Review

Edge Computing Systems for Streaming Video Analytics : Trail Behind and the Paths Ahead

Arun A Ravindran *

Posted Date: 4 August 2023

doi: 10.20944/preprints202308.0383.v1

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

Article

Edge Computing Systems for Streaming Video Analytics : Trail Behind and the Paths Ahead

Arun Ravindran

Department of Electrical and Computer Engineering, University of North Carolina at Charlotte;
e-mail@arun.ravindran@charlotte.edu

Abstract: The falling cost of cameras, the advancement of AI based computer vision algorithms, and powerful hardware accelerators for deep learning have enabled wide-spread 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 done 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 the state-of-the-art computer vision algorithms, and 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 edge video analytics 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 edge video analytics 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 field of edge video analytics.

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

1. Introduction

The falling cost of video cameras [1], and the increasing capability of deep learning based computer vision algorithms [2,3], makes it possible to use streaming video analytics for visual sensing of the environment. Streaming video analytics refers to 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 streaming analytics 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 video 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 compute intensive algorithms needed to process it [5]. A number of other use cases of streaming video analytics exists in multiple areas including health care, manufacturing, environmental monitoring, and national security [6].

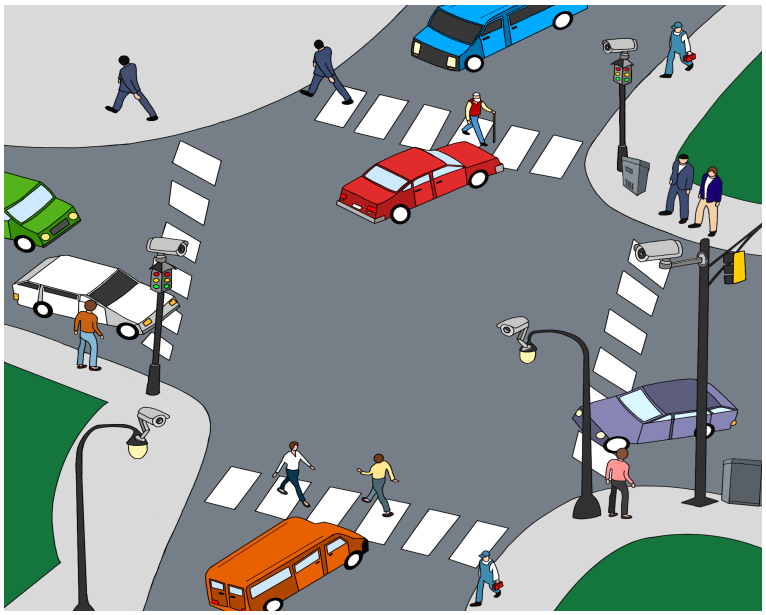

Figure 1. A smart-city traffic intersection equipped with surveillance cameras. Many city departments could consume the camera feeds for multiple applications including traffic monitoring, pedestrian safety, detecting traffic law violations, public safety, and environmental monitoring. Each application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

1.1. Need for Edge Computing

The compute could potentially be realized on the cloud taking advantage of powerful and scalable compute resources available on-demand. However, bandwidth, latency, and privacy necessitates the use of the edge computing paradigm for streaming video analytics [7,8]. To put this in context, a single H.265 1080p IP camera with high video quality operating at 15 fps requires a 2.3 Mbps uplink and generates over 25 GB of data per day [9]. A half 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 if 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 government use of surveillance technologies [11]. Further, 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. Further, the aggregate bandwidth requirements are reduced due to distributed processing at the edge enabling scaling of video cameras. Moreover, sensitive video data can be confined to the privacy perimeter expected by the user (for example, home, legal jurisdiction).

1.2. Key Contributions

In recent years, there has been a notable surge in research focused on streaming video analytics at the edge. Despite the widespread deployment of cameras in cities (for example, London, UK has more than half a million 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

 doi:10.20944/preprints202308.0383.v1

effectively utilizing AI to analyze these camera streams in real-time to derive actionable insights. The purpose of this paper is to thoroughly analyze the state-of-the-art on edge video analytic systems as reported in the research literature, so as to provide clarity to the research and practitioner community on the progress thus far, and the additional research and development that needs to be done to close the gap.

To achieve these goals, we begin by outlining the characteristics of an ideal edge computing system. By using this ideal system as a framework, we analyze existing research literature on video analytics along 17 unique dimensions. The analysis is presented in tabular format with 30 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 (1-3 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 (2-5 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 (3-10 years) envisions the future evolution of streaming video analytics at the edge and proposes exploratory research directions.

1.3. Paper Organization

The 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 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 edge video analytic systems. Section 6 then outlines our research vision, breaking it down into short-term, medium-term, and long-term objectives. Section 7 briefly examines how other advancements in computing might impact systems designed for video analytics at the edge. Finally, in Section 8, 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 provides an extensive perspective on the techniques employed, it can also make it challenging to understand how the different techniques are integrated in 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 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. We provide a comprehensive overview of 30 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

grasp the evolution of edge video analytics systems, the mix of techniques employed in a particular system, and areas that have received limited attention from the research community thus far. Further, 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.

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 challenges in implementing these systems. References to more detailed tutorial-like treatment of these topics is provided.

3.1. Streaming Video Analytics

Streaming video analytics involves real-time processing of video frames to extract valuable information [18,19]. For instance, 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 [21]. Figure 2 [20] illustrates an example of an video analytics pipeline (VAP) for recognizing multi-person actions. Many of these stages utilize computationally intensive deep learning algorithms. Additionally, multiple deep learning algorithms with varying performance-accuracy tradeoffs 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 tradeoffs. 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 decoding bit rate, dropping frames, or filtering specific parts of video frames, impact both the application's requirements and computational resources [23,24].

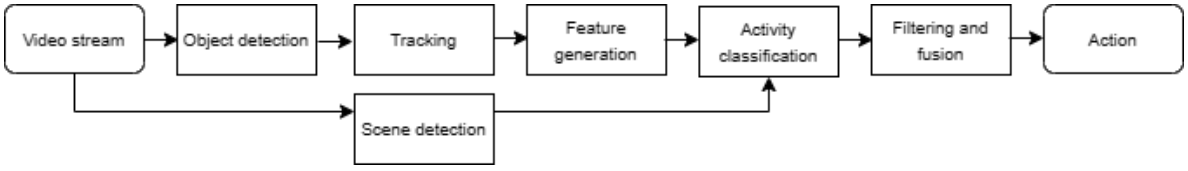

Figure 2. Activity detection video analytics pipeline (VAP) proposed in the Argus project [20]

3.2. Application Use Cases

This section provides a brief overview of the various applications of streaming video analytics across diverse industries.

 doi:10.20944/preprints202308.0383.v1

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 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 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 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 done on the edge of the network. As shown in Figure 3, the edge video analytics hardware hierarchy forms a tree structure with cameras at the leaf nodes; single board computers on a Local Area Network (LAN) shared with the 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 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. Further, 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 compute may be added to the tree structure as the system scales. End users connect to the system via web based or mobile application frontends.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

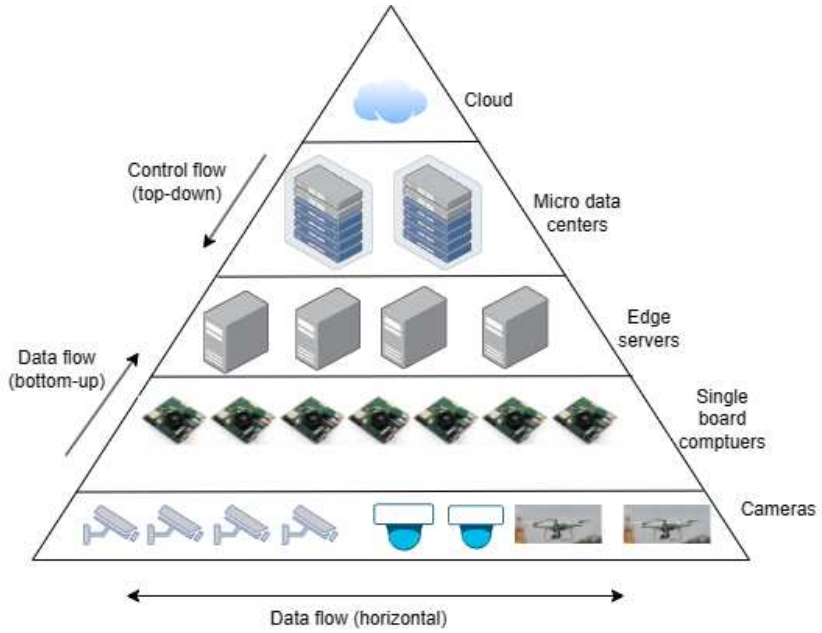

Figure 3. Edge video analytics systems hierarchy. Aside from the cameras, not all levels need be present in an implementation. The compute nodes may be organized as clusters for availability, and fault tolerance. Data flows vertically up the hierarchy starting from the cameras, while the control flows down the hierarchy. In some implementations , data may flow horizontally as well.

3.3.2. Hardware

Hardware components comprising the edge video analytics, include wired and wirless 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 lower speed (1 Mbps - 1 Gbps) less reliable networks. 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 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 virtualization of the physical compute, storage, and network. Users are able to provision these

 doi:10.20944/preprints202308.0383.v1

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], 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 compute, 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 remains 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 execute efficiently 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, a streaming video analytics edge platform, including the 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.

 doi:10.20944/preprints202308.0383.v1

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 compute 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 deep pocketed companies, community and non-profit owned edge infrastructure are often subject to tight budgetary constraints. Finally, from an environmental perspective minimizing the energy use, incorporation of sustainable energy sources, and maximizing as well as extending the life of equipment are important considerations.

4. Ideal System Requirements for Video Analytics at the Edge

In this Section, we present our view of the ideal characteristics of edge 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 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 thus, allowing for flexibility and scalability in their deployment. Additionally, the system is designed to take advantage of a wide variety of distributed compute resources including single board computers, edge servers, micro data centers, private and public clouds. These compute 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 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

Efficient operation of the surveillance system is a critical aspect. 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 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

4.4. Human Factor

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. The system provides multiple logical abstractions allowing application developers to choose abstractions appropriate for their use case. Furthermore, the system is designed to be easily maintained without the need for sophisticated engineering teams, minimizing the complexity and resources required for system upkeep.

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. Analysis of Reported End-to-End Systems

In this Section, we analyze research reported in the literature for realizing video analytics processing at the edge. Importantly, we exclude works solely dedicated to video analytics algorithms, or those involving cloud-only video analytics. While cloud video analytics has a longer history [19], and works such as Gabriel [58] and Vigil [59] involve the use of video processing at the edge, arguably 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 edge video analytics systems. In their article, the authors make a case of why edge computing is essential for video analytics, and sketch out the core technologies needed to realize such a system. Their application emphasis is on traffic analytics. Since their initial paper, the Microsoft Research group has been active in publishing research articles on this topic including 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.

Table 1 and Table 2 shows research projects reported in the literature on edge video analytics from 2015 - July 2023 (the date of the present review). Thirty papers are analyzed in chronological order along 17 different criteria.

Table 1. Literature review covering criteria 1 - 6.

Project (Year)	Focus	Perf. Obj.	Cross Cam.	VAP compon.	Profile	Arch.
Vigil (2015)[59]	SWS	MaxBW	Yes	FDR	No	EC
VideoStorm (2017)[61]	E2E	MaxAcc-MinLat	No	BS,OD,OT	OFF	DE,EC
Lavea (2017)[62]	E2E	MinLat	No	CR	OFF	DE,HE
VideoEdge (2018)[63]	E2E	MaxAcc-ResCon	No	OC,CR	OFF	DE,EC
AWStream (2018)[64]	SS	MaxAcc-ResCon	No	OD,OC	OFF	EC
Chameleon (2018)[65]	VAP KT	MaxAcc-ResCon	Yes	OD,OC	ON	SE
Wang et. al. (2018)[66]	VAP KT	MinBW-MaxAcc	No	OD,OC	No	HE
EdgeEye (2018)[67]	SS	MaxTh	No	OD	No	SE
VideoPipe (2019)[68]	SS	MinLat	No	PD,AR	No	DE
FilterForward (2019)[69]	VAP KT	MinBW-MaxAcc	No	OC	ON	EC
DeepQuery (2019)[70]	SS	MinLat	No	OD,OT,SD,VD	ON	ME
Couper (2019)[71]	SS	UD	No	OC	UD	EC

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

Table 1. *Cont.*

Project (Year)	Focus	Perf. Obj.	Cross Cam.	VAP compon.	Profile	Arch.
HeteroEdge (2019)[72]	SS	MinLat	No	3DR	OFF	DE
DiStream (2020)[73]	VCA	MaxTh	No	BS,OD	OFF	DE,HE
VU (2020)[74]	FD	MinBW-MaxAcc	No	LIP	OFF	EC
Clownfish (2020)[75]	VAP KT	MinBW-MaxAcc	No	AR	ON	EC
Spatula (2020)[76]	MC	MinBW-MaxAcc	No	OD,RID	OFF	HE
REVAMP^{2}T[77]	PP	MinPow-MaxAcc	No	PD,RID,OT	No	HE
Anveshak (2021)[78]	MC	MinBW-PerfCon	Yes	OD,RID,OT	No	HE,DE,EC
Jang et. al. (2021)[79]	SS	MinLat-MaxAcc	No	OD,RID,OT	No	DE
OneEdge (2021)[80]	SS	MinLat	No	OD,OT	OFF	HE,DE,EC
Yoda (2021)[81]	VCA	MaxAcc	No	OD	OFF	N/A
PECAM (2021)[82]	PP	MaxPriv-MaxAcc	Yes	GS	N/A	N/A
CASVA (2022)[83]	VAP KT	MaxAcc-MinLat	No	OD,SS	OFF	HE
MicroEdge (2022)[84]	SS	MaxTh	No	OD,OT,PS	OFF	HE,DE
EdgeDuet (2022)[85]	VAP KT	MaxAcc-MinLat	No	OD	ON	EC
Ekya (2022)[86]	OTr	MaxAcc	No	OC,OD	OFF	SE
Gemel (2023)[87]	MM	MinMem-MacAcc	No	M-DNN	OFF	EC
RECL (2023)[88]	OTr	MaxAcc	No	OC,OD	ON	HE
Runespoor (2023)[41]	VCA	MaxAcc-MinBW	No	SS,OD	OFF	EC

Focus — SWS: Scalable Wireless Systems E2E: End-to-End System SS: System Software VAP KT: VAP Knob Tuning VCA: Video Content Adaptation FD: Failure Detection MC: Multi Camera PP: Privacy Protection OTr: Online Training MM: Model Merging. **Performance Objective** — MaxAcc: Maximize Accuracy MinLat: MinimizeLatency 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: Charaacter 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: Hiearchical Edge SE: Single Edge ME: Mobile Edge

5.1. 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 ideal edge video analytics 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 focus of the paper. While all the papers considered present complete systems, many of them are focused on specific research objectives such as maximizing inference accuracy while minimizing bandwidth, or minimizing latency.
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 compute-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

 doi:10.20944/preprints202308.0383.v1

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 compute, memory, bandwidth, power, and cost constraints.

6. Profiling method: Profiling involves measuring the performance and resources associated with a VAP using benchmark videos. Profiling is typically done offline, and the information obtained is used to make scheduling decisions. However, offline profiling may fail to capture dynamic system characteristics (e.g., when the actual video significantly differs from the benchmark videos), requiring online profiling of the VAP. Additionally, VAPs have several tunable knobs that result in different performance-resource tradeoffs. These knobs include the choice of deep learning models, the choice of frames that need processing, the choice of regions in the frames, and the encoding bit rate. The combinatorial explosion results in a vast design space, necessitating strategies that efficiently explore the design space.
7. Architecture: Figure 3 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. Further, 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, HE in Table 1).
8. Scheduling: This describes algorithms for placing VAP components on the edge nodes such that performance and resource constraints are met. Scheduling algorithms proposed in the literature are either heuristic or based on constraint optimizations often solved with heuristics. Scheduling can be done either centrally or in a distributed manner.
9. Runtime adaptation: Performance tuning knobs are used to adapt the VAP at the edge to dynamic operating conditions.
10. Control plane: The control plane consists of the system software that controls the edge infrastructure. The organization of the control plane mirrors that of the edge architecture. Typically, control planes are implemented via agents communicating via REST/RPC APIs.
11. Data plane: The data plane consists of the system software that facilitates the flow of data between the VAP components within and across edge nodes.
12. Human interface: Users, developers, and operators are the different types of people that interact with edge video analytics system. 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: Securing the system from malicious use is of utmost importance, especially considering the sensitive nature of video data. Cybersecurity is a vast topic involving a number of subtopics such as system vulnerabilities, social engineering attacks, network security, malware, and cryptography [89]. While a well developed body of work exists in each of these areas, the research on the security of edge video analytics systems is still in the early stages.
14. Fault tolerance: Distributed systems are subject to hardware and software failures. In general, fault tolerance is challenging at the edge due to resource limitations. Some works have proposed the use of Kubernetes and its fault tolerance mechanisms at the edge. Only one work that we are aware of has studied different types of camera faults at the edge [74].
15. Observability: The ability to measure and analyze system operational information and application logs is critical to understanding the operational status of a large-scale network of cameras and edge compute infrastructure, as well as troubleshooting, locating, and repairing failures. AIOps involves the use of AI to analyze this operational data. While the use of AIOps is widespread in cloud computing [90], very little work has been reported in the literature on on edge video analytics systems.
16. Evaluation: The performance of the system needs to be evaluated experimentally for different operating conditions. Typical evaluation testbeds consist of emulating edge nodes using virtual machines, with video workloads from standard datasets. Some works have used actual edge hardware to build a testbed. Unfortunately, experimental testbeds are not scalable to the thousands of cameras employed in real-world applications.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

Table 2. Literature review covering criteria 7 - 17.

Project (Year)	Sched.	Run-time Adapt.	Ctrl. Plane	Data Plane	UI	Secu-rity	Priv-acy	Fault tol.	Obsv.	Sust.	Testbed
Vigil (2015)	HP	No	No	No	No	No	No	No	No	No	EXP,SIM
VideoStorm (2017)	HP	Yes	No	No	No	No	No	No	No	No	EMU
Lavea (2017)	MILP	Yes	No	Yes	No	No	No	No	No	No	EXP
VideoEdge (2018)	BILP	Yes	No	No	No	No	No	No	No	No	EMU
AWStream (2018)	HP	Yes	No	No	No	No	No	No	No	No	EMU
Chameleon (2018)	NA	Yes	No	No	No	No	No	No	No	No	EXP
Wang et. al. (2018)	No	Yes	No	No	No	No	No	No	No	No	EMU
EdgeEye (2018)	NA	No	Yes	Yes	Yes	No	No	No	No	No	EXP
VideoPipe (2019)	No	No	No	Yes	No	No	No	No	No	No	EXP
FilterForward (2019)	No	Yes	No	No	No	No	No	No	No	No	EXP
DeepQuery (2019)	HP	Yes	Yes	Yes	No	No	No	No	No	No	EXP
Couper (2019)	UD	No	Yes	Yes	No	No	No	No	Yes	No	EMP
HeteroEdge (2019)	HP	Yes	Yes	Yes	No	No	No	No	Yes	No	EXP
DiStream (2020)	NP	Yes	Yes	Yes	No	No	No	No	No	No	EXP
VU (2020)	NA	Yes	No	Yes	No	No	No	Yes	No	No	EMU
Clownfish (2020)	No	Yes	No	No	No	No	No	No	No	No	EMU
Spatula (2020)	No	Yes	No	No	No	No	No	No	No	No	EXP
REVAMP2T	No	No	No	Yes	No	No	Yes	No	No	Yes	EXP
Anveshak (2021)	RR	Yes	Yes	Yes	No	No	No	No	No	No	EMU
Jang et. al. (2021)	No	No	Yes	Yes	No	No	No	No	No	No	EXP
OneEdge (2021)	RR	Yes	Yes	Yes	No	No	No	Yes	Yes	No	EMU
Yoda (2021)	NA	Yes	NA	NA	No	No	No	No	No	No	EMU
PECAM (2021)	NA	Yes	No	No	No	No	Yes	No	No	No	EXP
CASVA (2022)	No	Yes	No	No	No	No	No	No	No	No	SIM
MicroEdge (2022)	BP	No	Yes	Yes	No	No	No	No	No	No	EXP
EdgeDuet (2022)	HP	Yes	No	No	No	No	No	No	No	No	EXP,SIM
Ekya (2022)	HP	Yes	No	No	No	No	No	No	No	No	EMU,SIM
Gemel (2023)	HP	Yes	No	No	No	No	No	No	No	No	EXP
RECL (2023)	HP	Yes	Yes	Yes	Yes	No	No	No	Yes	No	EXP
Runespoor (2023)	No	Yes	No	Yes	No	No	No	No	No	No	EMU

Scheduling Algorithm — HP: Heurisitc Programming MILP: Mixed Interger Linear Program BILP: Binary Integer Linear Program NP: Non-Linear Program RR: Round Robin BP: Bin Packing
Testbed — EMU: Emulation EXP: Experiment SIM: Simulator

 doi:10.20944/preprints202308.0383.v1

5.2. Discussion

The literature reviewed does not fully address all the outlined criteria. Specifically, the least discussed areas within edge systems for video analytics are cross-camera inference (3 papers), security (0 papers), privacy (2 papers), fault tolerance (2 papers), human interface (2 papers), observability (4 papers), and sustainability (1 paper).

In terms of scheduling methodologies, the predominance of heuristic algorithms is notable, often without providing a substantial rationale for choosing such an approach over alternative methods. Runtime adaptations commonly employ periodic rescheduling. Recent research such as the Casva project [83] have begun to explore the potential of Deep Reinforcement Learning in this context.

The design of the control plane is addressed by the OneEdge project [80], presenting a geo-distributed control plane. The authors uniquely focus on the issue of membership consistency in large-scale distributed edges, suggesting a two-phase commit protocol to handle the eventually consistent nature of the aggregate membership state.

As for data planes, several studies have utilized the ZeroMQ [91] messaging library, forgoing higher-performance, fault-tolerant, cloud-native message-broker based systems like NATS [48]. This choice is presumably due to ZeroMQ's simplicity compared to message brokers, though this presumption lacks experimental validation.

When examining Video Analytics Pipeline (VAP) components, the primary focus across studies lies in object detection, re-identification, and tracking using various deep learning models. Nevertheless, only the GEMEL project [87] investigates techniques for running multiple VAPs on a single machine, a requirement for supporting multiple applications and camera streams. The capability of online training of VAPs is explored by recent projects Ekya [86] and RECL [88].

Concerning edge architectures, Anveshak [78] and OneEdge [80] projects address highly scalable, distributed, hierarchical edge operation in conjunction with the cloud. In terms of evaluation platforms, most studies utilize an emulation-based testbed with standard video datasets, supplemented by a handful of small-scale experimental testbeds. This review underscores the necessity for real-world, large-scale surveillance data to be made available to researchers.

In conclusion, there has been considerable progress in edge video analytics over the past six years. However, significant work still remains. Guided by this literature review, and the ideal system requirements outline in Section 4, in Section 6 we propose tangible research suggestions spanning short, medium, and long-term timeframes.

6. Path Ahead

In this section, we present a comprehensive research and development roadmap for streaming analytics edge systems. We categorize our proposed projects based on their respective timeframes: short term (1-3 years), medium term (2-5 years), and long term (3-10 years).

6.1. Short Term Research

During the short term, our focus will be on integrating and extending existing ideas from the literature to create robust end-to-end edge video analytic systems. These research projects aim to combine various approaches proposed in previous works, resulting in practical solutions that can be implemented within the next three years.

Model tuning: The current body of research has investigated model tuning, adjusting it according to operating conditions such as resource and bandwidth fluctuations. However, we believe that these separate methodologies ought to be fused together. Specifically, attention must be given to situations where high video activity demands extra processing and low bandwidth availability necessitates careful consideration of communication-computation trade-offs. We believe AWSStream [64], Chameleon [65], CASVA [83], and YODA [83] would be good starting points for this research direction.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

Resource sharing: Another critical research domain that needs further exploration is resource sharing within Video Analytic Pipelines (VAPs). With multiple cameras in operation, each camera stream will necessitate its own distinct VAP, leaving a higher level application to consolidate the output. Although the Gemel [87] project has made strides in the realm of model sharing, additional methods for operating multiple coexisting VAPs warrant further investigation.

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 the developmental burdens of application creators. Open-source project such as KNative [92] and OpenFaaS [93] have been introduced for cloud-based applications; however, their suitability for various edge hardware types remains to be examined.

Data plane: Enhanced system software support for data plane management needs further research. In cloud-based environments, data replication with a replication factor of three is common to support fault-tolerant operations. On the edge, the data may have to be duplicated an arbitrary number of times, contingent on how Directed Acyclic Graphs (DAGs) comprising the VAP are scheduled and shared. The data plane system software should seamlessly abstract this operation from the developer. An additional challenge lies in the need to persist this replicated data for stateful applications. Replication inevitably brings data consistency issues to the forefront - a problem that becomes even more complex at the edge due to limited bandwidth and unreliable connectivity. One significant contribution in this field is Shepherd [55], a stream processing framework that sets itself apart from cloud stream projects like Apache Storm by effectively minimizing downtime during reconfiguration. This feature proves essential for the frequent readaptations required at the dynamic edge.

Testbeds: Current testbeds employ Virtual Machines (VMs) to emulate edge nodes. Emulation allows execution of VAP 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 [94] to the emulation platform.

6.2. 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 research could leverage the related body of work done in other areas of computing, but would require 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. 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 [95]. 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 [96]. The security triad of Confidentiality, Integrity, and Availability has been explored extensively for cloud and computing [97]. In their review on edge computing security, Xiao et. al. [98] have identified weak computation power, OS and protocol heterogeneity, attack unawareness, and coarse grained access control as key differences between the cloud and the edge from a security perspective. They list 6 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 4 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. [99], side channel attacks can be exploited to leak sensitive video information despite encryption. Defense strategies such as implementing fine grained access control [98], use of deep learning and machine learning algorithms to detect attacks [100], and use of hardware mechanisms for isolation of software components [101] are possible. However, their applicability to resource constrained edge nodes, and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 August 2023 doi:10.20944/preprints202308.0383.v1

its potential impact on VAP performance needs to be investigated systematically across multiple platforms.

Adversarial attacks could be directed at the deep learning VAP components such as classification, and object detection [102,103]. The goal of the adversarial attack is to insert small perturbations in the image to compromise the predictions of the deep learning based VAP component. Akhtar et. al. [102] provide a comprehensive review of adversarial attacks and defenses in computer vision. Proposed defenses against these attacks require model robustification [104], input modification for removing perturbations [105], and adding external detectors to the model [106]. 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 utmost importance to prevent leakage of unintended information. For example, an edge video analytics system for pedestrian safety might capture information regarding identities of individuals. The REVAMP^{2}T project [77] uses skeletal pose information to track pedestrian identity without storing any videos. The PECAM project [82] project proposes a novel Generative Adversarial Network to perform the privacy- enhanced securely-reversible video transformation at the edge nodes. Similar to security related counter measures, the cost of implementing privacy related computations on different types of research constrained edge devices need to be evaluated. Further, 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 [107] where the training data is 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 [86,88] required at the edge to mitigate model drift needs an in-depth investigation.

Human factors, reliability, and sustainability: 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 [108], and be provided with suitable system abstractions for VAP deployments. Operators should be 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 a API-driven model that enables developers and operators to interact with infrastructure programmatically, and at scale [109]. AIOps AIOps (Artificial Intelligence for IT Operations) [90,110], 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) [111]. Similar capabilites need to be developed to succesfully 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, deployment images) over bandwidth limited networks.

6.3. 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 project with a long time frame given the complexity of the problem, and the many stakeholders that need to be involved.

Real-world data sets: 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.

 doi:10.20944/preprints202308.0383.v1

Transactional data refers to the queries issued on the edge system by users and operators, while operational data refers to resource metrics, application logs, query traces, failure statuses. This would allow researchers to gain an understanding of real-world systems, and direct their efforts to 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. Further, updating of 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 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 [112]. As the technical community progresses towards 6G standards, with an anticipated initial rollout around 2030 [113], boasting impressive capabilities of 1,000 Gbps bandwidth and 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.

7. 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 [114]. Over the last few months the services of these LLMs models have been made available to the general public through services such as OpenAI's ChatGPT [115], Microsoft Bing AI [116], and Google's Bard [115]. More recently, Meta has made available it's LLama 2 LLM available for free download [117] potentially allowing the broader technical community to specialize these models for specific tasks based on training with proprietary data. In the edge vision analytics domain, we believe that these LLMs would be incorporated as a part of the VAPs 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 [118]. 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 [119]. Consequently, this enables a lightweight, portable, and highly secure alternative to container-based implementations of microservices, offering significant advantages, especially at the edge. It should be noted that development of the Wasm System Interface (WASI) is still ongoing [120].

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 [121] and the United States White House statement on responsible AI research, development, and deployment [122]. 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.

8. Conclusions

The widespread adoption of streaming video analytics 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. Diverging from prior reviews, our approach examines each system holistically, enabling a comprehensive assessment of 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, 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. By shedding light on the existing gaps and offering guidance for future investigations, this paper aims to contribute to the advancement and successful deployment of edge-based streaming video analytics systems.

References

1. PRNewswire. Artificial Intelligence (AI) Camera Market to Grow at a CAGR of 12.04 https://finance.yahoo.com/news/artificial-intelligence-ai-camera-market-100000236.html, Last accessed on July, 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. *Proceedings of the IEEE* **2019**, *107*, 1675–1696.
7. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal* **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. https://www.jvsg.com/storage-bandwidth-calculator/, Last accessed on July, 2023.
10. General Data Protection Regulation (GDPR), 2016. https://gdpr-info.eu/l, Last accessed on July, 2023.
11. Guiding Principles on Government Use of Surveillance Technologies), 2023. https://www.state.gov/wp-content/uploads/2023/04/Guiding-Principles-on-Government-Use-of-Surveillance-Technologies.pdfl, Last accessed on July, 2023.
12. Xu, R.; Razavi, S.; Zheng, R. Deep Learning-Driven Edge Video Analytics: A Survey. *arXiv preprint arXiv:2211.15751* **2022**.
13. Hu, M.; Luo, Z.; Pasdar, A.; Lee, Y.C.; Zhou, Y.; Wu, D. Edge-Based Video Analytics: A Survey. *arXiv preprint arXiv:2303.14329* **2023**.
14. Goudarzi, M.; Palaniswami, M.; Buyya, R. Scheduling IoT applications in edge and fog computing environments: a taxonomy and future directions. *ACM Computing Surveys* **2022**, *55*, 1–41.
15. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet of Things Journal* **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. *Proceedings of the IEEE* **2019**, *107*, 1537–1562.

17. Chen, J.; Ran, X. Deep learning with edge computing: A review. *Proceedings of the 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. *Proceedings of the 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. *Machine Vision and Applications* **2008**, *19*, 315–327.
20. 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 Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, 2020, pp. 126–133.
21. Szeliski, R. *Computer vision: algorithms and applications*; Springer, 2022.
22. TensorFlow 2 Detection Model Zoo. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.mdr/, Last accessed on July, 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 Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, 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 Transactions on Computers* **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. *Journal of Big Data Analytics in Transportation* **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 Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video, 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), 2019, pp. 1–6. https://doi.org/10.1109/ICB45273.2019.8987334.
31. Rashwan, H.A.; Solanas, A.; Puig, D.; et al.. Understanding Trust in Privacy-Aware Video Surveillance Systems. *International Journal of Information Security* **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.; et al.. Wildlife Insights: A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet. *Environmental Conservation* **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 Transactions on Industrial Informatics* **2019**, *16*, 5938–5947.
35. Ahmad, H.M.; Rahimi, A. Deep Learning Methods for Object Detection in Smart Manufacturing: A Survey. *Journal of Manufacturing Systems* **2022**, *64*, 181–196.
36. Kirkpatrick, K. Tracking Shoppers. *Communications of the 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.; 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, 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. *Frontiers in the Internet of 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 Transactions on Cloud Computing* **2022**.
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. *Communications of the ACM* **2010**, *53*, 50–58.
43. Pahl, C.; Jamshidi, P.; Zimmermann, O. Architectural principles for cloud software. *ACM Transactions on Internet Technology (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. *Communications of the ACM* **2021**, *64*, 76–84.
45. Docker: Accelerated, Containerized Application Development. https://www.docker.com/, Last accessed on July, 2023.
46. Kubernetes: Production-Grade Container Orchestration. https://kubernetes.io/, Last accessed on July, 2023.
47. Apache Kafka. https://kafka.apache.org/, Last accessed on July, 2023.
48. JetStream. https://docs.nats.io/nats-concepts/jetstream, Last accessed on July, 2023.
49. RabbitMQ. https://www.rabbitmq.com/, Last accessed on July, 2023.
50. Cattell, R. Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record* **2011**, *39*, 12–27.
51. Lightweight Kubernetes: The certified Kubernetes distribution built for IoT and Edge computing. https://k3s.io/, Last accessed on July, 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); USENIX Association: Renton, WA, 2019; pp. 929–946.
53. Sonbol, K.; Özkasap, Ö.; Al-Oqily, I.; Aloqaily, M. EdgeKV: Decentralized, Scalable, and Consistent Storage for the Edge. *Journal of Parallel and Distributed Computing* **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, 2022, pp. 40–53.
56. AWS Outposts Family. https://aws.amazon.com/outposts/, Last accessed on July, 2023.
57. Xu, M.; Liu, Y.; Liu, X. A Case for Camera-as-a-Service. *IEEE Pervasive Computing* **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 Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, 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 Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, 2015, pp. 426–438.
60. Microsoft Rocket for Live Video Analytics. https://www.microsoft.com/en-us/research/project/live-video-analytics/, Last accessed on July, 2023.
61. 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), 2017, pp. 377–392.
62. 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 Proceedings of the Second ACM/IEEE Symposium on Edge Computing, 2017, pp. 1–13.
63. 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, 2018, pp. 115–131.

64. Zhang, B.; Jin, X.; Ratnasamy, S.; Wawrzynek, J.; Lee, E.A. Awstream: Adaptive wide-area streaming analytics. In Proceedings of the Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, 2018, pp. 236–252.
65. Jiang, J.; Ananthanarayanan, G.; Bodik, P.; Sen, S.; Stoica, I. Chameleon: scalable adaptation of video analytics. In Proceedings of the Proceedings of the 2018 conference of the ACM special interest group on data communication, 2018, pp. 253–266.
66. 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, 2018, pp. 159–173.
67. Liu, P.; Qi, B.; Banerjee, S. Edgeeye: An edge service framework for real-time intelligent video analytics. In Proceedings of the Proceedings of the 1st international workshop on edge systems, analytics and networking, 2018, pp. 1–6.
68. Salehe, M.; Hu, Z.; Mortazavi, S.H.; Mohomed, I.; Capes, T. Videopipe: Building video stream processing pipelines at the edge. In Proceedings of the Proceedings of the 20th international middleware conference industrial track, 2019, pp. 43–49.
69. Canel, C.; Kim, T.; Zhou, G.; Li, C.; Lim, H.; Andersen, D.G.; Kaminsky, M.; Dulloor, S. Scaling video analytics on constrained edge nodes. *Proceedings of Machine Learning and Systems* **2019**, *1*, 406–417.
70. 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 Proceedings of the 10th ACM Multimedia Systems Conference, 2019, pp. 36–47.
71. Hsu, K.J.; Bhardwaj, K.; Gavrilovska, A. Couper: Dnn model slicing for visual analytics containers at the edge. In Proceedings of the Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, 2019, pp. 179–194.
72. 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, 2019, pp. 1270–1278.
73. Zeng, X.; Fang, B.; Shen, H.; Zhang, M. Distream: scaling live video analytics with workload-adaptive distributed edge intelligence. In Proceedings of the Proceedings of the 18th Conference on Embedded Networked Sensor Systems, 2020, pp. 409–421.
74. 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 of Things Journal* **2019**, *7*, 800–817.
75. 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, 2020, pp. 55–69.
76. 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, 2020, pp. 110–124.
77. 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 of Things Journal* **2019**, *7*, 2591–2602.
78. Khochare, A.; Krishnan, A.; Simmhan, Y. A scalable platform for distributed object tracking across a many-camera network. *IEEE Transactions on Parallel and Distributed Systems* **2021**, *32*, 1479–1493.
79. 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), 2021, pp. 165–177.
80. Saurez, E.; Gupta, H.; Daglis, A.; Ramachandran, U. Oneedge: An efficient control plane for geo-distributed infrastructures. In Proceedings of the Proceedings of the ACM Symposium on Cloud Computing, 2021, pp. 182–196.
81. 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, 2021, pp. 148–164.
82. 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 Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, 2021, pp. 229–241.

83. 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, 2022, pp. 2168–2177.
84. 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 Proceedings of the 23rd ACM/IFIP International Middleware Conference, 2022, pp. 322–334.
85. 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. *IEEE/ACM Transactions on Networking* **2022**.
86. 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), 2022, pp. 119–135.
87. 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), 2023, pp. 973–994.
88. 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), 2023, pp. 917–932.
89. Shostack, A. *Threat Modeling: Designing for Security*; John Wiley & Sons, 2014.
90. Notaro, P.; Cardoso, J.; Gerndt, M. A Survey of AIOps Methods for Failure Management. *ACM Transactions on Intelligent Systems and Technology (TIST)* **2021**, *12*, 1–45.
91. ZeroMQ: An open-source universal messaging library. https://zeromq.org, Last accessed on July, 2023.
92. Knative is an Open-Source Enterprise-level solution to build Serverless and Event Driven Applications. https://knative.dev/docs/, Last accessed on July, 2023.
93. Serverless Functions, Made Simple. https://www.openfaas.com/, Last accessed on July, 2023.
94. ns3 Network Simulator. https://www.nsnam.org/, Last accessed on July, 2023.
95. Hack of '150,000 cameras' investigated by camera firm. https://www.bbc.com/news/technology-56342525, Published, 10 March 2021.
96. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; et al.. Understanding the Mirai Botnet. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), 2017, pp. 1093–1110.
97. Vacca, J.R., Ed. *Cloud Computing Security: Foundations and Challenges*; CRC Press, 2016.
98. Xiao, Y.; Jia, Y.; Liu, C.; Cheng, X.; Yu, J.; Lv, W. Edge Computing Security: State of the Art and Challenges. *Proceedings of the IEEE* **2019**, *107*, 1608–1631.
99. 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, 2016, pp. 1–9.
100. 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 of Things Journal* **2021**, *9*, 236–260.
101. Coppolino, L.; D'Antonio, S.; Mazzeo, G.; Romano, L. A comprehensive survey of hardware-assisted security: From the edge to the cloud. *Internet of Things* **2019**, *6*, 100055.
102. 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.
103. Serban, A.; Poll, E.; Visser, J. Adversarial examples on object recognition: A comprehensive survey. *ACM Computing Surveys (CSUR)* **2020**, *53*, 1–38.
104. Bai, T.; Luo, J.; Zhao, J.; Wen, B.; Wang, Q. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356* **2021**.
105. Guo, C.; Rana, M.; Cisse, M.; Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117* **2017**.
106. Qin, Y.; Frosst, N.; Sabour, S.; Raffel, C.; Cottrell, G.; Hinton, G. Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. *arXiv preprint arXiv:1907.02957* **2019**.

107. Yin, X.; Zhu, Y.; Hu, J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)* **2021**, *54*, 1–36.
108. Bastani, F.; Moll, O.; Madden, S. Vaas: video analytics at scale **2020**.
109. Leite, L.; Rocha, C.; Kon, F.; Milojicic, D.; Meirelles, P. A survey of DevOps concepts and challenges. *ACM Computing Surveys (CSUR)* **2019**, *52*, 1–35.
110. 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 Transactions on Software Engineering and Methodology (TOSEM)* **2020**, *29*, 1–24.
111. 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.
112. 5G network coverage outlook. https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/network-coverage, Last accessed on July, 2023.
113. Next-gen mobile internet — 6G — will launch in 2030, telecom bosses say, even as 5G adoption remains low. https://www.cnbc.com/2023/03/08/what-is-6g-and-when-will-it-launch-telco-execs-predict.html, Published 7 March, 2023.
114. 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 preprint arXiv:2206.07682* **2022**.
115. ChatGPT: get instant answers, find creative inspiration, and learn something new. https://openai.com/chatgptl, Last accessed on July, 2023.
116. Bing helps you turn information into action, making it faster and easier to go from searching to doing. https://www.bing.com/?/ai, Last accessed on July, 2023.
117. Introducing Llama 2 - The next generation of our open source large language model. https://ai.meta.com/llama/, Last accessed on July, 2023.
118. Web Assemblyl. https://webassembly.org/, Last accessed on July, 2023.
119. Containers vs. WebAssembly: What's the Difference? https://www.fermyon.com/blog/webassembly-vs-containers, Published March, 2022.
120. WebAssembly System Interface. https://github.com/WebAssembly/WASI, Published March, 2022.
121. Artificial Intelligence Act. https://artificialintelligenceact.eu, Last accessed on July, 2023.
122. Ensuring Safe, Secure, and Trustworthy AI. https://www.whitehouse.gov/wp-content/uploads/2023/07/Ensuring-Safe-Secure-and-Trustworthy-AI.pd, Published 21 July, 2023.