

Received July 27, 2020, accepted August 8, 2020, date of publication August 17, 2020, date of current version August 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017135

Video Big Data Analytics in the Cloud: A Reference Architecture, Survey, Opportunities, and Open Research Issues

AFTAB ALAM^{ID}, IRFAN ULLAH^{ID}, AND YOUNG-KOO LEE^{ID}, (Member, IEEE)

Department of Computer Science and Engineering, Kyung Hee University (Global Campus), Yongin 1732, South Korea

Corresponding author: Young-Koo Lee (ykle@khu.ac.kr)

This work was supported by the Institute for Information and Communications Technology Promotion Grant through the Korean Government (MSIT) (SIAT CCTV Cloud Platform) under Grant R7120-17-1007.

ABSTRACT The proliferation of multimedia devices over the Internet of Things (IoT) generates an unprecedented amount of data. Consequently, the world has stepped into the era of big data. Recently, on the rise of distributed computing technologies, video big data analytics in the cloud has attracted the attention of researchers and practitioners. The current technology and market trends demand an efficient framework for video big data analytics. However, the current work is too limited to provide a complete survey of recent research work on video big data analytics in the cloud, including the management and analysis of a large amount of video data, the challenges, opportunities, and promising research directions. To serve this purpose, we present this study, which conducts a broad overview of the state-of-the-art literature on video big data analytics in the cloud. It also aims to bridge the gap among large-scale video analytics challenges, big data solutions, and cloud computing. In this study, we clarify the basic nomenclatures that govern the video analytics domain and the characteristics of video big data while establishing its relationship with cloud computing. We propose a service-oriented layered reference architecture for intelligent video big data analytics in the cloud. Then, a comprehensive and keen review has been conducted to examine cutting-edge research trends in video big data analytics. Finally, we identify and articulate several open research issues and challenges, which have been raised by the deployment of big data technologies in the cloud for video big data analytics. To the best of our knowledge, this is the first study that presents the generalized view of the video big data analytics in the cloud. This paper provides the research studies and technologies advancing the video analyses in the era of big data and cloud computing.

INDEX TERMS Big data, intelligent video analytics, cloud-based video analytics system, video analytics survey, deep learning, distributed computing, intermediate results orchestration, cloud computing.

I. INTRODUCTION

Videos are generated and uploaded regularly to the cloud. Many sources include CCTV, smartphones, drones, etc., are actively contributing to video generation leads to the evolution of Intelligent Video Analytics (IVA) and management systems. IVA is a domain that uses advanced computer vision technologies to process and extract insights from streaming or stored videos automatically. Over the past two decades, IVA is extensively rising from real-world needs driving by a broader range of application domains ranging from security

The associate editor coordinating the review of this manuscript and approving it for publication was Xin Luo^{ID}.

and surveillance to transportation and healthcare. The IVA market is anticipated to rise from 4.9 billion USD in 2020 to 11.7 billion USD by 2025 at a compound annual growth rate of 19.0% [1].

Video management and services providers such as Facebook [2], YouTube [3], and Netflix [4] are considered as valuable sources of large-scale video data. Along with these, various leading industrial organizations have successfully deployed video management and analytics platforms that provide more bandwidth and high-resolution cameras collecting videos at scale and has become one of the latest trends in the video surveillance industry. For example, more than 400 hours of videos are uploaded in a minute

on Youtube [5], and more than one hundred and seventy million video surveillance cameras have been installed in china only [6]. It has been reported that the data generated by various IoT devices will see a growth rate of 28.7% over the period 2018-2025, where surveillance videos are the majority shareholder [7].

Such an enormous video data is considered as “big data” because a variety of sources generates a large volume of video data at high velocity that holds high Value. Even though 65% of the big data shares hold by surveillance videos are monitored, but still, a significant portion of video data has been failed to notice [8]. That neglected data contain valuable information directly related to real-world situations. Video data provide information about interactions, behaviors, and patterns, whether its traffic or human patterns. However, handling such a large amount of complex video data is not worthwhile utilizing traditional data analytical approaches. Therefore, more comprehensive and sophisticated solutions are required to manage and analyses such large-scale unstructured video data.

Due to the data-intensive and resources hungry nature of large scale video data processing, extracting the insights from the video is a challenging task. A considerable size of video data poses significant challenges for video management and mining systems that require powerful machines to deal with large-scale video data. Moreover, a flexible solution is necessary to store and mine this large volume of video data for decision making. However, large-scale video analytics becomes a reality due to the popularity of big data and cloud computing technologies.

Cloud computing is an infrastructure for providing convenient and ubiquitous remote access to a shared pool of configurable computing resources. These resources can be managed with minimal management effort or service [9]. Big data technologies, such as Hadoop or Spark echo system, are software platforms designed for distributed computing to process, analyze, and extract the valuable insights from large datasets in a scalable and reliable way. The cloud is preferably appropriate to offer the big data computation power required for the processing of these large datasets. [10], Amazon web service [11], Microsoft Azure [12], and Oracle Big Data Analytics [13] are some examples of video big data analytics platforms. Large-scale video analytics in the cloud is a multi-disciplinary area, and the next big thing in big data, which opens new research avenues for researchers and practitioners.

This work aims to conduct a comprehensive study on the status of large scale video analytics in the cloud-computing environment while deploying video analytics techniques. First, this study builds the relationship between video big data and cloud computing and defines the terminologies that govern the study. Then service-oriented and a layered reference architecture have been proposed for large-scale video analytics in the cloud while focusing on architectural properties like reliability, scalability, fault-tolerance, extensibility, and intermediate results orchestration. Further, an intensive

survey has been conducted to project the current research trends in video analytics that encompass the taxonomy of video analytics approaches, and cloud-based scholarly and industrial study. Finally, open research issues and challenges are discussed, with a focus on proposed architecture, i.e., the deployment of an array of computer vision algorithms for large-scale videos in the cloud.

A. VIDEO BIG DATA, CLOUD COMPUTING, AND THEIR RELATIONSHIP

The term big data appeared and popularized by John R. Masey in the late 1990s [14], which refers to a large volume of data that are impractical to be stored, processed and analyzed using traditional data management and processing technologies [15]. The data can be unstructured, semi-structured, and structured data, but mostly unstructured data is considered. The definition of big data evolved and has been described in terms of three, four, or five characteristics. In literature, among these characteristics, three are shared, i.e., Volume, Velocity, and Variety, while the others are Veracity and Value [16]–[19]. Various video stream sources generate a considerable amount of unstructured video data on a regular bases and becoming a new application field of big data. The data generated by such sources are further subject to contextual analysis and interpretation to uncover the hidden patterns for decision-making and business purposes.

In the context of a large volume of video data, we specialize the generic big data characteristics. The size of data is referred to as Volume [20], but the majority of the shares, i.e., 65%, are held only by surveillance videos. The type of data generated by various sources such as text, picture, video, voice, and logs are known as Variety [20]. The video data are acquired from multimodal video stream sources, e.g., IP-Camera, depth camera, body-worn camera, etc., and from different geolocations, which augments the Variety property. The pace of data generation and transmission is known as Velocity [21]. The video data also possess the Velocity attribute, i.e., the Video Stream Data Source (VSDS) primarily produce video stream 24/7 and acquired by the data center storage servers. Veracity can be defined as the diversity of quality, accuracy, and trustworthiness of the data [22]. Video data are acquired directly from real-world domains and meet the Veracity characteristic. The Value refers to contextual analysis to extract the significant values for decision-making and business purpose [23], [24]. Video data has high Value because of its direct relation with real-word. Automatic criminal investigation, illegal vehicle detection, and abnormal activity recognition are some of the examples of Value extraction. Almost all the big data properties are dominated by the video data, which encourage us to give birth to Video Big Data.

These five characteristics impose many challenges on the organizations when embracing video big data analytics. Storing, scaling, and analyzing are some apparent challenges associated with video big data. To cope with these challenges, converged and hyper-converged infrastructure and software-defined storage are the most convenient solutions.

Distributed databases, data processing engines, and machine learning libraries have been introduced to overcome video big data management, processing, and analysis issues, respectively.

These big data technologies are deployed over a computer cluster to process and manage a massive amount of video data in parallel. A computer cluster may consist of few to hundreds, and even thousands of nodes work together as a single integrated computing resource, on different parts of the same program [25], [26]. Deploying an indoor computer cluster is an option for big data technologies, but hardware cost and maintenance issues are associated with it. An alternative solution can be cloud computing that elegantly reduces the costs associated with the management of hardware and software resources [27].

Typically, cloud services are provided on-demand in a “pay-as-you-go” manner for the conveniences of end-users and organizations [28]. Cloud computing follows the philosophy of the “as-a-Service” and offers its “services” according to different models, for example, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS) [9].

Under IaaS (e.g., Amazon’s EC2), the cloud service provider facilitates and allows the consumers to provision fundamental computing resources and deploy arbitrary software. In PaaS, the service provider provides a convenient platform enabling customers to develop, run, and manage applications without considering the complexities of building and maintaining the infrastructure. The examples of PaaS are Google’s Apps Engine and Microsoft Azure. In SaaS, applications (e.g., email, docs, etc.) are deployed on cloud infrastructure by service providers and allow the consumer to subscribe. These applications can be easily accessed from various client devices using a thin client or program interfaces [9].

B. RESEARCH OBJECTIVES AND CONTRIBUTIONS

This paper presents a detailed survey and review of cloud-based large-scale IVA. We also propose big data technological solutions for the challenges faced by IVA researchers and practitioners. The contributions of this paper are listed below:

- We standardize the basic nomenclatures that govern the IVA domain. The term video big data has been coined and clarified how it inherits the big data characteristics while establishing its relationship with cloud computing.
- We propose a distributed, layered, service-oriented, and lambda style [29] inspired reference architecture for large-scale IVA in the cloud. Each layer of the proposed Cloud-based Video Analytics System (CVAS) has been elaborated technologically, i.e., layerwise available big data technological alternatives. The base layer of the proposed architecture, i.e., video big data curation layer, is based on the notion of Intermediate Results (IR) orchestration, which can play a significant role in the

optimization of the IVA pipeline. Under the proposed architecture, we perform a thorough investigation of scalable traditional video analytics and deep learning techniques and tools on distributed infrastructures along with popular computer vision benchmark datasets.

- To the best of our knowledge, this is a first surveying video big data analytics, our research targets the most recent approaches that encompass broad IVA research domains like content-based video retrieval, video summarization, semantic-based approaches, and surveillance and security.
- We also investigate how the researchers exploit big data technologies for large-scale video analytics and show how the industrial IVA solutions are provided.
- In this study, we further write real word IVA application areas, which projects the significance of big data and cloud computing in IVA.
- Finally, we identify the research gap and list several open research issues and challenges. These research issues encompass orchestration and optimization of IVA pipeline, big dimensionality, online learning on video big data, model management, parameter servers, and distributed learning, evaluation issues and opportunities, IVA services statistics maintenance and ranking in the cloud, IVA-as-a-Service (IVAAA) and cost model, video big data management, and privacy, security and trust.

The remaining paper is organized as follows. Section II and III is about recent studies, and scope and nomenclature, respectively. Section IV discusses the proposed CVAS. Literature review has been presented in Section V and VI. The IVA applications can be seen in Section VII. In Section VIII several video big data challenges and opportunities are discussed. Finally, Section IX concludes this study.

II. RECENT STUDIES

Various studies have been conducted that discuss video analytics, which can be classified as IVA and Big data, as shown in Table 1. In the former class, the surveys overlook the big data characteristics and challenges of video analysis in the cloud. The majority of the work focuses on a specific domain of IVA and the application of video analyses. For instance, a comprehensive survey was presented by Liu *et al.* [30], and Olatunji *et al.* [31], where they have discussed IVA and its applications in the context of surveillance conventionally. Context-Based Video Retrieval (CBVR) was reviewed by Hu *et al.* [32], Patel *et al.* [33], and Haseyama *et al.* [34]. Similarly, vehicle surveillance systems in Intelligent Transportation System (ITS), abnormal behavior analytics, in surveillance videos were studied by Tian *et al.* [35], and [36], respectively.

Furthermore, big data studies were conducted from different perspectives, i.e., cloud computing, ML, mining, multimedia, and ITS. Hashem *et al.* [16] and Agrawal *et al.* [39] presented research issues and challenges on big data analytics in the cloud computing. Tsai *et al.* [38] and Khan *et al.* [37]

TABLE 1. Recent surveys related to video big data analytics.

Class	References	Target Domain	Pros	Cons
Int. Video Analytics	Liu <i>et al.</i> [30] Olatunji <i>et al.</i> [31]	Video analytics based surveillance	Comprehensive theory & applications of video analytics-based surveillance	-Lacking big data & cloud computing and its research issues - Not service oriented
	Hu <i>et al.</i> [32] Patel <i>et al.</i> [33] Haseyama <i>et al.</i> [34]	CBVR	- CBVR components, research trends and issues	-Lack of big data & cloud computing, & its research issues. -Limited to CBVR
	Khan <i>et al.</i> [37] Tsai <i>et al.</i> [38]	Vehicle surveillance systems in ITS	Generic layered architecture for video analytics-based vehicle surveillance	- Lacking big data and cloud computing techniques and challenges - Abstract architecture - Consider IVA for ITS
	Mabrouk <i>et al.</i> [36]	Abnormal behavior in surveillance videos	A generic architecture for abnormal behaviors in video surveillance systems	-Lacking big data and cloud computing discussion. -Limited to abnormal behavior
Big data	Hashem <i>et al.</i> [16] Agrawal <i>et al.</i> [39]	Cloud computing	A thorough overview of big data techniques, challenges and open issues in cloud computing	-Big data processing over cloud computing only. -Lack of video big data analytics, service-oriented architecture, research issues, and challenges.
	Khan <i>et al.</i> [37] Tsai <i>et al.</i> [38]	Big data technologies	Overview of big data techniques, opportunities, and challenges	Limited discussion on video big data analytics
	Zhou <i>et al.</i> [40]	Machine Learning (ML)	A framework of ML on big data, and open research issues	Lack of IVA discussion
	Che <i>et al.</i> [41]	Big data mining	A comprehensive overview on big data mining and challenges	Lack of video analytics and its research issues
	Pouyanfar <i>et al.</i> [5]	Multimedia	A comprehensive overview of multimedia data	Limited discussion on video big data analytics
	Zhu <i>et al.</i> [42]	ITS	-A study of big data analytics in ITS -Consider multimedia for ITS	Limited discussion of video big data in the context of ITS analytics
	H. Zahid <i>et al.</i> [43]	Telecommunications	A study of big data analytics in telecommunication.	Limited discussion of video big data in the context of telecommunication analytics

focus on big techniques and general applications along with challenges. Zhou *et al.* [40], Che *et al.* [41], and Pouyanfar *et al.* [5] studied and reported the research issues of big data practices concerning ML techniques, data mining algorithms, and multimedia data respectively. Zhu *et al.* [42] presented a comprehensive study on big data analytics in ITS, analytical methods and platforms, and categories of big data video analytics. In the study of big data, IVA has been overlooked, and minimal discussion can be found in the study of Khan *et al.* [37], Tsai *et al.* [38], Pouyanfar *et al.* [5], Zhu *et al.* [42], H. and Zahid *et al.* [43].

From the literature, it is clear that the recent studies ignored large-scale unstructured video analytics in the cloud in the discussion of big data. Some surveys focus on big data management and its related tools, while others have limited investigation of IVA in a particular context. They also do not consider the growing unstructured videos as video big data. Unlike existing work, this paper provides a comprehensive assessment of state-of-the-art literature and proposes an in-depth distributed cloud computing-based IVA reference architecture.

III. SCOPE AND NOMENCLATURE

We clarify some nomenclature being used and scoping this study. The first one is IVA, which is “any surveillance solution that utilizes video technologies to automatically manipulate and/or perform actions on live or stored video images [44]”. The IVA services are implemented through hardware called Video Analytics System (VAS) [30]. VAS assist acquires videos continuously and monitors unblinkingly. The VAS falls into four categories, i.e., Embedded Video Analytics System (EVAS), On-site Video Analytics System (OVAS), Fog-based Video Analytics System (FVAS), and CVAS.

EVAS embeds IVA solutions and performs video analysis directly on the edge device, e.g., camera or encoder, and can

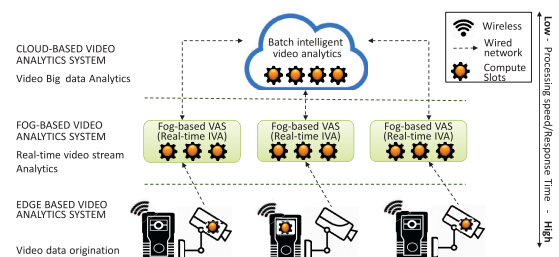


FIGURE 1. Geo-distributed video analytics infrastructure.

produce alerts in case of abnormality. EVAS provides very plain video analytics solutions and can simultaneously perform two or three rules on its stream and cannot accomplish complex algorithms such as fire detection, facial recognition, or cross video stream analytics. Under OVAS, small and middle-sized companies consist of networked or wireless cameras, a network router, a system running the video analytics and management software (e.g., IBM smart surveillance system [45], and Zoneminder [46]), and a storage device. All the cameras send the video data for analytics against contextual video analytics algorithms and warn the operator if anomaly detected. OVAS has many limitations, e.g., maintenance, software up-gradation, expensive hardware, scalability, and unable to deal with large-scale video data.

When IVA solutions are provided in fog and cloud computing environment, then it is called FVAS, and CVAS, respectively. In such environments the IVA solutions are made available under the as-a-Service (aaS) paradigm, i.e., IVaaS. In FVAS, the IVA solutions are geographically distributed and configured near the edge devices, i.e., video stream sources, to meet the strict real-time IVA requirements of large-scale video analytics, which must address latency, bandwidth, and provisioning challenges. Whereas the CVAS is more suitable for offline IVA because of the relatively high response time

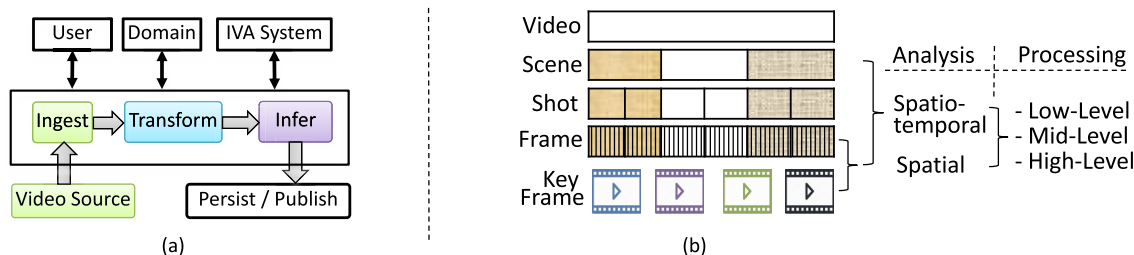


FIGURE 2. (a) A generic IVA service pipeline. (b) Hierarchical representation of video units.

and latency. The hierarchy and relation among CVAS, FVAS, and EVAS is shown in Fig. 1. The batch video data analytics are performed in the cloud, while the real-time IVA is performed in the fog. The EVAS can play a passive role in the proposed architecture, e.g., feed the video streams to the CVAS if motion is detected.

The scope of this paper is FVAS, and CVAS, i.e., the real-time and batch IVA solutions are deployed in the fog and cloud computing environment, respectively, while utilizing big data computing technology. However, for the ease of understandability, throughout this paper, we use the notion of CVAS.

To analyze a video in CVAS, a video undergoes through different phases, as shown in Fig. 2a. In Fig. 2a, Video Source are the sources which either generate videos streams from sources connected directly to real-world domains such as IP-camera or can be already acquired videos in the form of datasets residing in a file system. If IVA in the cloud are performed on real-time video streams, then we called it Real-time IVA (RIVA) and Batch IVA (BIVA) if performed on batch videos.

The Ingest phase implement interfaces to acquire videos from the Video Source. In the context of IVA, the acquired videos can be represented as a hierarchy, as shown in Fig. 2b. An acquired video from a Video Source may be decomposed into its constituent units either in the temporal or spatial domain. In a video, a frame represents a single image, whereas a shot denotes a consecutive sequence of frames recorded by a single camera. A scene is semantically related shots in a sequence that depicts a high-level story. A collection of scenes composes a sequence/story. Frames and shots are low-level temporal features suitable for machines, while scenes and sequence/story are considered to be the high-level features that are suitable for human perceptions. Such constituent units are further subject to low, mid, or/and high-level processing. In low-level processing, primitive operations (in Transformations phase) are performed e.g., noise reduction, histogram equalizer. The Infer phase encompasses mid and high-level processing. The mid-level processing extracts features from the sequence of frames, e.g., segmentation, description, classification, etc. The high-level processing, make sense of an ensemble of recognized objects; perform the cognitive functions normally associated with vision. Finally, the extracted

information can be persisted to the data store and/or published to the end-user.

Furthermore, the basic unit of an IVA service pipeline is an algorithm, e.g., encoder, feature extractor, classifier, etc. The input of an algorithm can be a Video Source, keyframes, or features. Similarly, the output of an IVA algorithm can be high, mid, or low-features. Throughout this study, all possible outputs of IVA algorithms are termed as IR. Multiple algorithms can be pipelined to build a domain-specific IVA service. The input and output of an IVA service are restricted to the Video Source and IR, respectively. The User represents the stakeholder of the CVAS, such as administrator, consumer, IVA researchers, and practitioners. A Domain is a specific real-world environment, e.g., street, shop, road traffic, etc., for which an IVA service needs to be built for automatic monitoring. Domain knowledge facilities IVA in discovering interesting patterns from domain video streams. The combination of software and hardware constitutes a distributed System (cloud environment) where IVA service and algorithms can run fast. Nevertheless, upgrading existing IVA algorithms to distributed architecture requires customization of how IVA algorithms should be implemented and deployed. Moreover, the specific needs of IVA may require the design and development of new system architecture. All the symbols being used in this paper are listed below.

LIST OF ABBREVIATIONS

- aaS as-a-Service.
- ACID Atomicity, Consistency, Isolation, Durability.
- AFS Adaptive Feature Scaling.
- AI Artificial Intelligence.
- B2B Business-to-Business.
- B2C Business-to-Customer.
- BIVA Batch IVA.
- C2C Customer-to-Customer.
- CAP Consistency, Availability, Partition Tolerance.
- CBVR Context-Based Video Retrieval.
- CCDG Controlled Cyclic Dependency Graph.
- CNN Convolutional Neural Network.
- CVAS Cloud-based Video Analytics System.
- DAG Directed Acyclic Graph.
- DBDS Distributed Big Datastore.
- DFS Distributed File System.
- DMBM Distributed Message Broker Manager.

DPDS	Distributed Persistent Data Store.
EVAS	Embedded Video Analytics System.
FVAS	Fog-based Video Analytics System.
FVAS	Fused Video Surveillance Architecture.
HDFS	Hadoop File System.
HOG	Histogram of Oriented Gradients.
IaaS	Infrastructure-as-a-Service.
IoT	Internet of Things.
IR	Intermediate Results.
IRM	Intermediate Results Manager.
ISDDS	Immediate Structured Distributed Data Store.
ITS	Intelligent Transportation System.
IVA	Intelligent Video Analytics.
IVAAaaS	IVA-Algorithm-as-a-Service.
IVAAaaS	IVA-as-a-Service.
IVAS	IVA services.
KCL	Knowledge Curation Layer.
L-CVAS	Lambda CVAS.
LBP	Local Binary Pattern.
LVSM	Lifelong Video Stream Monitor.
ML	Machine Learning.
OVAS	On-site Video Analytics System.
PaaS	Platform-as-a-Service.
PCA	Principal Component Analysis.
QoS	Quality of Service.
RDBMS	Relational Database Management System.
RIVA	Real-time IVA.
RNN	Recurrent neural network.
RVSAS	Real-time Video Stream Acquisition and Synchronization.
SaaS	Software-as-a-Service.
SIFT	Scale Invariant Feature Transform.
SVM	Support Vector Machine.
UPDDS	Unstructured Persistent Distributed Data Store.
VAS	Video Analytics System.
VBDCL	Video Big Data Curation Layer.
VBDML	Video Big Data Mining Layer.
VBDPL	Video Big Data Processing Layer.
VSAC	Video Stream Analytics Consumer.
VSAS	Video Stream Acquisition Service.
VSCS	Video Stream Consumer Service.
VSDS	Video Stream Data Source.
WSL	Web Service Layer.

IV. LAMBDA CVAS: A REFERENCE ARCHITECTURE

In this section, we briefly presents the technical details of the proposed CVAS (called Lambda CVAS (L-CVAS)) and the technical detail of each layer in the consecutive sub-sections. Fig. 3 presents the proposed reference cloud-based layered architecture for distributed RIVA and BIVA. The proposed L-CVAS architecture consists of five layers i.e., Video Big Data Curation Layer (VBDCL), Video Big Data Processing Layer (VBDPL), Video Big Data Mining Layer (VBDML), Knowledge Curation Layer (KCL), and Web Service Layer (WSL).

VBDCL is the foundation layer and is responsible for large-scale big data management throughout the life-cycle of IVA, i.e., from data acquisition to early persistence to archival and deletion [47]. VBDPL is responsible for distributed video pre-processing, feature extraction, etc. The VBDML deploys IVA algorithms on the top of distributed processing engines intending to produce high-level semantics from the processed sequence of frames. On top of the VBDML layer, the KCL layer has been designed to link the low-level features in spatial and temporal relation across videos in a multi-stream environment. KCL deploys a generic video ontology. The KCL layer maps the extracted IR to the video ontology to bridge the semantic gap between the low-level features in Euclidean space and temporal relation across videos while utilizing semantic rich queries. The proposed architecture incorporates top-notch functionalities of the above four layers into a simple unified role base WSL, which enables the L-CVAS users to manage, built, and deploy a wide array of domain-specific near RIVA and BIVA services.

All the layers are made available asaaS. These IVAAaaS are provided to the domain experts and allow them to pipelined in a specific context to built an IVA service. These IVA services are made available as IVAAaaS to which users can subscribe Video Sources.

Functionalities like security, scalability, load-balancing, fault-tolerance, and performance are mandatory and common to all the layers, which are shown as a cross-cutting in Fig. 3. The cloud infrastructure provides the underlying hardware and software under IaaS, on which the L-CVAS can be deployed. The cloud infrastructure is out of the scope of this paper. It has already been studied in detail in the context of big data by [16], [36], [39]. However, in the discussion of IVA in the cloud, some resources like CPU, GPU, FPGA, HDD, and SSD can be considered.

A. VIDEO BIG DATA CURATION LAYER

Effective data management is key to extract insights from the data. It is a petascale storage architecture that can be accessed in a highly efficient and convenient manner. We design the VBDCL for L-CVAS to efficiently manage video big data. L-CVAS's data storage stack consists of three main components: Real-time Video Stream Acquisition and Synchronization (RVSAS), Distributed Persistent Data Store (DPDS), and VBDCL Business Logic.

1) REAL-TIME VIDEO STREAM ACQUISITION AND SYNCHRONIZATION

The real-time video stream needs to be collected from the source device and forwarded to the executors for on-the-fly processing against the subscribed IVA service. Handling a tremendous amount of video streams, both processing and storage are subject to lose [51]. To handle, large-scale video stream acquisition in real-time, to manage the IR, anomalies, and the communication among RIVA services, we design the RVSAS component while assuming a distributed messaging system.

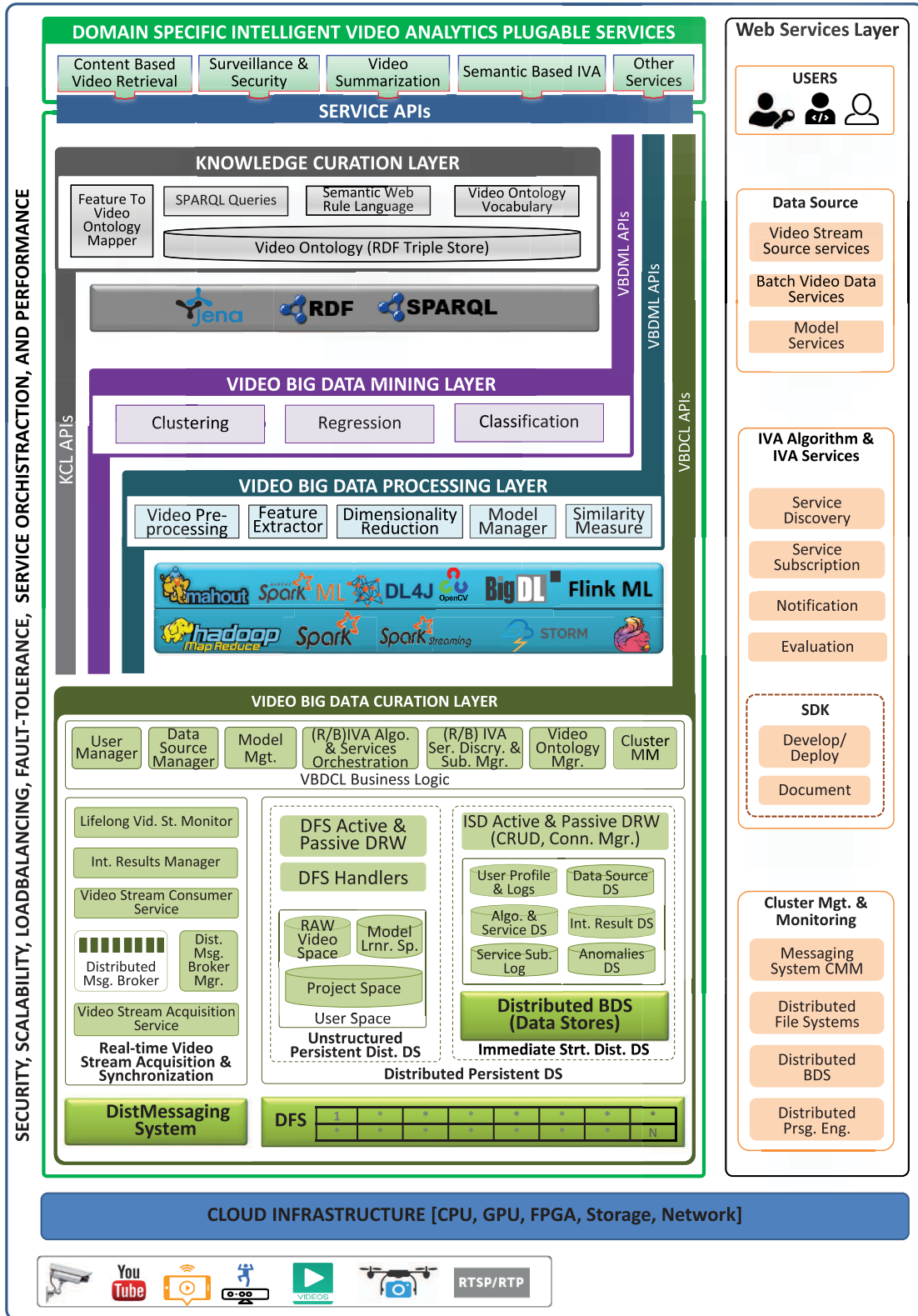


FIGURE 3. Lambda CVAS: A reference architecture for real-time and batch intelligent video analytics in the cloud.

Distributed Message Broker, also known as message-oriented-middleware [52], is an independent application that is responsible for buffering, queuing, routing, and delivering

the messages to the consumers being received from the message producer [53]. Message broker should be able to handle permission control and failure recovery. A message broker

TABLE 2. Popular message-oriented middlewares.

	Kafka [48]	RabbitMQ [49]	RocketMQ [1]	ActiveMQ [50]
Format	String, Byte	Map, String	String, Byte	String, Map, Byte, Object
Protocol	Kafka	AMQP, MQTT, STOMP, XMPP	Rocket, MQ	STOMP, REST, XMPP, AMQP
Pub/Sub	✓	✓	✓	✓
Point-to-Point	-	✓	✓	✓
Cluster	Stateless cluster	Simple clusters	Stateless cluster	Simple clusters
Load balancing	✓	✓	✓	✓
Persistence	✓	✓	✓	✓
Encryption	SSL	TLS	TLS	TLS
Authentication	SASL	SASL	-	JAAS
APIs	Java, .NET, PHP, Go, JS, Python, Ruby, Perl, Erlang	Java, .NET, PHP, Ruby, Python, Go, JavaScript	Java, C++, Go	Java, PHP, C++, Ruby, Perl, C#, Python, C

generally supports routing methods like direct worker queue, and/or publish-subscribe [54]. Similarly, the message consumer component receives the messages from the message broker either periodically (cron-like consumer) or continuously (daemon-like consumer). Generally and for the sake of scalability, message consumers are deployed on separate servers independently of message producers [54]. Some popular distributed messaging systems are shown in table 2.

RVSAS provides client APIs on the top of a distributed messaging system for the proposed framework. The RVSAS component is responsible for handling and collecting real-time video streams from device-independent video data sources. Once the video stream is acquired, then it is sent temporarily to the distributed broker server. The worker system, on which an IVA service is configured, e.g., activity recognition, reads the data from the distributed broker and process. The RVSAS component is composed of five sub-modules, i.e., Distributed Message Broker Manager (DMBM), Video Stream Acquisition Service (VSAS), Video Stream Consumer Service (VSCS), Intermediate Results Manager (IRM), and Lifelong Video Stream Monitor (LVSM).

a: DISTRIBUTED MESSAGE BROKER MANAGER

DMBM are used to manage the queues in the distributed message broker cluster considering RIVA services. Three types of queues, RIVA_ID, RIVA_IR_ID, and RIVA_A_ID as shown in Fig. 4, are automatically generated by the DMBM module on the distributed message broker when a new RIVA service is created. Here RIVA, ID, IR, and A stands for RIVA service, unique identifier of the service, IR, and Anomalies, respectively. These queues are used to hold the actual video stream being acquired by VSAS, IR produced by an algorithm, and anomalies detected by the video analytics services.

b: VIDEO STREAM ACQUISITION SERVICE

VSAS module is used to provide interfaces to VSDS and acquires large-scale streams from device-independent video data sources for on-the-fly processing. If a particular video stream source is subscribed against an RIVA service, then the VSAS gets the configuration metadata from the Data Source

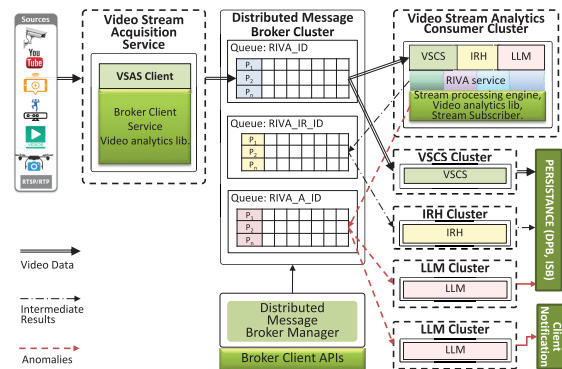


FIGURE 4. Real-time Video Stream Acquisition and Sync.

DS in Immediate Structured Distributed Data Store (ISDDS) and configure the source device for video streaming. After successful configuring the source device, VSAS decodes the video stream, detects the frames, and then performs some necessary operations on each frame such as meta-data extraction and frame resizing, which is then converted to a formal message. These messages are then serialized in the form of mini-batches, compressed, and sent to the Distributed Broker. If a video-stream source “C₁” is subscribed to the RIVA service “S₁” then the VSAS will rout the mini-batch of the video stream to queue RIVA_1 in the Broker Cluster as shown in Fig. 4.

c: VIDEO STREAM CONSUMER SERVICE

As the acquired video streams are now residing in the distributed broker in different queues in the form of mini-batches. To process these mini-batches of the video stream, we have different groups of computer cluster know as Video Stream Analytics Consumer (VSAC) Cluster. On each cluster, three types of client APIs are configured, i.e., RIVA services, VSCS, and LVSM. Each VSAC cluster has different domain-specific RIVA services where the VSCS are common for all. The VSCS assists the RIVA service to read the mini-batches of the video stream from the respective queue in the distributed broker for analytics, as shown in Fig. 4. The VSCS module has two main functions. First, this module

allows RIVA service to read the mini-batches of the video stream from the respective queue in the distributed broker. The second task is to save the consumed unstructured video streams and its meta-data to the row video space in the DPDS and to the Data Source DS meta-store, respectively.

d: INTERMEDIATE RESULTS MANAGER

During the IVA service life-cycle, a sequence of algorithms are executed. Thus the output of an algorithm can be the input of another algorithm. The IR demands proper management in the distributed environment because one algorithm may be on one computer while the other may be on another computer. Thus we design IRM that sends and gets the IR to and from the topic `RIVA_IR_ID` in the distributed broker cluster. Similarly, this module is also responsible for reading the IR from the respective queue and persists to the IR data store for future use so that to avoid recomputation.

e: LIFELONG VIDEO STREAM MONITOR

The domain-specific RIVA service process the video stream for anomalies or abnormal activities. If anomalies are detected, then it is sent to the distributed broker queue (i.e., `RIVA_A_ID`) by using the LVSM instance. To generate notification base response, LVSM follow standard observer-based concept [55]. Based on this approach, the LVSM module reads the anomalies from the respective distributed broker queue, i.e., `RIVA_A_ID` and notify the clients in near real-time and simultaneously persisted to the ISDDS.

2) DISTRIBUTED PERSISTENCE DATA STORE

The second component of the VBDCL is DPDS. The DPDS component provides the facilities of permanent and distributed big-data persistence storage of both structured and unstructured data. The DPDS provides abstraction in two levels on the acquired video data i.e. ISDDS and Unstructured Persistent Distributed Data Store (UPDDS). The philosophy behind DPDS and two levels of abstraction in the context of L-CVAS is many folds. From the users' perspective, the CVAS demands geo-based, real-time, low latency, and random read-write access to the data in the cloud. Similarly, the DPDS should also provide high-performance locality-based access to the data when the other layers deploy data-intensive IVA services. To meet such a diverse amount of requirements of the DPDS component, technologically, a Distributed File System (DFS) and Distributed Big Data-store (DBDS) can be leveraged.

a: IMMEDIATE STRUCTURED DISTRIBUTED DATASTORE

The ISDDS is provided to manage large-scale structured data in the distributed environment over DBDS. Because of the data-intensive operation and according to the requirements of the other layer, technologically, a distributed big data store, can be deployed. The ISDDS hosts five types of data. The detailed description of each type of data has been described in this section.

L-CVAS provides role-based access to its user. L-CVAS user logs and the respective role information are maintained through the User Profile and Logs meta-store. The proposed framework manages two types of video data sources through the Data Source meta-store. These are video data sources, for example, IP-cameras, Kinect, body-worn cameras, etc., and batch video datasets. The former one can be subscribed to RIVA service while the later one is eligible for BIVA services. The meta-information of these sources, along with access rights, are managed through the Data Source meta-store. Administrator and developer roles can develop, create, and deploy video analytics algorithms through the L-CVAS. Similarly, different IVA algorithms can be pipelined into an IVA service. The management of video analytics algorithms and services is managed through Video Analytics Algorithm and Service meta-store, respectively. As stated that in IVA pipelining environment, the output of one IVA algorithm can be the output of another algorithm. In this context, we design a general container called IR datastore to persist and index the output of an RIVA algorithm, and services. This datastore is significant and can play a vital role in IVA pipeline optimization, and fast content-based searching and retrieval. Finally, the L-CVAS users are allowed to subscribe to the data sources to the IVA services. The subscription information is maintained through the Subscription meta-store, and the anomalies are maintained through the Anomalies meta-store.

The ISDDS Data Model of the L-CVAS demands an efficient distributed data store. The distributed data store should have the ability of horizontal scalability, high availability, partition tolerance, consistency, and durability. Furthermore, the data store should fulfill the read/write access demands of the BIVA operations, and RIVA, interaction, and visualization. It is a fact that traditional relational databases have little or no ability to scale-out to accommodate the growing demands of the big data, and resultantly new distributed data stores have emerged. The distributed data stores can be grouped into two major categories, i.e., NoSQL (Not Only SQL) and NewSQL (excluding graph data stores).

NoSQL is a schema-free data store designed to support massive data storage across distributed servers [62], [63]. The features of NoSQL data stores include horizontal scalability, data replication, distributed indexing, simple API, flexibility, and consistency [64]. NoSQL lacks true Atomicity, Consistency, Isolation, Durability (ACID) transactions, unlike Relational Database Management System (RDBMS). In the context of Consistency, Availability, Partition Tolerance (CAP) theorem [65], it has to compromise on either consistency or availability while choosing partition tolerance. NoSQL can further be categorized as Document, Key-value, and Extensible stores. A key-value data store is responsible for storing values and indexes for searching. Document datastore is used for document storage, indexing, and retrieval. Extensible data store stores extensible records that can be partitioned vertically and horizontally across the nodes.

TABLE 3. Popular distributed file systems.

Features	CEPH [56]	GlusterFS [57]	HDFS [58]	Lustre [59]	MooseFS [60]	Quantcast File [61]
Initial release	2007	2007	2006	2003	2008	2011
License	LGPL 2.1	GPLv3+	Apache 2.0	GPLv2	GPLv2	Apache 2.0
Availability	✓	✓	Transparent master failover	✓	Master	Master
Tolerate disk failures	✓	✓	✓	✓	✓	✓
Scale	PBs	PBs	PBs	PBs	PBs	PBs
Shards	✓	✓	✗	✓	✗	✗
Commodity Hardware	✓	✓	✓	✓	✓	✓
Language	C++	C	Java	C	C	C
POSIX Compliant	✓	✓	✗	✓	✓	✓
APIs	Librados (C, Python, C++, Ruby), S3, Swift, FUSE	NFS, FUSE, SMB, Swift, libgfapi, libglusterfs	Java and C client, HTTP	NFS-Ganesha, NFS, SMB	FUSE	C++ client, FUSE

TABLE 4. Popular distributed data stores.

Type	DS Name	Release	Developer	OS	Language	D	HS	SL	Properties
Key-value	Voldemort	2009	LinkedIn	✓	Java	✓	✓	✓	<u>CAP</u>
	Riak	2009	Basho Tech.	✓	Erlang	✓	✓	✓	<u>CAP</u>
	Redis [67]	2009	Salvatore Sanfilippo	✓	C	✓	✓	✓	<u>CAP</u>
Document	SimpleDB	2007	Amazon	✗	Erlang	✓	✓	✓	<u>CAP</u>
	CouchDB	2005	ASF	✓	Erlang	✓	✓	✓	<u>CAP</u>
	MongoDB [68]	2009	10gen	✓	C, C++	✓	✓	✓	<u>CAP</u>
	DynamoDB	2012	Amazon	✗	Java	✓	✓	✓	<u>CAP</u>
Extensible	HBase [69]	2010	ASF	✓	Java	✓	✓	✓	<u>CAP</u>
	HyperTable	2008	Zvents	✓	C++	✓	✓	✓	<u>CAP</u>
	Cassandra	2008	ASF	✓	Java	✓	✓	✓	<u>CAP</u>
	BigTable [70]	2005	Google	✗	C, C++	✓	✓	✓	<u>CAP</u>
NewSQL	MySQL Cluster	2004	Oracle	✓	C++	✓	✓	✗	ACID
	VoldDB [71]	2009	VoltDB Inc	✓	Java, C++	✓	✓	✗	ACID
	ClustrixDB	2007	Clustrix, Inc.	✗	C	✓	✓	✗	ACID
	NuoDB	2008	NuoDB	✗	C++	✓	✓	✗	ACID

P=Persistence, D=Durability, HS=High Scalability, SL=Schema-less

CAP=Consistency, Availability, Partition Tolerance

ACID= Atomicity, Consistency, Isolation, Durability

In the properties column, the underline character represents the supported properties in the CAP theorem.

NewSQL data stores provide the characteristics of both NoSQL and RDBMS: ACID transactional consistency of relational databases with facilities of SQL; and the scalability and performance of NoSQL. MySQL Cluster, VoldDB, ClustrixDB are examples of NewSQL. Some of the popular NewSQL and NoSQL datastores are shown in Table 4 along with the respective properties. The ‘A’ and ‘C’ in CAP is not equal to the ‘A’ and ‘C’ in ACID [66].

b: UNSTRUCTURED PERSISTENT DISTRIBUTED DATASTORE

The UPDDS component built on the top of the DFS such as Hadoop File System (HDFS) that facilitates permanent and distributed big-data storage. The data are stored and mapped systematically according to the business logic of the proposed system. The UPDDS component is designed to effectively and dynamically manage the data workload in the life-cycle of an IVA service. Upon new registration with the L-CVAS,

a formal User Space is created on the top of DFS. The User Space is managed through the proper hierarchical directory structure and special read and writes access are granted to the owner. All the directories are synchronized and mapped in the corresponding user profile logs. Under the User Space, three types of distributed directories are created, i.e., Raw Video Space, Model Space, and Project Space. The hierarchical structure of the User Space in DFS is shown in Fig. 5.

Raw Video Space is used for the management of the video data. Raw Video Space is further divided into two types of video spaces. The first type is a batch video which has been uploaded to the L-CVAS for batch analytics, where the second type is acquired and persisted from the VSDS. The entire acquired stream is time-stamped on persistence. The granularity level of raw streaming videos is maintained through video data sources. IVA life-cycle may need different models for training and testing purposes. The Model Space is

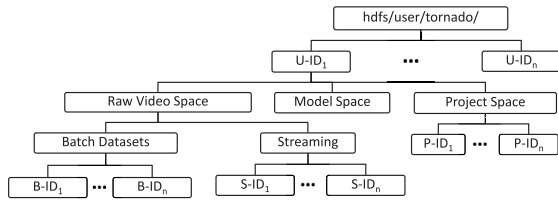


FIGURE 5. Hierarchical structure of the User Space in DFS.

provided to facilitate the developers to manage the training and testing model according to the deployed IVA algorithm. Similarly, a developer can develop a new algorithm or a video analytics service. The Project Space is provided to manage the source code of the respective developer and practitioners.

The UPDDS is supposed to be designed on the top of DFS. In this context, diverse types of open source DFS have been developed. The implementation quality of big data applications is relative to the storage tier's file system. From an architectural perspective, working and handling large volumes and throughput of data is challenging. Commonly, big data solutions exploit a cluster of computers, ranging from few to hundreds of computers, connected through high-speed networks while deploying specialized distributed data and management software. In distributed data-intensive applications, large-scale data is always moving across the cluster and thus demands a distributed, scalable, reliable, and fault-tolerant file system [72] known as DFS. DFS is a file system that provides access to replicated files across multiple hosts on a computer network, ensuring performance, data locality, high availability, scalability, reliability, security, uniform access, and fault-tolerance. Currently, various DFS are available and may differ in terms of performance, fault-tolerance, content mutability, and read/write policy. Some state-of-the-art popular DFS, which can be used for UPDDS, along with a short description, are shown in Fig 3.

c: ACTIVE AND PASSIVE DATA READERS AND WRITER

This module gives read-write access to the underlying data securely according to the business logic of the VBDCL Business Logic and according to the registered user access rights. This sub-module is composed of four types of readers and writers, i.e., ISDDS Active Data Reader, ISDDS Passive Data Reader, UPDDS Active Data Reader, and UPDDS Passive Writer. For real-time, read-write operation over the data residing in the ISDDS and UPDDS, such as CRUD operation, file creation, video stream writing to DFS, etc. the active data reader and writer are used. In the context of offline analytics over the bulk of videos while using distributed processing engines. The Passive Data Reader and Writer (PDRW) is provided to allow processing engines to load the bulk of data and persist the same to the ISDDS and UPDDS.

3) VBDCL BUSINESS LOGIC

VBDCL Business Logic provides the actual business logic. It implements six different modules, i.e., User Manager, Data

Source Manager, Model Manager, (R/B)IVA Algorithm and Service Manager, (R/B)IVA Service Discovery and Subscription Manager, Video Ontology Manager, Cluster Management and Monitoring.

The User Manager module encapsulates all the user-related operations such as new user account creation, access role assignment, and session management. Through the Data Source Manager Model Manager modules, the user can manage the VSIDS, video data uploading, and model management. The (R/B)IVA Algorithm and Service Manager are built to manage, develop, and deploy new IVA algorithms and services, respectively. The former one is provided aaS to the L-CVAS developers, while the latter one is provided aaS to the consumers. The developer role can create and publish a new video analytics algorithm. The algorithm is then made available aaS to other developers and can use it. Once IVA services are created, then the L-CVAS users are allowed to subscribe to the streaming video data sources and batch data against the provided RIVA and BIVA services, respectively.

Similarly, the Ontology Manager allows the developer to get the IR for decision making. The ontology manager provides a secure way of getting the IR and maps it according to ontology. This module also allows the user to manage the functionalities of the KCL. Finally, Cluster Management and Monitoring allows the administrator to monitor the health of the cluster.

B. DISTRIBUTED INTELLIGENT VIDEO ANALYTICS

IVA performs the complex tasks of extracting significant knowledge and information of interest from the video data, i.e., structural patterns, behavior patterns, content characteristics, event patterns [73], [74] and their relationship in the form of classification and clustering. Extracting knowledge and information from video big data is a processing-intensive task. Value extraction from video big data is behind the capabilities of traditional tools and demands for technological solutions to meet the processing requirements.

Therefore, video big data analytics is preferably performed in the distributed environment in a parallel manner while utilizing distributed scale-out computing technologies [75], such as MapReduce and Apache Spark. Distributed IVA not only significantly improves the performance, but also reduces the analytics cost. Based on the generic IVA life-cycle and motivated by scikit-learn [76], the distributed IVA are divided into two layers, i.e., VBDPL, and VBDML. These two layers are further elaborated in the following sub-sections.

1) VIDEO BIG DATA PROCESSING LAYER

IVA requires video data pruning and strong feature extraction. With such intentions, the VBDPL layer consists of three components, i.e., Video Preprocessing, Feature Extractor, and Dimensionality Reduction is designed.

a: VIDEO PREPROCESSING

The quality of data plays an active and significant role in solving a problem with ML. Raw videos have an unstructured format and contain noise/uncertainties, making it unsuitable for knowledge and information mining. Video Preprocessing component is designed with the same objectives and is supposed to deploy several distributed video preprocessing operations including frame extraction [77], frame-resizing, frame-conversion from RGB to grayscale [78], shot boundary detection [79], segmentation [80], transcoding [81], and many more. In the first step, frames are extracted from a video for processing. Several frame selection algorithms are available, for example, keyframe extraction. The number of frames to be extracted is dependent on the user objective and task-dependent. Candidate frames can be all frames, step frames (every second frame, fifth frame, etc.) or keyframes [82]. The spatial operations highly depend on the scenario and objective. Spatial operations include frame resizing (for reducing computational complexity), corrections (brightness, contrast, histogram equalization, cropping, keyframes), mode (RGB, Grayscale, etc.), and many other operations. Segmentation is used for various purposes, such as partitioning video into semantically related chunks.

b: FEATURE EXTRACTOR

The Feature Extractor component implements distributed feature extraction algorithms. The performance of ML is highly dependent upon the type of data representation or features [83]. Features represent the characteristics of classes in the dataset and have a heavy impact on the ML algorithm's generalizability and performance. The data features that used to train ML models have a huge influence on the performance of the algorithm. Inappropriate or irrelevant features affect the performance of the algorithm. Thus, feature extraction extracts the features from the raw videos that can be interpreted by the ML algorithm [84], [85]. In this context, several feature extraction algorithms have been introduced for video data. These feature extraction approaches can be categorized into static features of keyframes [86]–[88], object features [89], [90], dynamic/motion feature extraction [84], [91], [92], trajectory-based features extraction [84], [85], [93], [94], and deep learning-based feature extraction [95]–[103].

c: FEATURE SELECTION AND DIMENSIONALITY REDUCTION

Feature selection and dimensionality reduction reduce the size of the features. Large sizes of feature sets are expensive in terms of time for training and/or performing classification acquired by trained classifiers. For example, Principal Component Analysis (PCA) and its variants are used to reduce the size of features. During feature selection, most relevant features are selected by discarding irrelevant and weak features. The performance of ML classifiers is also directly related to the quality of features; GIGO (garbage in garbage out).

Inappropriate or partially relevant features can negatively affect model performance. Therefore, only a limited set of features should be selected and used for training classifiers. This is what precisely the purpose of this component is and deploy different algorithms in this context. Similarly, some feature reduction techniques available that selects the specific set of limited features in real-time. For example, Online Feature Selection selects and inputs a specific number of small features to the classifiers in real-time [104]. In order to accelerate the training process, [105] used non-linear and group-based feature selection techniques, based on Adaptive Feature Scaling (AFS), to process the data with substantial dimension sizes. [106] proposed an unsupervised feature reduction technique that selects extremely relevant features and indicates suitable weights to the distinctive feature dimensions.

2) VIDEO BIG DATA MINING LAYER

The VBDML utilizes diverse types of machine-learning algorithms, i.e., supervised, semi-supervised, and unsupervised algorithms to find different type of information from the videos [73], [74]. In this context, VBDML layer hosts three types of components, i.e., Classification, Regression, Clustering.

Classification component provides various ML algorithms, e.g., Support Vector Machine (SVM), Nearest Neighbors, Random Forest, Decision Tree, Naïve Bayes, etc., that identifies that a particular object in a video frame belongs to which category while using predefined classes. The Regression component includes different algorithms, e.g., Linear Regression, Decision Tree Regression, Logistic Regression, and many more, predicting a continuous-valued attribute associated with objects rather than discrete values. The Clustering component encapsulates algorithms, e.g., K-Mean, spectral clustering, etc., that produces groups of data depending upon the similarity of data items.

L-CVAS has the ability of BIVA and RIVA. In this context, the VBDML should support batch learning and online learning. The former case considers the complete training data to learn and generates models. The batch-learning algorithm is expected to generalize, that usually does not perform well in the real environment. Unlike batch learning, online learning continuously learns from new input without making any statistical assumptions about the data [107]. In the context of model generalization, online learning is expected to work well by accurately predicting the predefined set of inputs [107]. Online learning is used in the environment when continuous learning from the data is required to learn new patterns instead of batch learning.

3) DISTRIBUTED DEEP LEARNING FOR IVA

Handcrafted features, e.g., Scale Invariant Feature Transform (SIFT) [108], Local Binary Pattern (LBP) [109], Histogram of Oriented Gradients (HOG) [110], etc., generates high dimensional features vectors and resultantly facing the issue of scalability. Recently, Convolutional Neural Network

TABLE 5. Comparison of popular deep learning architectures.

Architecture	Year	Error	Parameters	Depth	Description and Pros	Cons
LeNet	1998	-	60 K	5	<ul style="list-style-type: none"> - First popular CNN architecture - Utilize spatial correlation to decrease the parameters and computation - Automatic learning of feature hierarchies 	<ul style="list-style-type: none"> - Poor scaling to diverse classes of images - Large size filters - Extraction of low-level features
AlexNet	2012	15.3%	60 M	8	<ul style="list-style-type: none"> - Low, mid and high-level feature extraction using large and small size filters on initial (5x5 and 11x11) and last layers (3x3) - The notion of deep and wide CNN architecture - Introduced regularization in CNN - Initiated GPUs parallelization - Deeper and wider than the LeNet - Uses Relu, dropout and overlap Pooling - GPUs NVIDIA GTX 580 	<ul style="list-style-type: none"> - Inactive neurons in the first and second layers - Aliasing artifacts in the learned feature-maps due to large filter size
ZFNet	2013	14.8%	-	8	<ul style="list-style-type: none"> - The notion of parameter tuning by visualizing the output of intermediate layers - Reduced both the filter size and stride in the first two layers of AlexNet 	<ul style="list-style-type: none"> - Extra information processing is required for visualization
GoogleNet	2014	6.67%	4 M	22	<ul style="list-style-type: none"> - Introduces block concept - The notion of utilizing Multiscale Filters within the layers - The notion of a split, transform and merge - Parameters reduction by utilizing the bottleneck layer, global average pooling at the last layer and Sparse Connections - Utilization of auxiliary classifiers to enhance the convergence rate 	<ul style="list-style-type: none"> - Heterogeneous topology led to tedious parameter customization - Chances of losing the valuable information because of representational bottleneck
VGGNet	2014	7.3%	138 M	19	<ul style="list-style-type: none"> - Small kernel size - The notion of effective receptive field - The notion of simple and homogenous topology 	<ul style="list-style-type: none"> - Use of computationally expensive fully connected layers
ResNet	2015	3.6%	-	152	<ul style="list-style-type: none"> - The error rate was decreased for deeper networks - Introduced the notion of residual learning - Improves the effect of vanishing gradient issue - Identity mapping based skip connection 	<ul style="list-style-type: none"> - A bit complex architecture - Lowers information of feature-map in feed forwarding - Over adaption of hyperparameters for a specific task, due to the stacking of the same modules

(CNN) based approaches have shown performance superiority in tasks like optical character recognition [111], and object detection [112]. The motive of the deep learning is to scale the training in three dimensions, i.e., size and complexity of the models [113], proportionality of the accuracy to the amount of training data [114], and the hardware infrastructure scalability [115]. Results of deep learning are so promising that soon the deep learning will give equivalent or higher performance compared to humans when trained over large data sets [112]. A CNN or ConvNet is a type of neural network that can recognize visual patterns directly from the pixels of images with less preprocessing. CNN based video classification methods have been proposed in the literature to learn features from raw pixels from both short video and still images [96], [97], [116], [117]. In the proposed L-CVAS framework, both the VBDPL, and VBDML are capable to deploy deep-learning approaches for distributed IVA.

Since on the dawn of deep learning, various open-source architecture have been developed. Some of the well-known and state-of-the-art CNN architectures are LeNet-5 [118], AlexNet [117], ZFNet [119], GoogleNet [120], VGGNet [121], and ResNet [122]. The comparison of these architectures can be found in Table 5. Similarly, several frameworks have been developed to eliminate the need for a manual definition of gradient propagation. Table 6 summarizes these libraries along with the comparisons.

TensorFlow is a popular deep learning library designed for ML and deep learning. It supports the deployment of

computation on both CPUs and GPUs. TensorFlow allows the fast implementation of deep neural networks on the cloud. TensorFlow is also suitable for other data-driven research purposes and is equipped with TensorBoard (a visualization tool). Higher-level programming interfaces such as Luminoth, Kera, and TensorLayer were built on the top of TensorFlow. Caffe2, developed by Berkeley AI Research, is another library to build their deep learning models efficiently along with GPUs' support in a distributed environment. PyTorch, maintained by Facebook, is a scientific computing framework with wide support for machine learning models and algorithms. PyTorch offers rich pre-trained models that can be easily reused. MXNET is a deep learning library suitable for fast numerical computation for both single and distributed ecosystems. Likewise, some more deep learning libraries have been developed, such as CNTK, Deeplearning4j, Blocks, Gluon, and Lasagne, which can also be employed cloud environment.

Big DL models training with large-scale training data is a challenging task. For example, S Gao et al. [129] utilized six learning algorithms, i.e., biogeography-based optimization, particle swarm optimization, genetic algorithm, ant colony optimization, evolutionary strategy, and population-based incremental learning, for the best combination of neural network user-defined parameters during training. It is a hectic job for a single system when the training datasets are large. Distributed infrastructure with multiple computing nodes

TABLE 6. Summary of popular deep learning libraries.

Features	TensorFlow [123]	Caffe2 [124]	PyTorch [125]	MXNet [126]	CNTK [127]	Matlab DL Toolbox [128]
Release	2015	2013	2016	2015	2016	2017
Licence	Apache 2.0	BSD	BSD	Apache 2.0	MIT	Proprietary
Written in	C++, Python, CUDA	C++	Python, C, C++, CUDA	C++	C++	C, C++, Java, MATLAB
Interface	Python (Keras), C/C++, Java, Go, JavaScript, R, Julia, Swift	Python, MATLAB, C++	Python, C++	C++, Python, Julia, Matlab, JavaScript, R Go, Scala, Perl, Clojure	Python, C++, C#, and Java.	MATLAB
GPU	✓	✓	✓	✗	✓	✓
DNN Support	RNN, CNN, LSTM, etc.	CNN, RNN, LSTM	RNN, CNN, LSTM, etc.	RNN, CNN, LSTM, etc.	CNN, RNN	ConvNets, CNN, LSTM, etc.
Top-Level Libraries	TensorLayer, Keras,	-	-	Gluon	Keras	-
Pros	Tensorboard for visualization, well documented, large user community	Fast, scalable, lightweight, cross-platform support, server optimized interface	Easy to use and debug, flexible, well documented, declarative data parallelism	Lightweight and fast, memory-efficient, highly scalable	Good performance and scalability, highly optimized, support for Spark	easy for scientists, model visualization, high-performance CUDA code generation,
Cons	Debugging issues, heavy, difficult for novice	Small community, modest documentation	Lacking model serving, visualization tools	Small community, difficult to learn	Limited community	Closed-source, expensive in terms of execution and price

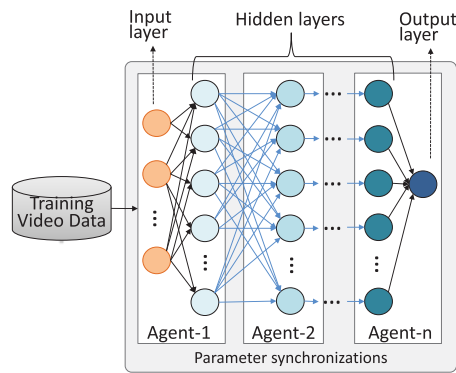


FIGURE 6. Scalable deep learning utilizing model distribution.

(equipped with powerful GPUs) is the best option, but it leads to numerous challenges. First is the effective utilization of resources (costly GPU stalling preventions). Second, the resources are shared among different users in the cloud for cost reduction and elasticity. Such challenges are attracting the attention of researchers. There are three approaches, i.e., model, data, and pipeline distribution for leveraging distributed computing in distributed computing [130], [131]. In the former, the DL model is partitioned in logical fragments, and loaded to different worker agents for training, as shown in Fig. 6.

In the second approach, the deep learning model is replicated to all the cluster’s worker-agents, as shown in Fig. 7. The training dataset is partitioned into non-overlapping sub-dataset, and each sub-dataset are loaded to the different worker-agents of the cluster. Each worker-agent executes the training on its sub-dataset of training data. The model parameters are synchronized among the cluster worker-agents to updates the model parameters. The data distribution approached naturally fits in the distribute computing MapReduce paradigm [131]. The MapReduce splits the input based on some predefined parameters. The map tasks, then, process these chunks in parallel a manner. After processing, the output is shuffled for relevance and is directed to map tasks for generating intermediate results. The output from the map tasks is shuffled for relevance and is given as input to the reduce tasks for generating intermediate results. The

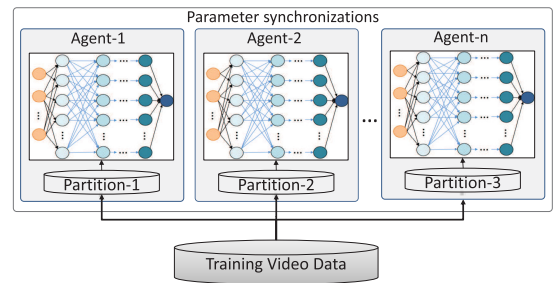


FIGURE 7. Scalable deep learning utilizing data distribution.

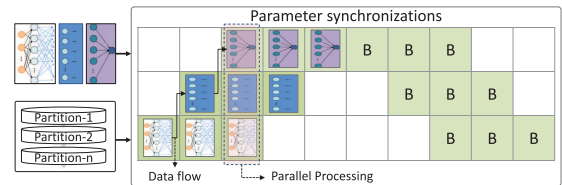


FIGURE 8. Deep learning with pipelining (Figure motivated by [132]).

intermediate results are combined to produce the complete result. Hadoop and Spark require and process data naturally distributed in the manner and popular research trend nowadays.

In the third case, the DL model is partitioned, and each worker-agent loads a different segment of the DL model for training, as shown in Fig. 8. The training data are given to the worker-agents that carry the input layer of the DL model. In the forward pass, the output signal is computed, which is transmitted to the worker-agents that hold the next layer of the DL model. In the backpropagation pass, gradients are calculated starting at the workers that carry the DL model’s output layer, propagating to the workers that hold the input layers of the DL model [132].

4) BIG DATA ENGINES, ML LIBRARIES, AND IVA

The VBDPL, and VBDML are assumed to be built on the top of distributed computing engines. This section overview some latest big data engines that can be utilized for scale-out IVA.

TABLE 7. Comparison of big data engines.

Features	Hadoop [133]	Spark [134]	Flink [135]	Storm [136]	Samza [137]
Batch	✓	✓	✓	✗	✗
Stream	✗	near real-time	Real-time	Real-time event based	Event based
Data Flow	No loops, Chain of stages	Direct Acyclic Graph	Controlled Cyclic Dependency Graph	Direct Acyclic Graph	Kafka Dependent
Fault-tolerance	High and application restart is not required in case of failure.	Recovers lost work in case of failure.	Based on distributed snapshots	Automatically restart dead workers or migrate in case of node failure.	Migrates the task into another node in case of node failure.
Scalability	✓	✓	✓	✓	✓
Latency	Higher	Low	Low	Low	Low
Cost	Less expensive	Expensive	Expensive	Expensive	Expensive
Security	Kerberos authentication	Shared secret, HDFS ACLS and Kerberos	Kerberos authentication, SSL	Kerberos authentication	Kerberos authentication
Scheduler	By default Fair Scheduler, Capacity Scheduler, and is pluggable.	Acts as its own flow scheduler because of in-memory computation.	Own scheduler and can use YARN scheduler.	Default, isolation, multi-tenant and resource aware scheduler.	Use YARN scheduler.
Language Support	Java, C/C++, Ruby, Groovy, Perl and Python	Java, Scala, Python, and R.	Java, Scala, Python, and R.	Java, Clojure, and can support non-JVM languages.	Java, scala and can support JVM languages.
Windowing	✗	Time-based	Time-based, Count based	Time-based, Count based	Time-based
ML Library	Mahout [138], DL4J [139], Keras [140], Caffe [124], TensorFlow [123], BigDL [141]	MMLib (Native) [142], DL4J, Mahout, DL4J, TensorFlow, Keras, Caffe, H2O, PyTorch [125], BigDL	FlinkML (Native), Mahout, TensorFlow, Keras, Caffe, H2O	Don't have native ML Lib, SAMOA 3rd Party [143]	Don't have native. SAMOA 3rd Party

Hadoop MapReduce [144] is a distributed programming model, developed based on GFS [72], for data-intensive tasks. Apache Spark follows a similar programming model like MapReduce but extends it with Resilient Distributed Datasets (RDDs), data sharing abstraction [134]. Hadoop's MapReduce operations are heavily dependent on the hard disk while Spark is based on in-memory computation, which makes Spark a hundred times faster than Hadoop [134], [145]. Spark support interactive operations, Directed Acyclic Graph (DAG) processing, and process streaming data in the form of mini-batches in near real-time [146]. Apache Spark is batch centric and treats stream processing as a special case, lacking support for cyclic operations, memory management, and windows operators. Such issues of Spark has been elegantly addressed by Apache Flink [135]. Apache Flink treats batch processing as a special and does not use micro-batching. Similarly, Apache Storm and Samza is another prominent solution focused on working with large data flow in real-time. A brief description and comparison of the open-source big data frameworks are listed in Table 7.

To achieve scalability, big data techniques can be exploited by existing video analytics modules. The VBDPL is not provided by default and needs its implementation on the top of these big data engines. However, The ML approaches can be categorized into two classes in the context of VBDML. One class re-implements the existing ML task by providing a middleware layer to run them on a big data platform. This general type of middleware layer provides general primitives/operations that assists in various learning tasks. Users who want to try different ML algorithms in the same framework can take benefits from it. In the second class, the individual video analytics and ML algorithm are executed on a big data platform that is directly built on top of a big data engine for better scalability.

Spark MLlib [142], Mahout [138], FlinKML [135] are list of some open-source ML packages built on the top of

Hadoop, Spark and Flink, respectively, that support many scalable learning algorithms. For deep learning, various open-source libraries have been develop including TensorFlow [123], DeepLearning4J [139], Keras [140], Caffe [124], H2O [147], BigDL [141], and PyTorch [125]. All these libraries provide support for various ML algorithms and feature engineering. These libraries introduce an independent layer between front-end algorithms and a back-end engine to facilitate the migration from one big data engine to another, as shown in Table 7. These algorithms can be used to process large datasets, just like processing it on a single machine by providing the distributed environment abstraction and optimization.

5) COMPUTER VISION BENCHMARK DATASETS

In the advancement of IVA, public datasets always play a vital and active role. Over the year, server benchmark datasets have emerged. Some of the recent popular datasets are listed in Table 8. ImageNet [148] is one of the significant datasets in deep learning and is utilized for training neural networks such as ResNet, AlexNet, and GoogleNet. Some more datasets have been developed aiming human action and motion recognition, including [150], [150], [152], [155], [156]. Google released YouTube-8M [159] and consisting of eight million diverse types of automatically labeled videos. Deng A *et al.* [158] proposed the HowTo100M dataset comprising of web videos with narrated instructions. Dataset like MediaEval2015 [153], and Trecvid2016 [150] are designed to support CBVR related research. For sports IVA, the Sports-1M dataset karpthy2014large is proposed and composed of 487 classes along with ground truth. YACVID [154] is a labeled image sequence dataset for benchmarking video surveillance algorithms. All these benchmark datasets are used for different IVA, such as action recognition, event detection, and classification.

TABLE 8. Popular computer vision benchmark datasets.

Dataset	Instances	Classes	Ground Truth	Description
YFCC100M [148]	100M	8M	Partially	Video and image, understanding
YouTube-8M [149]	8M	4,716	Automatic	Video classification
Trecvid 2016 [150]	Different	Different	Partially	Human action detection
UCF-101 [151]	13K	101	Yes	Human action detection
Kinetics [152]	306K	400	Yes	Human action detection
MediaEval [153]	-	-	-	Videos from BBC
YACVID [154]	75207 Frames	-	Yes	A multi-camera high-resolution dataset
ActivityNet [155]	849 video hours	203	Yes	Human activity
HMDB [156]	7K Videos	51 Classes	Yes	Human Motion Recognition
Sports-1M [97]	1M	487	Yes	The YouTube Sports-1M Dataset
AVA [157]	57K	Different	Yes	Spatiotemporally Localized Atomic Visual Actions
HowTo100M [158]	100M	23k	-	Narrated instructional web videos

C. KNOWLEDGE CURATION LAYER

Videos low-level processing produces feature descriptors that summarize characteristics of data quantitatively. The high-level analytics is more associated with the visual data understanding and reasoning. The features descriptors work as input to the high-level analytics and generate abstract descriptions about contents. The difficult problem is to bridge the semantic gap between the low-level features and high-level concepts suitable for human perception [160].

With the same indentations, the KCL layer has been proposed under L-CVAS architecture, on the top of VBDML, which map the IR (both online and offline) into the video ontology in order to allow domain-specific semantic video and complex event analysis. The KCL is composed of five components, i.e., Video Ontology Vocabulary, Video Ontology, Semantic Web Rules, FeatureOnto Mapper, and SPARQL queries.

Video Ontology Vocabulary standardizes the basic terminology that governs the video ontology, such as concept, attributes, objects, relations, video temporal relation, video spatial relation, and events. Video Ontology is a generic semantic-based model for the representation and organization of video resources that allow the L-CVAS users for contextual complex, event analysis, reasoning, search, and retrieval. Semantic Web Rules express domain-specific rules and logic for reasoning. When videos are classified and tagged by the VBDML then the respective IR are persistent to VBDCL and also mapped to the Video Ontology while using the FeatureOnto Mapper. Finally, SPARQL based semantic rich queries are allowed for knowledge graph, complex event reasoning, analysis, and retrieval.

D. WEB SERVICES LAYER

Finally, to provide the functionality of the proposed L-CVAS over the web, it incorporates top-notch functionality into simple unified role-based web services. The Web Service Layer is built on the top of VBDCL Business Logic. Sequence diagrams for IVA algorithm and service creation is shown in Fig. 10. Whereas, role-based use case diagram of the proposed platform is shown in Fig. 9.

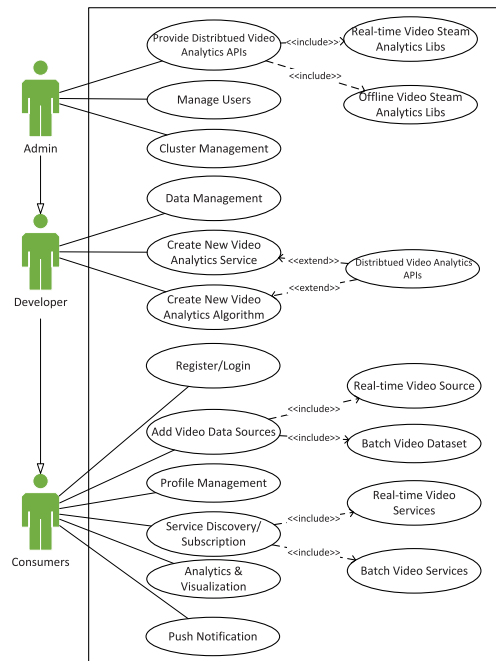


FIGURE 9. Lambda CVAS user roles and use case diagram.

E. IVA SERVICE EXAMPLE SCENARIOS

We show two example scenarios, i.e., how to develop BIVA and RIVA services under L-CVAS. Hadoop and Spark MapReduce type operations naturally fit in a BIVA. Fig. 11, shows the block diagram along with sample script for distributed BIVA, i.e., object classification, where a DFS is configured to read and store video files, e.g., in a standard video file format such as MPEG, AVI, H.264, etc. First, the videos are loaded, the distributed video transcoding (a preprocessing algorithm under VBDPL) are performed. For example, a user uploads a MPEG or other video files to the DFS. The transcoder algorithm first split the file into image frames, which may be throttled to key-value frames, and converts them into a sequence file format. These frames are then mapped, and features are extracted (utilizing some feature extractor from VBDPL). The features are then classified (using a classification algorithm of VBDPL). For each

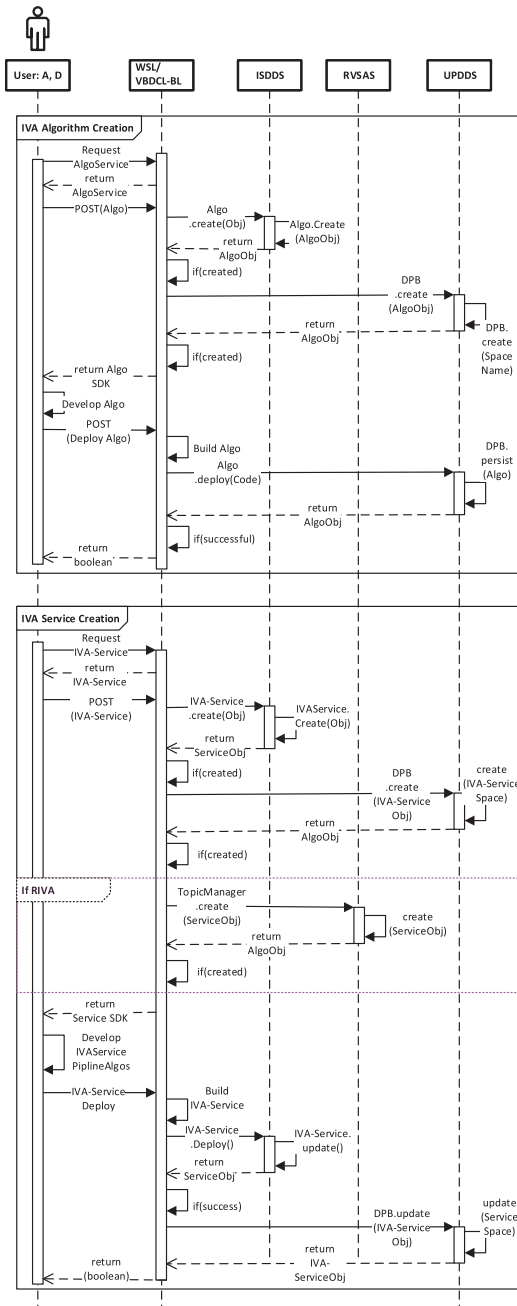


FIGURE 10. The sequence diagram for the IVA algorithm and service creation. In the Actor lifeline, A and D represent Admin and Developer, respectively. These two roles are allowed to create a new video analytics algorithm and service.

detected object, a key-value and coordinates of where the object is located within the image are computed. For each detected object in each frame, the map element process the frame. The map operation generates and provides output as composite visual value pair that includes the visual key, a time-stamp that identifies the frame, and the coordinates. The map stages then send the output in the form of a composite key-value pair to the respective reduce stages. The reduce stages provide the output to an output stage, which is persisted in the IR-DS of ISDDS.

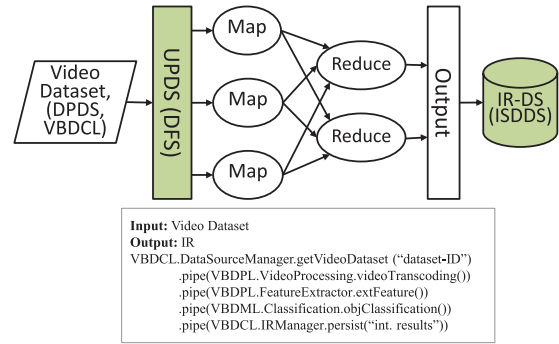


FIGURE 11. BIVA service example pipeline under Lambda CVAS.

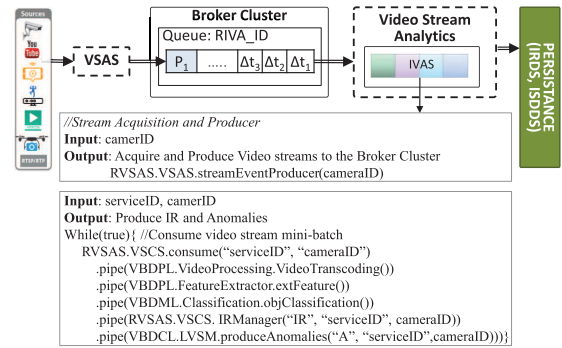


FIGURE 12. RIVA service example pipeline under Lambda CVAS.

The flow for a single RIVA along with an example pseudo-code is shown in Fig. 12. The VSAS component sends the video stream to the queue in the broker cluster. Then the consumer service (VSCS) is used to extract the mini-batch of video streams from the queue and process it. Once the mini-batch is consumed, then it is transcoded, features are extracted for classification, and finally, the classification results are persisted to the required destination (IR-DS and/or dashboard).

F. EXECUTION SCENARIOS

L-CVAS follows the lambda architecture style [29], and the execution scenarios undergo through two types of execution scenarios, i.e., Streaming Execution Scenario, and Batch Execution Scenario. These two scenarios aim to execute a massive amount of real-time video stream and batch videos against the subscribed IVA services. In literature, these execution scenarios are referred to as Speed Layer, and Batch Layer respectively [29]. The data of both the scenarios are managed through a common layer called Serving Layer. L-CVAS components are deployed on various types of clusters in the cloud, and each cluster is subject to scale-out on-demand. Fig. 13 illustrates these execution scenarios, and the explanation is given in the following subsections.

1) STREAMING EXECUTION SCENARIO

L-CVAS is supposed to deploy a pool of contextual RIVA services that are made available to the user for the subscription.

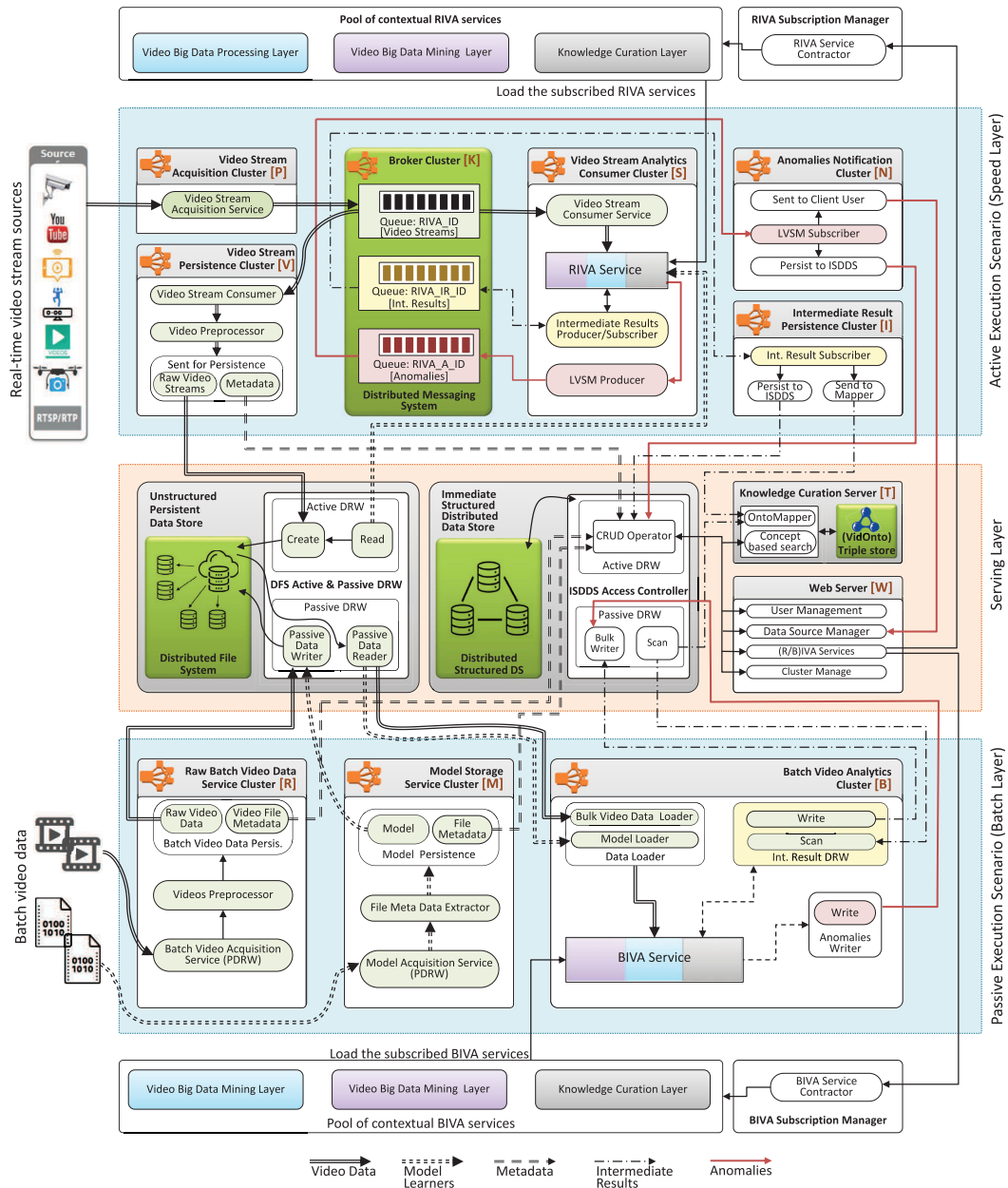


FIGURE 13. Lambda CVAS: Streaming and Batch execution scenarios.

Once a video stream data source is subscribed to service in the pool of contextual RIVA services, then the life-cycle of Streaming Execution Scenario encompasses through different stages while using distinct L-CVAS components. For the ease of understandability, these components are deployed on six types of computing clusters in the cloud, which are labeled explicitly as 'P', 'V', 'K', 'S', 'N', 'I', as shown in Fig. 13.

The cluster 'P' hosts VSAS and provides interfaces to external VSDS. On configuration, the video streams are captured and transformed into a proper message, which is then grouped into micro-batch, compressed, and loaded to the respective queue in the cluster 'K'.

The cluster 'K' deploys the distributed messaging system, where the acquired video streams, IR, and anomalies produced by LVSM are buffered. In this context, the cluster 'K' is composed of a set of three types of queues for each service, i.e., RIVA_ID, RIVA_IR_ID, and RIVA_A_ID, as described in IV-A1.

The mini-batches of video streams residing in the distributed broker's RIVA_ID queue need to be persisted to the UPDDS and ISDDS data stores. For this purpose, the cluster 'V' deploys three types of L-CVAS modules, i.e., VSCS, Video Processor and Persistence. The first module allows the cluster 'V' to read the video stream mini-batches from RIVA_ID topics in the Cluster 'K'. The

cluster 'V' then processes the consumed video data, encodes it, and extracts the metadata from the video. Finally, the video stream persistence module saves the video data and the respective metadata to the UPDDS and ISDDS, respectively.

The cluster 'S' is responsible for processing the video stream in near real-time while using the IVA services. Different stream processing engines, e.g., Apache Spark Stream, can be used for RIVA. The cluster 'S' deploys four types of L-CVAS modules. The first module is VSCS and is used to consume the video streams from the `RIVA_ID` queue in the cluster 'K'. The second type of module is RIVA services. The RIVA service is the actual video stream analytics service that analyzes the videos. The video RIVA service is loaded according to the RIVA services subscription contract made by the L-CVAS user. The RIVA services can be pipelined, and the IR might need other applications in the multi-subscription scenario. Thus the IR producer/subscriber is used to send and receive the IR according to the application logic to and from the IR queue in Cluster 'K'.

The fourth type of module is LVSM producer. The contextual IVA services (IVAS) instance deployed in the cluster 'S' should have some domain-specific goal and can produce anomalies if analyzed any. The L-CVAS support real-time anomalies delivery system. The IVAS sent the anomalies continuously to the LVSM producer and the LVSM producer to the respective anomalies queue `RIVA_A_ID` in the cluster 'K'.

The cluster group 'I' read the IR from the `RIVA_A_ID` queue in cluster 'K' continuously and sent it to the ISDDS's IR data store for persistence. If the subscribed service is using the ontology then the IR are also mapped to the VidOnto triple residing in the Knowledge Curation Server 'T'.

The final type of cluster in the Streaming Execution Scenario is cluster 'N' and is known as Anomalies Notification Cluster. This cluster aims to read anomalies from the `RIVA_A_ID` queue in cluster 'K' and send the same to the ISDDS for persistence and also delivered in real-time to the video stream source owner in the form of alerts.

2) BATCH EXECUTION SCENARIO

The L-CVAS architecture is also equipped with BIVA. Unlike the RIVA, the BIVA is analyzed as an offline manner. The execution time of offline analytics is proportional to the video dataset size and the subscribed BIVA service computation complexity. The Batch execution life-cycle undergoes through three types of clusters, i.e., 'R', 'M', 'B'.

The cluster 'R' allows the L-CVAS user to upload the batch video dataset to the L-CVAS cloud and configure three types of L-CVAS modules. The first type of service is Batch Video Acquisition Service, which is used to acquire batch video datasets. Once uploaded to the node buffer, the batch dataset is processed by the activated Video Processor to extract the metadata from the batch videos. After processing the batch video dataset and the respective metadata are persisted to the UPDDS and ISDDS, respectively. Similarly,

the cluster 'M' works the same way as that of cluster 'R', but this one is responsible for model management.

In the batch video analytics, the supporting layers deploy various contextual multi-domain offline BIVA services. This cluster loads the instance of the RIVA services as per user contract and processes the videos in an offline manner. Once subscribed, this cluster loads the batch video data set and model from the UPDDS. Similarly, the IR and anomalies are maintained in the ISDDS. The acquired video streams residing in the UPDDS is also illegible for offline analytics.

Finally, the Web Server 'W' deploys the Web User Interface (as described in IV-D), i.e., allow the users to interact with L-CVAS.

V. IVA; CONSTITUENTS, AND PREDOMINANT TRENDS

This section review the existing IVA literature and can be classified into four classes under the umbrella of IVA, i.e., CBVR, IVA Surveillance and Security, Video Summarization, and Semantics Approaches, as shown in Fig. 14. We also show that how L-CVAS can be used under these application areas.

A. CONTENT-BASED VIDEO RETRIEVAL

CBVR has applications from video browsing to intelligent management of video surveillance and analysis. To uphold advancement in CBVR, since 2001, the National Institute of Standards and Technology has been sponsoring the annual Text Retrieval Conference Video Retrieval Evaluation [161], [162]. The CBVR is an active research area, and several surveys are available, i.e., [32], [162]–[165]. However, here we discuss some of the scalable CBVR system being proposed in the literature.

In literature, some researchers tried to exploit distributed computing technology for the development of large-scale CBVR systems. Shang *et al.* [166] utilize the time-oriented temporal structure of videos and the relative gray-level intensity distribution of the frames as a feature base. Their method is expensive in terms of parallel processing due to the video semantics, and then all the frames of a video must be processed within the same execution environment. Hence this approach is challenging to parallelize accurately and efficiently even for the state-of-the-art big-data frameworks. Wang *et al.* [167] proposed a novel MapReduce framework called Multimedia and Intelligent Computing Cluster for near-duplicate video retrieval for large-scale multimedia data processing by joining the computing power of CPU's and GPU's to speed up the video data processing. They extract the keyframes using uniform sampling, store the keyframes to HDFS, perform local feature extraction using the Hessian-Affine detector [168] to detect interest points. K-means clustering over the feature vectors is utilized to generate visual words following the BoF [169] model, thus generating BoF-based feature vectors. Ding *et al.* [170] used big data processing technologies to design a human retrieval system on extensive surveillance video data called SurvSurf. Motion-based segments called M-clop were detected, which

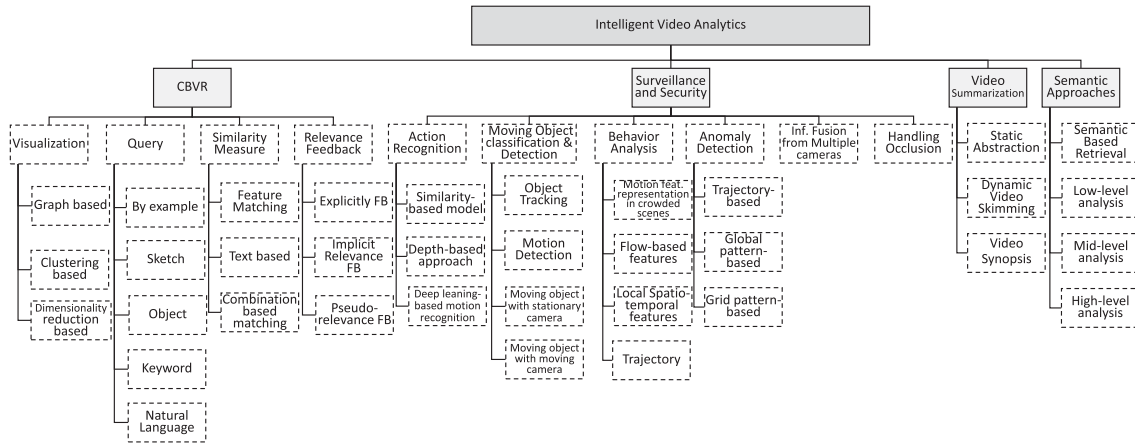


FIGURE 14. IVA taxonomy based on literature.

were utilized to remove redundant videos. Hadoop MapReduce framework was used to process M-clips for human detection and motion feature extraction. Vision algorithms were accelerated by processing only sub-areas with significant motion vectors rather than entire frames. Further, a distributed data store called V-BigTable on top of HBase was designed to structuralize M-clips’ semantic information and enables large-scale M-clips retrieval. They stated that Surv-Surf outperforms the baseline solutions in terms of computational time and with acceptable human retrieval accuracy.

Authors in [171] proposed Marlin for video big data similarity search. They used parallel computing to extract features from the acquired video micro-batches, which are then persisted in a distributed feature indexer. The proposed indexer was able to index incremental updates and real-time queries. They designed a fine-grained resource allocation with a resource-aware data abstraction layer over streaming videos to upsurge the system throughput. They reported Marlin achieved 25X and 23X speedup against the sequential feature extraction algorithm and similarity search, respectively. The challenge of extracting distinctive features is addressed by Lv *et al.* [172] for the efficiency of extraction closely related videos from the large scale data based on local and global features utilizing Spark. To balance precision and efficiency, they introduced a multi-feature based distributed system, including local and global features. They combined local feature SIFT, Local Maximal Occurrence, and global feature Color Name. Lastly, they developed the system in a distributed environment based on Apache Spark. Further, M. N. Khan *et al.* [173] proposed FALKON for large-scale CBVR that utilized distributed deep learning on top of Apache Spark for accuracy, efficiency, fault tolerance, and scalability. Motivated by the fact that Apache Spark, by default, does not provide native video data structure, they developed a wrapper on the top of Spark’s RDD called VidRDD. Utilizing VidRDD, first, they performed structural analysis on the videos, and then index the extracted deep spatial and temporal features in their designed distributed indexer. Finally, they evaluate

their proposed system and show performance improvement in terms of scalability and accuracy. Likewise, Lin FC. *et al.* [174] put forward a cloud-based face video retrieval system while utilizing deep learning. First, pre-processing operations like termination of blurry images, and face alignment are performed. Then the refined dataset is constructed and used to pre-train the CNN models, i.e., ArcFace, FaceNet, and VGGFace for face recognition. The results of these three models are compared, and the efficient one was chosen for the retrieval system development. The input query in their proposed system is a person’s name. If the system detects a new person, it performs enrolling that person. Finally, time-stamped results are returned against a query. A prototype of the proposed face retrieval system was implemented and reported its recognition accuracy and computational time.

1) CBVR UNDER L-CVAS ARCHITECTURE

L-CVAS provides an elegant and flexible six steps solution for the implementation and customization of scalable video indexing and retrieval, as shown in Fig. 15. First, the VSAS component acquires the video streams from VSDS in the form of mini-batches, and in the case of batch analytics, video data is loaded from the RAW DS and feeds to the VBDPL.

In the second step, VBDPL perform pre-processing operations and feature extraction. The former one encompasses structural exploration of a video, i.e., video scenes, shots detection, frames, and keyframes extraction, while the latter one extracts the low-level features. These low-level features can be keyframes’ static features (texture-based, color-based, and shape-based), object features, and/or motion features (trajectory-based, statistics-based, and objects’ spatial relationships-based). The extracted features are then handed over to the VBDML for classification and annotation.

In the third step, the semantic and high dimensional video feature vectors’ indexes make the representative index for persisted video sequences in IR-DS. The IR-DS is synchronized with RAW-DS.

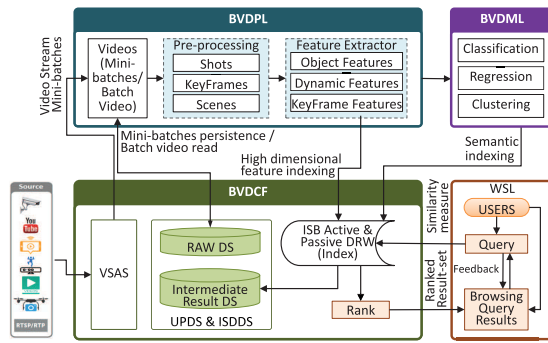


FIGURE 15. Components of CBVR under the Lambda CVAS architecture.

In the fourth step, through the WSL, the users are allowed to query the desired videos. In literature for video retrieval, various types of queries have been utilized for video retrieval. The queries can be categorized as Query by Example, Sketch, Objects, Keyword, and Natural Language. In the Query by Example, similar videos or images to the given sample video or image is extracted using feature similarity. In Query by Sketch, the features are extracted from users drawn sketches (sketch represent the required videos) and compared against the features generated from the stored videos [175]. In Query by Object, the user give an object image then the CBVR system search for all occurrences of the object in the video database [176]. Likewise, the last two approaches use Keywords and Natural Language [177] as query. Combination-based Query combines different types of queries for CBVR, was also adopted by some researchers such as [178], [179].

In the fifth step, the CBVR system search for similar videos against the user query. For similarity measure approaches can be classified as feature, text, Ontology and combination based matching. The similarity measure depends on the type of query. **Feature matching** based similarity measure is the average distance between the features of the corresponding frames [180]. Query uses low-level features for extracting relevant videos. The benefit of this approach is that the video similarity can be found easily from the features but is not appropriate for semantic similarity, which is conversant to users. In text matching based similarity measure, the name of each concept is matched with given query terms so that to find the videos that satisfy the query. The best example of this approach is that of Snoek *et al.* [181]. This approach is simple, but the query text must include the relevant concepts to get pleasing search results. In Ontology-Based Matching, the similarity between semantic concepts is measured utilizing ontology. Query descriptions are enriched from knowledge sources, such as the ontology of concepts. Adding extra concepts can improve the retrieval results [182] but can also decline search results unexpectedly. This approach is further explained in section V-D. Combination-based matching “leverages semantic concepts from a training collection by learning the combination strategies [86],

and query-class-dependent combination models [179]” [88]. Up to some extent, using this approach, the concept weights can be automated, and hidden semantic concepts can be handled but are difficult to learn query combination models.

Finally, the ranked result-set is presented to the user for browsing. To increase efficiency and to optimize the results of the CBVR system, many researchers use the obtained relevance feedback from the user. This feedback can be categorized as explicit, implicit, and pseudo feedback. In the first case, users are asked to select relevant videos from the previous results actively [183]. The explicit feedback approach obtains better results than the other two approaches, but direct user involvement is required. In the second case, the retrieval results are refined by exploiting the user’s interaction with the system, i.e., clicking pattern [184]. In the third case, there is no involvement of the users. The user’s feedback is produced through positive (closely related to the query sample) and negative sample (different from the query sample) from the previous retrieval. These samples are directed to the system for the next search. Yan *et al.* [185], and Hauptmann *et al.* [186] approaches are based on pseudo-relevance feedback approach. This approach substantially reduces the user interaction, but the semantic gap between low and high-level features obtained from different videos does not always agree with the similarities between the user-defined videos.

B. REAL-TIME INTELLIGENT VIDEO ANALYSIS AND SURVEILLANCE

In the context of growing security concerns, the surveillance meant to criminality and intrusion detection. Video surveillance is not limited to these, but it encapsulates all aspects of monitoring to capture the dynamics of diverse application areas, e.g., transportation, healthcare, retail, and service industries. A generic use case for RIVA has been shown in section IV-E, Fig. 12. Likewise, domain-specific RIVA services for security and surveillance can be created under L-CVAS architecture. In this section, we discuss recent literature on how the researchers advancing the activity and behavior analytics in the video streams with the aim of intrusion and crime detection, scene monitoring, and resource tracking.

1) VIDEO SEGMENTATION FOR ACTION RECOGNITION

A combination of numerous actions, objects, and scenes forms complex events [187]. Video analytics against complex events is a nontrivial task. Complex event detection demands the association of multiple semantic concepts because it is almost impossible to capture the complex event through a single event class label [188]. Video segmentation is required to mine informative segments regarding the event happening in the video. For effective event detection in video segmentation, it is vital to take into account the temporal relations between key segments in a particular event. The event videos hold intra-class variation, and several training videos are required to consider all possible instances of event classes.

Song and Wu [189] suggested a methodology to extract key segments automatically for event detection in videos by learning from a loosely tagged pool of web videos and images. For the positions of key segments and content depiction in the video, they used an adaptive latent structural SVM model and semantic concepts, correspondingly. They also developed two types of models, i.e., Temporal Relation Model and Segment-Event Interaction Model, for the temporal relations between video segments, and for evaluating the correlation between key segments and events respectively. They adapted labeled videos and images from the Web into their model and employed 'N' adjacent point sample consensus [190] for noise elimination. A knowledge-base was produced by Zhang *et al.* [191] to decrease the semantic gap between complex events. To effectively model event-centric semantic concepts, they used a large-scale of web images for learning noise-resistant classifiers.

Action recognition in videos encompasses both segmentation and classification. Action recognition can be tackled by sliding windows and aggregation in sequential as well as isolated manner, or by performing both tasks parallel. [188], [192] are some of the good examples of video segmentation based event detection. The related literature on video segmentation based action recognition can be classified as action segmentation, a depth-based approach, and deep learning-based motion recognition.

The most popular action segmentation model is dynamic time warping scheme [193]. Action segmentation from videos using an appearance-based method considers a comparison between the start and end frames of contiguous actions. Quantity of movement [194] and KNN along with HOG [195] are commonly used to identify the start and stop frames of actions. Similarly, for action recognition, depth-based approaches have been developed which consist of binary range-sample feature [196], capturing local motion and geometry information [197], histogram oriented 4D normal [198], and combination of depth motion map and HOG [199]. [200]–[202] have applied deep learning approaches to depth-based action recognition methods. Besides, for motion recognition, deep learning has been utilized in many ways. One is a suboptimal way, in which video represented as still images is fed to the channel of a CNN [96], [116]. Sometimes video as compact images are given as input to the already trained CNN to achieve good performance [203]. Video in the form of a sequence of images is input to the Recurrent neural network (RNN) for sequential parsing of video frames for long as well as short-term patterns [204], [205]. In order to introduce a temporal dimension, video is regarded as a volume and substitutes the 2D filters of CNN with 3D filters [206], [207].

In many application areas, e.g., resource tracking, action recognition, human behavior recognition, and traffic control, object tracking and motion detection in videos are vital. The process of detecting moving objects in videos is known as object tracking. For object tracking, initially, the foreground information is extracted in videos, and then the background

modeling of the scene is captured using a background subtraction algorithm, i.e., IAGMM [208], [209]. To increase the accuracy, subsequently, shadow elimination algorithms are applied to the foreground frame [210]. A connected component algorithm is used to determine the bounding box of an object. To ensure frame-to-frame matching of the detected object, a method such as adaptive mean shift [211] can be used for comparison. Factors like size and distance are used to object matching between frames. Finally, the occlusion scheme is used to detect and resolve occlusion. On the other hand, motion detection can be detected via foreground images extracted by the Gaussian Mixture Model background and connected component algorithm for noise removal. The area of detection is refined using a connected component algorithm and produces the bounding box information of the moving objects [208]. The output of the detection is a binary mask representing the moving object for each frame in a particular sequence. Object detection for moving objects is challenging, especially in the case of shadows and cloud movement [212]. The related literature for moving object detection and classification can be categorized as a stationary camera with a moving object and moving objects with moving camera.

a: MOVING OBJECT DETECTION WITH STATIONARY CAMERA

In fixed camera video, the background image pixels in each frame remain the same, and thus simple background subtraction techniques are required. The object detection approaches using a fixed camera can be grouped as feature-based, motion-based, classifier-based, and template-based models [213]. Categorization of object tracking in videos into point tracking, kernel tracking, and silhouette tracking as well as feature-based, region-based, and contour-based was performed by [213], [214] respectively. Unlike a fixed camera, moving object detection with a moving camera is relatively challenging because of camera motion and background modeling for generating foreground and background pixel fails [215].

- Trajectory classification involves computing long trajectories for feature point and discriminating trajectories that belong to different objects from those backgrounds using the clustering method. Some recommended algorithms include compensating long term motion based on flow optic technique [216], bag-of word classifier, and pre-trained CNN method for detecting moving object trajectories [217].
- In background modeling based methods, for each sequence, the frame-by-frame background is created utilizing the motion compensation method. Some popular algorithms are Mixture of Gaussian [218], complex homography [219], gaussian-based method [220], adaptiveMoG [221], multi-layer homography transform [222], thresholding [223], and CNN-based method [224] for background modeling.

- In extension of background subtraction method, low rank and sparse matrix decomposition method for static camera [225] are extended to moving camera. If there exists coherency between a set of image frames, then a low-rank representation of the matrix created by these frames contains the coherency, and the sparse matrix representation contains the outliers, which represents the moving object in these frames. Low rank and sparse decomposition involve segmenting moving objects from the fixed background by applying principal component pursuit. It is a valuable technique in background modeling. Mathematical formula and optimization of this method can be found in [226].

2) BEHAVIOR ANALYSIS

Usually, a camera is mounted nearby the digital displays to analyze and understand human behavior by investigating user interfacing with digital display [227]. Commercial tools have been developed to analyze audience behavior using video analytics while considering parameters like age, gender, distance from the display, and sight and spent time. The obtained data can then be used to improve advertising campaigns in combination with sales data [228].

Recently, crowd analytic, i.e., human detection and tracking, have attracted the attention of the researches. The exploration of both group and individual behavior to govern abnormality scope the crowd analysis. Congestion analysis, motion detection, tracking, and behavior analysis are the main attributes of crowd analytics. While performing crowd analysis, factors like terrain features, geometrical information, and crowd flow can be considered.

For analysis of the crowded scene, motion features are vital, and can be categorized as flow-based features, local spatiotemporal features, and trajectory features [229]. These features have applications in crowd behavior recognition, abnormality detection in-crowd, and motion pattern segmentation.

a: FLOW-BASED FEATURES

The flow-based features are pixel-level features, and in literature, different schemes have been proposed [230], [231], which be further classified as optical flow, partial flow, and streak flow. Optical flow technique encompasses computing pixel-wise motion between successive frames and can handle multi-camera objection motion. It has been applied to detection crowd motion and crowd segmentation [232]. This approach, optical flow, is unable to capture spatiotemporal attributes of the flow and long-range dependencies. Particle flow contains moving a grid of particles with the optical flow and providing trajectories that maps a particle's initial position to its future or current position. It has an application in crowd segmentation and detection of abnormal behavior [229]. Optical flow is unable to handle spatial changes. Mehran *et al.* [231] proposed streak flow to overcome the shortcomings of particle flow and to analyzing crowd video while computing motion

field. This approach, streak flow, captures motion information similar to particle flow; changes in the flow is faster and performs well in dynamic motion flow.

b: LOCAL SPATIOTEMPORAL FEATURES

Flow-based features fail on the very crowded scene, and resultantly local spatiotemporal features techniques are developed, which are 2D patches or 3D cubes representation of the scene. Spatiotemporal features can be categorized as spatiotemporal gradients, and motion histogram. To capture steady-state motion behavior, Kratz and Nishino [233] used a spatiotemporal motion pattern model and confirmed the detection of abnormal activities. On the other hand, motion histogram considers motion information within the local region. It is not appropriate for crowd analysis because it takes a substantial amount of time, and is subject to error. However, some improvements have been shown in the literature to motion histogram, e.g., [230], [234].

c: TRAJECTORY FEATURES

Trajectory features signify tracks in videos. The distance between object-based motion features can be extracted from the trajectories of objects and can be utilized to analyze crowd activities. The failure to obtain a full trajectory in dense crowd leads to the concepts of tracklet. The tracklets are extracted from the dense region and enforce the spatiotemporal correlation between them to detect patterns of behavior. Tracklet is a fragment of a trajectory obtained within a short period, and the occurrence of occlusion leads to closure. Tracklets have been used for human action recognition [31], [235] and for the representation of motion in crowded scenes [236], [237].

3) ANOMALY DETECTION

Anomaly detection an application area of crowd behavior analysis and is domain-dependent. Anomaly in a video occurs when the analyzed pattern drifts from the normal in a training video. The related literature of anomaly detection can be categorized into three, i.e., trajectory-based, global pattern-based, and grid pattern-based method of anomaly detection.

a: TRAJECTORY-BASED METHOD OF ANOMALY DETECTION

In trajectory-based anomaly detection, objects are formed from the segment scenes, and then the object is followed in the video. A trajectory is caused by the tracked object, which describes the behavior of the object [238]. For the evaluation of abnormality in trajectory-based methods have been used i.e., single-class SVM [239], zone-based analysis [240], semantic tracking [241], String kernels clustering [242], Spatiotemporal path search [243], and deep learning-based approach [244] have been used.

b: GLOBAL PATTERN-BASED METHOD OF ANOMALY DETECTION

In a global pattern-based technique, the video sequence is analyzed in whole, i.e. low, or medium-level features

are extracted from video using Spatiotemporal gradients or optical flow methods [245]. The technique is suitable for crowd analysis because it does not individually track each object in the video but is challenging while locating the position where the anomaly occurred. Approached used in the global pattern-based method are Gaussian Mixture Model [246], energy model [247], SFM [248], stationary-map [249], Gaussian regression [250], PCA model [251], global motion-map [252], motion influence map [253], and salient motion map [254].

c: GRID PATTERN-BASED METHOD OF ANOMALY DETECTION

In a grid pattern-based method, splits frames into blocks and individually analyze pattern on a block-level basis [255]. If ignoring inter-object connections that lead to processing efficiency. Spatiotemporal anomaly maps, local features probabilistic framework, joint sparsity model, mixtures of dynamic textures with Gaussian Mixture Model, low-rank and sparse decomposition, cell-based texture analysis, sparse coding and deep networks are used in evaluating grid pattern-based methods [256].

C. VIDEO SUMMARIZATION

Video big data are facing the challenge of sparsity and redundancy, i.e., hours of videos with less meaningful information, which creates many issues for viewing, mining, browsing, and storing videos. It has motivated researchers to find ways to shorten hours of videos and led to the field of video summarization. Video summarization is a process of generating a shorter video of the original one without spoiling the capability to comprehend the meaning of the whole video [257]. The video summarization can be classified as Static Video Abstracts, Dynamic Video Skimming, and Video Synopsis.

1) STATIC VIDEO ABSTRACT

These approaches include a video table of contents, a storyboard, and a pictorial video summary. For example, Xie and Wu [258] propose an algorithm to generate a video summary for broadcasting news videos automatically. An affinity propagation-based clustering algorithm is used to group the extracted keyframes into clusters, aiming to keep the relevant keyframes that distinguish one scene from the others and remove redundant keyframes. J. Wu *et al.* [259] were motivated by the notion from high-density peaks search clustering algorithm. They proposed a clustering algorithm by incorporating significant properties of video to gather similar frames into clusters. Finally, all clusters' centers were presented as a static video summary. Bhaumik *et al.* [260] proposed a summarization technique where they detect keyframes from each shot that eliminates redundancy at the intra-shot and inter-shot levels. For frames redundancy elimination, SURF and GIST feature descriptors were extracted for computing the similarity between the frames. The quality of the summaries obtained by using SURF and GIST descriptors are also compared in terms of precision and recall.

Similarly, Zhang *et al.* [261] propose a subset selection technique that leverages supervision in the form of human-created summaries to perform automatic keyframe-based video summarization. They were motivated by the intuition that similar videos share similar summary structures. The fundamental notion is to nonparametrically transfer summary structures from annotated videos to unseen test videos. Concretely, for each fresh video, they first compute the frame-level similarity between annotated and test videos. Then the summary structures are encoded in the annotated videos with kernel matrices made of binarized pairwise similarity among their frames. Those structures are then combined into a kernel matrix that encodes the summary structure for the test video. Finally, the summary is decoded by feeding the kernel matrix to a probabilistic model called the determinantal point process to extract a globally optimal subset of frames. M. Gygli *et al.* [262] used a supervised approach to learn the importance of the global characteristics in summary by extracting deep features of video frames. J. Mohan *et al.* [263] proposed a technique that utilizes sparse autoencoders. Motion vectors have been used for the elimination of redundant frames, and then high-level features are extracted from frames using sparse autoencoders. These high-level feature vectors are then clustered using the K-means algorithm. The frames closest to the centroid of each cluster are selected as keyframes of the input video. Ji, Zhong, *et al.* [264] employed tag information, i.e., titles and descriptions, as the side information for the generation of summarization. A sparse auto-encoder was used as the primary model to generate the final summary, where the input and output were multiple videos and keyframes set, respectively. They fused the visual and tag information to guide visual features, which constrained the sparse auto-encoder to select the candidate keyframes.

2) DYNAMIC VIDEO SKIMMING

A summary video is formed from the video segments of the original video to remove redundancy or to summarize based on object action or events. As an example of object-based skimming, Peker *et al.* [265] proposed a video skimming algorithm while utilizing face detection on broadcast video programs. In the algorithm, the attention was given to faces, as they establish the focus of most consumer video programs. Ngo *et al.* [266] represent a video as a complete undirected graph and exploit the normalized cut algorithm to form optimal graph clusters. One-shot was taken from each cluster of visually alike shots to remove duplicate shots. Xiao *et al.* [267] mine frequent patterns from a video. A video shot importance evaluation model is utilized to choose useful video shots to create a video summary. For personal videos, Gao *et al.* [268] developed a video summarization technique, which encompasses a two-level redundancy detection procedure. First, they terminated redundant video content with the hierarchical agglomerative cluster method at the shot level. Then parts of shots were selected, based on the ranking of the scenes and keyframes, to generate an initial video summary.

Finally, to terminate the redundant information, a repetitive frame segment detection step was utilized. They verified the proposed technique through a prototype while using TV datasets (movies and cartoons videos) and reported the performance in terms of compression ratio (81%) and recall (87.4%). In [269], for user-generated video summarization, both the representativeness and the quality of the selected segments from an original video were considered. They stated that user-generated videos contain semantic and emotional content, and its preservation is vital. They have designed a scheme to pick representative segments that include consistent semantics and emotions for the whole video. To ensure the quality of the summary, they computed quality measures, i.e., motion and lighting conditions, and integrate them with the semantic and emotional clues for segment selection.

Ji, Zhong, *et al.* [270] addresses the issue of supervised video summarization by formulating it as a sequence-to-sequence learning problem, where the input and output is a sequence of original video frames and a keyshot sequence, respectively. The notion is to learn a deep summarization network with attention mechanism to mimic the way of selecting the human keyshots. The proposed framework was called attentive encoder-decoder networks for video summarization. They utilized the BiLSTM encoder for encoding the contextual information among the input video frames. For the decoder, two attention-based LSTM networks, are explored by using additive and multiplicative objective functions, respectively. The results demonstrate the superiority of the proposed framework against the state-of-the-art approaches, with remarkable improvements. J. Wu *et al.* [271] were motivated by the fact that multi-video summarization is significant for video browsing and proposed a technique where multi-video summarization was formulated as a graph problem. They also introduced a dynamic graph convolutional network to measure the importance and relevance of each video shot locally as well as globally. They adopted two approaches to address the inherent class imbalance issue of video summarization. Additionally, a diversity regularization to encourage the model to generate a diverse summary was introduced. The results demonstrate the effectiveness of our proposed model in generating a representative summary for multiple videos with encouraging diversity. Z. Sheng-hua *et al.* [272] proposed a deep learning-based dynamic video summarization model. First, they addressed the issue of the imbalanced class distribution in video summarization. The over-sampling algorithm is used to balance the class distribution on training data. They proposed two-stream deep architecture with cost-sensitive learning to handle the class imbalance problem in feature learning. RGB images are utilized to represent the appearance of video frames in the spatial stream. Likely, multi-frame motion vectors with deep learning framework are introduced to represent and extract temporal information of the input video. Moreover, they stated that the proposed method highlights the video content with the active level of arousal in effective computing tasks and can automatically preserve the connection between consecutive frames.

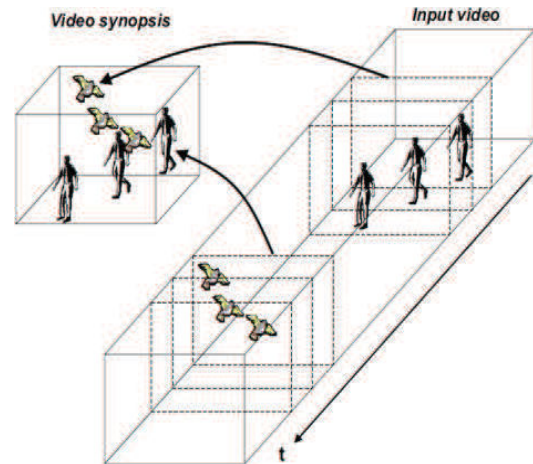


FIGURE 16. Simultaneous activities display in a video [273].

3) VIDEO SYNOPSIS

In this approach, activities from the stated time interval are collected and moved in time to form a smaller video synopsis showing maximum activity, as shown in Figure 16. The notion of video synopsis was pioneered by [273] in 2006 and proposed a two-phase approach, i.e., online and offline. The former phase includes the queuing of the generated activities. The later phase started after selecting a time interval of video synopsis with tube readjustment, background formation, and object stitching. A global energy function was defined and encompassing activity, temporal consistency, and collision cost. Then the simulated annealing method was applied for energy minimization. The video synopsis domain was further researched in single and multi-view scenarios. Some recent examples of a single-view are [274]–[276]. He *et al.* [274], [275] brought advancement in activity collision analysis by describing collision statuses between activities such as collision-free, colliding in the same direction and opposite directions. They also offered a graph-based optimization technique by considering these collision states to improve the activity density and put activity collisions at the center of their optimization strategy. Baskurt and Samet [276] concentrated on rising robustness of object detection by suggesting an adaptive background generation method. In another study, Baskurt and Samet [277] planned the object tracking method specified for video synopsis requirements. Their approach focused on long term tracking to represent each target with just one activity in video synopsis. Single view scalable approaches for video synopsis were projected by Lin *et al.* [278] while utilizing distributed computing technology. Their proposed video synopsis approach encompasses steps like object detection, tracking, classification, and optimization, which were performed in a distributed environment. Ahmed, S. A. [279] proposed a query-based method to generate a synopsis of long videos. Objects were tracked and utilized deep learning for objects classification (e.g., car, bike, etc.). Through unsupervised clustering, they identified regions in the surveillance scene. The source and the

destination represented spatiotemporal object trajectories. Finally, user queries were allowed to generate video synopsis by smoothly blending the appropriate tubes over the background frame through energy minimization.

Further, the examples of multi-view video analytics are that of [280], [281]. Zhu *et al.* [280] proposed a framework to generate a unified synopsis of multi-view videos. The synopsis is visualized by mapping multiple views to a common ground plane. Multiple camera activities were allied via trajectory matching in overlapping camera views. The process of producing a synopsis requires a balance among minimizing the synopsis length, maximizing the information coverage, and reducing the collision among objects' track that are presented concurrently. Likewise, Mahapatra *et al.* [281] proposed a multi-camera approach for an overlapping camera network and modeled the synopsis generation as a scheduling problem. They utilized three distinct methods, i.e., table-driven approach, contradictory binary graph coloring approach, and simulated annealing. Action recognition modules were integrated to recognize significant actions, i.e., walking, running, bending, jumping, hand-shaking, and one or both hands waving. The inclusion of such essential actions can help in the synopsis length reduction while preserving the value. The synopsis length was further reduced while utilizing a fuzzy inference system that computes the visibility score for each object tracking. They stated that through the contradictory binary graph coloring approach, they achieved a maximum reduction in synopsis length. Zhang, Z. *et al.* [282], tried to address the issue of video synopsis by joint object-shifting and camera view-switching to show multiple synopsis results more compactly and understandably. The input videos were synchronized and grouped the same object in different videos together. Then they shifted the grouped objects with respect to the time axis to gain multiple synopsis videos. They constructed a simultaneous object-shifting and view-switching optimization framework to achieve encouraging synopsis results. To address unified optimization, they further presented an optimization strategy composed of graph cuts and dynamic programming.

VIDEO SUMMARIZATION *aaS* UNDER L-CVAS

Fig. 17 shows the flow of the video summarization flow under the proposed L-CVAS architecture. Through the WSL, the user first subscribes to the video data-source to the video summarization service. Then the user preferences allow the users to set the parameters required for video summary service and personalization. The summary parameters encompass granularity level, type of summary to be performed (e.g., overview, highlights, synopsis, etc.) and any other as per the video summary service scenario. The personalization can be in terms of specific features of the video, like people, objects, events, etc. Through the VBDCL, videos are acquired from the video data-source and sent to the VBDPL for pre-processing. In pre-processing, video units are extracted (segmentation, shots, frame extraction, etc.) as per the requirements of the video summarization service. Then

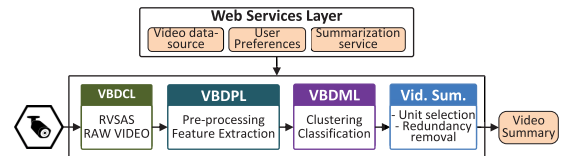


FIGURE 17. Generic flow of video summarization *aaS* under Lambda CVAS architecture.

multiple low and high-level features, such as motion, color, aesthetics, semantics, etc., are extracted from the video units. The extracted features from the basic video units are input to the VBDPL for object/activity identification and clustering. Once done then the next step is video summarization. The video summarization phase deploys the actual logic, i.e., video unit selection, and redundancy removal. The video unit selection and redundancy removal decide which video units should be included in the video summary based on unit significance, summary length, and other user's parameters. This block also removes similar video units within the video summary to achieve the best possible video summary, covering the required details in the original video. Finally, the summary results are delivered to the respective user through the WSL.

D. SEMANTIC-BASED VIDEO ANALYSIS

To bridge the semantic gap and to allow the machine to understand the visual data, Semantic Web technologies can be incorporated [160]. Semantic Web unlocked a new avenue for knowledge-based computer vision while enabling data exchange between video analysis systems in an open and extensible manner. The scientific community has exploited the Semantic Web concept for intelligent video analytics to bridge the so-called semantic gap between the low-level features and high-level human-understandable concepts. State-of-the-art scholarly work can roughly be classified as semantic-based low-level, mid-level, high-level analysis, and semantic-based video search and retrieval.

1) SEMANTIC-BASED LOW-LEVEL ANALYSIS

Semantic-based low-level analysis refers to the formalization of the extracted objects of interest from the videos. It then performs reasoning on formalization such as detection and tracking across domain-specific videos [283], [284]. Dasiopoulou *et al.* [285] proposed a multimedia ontology for domain-specific video analysis. In semantic concepts, they consider object attributed, low-level features, spatial object relations, and the processing approaches while defining F-logic rules for reasoning that govern the application of analysis methods. García *et al.* [286] proposed a knowledge-based framework for video object segmentation, where relationships among analysis phases are utilized. The main contribution is to provide a detailed description of the scene at low, mid, and high semantic levels through an ontology. The notion is to offer the semantic rich description of a

scene via an ontology that includes occurrences in the scene from high to low semantic level, controls iterative decisions on every stage. The low-level analysis modules (background subtraction, short-term change detection, point or region tracking, etc.) are provided with a structure to collaborate and achieve consistently, and contextual results. The results of the vision algorithms are mapped to the ontology, representing low-level scene occurrences. The following stages build Point Hypothesis Maps and Region Hypothesis Maps. These were the most probable occurrences of each point and region. The points and regions are coded according to the ScenePoint and SceneRegion hierarchies of the analysis ontology. The quality of the results was evaluated through a feedback path. Gomez *et al.* [287] proposed a computer vision framework for surveillance and consists of two layers, i.e., the tracking layer and a context layer. The proposed framework relies on an ontology-based representation of the scene in combination with contextual information and sensor data. The notion is the application of logical reasoning initiating from the acquired data from a classical tracker intending to construct an ontological model of the objects and the activities happening in the area of observation. Reasoning procedures were utilized to detect and predict tracking errors, sending feedback to the tracker to adjust the low-level image processing algorithms. Vision operation, like movement detection, blob-track association, and track and trajectory generation, were performed in the tracking layer. The context layer was supposed to produce a high-level interpretation of the scene. RACER reasoner [288] was utilized for scene interpretation since it allows abductive reasoning. Abductive rules are defined in the proposed framework to interpret what is happening in the scene from the primary tracking data.

2) SEMANTIC-BASED MID AND HIGH-LEVEL ANALYSIS

Atomic events such as loitering, fall, direction changes, group formations, and separations [289], and “complex” events such as aggressions, fights, thefts, and other general suspicious events [290], [291] falls in mid and high-level video analytics, respectively.

The utilization of semantic technology for video event representation in surveillance videos was initiated by Video Event Representation Language [292]. The main idea is to model simple events in a hierarchical framework intended for detecting complex events. They stated that the sequence of simple events (car door opening, leaving a car, car door closing, walking and opening a building door, and entering a building) forms complex events (a person arrived through a car and entered a building). They used Allen’s interval algebra to handle temporal relationships between subevents. They clarified their proposed application through the detection of an example event in a surveillance video, i.e., accessing a secure zone by entering behind an authorized individual. For complex events recognition in surveillance videos, Snidaro *et al.* [293] proposed an ontology and is composed of three high-level concepts, i.e., background, entities, and events. The event class, high-level in subclasses, describes

simple events, spatial events, and transitive events, allowing to show how complex events can be described through simple events sequencing. The event concept is composed of sub-concepts that represent simple events, spatial events, and transitive events. The focus was on complex events, which can be achieved by sequencing simple events. San-Miguel *et al.* [294] propose an ontology for representing the prior knowledge related to video event analysis. Such knowledge is described in terms of scene related entities (Object, Event, and Context), and system-related entities. The key contribution of the work is the integration of different types of knowledge in an ontology for detecting the objects and events in a video scene. In the same direction, Greco *et al.* [295] proposed a hybrid approach for simple abnormal (person falling) and complex abnormal events (person aggression) recognition using semantic web technologies. They modeled the extracted general tracking information to the proposed tracking ontology for advanced reasoning. The data from the videos were obtained using the tracking component (frames, bounding box), knowledge about the scene (static and dynamic objects, occluding objects), Situations and Events (people leaving the scene, falling ground, fighting). For event detection, SPIN rules and functions are used while SPARQL queries are employed for analytics tasks. The system has proven to successfully recognize mid-level events (ex. people falling to the ground) and high-level events (ex. person being attacked) on the PETS2016 dataset.

Researches also utilized semantic technologies aiming to address the issue of human activity recognition in daily living. In this context, Chen *et al.* [296] introduced a method for activity recognition while using ontological model, representation, and reasoning. They analyzed the nature and characteristics of daily life activities and modeling related concepts through ontologies. The authors describe the algorithms of activity recognition making full use of the reasoning power of semantic modeling and representation. They claimed that the proposed ontological models for daily life activities could easily be customized, deployed, and scaled up. Likewise, [297], an approach exploiting the synergy between the semantic technologies and tracking methods have been presented for object labeling. The work aims to augment and comprehend situation awareness, as well as critical alerting conditions. The unmanned aerial vehicles with an embedded camera were utilized to recognize moving and stationary objects along with relations between them. Contextual information was used for abnormal event detection. A prototype was designed and used a drone to capture videos on the University of Salerno. They stated that the proposed system could recognize an abnormal event by means of SWRL rules associated with mid-level activities, such as ball kicking by a human and a car passing through the same road.

3) SEMANTIC-BASED VIDEO RETRIEVAL

Some researchers exploited semantic technologies for video search and retrieval purposes. In this regard, Yao *et al.* [298] proposed the image to text framework to extract events from

TABLE 9. Scholarly state-of-the-art-work CVAS.

Ref	IVA			VDM		Core Technology			Performance Matrix	(a)Objectives (b)Target Domain (c) Issues
	R	B	Service	St	S	Storage	DP	Lib.		
[314]	✓	✓	-Face detection -Motion detection	✓	✓	MS SQL Server	✓	OpenCV	Workload for - Face detection - Storage avg. task time	(a) CVAS Architecture, Implements a prototype (b) NA (c) Not scalability and SOA
[315]	✓	✓	- Bg. Subtract -Object classification -Object tracking	✓	✓	-	-	-	✓	(a) CVAS Architecture, Implements a prototype. (b) Surveillance (c) Not scalable, consumer only, no evaluation.
[312]	✓	✓	Video compression	✓	✓	HDFS	MR	-	Scalability	(a) Merge-Split Based Video encoding. (b) Storage and transmission. (c) Video compression only.
[305]	-	✓	-	✓	✓	-	-	-	-	(a) Proposed video surveillance architecture for video analytics (b) Surveillance (c) Based on B2C models, and not SOA.
[306]	✓	x	-	✓	✓	HDFS, Hbase	-	-	-Bandwidth usage -Scalability	(a) Cloud-based architecture for video recordings. (b) Surveillance (c) IVA not supported
[307]	✓	✓	-Video summary - Encoding - Decoding - Bg. Subtract	✓	✓	HDFS, MySQL	MR	OpenCV, CUDA	-Throughput -Average I/O rate -Execution Time	(a) Distributed video management platform. (b) Distributed video management. (c) Not SOA, video stream not scalable.
[105]	✓	✓	-Face Recognition, -Motion Detection -Tracking	✓	✓	HDFS	MR	OpenCV, FFMPEG JavaCV	-Scalability	(a) Fast and distributed video processing. (b) Video surveillance. (c) Distributed video processing only.
[316] [317]	✓	✓	Object recognition and tracking vehicle classification	✓	✓	HDFS, HBase, SploutSQL	MR Storm	OpenCV	[8] -Accuracy Recognition [9] -Fault Tolerance, -Scalability	(a) Proposed a cloud platform for online and offline video analytics. (b) Security, surveillance, and traffic. (c) Not SOA,
[51]	✓	✓	Face recognition & counting	✓	✓	HDFS, Splout-SQL	MR Storm	OpenCV	-Scalability -Fault tolerance	(a) General cloud architecture for Video Analytics. (b) Security and transportation surveillance (c) Not SOA
[318]	✓	✓	-Face detection -Action recognition	✓	✓	HDFS, Splout-SQL	MR Storm	OpenCV	-Scalability -Fault tolerance	(a) CVAS Platform (b) Security and surveillance

B=Batch IVA, R=Real-time IVA, St = Video Stream Management, S = Video Storage Management, DP = Distributed Processing, MR = MapReduce

images (or video frames) and then provide semantic and text annotations. The And-or-Graph incorporates vocabularies of visual elements like objects and scenes along with stochastic image grammar that identifies semantic relations among the visual elements. In this way, low-level image features are linked with high-level concepts, and the parsed image can be transformed into semantic metadata to form the textual description. Video contents are expressed in both OWL and text format, and then the users are allowed to search images and video clips through keyword searching and semantic-based querying. Xue *et al.* [299] proposed an ontology-based content archive and retrieval framework for surveillance videos. A surveillance ontology was proposed that represents semantic information of video clips as a resource ontology. Such an ontology models the basic feature description in the low level, the video object description in the mid-level, and event description in the high-level. The proposed system was tested for object and event retrieval, such as walking and car parking.

Furthermore, Xu *et al.* in [300] and [301] propose a method to annotate video traffic events while considering their

spatial and temporal relations. They introduced a hierarchical semantic data model called structural video description, which consists of three layers, i.e., pattern recognition layer (ontological representation from the video the extracted video concepts), video resources layer (links video resources with their semantic relations), and demands layer (retrieval interface). They defined various concepts in the ontology, such as persons, vehicles, and traffic signs that can be used to annotate and represent video traffic events. Besides, the spatial and temporal relationships between objects in an event are defined. As a case study, an application to annotate and search traffic events is considered. Sah *et al.* [302] proposed a multimedia standard-based semantic metadata model and annotate globally inter-operable data about abnormal crowd behaviors from surveillance videos. Similar efforts are made by Sobhani *et al.* [303] and proposed an advanced intelligent forensic retrieval system by taking advantage of an ontological knowledge representation while considering the UK riots in 2011 as a use case. Similarly, A. Alam *et al.* [304] proposed a layered architecture for large-scale distributed intelligent video retrieval while exploiting deep-learning and semantic

approaches called IntelliBVR. The base layer is responsible for large-scale video data curation. The second and third layers are supposed to process and annotate videos, respectively while using deep learning on the top of a distributed in-memory computing engine. Finally, the knowledge curation layer, where the extracted low-level and high-level features are mapped to the proposed ontology, can be searched and retrieved using semantic rich queries. Finally, they projected the effectiveness of IntelliBVR through experimental evaluation.

VI. STATE-OF-THE-ART CVAS

In this section, we review the start-of-the-art CVAS. This discussion also supports our claim on the relationship between video big-data analytics and cloud computing. The discussion is further divided into two subsections, i.e., Scholarly CVAS and Industrial CVAS.

A. SCHOLARLY CVAS

In this subsection, we explore academic research trends (summarized in table 9) that how scientific community investigate and proposed cloud-based IVA solutions while utilizing big data technologies. In this direction, Ajiboye, S.O. *et al.* [305] stated that the network video recorder is already equipped with intelligent video processing capabilities but complained about its limitations, i.e., isolation, and scalability. To resolve such issues, they proposed a general high-level theoretical architecture called Fused Video Surveillance Architecture (FVSA). The design goals of the FVSA were cost reduction, unify data mining, public safety, and scalable IVA. The FVSA architecture consists of four-layer, i.e., Application layer (responsible for system administration and user management), Services Layer (for storage and analytics), Network Layer, and Physical Layer (physical devices like camera, etc.). They guaranteed the compatibility of FVSA with the hierarchical structure of computer networks and emerging technologies. Likewise, Lin, C.-F. *et al.* [306] implemented a prototype of a cloud-based video recorder system under gIaaS while using big data technologies like HDFS and Map Reduce. They showed the scalable of video recording, backup, and monitoring features only without implementing any video analytics services. Similarly, Liu, X. *et al.* [307] also came out with a cloud platform for large scale video analytics and management. They stated that the existing work failed to design a versatile video management platform in a user-friendly way and to effectively use Hadoop to tune the performance of video processing. They successfully develop a cloud platform and the same big data technologies, i.e., Hadoop and MapReduce. They also managed to develop three video processing services, i.e., video summary, video encoding and decoding, and background subtraction.

Tan, H. *et al.* [105] used Hadoop and MapReduce for fast distributed video processing and analytics. They developed two video analytics services, i.e., face recognition and motion detection, by using JavaCV. Furthermore, Ryu, C., *et al.* [308] proposed a cloud video analytics framework using HDFS and

MapReduce along with OpenCV [309] and FFmpeg for video analytics. They implemented face recognition and tracking algorithm and reported the scalability of the system and the accuracy of the algorithm. Ali M. *et al.* [310] proposed an edge enhanced stream analytics system for video big data called RealEdgeStream. They tried to investigate video stream analytics issues by offering filtration and identification phases to increase the value and to perform analytics on the streams, respectively. The stages are mapped onto available in-transit and cloud resources using a placement algorithm to satisfy the Quality of Service constraints recognized by a user. They demonstrate that for a 10K element data streams, with a frame rate of 15-100 per second, the job completion took 49% less time and saves 99% bandwidth compared to a centralized cloud-only based approach.

White *et al.* [311] researched MapReduce for IVA services, which comprises classifier training, clustering, sliding windows, bag-of-features, image registration, and background subtraction. However, experiments were performed for the k-means clustering and Gaussian background subtraction only. Tan and Chen [105] presented face detection, motion detection, and tracking using MapReduce-based clusters on Apache Hadoop. They utilized JavaCV since Hadoop is developed and designed for Java. Pereira, R. *et al.* [312] proposed a cloud-based distributed architecture for video compression based on the Split-Merge technique while using the MapReduce framework. They stated that they optimized the Split-Merge technique against two synchronization problems. The first optimization problem was split and merge video fragments without loss in synchronization, whereas the second optimization problem is the synchronization between audio and video can be greatly affected, since the frame size of each one may not be equal. Similarly, Liu *et al.* [313] used Hadoop and MapReduce for video sharing and transcoding purposes.

Zhang, W. *et al.* [51] proposed a cloud-based architecture for large scale intelligent video analytics called BiF. BiF combines the merits of RIVA and BIVA while exploiting distributed technologies like storm and MapReduce, respectively. BiF architecture considered non-functional architectural properties and constraints, i.e., usability, scalability, reliability, fault tolerance, data immutability, re-computation, storing large objects, batch processing capabilities, streaming data access, simplicity and consistency. The BiF architecture consists of four main layers, i.e., data collection layer, batch layer, real-time layer, and serving layer. The data collection layer collects the streaming video frames from the input video sources (camera). The data collection layer forwards the video frames to the batch layer and streaming layer for batch processing and real-time analytics, respectively. The service layer is to query both batch views and real-time views and integrate them to answer queries from a client. To evaluate the performance of the BiF architecture, they developed a video analytics algorithm, which was able to detect and count faces for a specific interval of time from the input source. During the evaluation, they showed that BiF is efficient in

terms of scalability and fault tolerance. Zhang *et al.* [319] introduced Apache Kafka and Spark Streaming framework for efficient real-time video data processing. They also proposed a fine-grained online video stream task management scheme to boost resource utilization and experimented with license plate extraction and human density analysis.

Azher *et al.* [318] proposed CVAS for RIVA and BIVA while using Spark Stream and Spark, respectively. They implemented IVA services such as human action recognition and face recognition services, respectively. In another work, Azher *et al.* [320] proposed a novel feature descriptor to recognize human action on Spark while utilized the Spark MLlib [142] to recognize the action from the feature vector generated by ALMD [320]. Wang *et al.* [321] also performed human action recognition on Spark. The aim was to speed up some key processes, including trajectory-based feature extraction, Gaussian Mixture Model generation, and Fisher Vector encoding. Distributed video processing called streaming video engine is also introduced in [322] for distributed IVA framework at Facebook scale against three major challenges, i.e., low latency, application-oriented flexibility, and robustness to faults and overload.

Zhang *et al.* [316], [317] stated that the historical video data could be used with the updated video stream to know the current status of an activity, e.g., status of traffic on the road, and to predict future. To make it possible, they proposed a video cloud-based service-oriented layered architecture called Depth Awareness Framework and consists of four layers, i.e., data retrieval layer, offline video analytics layer, online video processing layer, and domain service layer. The data service layer is supposed to handle large-scale video data and Webcam Stream. The offline layer is used to perform the operation on the batch videos, whereas online processing occurs in a real-time video processing layer. On the top of the proposed cloud platform, they implemented deep convolution neural network for obtaining in-depth raw context data inside the big video, and a deep belief network-based method to predict workload status of different cloud nodes, as part of knowledge on a system running status. They prepared a dataset consisting of seven traffic videos, each of size 2GB. During the evaluation, they stated the improvement in object prediction accuracy, fault tolerance, and scalability. Zhang *et al.* [323] performed pedestrian recognition on real-time video data using deep learning. Here, the CNN network is improved to fine-CNN, which consists of a nine-layer neural network. Moreover, the Apache Storm framework, along with a GPU-based scheduling procedure, is presented.

B. INDUSTRIAL CVAS

Various leading industrial organizations have successfully deployed CVAS. Some of the most popular are briefly described in the following subsections.

1) GOOGLE VISION

On March 8, 2017, at the Google Cloud Next conference in San Francisco, Google announced the release of

the IVA REST API [327]. The API lets the developer recognize objects in videos automatically and can detect and tag scene changes. Furthermore, it enables the users to search and discover the unstructured video contents by providing information about entities (20,000 labels). Its main features are label detection, explicit content detection, shot change detection, and regionalization [330]. It exploits deep-learning models and is built on the top of the TensorFlow framework. The Google IVA APIs is targeting the unstructured video content analytics rather than surveillance and security. The application domains of the API can be large media organizations that want to build their media catalogs or find easy ways to manage crowd-sourced content. It can also be helpful for product recommendations, medical-image analysis, fraud detection, and many more.

2) IBM CVAS

In April 2017 at the National Association of Broadcasters Show, IBM announced CVAS services [331]. BlueChasm, [328] development team, came up with a prototype app, known as “VideoRecon,” that combines IVA via IBM Watson and IBM cloud stack. The IBM CVAS service can extract metadata like keywords, concepts, visual imagery, tone, and emotional context from video data. The IBM CVAS allows the users to upload video footage to the IBM Clever-safe object storage [332] and subscribe to a service. When an object or event of interest is detected, the VideoRecon service creates a tag along with a timestamp of the point in the video when either the object was recognized or the event occurred. The tags are then stored in the IBM Cloudant fully managed NoSQL JSON document store [333] for future use.

3) AZURE CVAS

Microsoft Azure, a cloud computing service launched in 2010, started media services that enable developers to build scalable media management and delivery applications [329]. Media Services is based on REST APIs that enable the users to securely manage video or audio content for both on-demand and live streaming delivery to clients. Recently, they provide CVAS APIs to the customers for (R/B)IVA (as shown in Table 10).

4) CITILOG CVAS

Citilog [324], also known as CT-Cloud, provides intelligent video analytics and surveillance solutions in the domain of transportation. Citilog provides services like automatic incident detection, traffic data collection (vehicle counting, classification, average speed, occupancy and levels of service), interaction control, video management, and license plate recognition. The Citilog is an open platform, providing APIs, widgets for quick development of services. According to the Citilog, they process approximately 32000 hours of video data and detects about five incidents per minute from 900 sites worldwide.

TABLE 10. Industrial CVAS.

CVAS	Target Domain	RIVA Services	BIVA Services	BM	API
Citilog (1997) [324]	Security & Surveillance (Transportation, Highways, Tunnels, Bridges, and smart cities).	Incident detection, traffic data collection, interaction control, license plate recognition	NA	B2B, B2C	✓
Check-Video (1998) [325]	Security & Surveillance (education, guard, airports, retail, construction, services, residential, outdoor assets)	Basic Analytics (motion detection, zone violation, camera tamper) Object classification Business Analytics (object counting, tracks, transaction)	NA	B2C	✗
Intelli-Vision (2002) [326]	Security & Surveillance (home & IoT, retail, traffic & transportation)	Smart home, Smart security, Smart retail, Smart city, and ADAS	Video Search, video summary and, recognition	B2B, B2C	✓
Google Vision (2017) [327]	Video contents & analytics	NA	Label Detection (detect objects), Content Detection (Shot change and adult contents), detection (detect scene changes)	B2C, B2B	✓
IBM CVAS (2017) [328]	Video contents & analytics	NA	Deep learning based IVA (scenes, objects, faces)	B2B C2B	✓
Azure CVAS [329]	Video contents analytics & surveillance	Motion detection, Face detection, Emotion detection (sadness, fear, anger)	HyperLapse, summarization, character recognition, content moderation, face redactor	B2B C2B	✓

5) CHECK-VIDEO CVAS

Check-Video [325], founded in 1998, is one of the initiators in the domain of IVA. The company offers cameras, recorders, and a cloud video management solution for security dealers, integrators, and end-users. Initially, they used to provide OVAS, but with the advancement of cloud technology, CheckVideo launched CVAS. They provide domain-specific intelligent video analytics RIVA solution. The main features Check-Video are RIVA, a video search engine, cloud video storage, and an alert system. The provided services can be categorized as basic analytics, object classification, and business analytics (see table 10). According to the company's, they have successfully analyzed 108,458,000 hours of video and detected 61,233,000 events per month.

6) INTELLI-VISION CVAS

Intelli-Vision [326], founded in 2002, is a leading and notable company in the field of Artificial Intelligence (AI) and deep learning-based video analytics and video cloud platform. They are exploiting state-of-the-art technology in the area of AI for security and monitoring purposes while targeting multiple business domains, including home, retail, transportation, and advanced driver assistance systems for cars. Intelli-Vision's analytics adds the "Brains Behind the Eyes" for cameras by analyzing the video content, extracting metadata, sending out real-time alerts, and providing intelligence on the video. Currently, they are providing a wide range of video analytics services in the domains mentioned above, ranging from object left to night vision and enhancements (see table 10). In Feb 2018, in a press release, the Intelli-Vision stated that they have successfully deployed four million cameras worldwide, which have been subscribed to various IVA services.

VII. IVA APPLICATIONS

IVA at scale drives many application domains ranging from security and surveillance to self-driving and healthcare. Many application areas of video big data analytics are shown, which project the significant role of big data and cloud computing in IVA.

A. TRAFFIC AND TRANSPORTATION

IVA has been extensively used in traffic control and transportation, e.g., lane traffic counts, incident detection, illegal u-turn, and many more. One of the main reason for deaths and injuries are traffic-related misfortunes [334]. Proactive analytics is required to predict abnormal events so that to minimize or avoid such accidents. In this direction, VisonZero [335] has been developed and deployed successfully. In transportation, another application is vehicle tracking where chasing of a license plate, overspeeding, and collision cause analysis can be obtained by analyzing video data. Kestrel [336] is a vehicle tracking system and uses information from various non-overlapping cameras to detect vehicle path. Gao *et al.* [337] used an automatic particle filtering algorithm to track the vehicle and monitor its illegal lane changes. Chen *et al.* citechen2009machine used hidden Markov models to determine the traffic density state probabilistically. Incident detection framework based on generative adversarial networks were proposed in [338].

B. INTELLIGENT VEHICLE AND SELF-DRIVING CARS

Currently, the term self-driving cars mean that the vehicle exploits computer vision for safe and intelligent driving while assisting the driver. In an intelligent vehicle, different sensors and high definition cameras (cameras for vehicle cabin, forward roadway, and the instrument cluster) are integrated with

the vehicle, which generates multi-model data, and the same is sent to the cloud for real-time analytics [339], [340]. In this context, video analytics is vital with optimum algorithm accuracy. Researchers have developed several algorithms, including pedestrian detection, traffic light detection, and other driver assistance system. For example, Wang *et al.* [341] proposed a method for pedestrian detection in urban traffic conditions using a multilayer laser sensor mounted onboard a vehicle. An algorithm was proposed by Tsai *et al.* [342] to detect three condition changes: missing, tilted, and blocked signs, using GPS data, and video log images. An innovative CNN-based visual processing model is proposed in [343] to automatically detect traffic signs and dramatically reduces the sign inventory workload. Driver decision making was improved in taking the right turn in left-hand traffic at a signalized intersection utilizing simulation [344]. They used an in-car video assistant system to present the driver's occluded view when the driver's view is occluded by truck. An effort for driver body tracking and activity analysis, posture recognition, and action predication, have been studied in [345], [346] respectively.

C. HealthCare

Recently, video big data analytics is reshaping the health-care industry and yet another vital application area that demands special focus while deploying video analytics. Surveillance video streams can help understand the tracked person's behavior, such as monitoring elderly citizens or blind people against fall detection or detecting any possible threat. Fleck and Strasser reported a prototype 24/7 system installed in a home for assisted living for several months and shows quite promising performance [347]. Zhou *et al.* studied how video analytics can be used in eldercare to assist the independent living of elders and improve the efficiency of eldercare practice [348]. Some more studies were done [349]–[352] to analyze activities, recognize posture, and to detect falls or other substantial events. A smart gym can exploit the video cam stream to determine frequently used equipment, the duration of exercise, and time spent on a piece of particular equipment, which is useful for real-time assessment.

D. SMART CITY SECURITY (IoT) AND SURVEILLANCE

In many organizations ranging from large enterprises to schools, home and law enforcement agencies where security is becoming an essential concern and is turning the security centers to video analytics to keep their premises safe. Law enforcement agents can use a body-worn camera to identify criminals in real-time while transmitting the video stream to the video analytic cloud. IVAaaS can provide customized services that can adjust quickly to changing needs and demands. People detection, and tracking [353], motion detection, intrusion detection [354], line crossing [355], object lift, loitering [356], and license plate recognition [357] are the example of video analytics services for security. For security in subway stations, Krausz [358] developed as a surveillance system

to detect dangerous events. Shih *et al.* [359] tried to extract the color features of an employee's uniform to recognize the entry legality in a restricted area. In the context of security, abandoned object detection is indispensable and can lead to a terrorist attack. In the future, video analytics applications are developing fast, and they are changing the way the security industry works.

E. AUGMENTED REALITY AND PERSONAL DIGITAL ASSISTANCE

Among the many, one aspect of the augmented reality is visual, where devices like special glasses, helmets, or goggles are utilized for the projection of additional information or interactive experience of the surrounding real-world environment. The visual aspect of augmented reality may encompass complex IVA and demand powerful hardware. Likewise, vision-based digital assistants is a rising technology (e.g., personal robot Jibo) that could deeply alter our regular activities while offering personalized and interactive experiences. Such devices could be offloaded to the CVAS for low latency complex IVA uninterruptedly.

F. RETAIL, MANAGEMENT, AND BUSINESS INTELLIGENCE ANALYSIS

Large-scale products, services, and staff management while adjusting to consumer demands can be challenging without timely and up-to-date information. Smart Retail solution takes advantage of smart cameras combined with IVA to gather data on store operations and customer trends. Dwell analysis, face recognition, queue management, customer count, customer matrices, consumer traffic map, are some of the example services in this context [360]. Gaze analysis provides a means to learn customers' interest in merchandise by following their attention [361], [362] on a store display. Actions like reaching or grabbing products were analyzed by [363], [364] to understand customers' interest. Emotion analysis can identify customers' views regarding product and interaction with the company's representative [334]. IVA can also be used for business intelligence analysis to answer queries like "number of people visited per unite time?" or "customer interest in items?" while utilizing the same security infrastructure. Such information is beneficial for retailers in improving customer experience and marketing strategies.

VIII. RESEARCH ISSUES, OPPORTUNITIES, AND FUTURE DIRECTIONS

Intelligent video big data analytics in the cloud opens new research avenues, challenges, and opportunities. This section provides in-depth detail about such research challenges, which has been summarized in Table 11).

A. IVA ON VIDEO BIG DATA

Big data analytics engines are the general-purpose engine and are not mainly designed for big video analytics. Con-

TABLE 11. Open research issues in video big data analytics in the cloud.

Component	Aspect	Layer	Open Research Issues
Video Big Data	Volume	VBDC	Orchestration and Optimization of IVA Pipeline.
		VBDC, VBDC, VBDC	Big dimensionality reduction, and indexing.
	Velocity	VBDC, VBDC, VBDC	Cleaning and compressing video big data.
			Real-time video streams and online learning.
	Variety		Big dimensionality reduction.
			Data modality for single IVA goal.
	Veracity		Video big data veracity assessment.
		Learning with unreliable data.	
Value		Understandable IVA for decision support.	
User	Developer / Researcher	WSL	Semantic concepts extraction in distributed environment.
			Declarative IVA.
			IVA and distributed computing technologies abstraction.
		WSL	Comprehensive evaluation measures for IVA in the cloud environment.
		VBDC, WSL	Visualizing video big data.
		WSL, VBDC	IVA algorithm, model, and services statistics maintenance and ranking.
			Effective price scheme for IVA algorithm deployment, and subscription.
Consumer	WSL	Model management and algorithm selection.	
		IVA services utilization.	
		Improving consumer experience based on feedback.	
		VBDC, VBDC, VBDC, VBDC, VBDC	Effective price scheme for multiple IVA service subscription.
Security and Privacy	-		Privacy preserving distributed IVA, security, and trust.
Cloud System	Analytics engine	VBDC, VBDC	IVA on video big data (general big data middleware for IVA).
	Infrastructure		Parameter Server optimization.
			IVA on video big data (general big data middleware for IVA).

sequently, video big data analytics is challenging over such engines and demand optimization. Almost all the engines are inherently lacking the support of elementary video data structures and processing operations. Further, such engines are also not optimized, especially for iterative IVA and dependency among processes.

Optimizing cluster resource allocations among multiple workloads of iterative algorithms often involves an approximation of their runtime, i.e., predicting the number of iterations and the processing time of each iteration [365]. By default, Hadoop lacks iterative job support but can be handled through speculative execution. However, Spark supports not only MapReduce and fault tolerance but also cache data in memory between iterations. IVA on video big data creates an immense space for the research community to further crack in this direction. The research community is already trying to develop basic video processing and IVA support over big data, but it is still the beginning. How to optimize such engines for iterative IVA? It also allows us to research whether the existing distributed computing engines fulfill the demands of the IVA on video big data or need a specialized one.

Furthermore, the focus of the existing research on IVA are velocity, volume, velocity, but the veracity and value have been overlooked. One promising direction in addressing video big data veracity is to research methods and techniques capable of accessing the credibility of video data sources so that untrustworthy video data can be filtered. Another way is to come up with novel ML models that can make inferences with defective video data. Likewise, users' assistance is required to comprehend IVA results and the reason behind the decision to realize the value of video big data in decision support. Thus, understandable IVA can be a significant future research area.

B. IVA AND HUMAN-MACHINE COORDINATION

IVA on video big data grants a remarkable opportunity for learning with human-machine coordination for numerous reasons:

- IVA on video big data in cloud demands researchers and practitioners mastering both IVA and distributed computing technologies. Bridging both the worlds for most analysts is challenging. Especially in an educational environment, where the researcher focuses more on the understanding, configuration, and tons of parameters rather than innovation and research contribution. Thus there is a growing need to design such CVAS that provide high-level abstractions to hide the underlying complexity.
- IVS service to become commercially worthwhile and to achieve pervasive recognition, consumer lacking technical IVA knowledge. The consumers should be able to configure, subscribe, and maintain IVA services with comfort.
- In traditional IVA, consumers are usually passive. Further, research is required to build more interactive IVA services that assist consumers in gaining insight into video big data. An efficient interactive IVA service depends on the design of innovative interfacing practices based on an understanding of consumer abilities, behaviors, and requirements [366]. The interactive IVA services will learn from the consumer and decrease the need for administration by a specialist. It will also enable consumers to design custom IVA services to meet the domain-specific requirement.

C. ORCHESTRATION AND OPTIMIZATION OF IVA PIPELINE

L-CVAS is a service-oriented architecture, i.e., (R/B)IVAaaS. The real-time and batch workflow are deeply dependent on

the messaging middleware (Table 2) and distributed processing engines (Table 7). In L-CVAS, the dynamic (R/B)IVA service creation and multi-subscription environment demand the optimization and orchestration of the IVA service pipeline while guarantees opportunities for further research.

In literature, two types of scheduling techniques have been presented for real-time scheduling, i.e., static and dynamic [367], [368]. Static approaches are advantageous if the number of services and subscription sources is known priorly, but this is not the case with L-CVAS. The suitability of dynamic methods is reasonable but is expensive in terms of resource utilization. Likewise, the main issue in the BIVA service workflow on video big data is the data partitioning, scheduling, executing, and then integrate numerous predictions. The BIVA service workflow can be affected by the data flow feature of the underlying big data engine (as shown in Table 7). As Hadoop map-reduce lacking the loops or chain of stages, and Spark support DAG style of chaining, whereas Flink supports a Controlled Cyclic Dependency Graph (CCDG).

In the map-reduce infrastructure, a slowdown predictor can be utilized to improve the agility and timeliness of scheduling decisions [369]. Spark and Flink can accumulate a sequence of algorithms into a single pipeline but need research to examine its behavior in dynamic service creation and subscription environment. Further, concepts from the field of query and queuing optimization can be utilized while considering messaging middleware and distributed processing engines to orchestrate and optimization of IVA service Pipeline.

D. IVA AND BIG DIMENSIONALITY

The VSIDS multi-modality can produce diverse types of data streams. Similarly, the IVA algorithm, developer, and IR has a triangular relationship. An array of algorithms can be deployed, generating varied sorts of multi-dimensional features from the acquired data streams. The high-dimensionality factor poses many intrinsic challenges for data stream acquisition, transmission, learner, pattern recognition problems, indexing, and retrieval. In literature, it has been referred to as a “Big Dimensionality” challenge [370].

VSIDS variety leads to key challenges in acquiring and effectively processing the heterogeneous data. Most existing IVA approaches can consider a specific input, but in many cases, for a single IVA goal, different kinds, and formats can be considered.

With growing features dimensionality, current algorithms quickly become computationally inflexible and, therefore, inapplicable in many real-time applications [371]. Dimension reduction approaches are still going to be a hot research topic because of data diversity, increasing volume, and complexity. Effect-learning algorithms for first-order optimization, online learning, and paralleling computing will be more preferred.

Similarly, designing a generic, efficient, and scalable multi-level distributed data model for indexing and retrieving multi-dimensional features is becoming tougher than ever

because of the exponential growth and speed of video data. Considering varied situations, requirements, and parameters such as complex data type indexing (such as objects, which contains multiple types of data), multi-dimensional features (require different feature matching scheme for each type), cross scheme matching (e.g., spatiotemporal, spatial-object, object-temporal, etc.), giant search space, incremental updates, on the fly indexing, and concurrent query processing demands further investigation. It gives the research community the opportunities to optimize existing hashing schemes and indexing structures such as R-trees [372], M-tree [373], X-tree [374], locality-sensitive hashing [375] etc. on the top of big data engines. Log-structured merge-tree [376] based distributed data stores (see Table 4) can be leveraged to improve multi-dimensional query performances. Additionally, ML classification models can be used to capture the semantics by inspecting the association between features and the context among them to better index multi-dimensional data. Such methods make them more precise and effective than the non-ML methods [377]. Thus, ML classification models (including neural networks) can be used to capture the semantics by inspecting the association between features and the context among them to better index multi-dimensional data, that make them more precise and effective than the traditional indexing approaches [377].

E. ONLINE LEARNING ON VIDEO BIG DATA

The value of RIVA is dependent on the velocity of the video streams, i.e., newness and relatedness to ongoing happenings. Though existing big data de-facto standards are lacking to deal with the changing streams [378]. The RIVA services must address continuous and changing video streams. In this context, online learning can be utilized, representing a group of learning algorithms for constructing a predictive model incrementally from a sequence of data, e.g., Fourier Online Gradient Descent and Nystrom Online Gradient Descent algorithms [379]. In this context, Nallaperuma *et al.* [380] proposed ITS platform utilizing unsupervised online learning and deep learning approaches. It gives further research opportunities by involving data fusion from heterogeneous data sources [380].

F. MODEL MANAGEMENT

L-CVAS architecture is designed to deploy an array of IVA algorithms, i.e., both by administrator and developers. An algorithm might hold a list of parameters. The model selection process encompasses feature engineering (feature selection IV-B1), IVA algorithm selection, and hyperparameter tuning. Feature engineering is a laborious activity and is influenced by many key factors, e.g., domain-specific regulations, time, accuracy, video data, and IVA properties, which resultantly slow and hinder exploration. IVA algorithm selection is the process of choosing a model that fixes the hypothesis space of prediction function explored for a given application [381]. This process of IVA algorithm selection is reliant on technical and non-technical

aspects, which enforce the IVA developer to try manifold techniques at the cost of time and cloud resources. Hyperparameter is vital as they govern the trade-offs between accuracy and performance. IVA analysts usually do ad-hoc manual tuning by iteratively choosing a set of values or using heuristics such as grid search [381]. From IVA analysts' perspective, model selection is an expensive job in terms of time and resources that bringing down the video analytics lifecycle [382]. Model selection is an iterative and investigative process that generally creates an endless space, and it is challenging for IVA analysts to know a priori which combination will produce acceptable accuracy/insights. In this direction, theoretical design trade-offs are presented by Arun *et al.* [383], but further research is required that how to shape a unified framework that acts as a foundation for a novel class of IVA analytics while building the procedure of model selection easier and quicker.

G. PARAMETER SERVERS AND DISTRIBUTED LEARNING

Developing a model (such as Stochastic Gradient Descent) for video big data analytics in a distributed environment carries an intrinsic issue of sharing and updating high-dimension parameters that can easily run into orders billions to trillions. The Parameter Server notion has been introduced to address this issue, aiming to store the parameters of an ML model such as the weights of a neural network and serve them to clients. Parameter Server proposes a new framework for building distributed ML algorithms, and encompass diverse design goals, e.g., efficient communication, flexible consistency, elasticity when adding resources, resource utilization, and ease of use. In literature, recently, various studies tried to optimize Parameter Server. PS2 [384] builds the parameter server on top of Spark. SketchML [385] compresses the gradient values by a sketch-based method. FlexPS [386] introduces a multi-stage abstraction to support flexible parallel control. The Parameter Server can be optimized further against the stated design goals and need further investigation.

H. EVALUATION ISSUES AND OPPORTUNITIES

Haralick [387] initiated the discussion of IVA performance evaluation followed by dedicated workshops [388], and journals [389], [390]. As a result, performance evaluation tools (ViPER5), and datasets (ETISEO [391], TrecVID [392], iLIDS [393]) were introduced. It is a fact that traditional IVA has an established set of prediction accuracy based metrics for performance evaluation that is ranging from accuracy, error rate, and precision to optimization and estimation error. Some more evaluation parameters are adopted from big data analytics when IVA is tried on distributed computing, e.g., scalability, fault tolerance, memory usage, throughput, etc. [394] (as shown in Table 9. The amalgamation of two types of matrices might not be enough.

IVA services provided by a system like L-CVAS has to accomplish predictably through an intractable number of scenarios and environmental circumstances, meeting requirements that vary according to the situation, domain, and user.

For L-CVAS's consumer, the IVA services work as a black box where the significant metrics relate to overall system performance, such as false alarms, accuracy, and detection rate. However, from the developer and researcher perspective, L-CVAS consists of numerous computer vision algorithms, with complex interfaces among them. A proper performance evaluation matrix is required for IVA service developers to comprehend these relations and to revolutionize and address novel IVA services. Another critical issue is how to guarantee accurate and predictable IVA service performance when porting technology between distributed algorithm development environments and deployment code environments with hardware-specific optimizations. These shortcomings result in algorithmic alterations that can influence the performance of IVA services.

IVA performance evaluation is goal-oriented, and the factors should be determined carefully. Many key factors that influence the performance of video big data analytics utilizing distributed computing engines in the cloud are listed below (not limited to).

- VSDS: holds diverse types of parameters, i.e., video type (color, grey-scale, infrared, omnidirectional, depth map, etc.), property (frame-rate, field-depth), and quality (resolution, pixel depth) of the generated video as generated by the camera.
- VSDS and messaging middleware parameters: encompass VSDS connection, frames reducing and transformation, messaging queue (broker server), compression artifacts, mini-batch size, and possibly the involvement of internal and external network.
- VSDS environmental parameters: some features of the configuration remain constant in a given use case, but differ between configuration, possibly influencing performance. These parameters comprise camera location (mounting height, angle, indoor or outdoor), mounting type (still or in motion), camera view (roads, water, foliage), and weather (sun, cloud, rain, snow, fog, wind).
- Distributed processing environment: the video processing hardware (FPGA, CPU, GPU, etc.), and network communication channels potentially impose additional limitations in terms of locality, speed, and memory.
- Big data analytics engines: The nature and characteristics of big data engines affect the performance of the IVA, such as data flow, windowing, computation model, etc. Further, big video analytics in the cloud demand complex trade-offs between different evaluation criteria. In order to comprehend, assume the intricate trade-offs between accuracy and response time. Iterative tasks have an inverse relation with fault tolerance concerning scalability (e.g., MapReduce is high fault-tolerant but lacking iteration). Similarly, non-iterative IVA algorithms scale better than iterative at the cost of performance degradation.
- IVA service parameters: Application parameters are domain-specific parameters, e.g., vehicles, carts,

humans, etc., tolerable miss detection and false alarm rates and their desired trade-off, IVA type, and max acceptable latency.

- Computation and communication trade-off: IVA Algorithms and services in the distributed environment should be developed and designed wisely intending to minimize computation time, which is associated with data locality and loading.

Diverse types of factors affect video big data analytics performance in the cloud, constructing a comprehensive evaluation of all use cases are almost near impossible. It further provides opportunities for the researchers to design a framework that provides a unified and generic framework that can be adapted by any CVAS. Investigating these issues would significantly contribute to the academic and industrial communities interested in building IVA algorithms, services, and CVAS.

I. IVA ALGORITHM, MODEL, AND SERVICES STATISTICS MAINTENANCE, RANKING, AND RECOMMENDATION

L-CVAS architecture is designed under the Customer-to-Customer (C2C) business model. In L-CVAS, a user can develop and deploy an IVA algorithm, model, or service (here collectively we call it IVA service) that can be either extended, utilized, or subscribed by other users. The community members run such architecture, and rapidly, the number of IVA services can be reached to tons of domain-dependent or independent IVA services. This scenario develops a complex situation for the users, i.e., which IVA service (when sharing the parallel functionalities) in a specific situation, especially during service discover. Against each IVA service, there is a list of Quality of Service (QoS) parameters. Some of these QoS parameters (not limited to) are user trust, satisfaction, domain relevance, security, usability, availability, reliability, documentation, latency, response time, resource utilization, accuracy, and precision.

Such types of IVA services against the QoS parameters lead to the 0-1 knapsack issue. In this direction, one possible solution is utilizing multi-criteria decision-making approaches. It gives further opportunities to the research community to investigate how to rank and recommend IVA algorithms, models, and services. Similarly, it can lead to a high-dimensional sparse matrix [395]. In this direction, research is required on how to utilize such parameters for IVA services recommendation.

J. IVAaaS AND COST MODEL

Recently, cloud-based analytics platforms are the key means for enterprises to provide services on the pay-as-you-go cost model. Existing cost metrics usually are determined to utilize hardware usage comprising processing (CPU, GPU), disk space, and memory usage. These prices are often static or dynamic [396]. The example of the former one is Amazon's EC2, which offers tiered levels of service. In later cases, the cost model is used to determine the price of the service

using analytics. This takes into account factors such as peak hours and opponent cost model etc. The hardware cost is usually minimal compared to the cost of software such as L-CVAS where the cost of IVA analytics valued more.

L-CVAS is supposed to provide IVA-Algorithm-as-a-Service (IVAAaaS) and IVAaaS in the cloud while adopting the C2C business model. Unfortunately, current SaaS cost models might not be applicable because of the involvement of diverse types of parameters that drastically affect the cost model. Such parameters are, business model (Business-to-Business (B2B), Business-to-Customer (B2C), and C2C), unite of video, user type (developer, researcher, and consumers), services (IVAAaaS and IVAaaS), service subscription (algorithm, IVA service, single, multiple, dependent or independent), cloud resource utilization, user satisfaction, QoS, location, service subscription duration, and cost model fairness. The addition of further parameters is subject to discussion, but the listed are the basic that govern L-CVAS cost matrix. Additionally, the cost model demands further research and investigations to develop an effective price scheme for IVA services while considering the stated parameters.

K. VIDEO BIG DATA MANAGEMENT

Despite video big data pose high value, but its management, indexing, retrieval, and mining are challenging because of its volume, velocity, and unstructuredness. IVA has been investigated over the years (section V) but still evolving and need to address diverse types of issues such as:

- In the context of video big data management, the main issue is the extraction of semantic concepts from primitive features. A general domain-independent framework is required that can extract semantic features, analyze and model the multiple semantics from the videos by using the primitive features. Further, semantic event detection is still an open research issue because of the semantic gap and the difficulty of modeling temporal and multi-modality features of video streams. The temporal information is significant in the video big data mining mainly, in pattern recognition.
- Motion analysis is vital, and further research is required for moving objects analysis, i.e., object tracking, handling occlusion, and moving objects with statics cameras [213], moving cameras [215], and multiple camera fusion.
- Limited research is available on CBVR (see section V-A) while exploiting distributed computing. Further study is required to consider different features ranging from local to global spatiotemporal features utilizing and optimizing deep learning and distributed computing engines.
- For video retrieval, semantic-based approaches have been utilized because of the semantic gap between the low-level features and high-level human-understandable concepts. Ontology adds extra concepts that can improve the retrieval results [182] but can also lead to unexpected

deterioration of search results. In this context, a hybrid approach can be fruitful and need to design different query planes that can fulfill diverse queries in complex situations.

- Insufficient research is available on graph-based video big data retrieval and analysis, opening doors for further investigation. The researchers can conduct studies to answer questions like: the formation of the video graph, tuning the similar value and its effect on the graph formation, studying properties, and meta-analytic of the formed video graph.
- How the reinforcement learning and real-time feedback query expansion technique [397] can be exploited to improve the retrieval results?
- Recently, video query engines have been introduced to retrieve and analyze video at scale [398]–[400]. A special focus of the database community is required to design, implement, optimize, and operationalize such video query engines.

L. PRIVACY, SECURITY AND TRUST

Video big data, acquisition, storage, and subscriptions to shared IVA in the cloud become mandatory, which leads to privacy concerns. For the success of such platforms, privacy, security, and trust are always central. In literature, the word ‘trust’ is commonly used as a general term for ‘security’ and ‘privacy’ [401]. Trust is a social phenomenon where the user has expectations from the IVA service provider and willing to take action (subscription) on the belief based on evidence that the expected behavior occurs [402], [403]. In the cloud environment, security and privacy are playing an active role in the trust-building. To ensure security, the CVAS should offer different levels of privacy control. The phenomena of privacy and security are valid across VSDS, storage security, multi-level access controls, and privacy-aware IVA and analysis. We list some research directions that can provide opportunities for cloud security specialists.

- The video big data volume, variety, and velocity boost security threats. Recently, disputes and news were circulating regarding the misuse of user-generated content and hacking the cameras in an unauthorized manner. The CVAS vendor, in collaboration with or under the law enforcement agencies, must come up with new rules, laws, and agreements, which can differ from country to country. Utilizing such policies, the CVAS vendor should ensure that all the IVA service, subscription, and storage level agreements are adequately followed. The policies can be researched whether it offers adequate protection for individuals’ data while performing video big data analytics and public monitoring.
- Unlike other data, videos are more valuable for the owner and can be a direct threat, e.g., live broadcast, blackmailing, etc. Likewise, the focus of traditional privacy approaches is data management that becomes absolute when it comes to data security. Novel algo-

gorithms are required to secure user’s data both for shared IVA, storage to make the video stream acquisition more secure.

- In the context of security, the blockchain (popularized by Bitcoin) [404] has been studied and operationalized across academia and industry evenly. A blockchain is a modification resilient cryptography technique known as a distributed ledger, where records are linked and managed by a decentralized peer-to-peer network [404]. The blockchains techniques are still in early-stage and can be researched further to form a novel automated security system for CVAS.
- ML techniques have been matured over the years and have been successfully utilized for security, i.e., modeling attack patterns with their distinctive features. However, change in features in case of sophisticated attacks may lead to security failure. ML could enhance the performance of security solutions to alleviate the dangers of the existing cyberattacks. The research community can further investigate how ML techniques, especially deep learning, can be deployed to analyze logs produced by network traffic, IVA processes, and users to recognize doubtful activities.

IX. CONCLUSION

In the recent past, the number of public surveillance cameras has increased significantly, and an enormous amount of visual data is produced at an alarming rate. Such large-scale video data pose the characteristics of big data. Video big data offer opportunities to the video surveillance industry and permits them to gain insights in almost real-time. The deployment of big data technologies such as Hadoop, Spark, etc., in the cloud under aaS paradigm to acquire, persist, process and analyze a large amount of data has been in service from last few years. This approach has changed the context of information technology and has turned the on-demand service model’s assurances into reality.

This paper provides an extensive study on intelligent video big data in the cloud. First, we define basic terminologies and establish the relation between video big data analytics and cloud computing. A comprehensive layered architecture has been proposed for intelligent video big data analytics in the cloud under the aaS model called L-CVAS. VBDCL is the base layer that allows the other layers to develop IVA algorithms and services. This layer is based on the concept of IR orchestration and takes care of data curations throughout the life cycle of an IVA service. The VBDPL is in-charge of pre-processing and extracting the significant features from the raw videos. The VBDML is accountable for producing the high-level semantic result from the features generated by the VBDML. The KCL deploys video ontology and creates knowledge based on the extracted higher-level features obtained from VBDML. When all these layers are pipelined in a specific context, it becomes an IVA service to which the users can subscribe to video data sources under the IVAaaS paradigm.

Furthermore, to show the significance and recent research trends of IVA in the cloud, a broad literature review has been conducted. The research issues, opportunity, and challenges being raised by the uniqueness of the proposed L-CVAS, and the triangular relation among video big data analytics, distributed computing technologies, and cloud has been reported.

REFERENCES

- [1] M INC. (2012). *Video Analytics Market*. Accessed: Dec. 20, 2019. [Online]. Available: <https://github.com/apache/rocketmq>
- [2] Facebook.Com. (2020). *Facebook*. Accessed: Dec. 20, 2019. [Online]. Available: <https://www.facebook.com>
- [3] Youtube.com. (2019). *Youtube Statistics*. Accessed: Dec. 20, 2019. [Online]. Available: <https://www.youtube.com/about/press/>
- [4] Netflix.Com. (2020). *Netflix*. Accessed: Dec. 20, 2019. [Online]. Available: <https://www.netflix.com/>
- [5] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar, "Multimedia big data analytics: A survey," *ACM Comput. Surv.*, vol. 51, no. 1, p. 10, Jan. 2018, doi: [10.1145/3150226](https://doi.org/10.1145/3150226).
- [6] I. E. Olatunji and C.-H. Cheng, "Dynamic threshold for resource tracking in observed scenes," in *Proc. 9th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, 2018, pp. 1–6.
- [7] IDC Corporation. (2019). *The Growth in Connected IoT Devices*. Accessed: Dec. 7, 2019. [Online]. Available: <https://www.idc.com/getdoc.jsp?containerId=prUS45213219>
- [8] T. Huang, "Surveillance video: The biggest big data," *Comput. Now*, vol. 7, no. 2, pp. 82–91, 2014.
- [9] P. Mell and T. Grance, "The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-145, 2011.
- [10] P. Zikopoulos, D. Deroos, K. Parasuraman, T. Deutsch, J. Giles, and D. Corrigan, *Harness the Power of Big Data The IBM Big Data Platform*. New York, NY, USA: McGraw-Hill, 2012.
- [11] EWS Amazon. (Nov. 2012). *Amazon Web Services*. [Online]. Available: <http://aws.amazon.com/es/ec2/>
- [12] M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, "Overview of Microsoft azure services," in *Microsoft Azure*. Berkeley, CA, USA: Springer, 2015, pp. 27–69.
- [13] J.-P. Dijcks, "Oracle: Big data for the enterprise," Oracle, Redwood City, CA, USA, Oracle White Paper, 2012, p. 16. [Online]. Available: <http://www.oracle.com/us/products/database/big-data-for-enterprise-519135.pdf>
- [14] J. R. Mashey, "Big data and the next wave of infras-tress," in *Computer Science Division Seminar*. Berkeley, CA, USA: Univ. of California, 1997.
- [15] C. Snijders, U. Matzat, and U.-D. Reips, "'Big data': Big gaps of knowledge in the field of Internet science," *Int. J. Internet Sci.*, vol. 7, no. 1, pp. 1–5, 2012.
- [16] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, Jan. 2015.
- [17] N. Elgandy and A. Elragal, "Big data analytics: A literature review paper," in *Proc. Ind. Conf. Data Mining*. Cham, Switzerland: Springer, 2014, pp. 214–227.
- [18] H. Özköse, E. S. Ari, and C. Gencer, "Yesterday, today and tomorrow of big data," *Procedia-Social Behav. Sci.*, vol. 195, pp. 1042–1050, Jul. 2015.
- [19] S. O. Fadiya, S. Saydam, and V. V. Zira, "Advancing big data for humanitarian needs," *Procedia Eng.*, vol. 78, pp. 88–95, Jan. 2014.
- [20] B. Marr, *Big Data: Using SMART Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance*. Hoboken, NJ, USA: Wiley, 2015.
- [21] D. E. O'Leary, "Artificial intelligence and big data," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 96–99, Mar./Apr. 2013.
- [22] J. J. Berman, *Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information*. London, U.K.: Newnes, 2013.
- [23] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt, 2013.
- [24] B. Marr. (2015). *Why Only One of the 5 vs of Big Data Really Matters*. [Online]. Available: <http://www.ibmbigdatahub.com/blog/why-only-one-5-vs-big-data-really-matters>
- [25] G. F. Pfister, *Search Clusters*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.
- [26] R. Buyya, *High Performance Cluster Computing: Architectures and Systems*, vol. 1. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
- [27] H. Brian, "Cloud computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, 2008.
- [28] G. Aceto, A. Botta, W. de Donato, and A. Pescapè, "Cloud monitoring: A survey," *Comput. Netw.*, vol. 57, no. 9, pp. 2093–2115, Jun. 2013.
- [29] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. New York, NY, USA: Manning, 2015.
- [30] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: A survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1222–1233, Aug. 2013.
- [31] I. E. Olatunji and C.-H. Cheng, "Video analytics for visual surveillance and applications: An overview and survey," in *Machine Learning Paradigms*. Cham, Switzerland: Springer, 2019, pp. 475–515.
- [32] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 797–819, Nov. 2011.
- [33] B. V. Patel and B. B. Meshram, "Content based video retrieval systems," 2012, *arXiv:1205.1641*. [Online]. Available: <http://arxiv.org/abs/1205.1641>
- [34] M. Haseyama, T. Ogawa, and N. Yagi, "[Survey paper] a review of video retrieval based on image and video semantic understanding," *ITE Trans. Media Technol. Appl.*, vol. 1, no. 1, pp. 2–9, 2013.
- [35] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and networked vehicle surveillance in ITS: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 557–580, Apr. 2015.
- [36] A. Ben Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Syst. Appl.*, vol. 91, pp. 480–491, Jan. 2018.
- [37] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, M. Ali, W. Kamaleldin, M. Alam, M. Shiraz, and A. Gani, "Big data: Survey, technologies, opportunities, and challenges," *Sci. World J.*, vol. 2014, Jul. 2014, Art. no. 712826.
- [38] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," *J. Big Data*, vol. 2, no. 1, p. 21, 2015.
- [39] D. Agrawal, S. Das, and A. El Abbadi, "Big data and cloud computing: Current state and future opportunities," in *Proc. 14th Int. Conf. Extending Database Technol.*, 2011, pp. 530–533.
- [40] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, May 2017.
- [41] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: Challenges, issues, and opportunities," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Berlin, Germany: Springer, 2013, pp. 1–15.
- [42] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
- [43] H. Zahid, T. Mahmood, A. Morshed, and T. Sellis, "Big data analytics in telecommunications: Literature review and architecture recommendations," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 1, pp. 18–38, Jan. 2020.
- [44] D. Elliott, "Intelligent video solution: A definition," *Security*, vol. 47, no. 6, 2010. [Online]. Available: <https://www.securitymagazine.com/articles/80912-intelligent-video-solution-a-definition>
- [45] Y.-L. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, and C.-F. Shu, "IBM smart surveillance system (S3): Event based video surveillance system with an open and extensible framework," *Mach. Vis. Appl.*, vol. 19, nos. 5–6, pp. 315–327, 2008.
- [46] M. A. Sujana, "Zoneminder video surveillance by linux platform," Ph.D. dissertation, Universiti Teknologi MARA, Shah Alam, Malaysia, 2004.
- [47] A. Alam and Y.-K. Lee, "TORNADO: Intermediate results orchestration based service-oriented data curation framework for intelligent video big data analytics in the cloud," *Sensors*, vol. 20, no. 12, p. 3581, Jun. 2020.
- [48] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing," in *Proc. NetDB*, 2011, pp. 1–7.
- [49] A. Videla and J. J. Williams, *RabbitMQ in Action: Distributed Messaging for Everyone*. Greenwich, CT, USA: Manning, 2012.
- [50] B. Snyder, D. Bosnanac, and R. Davies, *ActiveMQ in Action*, vol. 47. Greenwich, CT, USA: Manning, 2011.

- [51] W. Zhang, L. Xu, P. Duan, W. Gong, Q. Lu, and S. Yang, "A video cloud platform combining online and offline cloud computing technologies," *Pers. Ubiquitous Comput.*, vol. 19, no. 7, pp. 1099–1110, Oct. 2015.
- [52] G. Banavar, T. Chandra, R. Strom, and D. Sturman, "A case for message oriented middleware," in *Distributed Computing*, P. Jayanti, Ed. Berlin, Germany: Springer, 1999, pp. 1–17.
- [53] A. Ejsmont, *Web Scalability for Startup Engineers*. New York, NY, USA: McGraw-Hill, 2015.
- [54] A. Ejsmont, *Web Scalability for Startup Engineers*, 1st ed. New York, NY, USA: McGraw-Hill, 2015.
- [55] E. Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*. New Delhi, India: Pearson, 1995.
- [56] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proc. 7th Symp. Operating Syst. Design Implement. (OSDI)*, Berkeley, CA, USA: USENIX Association, 2006, pp. 307–320. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1298455.1298485>
- [57] E. B. Boyer, M. C. Broomfield, and T. A. Perrotti, "Glusterfs one storage server to rule them all," Los Alamos Nat. Lab., Los Alamos, NM, USA, Rep. LA-UR-12-23586, 2012.
- [58] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *Proc. IEEE 26th Symp. Mass Storage Syst. Technol. (MSST)*, May 2010, pp. 1–10.
- [59] P. J. Braam and R. Zahir, "Lustre: A scalable, high performance file system. cluster file systems," Technic Inc., Cranston, RI, USA, White Paper, 2002. [Online]. Available: <https://cse.buffalo.edu/faculty/tkosar/cse710/papers/lustre-whitepaper.pdf>
- [60] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié, "MOOSE: A parallel computational framework for coupled systems of nonlinear equations," *Nucl. Eng. Des.*, vol. 239, no. 10, pp. 1768–1778, Oct. 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029549309002635>
- [61] M. Ovsianikov, S. Rus, D. Reeves, P. Sutter, S. Rao, and J. Kelly, "The quantcast file system," *Proc. VLDB Endowment*, vol. 6, no. 11, pp. 1092–1101, Aug. 2013, doi: [10.14478/2536222.2536234](https://doi.org/10.14478/2536222.2536234).
- [62] D. G. Chandra, "BASE analysis of NoSQL database," *Future Gener. Comput. Syst.*, vol. 52, pp. 13–21, Nov. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X15001788>
- [63] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL database: New era of databases for big data analytics—classification, characteristics and comparison," 2013, *arXiv:1307.0191*. [Online]. Available: <http://arxiv.org/abs/1307.0191>
- [64] R. Cattell, "Scalable sql and nosql data stores," *SIGMOD Rec.*, vol. 39, no. 4, pp. 12–27, May 2011. [Online]. Available: <http://doi.acm.org/10.1145/1978915.1978919>
- [65] E. Brewer, "Pushing the cap: Strategies for consistency and availability," *Computer*, vol. 45, no. 2, pp. 23–29, Feb. 2012, doi: [10.1109/MC.2012.37](https://doi.org/10.1109/MC.2012.37).
- [66] E. Brewer, "CAP twelve years later: How the 'rules' have changed," *Computer*, vol. 45, no. 2, pp. 23–29, Feb. 2012.
- [67] J. L. Carlson, *Redis in Action*. Greenwich, CT, USA: Manning, 2013.
- [68] K. Banker, *MongoDB in Action*. Greenwich, CT, USA: Manning, 2011.
- [69] M. N. Vora, "Hadoop-HBase for large-scale data," in *Proc. Int. Conf. Comput. Sci. Netw. Technol.*, Dec. 2011, pp. 601–605.
- [70] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 4:1–4:26, Jun. 2008, doi: [10.1145/1365815.1365816](https://doi.org/10.1145/1365815.1365816).
- [71] M. Stonebraker and A. Weisberg, "The VoltDB main memory DBMS," *IEEE Data Eng. Bull.*, vol. 36, no. 2, pp. 21–27, Jun. 2013.
- [72] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proc. 19th ACM Symp. Operating Syst. Princ. (SOSP)*, New York, NY, USA, 2003, pp. 29–43, doi: [10.1145/945445.945450](https://doi.org/10.1145/945445.945450).
- [73] L. Xie, H. Sundaram, and M. Campbell, "Event mining in multimedia streams," *Proc. IEEE*, vol. 96, no. 4, pp. 623–647, Apr. 2008.
- [74] M.-L. Shyu, Z. Xie, M. Chen, and S.-C. Chen, "Video semantic event/concept detection using a subspace-based multimedia data mining framework," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 252–259, Feb. 2008.
- [75] C. Regazzoni, A. Cavallaro, Y. Wu, J. Konrad, and A. Hampapur, "Video analytics for surveillance: Theory and practice [From the guest editors]," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 16–17, Sep. 2010.
- [76] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," 2013, *arXiv:1309.0238*. [Online]. Available: <http://arxiv.org/abs/1309.0238>
- [77] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Process.*, vol. 5, no. 6, pp. 996–1011, Jun. 1996.
- [78] C. Saravanan, "Color image to grayscale image conversion," in *Proc. 2nd Int. Conf. Comput. Eng. Appl.*, vol. 2, 2010, pp. 196–199.
- [79] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," *J. Electron. Imag.*, vol. 5, no. 2, pp. 122–129, 1996.
- [80] A. Hampapur, T. Weymouth, and R. Jain, "Digital video segmentation," in *Proc. 2nd ACM Int. Conf. Multimedia*, 1994, pp. 357–364.
- [81] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 18–29, Mar. 2003.
- [82] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 1, Oct. 1998, pp. 866–870.
- [83] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [84] F. Baumann, A. Ehlers, B. Rosenhahn, and J. Liao, "Computation strategies for volume local binary patterns applied to action recognition," in *Proc. 11th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2014, pp. 68–73.
- [85] F. Baumann, A. Ehlers, B. Rosenhahn, and J. Liao, "Recognizing human actions using novel space-time volume binary patterns," *Neurocomputing*, vol. 173, pp. 54–63, Jan. 2016.
- [86] A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock, J. R. Smith, B. Tseng, Y. Wu, and D. Zhang, "IBM research TRECVID-2003 video retrieval system," in *Proc. NIST TRECVID*, 2003, vol. 7, no. 8, p. 36.
- [87] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel, "FXPAL experiments for TRECVID 2004," in *Proc. TREC Video Retr. Eval. (TRECVID)*, 2004, pp. 70–81.
- [88] R. Yan and A. G. Hauptmann, "A review of text and image retrieval approaches for broadcast news video," *Inf. Retr.*, vol. 10, nos. 4–5, pp. 445–484, Sep. 2007.
- [89] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," in *Proc. Int. Conf. Image Video Retr.* Berlin, Germany: Springer, 2005, pp. 226–236.
- [90] R. Visser, N. Sebe, and E. Bakker, "Object recognition for video retrieval," in *Proc. Int. Conf. Image Video Retr.* Berlin, Germany: Springer, 2002, pp. 262–270.
- [91] R. Mattivi and L. Shao, "Human action recognition using LBP-top as sparse spatio-temporal feature descriptor," in *Proc. Int. Conf. Comput. Anal. Images Patterns*. Berlin, Germany: Springer, 2009, pp. 740–747.
- [92] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, Jun. 2007.
- [93] L. Nanni, S. Brahnam, and A. Lumini, "Local ternary patterns from three orthogonal planes for human action classification," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5125–5128, May 2011.
- [94] Y. Yi, Z. Zheng, and M. Lin, "Realistic action recognition with salient foreground trajectories," *Expert Syst. Appl.*, vol. 75, pp. 44–55, Jun. 2017.
- [95] M. Uddin and Y.-K. Lee, "Feature fusion of deep spatial features and handcrafted spatiotemporal features for human action recognition," *Sensors*, vol. 19, no. 7, p. 1599, Apr. 2019.
- [96] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
- [97] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
- [98] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4305–4314.

- [99] X. Lu, H. Yao, S. Zhao, X. Sun, and S. Zhang, "Action recognition with multi-scale trajectory-pooled 3D convolutional descriptors," *Multimedia Tools Appl.*, vol. 78, no. 1, pp. 507–523, Jan. 2019.
- [100] G. Yao, T. Lei, J. Zhong, and P. Jiang, "Learning multi-temporal-scale deep information for action recognition," *Int. J. Speech Technol.*, vol. 49, no. 6, pp. 2017–2029, Jun. 2019.
- [101] L. Wang, J. Zang, Q. Zhang, Z. Niu, G. Hua, and N. Zheng, "Action recognition by an attention-aware temporal weighted convolutional neural network," *Sensors*, vol. 18, no. 7, p. 1979, Jun. 2018.
- [102] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Action-VLAD: Learning spatio-temporal aggregation for action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 971–980.
- [103] S. Zhao, Y. Liu, Y. Han, R. Hong, Q. Hu, and Q. Tian, "Pooling the convolutional layers in deep convnets for video action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1839–1849, Aug. 2018.
- [104] S. C. Hoi, J. Wang, P. Zhao, and R. Jin, "Online feature selection for mining big data," in *Proc. 1st Int. Workshop Big Data, Streams Heterogeneous Source Mining Algorithms, Syst., Program. Models Appl.*, 2012, pp. 93–100.
- [105] H. Tan and L. Chen, "An approach for fast and parallel video processing on apache Hadoop clusters," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2014, pp. 1–6.
- [106] Y. Cong, S. Wang, B. Fan, Y. Yang, and H. Yu, "UDSFS: Unsupervised deep sparse feature selection," *Neurocomputing*, vol. 196, pp. 150–158, Jul. 2016.
- [107] O. Dekel, "From online to batch learning with cutoff-averaging," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 377–384.
- [108] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: A SIFT descriptor with color invariant characteristics," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, 2006, pp. 1978–1983.
- [109] Y. Fang and Z. Wang, "Improving LBP features for gender classification," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, vol. 1, Aug. 2008, pp. 373–377.
- [110] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognit. Lett.*, vol. 32, no. 12, pp. 1598–1603, 2011.
- [111] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 71–79.
- [112] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [113] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, "Large scale distributed deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1223–1231.
- [114] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. Mostofa Ali Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," 2017, *arXiv:1712.00409*. [Online]. Available: <http://arxiv.org/abs/1712.00409>
- [115] H. Zhang, Z. Zheng, S. Xu, W. Dai, Q. Ho, X. Liang, Z. Hu, J. Wei, P. Xie, and E. P. Xing, "Poseidon: An efficient communication architecture for distributed deep learning on GPU clusters," in *Proc. Annu. Tech. Conf. (USENIX ATC)*, 2017, pp. 181–193.
- [116] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4694–4702.
- [117] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [118] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [119] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [120] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. [Online]. Available: <http://arxiv.org/abs/1409.4842>
- [121] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [122] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [123] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
- [124] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
- [125] N. Ketkar, "Introduction to pytorch," in *Deep Learning With Python*. Berkeley, CA, USA: Springer, 2017, pp. 195–208.
- [126] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems," 2015, *arXiv:1512.01274*. [Online]. Available: <http://arxiv.org/abs/1512.01274>
- [127] F. Seide and A. Agarwal, "Cntk: Microsoft's open-source deep-learning toolkit," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, p. 2135.
- [128] M. Mathwork, *Deep Learning Toolbox*. Natick, MA, USA: Mathworks.
- [129] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.
- [130] R. Mayer and H.-A. Jacobsen, "Scalable deep learning on distributed infrastructures: Challenges, techniques and tools," 2019, *arXiv:1903.11314*. [Online]. Available: <http://arxiv.org/abs/1903.11314>
- [131] A. Deshpande and M. Kumar, *Artificial Intelligence for Big Data: Complete Guide to Automating Big Data Solutions Using Artificial Intelligence Techniques*. Birmingham, U.K.: Packt, 2018.
- [132] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 103–112.
- [133] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
- [134] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, G. Gonzalez, S. Shenker, and I. Stoica, "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.
- [135] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache flink: Stream and batch processing in a single engine," *Bull. IEEE Comput. Soc. Tech. Committee Data Eng.*, vol. 36, no. 4, pp. 28–38, 2015.
- [136] A. Toshniwal, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, D. Ryaboy, "Storm Twitter," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 147–156.
- [137] G. Wang, J. Koshy, S. Subramanian, K. Paramasivam, M. Zadeh, N. Narkhede, J. Rao, J. Kreps, and J. Stein, "Building a replicated logging system with apache kafka," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 1654–1655, Aug. 2015.
- [138] S. Owen and S. Owen, *Mahout in Action*. Stamford, CT, USA: Manning Publications Co., 2012.
- [139] D. Team, "Deeplearning4j: Open-source distributed deep learning for the JVM," *Apache Softw. Found. License*, vol. 2, 2016. [Online]. Available: <https://deeplearning4j.org/>
- [140] F. Chollet, "Keras," 2015.
- [141] J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang, Y. Wang, X. Jia, C. Zhang, Y. Wan, Z. Li, J. Wang, S. Huang, Z. Wu, Y. Wang, Y. Yang, B. She, D. Shi, Q. Lu, K. Huang, and G. Song, "BigDL: A distributed deep learning framework for big data," 2018, *arXiv:1804.05839*. [Online]. Available: <http://arxiv.org/abs/1804.05839>
- [142] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "MLlib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235–1241, 2016.

- [143] G. De Francisci Morales and A. Bifet, "SAMOA: Scalable advanced massive online analysis," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 149–153, 2015.
- [144] J. Dean and S. Ghemawat, "MapReduce: A flexible data processing tool," *Commun. ACM*, vol. 53, no. 1, pp. 72–77, Jan. 2010.
- [145] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," *HotCloud*, vol. 10, nos. 10–10, p. 95, 2010.
- [146] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, "Discretized streams: Fault-tolerant streaming computation at scale," in *Proc. 24th ACM Symp. Operating Syst. Princ.*, 2013, pp. 423–438.
- [147] D. Cook, *Practical Machine Learning With H2O: Powerful, Scalable Techniques for Deep Learning and AI*. Newton, MA, USA: O'Reilly Media, 2016.
- [148] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [149] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "YFCC100M: The new data in multimedia research," 2015, *arXiv:1503.01817*. [Online]. Available: <http://arxiv.org/abs/1503.01817>
- [150] Trecvid. *TREC Video Retrieval Evaluation: TRECVID*. Accessed: Dec. 7, 2019. [Online]. Available: <https://trecvid.nist.gov/>
- [151] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: <http://arxiv.org/abs/1212.0402>
- [152] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*. [Online]. Available: <http://arxiv.org/abs/1705.06950>
- [153] CEUR. (2015). *Multimedia Benchmark Workshop*. Accessed: Jul. 12, 2019. [Online]. Available: <http://ceur-ws.org/Vol-1436/>
- [154] A. Nambiar, M. Taiana, D. Figueira, J. C. Nascimento, and A. Bernardino, "A multi-camera video dataset for research on high-definition surveillance," *Int. J. Mach. Intell. Sensory Signal Process.*, vol. 1, no. 3, pp. 267–286, 2014.
- [155] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 961–970.
- [156] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.
- [157] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, "AVA: A video dataset of spatio-temporally localized atomic visual actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6047–6056.
- [158] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips," 2019, *arXiv:1906.03327*. [Online]. Available: <http://arxiv.org/abs/1906.03327>
- [159] S. Abu-El-Hajja, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark," 2016, *arXiv:1609.08675*. [Online]. Available: <http://arxiv.org/abs/1609.08675>
- [160] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra, "Video annotation and retrieval using ontologies and rule learning," *IEEE Multimedia Mag.*, vol. 17, no. 4, pp. 80–88, Oct. 2010.
- [161] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of TRECVID activity," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 411–418, Apr. 2010.
- [162] A. F. Smeaton, P. Over, and W. Kraaij, "High-level feature detection from video in TRECVID: A 5-year retrospective of achievements," in *Multimedia Content Analysis*. Boston, MA, USA: Springer, 2009, pp. 1–24.
- [163] H. Bhaumik, S. Bhattacharyya, M. D. Nath, and S. Chakraborty, "Hybrid soft computing approaches to content based video retrieval: A brief review," *Appl. Soft Comput.*, vol. 46, pp. 1008–1029, Sep. 2016.
- [164] S.-M. Hu, T. Chen, K. Xu, M.-M. Cheng, and R. R. Martin, "Internet visual media processing: A survey with graphics and vision applications," *Vis. Comput.*, vol. 29, no. 5, pp. 393–405, May 2013.
- [165] N. Spolaor, H. D. Lee, W. S. R. Takaki, L. A. Ensina, C. S. R. Coy, and F. C. Wu, "A systematic review on content-based video retrieval," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103557.
- [166] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate Web video retrieval," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 531–540.
- [167] H. Wang, F. Zhu, B. Xiao, L. Wang, and Y.-G. Jiang, "GPU-based mapreduce for large-scale near-duplicate video retrieval," *Multimedia Tools Appl.*, vol. 74, no. 23, pp. 10515–10534, 2015.
- [168] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.
- [169] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.
- [170] S. Ding, G. Li, Y. Li, X. Li, Q. Zhai, A. C. Champion, J. Zhu, D. Xuan, and Y. F. Zheng, "SurvSurf: Human retrieval on large surveillance video data," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6521–6549, Mar. 2017.
- [171] N. Zhu, W. He, Y. Hua, and Y. Chen, "Marlin: Taming the big streaming data in large scale video similarity search," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 1755–1764.
- [172] J. Lv, B. Wu, S. Yang, B. Jia, and P. Qiu, "Efficient large scale near-duplicate video detection base on spark," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 957–962.
- [173] M. N. Khan, A. Alam, and Y.-K. Lee, "FALKON: large-scale content-based video retrieval utilizing deep-features and distributed in-memory computing," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (Big-Comp)*, Feb. 2020, pp. 36–43.
- [174] F.-C. Lin, H.-H. Ngo, and C.-R. Dow, "A cloud-based face video retrieval system with deep learning," *J. Supercomput.*, vol. 76, pp. 1–21, Jan. 2020.
- [175] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1168–1181, Apr. 2007.
- [176] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*. Berlin, Germany: Springer, 2006, pp. 127–144.
- [177] Y. Aytar, M. Shah, and J. Luo, "Utilizing semantic word similarity measures for video retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [178] L. S. Kennedy, A. P. Natsev, and S.-F. Chang, "Automatic discovery of query-class-dependent models for multimodal search," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 882–891.
- [179] R. Yan, J. Yang, and A. G. Hauptmann, "Learning query-class dependent weights in automatic video retrieval," in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 548–555.
- [180] R. Lienhart, "A system for effortless content annotation to unfold the semantics in videos," in *Proc. cbavi1*, 2000, p. 45.
- [181] C. G. M. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring, "Adding semantics to detectors for video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 975–986, Aug. 2007.
- [182] S.-Y. Neo, J. Zhao, M.-Y. Kan, and T.-S. Chua, "Video retrieval using high level features: Exploiting query matching and confidence-based weighting," in *Proc. Int. Conf. Image Video Retr.* Berlin, Germany: Springer, 2006, pp. 143–152.
- [183] L.-H. Chen, K.-H. Chin, and H.-Y. Liao, "An integrated approach to video retrieval," in *Proc. 19th Conf. Australas. Database*, vol. 75, 2008, pp. 49–55.
- [184] F. Hopfgartner, J. Urban, R. Villa, and J. Jose, "Simulated testing of an adaptive multimedia information retrieval system," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, Jun. 2007, pp. 328–335.
- [185] R. Yan, A. G. Hauptmann, and R. Jin, "Negative pseudo-relevance feedback in content-based video retrieval," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 343–346.
- [186] A. G. Hauptmann, M. G. Christel, and R. Yan, "Video retrieval based on semantic concepts," *Proc. IEEE*, vol. 96, no. 4, pp. 602–622, Apr. 2008.
- [187] C. Sun and R. Nevatia, "DISCOVER: Discovering important segments for classification of video events and recounting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2569–2576.
- [188] A. Habibian and C. G. M. Snoek, "Recommendations for recognizing video events by concept vocabularies," *Comput. Vis. Image Understand.*, vol. 124, pp. 110–122, Jul. 2014.
- [189] H. Song, X. Wu, W. Yu, and Y. Jia, "Extracting key segments of videos for event detection by learning from Web sources," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1088–1100, May 2018.
- [190] H. Wang, X. Wu, and Y. Jia, "Video annotation via image groups from the Web," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1282–1291, Aug. 2014.

- [191] X. Zhang, Y. Yang, Y. Zhang, H. Luan, J. Li, H. Zhang, and T.-S. Chua, "Enhancing video event recognition using automatically constructed semantic-visual knowledge base," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1562–1575, Sep. 2015.
- [192] H. Song, X. Wu, W. Liang, and Y. Jia, "Recognizing key segments of videos for video annotation by learning from Web image sets," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6111–6126, Mar. 2017.
- [193] H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, "Principal motion components for one-shot gesture recognition," *Pattern Anal. Appl.*, vol. 20, no. 1, pp. 167–182, Feb. 2017.
- [194] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, "Multi-layered gesture recognition with Kinect," *J. Mach. Learn. Res.*, vol. 16, pp. 227–254, Feb. 2015.
- [195] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 7–12.
- [196] C. Lu, J. Jia, and C.-K. Tang, "Range-sample depth feature for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 772–779.
- [197] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 804–811.
- [198] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 716–723.
- [199] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 1057–1060.
- [200] P. Wang, W. Li, Z. Gao, C. Tang, J. Zhang, and P. Ogunbona, "Convnets-based action recognition from depth maps through virtual cameras and pseudocoloring," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1119–1122.
- [201] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona, "Action recognition from depth maps using deep convolutional neural networks," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 4, pp. 498–509, Aug. 2016.
- [202] D. Wu, L. Pigou, P.-J. Kindermans, N. D.-H. Le, L. Shao, J. Dambre, and J.-M. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, Aug. 2016.
- [203] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, "Depth pooling based large-scale 3-D action recognition with convolutional neural networks," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1051–1061, May 2018.
- [204] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4041–4049.
- [205] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1110–1118.
- [206] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.
- [207] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
- [208] P. Guler, D. Emeksiz, A. Temizel, M. Teke, and T. T. Temizel, "Real-time multi-camera video analytics system on GPU," *J. Real-Time Image Process.*, vol. 11, no. 3, pp. 457–472, Mar. 2016.
- [209] V. Pham, P. Vo, V. T. Hung, and L. H. Bac, "GPU implementation of extended Gaussian mixture model for background subtraction," in *Proc. IEEE RIVF Int. Conf. Commun. Technol., Res., Innov., Vis. Future (RIVF)*, Nov. 2010, pp. 1–4.
- [210] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting objects, shadows and ghosts in video streams by exploiting color and motion information," in *Proc. 11th Int. Conf. Image Anal. Process.*, 2001, pp. 360–365.
- [211] C. Beyan and A. Temizel, "Adaptive mean-shift for automated multi object tracking," *IET Comput. Vis.*, vol. 6, no. 1, pp. 1–12, 2012.
- [212] B. Risse, M. Mangan, L. Del Pero, and B. Webb, "Visual tracking of small animals in cluttered natural environments using a freely moving camera," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2840–2849.
- [213] S. Shantaiya, K. Verma, and K. Mehta, "A survey on approaches of object detection," *Int. J. Comput. Appl.*, vol. 65, no. 18, pp. 14–20, 2013.
- [214] B. Deori and D. M. Thounaojam, "A survey on moving object tracking in video," *Int. J. Inf. Theory*, vol. 3, no. 3, pp. 31–46, Jul. 2014.
- [215] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: An analysis of the state of the art in multiple object tracking," 2017, *arXiv:1704.02781*. [Online]. Available: <http://arxiv.org/abs/1704.02781>
- [216] X. Yin, B. Wang, W. Li, Y. Liu, and M. Zhang, "Background subtraction for moving cameras based on trajectory-controlled segmentation and label inference," *Ksii Trans. Internet Inf. Syst.*, vol. 9, no. 10, pp. 4092–4107, 2015.
- [217] S. Zhang, J.-B. Huang, J. Lim, Y. Gong, J. Wang, N. Ahuja, and M.-H. Yang, "Tracking persons-of-interest via unsupervised representation adaptation," 2017, *arXiv:1710.02139*. [Online]. Available: <http://arxiv.org/abs/1710.02139>
- [218] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2003, p. 67.
- [219] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 926–932.
- [220] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pffinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.
- [221] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, May 2006.
- [222] Y. Jin, L. Tao, H. Di, N. I. Rao, and G. Xu, "Background modeling from a free-moving camera by multi-layer homography algorithm," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1572–1575.
- [223] Y. Wu, X. He, and T. Q. Nguyen, "Moving object detection with a freely moving camera via background motion subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 236–248, Feb. 2017.
- [224] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2016, pp. 1–4.
- [225] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.*, vol. 23, pp. 1–71, Feb. 2017.
- [226] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," *Comput. Sci. Rev.*, vol. 28, pp. 157–177, May 2018.
- [227] I. Elhart, M. Mikusz, C. G. Mora, M. Langheinrich, and N. Davies, "Audience monitor: An open source tool for tracking audience mobility in front of pervasive displays," in *Proc. 6th ACM Int. Symp. Pervas. Displays*, 2017, p. 10.
- [228] G. M. Farinella, G. Farioli, S. Battiato, S. Leonardi, and G. Gallo, "Face re-identification for digital signage applications," in *Proc. Int. Workshop Video Anal. Audience Meas. Retail Digit. Signage*. Cham, Switzerland: Springer, 2014, pp. 40–52.
- [229] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 367–386, Mar. 2015.
- [230] X. Wang, X. Yang, X. He, Q. Teng, and M. Gao, "A high accuracy flow segmentation method in crowded scenes based on streakline," *Optik*, vol. 125, no. 3, pp. 924–929, Feb. 2014.
- [231] R. Mehran, B. E. Moore, and M. Shah, "A streakline representation of flow in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 439–452.
- [232] H. Su, H. Yang, S. Zheng, Y. Fan, and S. Wei, "The large-scale crowd behavior perception based on spatio-temporal viscous fluid field," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 10, pp. 1575–1589, Oct. 2013.
- [233] L. Kratz and K. Nishino, "Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 987–1002, May 2012.
- [234] P.-M. Jodoin, Y. Benezeth, and Y. Wang, "Meta-tracking for video scene understanding," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2013, pp. 1–6.
- [235] M. Lewandowski, D. Simonnet, D. Makris, S. A. Velastin, and J. Orwell, "Tracklet reidentification in crowded scenes using bag of spatio-temporal histograms of oriented gradients," in *Proc. Mex. Conf. Pattern Recognit.* Berlin, Germany: Springer, 2013, pp. 94–103.
- [236] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2871–2878.

- [237] W. Chongjing, Z. Xu, Z. Yi, and L. Yuncai, "Analyzing motion patterns in crowded scenes via automatic tracklets clustering," *China Commun.*, vol. 10, no. 4, pp. 144–154, Apr. 2013.
- [238] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1114–1127, Aug. 2008.
- [239] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1544–1554, Nov. 2008.
- [240] S. Coşar, G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, and F. Brémont, "Toward abnormal trajectory and event detection in video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 683–695, Mar. 2017.
- [241] X. Song, X. Shao, Q. Zhang, R. Shibasaki, H. Zhao, J. Cui, and H. Zha, "A fully online and unsupervised system for large and high-density area surveillance: Tracking, semantic scene learning and abnormality detection," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 2, p. 35, 2013.
- [242] L. Brun, A. Saggese, and M. Vento, "Dynamic scene understanding for behavior analysis based on string kernels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1669–1681, Oct. 2014.
- [243] D. Tran, J. Yuan, and D. Forsyth, "Video event detection: From sub-volume localization to spatiotemporal path search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 404–416, Feb. 2014.
- [244] A. R. Revathi and D. Kumar, "An efficient system for anomaly detection using deep learning classifier," *Signal, Image Video Process.*, vol. 11, no. 2, pp. 291–299, Feb. 2017, doi: [10.1007/s11760-016-0935-0](https://doi.org/10.1007/s11760-016-0935-0).
- [245] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognition—A review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 865–878, Nov. 2012.
- [246] Y. Yuan, Y. Feng, and X. Lu, "Statistical hypothesis detector for abnormal event detection in crowded scenes," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3597–3608, Nov. 2017.
- [247] G. Xiong, J. Cheng, X. Wu, Y.-L. Chen, Y. Ou, and Y. Xu, "An energy model approach to people counting for abnormal crowd behavior detection," *Neurocomputing*, vol. 83, pp. 121–135, Apr. 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231211006990>
- [248] Y. Zhang, L. Qin, R. Ji, H. Yao, and Q. Huang, "Social attribute-aware force model: Exploiting richness of interaction for abnormal crowd detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1231–1245, Jul. 2015.
- [249] S. Yi, X. Wang, C. Lu, and J. Jia, "L0 regularized stationary time estimation for crowd group analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Washington, DC, USA: IEEE Computer Society, 2014, pp. 2219–2226, doi: [10.1109/CVPR.2014.284](https://doi.org/10.1109/CVPR.2014.284).
- [250] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2909–2917.
- [251] Y.-J. Lee, Y.-R. Yeh, and Y.-C.-F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1460–1470, Jul. 2013.
- [252] B. Krausz and C. Bauckhage, "Loveparade 2010: Automatic video analysis of a crowd disaster," *Comput. Vis. Image Understand.*, vol. 116, no. 3, pp. 307–319, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1077314211002037>
- [253] D.-G. Lee, H.-I. Suk, S.-K. Park, and S.-W. Lee, "Motion influence map for unusual human activity detection and localization in crowded scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 10, pp. 1612–1623, Oct. 2015.
- [254] C. C. Loy, T. Xiang, and S. Gong, "Salient motion detection in crowded scenes," in *Proc. 5th Int. Symp. Commun., Control Signal Process.*, May 2012, pp. 1–4.
- [255] S. Vishwakarma and A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *Vis. Comput.*, vol. 29, no. 10, pp. 983–1009, 2013, doi: [10.1007/s00371-012-0752-6](https://doi.org/10.1007/s00371-012-0752-6).
- [256] K. Xu, X. Jiang, and T. Sun, "Anomaly detection based on stacked sparse coding with intraframe classification strategy," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1062–1074, May 2018.
- [257] D. Sen and B. Raman, "Video skimming: Taxonomy and comprehensive survey," *ACM Comput. Surv.*, vol. 52, no. 5, p. 106, 2019.
- [258] X.-N. Xie and F. Wu, "Automatic video summarization by affinity propagation clustering and semantic content mining," in *Proc. Int. Symp. Electron. Commerce Secur.*, 2008, pp. 203–208.
- [259] J. Wu, S.-H. Zhong, J. Jiang, and Y. Yang, "A novel clustering method for static video summarization," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 9625–9641, Apr. 2017.
- [260] H. Bhaumik, S. Bhattacharyya, S. Dutta, and S. Chakraborty, "Towards redundancy reduction in storyboard representation for static video summarization," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 344–350.
- [261] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1059–1067.
- [262] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3090–3098.
- [263] J. Mohan and M. S. Nair, "Static video summarization using sparse autoencoders," in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Feb. 2019, pp. 1–8.
- [264] Z. Ji, Y. Zhao, Y. Pang, and X. Li, "Cross-modal guidance based auto-encoder for multi-video summarization," *Pattern Recognit. Lett.*, vol. 135, pp. 131–137, Jul. 2020.
- [265] K. A. Peker, I. Otsuka, and A. Divakaran, "Broadcast video program summarization using face tracks," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 1053–1056.
- [266] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.
- [267] R.-G. Xiao, Y.-Y. Wang, H. Pan, and F. Wu, "Automatic video summarization by spatio-temporal analysis and non-trivial repeating pattern detection," in *Proc. Congr. Image Signal Process.*, vol. 4, 2008, pp. 555–559.
- [268] Y. Gao, W.-B. Wang, and J.-H. Yong, "A video summarization tool using two-level redundancy detection for personal video recorders," *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 521–526, May 2008.
- [269] B. Xu, X. Wang, and Y.-G. Jiang, "Fast summarization of user-generated videos: Exploiting semantic, emotional, and quality clues," *IEEE MultimediaMag.*, vol. 23, no. 3, pp. 23–33, Jul. 2016.
- [270] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder-decoder networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1709–1717, Jun. 2020.
- [271] J. Wu, S.-H. Zhong, and Y. Liu, "Dynamic graph convolutional network for multi-video summarization," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107382.
- [272] S.-H. Zhong, J. Wu, and J. Jiang, "Video summarization via spatio-temporal deep architecture," *Neurocomputing*, vol. 332, pp. 224–235, Mar. 2019.
- [273] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 435–441.
- [274] Y. He, Z. Qu, C. Gao, and N. Sang, "Fast online video synopsis based on potential collision graph," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 22–26, Jan. 2017.
- [275] Y. He, C. Gao, N. Sang, Z. Qu, and J. Han, "Graph coloring based surveillance video synopsis," *Neurocomputing*, vol. 225, pp. 64–79, Feb. 2017.
- [276] K. B. Başkurt and R. Same, "Improved adaptive background subtraction method using pixel-based segmenter," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 2017. [Online]. Available: <https://dspace5.zcu.cz/handle/11025/29610>
- [277] K. B. Başkurt and R. Samet, "Long-term multiobject tracking using alternative correlation filters," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 26, no. 5, pp. 2246–2259, 2018.
- [278] L. Lin, W. Lin, W. Xiao, and S. Huang, "An optimized video synopsis algorithm and its distributed processing model," *Soft Comput.*, vol. 21, no. 4, pp. 935–947, Feb. 2017.
- [279] S. A. Ahmed, D. P. Dogra, S. Kar, R. Patnaik, S.-C. Lee, H. Choi, G. P. Nam, and I.-J. Kim, "Query-based video synopsis for intelligent traffic monitoring applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3457–3468, Aug. 2020.
- [280] X. Zhu, J. Liu, J. Wang, and H. Lu, "Key observation selection-based effective video synopsis for camera network," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 145–157, Jan. 2014.
- [281] A. Mahapatra, P. K. Sa, B. Majhi, and S. Padhy, "MVS: A multi-view video synopsis framework," *Signal Process., Image Commun.*, vol. 42, pp. 31–44, Mar. 2016.
- [282] Z. Zhang, Y. Nie, H. Sun, Q. Zhang, Q. Lai, G. Li, and M. Xiao, "Multi-view video synopsis via simultaneous object-shifting and view-switching optimization," *IEEE Trans. Image Process.*, vol. 29, pp. 971–985, 2020.

- [283] J. Ferryman and A.-L. Ellis, "Performance evaluation of crowd image analysis using the PETS2009 dataset," *Pattern Recognit. Lett.*, vol. 44, pp. 3–15, Jul. 2014.
- [284] P. Foggia, G. Percannella, A. Saggese, and M. Vento, "Real-time tracking of single people and groups simultaneously by contextual graph-based reasoning dealing complex occlusions," in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill. (PETS)*, Jan. 2013, pp. 29–36.
- [285] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. G. Strintzis, "Knowledge-assisted semantic video object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1210–1224, Oct. 2005.
- [286] A. García and J. Bescós, "Video object segmentation based on feedback schemes guided by a low-level scene ontology," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Berlin, Germany: Springer, 2008, pp. 322–333.
- [287] J. Gómez-Romero, M. A. Patricio, J. García, and J. M. Molina, "Ontology-based context representation and reasoning for object tracking and scene interpretation in video," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7494–7510, Jun. 2011.
- [288] V. Haarslev and R. Möller, "Racer system description," in *Proc. Int. Joint Conf. Automated Reasoning.* Berlin, Germany: Springer, 2001, pp. 701–705.
- [289] N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs, "Detection of loitering individuals in public transportation areas," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 167–177, Jun. 2005.
- [290] B. Laxton, J. Lim, and D. Kriegman, "Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
- [291] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *Proc. CVPR*, 2011, pp. 3185–3192.
- [292] A. R. J. Francois, R. Nevatia, J. Hobbs, R. C. Bolles, and J. R. Smith, "VERL: An ontology framework for representing and annotating video events," *IEEE MultimediaMag.*, vol. 12, no. 4, pp. 76–86, Oct. 2005.
- [293] L. Snidaro, M. Belluz, and G. L. Foresti, "Representing and recognizing complex events in surveillance applications," in *Proc. IEEE Conf. Adv. Video Signal Based Surveill.*, Sep. 2007, pp. 493–498.
- [294] J. C. SanMiguel, J. M. Martínez, and A. Garcia, "An ontology for event detection and its application in surveillance video," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2009, pp. 220–225.
- [295] L. Greco, P. Ritrovato, A. Saggese, and M. Vento, "Abnormal event recognition: A hybrid approach using SemanticWeb technologies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1297–1304.
- [296] L. Chen and C. Nugent, "Ontology-based activity recognition in intelligent pervasive environments," *Int. J. Web Inf. Syst.*, vol. 5, pp. 410–430, Nov. 2009.
- [297] D. Cavaliere, S. Senatore, M. Vento, and V. Loia, "Towards semantic context-aware drones for aerial scenes understanding," in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2016, pp. 115–121.
- [298] B. Z. Yao, X. Yang, L. Lin, M. Wai Lee, and S.-C. Zhu, "I2T: Image parsing to text description," *Proc. IEEE*, vol. 98, no. 8, pp. 1485–1508, Aug. 2010.
- [299] M. Xue, S. Zheng, and C. Zhang, "Ontology-based surveillance video archive and retrieval system," in *Proc. IEEE 5th Int. Conf. Adv. Comput. Intell. (ICACI)*, Oct. 2012, pp. 84–89.
- [300] Z. Xu, L. Mei, Y. Liu, and C. Hu, "Video structural description: A semantic based model for representing and organizing video surveillance big data," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Eng.*, Dec. 2013, pp. 802–809.
- [301] Z. Xu, Y. Liu, L. Mei, C. Hu, and L. Chen, "Semantic based representing and organizing surveillance big data using video structural description technology," *J. Syst. Softw.*, vol. 102, pp. 217–225, Apr. 2015.
- [302] M. Sah and C. Direkoglu, "Semantic annotation of surveillance videos for abnormal crowd behaviour search and analysis," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
- [303] F. Sobhani, N. F. Kahar, and Q. Zhang, "An ontology framework for automated visual surveillance system," in *Proc. 13th Int. Workshop Content-Based Multimedia Indexing (CBMI)*, Jun. 2015, pp. 1–7.
- [304] A. Alam, M. N. Khan, J. Khan, and Y.-K. Lee, "Intellibr-intelligent large-scale video retrieval for objects and events utilizing distributed deep-learning and semantic approaches," in *Proc. IEEE Int. Conf. Big Data Smart Comput. BigComp*, Feb. 2020, pp. 28–35.
- [305] S. O. Ajiboye, P. Birch, C. Chatwin, and R. Young, "Hierarchical video surveillance architecture: A chassis for video big data analytics and exploration," *Proc. SPIE*, vol. 9407, Mar. 2015, Art. no. 94070K.
- [306] C.-F. Lin, S.-M. Yuan, M.-C. Leu, and C.-T. Tsai, "A framework for scalable cloud video recorder system in surveillance environment," in *Proc. 9th Int. Conf. Ubiquitous Intell. Comput. 9th Int. Conf. Autonomic Trusted Comput.*, Sep. 2012, pp. 655–660.
- [307] X. Liu, D. Zhao, L. Xu, W. Zhang, J. Yin, and X. Chen, "A distributed video management cloud platform using Hadoop," *IEEE Access*, vol. 3, pp. 2637–2643, 2015.
- [308] C. Ryu, D. Lee, M. Jang, C. Kim, and E. Seo, "Extensible video processing framework in apache Hadoop," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, vol. 2, Dec. 2013, pp. 305–310.
- [309] G. Bradski, "The OpenCV library," *Dr. Dobbs's J. Softw. Tools*, vol. 3, 2000, Art. no. 2236121.
- [310] M. Ali, A. Anjum, O. Rana, A. R. Zamani, D. Balouek-Thomert, and M. Parashar, "RES: Real-time video stream analytics using edge enhanced clouds," *IEEE Trans. Cloud Comput.*, early access, May 6, 2020, doi: 10.1109/TCC.2020.2991748.
- [311] B. White, T. Yeh, J. Lin, and L. Davis, "Web-scale computer vision using mapreduce for multimedia data mining," in *Proc. 10th Int. Workshop Multimedia Data Mining*, 2010, p. 9.
- [312] R. Pereira, M. Azambuja, K. Breitman, and M. Endler, "An architecture for distributed high performance video processing in the cloud," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 482–489.
- [313] C. Liu, K. Fan, Z. Yang, and J. Xiu, "A distributed video share system based on Hadoop," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2014, pp. 587–590.
- [314] M. A. Hossain, "Framework for a cloud-based multimedia surveillance system," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 5, May 2014, Art. no. 135257.
- [315] L. Valentín, S. A. Serrano, R. O. García, A. Andrade, M. A. Palacios-Alonso, and L. E. Sucar, "A cloud-based architecture for smart video surveillance," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, p. 99, 2017.
- [316] W. Zhang, P. Duan, Z. Li, Q. Lu, W. Gong, and S. Yang, "A deep awareness framework for pervasive video cloud," *IEEE Access*, vol. 3, pp. 2227–2237, 2015.
- [317] W. Zhang, L. Xu, Z. Li, Q. Lu, and Y. Liu, "A deep-intelligence framework for online video processing," *IEEE Softw.*, vol. 33, no. 2, pp. 44–51, Mar. 2016.
- [318] M. A. Uddin, A. Alam, N. A. Tu, M. S. Islam, and Y.-K. Lee, "SIAT: A distributed video analytics framework for intelligent video surveillance," *Symmetry*, vol. 11, no. 7, p. 911, Jul. 2019.
- [319] H. Zhang, J. Yan, and Y. Kou, "Efficient online surveillance video processing based on spark framework," in *Proc. Int. Conf. Big Data Comput. Commun.* Cham, Switzerland: Springer, 2016, pp. 309–318.
- [320] M. A. Uddin, J. B. Joollee, A. Alam, and Y.-K. Lee, "Human action recognition using adaptive local motion descriptor in spark," *IEEE Access*, vol. 5, pp. 21157–21167, 2017.
- [321] H. Wang, X. Zheng, and B. Xiao, "Large-scale human action recognition with spark," in *Proc. IEEE 17th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2015, pp. 1–6.
- [322] Q. Huang, P. Ang, P. Knowles, T. Nykiel, I. Tverdokhlib, A. Yajurvedi, P. Dapolito, X. Yan, M. Bykov, C. Liang, M. Talwar, A. Mathur, S. Kulkarni, M. Burke, and W. Lloyd, "SVE: Distributed video processing at facebook scale," in *Proc. 26th Symp. Operating Syst. Princ.*, Oct. 2017, pp. 87–103.
- [323] W. Zhang, Z. Wang, X. Liu, H. Sun, J. Zhou, Y. Liu, and W. Gong, "Deep learning-based real-time fine-grained pedestrian recognition using stream processing," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 602–609, Sep. 2018.
- [324] Citilog, Citilog, Bagnex, France, 1997.
- [325] checkvideo. 1998. *Checkvideo*. Accessed: Dec. 7, 2018. [Online]. Available: <https://www.checkvideo.com/>
- [326] Intell Vision. (2002). *Intelli Vision*. [Online]. Available: <https://www.intelli-vision.com/>
- [327] Google Inc. (2017). *Video Ai*. [Online]. Available: <https://cloud.google.com/video-intelligence/>
- [328] BlueChasm LLC. (2015). *Usher in the New Cognitive Era*. [Online]. Available: <https://bluechasm.com/>
- [329] J. H. J. Kornich and C. Casey. (2010). *Media Analytics on the Media Services Platform*. [Online]. Available: <https://docs.microsoft.com/en-gb/azure/media-services/previous/media-services-analytics-overview>

- [330] Google Inc. (2017). *Features*. [Online]. Available: <https://cloud.google.com/video-intelligence/docs/features>
- [331] J. G. Collier. (2017). *IBM Uses Watson's Power to Unlock Insights From Video in the Cloud*. [Online]. Available: <https://www-03.ibm.com/press/us/en/pressrelease/52126.wss>
- [332] IBM Cloud. (2017). *IBM Cloud Object Storage*. [Online]. Available: <https://www.ibm.com/cloud/object-storage>
- [333] (2017). *Cloudant*. [Online]. Available: <https://www.ibm.com/cloud/cloudant>
- [334] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.
- [335] F. Loewenherz, V. Bahl, and Y. Wang, "Video analytics towards vision zero," *Inst. Transp. Engineers. ITE J.*, vol. 87, no. 3, p. 25, 2017.
- [336] H. Qiu, X. Liu, S. Rallapalli, A. J. Bency, K. Chan, R. Urgaonkar, B. S. Manjunath, and R. Govindan, "Kestrel: Video analytics for augmented multi-camera vehicle tracking," in *Proc. IEEE/ACM 3rd Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, Apr. 2018, pp. 48–59.
- [337] T. Gao, Z.-G. Liu, S.-H. Yue, J.-Q. Mei, and J. Zhang, "Traffic video-based moving vehicle detection and tracking in the complex environment," *Cybern. Syst.*, vol. 40, no. 7, pp. 569–588, Sep. 2009.
- [338] Y. Lin, L. Li, H. Jing, B. Ran, and D. Sun, "Automated traffic incident detection with a smaller dataset based on generative adversarial networks," *Accident Anal. Prevention*, vol. 144, Sep. 2020, Art. no. 105628.
- [339] L. Fridman, D. E. Brown, M. Glazer, W. Angell, S. Dodd, B. Jenik, J. Terwilliger, A. Patsek, J. Kindelsberger, L. Ding, S. Seaman, A. Mehler, A. Sipperley, A. Pettinato, B. Seppelt, L. Angell, B. Mehler, and B. Reimer, "MIT advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation," 2017, *arXiv:1711.06976*. [Online]. Available: <http://arxiv.org/abs/1711.06976>
- [340] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.
- [341] J. Wang, G. Bebis, M. Nicolescu, M. Nicolescu, and R. Miller, "Improving target detection by coupling it with tracking," *Mach. Vis. Appl.*, vol. 20, no. 4, pp. 205–223, Jun. 2009.
- [342] Y. J. Tsai, Z. Hu, and C. Alberti, "Detection of roadway sign condition changes using multi-scale sign image matching (M-SIM)," *Photogramm. Eng. Remote Sens.*, vol. 76, no. 4, pp. 391–405, Apr. 2010.
- [343] R. Ayachi, M. Afif, Y. Said, and M. Atri, "Traffic signs detection for real-world application of an advanced driving assisting system using deep learning," *Neural Process. Lett.*, vol. 51, no. 1, pp. 837–851, Feb. 2020.
- [344] A. M. Callan, R. Osu, Y. Yamagishi, D. E. Callan, and N. Inoue, "Neural correlates of resolving uncertainty in driver's decision making," *Hum. Brain Mapping*, vol. 30, no. 9, pp. 2804–2812, Sep. 2009.
- [345] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by convolutional neural networks," *IET Comput. Vis.*, vol. 10, no. 2, pp. 103–114, Mar. 2016.
- [346] S. Gite and H. Agrawal, "Early prediction of Driver's action using deep neural networks," *Int. J. Inf. Retr. Res.*, vol. 9, no. 2, pp. 11–27, Apr. 2019.
- [347] S. Fleck and W. Straßer, "Smart camera based monitoring system and its application to assisted living," *Proc. IEEE*, vol. 96, no. 10, pp. 1698–1714, Oct. 2008.
- [348] Z. Zhou, X. Chen, Y.-C. Chung, Z. He, T. X. Han, and J. M. Keller, "Activity analysis, summarization, and visualization for indoor human activity monitoring," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1489–1498, Nov. 2008.
- [349] J. Aertssen, M. Rudinac, and P. Jonker, *Fall and Action Detection in Elderly Homes*. Maastricht, The Netherlands: AAATE, 2011.
- [350] D. Brulin, Y. Benezeth, and E. Courtial, "Posture recognition based on fuzzy logic for home monitoring of the elderly," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 5, pp. 974–982, Sep. 2012.
- [351] G. Sacco, V. Joumier, N. Darmon, A. Dechamps, A. Derreumaux, J.-H. Lee, J. Piano, N. Bordone, A. König, B. Teboul, R. David, O. Guerin, F. Bremond, and P. Robert, "Detection of activities of daily living impairment in Alzheimer's disease and mild cognitive impairment using information and communication technology," *Clin. Interventions Aging*, vol. 7, p. 539, Dec. 2012.
- [352] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1274–1286, Nov. 2012.
- [353] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1577–1591, Jul. 2013.
- [354] V. Pragada, "Intrusion detection and video surveillance activation and processing," U.S. Patent 14 172 880, Aug. 6, 2015.
- [355] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2016, pp. 712–726.
- [356] G. Xu and W. K. Cobb, "Loitering detection in a video surveillance system," U.S. Patent 9 208 675, Dec. 8, 2015.
- [357] P. Shivakumara, D. Tang, M. Asadzadehkalajahi, T. Lu, U. Pal, and M. H. Anisi, "CNN-RNN based method for license plate recognition," *CAAI Trans. Intell. Technol.*, vol. 3, no. 3, pp. 169–175, Sep. 2018.
- [358] B. Krausz and R. Herpers, "MetroSurv: Detecting events in subway stations," *Multimedia Tools Appl.*, vol. 50, no. 1, pp. 123–147, Oct. 2010.
- [359] J.-L. Shih, C.-C. Han, and K.-C. Yan, "Illegal entrant detection at a restricted area in open spaces using color features," in *Proc. 40th Annu. Int. Carnahan Conf. Secur. Technol.*, Oct. 2006, pp. 68–74.
- [360] H. Singh, "Applications of intelligent video analytics in the field of retail management: A study," in *Supply Chain Management Strategies and Risk Assessment in Retail Environments*. Hershey, PA, USA: IGI Global, 2018, pp. 42–59.
- [361] H. Xu, P. Lv, and L. Meng, "A people counting system based on head-shoulder detection and tracking in surveillance video," in *Proc. Int. Conf. Comput. Design Appl.*, vol. 1, Jun. 2010, pp. V1–394.
- [362] I. Haritaoglu and M. Flickner, "Attentive billboards: Towards to video based customer behavior understanding," in *Proc. 6th IEEE Workshop Appl. Comput. Vis. (WACV)*, Dec. 2002, pp. 127–131.
- [363] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 128–135.
- [364] R. Sicre and H. Nicolas, "Human behaviour analysis and event recognition at a point of sale," in *Proc. 4th Pacific-Rim Symp. Image Video Technol.*, Nov. 2010, pp. 127–132.
- [365] A. D. Popescu, A. Balmin, V. Ercegovac, and A. Ailamaki, "PRE-DIC: Towards predicting the runtime of large scale iterative analytics," *Proc. VLDB Endowment*, vol. 6, no. 14, pp. 1678–1689, Sep. 2013.
- [366] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Mag.*, vol. 35, no. 4, pp. 105–120, 2014.
- [367] Y.-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *ACM Comput. Surv.*, vol. 31, no. 4, pp. 406–471, Dec. 1999.
- [368] X. Zhu, J. Wang, J. Wang, and X. Qin, "Analysis and design of fault-tolerant scheduling for real-time tasks on Earth-observation satellites," in *Proc. 43rd Int. Conf. Parallel Process.*, Sep. 2014, pp. 491–500.
- [369] E. Bortnikov, A. Frank, E. Hillel, and S. Rao, "Predicting execution bottlenecks in map-reduce clusters," in presented at the 2012.
- [370] Y. Zhai, Y.-S. Ong, and I. W. Tsang, "The emerging 'big dimensionality,'" *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Aug. 2014.
- [371] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, and J. Shao, "Learning in high-dimensional multimedia data: The state of the art," *Multimedia Syst.*, vol. 23, no. 3, pp. 303–313, Jun. 2017.
- [372] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, New York, NY, USA, 1984, pp. 47–57, doi: [10.1145/602259.602266](https://doi.org/10.1145/602259.602266).
- [373] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proc. 23rd Int. Conf. Very Large Data Bases (VLDB)*, San Francisco, CA, USA: Morgan Kaufmann, 1997, pp. 426–435. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645923.671005>
- [374] K.-I. Lin, H. V. Jagadish, and C. Faloutsos, "The TV-tree: An index structure for high-dimensional data," *VLDB J.*, vol. 3, no. 4, pp. 517–542, Oct. 1994.
- [375] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geometry (SCG)*, 2004, pp. 253–262.

- [376] P. O'Neil, E. Cheng, D. Gawlick, and E. O'Neil, "The log-structured merge-tree (LSM-tree)," *Acta Inf.*, vol. 33, no. 4, pp. 351–385, Jun. 1996.
- [377] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis, "The case for learned index structures," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 489–504.
- [378] G. De Francisci Morales, "SAMOA: A platform for mining big data streams," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 777–778.
- [379] J. Lu, S. C. H. Hoi, J. Wang, P. Zhao, and Z.-Y. Liu, "Large scale online kernel learning," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1613–1655, Jan. 2016.
- [380] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitaya, D. De Silva, D. Alahakoon, and D. Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4679–4690, Dec. 2019.
- [381] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1, no. 10. New York, NY, USA: Springer, 2001.
- [382] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, "Enterprise data analysis and visualization: An interview study," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2917–2926, Dec. 2012.
- [383] A. Kumar, R. McCann, J. Naughton, and J. M. Patel, "Model selection management systems: The next frontier of advanced analytics," *ACM SIGMOD Rec.*, vol. 44, no. 4, pp. 17–22, May 2016.
- [384] Z. Zhang, B. Cui, Y. Shao, L. Yu, J. Jiang, and X. Miao, "PS2: Parameter server on spark," in *Proc. Int. Conf. Manage. Data*, 2019, pp. 376–388.
- [385] J. Jiang, F. Fu, T. Yang, and B. Cui, "SketchML: Accelerating distributed machine learning with data sketches," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 1269–1284.
- [386] Y. Huang, T. Jin, Y. Wu, Z. Cai, X. Yan, F. Yang, J. Li, Y. Guo, and J. Cheng, "FlexPS: Flexible parallelism control in parameter server architecture," *Proc. VLDB Endowment*, vol. 11, no. 5, pp. 566–579, Jan. 2018.
- [387] R. M. Haralick, "Performance characterization in computer vision," in *Proc. BMVC*. London, U.K.: Springer, 1992, pp. 1–8.
- [388] W. Förstner, "DAGM workshop on performance characteristics and quality of computer vision algorithms," Tech. Univ. Braunschweig, Braunschweig, Germany, Tech. Rep., 1997.
- [389] P. J. Phillips and K. W. Bowyer, "Empirical evaluation of computer vision algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 289–290, Apr. 1999.
- [390] H. Christensen and W. Foerstner, "Special issue on performance evaluation," *Mach. Vis. Appl.*, vol. 9, no. 5, 1997.
- [391] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, performance evaluation for video surveillance systems," in *Proc. IEEE Conf. Adv. Video Signal Based Surveill.*, Sep. 2007, pp. 476–481.
- [392] A. G. Hauptmann and M. G. Christel, "Successful approaches in the TREC video retrieval evaluations," in *Proc. 12th Annu. ACM Int. Conf. Multimedia (MULTIMEDIA)*, 2004, pp. 668–675.
- [393] The Home Office Scientific Development Branch, "Imagery library for intelligent detection systems (i-LIDS)," in *Proc. IET Conf. Crime Secur.*, 2006, pp. 445–448.
- [394] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *J. Big Data*, vol. 2, no. 1, p. 8, 2015.
- [395] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and M. Zhou, "A deep latent factor model for high-dimensional and sparse matrices in recommender systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 15, 2019, doi: 10.1109/TSMC.2019.2931393.
- [396] A. Mazrekaj, I. Shabani, and B. Sejdiu, "Pricing schemes in cloud computing: An overview," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 80–86, 2016.
- [397] S. Khalid, S. Wu, A. Alam, and I. Ullah, "Real-time feedback query expansion technique for supporting scholarly search using citation network analysis," *J. Inf. Sci.*, vol. 45, no. 4, Jul. 2019, Art. no. 0165551519863346.
- [398] D. Kang, P. Bailis, and M. Zaharia, "Challenges and opportunities in DNN-based video analytics: A demonstration of the Blazeit video query engine," in *Proc. CIDR*, 2019, pp. 1–4.
- [399] Y. Lu, A. Chowdhery, and S. Kandula, "VisFlow: A relational platform for efficient large-scale video analytics," in *Proc. ACM Symp. Cloud Comput. (SoCC)*, 2016. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/visflow-relational-platform-efficient-large-scale-video-analytics/>
- [400] D. Kang, P. Bailis, and M. Zaharia, "Blazeit: Optimizing declarative aggregation and limit queries for neural network-based video analytics," *Proc. VLDB Endowment*, vol. 13, no. 4, pp. 533–546, Dec. 2019. [Online]. Available: <https://doi.org/10.14778/3372716.3372725>, doi: 10.14778/3372716.3372725.
- [401] A. Ghosh and I. Arce, "Guest editors' introduction: In cloud computing we trust—but should we?" *IEEE Secur. Privacy Mag.*, vol. 8, no. 6, pp. 14–16, Nov. 2010.
- [402] S. Khusro, A. Alam, and S. Khalid, "Social question and answer sites: The story so far," *Program*, vol. 51, no. 2, pp. 170–192, Jul. 2017.
- [403] A. Alam, S. Khusro, I. Ullah, and M. S. Karim, "Confluence of social network, social question and answering community, and user reputation model for information seeking and experts generation," *J. Inf. Sci.*, vol. 43, no. 2, pp. 260–274, Apr. 2017.
- [404] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Appl. Innov.*, vol. 2, nos. 6–10, p. 71, 2016.



ing, information service engineering, knowledge engineering, knowledge graph, and graph mining.

AFTAB ALAM received the master's degree in computer science with a specialization in Web engineering from the Department of Computer Science, University of Peshawar, Peshawar, Pakistan. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University (Global Campus), South Korea. His research interests include big data analytics, distributed computing, video retrieval, social computing, machine/deep learning, information service engineering, knowledge engineering, knowledge graph, and graph mining.



IRFAN ULLAH received the master's degree in computer science from the National University of Sciences and Technology (NUST), Islamabad, Pakistan. He is currently serving as a Research Assistant at the Department of Computer Science and Engineering, Kyung Hee University (Global Campus), South Korea. His research interests include natural language processing, social computing, crisis informatics, machine/deep learning, OS design and optimization on memory systems, big data analytics, and distributed computing.



YOUNG-KOO LEE (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1992, 1994, and 2002, respectively. Since 2004, he has been with the Department of Computer Science and Engineering, Kyung Hee University (Global Campus). His research interests include data mining, online analytical processing, and big data processing.