrid approaches such as experience replay [96]. Another promising line of research would be the merging of the two techniques described above, with a continuum of hybrid edge-cloud processing depending on the bandwidth and resource constraints available at the edge.

6.2. Computational Efficiency

The high computational requirements of DNN models, and resource constraints at the edge have led researchers to explore techniques to effectively use DNNs at the edge.

6.2.1. Technique 1: Trade-Offs in Application Accuracy vs. Resource Usage

Similarly to bandwidth reduction, techniques that sacrifice the accuracy of the application can potentially reduce the computational load. The VideoEdge [65] project highlights the use of this technique.

In the VideoEdge project, video analytics queries are processed through a pipeline of computer vision components. These components have varying resource requirements and produce results with different levels of accuracy. Additionally, each component can be adjusted through various parameters such as frame resolution and frame rate, creating a large configuration search space. VideoEdge narrows this search space by identifying Pareto optimal configurations that balance resource demand and output accuracy. The project also introduces the concept of "dominant demand", defined as the maximum ratio of demand to capacity across all resources and clusters in the system hierarchy. This metric facilitates the direct comparison of different configurations in terms of both demand and accuracy, preventing the excessive consumption of any single resource. To optimize the system further, VideoEdge employs a greedy heuristic that iteratively explores configurations within the Pareto optimal band, switching to configurations that offer increased accuracy with a minimal increase in dominant demand.

Chameleon [67] is another work that employs this technique.

6.2.2. Technique 2: Edge Efficient DNN Models

The goal here is to come up with strategies to maximally utilize accelerators such as GPUs and TPUs on the edge nodes. Gemel [92], DeepQuery [72], DeepRT [87], and MicroEdge [89] are examples of applications of this technique.

To address the challenge of running multiple deep-learning-based video analytics applications on resource-limited edge GPUs, the GEMEL project introduces an innovative memory management strategy. They leverage the structural similarities among various edge vision models to share layers, including their weights, thereby reducing memory consumption and minimizing model-swapping delays. The process employs a step-by-step layer merging technique, prioritizing the layers that consume the most memory. Additionally, the method uses an adaptive retraining strategy based on the success or failure of each merging attempt. The GPU scheduling is also optimized to enhance the benefits of model merging by minimizing the frequency and duration of model swaps. Before deploying the merged models to the edge, GEMEL validates that they meet predetermined accuracy standards and continuously monitor for data drift.

The DeepQuery project focuses on multiple DNN models on resource-constrained edge GPUs by co-locating real-time and delay-tolerant tasks, and exploits a predictive and

plan ahead approach to alleviate resource contention due to co-locating by using dynamic batch sizing for delay-tolerant tasks so that they can finish before real-time tasks are to be scheduled.

The DeepRT project proposes a soft real-time GPU scheduler that employs an admission control mechanism that uses schedulability analysis, batches image frames from multiple requests using earliest deadline first (EDF) to perform the real-time scheduling of the batches, and an adaptation module that penalizes jobs that overrun deadlines so as to avoid the unpredictable deadline misses of other jobs in the system.

The MicroEdge project provides multi-tenancy support for coral TPUs by extending K3s [51], an edge-specific distribution of Kubernetes, through an admission control algorithm that allows for a fraction of TPU usage.

6.2.3. Technique 3: Continuous Learning at the Edge

To address data drift in dynamic video environments, DNN-based video analytics models deployed at the edge require regular retraining. Traditional retraining methods, however, are both resource-intensive and slow, making them unsuitable for edge devices with limited resources. The RECL [93] project offers a solution by selectively reusing pre-trained, specialized DNNs from a historical model repository. The key innovation lies in a rapid and reliable model selection process, achieved through a lightweight DNN-based model selector. This enables the quick identification and deployment of an appropriate model. Moreover, RECL incorporates an efficient scheduler that optimizes the retraining of multiple models. It does so by monitoring real-time accuracy gains during training and dynamically reallocating GPU resources to models that show greater improvement.

The Ekya [91] project also addresses the topic of continuous model retraining via a microprofiling approach that identifies the models that need to be retrained, and a resource scheduler for supporting both training and inference on a resource-constrained edge device.

6.2.4. Research Gaps

To enhance the efficiency of real-time video analytics workloads on edge devices, it is important to expand the focus beyond Nvidia GPUs to include accelerators from other vendors, such as AMD and Intel. One challenge in effectively utilizing Nvidia GPUs is the proprietary nature of their drivers, which limits the research flexibility. According to Otterness and Anderson [97], the open-source architecture of AMD GPUs facilitates better support for real-time tasks.

Addressing continuous learning on edge devices poses significant challenges due to resource constraints and the computational demands of training DNN models. Federated learning [98], a technique that maintains user privacy by locally aggregating data on user devices for collaborative model training, has gained considerable traction in recent years. Exploring the applicability of federated learning in a peer-to-peer context could capitalize on the diverse capabilities of edge nodes and periods of low activity, (for example, such as late-night hours in traffic monitoring systems) to efficiently perform distributed model training.

6.3. Scheduling

The edge represents a form of distributed computing that operates on heterogeneous resources. Scheduling algorithms are critical for optimizing the use of this complex environment, prompting researchers to explore more efficient algorithmic solutions. Some of the research works on edge video analytics, wherein scheduling is not the primary focus, have used simple approaches such as round-robin [82] and or bin-packing heuristics such as worst-fit [74]. Sophisticated approaches formulate the problem as a constraint optimization, and propose heuristics to solve it.

6.3.1. Technique: Constraint Optimization Problem Formulation

The scheduling problem is expressed as a cost optimization problem subject to constraints. As an illustrative example, in the VideoEdge [65] project, the application configuration and placement is modeled as the following binary integer problem (BIP)—maximize the sum of the accuracies of all queries (cost function) and subject it to the computation capacity (constraint), minimum accuracy (constraint), and the configuration and placement which are chosen for each task (constraint). Since solving the above optimization problem has an exponential time complexity, they propose a greedy heuristic.

Similar approaches have been adopted in the Lavea project [64] (mixed-integer linear problem), Distream [77] (nonlinear optimization), and VideoStorm [62] (utility maximization heuristic) projects.

6.3.2. Research Gaps

While the existing projects offer valuable scheduling techniques, there is a need for a comprehensive comparison of these approaches. Additionally, it is important to evaluate the suitability of distributed scheduling methods proposed for the cloud [99] adapted to the heterogeneous nodes of IE-SVA systems.

6.4. Control and Data Plane

The control plane serves as the mechanism to implement the scheduling decisions, while the data plane allows the transparent movement of data among the video analytics pipeline components. The distributed heterogeneous nature of the edge, and the potentially large data transfers involved in streaming video analytics makes the design of the control and data plane challenging. We highlight two recent works on control and data planes for IE-SVA systems.

6.4.1. Technique: Distributed Hierarchical Architecture

One approach to designing control planes for edge video analytics is to use industry standard distributed systems and adapt it to the edge such as in the open source K3s [51] and KubeEdge [100] projects. In contrast, instead of retrofitting Kubernetes, which is optimized for cloud-based, throughput-focused applications, and has a centralized control plane design, the OneEdge [84] project proposes a control plane that enables autonomous scheduling at individual edge sites without the need for central coordination. This is particularly useful for applications that largely operate independently. For applications requiring global coordination (for example, a drone fleet), OneEdge incorporates a centralized component that maintains an eventually consistent state of the system across all sites. This centralized component helps in effectively deploying multi-site applications.

To ensure reliable deployment decisions, OneEdge utilizes an enhanced two-phase commit protocol that synchronizes with the edge sites involved. It also offers specialized interfaces for developers, allowing them to implement latency-sensitive and location-aware scheduling. OneEdge continuously monitors end-to-end latency, ensuring it meets the specified service-level objectives (SLOs). To support location awareness, OneEdge triggers a migration process that relocates the client—such as a connected vehicle or drone—to a geographically suitable application instance.

6.4.2. Technique: Flexible Stream Processing Framework

Many projects have used an embedded queuing library such as ZeroMQ for the data plane [73,77,82]. While embedded libraries perform well, they lack features such as data persistence, encryption, and replication, which are available in comprehensive data streaming frameworks like Apache Flink [101] and Apache Storm [102]. However, these industry-supported frameworks have a limitation since they use a "stop-the-world" strategy for application reconfiguration, requiring global coordination. This is problematic at the edge, where resource constraints and dynamic conditions often lead to frequent reconfigurations and associated latency spikes.

To address this, the Shepherd [55] project implements a late-binding routing strategy. In this setup, a computation operation does not need to know in advance the location of the next operation that will consume its data. This flexibility is achieved through a separable transport layer that can be modified independently of the data processing layer. As a result, Shepherd allows for quick and flexible reconfigurations without the need for global coordination and with minimal system downtime.

6.4.3. Research Gaps

While the OneEdge project investigated a hierarchical control plane architecture for the edge, a peer-to-peer control plane architecture that avoids the need for a central cloud-based controller needs to be explored for IE-SVA systems. On the data plane side, incorporating persistence into the strategies proposed in the Shepherd project is needed to support stateful video analytics applications.

6.5. Multi-Camera Analytics

Collaborative video analytics of output from multiple cameras is necessary for improving the accuracy of analytics, and in pruning the search space exploiting temporal and spatial collaborations between cameras. We highlight the implementations of these two techniques that achieve this.

6.5.1. Technique 1: Multi-Camera Analysis to Improve Accuracy

In camera networks with overlapping fields of view, the ability to capture scenes from multiple angles can mitigate issues related to model limitations and object occlusions. Within the Vigil [59] project, cameras grouped into a cluster performed edge-based data fusion to enhance the surveillance capabilities. Specifically, video frames within the cluster are prioritized based on a utility metric that quantifies the number of "re-identified" objects, thereby improving the detection and tracking accuracy.

6.5.2. Technique 2: Cross-Camera Analytics to Improve Efficiency

In large-camera networks, implementing cross-camera analytics presents significant system challenges due to its computational and network resource demands. Unlike single-camera "stateless" tasks, such as object detection in a single feed, cross-camera analytics involves identifying correlations both within and between multiple video streams. The Spatula [80] project addresses these challenges through a three-stage approach: (1) Offline profiling phase: Spatula creates a spatio-temporal correlation model using unlabeled video data that encapsulate historically observed patterns of object locations and movements across the camera network. (2) Inference phase: during real-time analytics, Spatula consults the spatio-temporal model to eliminate camera feeds that are not likely to contain information relevant to the query identity's current location. This selective filtering effectively reduces the computational load and network bandwidth usage. (3) Recovery phase: To address any missed detections, Spatula performs a rapid review of recently filtered frames stored in its memory to identify any overlooked query instances.

The Anveshak [82] project similarly exploits cross-camera correlations in pedestrian tracking. In contrast with the Spatula project, they use a physics-based approach, where the information on the road network and the speed of person movement are used to constrain the cameras that need to be activated.

6.5.3. Research Gaps

As camera deployments grow, we believe the two above techniques of redundancy with multi-cameras, and efficiency based on predictive modeling could be combined in novel ways to further increase efficiency and accuracy. Additionally, the incorporation of PTZ cameras (pan, tilt, and zoom) incorporated into IE-SVA systems has not been explored. PTZ cameras allows for the reorienting and zooming of a particular camera (for example,

to zoom or change the field of view to track a particular object) potentially based on the output generated by neighboring cameras.

6.6. Video Analytics Pipeline Components

Most of the works surveyed have used DNN-based video analytics components, primarily for object detection, object tracking, and re-identification. The DNN-based components are computationally intensive, and require GPUs for model training. While CPU-based inference is possible, the use of embedded GPUs allows applications to sustain tens-of-frames-per-second throughput needed for real-time tracking.

Research Gaps

Recently, more complex video processing pipelines have been proposed for high-level cognitive tasks such as human activity recognition [103], and reasoning over the vision module output using large-language models (LLMs) [104]. Future research should investigate the edge computing approaches involving these complex models. Given the computational complexity of these models, distributed approaches may be required at the edge, even for inference purposes.

6.7. Fault Tolerance

While fault tolerance has received considerable attention in cloud computing research [105], we observed that IE-SVA systems' specific issues are only addressed in a single recent paper on the topic. The VU [78] project investigates surveillance camera failure modes from the study of a large-scale camera network. They identified 12 such failure modes and proposed an online failure detection approach.

Research Gaps

To attain the widespread adoption of edge video analytics, fault-tolerant operation is necessary to build reliable applications. Failures at the edge could include hardware, software, network, and power failures. The traditional cloud data center approaches of fault tolerance through redundancy are only partially applicable at the edge due to resource constraints. Unfortunately, we are unable to envision any straightforward approach to tackling this problem to all IE-SVA application contexts. We believe that successful solutions will be application- and customer-dependent. For example, in non-critical applications, if an edge node fails, camera streams could bypass the failed node, and directly transmit the video stream at a reduced rate to a backend cloud until the failure is fixed. This approach may allow some analytics operations to continue, albeit at a reduced accuracy.

6.8. Privacy

While much work has been performed regarding the privacy aspects of video analytics (see Section 7), some works on edge computing are limited to computing and discarding videos in IE-SVA systems as the sole privacy mechanism [81]. However, in practice, post incident analyses of videos may be needed for forensics and other investigative purposes. The OpenFaace project [63] proposes a mechanism for denaturing video streams that selectively blur faces according to specified policies at full-frame rates. However, privacy is limited to detecting and encrypting the regions of the image containing faces. The recently published PECAM [86] project proposes a video transformation technique that achieves broader visual privacy at the edge.

6.8.1. Technique: Reversible Video Transformations That Preserve Privacy While Allowing Analytics

The PECAM [86] project proposes a security-reinforced cycle-consistent generative adversarial network (GAN) to generate camera-specific video transformer and reconstructor pairs. Privacy preserving video transformation is performed by the transformer so that enough information is present for analytics tasks (for example, vehicle counting) while

preserving privacy (for example, license plate information). The reconstructor allows authorized parties to restore the requested frames to the original version. PECAM also proposes techniques to reduce the computational and bandwidth costs of the proposed privacy-preserving mechanisms. They evaluate the efficacy of the proposed approach under different attack scenarios.

6.8.2. Research Gaps

Homomorphic encryption [106] allows performing computations and analytics directly on encrypted data without requiring the data to be decrypted. The use of homomorphic encryption in video analytics is in the early stages [107]. We note that the computational costs of these methods will need to be addressed to make it amenable to edge video analytics.

6.9. Sustainability

Sustainability is a complex topic that includes the use of sustainable materials, reducing e-waste, reducing the carbon footprint by energy-efficient computing, and reducing social inequalities. In the existing literature concerning IE-SVA systems, energy efficiency is only addressed in a limited number of studies, while other topics remain unexplored.

6.9.1. Technique: Activate Video Analytics Only When Necessary

In the RL-CamSleep [95] project, for a smart parking application, a deep reinforcement learning-based controller automatically adapts the camera operation to parking patterns, saving energy while preserving the operational utility. For example, parking assistance is less needed in empty parking lots. Furthermore, they study the operation of the controller in the cloud, and at the edge powered by solar.

6.9.2. Research Gaps

To enhance the sustainability of IE-SVA systems, the focus on this aspect must be intensified, especially in large-scale deployments. Running applications with lower accuracy may facilitate the workload consolidation and enable the energy-efficient idling of edge nodes and IoT cameras. Additionally, system costs should be prioritized as a key design parameter to make IE-SVA systems accessible to economically disadvantaged communities.

Other works with a focus on energy-efficient IS-SVA systems include the REVAMP^{2}T [81] and the Marlin project [76]. Both of these projects explore techniques involving the energy-efficient operation of DNN models.

7. Path Ahead

In this section, we present a comprehensive research and development roadmap for streaming analytics edge systems. The Gantt chart of Figure 3 shows our proposed research roadmap in the short term (3 years), medium term (5 years), and long term (10 years).

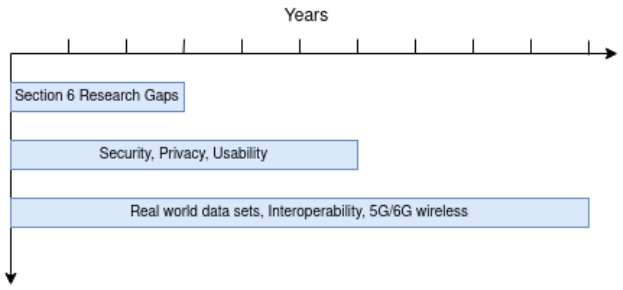

Figure 3. Gantt Chart with a proposed research roadmap over short-, medium- and long-term time horizons.

7.1. Short-Term Research

Section 6 identifies multiple research gaps in the existing state-of-the-art. Many of these could be implemented over the next three years. Additionally, we would like to highlight a few other research directions that could be pursued over the short term.

Joint compute and bandwidth optimization: In existing research, the tuning of DNN models has been studied in isolation, focusing on adjustments based on operating conditions like resource limitations and bandwidth variability. We propose that these isolated approaches should be integrated. Special consideration should be given to scenarios involving high-activity video content, where trade-offs between computation and communication are required based on the current computational load and network state.

Serverless: From a system standpoint, the application of the serverless paradigm, and more precisely, the function-as-a-service (FaaS) framework, at the edge merits further investigation to alleviate deployment burdens. Serverless computing hides the servers by providing programming abstractions for application builders that simplify development, making software easier to write [44]. Serverless computing is an active research topic in the cloud with many public cloud offerings. Open source serverless frameworks like KNative [108] and OpenFaaS [109] are available for use at the edge, but their compatibility with different types of edge hardware has not yet been evaluated. Furthermore, in the context of IE-SVA systems, given the resource constraints at the edge, it is not apparent what serverless abstractions are appropriate.

Testbeds: Current testbeds employ virtual machines (VMs) to emulate edge nodes. Emulation allows the direct execution of video analytics, albeit at a slower speed. An advantageous initiative would be the creation of a library of VMs that mimic various types of edge hardware, which researchers could readily use. Furthermore, to scrutinize communication-related bandwidth and latency issues, it would be beneficial to incorporate network simulators like NS3 [110] to the emulation platform.

7.2. Medium-Term Research

The medium-term work described in this section would require launching new research projects to tackle problems for edge video analytics systems. In our view, these studies could leverage the related body of work performed in other areas of computing, but would require the non-trivial adoption of these techniques to satisfy the constraints and peculiarities of video analytics at the edge.

Security: Foremost among these is security, a topic that has not been explicitly addressed by any of the video analytic edge systems reviewed in Section 5. In the context of IE-SVA systems, security involves a combination of traditional IoT security issues (for example, DDoS attacks), adversarial attacks on DNN models, and the compromise of privacy expectations (see below). The criticality of security for edge video analytics is highlighted by a March 2021 incident where a hacker group was able to publish live video feeds from 150,000 surveillance cameras [111]. An additional threat is that the large-scale edge computing infrastructure deployed for edge video analytics could be compromised, and recruited for running BotNet similar to the 2016 Mirai BotNet of compromised IoT devices [112]. The security triad of confidentiality, integrity, and availability has been extensively explored for cloud computing [113]. In their review on edge computing security, Xiao et al. [114] identified weak computation power, OS and protocol heterogeneity, attack unawareness, and coarse-grained access control as key differences between the cloud and edge from a security perspective. They list six major classes of attacks applicable to edge computing—DDoS attacks, side-channel attacks, malware injection attacks, authentication and authorization attacks, man-in-the-middle attacks, and bad data injection attacks. They review solutions proposed in the literature on the first four of these attack classes as they are particularly relevant to the edge. In the case of edge video analytics, as shown by Li et al. [115], side-channel attacks can be exploited to leak sensitive video information despite encryption. Defense strategies such as implementing fine-grained access control [114], the use of deep learning and machine learning algorithms to detect attacks [116], and the

use of hardware mechanisms for the isolation of software components [117] are possible. However, their applicability to resource-constrained edge nodes, and its potential impact on VAP performance needs to be systematically investigated across multiple platforms.

Adversarial attacks could be directed at the deep learning VAP components such as classification and object detection [118,119]. The goal of the adversarial attack is to insert small perturbations in the image to compromise the predictions of the deep-learning-based VAP components. Akhtar et al. [118] provide a comprehensive review of adversarial attacks and defenses in computer vision. Proposed defenses against these attacks require model robustification [120], input modification for removing perturbations [121], and adding external detectors to the model [122]. The performance impact of these defense strategies, as implemented on resource-constrained edge devices, needs to be comprehensively explored and evaluated.

Privacy: Since videos are a rich source of information, preserving privacy is of the utmost importance to prevent the leakage of unintended information. For example, an edge video analytics system for pedestrian safety might capture information regarding the identities of individuals. The REVAMP^{2}T project [81] uses skeletal pose information to track a pedestrian identity without storing any videos. The PECAM project [86] project proposes a novel generative adversarial network to perform the privacy-enhanced securely-reversible video transformation at the edge nodes. Similarly to security-related counter measures, the cost of implementing privacy-related computations on different types of research-constrained edge devices needs to be evaluated. Furthermore, since a video stream might be used for multiple applications, the ability of the proposed techniques to serve multiple applications needs to considered. A related technique would be the use of federated learning approaches [98] where the training data are used to train a local model that is then transmitted to a central coordinator, thus avoiding the need to send training data outside a specified privacy domain. The interplay between federated learning approaches and continuous learning [91,93] required at the edge to mitigate model drift needs an in-depth investigation.

Usability: The success of edge video analytics critically depends on how easily different types of personnel can interact with the system. Developers should be able to readily explore different VAP designs [123] and be provided with suitable system abstractions for VAP deployments. Operators should be able to readily determine the operational status of the complex distributed edge hierarchy, possibly involving hundreds of edge nodes, and thousands of cameras spread over a large geographic area. They should be quickly notified of system failures, and be able to perform root cause analysis to identify and rectify these. Users such as city personnel and law enforcement should be able to query the system in an intuitive fashion, preferably through the use of natural language.

In the cloud, the DevOps workflow uses an API-driven model that enables developers and operators to interact with infrastructure programmatically and at scale [124]. Artificial intelligence for IT operations (AIOps) [125,126], a recently introduced approach in DevOps, leverages data analytics and machine learning to improve the quality of computing platforms in a cost-effective manner using practices such as continuous integration and continuous deployment (CI/CD) [127]. Similar capabilities need to be developed to successfully implement and manage large-scale edge video analytic systems. An important consideration in applying successful cloud DevOps and AIOps practices at the edge is the challenge of moving large amounts of data (operational data and deployment images) over bandwidth-limited networks.

7.3. Long-Term Research

While predicting technology trajectories over the 5–10 year horizon is challenging given the ongoing rapid advances in all areas of computing especially in AI, we believe that overcoming certain problems would require research projects with a long time frame given the complexity of the problem, and the many stakeholders that need to be involved.

Real-world datasets: As the deployment of edge video analytics expands, we should seek to collect and open source real-world transactions and operational data with suitable privacy guards. Transactional data refer to the queries issued on the edge system by users and operators, while operational data refer to resource metrics, application logs, query traces, and failure statuses. This would allow researchers to gain an understanding of real-world systems, and direct their efforts towards impactful solutions.

Interoperability: As edge vision systems proliferate, there is a danger of these systems lacking interoperability due to custom protocols, data formats, and lack of standardization. Furthermore, updating legacy systems may become problematic, resulting in communities where these systems are deployed getting stuck with outdated technology. Standardization and modular design are two approaches that can be used to tackle this issue; the technical and standards community would need to take strong leadership roles in this regard within the next few years before these systems see widespread deployment.

5G and 6G wireless: High-speed communication technologies, like 5G and optic fiber networks, enable the widespread deployment of edge video analytics. However, it is crucial to recognize that the availability of these technologies is not uniform across the global population. Many areas still lack access to high-speed networks due to financial constraints and limited spectrum availability [128]. As the technical community progresses towards 6G standards, with an anticipated initial rollout around 2030 [129] boasting impressive capabilities of 1000 Gbps bandwidth and a latency of less than 100 microseconds, it becomes even more critical for the edge video analytics systems community to proactively explore and understand these emerging possibilities and challenges in order to make the most of 6G advancements.

8. Impact of Advancements in Other Areas in Computing

In this section, we provide a brief review of important developments and other areas of computing and related societal concerns that in our opinion are highly relevant to edge video analytics. Since these are fast-moving areas of research, the exact nature of their impacts on edge video analytics systems in not clear at this point.

Large language models: In recent years, significant progress has been observed in the realm of large language models (LLMs). These advancements have paved the way for more sophisticated and accurate AI applications with emergent abilities to tackle a wide range of tasks [130]. In recent months, the services of these LLMs models have been made available to the general public through services such as OpenAI's ChatGPT [131], Microsoft Bing AI [132], and Google's Bard [131]. More recently, Meta has made available its LLama 2 LLM available for free download [133] potentially allowing the broader technical community to specialize these models for specific tasks based on training with proprietary data. We believe that IE-SVA systems could incorporate these LLMs as a part of their analytics pipelines possibly to reason about relationships between events detected.

Web assembly: WebAssembly (Wasm)-based sandboxing has experienced a rapid rise as a notable technology. Wasm is a binary instruction format designed for a stack-based virtual machine, functioning as a portable compilation target for various programming languages, making it suitable for deployment on the web in client and server applications [134]. The platform-neutral nature of Wasm allows a single binary to be compiled and executed on diverse architectures and operating systems, eliminating the need for dealing with platform-specific information at the container level [135]. Consequently, this enables a lightweight, portable, and highly secure alternative to the container-based implementations of microservices, offering significant advantages, especially at the edge. It should be noted that the development of the Wasm system interface (WASI) is still ongoing [136].

AI regulation: The regulatory status of AI models is still evolving, but there is a growing awareness of the need for some form of regulation. Among the recent developments are the European Union Artificial Intelligence Act [137] and the United States White House statement on responsible AI research, development, and deployment [138]. In our opinion,

the edge video analytics research community should keep themselves abreast of these and other emerging regulations, so that the systems they design are compliant with them.

9. Conclusions

The widespread adoption of IoT edge streaming video analytics (IE-SVA) systems is propelled by the rapid advancements in deep-learning-based computer vision algorithms. These algorithms have revolutionized the automatic analysis of streaming video feeds, enabling the detection of events of interest. To facilitate this development, edge computing has emerged as a crucial component, offering advantages such as low latency, reduced bandwidth, and enhanced privacy. However, despite its potential, a significant gap remains in the successful practical implementation of edge-based streaming video analytics systems.

This paper presents an in-depth review of more than 30 studies on edge video analytics systems, assessed across 17 dimensions published over the last 8 years. Diverging from prior reviews, our approach examines each system holistically, enabling a comprehensive assessment of the strengths and weaknesses in various implementations. Our analysis reveals that certain crucial aspects essential for the practical realization of edge video analytics systems, such as security, privacy, and user support, and energy efficient operation, have not received sufficient attention in current research.

Based on these findings, we propose research trajectories spanning short-, medium-, and long-term horizons to address the identified challenges. Moreover, we explore trending topics in other computing domains that hold considerable potential to significantly impact the field of edge video analytics. This article aims to help new researchers rapidly understand the current state-of-the-art and inspire research initiatives that contribute to the widespread deployment of IE-SVA systems.

Funding: This research received no external funding.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. PRNewswire. Artificial Intelligence (AI) Camera Market to Grow at a CAGR of 12.04% from 2022 to 2027, 2023. Available online: https://finance.yahoo.com/news/artificial-intelligence-ai-camera-market-100000236.html (accessed on 19 October 2023).
2. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
3. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A survey of deep learning-based object detection. *IEEE Access* **2019**, *7*, 128837–128868.
4. Pop, D.O.; Rogozan, A.; Chatelain, C.; Nashashibi, F.; Bensrhair, A. Multi-task deep learning for pedestrian detection, action recognition and time to cross prediction. *IEEE Access* **2019**, *7*, 149318–149327.
5. Ananthanarayanan, G.; Bahl, P.; Bodík, P.; Chintalapudi, K.; Philipose, M.; Ravindranath, L.; Sinha, S. Real-time video analytics: The killer app for edge computing. *Computer* **2017**, *50*, 58–67.
6. Zhang, Q.; Sun, H.; Wu, X.; Zhong, H. Edge video analytics for public safety: A review. *Proc. IEEE* **2019**, *107*, 1675–1696.
7. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. https://doi.org/10.1109/JIOT.2016.2579198.
8. Barthélemy, J.; Verstaevel, N.; Forehead, H.; Perez, P. Edge-computing video analytics for real-time traffic monitoring in a smart city. *Sensors* **2019**, *19*, 2048.
9. IP Camera Bandwidth Calculator & CCTV Storage Calculator. Available online: https://www.jvsg.com/storage-bandwidth-calculator/ (accessed on 19 October 2023).
10. General Data Protection Regulation (GDPR). 2016. Available online: https://gdpr-info.eu/l (accessed on 19 October 2023).
11. Guiding Principles on Government Use of Surveillance Technologies) 2023. Available online: https://www.state.gov/wp-content/uploads/2023/04/Guiding-Principles-on-Government-Use-of-Surveillance-Technologies.pdf (accessed on 19 October 2023).
12. Xu, R.; Razavi, S.; Zheng, R. Deep Learning-Driven Edge Video Analytics: A Survey. *arXiv* **2022**, arXiv:2211.15751.
13. Hu, M.; Luo, Z.; Pasdar, A.; Lee, Y.C.; Zhou, Y.; Wu, D. Edge-Based Video Analytics: A Survey. *arXiv* **2023**, arXiv:2303.14329.
14. Goudarzi, M.; Palaniswami, M.; Buyya, R. Scheduling IoT applications in edge and fog computing environments: A taxonomy and future directions. *ACM Comput. Surv.* **2022**, *55*, 1–41.
15. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **2017**, *5*, 450–465.
16. Liu, F.; Tang, G.; Li, Y.; Cai, Z.; Zhang, X.; Zhou, T. A survey on edge computing systems and tools. *Proc. IEEE* **2019**, *107*, 1537–1562.

17. Chen, J.; Ran, X. Deep learning with edge computing: A review. *Proc. IEEE* **2019**, *107*, 1655–1674.
18. Greiffenhagen, M.; Comaniciu, D.; Niemann, H.; Ramesh, V. Design, analysis, and engineering of video monitoring systems: An approach and a case study. *Proc. IEEE* **2001**, *89*, 1498–1517.
19. Tian, Y.L.; Brown, L.; Hampapur, A.; Lu, M.; Senior, A.; Shu, C.F. IBM smart surveillance system (S3): Event based video surveillance system with an open and extensible framework. *Mach. Vis. Appl.* **2008**, *19*, 315–327.
20. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: Berlin/Heidelberg, Germany, 2022.
21. Liu, W.; Kang, G.; Huang, P.Y.; Chang, X.; Qian, Y.; Liang, J.; Gui, L.; Wen, J.; Chen, P. Argus: Efficient activity detection system for extended video analysis. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, Waikoloa, HI, USA, 3–7 January 2020; pp. 126–133.
22. TensorFlow 2 Detection Model Zoo. Available online: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md/ (accessed on 19 October 2023).
23. Li, Y.; Padmanabhan, A.; Zhao, P.; Wang, Y.; Xu, G.H.; Netravali, R. Reducto: On-camera filtering for resource-efficient real-time video analytics. In Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, Online, 10–14 August 2020; pp. 359–376.
24. Zhang, C.; Cao, Q.; Jiang, H.; Zhang, W.; Li, J.; Yao, J. A fast filtering mechanism to improve efficiency of large-scale video analytics. *IEEE Trans. Comput.* **2020**, *69*, 914–928.
25. Jebamikyous, H.H.; Kashef, R. Autonomous Vehicles Perception (AVP) Using Deep Learning: Modeling, Assessment, and Challenges. *IEEE Access* **2022**, *10*, 10523–10535.
26. Haghighat, A.K.; Ravichandra-Mouli, V.; Chakraborty, P.; Esfandiari, Y.; Arabi, S.; Sharma, A. Applications of Deep Learning in Intelligent Transportation Systems. *J. Big Data Anal. Transp.* **2020**, *2*, 115–145.
27. Fei, L.; Han, B. Multi-Object Multi-Camera Tracking Based on Deep Learning for Intelligent Transportation: A Review. *Sensors* **2023**, *23*, 3852. https://doi.org/10.3390/s23083852.
28. Cheong, K.H.; Poeschmann, S.; Lai, J.W.; Koh, J.M.; Acharya, U.R.; Yu, S.C.M.; Tang, K.J.W. Practical Automated Video Analytics for Crowd Monitoring and Counting. *IEEE Access* **2019**, *7*, 183252–183261.
29. Li, J.; Liao, J.; Chen, B.; Nguyen, A.; Tiwari, A.; Zhou, Q.; Yan, Z.; Nahrstedt, K. Latency-Aware 360-Degree Video Analytics Framework for First Responders Situational Awareness. In Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video, Vancouver, BC, Canada, 7–10 June 2023; pp. 8–14.
30. Garcia, R.V.; Wandzik, L.; Grabner, L.; Krueger, J. The Harms of Demographic Bias in Deep Face Recognition Research. In Proceedings of the 2019 International Conference on Biometrics (ICB), Crete, Greece, 4–7 June 2019; pp. 1–6. https://doi.org/10.1109/ICB45273.2019.8987334.
31. Rashwan, H.A.; Solanas, A.; Puig, D.; Martínez-Ballesté, A. Understanding Trust in Privacy-Aware Video Surveillance Systems. *Int. J. Inf. Secur.* **2016**, *15*, 225–234. https://doi.org/10.1007/s10207-015-0286-9.
32. Zhang, J.; Wu, C.; Wang, Y. Human Fall Detection Based on Body Posture Spatio-Temporal Evolution. *Sensors* **2020**, *20*, 946.
33. Ahumada, J.A.; Fegraus, E.; Birch, T.; Flores, N.; Kays, R.; O'Brien, T.G.; Palmer, J.; Schuttler, S.; Zhao, J.Y.; Jetz, W.; et al. Wildlife Insights: A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet. *Environ. Conserv.* **2020**, *47*, 1–6. https://doi.org/10.1017/S0376892919000298.
34. Muhammad, K.; Hussain, T.; Del Ser, J.; Palade, V.; De Albuquerque, V.H.C. DeepReS: A Deep Learning-Based Video Summarization Strategy for Resource-Constrained Industrial Surveillance Scenarios. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5938–5947.
35. Ahmad, H.M.; Rahimi, A. Deep Learning Methods for Object Detection in Smart Manufacturing: A Survey. *J. Manuf. Syst.* **2022**, *64*, 181–196.
36. Kirkpatrick, K. Tracking Shoppers. *Commun. ACM* **2020**, *63*, 19–21.
37. Lygouras, E.; Santavas, N.; Taitzoglou, A.; Tarchanidis, K.; Mitropoulos, A.; Gasteratos, A. Unsupervised Human Detection with an Embedded Vision System on a Fully Autonomous UAV for Search and Rescue Operations. *Sensors* **2019**, *19*, 3542.
38. Sambolek, S.; Ivasic-Kos, M. Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors. *IEEE Access* **2021**, *9*, 37905–37922.
39. Liu, D.; Abdelzaher, T.; Wang, T.; Hu, Y.; Li, J.; Liu, S.; Caesar, M.; Kalasapura, D.; Bhattacharyya, J.; Srour, N.; et al. IoBT-OS: Optimizing the Sensing-to-Decision Loop for the Internet of Battlefield Things. In Proceedings of the 2022 International Conference on Computer Communications and Networks (ICCCN), IEEE, Honolulu, HI, USA, 25–28 July 2022; pp. 1–10.
40. Satyanarayanan, M.; Harkes, J.; Blakley, J.; Meunier, M.; Mohandoss, G.; Friedt, K.; Thulasi, A.; Saxena, P.; Barritt, B. Sinfonia: Cross-tier Orchestration for Edge-Native Applications. *Front. Internet Things* **2022**, *1*, 1025247.
41. Wang, Y.; Wang, W.; Liu, D.; Jin, X.; Jiang, J.; Chen, K. Enabling edge-cloud video analytics for robotics applications. *IEEE Trans. Cloud Comput.* **2022**, *11*, 1500–1513.
42. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; et al. A view of cloud computing. *Commun. ACM* **2010**, *53*, 50–58.
43. Pahl, C.; Jamshidi, P.; Zimmermann, O. Architectural principles for cloud software. *ACM Trans. Internet Technol. (TOIT)* **2018**, *18*, 1–23.
44. Schleier-Smith, J.; Sreekanti, V.; Khandelwal, A.; Carreira, J.; Yadwadkar, N.J.; Popa, R.A.; Gonzalez, J.E.; Stoica, I.; Patterson, D.A. What serverless computing is and should become: The next phase of cloud computing. *Commun. ACM* **2021**, *64*, 76–84.

45. Docker: Accelerated, Containerized Application Development. Available online: https://www.docker.com/ (accessed on 19 October 2023).
46. Kubernetes: Production-Grade Container Orchestration. Available online: https://kubernetes.io/ (accessed on 19 October 2023).
47. Apache Kafka. Available online: https://kafka.apache.org/ (accessed on 19 October 2023).
48. JetStream. Available online: https://docs.nats.io/nats-concepts/jetstream (accessed on 19 October 2023).
49. RabbitMQ. Available online: https://www.rabbitmq.com/ (accessed on 19 October 2023).
50. Cattell, R. Scalable SQL and NoSQL Data Stores. *ACM Sigmod Rec.* **2011**, *39*, 12–27.
51. Lightweight Kubernetes: The Certified Kubernetes Distribution Built for IoT and Edge Computing. Available online: https://k3s.io/ (accessed on 19 October 2023).
52. Fu, X.; Ghaffar, T.; Davis, J.C.; Lee, D. EdgeWise: A Better Stream Processing Engine for the Edge. In Proceedings of the 2019 USENIX Annual Technical Conference (USENIX ATC 19), Renton, WA, USA, 10–12 July 2019; pp. 929–946.
53. Sonbol, K.; Özkasap, Ö.; Al-Oqily, I.; Aloqaily, M. EdgeKV: Decentralized, Scalable, and Consistent Storage for the Edge. *J. Parallel Distrib. Comput.* **2020**, *144*, 28–40.
54. George, A.; Ravindran, A.; Mendieta, M.; Tabkhi, H. Mez: An Adaptive Messaging System for Latency-Sensitive Multi-Camera Machine Vision at the IoT Edge. *IEEE Access* **2021**, *9*, 21457–21473.
55. Ramprasad, B.; Mishra, P.; Thiessen, M.; Chen, H.; da Silva Veith, A.; Gabel, M.; Balmau, O.; Chow, A.; de Lara, E. Shepherd: Seamless Stream Processing on the Edge. In Proceedings of the 2022 IEEE/ACM 7th Symposium on Edge Computing (SEC), IEEE, Seattle, WA, USA, 5–8 December 2022; pp. 40–53.
56. AWS Outposts Family. Available online: https://aws.amazon.com/outposts/ (accessed on 19 October 2023).
57. Xu, M.; Liu, Y.; Liu, X. A Case for Camera-as-a-Service. *IEEE Pervasive Comput.* **2021**, *20*, 9–17.
58. Ha, K.; Chen, Z.; Hu, W.; Richter, W.; Pillai, P.; Satyanarayanan, M. Towards Wearable Cognitive Assistance. In Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, Bretton Woods NH, USA, 16–19 June 2014; pp. 68–81.
59. Zhang, T.; Chowdhery, A.; Bahl, P.; Jamieson, K.; Banerjee, S. The Design and Implementation of a Wireless Video Surveillance System. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 426–438.
60. Microsoft Rocket for Live Video Analytics. Available online: https://www.microsoft.com/en-us/research/project/live-video-analytics/ (accessed on 19 October 2023).
61. Chen, T.Y.H.; Ravindranath, L.; Deng, S.; Bahl, P.; Balakrishnan, H. Glimpse: Continuous, real-time object recognition on mobile devices. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, Seoul, Republic of Korea, 1–4 November 2015; pp. 155–168.
62. Zhang, H.; Ananthanarayanan, G.; Bodik, P.; Philipose, M.; Bahl, P.; Freedman, M.J. Live video analytics at scale with approximation and Delay-Tolerance. In Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), Boston, MA, USA, 27–29 March 2017; pp. 377–392.
63. Wang, J.; Amos, B.; Das, A.; Pillai, P.; Sadeh, N.; Satyanarayanan, M. A scalable and privacy-aware IoT service for live video analytics. In Proceedings of the 8th ACM on Multimedia Systems Conference, Taipei , China, 20–23 June 2017; pp. 38–49.
64. Yi, S.; Hao, Z.; Zhang, Q.; Zhang, Q.; Shi, W.; Li, Q. Lavea: Latency-aware video analytics on edge computing platform. In Proceedings of the Second ACM/IEEE Symposium on Edge Computing, San Jose, CA, USA, 12–14 October 2017; pp. 1–13.
65. Hung, C.C.; Ananthanarayanan, G.; Bodik, P.; Golubchik, L.; Yu, M.; Bahl, P.; Philipose, M. Videoedge: Processing camera streams using hierarchical clusters. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, Seattle, WA, USA, 25–27 October 2018; pp. 115–131.
66. Zhang, B.; Jin, X.; Ratnasamy, S.; Wawrzynek, J.; Lee, E.A. Awstream: Adaptive wide-area streaming analytics. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, Budapest, Hungary, 20–25 August 2018; pp. 236–252.
67. Jiang, J.; Ananthanarayanan, G.; Bodik, P.; Sen, S.; Stoica, I. Chameleon: Scalable adaptation of video analytics. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, Budapest, Hungary, 20–25 August 2018; pp. 253–266.
68. Wang, J.; Feng, Z.; Chen, Z.; George, S.; Bala, M.; Pillai, P.; Yang, S.W.; Satyanarayanan, M. Bandwidth-efficient live video analytics for drones via edge computing. In Proceedings of the 2018 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, Seattle, WA, USA, 25–27 October 2018; pp. 159–173.
69. Liu, P.; Qi, B.; Banerjee, S. Edgeeye: An edge service framework for real-time intelligent video analytics. In Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking, Munich, Germany, 10–15 June 2018; pp. 1–6.
70. Salehe, M.; Hu, Z.; Mortazavi, S.H.; Mohomed, I.; Capes, T. Videopipe: Building video stream processing pipelines at the edge. In Proceedings of the 20th International Middleware Conference Industrial Track, Davis, CA, USA, 9–13 December 2019; pp. 43–49.
71. Canel, C.; Kim, T.; Zhou, G.; Li, C.; Lim, H.; Andersen, D.G.; Kaminsky, M.; Dulloor, S. Scaling video analytics on constrained edge nodes. *Proc. Mach. Learn. Syst.* **2019**, *1*, 406–417.
72. Fang, Z.; Hong, D.; Gupta, R.K. Serving deep neural networks at the cloud edge for vision applications on mobile platforms. In Proceedings of the 10th ACM Multimedia Systems Conference, Amherst, MA, USA, 18–21 June 2019; pp. 36–47.

73. Hsu, K.J.; Bhardwaj, K.; Gavrilovska, A. Couper: Dnn model slicing for visual analytics containers at the edge. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, Arlington, VA, USA, 7–9 November 2019; pp. 179–194.
74. Zhang, W.; Li, S.; Liu, L.; Jia, Z.; Zhang, Y.; Raychaudhuri, D. Hetero-edge: Orchestration of real-time vision applications on heterogeneous edge clouds. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, Paris, France, 29 April–2 May 2019; pp. 1270–1278.
75. Liu, L.; Li, H.; Gruteser, M. Edge assisted real-time object detection for mobile augmented reality. In Proceedings of the the 25th Annual International Conference on Mobile Computing and Networking, Los Cabos, Mexico, 21–25 October 2019; pp. 1–16.
76. Apicharttrisorn, K.; Ran, X.; Chen, J.; Krishnamurthy, S.V.; Roy-Chowdhury, A.K. Frugal following: Power thrifty object detection and tracking for mobile augmented reality. In Proceedings of the 17th Conference on Embedded Networked Sensor Systems, New York, NY, USA, 10–13 November 2019; pp. 96–109.
77. Zeng, X.; Fang, B.; Shen, H.; Zhang, M. Distream: Scaling live video analytics with workload-adaptive distributed edge intelligence. In Proceedings of the 18th Conference on Embedded Networked Sensor Systems, Virtual, 16–19 November 2020; pp. 409–421.
78. Sun, H.; Shi, W.; Liang, X.; Yu, Y. VU: Edge computing-enabled video usefulness detection and its application in large-scale video surveillance systems. *IEEE Internet Things J.* **2019**, *7*, 800–817.
79. Nigade, V.; Wang, L.; Bal, H. Clownfish: Edge and cloud symbiosis for video stream analytics. In Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, San Jose, CA, USA, 12–14 November 2020; pp. 55–69.
80. Jain, S.; Zhang, X.; Zhou, Y.; Ananthanarayanan, G.; Jiang, J.; Shu, Y.; Bahl, P.; Gonzalez, J. Spatula: Efficient cross-camera video analytics on large camera networks. In Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, San Jose, CA, USA, 12–14 November 2020; pp. 110–124.
81. Neff, C.; Mendieta, M.; Mohan, S.; Baharani, M.; Rogers, S.; Tabkhi, H. REVAMP 2 T: Real-time edge video analytics for multicamera privacy-aware pedestrian tracking. *IEEE Internet Things J.* **2019**, *7*, 2591–2602.
82. Khochare, A.; Krishnan, A.; Simmhan, Y. A scalable platform for distributed object tracking across a many-camera network. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1479–1493.
83. Jang, S.Y.; Kostadinov, B.; Lee, D. Microservice-based edge device architecture for video analytics. In Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 14–17 November 2021; pp. 165–177.
84. Saurez, E.; Gupta, H.; Daglis, A.; Ramachandran, U. Oneedge: An efficient control plane for geo-distributed infrastructures. In Proceedings of the ACM Symposium on Cloud Computing, Seattle, WA, USA, 1–4 November 2021; pp. 182–196.
85. Xiao, Z.; Xia, Z.; Zheng, H.; Zhao, B.Y.; Jiang, J. Towards performance clarity of edge video analytics. In Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, San Jose, CA, USA, 14–17 November 2021; pp. 148–164.
86. Wu, H.; Tian, X.; Li, M.; Liu, Y.; Ananthanarayanan, G.; Xu, F.; Zhong, S. Pecam: Privacy-enhanced video streaming and analytics via securely-reversible transformation. In Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, New Orleans, LA, USA, 25–29 October 2021; pp. 229–241.
87. Yang, Z.; Nahrstedt, K.; Guo, H.; Zhou, Q. Deeprt: A soft real time scheduler for computer vision applications on the edge. In Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, San Jose, CA, USA, 14–17 December 2021; pp. 271–284.
88. Zhang, M.; Wang, F.; Liu, J. CASVA: Configuration-Adaptive Streaming for Live Video Analytics. In Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications, IEEE, Online, 2–5 May 2022; pp. 2168–2177.
89. Cao, D.; Yoo, J.; Xu, Z.; Saurez, E.; Gupta, H.; Krishna, T.; Ramachandran, U. MicroEdge: A multi-tenant edge cluster system architecture for scalable camera processing. In Proceedings of the 23rd ACM/IFIP International Middleware Conference, Quebec, QC, Canada, 7–11 November 2022; pp. 322–334.
90. Yang, Z.; Wang, X.; Wu, J.; Zhao, Y.; Ma, Q.; Miao, X.; Zhang, L.; Zhou, Z. Edgeduet: Tiling small object detection for edge assisted autonomous mobile vision. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021.
91. Bhardwaj, R.; Xia, Z.; Ananthanarayanan, G.; Jiang, J.; Shu, Y.; Karianakis, N.; Hsieh, K.; Bahl, P.; Stoica, I. Ekya: Continuous learning of video analytics models on edge compute servers. In Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), Hyatt Regency Lake, WA, USA, 4–6 April 2022; pp. 119–135.
92. Padmanabhan, A.; Agarwal, N.; Iyer, A.; Ananthanarayanan, G.; Shu, Y.; Karianakis, N.; Xu, G.H.; Netravali, R. Gemel: Model Merging for Memory-Efficient, Real-Time Video Analytics at the Edge. In Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23), Boston, MA, USA, 17–19 April 2023; pp. 973–994.
93. Khani, M.; Ananthanarayanan, G.; Hsieh, K.; Jiang, J.; Netravali, R.; Shu, Y.; Alizadeh, M.; Bahl, V. RECL: Responsive Resource-Efficient Continuous Learning for Video Analytics. In Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23), Boston, MA, USA, 17–19 April 2023; pp. 917–932.
94. Ghosh, A.; Iyengar, S.; Lee, S.; Rathore, A.; Padmanabhan, V.N. REACT: Streaming Video Analytics On The Edge with Asynchronous Cloud Support. In Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation, San Antonio, TX, USA, 9–12 May 2023; pp. 222–235.
95. Rezaei, Y.; Khan, T.; Lee, S.; Mossé, D. Solar-powered Parking Analytics System Using Deep Reinforcement Learning. *ACM Trans. Sens. Netw.* **2023**, *19*, 1–27.
96. Zhang, S.; Sutton, R.S. A deeper look at experience replay. *arXiv* **2017**, arXiv:1712.01275.

97. Otterness, N.; Anderson, J.H. AMD GPUs as an Alternative to NVIDIA for Supporting Real-Time Workloads. In *Leibniz International Proceedings in Informatics (LIPIcs), Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020), Online, 7–10 July 2020*; Völp, M., Ed.; Schloss Dagstuhl: Dagstuhl, Germany, 2020; Volume 165, pp. 10:1–10:23. https://doi.org/10.4230/LIPIcs.ECRTS.2020.10.
98. Yin, X.; Zhu, Y.; Hu, J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–36.
99. Bittencourt, L.F.; Goldman, A.; Madeira, E.R.; da Fonseca, N.L.; Sakellariou, R. Scheduling in distributed systems: A cloud computing perspective. *Comput. Sci. Rev.* **2018**, *30*, 31–54.
100. KubeEdge Kubernetes Native Edge Computing Framework. Available online: https://kubeedge.io/ (accessed on 1 September 2023).
101. Apache Flink—Stateful Computations over Data Streams. Available online: https://flink.apache.org/ (accessed on 1 September 2023).
102. Apache Storm. Available online: https://storm.apache.org/ (accessed on 1 September 2023).
103. Pazho, A.D.; Neff, C.; Noghre, G.A.; Ardabili, B.R.; Yao, S.; Baharani, M.; Tabkhi, H. Ancilia: Scalable intelligent video surveillance for the artificial intelligence of things. *IEEE Internet Things J.* **2023**, *10*, 14940–14951.
104. Berrios, W.; Mittal, G.; Thrush, T.; Kiela, D.; Singh, A. Towards language models that can see: Computer vision through the lens of natural language. *arXiv* **2023**, arXiv:2306.16410.
105. Mukwevho, M.A.; Celik, T. Toward a smart cloud: A review of fault-tolerance methods in cloud systems. *IEEE Trans. Serv. Comput.* **2018**, *14*, 589–605.
106. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv. (Csur)* **2018**, *51*, 1–35.
107. Bentafat, E.; Rathore, M.M.; Bakiras, S. Towards real-time privacy-preserving video surveillance. *Comput. Commun.* **2021**, *180*, 97–108.
108. Knative Is an Open-Source Enterprise-Level Solution to Build Serverless and Event Driven Applications. Available online: https://knative.dev/docs/ (accessed on 19 October 2023).
109. Serverless Functions, Made Simple. Available online: https://www.openfaas.com/ (accessed on 19 October 2023).
110. ns3 Network Simulator. https://www.nsnam.org/ (accessed on 19 October 2023).
111. Hack of '150,000 Cameras' Investigated by Camera Firm. Published on 10 March 2021. Available online: https://www.bbc.com/news/technology-56342525 (accessed on 19 October 2023).
112. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the Mirai Botnet. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
113. Vacca, J.R., Ed. *Cloud Computing Security: Foundations and Challenges*; CRC Press: Boca Raton, FL, USA, 2016.
114. Xiao, Y.; Jia, Y.; Liu, C.; Cheng, X.; Yu, J.; Lv, W. Edge Computing Security: State of the Art and Challenges. *Proc. IEEE* **2019**, *107*, 1608–1631.
115. Li, H.; He, Y.; Sun, L.; Cheng, X.; Yu, J. Side-channel Information Leakage of Encrypted Video Stream in Video Surveillance Systems. In Proceedings of the IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, IEEE, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
116. Singh, S.; Sulthana, R.; Shewale, T.; Chamola, V.; Benslimane, A.; Sikdar, B. Machine-learning-assisted security and privacy provisioning for edge computing: A survey. *IEEE Internet Things J.* **2021**, *9*, 236–260.
117. Coppolino, L.; D'Antonio, S.; Mazzeo, G.; Romano, L. A comprehensive survey of hardware-assisted security: From the edge to the cloud. *Internet Things* **2019**, *6*, 100055.
118. Akhtar, N.; Mian, A.; Kardan, N.; Shah, M. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access* **2021**, *9*, 155161–155196.
119. Serban, A.; Poll, E.; Visser, J. Adversarial examples on object recognition: A comprehensive survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–38.
120. Bai, T.; Luo, J.; Zhao, J.; Wen, B.; Wang, Q. Recent advances in adversarial training for adversarial robustness. *arXiv* **2021**, arXiv:2102.01356.
121. Guo, C.; Rana, M.; Cisse, M.; Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv* **2017**, arXiv:1711.00117.
122. Qin, Y.; Frosst, N.; Sabour, S.; Raffel, C.; Cottrell, G.; Hinton, G. Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. *arXiv* **2019**, arXiv:1907.02957.
123. Bastani, F.; Moll, O.; Madden, S. Vaas: Video analytics at scale. *Proc. VLDB Endow.* **2020**, *13*, 2877–2880.
124. Leite, L.; Rocha, C.; Kon, F.; Milojicic, D.; Meirelles, P. A survey of DevOps concepts and challenges. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–35.
125. Notaro, P.; Cardoso, J.; Gerndt, M. A Survey of AIOps Methods for Failure Management. *ACM Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–45.
126. Li, Y.; Jiang, Z.M.; Li, H.; Hassan, A.E.; He, C.; Huang, R.; Zeng, Z.; Wang, M.; Chen, P. Predicting node failures in an ultra-large-scale cloud computing platform: An aiops solution. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **2020**, *29*, 1–24.

127. Shahin, M.; Babar, M.A.; Zhu, L. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access* **2017**, *5*, 3909–3943.
128. 5G Network Coverage Outlook. Available online: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/network-coverage (accessed on 19 October 2023).
129. Next-Gen Mobile Internet—6G—Will Launch in 2030, Telecom Bosses Say, Even as 5G Adoption Remains Low. Published on 7 March 2023. Available online: https://www.cnbc.com/2023/03/08/what-is-6g-and-when-will-it-launch-telco-execs-predict.html (accessed on 19 October 2023).
130. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent abilities of large language models. *arXiv* **2022**, arXiv:2206.07682.
131. ChatGPT: Get Instant Answers, Find Creative Inspiration, and Learn Something New. Available online: https://openai.com/chatgptl (accessed on 19 October 2023).
132. Bing Helps You Turn Information into Action, Making It Faster and Easier to Go from Searching to Doing. Available online: https://www.bing.com/?/ai (accessed on 19 October 2023).
133. Introducing Llama 2—The Next Generation of oUr Open Source Large Language Model. Available online: https://ai.meta.com/llama/ (accessed on 19 October 2023).
134. Web Assemblyl. Available online: https://webassembly.org/ (accessed on 19 October 2023).
135. Containers vs. WebAssembly: What Is the Difference? Published on March 2022. Available online: https://www.fermyon.com/blog/webassembly-vs-containers (accessed on 19 October 2023).
136. WebAssembly System Interface. Published on March 2022. Available online: https://github.com/WebAssembly/WASI (accessed on 19 October 2023).
137. Artificial Intelligence Act. Available online: https://artificialintelligenceact.eu(accessed on 19 October 2023).
138. Ensuring Safe, Secure, and Trustworthy AI. Published on 21 July 2023. Available online: https://www.whitehouse.gov/wp-content/uploads/2023/07/Ensuring-Safe-Secure-and-Trustworthy-AI.pd (accessed on 19 October 2023).