Live Video Analytics as a Service

Guilherme H. Apostolo, Pablo Bauszat, Vinod Nigade, Henri E. Bal, Lin Wang
Vrije Universiteit Amsterdam
{g.apostolo,p.bauszat,v.v.nigade,h.e.bal,lin.wang}@vu.nl

Abstract

Many private and public organizations deploy large numbers of cameras, which are used in application services for public safety, healthcare, and traffic control. Recent advances in deep learning have demonstrated remarkable accuracy on computer analytics tasks that are fundamental for these applications, such as object detection and action recognition. While deep learning opens the door for the automation of camera-based applications, deploying pipelines for live video analytics is still a complicated process that requires domain expertise in the fields of machine learning, computer vision, computer systems, and networks. The problem is further amplified when multiple pipelines need to be deployed on the same infrastructure to meet different users' diverse and yet dynamic needs. In this paper, we present a live-video-analytics-as-a-service vision, aiming to remove the complexity barrier and achieve flexibility, agility, and efficiency for applications based on live video analytics. We motivate our vision by identifying its requirements and the shortcomings of existing approaches. Based on our analysis, we present our envisioned system design and discuss the challenges that need to be addressed to make it a reality.

CCS Concepts: • **Computer systems organization → Embedded and cyber-physical systems**; • **Information systems → Data management systems**.

Keywords: live video analytics, service systems, machine learning, privacy preservation

ACM Reference Format:
Guilherme H. Apostolo, Pablo Bauszat, Vinod Nigade, Henri E. Bal, Lin Wang. 2022. Live Video Analytics as a Service. In *2nd European Workshop on Machine Learning and Systems (EuroMLSys'22), April 5–8, 2022, RENNES, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3517207.3526973

1 Introduction

Today, public places like airports, traffic intersections or tunnels, and nursing homes deploy a large number of cameras to support services such as security and safety, crowd and traffic control, and patient monitoring [18, 38, 39]. Traditionally, these cameras are static and the videos collected by them are streamed to big monitors in a security office and watched manually by humans to detect anomalies or recorded for offline investigation. The recent advances in deep learning (DL) have brought new hope to this tedious, cumbersome practice, where deep neural networks (DNNs) have shown numerous successful examples of outperforming humans on analytics tasks, including object recognition, tracking, and action recognition [5, 13, 31]. Complex video analytics tasks can be automatically accomplished by composing multiple DNNs together in a video processing pipeline that takes live videos streamed from cameras as input [6, 22, 29]. The pipeline is then provisioned either on an edge [2] or cloud [28] platform or on a hybrid one spanning both the edge and cloud [26] to meet real-time performance.

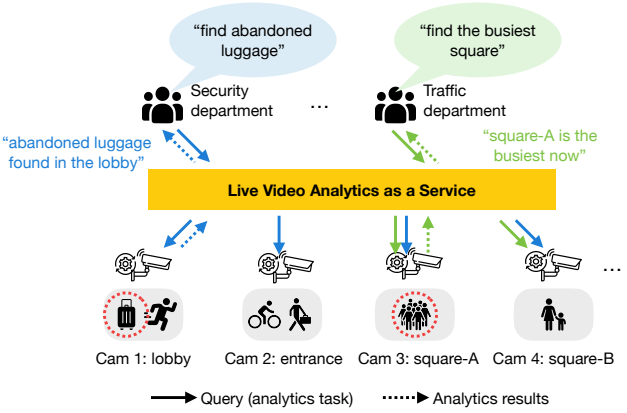

Figure 1. A live video analytics system at an airport serving multiple departments (i.e., users) with diverse needs.

However, the real-world adoption of DL-based live video analytics systems still faces many challenges [35]. One recent challenge is the provisioning of data and user privacy [4]. Another crucial challenge is the complexity in deploying and managing such systems [16, 30, 34]. Given a video analytics task, provisioning an efficient video processing pipeline requires extensive knowledge in multiple domains, including machine learning, computer vision, computer systems, and networks. Further, real-world scenarios often involve multiple video streams from a set of cameras and a diverse group of non-expert users with different needs, as depicted in Fig. 1.

Despite the rich literature on video analytics, most existing DL-based video analytics systems fall into the "piecemeal" category, lacking an *holistic, end-to-end* mindset in

EuroMLSys'22, April 5–8, 2022, RENNES, France

ACM ISBN 978-1-4503-9254-9/22/04.
https://doi.org/10.1145/3517207.3526973

their designs. Following the traditional store-and-analyze approach, systems like Blazeit [20] and Miris [3] focus on optimizing the performance of DL-based analytics tasks on *static* (data-at-rest) video databases, without support for live video streams. There exist also a considerable number of live video analytics systems that are tailored for a particular analytics task or scenario [17, 24, 26, 37]. These "domain-specific" solutions are hard to generalize and thus cannot accommodate the diverse and dynamic needs of different users. A few recent approaches aim at providing a more unified and versatile environment for live video analytics [34], but still lack support for several critical requirements (see requirement identification in §2 and gap analysis in §3).

In this paper, we present our vision for live video analytics, advocating for a concept known as live-video-analytics-as-a-service [1, 28] targeting the device-edge-cloud computing continuum. The goal of our vision is to remove the complexity barrier for the adoption of video analytics in real-world scenarios and to bring more flexibility, agility, and efficiency to the space. Ideally, a live video analytics system should be able to perform diverse, dynamic video analytics tasks across a large number of cameras in real-time, meeting the needs of different non-expert users in a unified manner while preserving privacy. *Our envisioned system* addresses this challenge by featuring the following design proposal:

- A *declarative interface* that allows users to describe their analytics tasks with customized camera scope and lifetime as well as requirements for latency and accuracy, requiring no domain expertise,
- An *adaptive analytics engine* that can automatically compose video analytics pipelines to serve user-specified analytics tasks, with privacy constraints enforced, and adapt these pipelines dynamically according to the changing computational environment and content in the video streams,
- An *efficient runtime system* that optimizes the provisioning of the video analytics pipelines across the device-edge-cloud computing continuum to achieve real-time analytics performance with high resource efficiency.

In the following, we first provide a thorough analysis for identifying the requirements of an ideal live-video-analytics-as-a-service system (§2). Then, we discuss in detail how existing approaches fall short in meeting these requirements (§3). Finally, we sketch the design of our envisioned system, highlighting the challenges in the design and pointing out future directions to explore (§4).

2 Identifying the Requirements

In this section, we dissect an ideal live video analytics system (as depicted in Fig. 1) and identify the requirements that need to be fulfilled by such a system towards achieving our vision of live-video-analytics-as-a-service. We will follow the general workflow of such a system guided by the following questions: (a) How do users interact with the system? (b) How to synthesize video analytics pipelines? (c) How to deploy and execute these pipelines?

2.1 User Interaction

The live video analytics system needs to serve various non-expert users who submit analytics tasks to one or more cameras and persist in the system for a customized time period. User interaction should meet the following requirements.

Intent-oriented interface. Users are unlikely to be experts on video analytics. Hence, it is critically important that users can describe their analytics tasks with a simple, high-level, intent-oriented interface. We advocate for a declarative interface that allows the users to focus on *what* they need ("find the busiest store"), instead of *how* it should be done. Despite its simplicity, the interface needs to be expressive enough to cover a wide variety of use cases.

Cross-camera exploration. Many real-world live video analytics use cases involve a large set of interconnected cameras. To support these use cases, the interface should provide syntax that allows describing cross-camera analytics tasks. This is essential for tracking-based tasks where the tracked objects may move across cameras. The interface should also support customizing the scope of a cross-camera analytics task, e.g., specifying the location or subset of cameras to consider.

Temporality. Depending on the user's needs, a task may need to be performed just once, for a specific period of time, or indefinitely. Taking the scenario of Fig. 1 as an example, the security department may submit an analytics task "find abandoned luggage" to run indefinitely, while the traffic department may require the task "find the busiest square" to run only during rush hours. The interface should provide syntax to support such temporality requirements. Tasks may also have an event matching windows where the analysis is continuously made over fixed intervals of a video stream [33]. Note that windowing is independent of task lifetime.

2.2 Video Analytics Pipeline Synthesis

Since the users will only declare their task goals, the live analytics system must automatically synthesize video analytics pipelines that consist of DL models. The synthesis procedure needs to take into account the following aspects.

Cross-task sharing. Often, video analytics pipelines may share DL models between user tasks. This becomes possible when tasks want to apply the same DL model to the same input, which frequently occurs for models that "pre-process" the video streams early on in the pipelines. The pipeline synthesis procedure should eliminate these redundancies during deployment as much as possible by constructing joint pipelines for multiple tasks to improve resource efficiency.

Content-awareness. The synthesized pipeline for a task should also adapt based on the content of the video streams.

The motivation for content adaptation is twofold. First, DL models tend to show varying accuracy for different video contents (e.g., due to changing numbers of objects or lighting conditions) and, thus, the pipeline should select the most accurate model among the functionally-equivalent ones with respect to the current condition. Second, the set of cameras that contribute to a task may change over time (e.g., for the analytics task of "tracking a unique object", other camera feeds can be ignored once the object has been detected in a specific camera). Being *adaptive* to these dynamic factors can improve the overall efficiency of the system.

PTZ support. Modern cameras increasingly support "pan-tilt-zoom" (PTZ) features that allow them to change their field and angle of view when instructed. However, current systems require humans to send these instructions manually or execute a preset of instructions in a timely manner [23]. We argue the need for a "human-out-of-the-loop" design, where the system can take advantage of the PTZ features and actuate automatically. For example, the system can automatically instruct a camera to zoom in when tracking a slow-moving object. These actuation decisions need to be coordinated between all user tasks.

Privacy preservation. Users of a video analytics system can have different privileges to access the contents of the video streams. For example, the security department may have full access to the camera feeds, while the traffic department cannot access sensitive information such as human faces. This requires the system to support the following privacy-preserving features: (a) task admission control based on the user role, and (b) video privacy-preserving techniques (such as blurring and generative adversarial networks [32]). Ideally, analytics tasks should be performed at the smallest information granularity without violating privacy rules.

2.3 Pipeline Deployment and Execution

The synthesized video analytics pipelines for all submitted tasks need to be deployed and executed efficiently, meeting individual real-time performance and accuracy goals. In particular, a system for deployment and execution needs to consider the following aspects.

Locality. The computing infrastructure (spanning the camera device, the edge, and the cloud) constitutes a continuum that presents a large space for locality-capability tradeoffs. The system should leverage such tradeoffs when deploying the operators from the video analytics pipelines in order to make the most out of the available resources.

Environment-awareness. The edge and cloud platforms are typically accessed via a network that exhibits variable throughput and latency [37]. The deployment and execution of pipelines should be adapted continuously in order to handle dynamic changes in the environment.

Performance guarantees. The system needs to meet the performance goals for different tasks, e.g., tracking-based tasks need to be performed in real-time, while a delay of a few seconds might be acceptable when responding to search-based tasks. Generally, service-level objectives (SLOs) for latency and accuracy are specified individually per task and need to be enforced during the pipeline deployment and execution process.

3 Identifying the Gap

Video analytics are applied in various domains and have therefore been widely studied. In the following, we present a detailed gap analysis on how existing systems, to the best of our knowledge, still fall short in meeting the previously discussed requirements. An overview of our analysis is shown in Tab. 1. At a high level, we divide existing video analytics systems into the following four major categories: DL-based video databases, DL inference serving systems, domain-specific video analytic pipelines, and generic video analytics engines. Requirements that do not apply to some systems because of differences in target scenarios are marked by "–". System requirements are marked as partially supported (◐) if (a) they support queries that take input from multiple cameras but do not exploit redundancy across cameras (cross-camera requirement); (b) they present event-matching windows but no lifetime support for queries (temporality requirement) ; (c) they support pipelines for multiple tasks but do not eliminate their redundancies (cross-task requirement); (d) they guarantee privacy, but only at a coarse-grained level (privacy requirement).

DL-based video databases store static video data and allow users to perform analytics tasks in the form of "queries" over the stored videos [3, 4, 19–21]. These systems typically provide a declarative language (often a variant of SQL) for submitting queries. These queries are generally one-shot operations applied over (a part of) the stored videos and do not consider any latency SLO guarantees. While queries may look at video feeds from multiple cameras, support for cross-camera tracking is lacking. Video database systems focus on optimizing query execution time through cross-task optimizations (e.g., caching intermediate results of a query for subsequent ones), content-awareness (e.g., adapting the query execution based on previously explored content), and data pre-processing techniques (e.g., NoScope [21] uses previously processed video annotations to train proxy models that are cheaper for filtering video frames). Consequently, these systems are powerful for offline video analytics but not directly suitable for the live video setting. Regarding user privacy, video database systems are designed with a binary mentality (no access or full access) that lacks fine-grained privacy control. An exception is Privid [4] which enforces duration-based differential privacy, but only to aggregate queries.

DL inference serving systems provide inference services for DNN pipelines and multiple users with the goal of high throughput and accuracy while meeting latency

Table 1. Gap analysis for existing systems and our identified requirements.

	Interface	Cross-camera	Temp.	Cross-task	Content-aware	PTZ	Privacy	Locality	Env.-aware	SLO
DL-based video database systems										
Viva [19]	ⓓ	◐	○	◐	●	–	○	C	–	○
Blazeit [20]	ⓓ	◐	○	◐	●	–	○	C	–	○
Miris [3]	ⓓ	◐	○	○	●	–	○	C	–	○
Privid [4]	ⓓ	◐	○	◐	●	–	◐	C	–	○
DL inference serving systems										
Clipper [7]	ⓘ	–	○	●	○	–	○	C	–	●
Nexus [29]	ⓘ	–	○	●	○	–	○	C	–	●
InferLine [6]	ⓘ	–	○	●	○	–	○	C	–	●
Domain-specific video analytics pipelines										
Awstream [37]	ⓘ	○	◐	◐	○	○	○	E-C	●	●
Caesar [24]	ⓓ	◐	◐	◐	○	○	○	D-E-C	○	●
Clownfish [26]	ⓘ	○	◐	○	○	○	○	E-C	●	●
Amadeus [8]	ⓘ	○	◐	◐	○	○	●	E-C	○	○
Spatula [17]	ⓘ	●	◐	○	○	○	○	D	●	○
Distream [36]	ⓘ	◐	◐	◐	○	○	○	E-D	●	●
Generic video analytics engines										
Gnosis [34]	ⓓ	◐	◐	◐	○	○	○	E-C	●	●
Nguyen et al. [9]	ⓓ	●	◐	◐	○	○	○	D-E-C	●	○

ⓓ: declarative, ⓘ: imperative, ●: full support, ◐: partial support, ○: no support, –: n/a, D: device (camera), E: edge, C: cloud

SLOs [6, 7, 29]. These systems expose an imperative interface for users to describe their DNN inference pipelines that are executed on a shared cloud platform equipped with high-end accelerators (generally GPUs and TPUs). While video analytics is typically treated as a representative use case, DL inference serving systems lack several essential features for live video analytics (e.g., temporality, content-awareness, camera actuation, privacy, and locality).

Domain-specific video analytics pipelines are systems that specialize in one specific live video analytics task by deploying a tailor-made DL pipeline [8, 17, 24, 26, 36, 37]. Development tools such as Microsoft Rocket [25], and NVIDIA DeepStream [27] have been introduced to simplify the development of such DL pipelines. These systems are highly optimized for a specific computer analytics task such as object detection [37] or action recognition [24, 26], and primarily focus on the efficient provisioning of a fixed-function pipeline to meet real-time performance. Consequently, they are hard to generalize to support arbitrary or multiple video analytics tasks. Some of these systems explore cross-camera similarities to reduce the computational workload [17], or expose fine-grained control on user privacy privileges [8]. However, they cannot satisfy the diverse and dynamic needs of different users and fall short in meeting the requirements for cross-task optimization, content-awareness, and PTZ actuation.

Generic video analytics engines treat each analytics task as a stream of operators (e.g., DL models) that need to be executed in order to transform raw video data from cameras into the results requested by users [9, 34]. Video analytics engines like Gnosis [34], and the approach proposed by Nguyen et al. [9] allow users to express queries with SQL-like languages. However, no support for automatic synthesis of video analytics pipelines is provided. While these systems are the closest to the live-video-analytics-as-a-service vision, they do not account for key requirements, including content-awareness, privacy preservation, and PTZ actuation. Recently, Yi et al. [35] presented a vision for live video analytics that incorporates cross-camera and cross-task support. However, their design lacks several important features that we identified as critical such as intent-oriented interface, content-awareness, and privacy preservation.

4 Vision and Challenges

After discussing the requirements of video analytics systems and the gap in state-of-the-art approaches, we now present our vision for live-video-analytics-as-a-service. In the following, we will sketch the design of our envisioned system and identify the challenges that call for future exploration.

4.1 System Design Sketch

A high-level overview of the key components of our envisioned system is sketched in Fig. 2. Our system consists of three conceptual layers that take user-specified video analytics tasks as input through a declarative interface, synthesizes video analytics pipelines composed of DL models, and deploys and executes these pipelines on the computing platform, meeting user-provided performance objectives.

4.1.1 A declarative interface. We envision the use of a declarative high-level language for users to specify their tasks as queries without requiring any domain expertise on

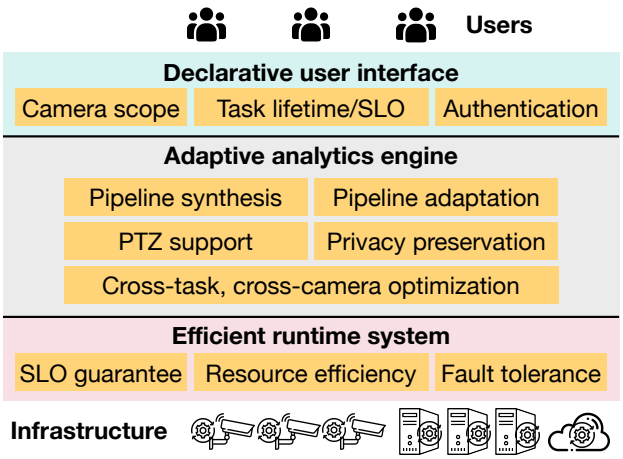

Figure 2. Sketch of the layered architecture of our envisioned live-video-analytics-as-a-service system.

video analytics. Our language design allows users to specify complex cross-camera analytics tasks across a customized set of cameras (e.g., based on the camera location) and supports syntax for specifying SLO requirements. As a novel feature, the language also allows users to declare the *lifetime of tasks* (temporality requirement), which can be done in two ways: (a) explicitly by time conditions like "for the next 20 min" or "indefinitely," or (b) implicitly by content-dependent conditions like "until a car has stopped for 10 min". To illustrate, we list four basic types of queries that are supported by our design:

- **Object Recognition** queries return the set of cameras that have a specified object in it (e.g., " find people with covered face in the stadium's grandstand for the *next two hours*")
- **Limit** queries return a set of cameras that have at least a certain number of objects in it (e.g., "show every camera from the highway with more than four SUVs in it for the *next 30 min*")
- **Tracking** queries return a set of cameras that track a pre-specified object in it (e.g., "follow the car with the global id of 102 *for 10 minutes*")
- **Action recognition** queries return a set of cameras where a specific action is happening in it (e.g., "show cameras from stores where there are people shoplifting *during the opening hours*")

The user interface also includes an authentication service for providing user credentials to the system.

4.1.2 An adaptive analytics engine.

Our system includes an adaptive analytics engine, which takes as input the queries and automatically synthesizes video analytics pipelines. The synthesis process is dynamic, i.e., the synthesized pipelines are continuously adapted based on the status of the system's environment, and the current content of the camera feeds.

Automatic video analytics pipeline synthesis. The analytics engine automatically generates a pipeline for each received analytics task. It identifies which DL models are needed, to which cameras they are applied to, and how they are chained to accomplish the tasks. It does so by applying optimizations (e.g., for DL model de-duplication) that account for cross-task and cross-camera similarities, as well as task lifetimes. Finally, the engine can leverage the type and requested information of a query to incorporate privacy-preserving operators that provide privacy guarantees in a more fine-grained way (e.g., enforce duration-based differential privacy for limit queries [4]).

Pipeline adaptation. The engine adapts to changing environmental conditions and video contents by continuously re-synthesizing the most suitable pipelines for all tasks collectively. Different DL models may be preferred for different environmental conditions to maximize accuracy while respecting the user SLOs. Content-awareness, which is beneficial in multi-camera scenarios under changing conditions (as discussed in §2.2), is achieved by the engine through the following aspects: (a) implicit lifetime of tasks steered by content-dependent conditions, (b) cross-camera object movements, and (c) PTZ support where cameras can actuate based on the detected content.

4.1.3 An efficient runtime system.

Finally, our design features a runtime system that is responsible for deploying and executing the pipelines generated by the analytics engine on the available compute resources. The goal is to collectively meet the performance requirements (e.g., SLOs) for all users while achieving maximum resource efficiency. The runtime system accounts for the variability of the underlying compute (e.g., heterogeneous camera hardware) and network platforms (e.g., changing network throughput) and dynamically adapts the pipeline deployment [37]. Such adaptation can also explore tradeoff "knobs" like the DL model batch size [29] and DNN architecture or variant [11] for a particular operator in the pipeline. The runtime system also deals with the operators' reliability (e.g., recovering a faulty operator) and network connectivity (e.g., camera connection drops).

4.2 Challenges and Discussion

Incorporating the aforementioned new features and requirements into a video analytics system introduces various new challenges for the system design. In the following, we describe those challenges and some possible features and techniques that can be leveraged to address them.

How to explore cross-camera optimizations? The engine can explore cross-camera similarities and identify overlapping fields of view to reduce the amount of video data that needs to be processed. However, identifying cameras with overlapping fields of view is a challenging problem [10, 12]. The engine requires new lightweight techniques

to efficiently identify similarities for large-scale deployments where the amount of cameras is considerable. The problem becomes even more challenging when overlapping fields of view change over time due to the camera's PTZ actuation. Applying clustering techniques with cheaper feature detection algorithms such as in [12] allied with the camera's subset expression might be a solution to reduce the cross-camera similarity search space. Furthermore, cameras might experience failures during execution (e.g., power outage or connection loss), which need to be considered when identifying similarities. A possible solution could be to utilize camera actuation to find new transformations for cameras that can cover the region lost by the faulty cameras.

How to explore cross-query optimizations? A typical approach for exploiting cross-query redundancies is sharing of operators (i.e., DL models) and their outputs [15]. In our design queries have individual lifetimes and, thus, distinct arrival and finishing times. An adaption engine that utilizes operator sharing but is agnostic of query lifetime might make sub-optimal placement and scheduling decisions. Our engine must consider lifetime during the decision-making process leading to a more complicated optimization process.

Another popular approach for improving resource utilization when serving multiple users at the same time is input batching for the DNN models [7, 29]. In cases where the user SLOs provide some latency slack, an increase in individual query latency is acceptable, and input batching can improve the system's overall throughput. Previous approaches decide the best batch size by considering user SLOs and the amount of video frames that need to be analyzed for a specific DNN operator. However, these frameworks do not consider an edge-cloud scenario where network bandwidth and latency vary over time. Our runtime system now must also consider environmental conditions for scheduling, operator placement as well as choosing batch sizes.

How to enable adaptivity based on video content? Our declarative interface provides the analytics engine with information about the semantic objects that are required to match user queries. When dynamically adapting the pipelines for these queries, however, more information than these provided objects might be necessary. Identifying changes in the camera content might require additional computer vision operators to additionally extract object sizes, types, quantities, and camera conditions. There are some possible techniques to explore, such as background subtraction to detect lighting conditions of the background [14], or using specialized models to better detect the appearances of a certain object based on previously executed frames [21]. However, any additional operator needs to be applied with high frequency to the video data and, thus, should be as lightweight as possible to avoid introducing computational overhead.

How to reconcile PTZ control decisions? Once the analytics engine receives a query, it has to automatically understand the task and how camera actuation for that task needs to be performed, which may not be straightforward. Additionally, PTZ actuation for a single camera might conflict when serving multiple tasks simultaneously. For example, if multiple queries require a camera to analyze the same area and one of those queries instructs the camera to zoom in onto a certain object, the camera will lose the broader view of the area and potentially compromise results for other queries. An engine that is aware of this problem could compensate by negotiating PTZ actuation between queries or by exploring cross-camera similarities to identify alternative cameras that can provide the missing information. Alternatively, the engine can decide to prioritize actuation from users with higher privileges or more important SLOs.

How to achieve fine-grained privacy protection? Our system utilizes privacy-preserving operators to create secure streams of video data and semantic objects. Care must be taken when sharing these streams between users because they might not share the same access privileges. Further, those privacy-aware operations are generally more computationally expensive and ultimately lead to larger delays during pipeline execution [8]. Our adaptation engine must acknowledge varying user privileges as well as additional latency costs from choosing privacy-aware operators when synthesizing and adapting pipelines. Overall, this leads to a much more complicated decision-making process. Finally, the runtime has to provide the best operation scheduling and computation offloading strategy while considering privacy requirements (e.g., by executing queries with higher privacy concerns more often on the edge instead of in the cloud).

5 Conclusion

Driven by the recent use cases of live video analytics, we tackle the unprecedented complexity of deploying live video analytics pipelines when facing users with diverse and dynamic needs. Based on identifying the requirements and analyzing the pitfalls of existing systems, we present our vision of a live-video-analytics-as-a-service system that features a declarative interface with lifetime support, an adaptive analytics engine that synthesizes and adapts video analytics pipelines automatically applying cross-task, cross-camera optimizations and supporting privacy and PTZ features, and an efficient runtime system to handle the execution of pipelines across the device-edge-cloud computing continuum, achieving both performance guarantee and resource efficiency.

We highlight that our vision poses multiple important challenges that would require further exploration by the research community.

Acknowledgments

This work is part of the Real-Time Video Surveillance Search project (grant number 18038), financed by the Dutch Research Council (NWO).

References

[1] Amazon. 2022. Amazon Rekognition: Automate your image and video analysis with machine learning. https://aws.amazon.com/rekognition/. Accessed: 05-02-2022.

[2] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. 2017. Real-Time Video Analytics: The Killer App for Edge Computing. *Computer* 50, 10 (2017), 58–67. https://doi.org/10.1109/MC.2017.3641638

[3] Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan, Mohammad Alizadeh, Hari Balakrishnan, Michael J. Cafarella, Tim Kraska, and Sam Madden. 2020. MIRIS: Fast Object Track Queries in Video. In *ACM SIGMOD*. 1907–1921.

[4] Frank Cangialosi, Neil Agarwal, Venkat Arun, Junchen Jiang, Srinivas Narayana, Anand Sarwate, and Ravi Netravali. 2022. Privid: Practical, Privacy-Preserving Video Analytics Queries. In *USENIX NSDI*.

[5] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE CVPR*. 4724–4733. https://doi.org/10.1109/CVPR.2017.502

[6] Daniel Crankshaw, Gur-Eyal Sela, Xiangxi Mo, Corey Zumar, Ion Stoica, Joseph Gonzalez, and Alexey Tumanov. 2020. InferLine: latency-aware provisioning and scaling for prediction serving pipelines. In *ACM SoCC*. 477–491.

[7] Daniel Crankshaw, Xin Wang, Giulio Zhou, Michael J. Franklin, Joseph E. Gonzalez, and Ion Stoica. 2017. Clipper: A Low-Latency Online Prediction Serving System. In *USENIX NSDI*. 613–627.

[8] Sandeep D'Souza, Victor Bahl, Lixiang Ao, and Landon P. Cox. 2020. Amadeus: Scalable, Privacy-Preserving Live Video Analytics. *CoRR* abs/2011.05163 (2020). arXiv:2011.05163 https://arxiv.org/abs/2011.05163

[9] Manh Nguyen Duc, Anh Lê Tuán, Manfred Hauswirth, and Danh Le Phuoc. 2021. Towards autonomous semantic stream fusion for distributed video streams. In *ACM DEBS*. 172–175. https://doi.org/10.1145/3465480.3467837

[10] Hongpeng Guo, Shuochao Yao, Zhe Yang, Qian Zhou, and Klara Nahrstedt. 2021. CrossRoI: cross-camera region of interest optimization for efficient real time video analytics at scale. In *MMSys '21: 12th ACM Multimedia Systems Conference, Istanbul, Turkey, 28 September 2021 - 1 October 2021*, Özgü Alay, Cheng-Hsin Hsu, and Ali C. Begen (Eds.). ACM, 186–199. https://doi.org/10.1145/3458305.3463381

[11] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy. 2016. MCDNN: An Approximation-Based Execution Framework for Deep Stream Processing Under Resource Constraints. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2016, Singapore, June 26-30, 2016*, Rajesh Krishna Balan, Archan Misra, Sharad Agarwal, and Cecilia Mascolo (Eds.). ACM, 123–136. https://doi.org/10.1145/2906388.2906396

[12] Brandon Haynes, Maureen Daum, Dong He, Amrita Mazumdar, Magdalena Balazinska, Alvin Cheung, and Luis Ceze. 2021. VSS: A Storage System for Video Analytics. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, Guoliang Li, Zhanhuai Li, Stratos Idreos, and Divesh Srivastava (Eds.). ACM, 685–696. https://doi.org/10.1145/3448016.3459242

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE CVPR*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[14] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodík, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B. Gibbons, and Onur Mutlu. 2018. Focus: Querying Large Video Datasets with Low Latency and Low Cost. In *13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018, Carlsbad, CA, USA, October 8-10, 2018*, Andrea C. Arpaci-Dusseau and Geoff Voelker (Eds.). USENIX Association, 269–286.

[15] Chien-Chun Hung, Ganesh Ananthanarayanan, Peter Bodík, Leana Golubchik, Minlan Yu, Paramvir Bahl, and Matthai Philipose. 2018. VideoEdge: Processing Camera Streams using Hierarchical Clusters. In *2018 IEEE/ACM Symposium on Edge Computing, SEC 2018, Seattle, WA, USA, October 25-27, 2018*. IEEE, 115–131. https://doi.org/10.1109/SEC.2018.00016

[16] Samvit Jain, Ganesh Ananthanarayanan, Junchen Jiang, Yuanchao Shu, and Joseph Gonzalez. 2019. Scaling Video Analytics Systems to Large Camera Deployments. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications, HotMobile 2019, Santa Cruz, CA, USA, February 27-28, 2019*, Alec Wolman and Lin Zhong (Eds.). ACM, 9–14. https://doi.org/10.1145/3301293.3302366

[17] Samvit Jain, Xun Zhang, Yuhao Zhou, Ganesh Ananthanarayanan, Junchen Jiang, Yuanchao Shu, Paramvir Bahl, and Joseph Gonzalez. 2020. Spatula: Efficient cross-camera video analytics on large camera networks. In *IEEE/ACM Symposium on Edge Computing (SEC)*. 110–124. https://doi.org/10.1109/SEC50012.2020.00016

[18] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodík, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: scalable adaptation of video analytics. In *ACM SIGCOMM*. 253–266.

[19] Daniel Kang, Francisco Romero Peter Bailis, Christos Kozyrakis, and Matei Zaharia. 2022. VIVA: An End-to-End System for Interactive Video Analytics. (2022).

[20] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *Proc. VLDB Endow.* 13, 4 (2019), 533–546. https://doi.org/10.14778/3372716.3372725

[21] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Deep CNN-Based Queries over Video Streams at Scale. *Proc. VLDB Endow.* 10, 11 (2017), 1586–1597. https://doi.org/10.14778/3137628.3137664

[22] Ram Srivatsa Kannan, Lavanya Subramanian, Ashwin Raju, Jeongseob Ahn, Jason Mars, and Lingjia Tang. 2019. GrandSLAm: Guaranteeing SLAs for Jobs in Microservices Execution Frameworks. In *ACM EuroSys*. 34:1–34:16.

[23] Pratibha Kumari, Nikhil Nandyala, Allu Krishna Sai Teja, Neeraj Goel, and Mukesh Saini. 2020. Dynamic Scheduling of an Autonomous PTZ Camera for Effective Surveillance. In *17th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2020, Delhi, India, December 10-13, 2020*. IEEE, 437–445. https://doi.org/10.1109/MASS50613.2020.00060

[24] Xiaochen Liu, Pradipta Ghosh, Oytun Ulutan, B. S. Manjunath, Kevin S. Chan, and Ramesh Govindan. 2019. Caesar: cross-camera complex activity recognition. In *ACM SenSys*. 232–244. https://doi.org/10.1145/3356250.3360041

[25] Microsoft. 2022. Microsoft Rocket for Live Video Analytics. https://www.microsoft.com/en-us/research/project/live-video-analytics/. Accessed: 09-02-2022.

[26] Vinod Nigade, Lin Wang, and Henri E. Bal. 2020. Clownfish: Edge and Cloud Symbiosis for Video Stream Analytics. In *IEEE/ACM Symposium on Edge Computing (SEC)*. 55–69. https://doi.org/10.1109/SEC50012.2020.00012

[27] NVIDIA. 2022. DeepStream. https://developer.nvidia.com/deepstream-sdk. [Online; accessed 04-Feb-2022].

[28] Rishabh Poddar, Ganesh Ananthanarayanan, Srinath Setty, Stavros Volos, and Raluca Ada Popa. 2020. Visor: Privacy-Preserving Video Analytics as a Cloud Service. In *USENIX Security*. 1039–1056.

[29] Haichen Shen, Lequn Chen, Yuchen Jin, Liangyu Zhao, Bingyu Kong, Matthai Philipose, Arvind Krishnamurthy, and Ravi Sundaram. 2019. Nexus: a GPU cluster engine for accelerating DNN-based video analysis. In *ACM SOSP*. 322–337. https://doi.org/10.1145/3341301.3359658

[30] Can Wang, Sheng Zhang, Yu Chen, Zhuzhong Qian, Jie Wu, and Mingjun Xiao. 2020. Joint Configuration Adaptation and Bandwidth Allocation for Edge-based Real-time Video Analytics. In *39th IEEE*

Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020. IEEE, 257–266. https://doi.org/10.1109/INFOCOM41043.2020.9155524

[31] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. 2019. Fast Online Object Tracking and Segmentation: A Unifying Approach. In *IEEE CVPR*. 1328–1338.

[32] Hao Wu, Xuejin Tian, Minghao Li, Yunxin Liu, Ganesh Ananthanarayanan, Fengyuan Xu, and Sheng Zhong. 2021. PECAM: privacy-enhanced video streaming and analytics via securely-reversible transformation. In *ACM MobiCom '21: The 27th Annual International Conference on Mobile Computing and Networking, New Orleans, Louisiana, USA, October 25-29, 2021*. ACM, 229–241. https://doi.org/10.1145/3447993.3448618

[33] Piyush Yadav and Edward Curry. 2020. VidCEP: Complex Event Processing Framework to Detect Spatiotemporal Patterns in Video Streams. *CoRR* abs/2007.07817 (2020). arXiv:2007.07817 https://arxiv.org/abs/2007.07817

[34] Piyush Yadav, Dhaval Salwala, Felipe Pontes, Praneet Dhingra, and Edward Curry. 2021. Query-Driven Video Event Processing for the Internet of Multimedia Things. *Proc. VLDB Endow.* 14, 12 (2021), 2847–2850.

[35] Juheon Yi, Chulhong Min, and Fahim Kawsar. 2021. Vision Paper: Towards Software-Defined Video Analytics with Cross-Camera Collaboration. In *ACM SenSys*. 474–477. https://doi.org/10.1145/3485730.3493453

[36] Xiao Zeng, Biyi Fang, Haichen Shen, and Mi Zhang. 2020. Distream: scaling live video analytics with workload-adaptive distributed edge intelligence. In *ACM SenSys*. 409–421. https://doi.org/10.1145/3384419.3430721

[37] Ben Zhang, Xin Jin, Sylvia Ratnasamy, John Wawrzynek, and Edward A. Lee. 2018. AWStream: adaptive wide-area streaming analytics. In *ACM SIGCOMM*. 236–252. https://doi.org/10.1145/3230543.3230554

[38] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodík, Matthai Philipose, Paramvir Bahl, and Michael J. Freedman. 2017. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *USENIX NSDI*. 377–392.

[39] Qingyang Zhang, Hui Sun, Xiaopei Wu, and Hong Zhong. 2019. Edge Video Analytics for Public Safety: A Review. *Proc. IEEE* 107, 8 (2019), 1675–1696. https://doi.org/10.1109/JPROC.2019.2925910