This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

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

**AFTAB ALAM, IRFAN ULLAH, AND YOUNG-KOO LEE**

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

Corresponding author: Young-Koo Lee (e-mail: yklee@khu.ac.kr)

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

 **ABSTRACT** The proliferation of multimedia devices over the Internet of Things (IoT) generates anunprecedented 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 analyticssurvey, deep learning, distributed computing, intermediate results orchestration, cloud computing.

**I. INTRODUCTION**

IDEOS are generated and uploaded regularly Vto the cloud. Many sources include CCTV, smartphones, drones, etc., are actively contributing to video generation leads to the evolution of Intel-ligent Video Analytics (IVA) and management sys-tems. 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 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].

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 1 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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 or-ganizations have successfully deployed video man-agement 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 exam-ple, 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 con-tain 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 ap-proaches. Therefore, more comprehensive and so-phisticated 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, extract-ing the insights from the video is a challenging task. A considerable size of video data poses significant challenges for video management and mining sys-tems 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 tech-nologies.

Cloud computing is an infrastructure for provid-ing convenient and ubiquitous remote access to a shared pool of configurable computing resources. These resources can be managed with minimal man-agement effort or service [9]. Big data technologies, such as Hadoop or Spark echo system, are soft-ware 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 gov-ern 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, scalabil-ity, 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 taxon-omy 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.

1. **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 con-sidered. The definition of big data evolved and has been described in terms of three, four, or five char-acteristics. 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 consider-able 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

2 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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 aug-ments the Variety property. The pace of data genera-tion 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 charac-teristic. 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 de-tection, 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 chal-lenges 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 thou-sands 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 comput-ing 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 com-puting 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 ser-vice 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 var-ious client devices using a thin client or program interfaces [9].

1. **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 in-herits the big data characteristics while estab-lishing its relationship with cloud computing.

We propose a distributed, layered, service-oriented, and lambda style [29] inspired ref-erence 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) orches-tration, 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 ana-lytics and deep learning techniques and tools on distributed infrastructures along with popu-lar computer vision benchmark datasets.

To the best of our knowledge, this is a first sur-veying video big data analytics, our research

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 3 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

targets the most recent approaches that encom-pass 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 an-alytics and show how the industrial IVA solu-tions are provided.

In this study, we further write real word IVA application areas, which projects the signifi-cance 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 dimen-sionality, online learning on video big data, model management, parameter servers, and distributed learning, evaluation issues and op-portunities, IVA services statistics maintenance and ranking in the cloud, IVA-as-a-Service (IVAaaS) and cost model, video big data man-agement, and privacy, security and trust.

The remaining paper is organized as follows. Sec-tion II and III is about recent studies, and scope and nomenclature, respectively. Section IV discusses the proposed CVAS. Literature review has been pre-sented 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 dis-cussed. Finally, Section IX concludes this study.

**II. RECENT STUDIES**

Various studies have been conducted that discus 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 con-text 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.

1. 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.
2. focus on big techniques and general applica-tions along with challenges. Zhou et al. [40], Che et al. [41], and Pouyanfar et al. [42] studied and reported the research issues of big data practices concerning ML techniques, data mining algorithms, and multimedia data respectively. Zhu et al. [43] presented a comprehensive study on big data ana-lytics 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. [42], Zhu et al. [43], H. and Zahid et al. [44].

From the literature, it is clear that the recent studies ignored large-scale unstructured video an-alytics 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 lit-erature and proposes an in-depth distributed cloud computing-based IVA reference architecture.

**III. SCOPE AND NOMENCLATURE**

We clarify some nomenclature being used and scop-ing this study. The first one is IVA, which is “any surveillance solution that utilizes video technolo-gies to automatically manipulate and/or perform actions on live or stored video images [45]”. 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 produce alerts in case of abnor-mality. 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 [46],

4 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 1.** Recent Surveys Related to Video Big Data Analytics,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | References | Target Domain | Pros | Cons |  |
| Analytics | Liu et al. [30] | Video analytics | Comprehensive theory & applications of | -Lacking big data & cloud computing |  |
| -Lack of big data & cloud computing, |  |
| Hu et al. [32] |  | - CBVR components, research trends |  |
|  | Olatunji et al. [31] | based surveillance | video analytics-based surveillance | and its research issues |  |
|  | - Not service oriented |  |
|  |  |  |  |  |
| Video | Patel et al. [33] | CBVR | and issues | & its research issues. |  |
| Haseyama et al. [34] |  | -Limited to CBVR |  |
|  |  |  |
| Khan et al. [37] | Vehicle surveillance | Generic layered architecture for video | - Lacking big data and cloud computing |  |
| Int. | techniques and challenges |  |
| Tsai et al. [38] | systems in ITS | analytics-based vehicle surveillance |  |
|  |  |  |  | - Abstract architecture - Consider IVA for ITS |  |
|  |  | Abnormal behavior in | A generic architecture for abnormal behaviors | -Lacking big data and cloud computing |  |
|  | Mabrouk et al. [36] | discussion. |  |
|  | surveillance videos | in video surveillance systems |  |
|  |  | -Limited to abnormal behavior |  |
|  |  |  |  |  |
|  | Hashem et al. [16] |  | A thorough overview of big data techniques, | -Big data processing over cloud computing only. |  |
|  | Cloud computing | -Lack of video big data analytics, |  |
|  | Agrawal et al. [39] | challenges and open issues in cloud computing | service-oriented architecture, |  |
|  |  |  |
| data |  |  |  | research issues, and challenges. |  |
| Khan et al. [37] | Big data technologies | Overview of big data techniques,opportunities, | Limited discussion on video big data |  |
| Big | Tsai et al. [38] | and challenges | analytics |  |
|  |  |
| Zhou et al. [40] | Machine Learning (ML) | A framework of ML on big data, and open | Lack of IVA discussion |  |
|  |  |
|  | research issues |  |
|  |  |  |  |  |
|  | Che et al. [41] | Big data mining | A comprehensive overview on big data | Lack of video analytics and its |  |
|  | mining and challenges | research issues |  |
|  |  |  |  |
|  | Pouyanfar et al. [42] | Multimedia | A comprehensive overview of multimedia data | Limited discussion on video big data analytics |  |
|  | Zhu et al. [43] | ITS | -A study of big data analytics in ITS | Limited discussion of video big data |  |
|  | -Consider multimedia for ITS | in the context of ITS analytics |  |
|  |  |  |  |
|  | H. Zahid et al. [44] | Telecommunications | A study of big data analytics in telecommunication. | Limited discussion of video big data |  |
|  | in the context of telecommunication analytics |  |
|  |  |  |  |  |

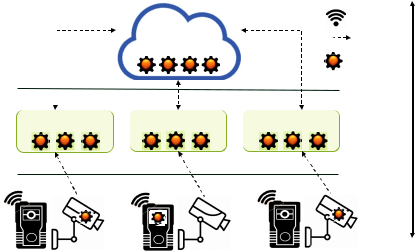
and Zoneminder [47]), and a storage device. All the cameras send the video data for analytics against contextual video analytics algorithms and warn the operator if anomaly detected. OVAS has many lim-itations, e.g., maintenance, software up-gradation, expensive hardware, scalability, and unable to deal 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 environ-ments the IVA solutions are made available under the as-a-Service (aaS) paradigm, i.e., IVAaaS. In FVAS, the IVA solutions are geographically dis-tributed 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 pro-visioning challenges. Whereas the CVAS is more suitable for offline IVA because of the relatively high response time and latency. The hierarchy and relation among CVAS, FVAS, and EVAS is shown in Fig. 1. The batch video data analytics are per-formed 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, re-spectively, while utilizing big data computing tech-nology. However, for the ease of understandability, throughout this paper, we use the notion of CVAS.

To analyze a video in CVAS, a video undergoes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | Wireless | -**Low** |  |
| CLOUD-BASED VIDEO |  | Batch intelligent | | |  |  | Wired | Processing |  |
|  | video analytics | | |  |  | network |  |
| ANALYTICS SYSTEM |  |  |  |  |
|  |  |  |  |  |  | Compute |  |  |
|  |  |  |  |  |  |  |  |  |
| Video Big data Analytics |  |  |  |  |  |  | Slots |  |  |
| Analytics |  |  |  |  |  |  |  | speed/Response |  |
| FOG-BASED VIDEO | Fog-based VAS | Fog-based VAS | | | Fog-based VAS | |  |  |  |
| ANALYTICS SYSTEM | (Real-time IVA) | (Real-time IVA) | | | (Real-time IVA) | |  |  |  |
| Real-time video stream |  |  |  |  |  |  |  |  |  |
| EDGE BASED VIDEO |  |  |  |  |  |  |  | Time |  |
|  |  |  |  |  |  |  | - |  |
| ANALYTICS SYSTEM |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **High** |  |
| Video data origination |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |



**FIGURE 1.** Geo-distributed video analytics infrastructure.

through different phases, as shown in Fig. 2a. In Fig. 2a, Video Source are the sources which either generate videos streams from sources con-nected 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 ac-quire videos from the Video Source. In the con-text 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

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 5 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

low-level temporal features suitable for machines, while scenes and sequence/story are considered to be the high-level features that are suitable for hu-man 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 ex-tractor, 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 out-put of an IVA service are restricted to the Video Source and IR, respectively. The User represents the stakeholder of the CVAS, such as administra-tor, consumer, IVA researchers, and practitioners. A Domain is a specific real-word 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 con-stitutes s 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. | | |
| FVSA | Fused Video Surveillance Architec- | | |
|  | ture. |  |  |
| 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. | | |
| IVAaaS | 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. | |  |

6 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access

A Alam et al.: Video Big Data Analytics in the cloud

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User | Domain |  | IVA System | Video |  |
|  |  |  |  | Scene |  |
| Ingest | Transform | | Infer | Shot |  |
| Frame |  |
|  |  |  |  |  |
|  |  |  |  | Key |  |
| Video Source | | Persist / Publish | | Frame |  |
|  | (a) |  |  | (b) |  |

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

Analysis Processing

Spatio- - Low-Level temporal - Mid-Level

Spatial - High-Level

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 de-tails 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 Min-ing Layer (VBDML), Knowledge Curation Layer (KCL), and Web Service Layer (WSL).

VBDCL is the foundation layer and is responsi-ble for large-scale big data management throughout the life-cycle of IVA, i.e., from data acquisition to early persistence to archival and deletion [48]. 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 spa-tial 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 as aaS. These IVAaaS 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 IVAaaS 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 conve-nient 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 Synchro-nization (RVSAS), Distributed Persistent Data Store (DPDS), and VBDCL Business Logic.

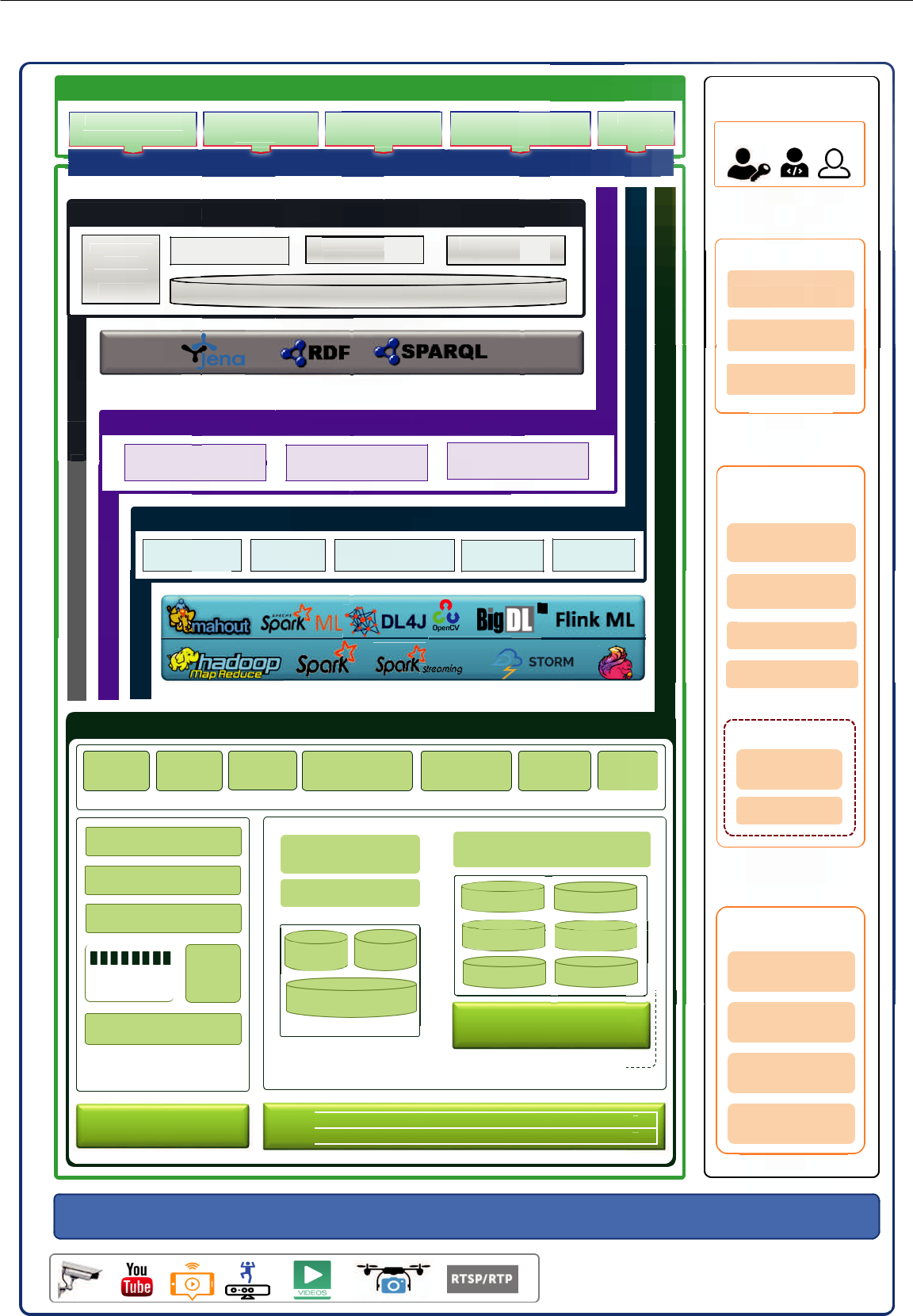
|  |  |
| --- | --- |
| VOLUME 4, 2016 | 7 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud



|  |
| --- |
| **FAULT-TOLERANCE, SERVICE ORCHISTRACTION, AND PERFORMANCE** |

**DOMAIN SPECIFIC INTELLIGENT VIDEO ANALYTICS PLUGABLE SERVICES**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Content Based | Surveillance & | |  | Video | Semantic Based IVA | | | **O**ther | |  |
| Videodeo Retrieval | | | Securitycurity |  | Summarizationumma | **Se**rvices | |  |
|  |  |  |  |  |
|  |  |  |  | **SERVICE APIs** | |  |  |  |  |  |  |
|  |  | **KNOWLEDGE CURATION LAYERLAYER** | | | |  |  |  |  |  |  |
|  | Feature To | SPARQL Queries | | Semantic Web | | Video On**t**ology | |  |  |  |  |
|  | Ruleule Language | | Vocabuocabulary | |  |  |  |  |
|  | Videoideo |  |  |  | APIs |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Ontology |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Mapper |  | Video Ontology (RDF Triple Store) | | | |  |  | VBDML |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | APIs |  |
|  |  |  |  |  |  |  |  |  |  | VBDML |  |
| KCLAPIs |  | **VIDEO BIG DATADATA MININGMININ LAYER** | | | | | |  |  | VBDCLAPIs |  |
| Clust**ering** | |  | Regression | | Cl**assi**fication | | |  |  |
|  |  |  |  |  |
|  |  |  | **VIDEO BIGG DATADATA PROCESSINGPRO** | | | | **LAYER** |  |  |  |  |
|  | Video Pre- | | Feature |  | Dimensionality | Model | | Similarity | |  |  |
|  | processing | | Extractor | | Reductionuction | Manager | | Measure | |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **VIDEO BIG DATA CURATION LAYER** | | |  |  |  |
| User | Data | Model | (R/B)IVA Algo. | (R/B) IVA | Video | Cluster |  |
| Source | & Services | Ser. Discry. & | Ontology |  |
| Manager | Mgt. | MM |  |
| Manager | Orchestration | Sub. Mgr. | Mgr. |  |
|  |  |  |  |

VBDCL Business Logic

**Web Services Layer**

**USERS**

**Data Source**

Video Stream

Source services

Batch Video Data

Services

Model

Services

**IVA Algorithm &**

**IVA Services**

Service

Discovery

Service

Subscription

Notification

Evaluation

**SDK**

Develop/

Deploy

Lifelong Vid. St. Monitor

Int. Results Manager

|  |  |
| --- | --- |
| DFS Active & | ISD Active & Passive DRW |
| Passive DRW | (CRUD, Conn. Mgr.) |

Video Stream Consumer

Service

Dist.

Msg.

Distributed Broker

Msg. Broker Mgr.

Video Stream Acquisition

Service

**Real-time Video**

**Stream Acquisition &**

DFS Handlers

|  |  |  |
| --- | --- | --- |
| RAW | Model |  |
| Video |  |
| Lrnr. Sp. |  |
| Space |  |
|  |  |

Project Space

User Space

**Unstructured**

**Persistent Dist. DS**

|  |  |
| --- | --- |
| User Profile | Data Source |
| & Logs | DS |

|  |  |  |
| --- | --- | --- |
| Algo. & | Int. Result DS |  |
| Service DS |  |
|  |  |

|  |  |
| --- | --- |
| Service Sub. | Anomalies |
| Log | DS |

**Distributed BDS**

**(Data Stores)**

**Immediate Strt. Dist. DS**

**Cluster Mgt. &**

**Monitoring**

Messaging

System CMM

Distributed

File Systems

Distributed

**Synchronization**

**Distributed Persistent DS**

BDS

|  |
| --- |
| **SECURITY, SCALABILITY, LOADBALANCING,** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DistMessaging** |  |  |  |  |  |  |  |  |  |  |  |
| **DFS** | **1** | \* | **\*** | \* | **\*** | \* | **\*** | \* | |  |
| **System** |  |  |  |  |  |  |  |  |  |  |
| \* | \* | \* | \* | \* | \* | \* |  | NN |  |
|  |  |  |

Distributed

Prsg. Eng.

**CLOUD INFRASTRUCTURE [CPU, GPU, FPGA, Storage, Network]**

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

8 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 2.** Popular message-oriented middlewares.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kafka [49] | RabbitMQ [50] | RocketMQ [1] | ActiveMQ [51] |  |
| Format | String, Byte | Map, String | String, Byte | String, Map, Byte, Object |  |
| Protocol | Kafka | AMQP, MQTT, | Rocket, MQ | STOMP, REST, |  |
| STOMP, XMPP | XMPP, AMQP |  |
|  |  |  |  |
| Pub/Sub | 3 | 3 | 3 | 3 |  |
| Point-to-Point | - | 3 | 3 | 3 |  |
| Cluster | Stateless cluster | Simple clusters | Stateless cluster | Simple clusters |  |
| Load balancing | 3 | 3 | 3 | 3 |  |
| Persistence | 3 | 3 | 3 | 3 |  |
| Encryption | SSL | TLS | TLS | TLS |  |
| Authentication | SASL | SASL | - | JAAS |  |
| APIs | Java, .NET, PHP, Go, JS | Java, .NET, PHP, Ruby, | Java, C++, Go | Java, PHP, C++, Ruby, |  |
| Python, Ruby, Perl, Erlang | Python, Go, JavaScript | Perl, C#, Python, C |  |
|  |  |  |

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 ex-ecutors for on-the-fly processing against the sub-scribed IVA service. Handling a tremendous amount of video streams, both processing and storage are subject to lose [52]. 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.

Distributed Message Broker, also known as message-oriented-middleware [53], is an indepen-dent application that is responsible for buffering, queuing, routing, and delivering the messages to the consumers being received from the message producer [54]. Message broker should be able to handle permission control and failure recovery. A message broker generally supports routing methods like direct worker queue, and/or publish-subscribe

1. Similarly, the message consumer component receives the messages from the message broker ei-ther periodically (cron-like consumer) or continu-ously (daemon-like consumer). Generally and for the sake of scalability, message consumers are de-ployed on separate servers independently of mes-sage producers [55]. Some popular distributed mes-saging systems are shown in table 2.

RVSAS provides client APIs on the top of a dis-tributed messaging system for the proposed frame-work. 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 tem-porarily to the distributed broker server. The worker system, on which an IVA service is configured, e.g., activity recognition, reads the data from the dis-

tributed 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 Con-sumer Service (VSCS), Intermediate Results Man-ager (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 algo-rithm, 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 pro-cessing. If a particular video stream source is sub-scribed against an RIVA service, then the VSAS gets the configuration metadata from the Data Source 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 opera-tions 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

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 9 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



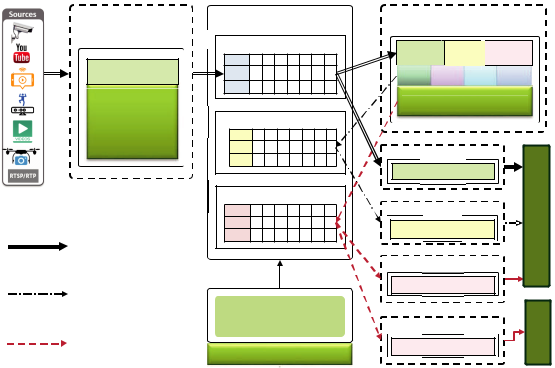
Distributed Broker. If a video-stream source “C1” is subscribed to the RIVA service “S1” then the VSAS will rout the mini-batch of the video stream to queue RIVA\_1 in the Broker Cluster as shown in Fig. 4.

A Alam et al.: Video Big Data Analytics in the cloud

queue and persists to the IR data store for future use so that to avoid recomputation.

e: Lifelong Video Stream Monitor

|  |  |  |
| --- | --- | --- |
| **Video Stream** | **Distributed Message** |  |
| **Acquisition** | **Broker Cluster** |  |
| **Service** | Queue: RIVA\_ID |  |
|  | P1 |  |
| **VSAS Client** | P2 |  |
|  | Pn |  |
| Broker Client |  |  |
| Service | Queue: RIVA\_IR\_ID |  |
|  |  |
| Video analytics lib. | P1 |  |
|  | P2 |  |
|  | Pn |  |



Queue: RIVA\_A\_ID

P1

P2

Pn

Video Data

|  |  |
| --- | --- |
| Intermediate | Distributed |
| Results | Message |
|  | Broker Manager |
| Anomalies | Broker Client APIs |

**Video Stream Analytics**

**Consumer Cluster**

VSCS IRH  LLM

RIVA service

Stream processing engine,

Video analytics lib,

Stream Subscriber.

|  |  |  |
| --- | --- | --- |
| **VSCS Cluster** | **PERSISTANCE** |  |
| VSCS |  |
| **IRH Cluster** |  |  |
| IRH | **(DPB,** |  |
|  |  |
| **LLM Cluster** | **ISB)** |  |
| LLM |  |  |

|  |  |  |
| --- | --- | --- |
| **LLM Cluster** | **NotificationClient** |  |
| LLM |  |
|  |  |

The domain-specific RIVA service process the video stream for anomalies or abnormal activities. If ~~an~~ anomalies are detected, then it is sent to the dis-tributed broker queue (i.e., RIVA\_A\_ID) by using the LVSM instance. To generate notification base response, LVSM follow standard observer-based concept [56]. Based on this approach, the LVSM module reads the anomalies from the respective distributed broker queue, i.e., RIVA\_A\_ID and no-tify the clients in near real-time and simultaneously persisted to the ISDDS.

2) Distributed Persistence Data Store

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

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 ser-vices, 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 al-gorithm 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 com-puter. Thus we design IRM that sends and gets the IR to and from the topic RIVA\_IR\_ID in the dis-tributed broker cluster. Similarly, this module is also responsible for reading the IR from the respective

The second component of the VBDCL is DPDS. The DPDS component provides the facilities of per-manent and distributed big-data persistence storage of both structured and unstructured data. The DPDS provides abstraction in two levels on the acquired video data ie. 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 ac-cess 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 Datastore (DBDS) can be lever-aged.

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 exam-ple, IP-cameras, Kinect, body-worn cameras, etc.,

10 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 3.** Popular Distributed File Systems.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Features | CEPH [57] | GlusterFS [58] | HDFS [59] | Lustre [60] | MooseFS [61] | Quantcast File [62] |  |
| Initial release | 2007 | 2007 | 2006 | 2003 | 2008 | 2011 |  |
| License | LGPL 2.1 | GPLv3+ | Apache 2.0 | GPLv2 | GPLv2 | Apache 2.0 |  |
| Availability | 3 | 3 | Transparent | 3 | Master | Master |  |
| master failover |  |
|  |  |  |  |  |  |  |
| Tolerate disk failures | 3 | 3 | 3 | 3 | 3 | 3 |  |
| Scale | PBs | PBs | PBs | PBs | PBs | PBs |  |
| Shards | 3 | 3 | 7 | 3 | 7 | 7 |  |
| Commodity Hardware | 3 | 3 | 3 | 3 | 3 | 3 |  |
| Language | C++ | C | Java | C | C | C |  |
| POSIX Complaint | 3 | 3 | 7 | 3 | 3 | 3 |  |
|  | Librados (C, | NFS, FUSE, SMB, |  |  |  |  |  |
|  | Python, C++, | Java and | NFS-Ganesha, |  | C++ client, |  |
| APIs | Swift, libgfapi, | FUSE |  |
| Ruby), S3, | C client, HTTP | NFS, SMB | FUSE |  |
|  | libglusterfs |  |  |
|  | Swift, FUSE |  |  |  |  |  |
|  |  |  |  |  |  |  |

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 an-alytics algorithms and services is managed through Video Analytics Algorithm and Service meta-store, respectively. As stated that in IVA pipelining envi-ronment, 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 Subscrip-tion 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 scal-ability, high availability, partition tolerance, consis-tency, and durability. Furthermore, the data store should fulfill the read/write access demands of the BIVA operations, and RIVA, interaction, and vi-sualization. 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 [63], [64]. The features of NoSQL data stores include horizontal scalability, data replica-tion, distributed indexing, simple API, flexibility, and consistency [65]. NoSQL lacks true Atomicity, Consistency, Isolation, Durability (ACID) transac-tions, unlike Relational Database Management Sys-tem (RDBMS). In the context of Consistency, Avail-ability, Partition Tolerance (CAP) theorem [66], it has to compromise on either consistency or avail-ability while choosing partition tolerance. NoSQL can further be categorized as Document, Key-value, and Extensible stores. A key-value data store is re-sponsible for storing values and indexes for search-ing. Document datastore is used for document stor-age, indexing, and retrieval. Extensible data store stores extensible records that can be partitioned vertically and horizontally across the nodes.

NewSQL data stores provide the characteris-tics of both NoSQL and RDBMS: ACID transac-tional consistency of relational databases with facil-ities 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 Ta-ble 4 along with the respective properties. The ‘A’ and ‘C’ in CAP is not equal to the ‘A’ and ‘C’ in ACID [67].

1. 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 accord-ing 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

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 11 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

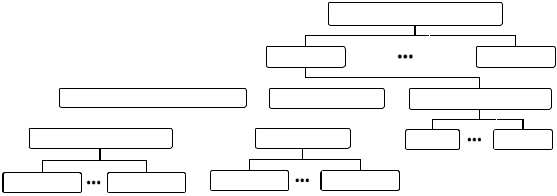
**TABLE 4.** Popular Distributed Data Stores.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Type | DS Name | Release | Developer | OS | Language | D | HS | SL |  | Properties | | | | | | |  |
| -value | Voldemort | 2009 | LinkedIn | 3 | Java | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Riak | 2009 | Basho | 3 | Erlang | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Key |  |  | Tech. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Redis [68] | 2009 | Salvatore | 3 | C | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  | Sanfilippo |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Document | SimpleDB | 2007 | Amazon | 7 | Erlang | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  |
| DynamoDB | 2012 | Amazon | 7 | Java | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  | CouchDB | 2005 | ASF | 3 | Erlang | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  | |  | |  | |  |
|  | MongoDB [69] | 2009 | 10gen | 3 | C, C++ | 3 | 3 | 3 |  | C |  | | AP | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | | |  |  | |  |  |
|  |  |  |  |  |  |  |  |  |  |  | |  | | |  | |  |
|  |  |  |  |  |  |  |  |  |  |  | | | | | | |  |
| Extensible | HBase [70] | 2010 | ASF | 3 | Java | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  | | |  |  | |  |  |
| BigTable [71] | 2005 | Google | 7 | C, C++ | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  | HyperTable | 2008 | Zvents | 3 | C++ | 3 | 3 | 3 |  | CAP | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  | | |  |  | |  |  |
|  | Cassandra | 2008 | ASF | 3 | Java | 3 | 3 | 3 |  | C | | | AP | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | | | |  | |  |
|  |  |  |  |  |  |  |  |  |  |  | | |  |  | |  |  |
|  |  |  |  |  |  |  |  |  |  |  | | | | | | |  |
| NewSQL | MySQL Cluster | 2004 | Oracle | 3 | C++ | 3 | 3 | 7 |  | ACID | | | | | | |  |
| VoldDB [72] | 2009 | VoltDB Inc | 3 | Java, C++ | 3 | 3 | 7 |  | ACID | | | | | | |  |
|  |  |  |
|  | ClustrixDB | 2007 | Clustrix, | 7 | C | 3 | 3 | 7 |  | ACID | | | | | | |  |
|  | Inc. |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | NuoDB | 2008 | NuoDB | 7 | C++ | 3 | 3 | 7 |  | 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.

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.



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | hdfs/user/tornado/ | | | |  |  |
|  |  |  |  | U-ID1 | | |  |  |  | U-IDn |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Raw Video Space | |  | Model Space | | |  |  | Project Space | |  |
|  |  | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |
| Batch Datasets | | |  | Streaming | | |  | P-ID1 | | P-IDn |  |
| B-ID1 | B-IDn | | S-ID1 |  |  | S-IDn |  |  |  |  |  |
|  |  |  |  |  |  |  |

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

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 main-tained through video data sources. IVA life-cycle

may need different models for training and testing purposes. The Model Space is 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 respec-tive 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 implemen-tation 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, con-nected through high-speed networks while deploy-ing specialized distributed data and management software. In distributed data-intensive applications, large-scale data is always moving across the clus-ter and thus demands a distributed, scalable, reli-able, and fault-tolerant file system [73] known as DFS. DFS is a file system that provides access to replicated files across multiple hosts on a com-puter network, ensuring performance, data local-ity, high availability, scalability, reliability, security, uniform access, and fault-tolerance. Currently, var-

12 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

ious 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.

1. Active and Passive Data Readers and Writer

This module gives read-write access to the under-lying 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 process-ing 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 busi-ness logic. It implements six different modules, i.e., User Manager, Data Source Manager, Model Manager, (R/B)IVA Algorithm and Service Man-ager, (R/B)IVA Service Discovery and Subscription Manager, Video Ontology Manager, Cluster Man-agement and Monitoring.

The User Manager module encapsulates all the user-related operations such as new user account creation, access role assignment, and session man-agement. Through the Data Source Manager Model Manager modules, the user can man-age the VSDS, 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, re-spectively. 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 cre-ate 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 sub-scribe 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 on-tology 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.

* 1. **DISTRIBUTED INTELLIGENT VIDEO ANALYTICS**

IVA performs the complex tasks of extracting sig-nificant knowledge and information of interest from the video data, i.e., structural patterns, behavior patterns, content characteristics, event patterns [74], [75] and their relationship in the form of classi-fication 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 par-allel manner while utilizing distributed scale-out computing technologies [76], such as MapReduce and Apache Spark. Distributed IVA not only sig-nificantly improves the performance, but also re-duces the analytics cost. Based on the generic IVA life-cycle and motivated by scikit-learn [77], 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 Prepro-cessing, Feature Extractor, and Dimensionality Re-duction is designed.

a: Video Preprocessing

The quality of data plays an active and sig-nificant 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 includ-ing frame extraction [78], frame-resizing, frame-conversion from RGB to grayscale [79], shot bound-ary detection [80], segmentation [81], transcoding [82], and many more. In the first step, frames

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 13 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

are extracted from a video for processing. Several frame selection algorithms are available, for exam-ple, 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 [83]. The spatial operations highly depend on the scenario and objective. Spatial opera-tions include frame resizing (for reducing computa-tional 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 imple-ments distributed feature extraction algorithms. The performance of ML is highly dependent upon the type of data representation or features [84]. 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 fea-tures from the raw videos that can be interpreted by the ML algorithm [85], [86]. In this context, several feature extraction algorithms have been in-troduced for video data. These feature extraction approaches can be categorized into static features of keyframes [87]–[89], object features [90], [91], dynamic/motion feature extraction [85], [92], [93], trajectory-based features extraction [85], [86], [94], [95], and deep learning-based feature extraction [96]–[104].

c: Feature Selection and Dimensionality Reduction

Feature selection and dimensionality reduction re-duce the size of the features. Large sizes of fea-ture sets are expensive in terms of time for train-ing and/or performing classification acquired by trained classifiers. For example, Principal Compo-nent 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). In-appropriate or partially relevant features can nega-tively 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 spe-cific set of limited features in real-time. For exam-ple, Online Feature Selection selects and inputs a specific number of small features to the classifiers in real-time [105]. In order to accelerate the training process, [106] used non-linear and group-based fea-ture selection techniques, based on Adaptive Feature Scaling (AFS), to process the data with substantial dimension sizes. [107] 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 [74], [75]. In this context, VBDML layer hosts three types of compo-nents, 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 cate-gory while using predefined classes. The Regres-sion component includes different algorithms, e.g., Linear Regression, Decision Tree Regression, Lo-gistic Regression, and many more, predicting a continuous-valued attribute associated with objects rather than discrete values. The Clustering compo-nent encapsulates algorithms, e.g., K-Mean, spec-tral 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 con-siders the complete training data to learn and gen-erates models. The batch-learning algorithm is ex-pected 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 [108]. In the context of model generaliza-tion, online learning is expected to work well by accurately predicting the predefined set of inputs

1. Online learning is used in the environment when continuous learning from the data is required to learn new patterns instead of batch learning.

14 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 5.** Comparision of popular deep learning architectures.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Architecture | Year | Error | Parameters | Depth | Description and Pros | Cons |  |
|  |  |  |  |  | – First popular CNN architecture | -Poor scaling to diverse classes |  |
| LeNet | 1998 | - | 60 K | 5 | -Utilize spatial correlation to decrease the | of images |  |
| parameters and computation | -Large size filters |  |
|  |  |  |  |  |  |
|  |  |  |  |  | -Automatic learning of feature hierarchies | -Extraction of low-level features |  |
|  |  |  |  |  | -Low, mid and high-level feature extraction |  |  |
|  |  |  |  |  | using large and small size filters on initial |  |  |
|  |  |  |  |  | (5x5 and 11x11) and last layers (3x3) | -Inactive neurons in the first and |  |
|  |  |  |  |  | -The notion of deep and wide CNN architecture |  |
|  |  |  |  |  | second layers |  |
| AlexNet | 2012 | 15.3% | 60 M | 8 | -Introduced regularization in CNN |  |
| -Aliasing artifacts in the learned |  |
|  |  |  |  |  | -Initiated GPUs parallelization |  |
|  |  |  |  |  | feature-maps due to large filter size |  |
|  |  |  |  |  | - Deeper and wider than the LeNet |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  | - Uses Relu ,dropout and overlap Pooling |  |  |
|  |  |  |  |  | - GPUs NVIDIA GTX 580 |  |  |
|  |  |  |  |  | -The notion of parameter tuning by visualizing |  |  |
| ZFNet | 2013 | 14.8% | - | 8 | the output of intermediate layers | -Extra information processing is |  |
| -Reduced both the filter size and stride in the | required for visualization |  |
|  |  |  |  |  |  |
|  |  |  |  |  | first two layers of AlexNet |  |  |
|  |  |  |  |  | - Introduces block concept |  |  |
|  |  |  |  |  | -The notion of utilizing Multiscale Filters within |  |  |
|  |  |  |  |  | the layers | -Heterogeneous topology led to |  |
|  |  |  |  |  | -The notion of a split, transform and merge | tedious parameter customization |  |
| GoogleNet | 2014 | 6.67% | 4 M | 22 | -Parameters reduction by utilizing the bottleneck | -Chances of losing the |  |
|  |  |  |  |  | layer, global average pooling at the last layer and | valuable information because of |  |
|  |  |  |  |  | Sparse Connections | representational bottleneck |  |
|  |  |  |  |  | -Utilization of auxiliary classifiers to enhance the |  |  |
|  |  |  |  |  | convergence rate |  |  |
|  |  |  |  |  | - Small kernel size | -Use of computationally expensive |  |
| VGGNet | 2014 | 7.3% | 138 M | 19 | -The notion of effective receptive field |  |
| fully connected layers |  |
|  |  |  |  |  | -The notion of simple and homogenous topology |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | -A bit complex architecture |  |
|  |  |  |  |  | -The error rate was decreased for deeper networks | -Lowers information of feature-map |  |
| ResNet | 2015 | 3.6% | - | 152 | -Introduced the notion of residual learning | in feed forwarding |  |
| -Improves the effect of vanishing gradient issue | -Over adaption of hyperparameters |  |
|  |  |  |  |  |  |
|  |  |  |  |  | - Identity mapping based skip connection | for a specific task, due to the |  |
|  |  |  |  |  |  | stacking of the same modules |  |

3) Distributed Deep Learning for IVA

Handcrafted features, e.g., Scale Invariant Fea-ture Transform (SIFT) [109], Local Binary Pat-tern (LBP) [110], Histogram of Oriented Gradi-ents (HOG) [111], etc., generates high dimensional features vectors and resultantly facing the issue of scalability. Recently, Convolutional Neural Network (CNN) based approaches have shown performance superiority in tasks like optical character recognition [112], and object detection [113]. The motive of the deep learning is to scale the training in three dimensions, i.e., size and complexity of the models [114], proportionality of the accuracy to the amount of training data [115], and the hardware infras-tructure scalability [116]. 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 [113]. A CNN or ConvNet is a type of neural network that can recognize visual patterns directly from the pix-els 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 [97], [98], [117],

1. 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, vari-ous open-source architecture have been developed. Some of the well-known and state-of-the-art CNN architectures are LeNet-5 [119], AlexNet [118], ZFNet [120], GoogleNet [121], VGGNet [122], and ResNet [123]. The comparison of these architec-tures can be found in Table 5. Similarly, several frameworks have been developed to eliminate the need for a manual definition of gradient propaga-tion. Table 6 summarizes these libraries along with the comparisons.

TensorFlow is a popular deep learning library de-signed for ML and deep learning. It supports the de-ployment 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 Lu-minoth, Kera, and TensorLayer were built on the top of TensorFlow. Caffe2, developed by Berkeley AI Research, is another library to build their deep learn-ing models efficiently along with GPUs’ support in a distributed environment. PyTorch, maintained by Facebook, is a scientific computing framework

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 15 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access

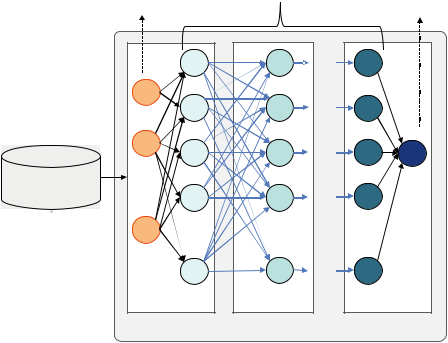


with wide support for machine learning models and algorithms. PyTorch offers rich pre-trained models that can be easily reused. MXNET is a deep learning library suitable for fast numerical computation for both single and distributed ecosystems. Likewise, some more deep learning libraries have been de-veloped, 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 at al. [130] utilized six learning algorithms, i.e., biogeography-based optimization, particle swarm optimization, genetic algorithm, ant colony op-timization, 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 infras-tructure with multiple computing nodes (equipped with powerful GPUs) is the best option, but it leads to numerous challenges. First is the effective utiliza-tion 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 re-searchers. There are three approaches, i.e., model, data, and pipeline distribution for leveraging dis-tributed computing in distributed computing [131],

1. In the former, the DL model is partitioned in logical fragments, and loaded to different worker agents for training, as shown in Fig. 6.

|  |  |  |
| --- | --- | --- |
| Input | Hidden layers | Output |
| layer |  | layer |



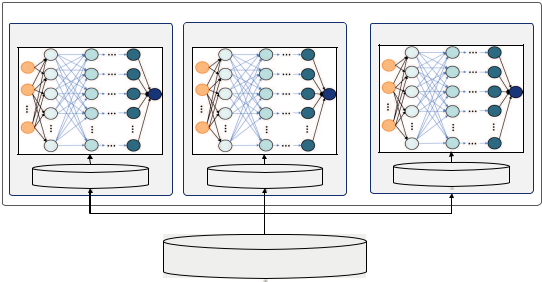
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | … |  |
|  |  |  | … |  |
|  |  |  | … |  |
| Training | … |  | … |  |
| Video Data |  |  |
|  | … | … | … |  |
|  |  |  | … |  |
|  | Agent-1 | Agent-2Agent-n | |  |
|  | Parameter synchronizations | | |  |

**FIGURE 6.** Scalable deep learning utilizing model distribution.

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 train-

A Alam et al.: Video Big Data Analytics in the cloud

ing 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 [132]. 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 generat-ing intermediate results. The intermediate results are combined to produce the complete result. Hadoop and Spark require and process data naturally dis-tributed in the manner and popular research trend nowadays.



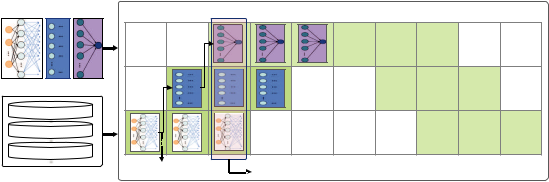
|  |  |  |
| --- | --- | --- |
|  | Parameter synchronizations |  |
| Agent-1 | Agent-2 | Agent-n |

|  |  |  |
| --- | --- | --- |
|  |  | … |
| Partition-1 | Partition-2 | Partition-3 |

Training Video Data

**FIGURE 7.** Scalable deep learning utilizing data distribution.

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 [133].



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Parameter synchronizations | |  |  |  |
|  | B | B | B |  |  |
|  |  | B | B | B |  |
| Partition-1 |  |  |  |  |  |
| Partition-2 |  |  | B | B | B |
| Partition-n |  |  |  |  |  |
| Data flow | Parallel Processing |  |  |  |  |

**FIGURE 8.** Deep learning with pipelining (Figure motivated by [133]).

16 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 6.** Summary of popular deep learning libraries.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Features | TensorFlow [124] | Caffe2 [125] | PyTorch [126] | MXNet [127] | CNTK [128] | Matlab DL Toolbox [129] |  |
| 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 |  |
|  | Python (Keras), C/C++, | Python, |  | C++, Python, Julia, | Python, C++, |  |  |
| Interface | Java, Go, JavaScript, | Python, C++ | Matlab, JavaScript, | MATLAB |  |
| MATLAB, C++ | C#, and Java. |  |
|  | R, Julia, Swift |  | R Go, Scala, Perl, Clojure |  |  |
|  |  |  |  |  |  |
| GPU | 3 | 3 | 3 | 7 | 3 | 3 |  |
| 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 | TensorLayer, Keras, | - | - | Gluon | Keras | - |  |
| Libraries |  |
|  |  |  |  |  |  |  |
|  | Tensorboard for | Fast, scalable, |  |  | Good performance | easy for scientists, |  |
|  | lightweight, | Easy to use and debug, | Lightweight and fast, |  |
|  | visualization, well | and scalability, | model visualization, |  |
| Pros | cross-platform | flexible, well documented, | memory-efficient, highly |  |
| documented, large | highly optimized, | high-performance |  |
|  | support, server | declarative data parallelism | scalable |  |
|  | user community | support for Spark | CUDA code generation, |  |
|  | optimized interface |  |  |  |
|  |  |  |  |  |  |  |
|  | Debugging issues, heavy, | Small community, | Lacking model serving, | Small community, |  | Closed-source, |  |
| Cons | Limited community | expensive in terms of |  |
| difficult for novice | modest documentation | visualization tools | difficult to learn |  |
|  |  | execution and price |  |
|  |  |  |  |  |  |  |

**TABLE 7.** Comparison of big data engines.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Features | Hadoop [134] | Spark [135] | Flink [136] | Storm [137] | Samza [138] |  |
| Batch | 3 | 3 | 3 | 7 | 7 |  |
| Stream | 7 | near real-time | Real-time | Real-time event based | Event based |  |
| Data Flow | No loops, Chain of stages | Direct Acyclic Graph | Controlled Cyclic | Direct Acyclic Graph | Kafka Dependent |  |
| Dependency Graph |  |
|  |  |  |  |  |  |
| Fault- | High and application | Recovers lost work | Based on distributed | Automatically restart dead | Migrates the task into |  |
| restart is not required | workers or migrate in case | another node in case of |  |
| tolerance | in case of failure. | snapshots |  |
| in case of failure. | of node failure. | node failure. |  |
|  |  |  |  |
| Scalability | 3 | 3 | 3 | 3 | 3 |  |
| Latency | Higher | Low | Low | Low | Low |  |
| Cost | Less expensive | Expensive | Expensive | Expensive | Expensive |  |
| Security | Kerberos authentication | Shared secret, HDFS | Kerberos authentication, | Kerberos authentication | Kerberos authentication |  |
| ACLS and Kerberos | SSL |  |
|  |  |  |  |  |
|  | By default Fair Scheduler, | Acts as its own flow | Own scheduler and can | Default, isolation, |  |  |
| Scheduler | Capacity Scheduler, | scheduler because of | multi-tenant and resource | Use YARN scheduler. |  |
| use YARN scheduler. |  |
|  | and is pluggable. | in-memory computation. | aware scheduler. |  |  |
|  |  |  |  |
| Language | Java, C/C++, Ruby, | Java, Scala, Python, | Java, Scala, | Java, Clojure, and | Java, scala and can |  |
| can support non-JVM |  |
| Support | Groovy, Perl and Python | and R. | Python, and R. | support JVM languages. |  |
| languages. |  |
|  |  |  |  |  |  |
| Windowing | 7 | Time-based | Time-based, Count based | Time-based, Count based | Time-based |  |
|  | Mahout [139], DL4J [140], | MLLib (Native) [143], |  |  |  |  |
|  | DL4J, Mahout, DL4J, | FlinkML (Native), | Don’t have native |  |  |
|  | Keras [141], Caffe [125], | Don’t have native. |  |
| ML Library | TensorFlow, Keras, Caffe, | Mahout, TensorFlow, | ML Lib, SAMOA 3rd |  |
| TensorFlow [124], | SAMOA 3rd Party |  |
|  | H20, PyTorch [126], | Keras, Caffe, H20 | Party [144] |  |
|  | BigDL [142] |  |  |
|  | BigDL |  |  |  |  |
|  |  |  |  |  |  |

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.

Hadoop MapReduce [145] is a distributed pro-gramming model, developed based on GFS [73], for data-intensive tasks. Apache Spark follows a similar programming model like MapReduce but extends it with Resilient Distributed Datasets (RDDs), data sharing abstraction [135]. 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 [135], [146]. Spark support interactive op-erations, Directed Acyclic Graph (DAG) processing, and process streaming data in the form of mini-batches in near real-time [147]. 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 [136]. 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 pro-viding 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 differ-

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 17 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

ent 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 [143], Mahout [139], FlinKML

1. are list of some open-source ML packages built on the top of Hadoop, Spark and Flink, re-spectively, that support many scalable learning al-gorithms. For deep learning, various open-source libraries have been develop including TensorFlow [124], DeepLearning4J [140], Keras [141], Caffe [125], H20 [148], BigDL [142], and PyTorch [126]. 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 environ-ment 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. Im-ageNet [149] 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 [151], [151], [153], [156], [157]. Google released YouTube-8M [160] and consisting of eight mil-lion diverse types of automatically labeled videos. Deng A et al. [159] proposed the HowTo100M dataset comprising of web videos with narrated instructions. Dataset like MediaEval2015 [154], and Trecvid2016 [151] are designed to support CBVR related research. For sports IVA, the Sports-1M dataset karpathy2014large is proposed and com-posed of 487 classes along with ground truth. YACVID [155] 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 clas-sification.

**C. KNOWLEDGE CURATION LAYER**

Videos low-level processing produces feature de-scriptors that summarize characteristics of data quantitatively. The high-level analytics is more as-

sociated 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 [161].

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 standard-izes the basic terminology that governs the video on-tology, 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 ex-press domain-specific rules and logic for reasoning. When videos are classified and tagged by the VB-DML then the respective IR are persistent to VB-DCL 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.

**E. IVA SERVICE EXAMPLE SCENARIOS**

We show two example scenarios, i.e., how to de-velop BIVA and RIVA services under L-CVAS. Hadoop and Spark MapReduce type operations nat-urally fit in a BIVA. Fig. 11, shows the block dia-gram 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

18 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access

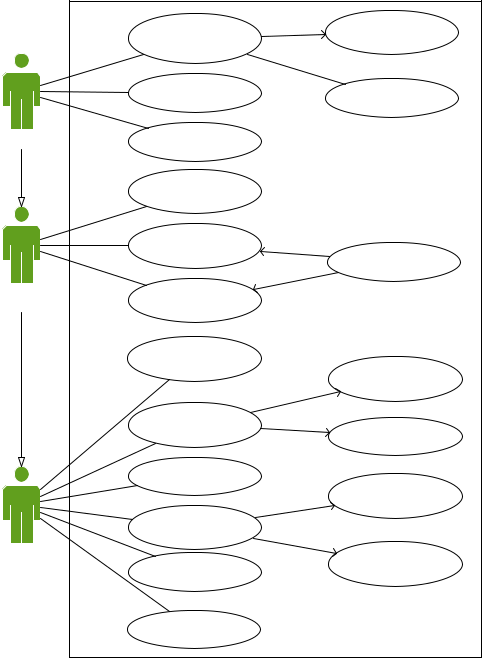


A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 8.** Popular computer vision benchmark datasets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Instances | Classes | Ground Truth | Description |
| YFCC100M [149] | 100M | 8M | Partially | Video and image, understanding |
| YouTube-8M [150] | 8M | 4,716 | Automatic | Video classification |
| Trecvid 2016 [151] | Different | Different | Partially | Human action detection |
| UCF-101 [152] | 13K | 101 | Yes | Human action detection |
| Kinetics [153] | 306K | 400 | Yes | Human action detection |
| MediaEval [154] | - | - | - | Videos from BBC |
| YACVID [155] | 75207 Frames | - | Yes | A multi-camera high-resolution dataset |
| ActivityNet [156] | 849 video hours | 203 | Yes | Human activity |
| HMDB [157] | 7K Videos | 51 Classes | Yes | Human Motion Recognition |
| Sports-1M [98] | IM | 487 | Yes | The YouTube Sports-1M Dataset |
| AVA [158] | 57K | Different | Yes | Spatiotemporally Localized Atomic Visual Actions |
| HowTo100M [159] | 100M | 23k | - | Narrated instructional web videos |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **WƌŽǀŝ Ğ ŝƐƚƌŝ ƚ Ğ sŝ ĞŽ** | | | | **ффŝŶĐů Ğхх** | | **ZĞĂůͲƚŝŵĞ sŝ ĞŽ ^ƚĞĂŵ** | | |  |
| **ŶĂůǇƚŝĐƐ >ŝ** | | **Ɛ** |  |
|  | **ŶĂůǇƚŝĐƐ W/Ɛ** | |  |  |  |  |  |  |  |
|  |  |  |  | **ффŝŶĐů** | **Ğхх** |  |  |  |  |
|  | **DĂŶĂŐĞ hƐĞƌƐ** | | |  |  | **KĨĨůŝŶĞ sŝ ĞŽ ^ƚĞĂŵ** | | |  |
|  |  |  |  |  |  | **ŶĂůǇƚŝĐƐ >ŝ** | | **Ɛ** |  |
| **ŵŝŶ** | **ů ƐƚĞƌ DĂŶĂŐĞŵĞŶƚ** | | |  |  |  |  |  |  |
|  | **ĂƚĂ DĂŶĂŐĞŵĞŶƚ** | | |  |  |  |  |  |  |
|  | **ƌĞĂƚĞ EĞǁ sŝ** | | **ĞŽ** |  |  |  |  |  |  |
|  | **ŶĂůǇƚŝĐƐ ^ĞƌǀŝĐĞ** | | | **ффĞǆƚĞŶ** | **хх** | **ŝƐƚƌŝ ƚ** | **Ğ sŝ ĞŽ ŶĂůǇƚŝĐƐ** | |  |
|  |  |  |  | **ффĞǆƚĞŶ** | **хх** |  | **W/Ɛ** |  |  |
| **ĞǀĞůŽ Ğƌ** | **ƌĞĂƚĞ EĞǁ sŝ** | | **ĞŽ** |  |  |  |  |
|  |  |  |  |  |  |
|  | **ŶĂůǇƚŝĐƐ ůŐŽƌŝƚ** | | **ŵ** |  |  |  |  |  |  |
|  | **ZĞŐŝƐƚĞƌͬ>ŽŐŝŶ** | | |  |  |  |  |  |  |
|  |  |  |  |  |  | **ZĞĂůͲƚŝŵĞ sŝ ĞŽ ^Ž ƌĐĞ** | | |  |
|  |  |  |  | **ффŝŶĐů** | **Ğхх** |  |  |  |  |
|  | **sŝ** | **ĞŽ ĂƚĂ ^Ž** | **ƌĐĞƐ** | **ффŝŶĐů** | **Ğхх** | **ĂƚĐ** | **sŝ ĞŽ ĂƚĂƐĞƚ** | |  |
|  |  |  |  |  |  |  |
|  | **WƌŽĨŝůĞ DĂŶĂŐĞŵĞŶƚ** | | |  |  | **ZĞĂůͲƚŝŵĞ sŝ** | | **ĞŽ** |  |
|  |  |  |  |  |  |  |
|  | **^ĞƌǀŝĐĞ ŝƐĐŽǀĞƌǇͬ** | | | **ффŝŶĐů** | **Ğхх** |  | **^ĞƌǀŝĐĞƐ** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **^** | **ƐĐƌŝ ƚŝŽŶ** |  | **ффŝŶĐů** | **Ğхх** |  |  |  |  |
| **ŽŶƐ ŵĞƌƐ** |  |  |  |  |  |  |  |
| **ŶĂůǇƚŝĐƐ Θ** | |  |  |  | **ĂƚĐ** | **sŝ ĞŽ ^ĞƌǀŝĐĞƐ** | |  |
|  |  |  |  |  |
|  | **sŝƐ ĂůŝǌĂƚŝŽŶ** | |  |  |  |  |  |  |  |
|  | **W Ɛ** | **EŽƚŝĨŝĐĂƚŝŽŶ** | |  |  |  |  |  |  |



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

videos are loaded, the distributed video transcod-ing (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 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.

The flow for a single RIVA along with an exam-ple 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, fea-tures 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 Execu-tion 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 execu-tion 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 de-ployed 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

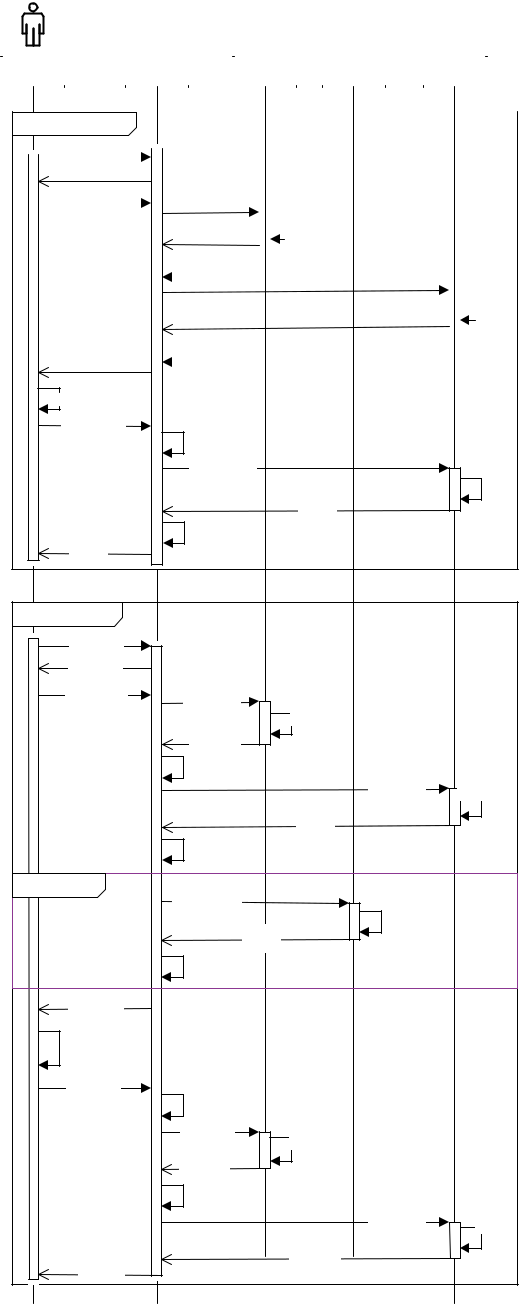
|  |  |
| --- | --- |
| VOLUME 4, 2016 | 19 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User: A, D** | | | |  |  |  |  |  | **WSL/** | | | | | |  |  |  |  | **ISDDS** | | | |  |  |  | **RVSAS** | |  |  | **UPDDS** | | | | |  |  |  |
|  |  |  |  |  | **VBDCL-BL** | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **IVA Algorithm Creation** | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | Request | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | AlgoService | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | return |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | AlgoService | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | POST(Algo) |  |  |  |  |  |  |  |  |  | Algo | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | .create(Obj) | | | |  |  |  |  | Algo.Create | | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | (AlgoObj) | | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | return | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | AlgoObj | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | if(created) | |  | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | DPB | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | .create | |  |  |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | (AlgoObj) | |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | return | |  | |  |  |  |  |  |  | DPB. | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | AlgoObj | |  |  |  |  |  |  |  | create | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | if(created) | | | | | | |  |  |  |  |  |  |  |  |  |  | (Space | | |  |  |
|  |  |  |  |  |  | return Algo | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Name) | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | SDK | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



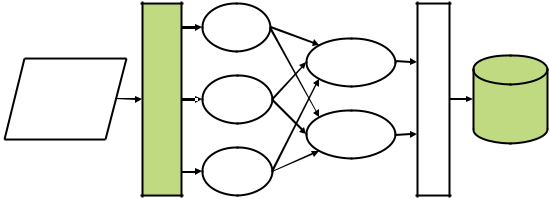
Develop Algo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| POST |  |  |  |  |  |
| (Deploy Algo) |  |  |  | DPB. |  |
|  | Build Algo |  |  |  |
|  | Algo |  |  | persist |  |
|  |  |  | (Algo) |  |
|  | .deploy(Code) |  |  |  |
|  |  |  |  |  |
|  |  | return |  |  |  |
|  |  | AlgoObj |  |  |  |
| return | if(successful) |  |  |  |  |
|  |  |  |  |  |
| boolean |  |  |  |  |  |
| **IVA Service Creation** |  |  |  |  |  |
| Request |  |  |  |  |  |
| IVA-Service |  |  |  |  |  |
| return |  |  |  |  |  |
| IVA-Service |  |  |  |  |  |
| POST | IVA-Service | IVAService. |  |  |  |
| (IVA-Service) | .create(Obj) |  |  |  |
| Create(Obj) |  |  |  |
|  |  |  |  |  |
|  | return |  |  |  |  |
|  | ServiceObj |  |  |  |  |
|  | if(created) |  | DPB | create |  |
|  |  |  |  |
|  |  |  | .create | (IVA-Service |  |
|  |  |  |  |  |
|  |  |  | (IVA-Service | Space) |  |
|  |  |  |  |  |
|  |  | return | Obj) |  |  |
|  |  |  |  |  |
|  |  | AlgoObj |  |  |  |
|  | if(created) |  |  |  |  |
| **If RIVA** | TopicManager |  |  |  |  |
|  | .create |  |  |  |  |
|  | (ServiceObj) |  | create |  |  |
|  |  |  |  |  |
|  | return |  | (ServiceObj) |  |  |
|  |  |  |  |  |
|  | AlgoObj |  |  |  |  |
|  | if(created) |  |  |  |  |
| return |  |  |  |  |  |
| Service SDK |  |  |  |  |  |
| Develop |  |  |  |  |  |
| IVAService |  |  |  |  |  |
| PiplineAlgos |  |  |  |  |  |
| IVA-Service | Build |  |  |  |  |
| Deploy | IVA-Service |  |  |  |  |
|  | IVA-Service | IVA-Service. |  |  |  |
|  |  |  |  |  |
|  | .Deploy() | update() |  |  |  |
|  |  |  |  |  |
|  | return |  |  |  |  |
|  | ServiceObj |  |  |  |  |
|  | if(success) |  |  | update |  |
|  |  | DPB.update |  |  |
|  |  |  | (Service |  |
|  |  |  |  |  |
|  |  |  | (IVA-Service | Space) |  |
|  |  |  |  |  |
|  |  | return | Obj) |  |  |
|  |  |  |  |  |
| return |  | IVA- |  |  |  |
| (boolean) |  | ServiceObj |  |  |  |

**FIGURE 10.** The sequence diagram for the IVA algorithm andservice 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.

the subscription. 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

A Alam et al.: Video Big Data Analytics in the cloud



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Map |  |  |  |  |
| Video | (DFS)UPDS |  | Reduce | Output |  |  |
|  |  |  |  |
| Dataset, | Map |  | IR-DS |  |
|  |  |  |  |
| (DPDS, |  |  |  |  |
|  |  |  |  | (ISDDS) |  |
| VBDCL) |  |  |  |  |  |
|  |  | Reduce |  |  |  |
|  |  |  |  |  |  |
|  |  | Map |  |  |  |  |
|  | **Input:** Video Dataset | |  |  |  |  |
|  | **Output:** IR |  |  |  |  |  |
|  | VBDCL.DataSourceManager.getVideoDataset (“dataset-ID”) | | | |  |  |



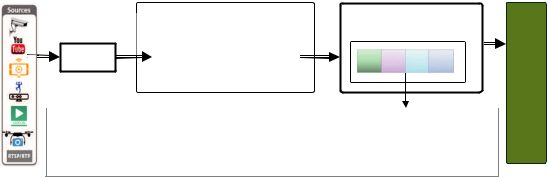
.pipe(VBDPL.VideoProcessing.videoTranscoding())

.pipe(VBDPL.FeatureExtractor.extFeature())

.pipe(VBDML.Classification.objClassification())

.pipe(VBDCL.IRManager.persist(“int. results”))

**FIGURE 11.** BIVA service example pipeline under Lambda CVAS.



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Broker Cluster** | | | | | | **Video Stream** | |  |  |
|  |  | QUEUE: RIVA\_ID | | | | | | | **Analytics** | |  |  |
| **VSAS** |  |  |  |  |  |  |  |  |  | **ISDDS)(IRDS,** | **PERSISTANCE** |  |
|  |  | P1 | ….. | T3 | T2 | T1 |  |  |
|  |  |  | IVAS | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | |  |  |  |  |
| *//Stream Acquisition and Producer* | | | | | | | | |  |  |  |  |
| **Input**: camerID | |  |  |  |  |  |  |  |  |  |  |  |

**Output**: Acquire and Produce Video streams to the Broker Cluster

RVSAS.VSAS.streamEventProducer(cameraID)



**Input**: serviceID, camerID

**Output**: Produce IR and Anomalies

While(true){ //Consume video stream mini-batch

RVSAS.VSCS.consume(“serviceID”, “cameraID”)

.pipe(VBDPL.VideoProcessing.VideoTranscoding())

.pipe(VBDPL.FeatureExtractor.extFeature())

.pipe(VBDML.Classification.objClassification()) .pipe(RVSAS.VSCS. IRManager(“IR”, “serviceID”, cameraID)) .pipe(VBDCL.LVSM.produceAnomalies(“A”, “serviceID”,cameraID)))}

**FIGURE 12.** RIVA service example pipeline under Lambda CVAS.

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 inter-faces 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 messag-ing 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 per-sisted 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 clus-ter ’V’ then processes the consumed video data, encodes it, and extracts the metadata from the video.

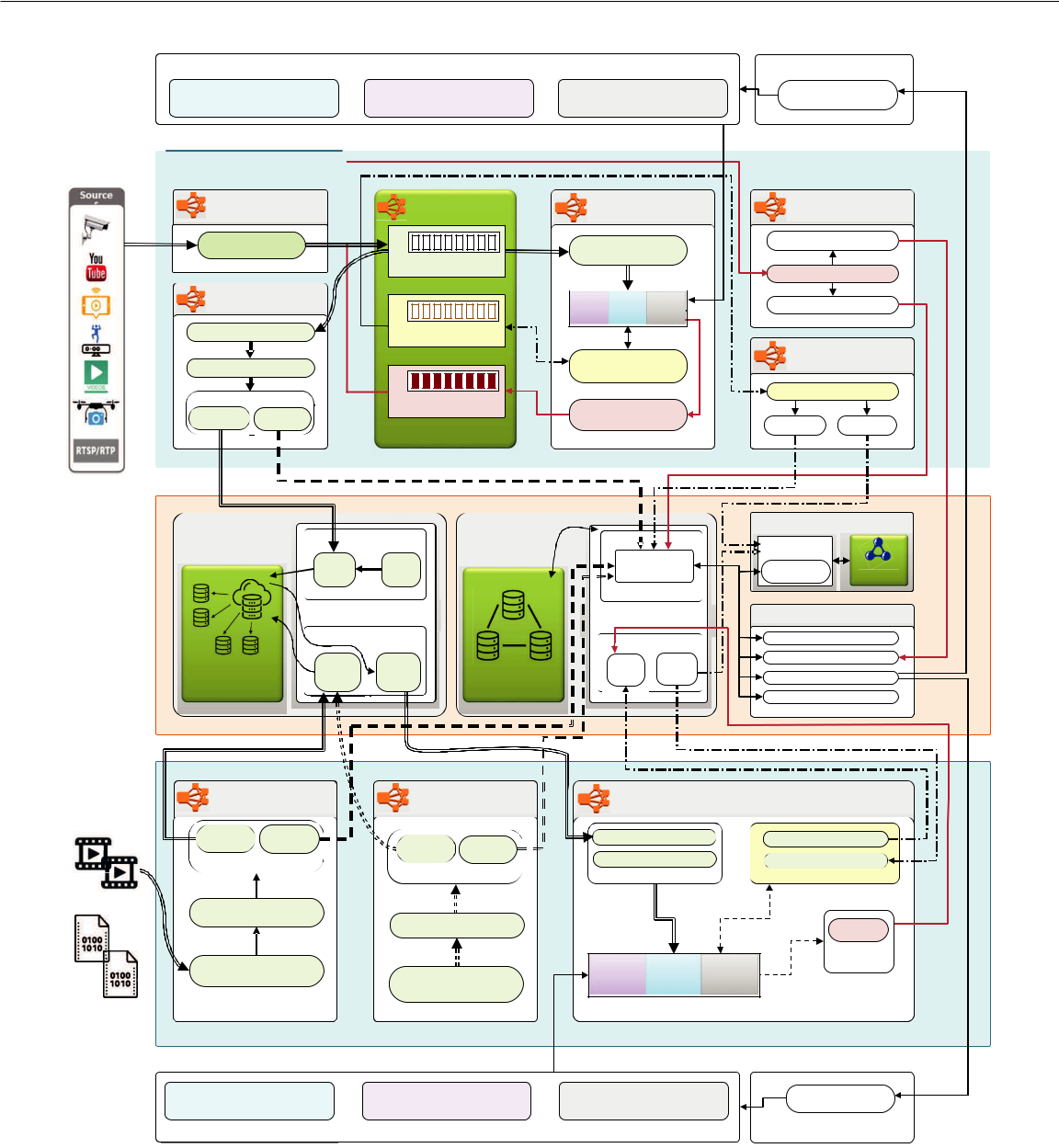
20 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | **Pool of contextual RIVA services** | | | | | | | | | | | | | **RIVA Subscription Manager** |  |
|  |  | Video Big Data Processing Layer | | | Video Big Data Mining Layer | | | | | | | | | |  | Knowledge Curation Layer | | RIVA Service |  |
|  |  |  | Contractor |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Load the subscribed RIVA services | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **Video Stream** | | |  | **Broker Cluster [K]** | | | | | | | | |  | **Video Stream Analytics** | | **Anomalies Notification** |  |
|  |  |  |  |  |
| sources | | **Acquisition Cluster [P]** |  | |  |  |  |  |  |  |  |  |  |  |  | **Consumer Cluster [S]** | | **Cluster [N]** |  |
| Video Stream | | |  |  |  |  |  |  |  |  |  |  |  | Video Stream | | Sent to Client User |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Acquisition Service | | | Queue: RIVA\_ID | | | | | | | | | |  | Consumer Service | |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| stream | |  |  |  | [Video Streams] | | | | | | | | | |  |  |  | LVSM Subscriber |  |
| **Video Stream** | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **Persistence Cluster [V]** | | |  |  |  |  |  |  |  |  |  |  |  | RIVAService | | Persist to ISDDS |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| video | Video Stream Consumer | |  | Queue: RIVA\_IR\_ID |  |  |  |  |  |
|  | [Int. Results] |  |  |  |  |  |
|  |  |  |  | **Intermediate Result** | | |  |
|  |  |  |  | Intermediate Results |  |
| Video Preprocessor | |  |  | **Persistence Cluster [I]** | | |  |
| time |  |  | Producer/Subscriber |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Sent for Persistence | |  | Queue: RIVA\_A\_ID |  | Int. Result Subscriber | | |  |
| Real- |  |  |  |  |  |  |
| Raw Video | Metadata |  | [Anomalies] | LVSM Producer | Persist to |  | Send to |  |
| Streams | **Distributed Messaging** | |  |  |  |
|  |  |  | ISDDS |  | Mapper |  |
|  |  |  |  | **System** |  |  |  |  |  |
|  | **Unstructured** |  |  | **Immediate** |  | **Knowledge Curation Server [T]** | | |  |
|  | **Persistent** | Active DRW | | **Structured** |  |  |  |  |  |
|  |  |  | **Distributed** |  | OntoMapper | |  |  |
|  | **Data Store** |  |  |  |  |  |
|  |  |  | **Data Store** |  |  |  |  |  |
|  |  |  |  | CRUD Operator | Concept |  | **(VidOnto)** |  |
|  |  | Create | Read | |  |  |
|  |  |  |  |  |  | based search | | **Triple store** |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | Active DRW |  |  |  |  |
|  |  | **DFS Active & Passive DRW** | | | **ISDDS Access Controller** |  | **Web Server [W]** | |  |
|  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Passive DRW | Passive DRW | User Management |  |
|  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Passive | Bulk | Data Source Manager |  |
|  |  | Scan |  |
|  |  | Data |  |
|  |  | Writer |  |
| **Distributed File** | Writer | Reader | (R/B)IVA Services |  |
| **Distributed** |  |  |
| **System** |  |  | **Structured DS** | Cluster Manage |  |

|  |
| --- |
| Active Execution Scenario (Speed Layer) |

|  |
| --- |
| Serving Layer |

|  |
| --- |
| Batch video data |

|  |  |  |  |
| --- | --- | --- | --- |
| **Raw Batch Video Data** | | | **Model Storage** |
| **Service Cluster [R]** | | | **Service Cluster [M]** |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Raw Video | Video File |  |  |  |
| Data | Metadata | Model | File |  |
|  |  | Metadata |  |
| Batch Video Data Persis. | |  |  |
|  |  |  |
|  |  | Model Persistence | |  |
| Videos Preprocessor | |  |  |  |
|  |  | File Meta Data Extractor | |  |
| Batch Video Acquisition | |  |  |  |
| Service (PDRW) | | Model Acquisition Service | |  |
|  |  |  |
|  |  | (PDRW) | |  |

Load the subscribed BIVA services

|  |  |
| --- | --- |
| Video Big Data Mining Layer | Video Big Data Mining Layer |

Pool of contextual BIVA services



**Batch Video Analytics**

**Cluster [B]**

|  |  |
| --- | --- |
| Bulk Video Data Loader | Write |
| Model Loader | Scan |

|  |  |  |
| --- | --- | --- |
| Data Load**er** | Int. **Result D**RW |  |
|  | Write |  |
|  | Anomalies |  |
| BIVA Service | Writer |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Knowledge Curation Layer | BIVA Service |  |
| Contractor |  |
|  |  |
|  | **BIVA Subscription Manager** |  |



|  |
| --- |
| Passive Execution Scenario (Batch Layer) |

Video Data Model Metadata Intermediate Anomalies

Learners Results

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

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

VOLUME 4, 2016

queue in the cluster ’K’. The second type of mod-ule is RIVA services. The RIVA service is the ac-tual 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 pro-ducer/subscriber is used to send and receive the IR according to the application logic to and from the IR

21

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

queue in Cluster ’K’.

The fourth type of module is LVSM producer. The contextual IVA services (IVAS) instance de-ployed in the cluster ’S’ should have some domain-specific goal and can produce anomalies if ana-lyzed any. The L-CVAS support real-time anoma-lies delivery system. The IVAS sent the anoma-lies 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 persis-tence. If the subscribed service is using the ontology then the IR are also mapped to the VidOnto triple re-siding in the Knowledge Curation Server ‘T’.

The final type of cluster in the Streaming Ex-ecution 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 deliv-ered 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 analyt-ics 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 up-load 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 Ser-vice, 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 lay-ers 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.

1. **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 [162], [163]. The CBVR is an active research area, and several surveys are available, i.e., [32], [163]–

1. However, here we discuss some of the scal-able CBVR system being proposed in the literature.

In literature, some researchers tried to exploit distributed computing technology for the develop-ment of large-scale CBVR systems. Shang et al.

1. utilize the time-oriented temporal structure of videos and the relative gray-level intensity distribu-tion 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 envi-ronment. Hence this approach is challenging to par-allelize accurately and efficiently even for the state-of-the-art big-data frameworks. Wang et al. [168] proposed a novel MapReduce framework called Multimedia and Intelligent Computing Cluster for near-duplicate video retrieval for large-scare mul-timedia 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 [169] to detect interest points. K-means clustering over the feature vectors is utilized to generate visual words following the BoF [170] model, thus generating BoF-based feature vectors. Ding et al. [171] used big data processing technolo-

22 VOLUME 4, 2016

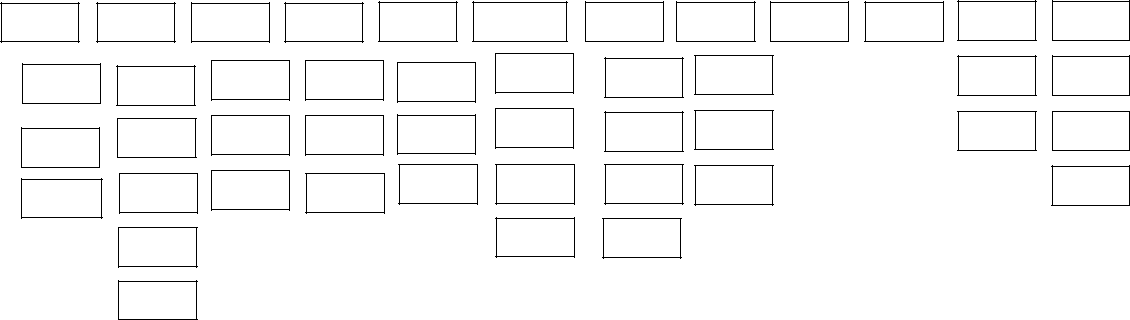
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  | | | Intelligent Video Analytics | | | | | | | | | | | | | | | | | |  | |  | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  | | |  | | | |  | | |  |  | | |  | |  |  | | | |  | |  | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | | | CBVR | | | | | |  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | Surveillance | | | | | |  | | | | | | | | | | Video | | | |  |  | Semantic | | | | |  |  |
|  | | | | | | | | | | | | |  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | and Security | | | | | |  | | | | | | | | | | Summarization | | | |  | Approaches | | | | | |  |  |
|  | | | | | | | | | | | | |  | | | | | |  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | | | | |  |  |  |
|  | | | |  | | | | | | | |  | | | | | | | | | |  | | | | | | |  | | | | | |  | | | | | |  | | | | | | |  | | | | | | |  | | | |  | | | |  |  | | | |  |  |  | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Similarity | | | |  |  |  | Relevance | | | |  |  |  | Action | | |  | Moving Object | | | | |  |  |  | Behavior | | | |  |  |  | Anomaly | | | |  |  | Inf. Fusion | |  |  | Handling | |  |  |  | Static | |  |  |  | Semantic | | | |  |
| Visualization | | | | |  |  |  |  | Query | | | | |  |  |  |  |  |  |  |  |  |  |  |  | classification & | | | | |  |  |  |  |  |  |  |  | from Multiple | |  |  |  |  |  |  |  |  |  |  | Based | |  |
|  |  |  |  |  | | |  |  | Measure | | | |  |  |  | Feedback | | | |  |  | Recognition | | | |  |  |  |  | Analysis | | | |  |  |  | Detection | | | |  |  |  |  | Occlusion | |  | Abstraction | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Detection | | | |  |  |  |  |  |  |  |  | cameras | |  |  |  |  |  |  | Retrieval | | | |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | |  | | | |  | | | | | |  | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | Object | | |  | |  | | Motion feat. | | | | |  | | Trajectory- | | | |  | | | | | |  | Dynamic | | | |  |  | | Low-level | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | |  | |  | |  | |  | | | | | |  |  |  | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | |  | |  | |  | |  | | | | | |  |  |  | |  |
|  |  | | | | | | |  | | | | | | | | | | | Feature | | | | | | |  | | | | | | | Similarity- | | | | |  |  |  | | representation | | | | | | |  |  |  | | | | | | Video | | | | | |  | |  |
|  |  |  | Graph based | | | |  |  |  |  | By example | | | | | | |  |  |  | Matching | | | |  |  |  | Explicitly FB | | | |  | based model | | | |  |  |  | Tracking | | |  |  |  |  | in crowded | | | | |  |  |  | based | | |  |  |  |  |  |  |  | Skimming | | | |  |  |  | analysis | | | |  |
|  |  | | | | | | |  | | |  |  | |  |  | | | | | | |  |  | | | | | | |  | | | scenes | | | |  | | | | | | | | | | | |  |  |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | |  | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | Implicit | | | |  | Depth-based | | | |  |  | | Motion | | |  | |  | | Flow-based | | | | | Global | | | | | | | | | | | |  | | | Video | |  |  | | Mid-level | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | |  |  |  |  |  | |  | |  | pattern- | | | | | | | | | | |  |  | |  |  | |  |
|  |  |  | Clustering | | | |  |  |  |  |  | Sketch | | | | | |  |  | Text based | | | | |  |  |  | Relevance | | | |  |  | approach | | |  |  |  | Detection | | |  |  |  |  | features | | | | |  |  |  | based | | |  |  |  |  |  |  |  | Synopsis | | | |  |  |  | analysis | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | based | | |  |  | | | | | | | | | | |  | | | | | | | FB | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | |  |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | |  |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | |  |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | | | | |  | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  | Deep leaning- | | |  |  | | Moving object | | |  | |  | | Local Spatio- | | | | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | | | | | | |  | | | | | | |  |  |  | |  | |  | |  | | | | | | | | | | | | | | | | | |  | | | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  | Combination | | | | |  |  | | | | | | |  |  |  | |  | |  | |  | Grid pattern- | | | | |  | | | | | | | | | | | |  | | High-level | | | |  |
|  |  | | | | | | |  | | | | | | | | | | |  |  |  | | | | | | |  |  |  | |  | |  | |  |  | | | | | | | | | | | |  | |  |
|  |  |  | Dimensionality | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pseudo- | | | |  |  | based motion | | |  |  |  | with stationary | | |  |  |  |  | temporal | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Object | | | | | | |  |  |  | based | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | based | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | analysis | | | |  |
|  |  |  | reduction | | | | |  |  |  |  |  |  |  |  |  |  | relevance FB | | | |  |  | recognition | | |  |  |  | camera | | |  |  |  |  | features | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | |  | | | | | | | | | | | | | matching | | | |  | | |  | |  |  | |  | |  | |  | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | based | | | |  | | | | | | | | | | | | |  | | | | | | | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | |  | | | | | | | | | | | | |  | | | | | | | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | Moving object | | |  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | |  | |  | | Trajectory | | | |  | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | |  | |  | |  | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | Keyword | | | | | | | |  | | | | | | | | | | | | | | | | | | |  | | with moving | | | | |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | |  |  | | | | | | | | | | | | | | | | | | | | | camera | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | |  | | | | | | | | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | |  | | Natural | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
| Language | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |



**FIGURE 14.** IVA taxonomy based on literature.

gies to design a human retrieval system on extensive surveillance video data called SurvSurf. Motion-based segments called M-clop were detected, which 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 SurvSurf outperforms the baseline solutions in terms of computational time and with acceptable human retrieval accuracy.

Authors in [172] 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 through-put. 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 ad-dressed by Lv et al. [173] 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

VOLUME 4, 2016

the system in a distributed environment based on Apache Spark. Further, M. N. Khan et al. [174] proposed FALKON for large-scale CBVR that uti-lized 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 struc-ture, 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. Fi-nally, they evaluate their proposed system and show performance improvement in terms of scalability and accuracy. Likewise, Lin FC. et al. [175] 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 sys-tem detects a new person, it performs enrolling that person. Finally, timestamped 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

23

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

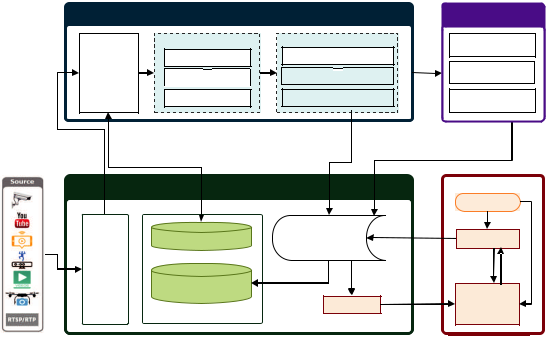
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



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 ex-tracts the low-level features. These low-level fea-tures can be keyframes’ static features (texture-based, color-based, and shape-based), object fea-tures, 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 dimen-sional video feature vectors’ indexes make the rep-resentative index for persisted video sequences in IR-DS. The IR-DS is synchronized with RAW-DS.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| VideoStreamMini-batches |  | **BVDPL** |  |  | **BVDML** |  |
| Video) | Scenes | KeyFrame Features |  | Clustering |  |
|  | Videos | Pre-processing | Feature Extractor |  | Classification |  |
|  |  |  |  |  |
|  | (Mini- | Shots | Object Features |  | Regression |  |
|  | batches/ | KeyFrames | Dynamic Features |  |  |
|  | Batch |  |  |  |
|  |  |  |  |  |  |
|  | Mini-batches persistence / | |  |  | Semantic |  |
|  | Batch video read | |  |  |  |
|  |  |  | indexing |  |
|  |  | High dimensional | |  |  |
|  |  | feature indexing | | Similaritymeasure |  |  |
|  |  | **BVDCF** |  | WSL |  |
|  |  |  |  |  | USERS |  |
|  |  |  | ISB Active & |  |  |  |
|  |  | RAW DS | Passive DRW | RankedResult-set | Query |  |
|  |  |  | (Index) |  |  |
|  | VSAS |  |  | Feedback |  |
|  | Result DS |  | Browsing |  |
|  |  |  |  |
|  |  | Intermediate |  |  |  |  |
|  |  |  | Rank |  | Query |  |
|  |  | UPDS & ISDDS |  |  | Results |  |

**FIGURE 15.** Components of CBVR under the Lambda CVASarchitecture.

In the fourth step, through the WSL, the users are allowed to query the desired videos. In liter-ature 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 theQuery by Example, simi-lar 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 [176]. 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 [177]. Likewise, the last two approaches use Keywords and Natural Language [178] as query. Combination-based Query combines different types of queries for

A Alam et al.: Video Big Data Analytics in the cloud

CBVR, was also adopted by some researchers such as [179], [180].

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 similar-ity measure is the average distance between the features of the corresponding frames [181]. Query uses 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. [182]. 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 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 [183] but can also decline search results unexpectedly. This approached is further ex-plained in section V-D. Combination-based matching “leverages semantic concepts from a training collection by learning the combination strategies [87], and query-class-dependent combi-nation models [180]” [89]. 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 feed-back from the user. This feedback can be cate-gorized as explicit, implicit, and pseudo feedback. In the first case, users are asked to select relevant videos from the previous results actively [184]. The explicit feedback ap-proach 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 [185]. In the third case, there is no involvement of the users. The user’s feedback is produced through positive (closely related to the

24 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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. [186], and Hauptmann et al. [187] approaches are based on pseudo-relevance feedback approach. This approach substantially re-duces 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.

1. **REAL-TIME INTELLIGENT VIDEO ANALYSIS AND SURVEILLANCE**

In the context of growing security concerns, the surveillance meant to criminality and intrusion de-tection. Video surveillance is not limited to these, but it encapsulates all aspects of monitoring to cap-ture the dynamics of diverse application areas, e.g., transportation, healthcare, retail, and service indus-tries. 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 [188]. 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 [189]. 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 [190] suggested a methodology to extract key segments automatically for event detec-tion 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 Rela-tion Model and Segment-Event Interaction Model, for the temporal relations between video segments,

and for evaluating the correlation between key seg-ments and events respectively. They adapted labeled videos and images from the Web into their model and employed ’N’ adjacent point sample consensus

1. for noise elimination. A knowledge-base was produced by Zhang et al. [192] to decrease the semantic gap between complex events. To effec-tively 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 per-forming both tasks parallel. [189], [193] are some of the good examples of video segmentation based event detection. The related literature on video seg-mentation based action recognition can be classi-fied as action segmentation, a depth-based approach, and deep learning-based mo-tion recognition.

The most popular action segmentation model is dynamic time warping scheme [194]. Action segmentation from videos using an appearance-based method considers a comparison between the start and end frames of contiguous actions. Quantity of movement [195] and KNN along with HOG

1. are commonly used to identify the start and stop frames of actions. Similarly, for action recog-nition, depth-based approaches have been developed which consist of binary range-sample feature [197], capturing local motion and geometry information [198], histogram oriented 4D normal [199], and combination of depth motion map and HOG [200]. [201]–[203] have applied deep learn-ing 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 [97], [117]. Sometimes video as compact images are given as input to the already trained CNN to achieve good performance [204]. 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 [205], [206]. In order to introduce a temporal dimension, video is regarded as a volume and substitutes the 2D filters of CNN with 3D filters [207], [208].

In many application areas, e.g., resource tracking, action recognition, human behavior recognition, and traffic control, object tracking and motion detec-tion in videos are vital. The process of detecting

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 25 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

moving objects in videos is known as object track-ing. 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 [209], [210]. To increase the accuracy, subsequently, shadow elimination algorithms are applied to the foreground frame [211]. 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 adap-tive mean shift [212] 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 com-ponent algorithm for noise removal. The area of detection is refined using a connected component algorithm and produces the bounding box infor-mation of the moving objects [209]. 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 [213]. 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.

1. Moving object detection with stationary camera

In fixed camera video, the background image pix-els in each frame remain the same, and thus sim-ple background subtraction techniques are required. The object detection approaches using a fixed cam-era can be grouped as feature-based, motion-based, classifier-based, and template-based models [214]. Categorization of object tracking in videos into point tracking, kernel tracking, and silhouette track-ing as well as feature-based, region-based, and contour-based was performed by [214], [215] re-spectively. Unlike a fixed camera, moving object de-tection with a moving camera is relatively challeng-ing because of camera motion and background mod-eling for generating foreground and background pixel fails [216].

Trajectory classification involves comput-ing long trajectories for feature point and dis-criminating trajectories that belong to differ-ent objects from those backgrounds using the clustering method. Some recommended algo-

rithms include compensating long term motion based on flow optic technique [217], bag-of word classifier, and pre-trained CNN method for detecting moving object trajectories [218].

In background modeling based meth-ods, for each sequence, the frame-by-frame background is created utilizing the motion compensation method. Some popular algo-rithms are Mixture of Gaussian [219], complex homography [220], gaussian-based method [221], adaptiveMoG [222], multi-layer ho-mography transform [223], thresholding [224], and CNN-based method [225] for background modeling.

In extension of background subtraction method, low rank and sparse matrix decom-position method for static camera [226] 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 decomposi-tion 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 [227].

1. Behavior Analysis

Usually, a camera is mounted nearby the digital displays to analyze and understand human behavior by investigating user interfacing with digital display

1. Commercial tools have been developed to an-alyze audience behavior using video analytics while considering parameters like age, gender, distance from the display, and sight and spent time. The ob-tained data can then be used to improve advertising campaigns in combination with sales data [229].

Recently, crowd analytic, i.e., human detection and tracking, have attracted the attention of the researches. The exploration of both group and in-dividual behavior to govern abnormality scope the crowd analysis. Congestion analysis, motion detec-tion, tracking, and behavior analysis are the main at-tributes 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, mo-tion features are vital, and can be catego-rized as flow-based features, local

26 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

spatiotemporal features, and trajectory features [230]. These features have applications in crowd behavior recognition, abnormality detec-tion 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 pro-posed [231], [232], 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 mo-tion. It has been applied to detection crowd mo-tion and crowd segmentation [233]. This approach, optical flow, is unable to capture spatiotemporal attributes of the flow and long-range dependencies. Particle flow contains moving a grid of par-ticles with the optical flow and providing trajec-tories that maps a particle’s initial position to its future or current position. It has an application in crowd segmentation and detection of abnormal be-havior [230]. Optical flow is unable to handle spatial changes. Mehran et al. [232] 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 fea-tures 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 be-havior, Kratz and Nishino [234] used a spatiotem-poral motion pattern model and confirmed the de-tection of abnormal activities. On the other hand, motion histogram considers motion informa-tion 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., [231], [235].

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 spatiotem-poral 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], [236] and for the representation of motion in crowded scenes [237], [238].

3) Anomaly Detection

Anomaly detection an application area of crowd be-havior 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.

1. 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 [239]. For the evaluation of abnormality in trajectory-based methods have been used i.e., single-class SVM [240], zone-based analysis [241], semantic tracking [242], String kernels clustering [243], Spatiotemporal path search [244], and deep learning-based approach [245] have been used.

1. Global pattern-based method of anomaly detection

In a global pattern-based technique, the video se-quence is analyzed in whole, i.e. low, or medium-level features are extracted from video using Spa-tiotemporal gradients or optical flow methods [246]. 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 Mix-ture Model [247], energy model [248], SFM [249], stationary-map [250], Gaussian regression [251], PCA model [252], global motion-map [253], motion influence map [254], and salient motion map [255].

1. Grid pattern-based method of anomaly detection

In a grid pattern-based method, splits frames into blocks and individually analyze pattern on a block-

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 27 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

level basis [256]. If ignoring inter-object connec-tions that lead to processing efficiency. Spatiotem-poral anomaly maps, local features probabilistic framework, joint sparsity model, mixtures of dy-namic 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 [257].

**C. VIDEO SUMMARIZATION**

Video big data are facing the challenge of spar-sity 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 summa-rization. Video summarization is a process of gen-erating a shorter video of the original one without spoiling the capability to comprehend the meaning of the whole video [258]. The video summarization can be classified as Static Video Abstracts, Dynamic Video Skimming, and Video Synopsis.

a: Static Video Abstract

These approaches include a video table of contents, a storyboard, and a pictorial video summary. For example, Xie and Wu [259] 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. [260] 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. [261] proposed a summarization technique where they detect keyframes from each shot that eliminates re-dundancy 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. [262] propose a subset selection technique that leverages supervision in the form of human-created summaries to perform auto-matic 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 anno-tated and test videos. Then the summary structures are encoded in the annotated videos with kernel ma-trices made of binarized pairwise similarity among their frames. Those structures are then combined into a kernel matrix that encodes the summary struc-ture for the test video. Finally, the summary is de-coded by feeding the kernel matrix to a probabilistic model called the determinantal point process to ex-tract a globally optimal subset of frames. M. Gygli et al. [263] used a supervised approach to learn the importance of the global characteristics in summary by extracting deep features of video frames. J. Mo-han et al. [264] 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 vec-tors 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. [265] 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 fea-tures, which constrained the sparse auto-encoder to select the candidate keyframes.

b: Dynamic Video Skimming

A summary video is formed from the video seg-ments 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. [266] 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 con-sumer video programs. Ngo et al. [267] 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. [268] mine frequent patterns from a video. A video shot importance evaluation model is utilized to choose useful video shots to create a video sum-mary. For personal videos, Gao et al. [269] devel-

28 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

oped a video summarization technique, which en-compasses a two-level redundancy detection proce-dure. 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 termi-nate 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 [270], for user-generated video summarization, both the represen-tativeness 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 seg-ments that include consistent semantics and emo-tions 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. [271] 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 se-lecting the human keyshots. The proposed frame-work was called attentive encoder-decoder networks for video summarization. They utilized the BiL-STM encoder for encoding the contextual infor-mation among the input video frames. For the de-coder, two attention-based LSTM networks, are ex-plored by using additive and multiplicative objec-tive functions, respectively. The results demonstrate the superiority of the proposed framework against the state-of-the-art approaches, with remarkable im-provements. J. Wu et al. [272] were motivated by the fact that multi-video summarization is signifi-cant for video browsing and proposed a technique where multi-video summarization was formulated as a graph problem. They also introduced a dy-namic 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 reg-ularization to encourage the model to generate a di-

verse summary was introduced. The results demon-strate the effectiveness of our proposed model in generating a representative summary for multiple videos with encouraging diversity. Z. Sheng-hua et al. [273] 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 im-balance 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 infor-mation of the input video. Moreover, they stated that the proposed method highlights the video con-tent with the active level of arousal in effective computing tasks and can automatically preserve the connection between consecutive frames.

c: 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 [274] 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 readjust-ment, background formation, and object stitching. A global energy function was defined and encom-passing activity, temporal consistency, and collision cost. Then the simulated annealing method was ap-plied for energy minimization. The video synopsis domain was further researched in single and multi-view scenarios. Some recent examples of a single-view are [275]–[277]. He et al. [275], [276] brought advancement in activity collision analysis by de-scribing 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 colli-sion states to improve the activity density and put activity collisions at the center of their optimization strategy. Baskurt and Samet [277] concentrated on rising robustness of object detection by suggesting an adaptive background generation method. In an-other study, Baskurt and Samet [278] planned the object tracking method specified for video synop-sis requirements. Their approach focused on long

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 29 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

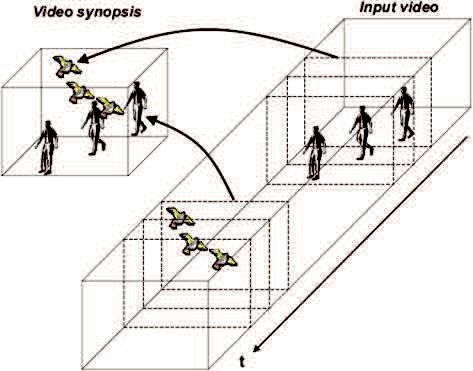
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

term tracking to represent each target with just one activity in video synopsis. Single view scalable ap-proaches for video synopsis were projected by Lin et al. [279] while utilizing distributed computing tech-nology. Their proposed video synopsis approach encompasses steps like object detection, tracking, classification, and optimization, which were per-formed in a distributed environment. Ahmed, S. A.

1. 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 spatiotem-poral object trajectories. Finally, user queries were allowed to generate video synopsis by smoothly blending the appropriate tubes over the background frame through energy minimization.



**FIGURE 16.** Simultaneous activities display in a video [274].

Further, the examples of multi-view video an-alytics are that of [281], [282]. Zhu et al. [281] 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 bal-ance among minimizing the synopsis length, maxi-mizing the information coverage, and reducing the collision among objects’ track that are presented concurrently. Likewise, Mahapatra et al. [282] pro-posed a multi-camera approach for an overlap-ping 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. [283], 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 con-structed a simultaneous object-shifting and view-switching optimization framework to achieve en-couraging synopsis results. To address unified op-timization, they further presented an optimization strategy composed of graph cuts and dynamic pro-gramming.

1) Video summarization aaS under L-CVAS

Fig. 17 shows the flow of the video summariza-tion flow under the proposed L-CVAS architecture. Through the WSL, the user first subscribes to the video data-source to the video summarization ser-vice. Then the user preferences allow the users to set the parameters required for video summary service and personalization. The summary parame-ters encompass granularity level, type of summary to be performed (e.g., overview, highlights, syn-opsis, etc.) and any other as per the video sum-mary 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 require-ments of the video summarization service. Then multiple low and high-level features, such as mo-tion, 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.

30 VOLUME 4, 2016

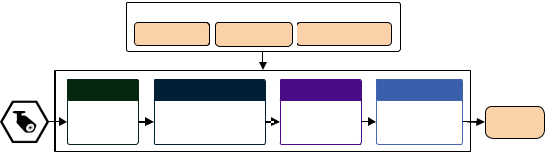
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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 sum-mary 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.



**Web Services Layer**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Video data- | User |  | Summarization | |  |  |
|  | source | Preferences | | service |  |  |  |
| **VBDCL** | **VBDPL** | |  | **VBDML** | **Vid. Sum.** |  |  |
| RVSAS | Pre-processing | |  | Clustering | - Unit selection | Video |  |
|  | - Redundancy |  |
| RAW VIDEO | Feature Extraction | | Classification | | Summary |  |
| removal |  |
|  |  |  |  |  |  |  |

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

**D. SEMANTIC-BASED VIDEO ANALYSIS**

To bridge the semantic gap and to allow the ma-chine to understand the visual data, Semantic Web technologies can be incorporated [161]. Seman-tic Web unlocked a new avenue for knowledge-based computer vision while enabling data ex-change between video analysis systems in an open and extensible manner. The scientific community has exploited the Semantic Web concept for in-telligent 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 re-trieval.

1) Semantic-based Low-level analysis

Semantic-based low-level analysis refers to the for-malization of the extracted objects of interest from the videos. It then performs reasoning on formaliza-tion such as detection and tracking across domain-specific videos [284], [285]. Dasiopoulou et al.

1. proposed a multimedia ontology for domain-specific video analysis. In semantic concepts, they consider object attributed, low-level features, spa-tial object relations, and the processing approaches while defining F-logic rules for reasoning that gov-ern the application of analysis methods. García et al. [287] proposed a knowledge-based framework for video object segmentation, where relationships among analysis phases are utilized. The main con-tribution 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 con-textual 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 hi-erarchies of the analysis ontology. The quality of the results was evaluated through a feedback path. Gomez et al. [288] proposed a computer vision framework for surveillance and consists of two lay-ers, 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 observa-tion. 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 detec-tion, 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

1. 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 [290], and “complex” events such as aggressions, fights, thefts, and other general suspicious events [291],

1. falls in mid and high-level video analytics, respectively.

The utilization of semantic technology for video event representation in surveillance videos was ini-tiated by Video Event Representation Language

1. The main idea is to model simple events in a hierarchical framework intended for detecting

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 31 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

complex events. They stated that the sequence of simple events (car door opening, leaving a car, car door closing, walking and opening a build-ing 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.

1. 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 transi-tive 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 tran-sitive events. The focus was on complex events, which can be achieved by sequencing simple events. SanMiguel et al. [295] propose an ontology for rep-resenting 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. [296] proposed a hybrid approach for simple abnormal (person falling) and complex ab-normal events (person aggression) recognition using semantic web technologies. They modeled the ex-tracted general tracking information to the proposed tracking ontology for advanced reasoning. The data from the videos were obtained using the track-ing component (frames, bounding box), knowledge about the scene (static and dynamic objects, occlud-ing objects), Situations and Events (people leaving the scene, falling ground, fighting). For event de-tection, 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 recog-nition in daily living. In this context, Chen et al.

1. introduced a method for activity recognition while using ontological model, representation, and reasoning. They analyzed the nature and character-

istics 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. Like-wise, [298], an approach exploiting the synergy be-tween the semantic technologies and tracking meth-ods 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 sta-tionary 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 recog-nize 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 re-gard, Yao et al. [299] proposed the image to text framework to extract events from images (or video frames) and then provide semantic and text annota-tions. 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 descrip-tion. 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.

1. 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 fea-ture 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 [301] and [302] propose a method to annotate video traffic events while con-

32 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

sidering 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 re-sources with their semantic relations), and demands layer (retrieval interface). They defined various con-cepts 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. [303] 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. [304] and proposed an advanced intelligent forensic retrieval system by taking ad-vantage of an ontological knowledge representation while considering the UK riots in 2011 as a use case. Similarly, A. Alam et al. [305] proposed a layered architecture for large-scale distributed intel-ligent 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 on-tology, 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 rela-tionship 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 technolo-gies. In this direction, Ajiboye, S.O. et al. [306] stated that the network video recorder is already equipped with intelligent video processing capabil-

ities but complained about its limitations, i.e., iso-lation, and scalability. To resolve such issues, they proposed a general high-level theoretical architec-ture called Fused Video Surveillance Architecture (FVSA). The design goals of the FVSA were cost reduction, unify data mining, public safety, and scal-able IVA. The FVSA architecture consists of four-layer, i.e., Application layer (responsible for sys-tem 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. [307] implemented a prototype of a cloud-based video recorder system under glsIaaS 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. [308] also came out with a cloud platform for large scale video analytics and management. They stated that the existing work failed to design a ver-satile 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. [106] 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. [309] proposed a cloud video analytics framework using HDFS and MapReduce along with OpenCV [310] and FFm-peg 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. [311] 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 sat-isfy 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

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 33 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

99% bandwidth compared to a centralized cloud-only based approach.

White et al. [312] researched MapReduce for IVA services, which comprises classifier training, clus-tering, sliding windows, bag-of-features, image reg-istration, and background subtraction. However, ex-periments were performed for the k-means cluster-ing and Gaussian background subtraction only. Tan and Chen [106] presented face detection, motion de-tection, and tracking using MapReduce-based clus-ters on Apache Hadoop. They utilized JavaCV since Hadoop is developed and designed for Java. Pereira, R. el al. [313] 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 synchro-nization, 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. [314] used Hadoop and MapReduce for video sharing and transcoding purposes.

Zhang, W. et al. [52] proposed a cloud-based architecture for large scale intelligent video analyt-ics called BiF. BiF combines the merits of RIVA and BIVA while exploiting distributed technologies like storm and MapReduce, respectively. BiF ar-chitecture considered non-functional architectural properties and constraints, i.e., usability, scalabil-ity, reliability, fault tolerance, data immutability, re-computation, storing large objects, batch process-ing 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 algo-rithm, which was able to detect and count faces for a specific interval of time from the input source. Dur-ing the evaluation, they showed that BiF is efficient in terms of scalability and fault tolerance. Zhang et al. [320] 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. [319] 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.

1. proposed a novel feature descriptor to recog-nize human action on Spark while utilized the Spark MLlib [143] to recognize the action from the fea-ture vector generated by ALMD [321]. Wang et al.
2. also performed human action recognition on Spark. The aim was to speed up some key processes, including trajectory-based feature extraction, Gaus-sian Mixture Model generation, and Fisher Vec-tor encoding. Distributed video processing called streaming video engine is also introduced in [323] 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. [317], [318] 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 hander 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 knowl-edge 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 tol-erance, and scalability. Zhang et al. [324] 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.

34 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ref |  |  | IVA | VDM | | Core Technology | | | Performance | (a)Objectives |  |
|  | R | B | Service | St | S | Storage | DP | Lib. | Matrix | (b)Target Domain (c) Issues |  |
|  |  |  |  |  |  |  |  |  | Workload for | (a) CVAS Architecture, |  |
| [315] | 7 | 3 | -Face detection | 7 | 7 | MS SQL | 7 | OpenCV | - Face detection | Implements a prototype |  |
| -Motion detection | Server | - Storage avg. | (b) NA |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | task time | (c) Not scalability and SOA |  |
|  |  |  | - Bg. Subtract |  |  |  |  |  |  | (a) CVAS Architecture, |  |
|  |  |  |  |  |  |  |  |  | Implements a prototype. |  |
|  |  |  | -Object |  |  |  |  |  |  |  |
| [316] | 7 | 3 | 3 | 7 | - | - | - | 7 | (b) Surveillance |  |
| classification |  |
|  |  |  |  |  |  |  |  |  | (c) Not scalable, consumer |  |
|  |  |  | -Object tracking |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | only, no evaluation. |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | (a) Merge-Split Based |  |
| [313] | 3 | 7 | Video | 7 | 3 | HDFS | MR | - | Scalability | Video encoding. |  |
| compression | (b) Storage and transmission. |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | (c) Video compression only. |  |
|  |  |  |  |  |  |  |  |  |  | (a) Proposed video surveillance |  |
|  |  |  |  |  |  |  |  |  |  | architecture for video analytics |  |
| [306] | - | 3 | - | 3 | 3 | - | - | - | - | (b) Surveillance |  |
|  |  |  |  |  |  |  |  |  |  | (c) Based on B2C models, and |  |
|  |  |  |  |  |  |  |  |  |  | not SOA. |  |
|  |  |  |  |  |  |  |  |  |  | (a) Cloud-based architecture for |  |
| [307] | 3 | x | - | 3 | 3 | HDFS, | - | - | -Bandwidth usage | video recordings. |  |
| Hbase | -Scalability | (b) Surveillance |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | (c) IVA not supported |  |
|  |  |  | -Video summary |  |  |  |  |  |  | (a) Distributed video management |  |
|  |  |  |  |  |  |  |  | -Throughput | platform. |  |
|  |  |  | - Encoding |  |  | HDFS, |  | OpenCV, |  |
| [308] | 3 | 7 | 3 | 3 | MR | -Average I/O rate | (b) Distributed video management. |  |
| - Decoding | MySQL | CUDA |  |
|  |  |  |  |  |  | -Execution Time | (c) Not SOA, video stream not |  |
|  |  |  | - Bg. Subtract |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | scalable. |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | (a) Fast and distributed video |  |
|  |  |  | -Face Recognition, |  |  |  |  | OpenCV, |  | processing. |  |
| [106] | 3 | 3 | -Motion Detection | 7 | 3 | HDFS | MR | FFMPEG | -Scalability | (b) Video surveillance. |  |
|  |  |  | -Tracking |  |  |  |  | JavaCV |  | (c) Distributed video processing |  |
|  |  |  |  |  |  |  |  |  |  | only. |  |
|  |  |  | Object recognition |  |  | HDFS, |  |  | [8] -Accuracy | (a) Proposed a cloud platform for |  |
|  |  |  |  |  |  |  | Recognition | online and offline video analytics. |  |
| [317] |  |  | and tracking |  |  | MR |  |  |
| 3 | 3 | 3 | 3 | HBase, | OpenCV | [9] -Fault | (b) Security, surveillance. and |  |
| [318] | vehicle | Storm |  |
|  |  |  |  | SploutSQL |  | Tolerance, | traffic. |  |
|  |  |  | classification |  |  |  |  |  |
|  |  |  |  |  |  |  |  | -Scalability | (c) Not SOA, |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | (a) General cloud architecture for |  |
|  |  |  | Face recognition |  |  | HDFS, | MR |  | -Scalability | Video Analytics. |  |
| [52] | 3 | 3 | 3 | 3 | OpenCV | (b) Security and transportation |  |
| & counting | Splout-SQL | Storm | -Fault tolerance |  |
|  |  |  |  |  |  | surveillance |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | (c) Not SOA |  |
| [319] | 3 | 3 | -Face detection | 3 | 3 | HDFS, | MR | OpenCV | -Scalability | (a) CVAS Platform |  |
| -Action recognition | Splout-SQL | Storm | -Fault tolerance | (b) Security and surveillance |  |
|  |  |  |  |  |  |  |

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

**B. INDUSTRIAL CVAS**

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

a: Google Vision

On March 8, 2017, at the Google Cloud Next con-ference in San Francisco, Google announced the release of the IVA REST API [328]. 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 informa-tion about entities (20,000 labels). Its main fea-tures are label detection, explicit content detection, shot change detection, and regionalization [331]. 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.

b: IBM CVAS

In April 2017 at the National Association of Broad-casters Show, IBM announced CVAS services [332]. BlueChasm, [329] development team, came up with a prototype app, known as “VideoRecon,” that com-bines IVA via IBM Watson and IBM cloud stack. The IBM CVAS service can extract metadata like keywords, concepts, visual imagery, tone, and emo-tional context from video data. The IBM CVAS

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 35 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

**TABLE 10.** Industrial CVAS.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CVAS | Target Domain | RIVA Services | BIVA Services | BM | API |  |
| Citilog | Security & Surveillance | Incident detection, traffic |  |  |  |  |
| (Transportation, Highways, | data collection, interaction |  | B2B, |  |  |
| (1997) | NA | 3 |  |
| Tunnels, Bridges, and smart | control, license plate | B2C |  |
| [325] |  |  |  |
| cities). | recognition |  |  |  |  |
|  |  |  |  |  |
| Check- | Security & Surveillance | Basic Analytics (motion detection, |  |  |  |  |
| zone violation, camera tamper) |  |  |  |  |
| Video | (education, guard, airports, |  |  |  |  |
| Object classification Business | NA | B2C | 7 |  |
| (1998) | retail, construction, services, |  |
| Analytics (object counting, tracks, |  |  |  |  |
| [326] | residential, outdoor assets) |  |  |  |  |
| transaction) |  |  |  |  |
|  |  |  |  |  |  |
| Intelli- | Security & Surveillance | Smart home, Smart security, |  |  |  |  |
| Vision | Video Search, video summary | B2B, |  |  |
| (home & IoT, retail, traffic | Smart retail, Smart city, | 3 |  |
| (2002) | and, recognition | B2C |  |
| & transportation) | and ADAS |  |  |
| [327] |  |  |  |  |
|  |  |  |  |  |  |
| Google |  |  | Label Detection (detect objects), |  |  |  |
| Vision | Video contents & analytics | NA | Content Detection (Shot change | B2C, | 3 |  |
| (2017) | and adult contents), detection (detect | B2B |  |
|  |  |  |  |
| [328] |  |  | scene changes) |  |  |  |
| IBM |  |  |  |  |  |  |
| CVAS | Video contents & analytics | NA | Deep learning based IVA | B2B | 3 |  |
| (2017) | (scenes, objects, faces) | C2B |  |
|  |  |  |  |
| [329] |  |  |  |  |  |  |
| Azure | Video contents analytics & | Motion detection, Face detection, | HyperLapse, summarization, character | B2B |  |  |
| CVAS | Emotion detection (sadness, fear, | recognition, content moderation, | 3 |  |
| surveillance | C2B |  |
| [330] | anger) | face redactor |  |  |
|  |  |  |  |

allows the users to upload video footage to the IBM Cleversafe object storage [333] 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 ei-ther the object was recognized or the event occurred. The tags are then stored in the IBM Cloudant fully managed NoSQL JSON document store [334] for future use.

c: 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 [330]. 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).

d: Citilog CVAS

Citilog [325], also known as CT-Cloud, provides intelligent video analytics and surveillance solutions in the domain of transportation. Citilog provides ser-vices like automatic incident detection, traffic data collection (vehicle counting, classification, average speed, occupancy and levels of service), interaction control, video management, and license plate recog-nition. The Citilog is an open platform, providing APIs, widgets for quick development of services. According to the Citilog, they process approxi-

mately 32000 hours of video data and detects about five incidents per minute from 900 sites worldwide.

e: Check-Video CVAS

Check-Video [326], 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 in-telligent video analytics RIVA solution. The main features Check-Video are RIVA, a video search en-gine, cloud video storage, and an alert system. The provided services can be categorized as basic analyt-ics, 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.

f: Intelli-Vision CVAS

Intelli-Vision [327], founded in 2002, is a leading and notable company in the field of Artificial Intel-ligence (AI) and deep learning-based video analyt-ics 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 sys-tems for cars. Intelli-Vision’s analytics adds the “Brains Behind the Eyes” for cameras by analyzing the video content, extracting meta-data, sending out

36 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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 suc-cessfully deployed four million cameras worldwide, which have been subscribed to various IVA services.

**VII. IVA APPLICATIONS**

IVA at scale drives many application domains rang-ing from security and surveillance to self-driving and healthcare. Many application areas of video big data analytics are shown, which project the signifi-cant 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 [335]. Proactive analytics is re-quired to predict abnormal events so that to min-imize or avoid such accidents. In this direction, VisonZero [336] 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 [337] is a vehicle tracking system and uses informa-tion from various non-overlapping cameras to de-tect vehicle path. Gao et al. [338]used an auto-matic 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 probabilisti-cally. Incident detection framework based on gener-ative adversarial networks were proposed in [339].

1. Intelligent Vehicle and Self-driving cars

Currently, the term self-driving cars mean that the vehicle exploits computer vision for safe and intel-ligent driving while assisting the driver. In an intel-ligent vehicle, different sensors and high definition cameras (cameras for vehicle cabin, forward road-way, 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 [340], [341]. 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. [342]

proposed a method for pedestrian detection in urban traffic conditions using a multilayer laser sensor mounted onboard a vehicle. An algorithm was pro-posed by Tsai et al. [343] 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 [344] 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 [345]. 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 predica-tion, have been studied in [346], [347] respectively.

c: HealthCare

Recently, video big data analytics is reshaping the healthcare 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 in-stalled in a home for assisted living for several months and shows quite promising performance

1. 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 [349]. Some more studies were done [350]–
2. 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 enter-prises to schools, home and law enforcement agen-cies 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 cus-tomized services that can adjust quickly to changing needs and demands. People detection, and tracking [354], motion detection, intrusion detection [355], line crossing [356], object lift, loitering [357], and

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 37 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

license plate recognition [358] are the example of video analytics services for security. For security in subway stations, Krausz [359] developed as a surveillance system to detect dangerous events. Shih et al. [360] 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, aban-doned object detection is indispensable and can lead to a terrorist attack. In the future, video analytics ap-plications are developing fast, and they are changing the way the security industry works.

1. 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 com-plex 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.

1. Retail, Management, and Business Intelligence Analysis

Large-scale products, services, and staff manage-ment while adjusting to consumer demands can be challenging without timely and up-to-date informa-tion. 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 matrics, consumer traffic map, are some of the example services in this context [361]. Gaze analysis provides a means to learn customers’ in-terest in merchandise by following their attention [362], [363] on a store display. Actions like reaching or grabbing products were analyzed by [364], [365] to understand customers’ interest. Emotion analysis can identify customers’ views regarding product and interaction with the company’s representative

1. 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 infras-tructure. 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 oppor-tunities. This section provides in-depth detail about such research challenges, which has been summa-rized 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. Consequently, video big data analytics is challenging over such engines and demand opti-mization. Almost all the engines are inherently lack-ing 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 pro-cessing time of each iteration [366]. By default, Hadoop lacks iterative job support but can be han-dled 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 exiting 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 ac-cessing 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. Like-wise, users’ assistance is required to comprehend IVA results and the reason behind the decision to realize the value of video big data in decision sup-port. Thus, understandable IVA can be a significant future research area.

38 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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

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

**B. IVA AND HUMAN-MACHINE COORDINATION**

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

IVA on video big data in cloud demands re-searchers and practitioners mastering both IVA and distributed computing technologies. Bridg-ing both the worlds for most analysts is chal-lenging. Especially in an educational environ-ment, 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 ab-stractions to hide the underlying complexity.

IVS service to become commercially worth-while 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 pas-sive. 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 abil-ities, behaviors, and requirements [367]. The interactive IVA services will learn from the consumer and decrease the need for admin-

istration by a specialist. It will also enable consumers to design custom IVA services to meet the domain-specific requirement.

1. **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 middle-ware (Table 2) and distributed processing engines (Table 7). In L-CVAS, the dynamic (R/B)IVA ser-vice 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 [368], [369]. 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

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 39 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

Cyclic Dependency Graph (CCDG).

In the map-reduce infrastructure, a slowdown pre-dictor can be utilized to improve the agility and timeliness of scheduling decisions [370]. 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 subscrip-tion environment. Further, concepts from the field of query and queuing optimization can be utilized while considering messaging middleware and dis-tributed processing engines to orchestrate and opti-mization of IVA service Pipeline.

**D. IVA AND BIG DIMENSIONALITY**

The VSDS 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 fac-tor poses many intrinsic challenges for data stream acquisition, transmission, learner, pattern recogni-tion problems, indexing, and retrieval. In literature, it has been referred to as a “Big Dimensionality” challenge [371].

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

With growing features dimensionality, current al-gorithms quickly become computationally inflexi-ble and, therefore, inapplicable in many real-time applications [372]. 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 optimiza-tion, online learning, and paralleling computing will be more preferred.

Similarly, designing a generic, efficient, and scal-able multi-level distributed data model for indexing and retrieving multi-dimensional features is becom-ing 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., spa-tiotemporal, 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 commu-

nity the opportunities to optimize existing hashing schemes and indexing structures such as R-trees [373], M-tree [374], X-tree [375], locality-sensitive hashing [376] etc. on the top of big data engines. Log-structured merge-tree [377] based distributed data stores (see Table 4) can be leveraged to im-prove multi-dimensional query performances. Ad-ditionally, 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 [378]. 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 in-dex multi-dimensional data, that make them more precise and effective than the traditional indexing approaches [378].

**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 [379]. The RIVA services must address continuous and changing video streams. In this con-text, 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 [380]. In this context, Nallaperuma et al. [381] proposed ITS platform utilizing unsupervised online learning and deep learning approaches. It gives further re-search opportunities by involving data fusion from heterogeneous data sources [381].

**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 param-eters. The model selection process encompasses fea-ture engineering (feature selection IV-B1), IVA al-gorithm selection, and hyperparameter tuning. Fea-ture engineering is a laborious activity and is in-fluenced by many key factors, e.g., domain-specific regulations, time, accuracy, video data, and IVA properties, which resultantly slow and hinder ex-ploration. IVA algorithm selection is the process of choosing a model that fixes the hypothesis space of prediction function explored for a given applica-tion [382]. This process of IVA algorithm selection

40 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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 ana-lysts usually do ad-hoc manual tuning by iteratively choosing a set of values or using heuristics such as grid search [382]. From IVA analysts’ perspective, model selection is an expensive job in terms of time and resources that bringing down the video analyt-ics lifecycle [383]. 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, the-oretical design trade-offs are presented by Arun et al. [384], 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.

1. **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 pro-poses a new framework for building distributed ML algorithms, and encompass diverse design goals, e.g., efficient communication, flexible consistency, elasticity when adding resources, resource utiliza-tion, and ease of use. In literature, recently, vari-ous studies tried to optimize Parameter Server. PS2

1. builds the parameter server on top of Spark. SketchML [386] compresses the gradient values by a sketch-based method. FlexPS [387] 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 [388] initiated the discussion of IVA per-formance evaluation followed by dedicated work-shops [389], and journals [390], [391]. As a re-sult, performance evaluation tools (ViPER5), and datasets (ETISEO [392], TrecVID [393], i-LIDS [394]) were introduces. 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 comput-ing, e.g., scalability, fault tolerance, memory usage, throughput, etc. [395] (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 circum-stances, 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 sys-tem performance, such as false alarms, accuracy, and detection rate. However, from the developer and researcher perspective, L-CVAS consists of nu-merous computer vision algorithms, with complex interfaces among them. A proper performance eval-uation 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 envi-ronments and deployment code environments with hardware-specific optimizations. These shortcom-ings result in algorithmic alterations that can influ-ence 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, omni-directional, depth map, etc.), property (frame-rate, field-depth), and quality (resolution, pixel depth) of the generated video as generated by the camera.

VSDSand 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.

VSDSenvironmental parameters: some fea-tures of the configuration remain constant in a given use case, but differ between configura-

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 41 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

tion, possibly influencing performance. These parameters comprise camera location (mount-ing 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 poten-tially 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 com-plex trade-offs between different evaluation criteria. In order to comprehend, assume the intricate trade-offs between accuracy and re-sponse time. Iterative tasks have an inverse relation with fault tolerance concerning scal-ability (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 parame-ters are domain-specific parameters, e.g., ve-hicles, carts, humans, etc., tolerable miss de-tection and false alarm rates and their desired trade-off, IVA type, and max acceptable la-tency.

Computation and communication trade-off: IVA Algorithms and services in the distributed environment should be developed and de-signed wisely intending to minimize computa-tion 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 sig-nificantly contribute to the academic and industrial communities interested in building IVA algorithms, services, and CVAS.

1. **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 com-munity 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 pa-rameters 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 investi-gate how to rank and recommend IVA algorithms, models, and services. Similarly, it can lead to a high-dimensional sparse matric [396]. In this direction, research is required on how to utilize such parame-ters 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 [397]. 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 oppo-nent 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. Unfortu-nately, current SaaS cost models might not be ap-plicable 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, re-searcher, and consumers), services (IVAAaaS and

42 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

IVAaaS), service subscription (algorithm, IVA ser-vice, 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 investiga-tions 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 manage-ment, the main issue is the extraction of se-mantic concepts from primitive features. A general domain-independent framework is re-quired that can extract semantic features, an-alyze and model the multiple semantics from the videos by using the primitive features. Fur-ther, 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 mov-ing objects with statics cameras [214], moving cameras [216], and multiple camera fusion.

Limited research is available on CBVR (see section V-A) while exploiting distributed com-puting. Further study is required to consider different features ranging from local to global spatiotemporal features utilizing and optimiz-ing 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. Ontol-ogy adds extra concepts that can improve the retrieval results [183] but can also lead to un-expected 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 re-searchers can conduct studies to answer ques-tions 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 [398] can be exploited to improve the retrieval results?

Recently, video query engines have been intro-duced to retrieve and analyze video at scale [399]–[401]. A special focus of the database community is required to design, implement, optimize, and operationalize such video query engines.

1. **PRIVACY, SECURITY AND TRUST**

Video big data, acquisition, storage, and subscrip-tions 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 com-monly used as a general term for ‘security’ and ‘privacy’ [402]. 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 [403], [404]. In the cloud environ-ment, 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 ve-locity boost security threats. Recently, disputes and news were circulating regarding the mis-use 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 dif-fer 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

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 43 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

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 algorithms are required to secure user’s data both for shared IVA, storage to make the video stream acquisi-tion more secure.

In the context of security, the blockchain (pop-ularized by Bitcoin) [405] 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 [405]. 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 surveil-lance 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 of-fer 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 intel-ligent video big data in the cloud. First, we define basic terminologies and establish the relation be-tween video big data analytics and cloud computing.

A comprehensive layered architecture has been pro-posed 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 account-able 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 IVAAaaS paradigm.

Furthermore, to show the significance and recent research trends of IVA in the cloud, a broad lit-erature 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.

**ACKNOWLEDGMENT**

This work was supported by Institute for Infor-mation and communications Technology Promo-tion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00406, SIAT CCTV Cloud Platform).

**REFERENCES**

1. M. INC., “Video analytics market,” 2012, accessed: 2019-12-

20. [Online]. Available: https://github.com/apache/rocketmq

1. Facebook.Com, “Facebook,” 2020, accessed: 2019-12-20.

[Online]. Available: https://www.facebook.com

1. youtube.com, “Youtube statistics,” 2019, accessed: 2019-

12-20. [Online]. Available: https://www.youtube.com/about/ press/

1. Netflix.Com, “Netflix,” 2020, accessed: 2019-12-20.

[Online]. Available: https://www.netflix.com/

1. S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. Iyen-gar, “Multimedia big data analytics: A survey,” ACM Computing Surveys (CSUR), vol. 51, no. 1, p. 10, 2018.
2. I. E. Olatunji and C.-H. Cheng, “Dynamic threshold for re-source tracking in observed scenes,” in 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA). IEEE, 2018, pp. 1–6.
3. I. D. Corporation, “The growth in connected iot devices,” 2019, accessed: 2019-12-07. [Online]. Available: https: //www.idc.com/getdoc.jsp?containerId=prUS45213219
4. T. Huang, “Surveillance video: The biggest big data,” Computing Now, vol. 7, no. 2, pp. 82–91, 2014.
5. P. Mell, T. Grance et al., “The nist definition of cloud com-puting,” 2011.

44 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

1. P. Zikopoulos, D. Deroos, K. Parasuraman, T. Deutsch,
   1. Giles, and D. Corrigan, Harness the power of big data The IBM big data platform. McGraw Hill Professional, 2012.
2. E. Amazon, “Amazon web services,” Available in: http://aws. amazon. com/es/ec2/(November 2012), 2015.
3. M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, “Overview of microsoft azure services,” in Microsoft Azure. Springer, 2015, pp. 27–69.
4. J.-P. Dijcks, “Oracle: Big data for the enterprise,” Oracle white paper, p. 16, 2012.
5. J. R. Mashey, “Big data and the next wave of infras-tress,” in Computer Science Division Seminar, University of California, Berkeley, 1997.
6. C. Snijders, U. Matzat, and U.-D. Reips, “" big data": big gaps of knowledge in the field of internet science,” International Journal of Internet Science, vol. 7, no. 1, pp. 1–5, 2012.
7. I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, “The rise of “big data” on cloud comput-ing: Review and open research issues,” Information Systems, vol. 47, pp. 98–115, 2015.
8. N. Elgendy and A. Elragal, “Big data analytics: a literature review paper,” in Industrial Conference on Data Mining. Springer, 2014, pp. 214–227.
9. H. Özköse, E. S. Arı, and C. Gencer, “Yesterday, today and tomorrow of big data,” Procedia-Social and Behavioral Sciences, vol. 195, pp. 1042–1050, 2015.
10. S. O. Fadiya, S. Saydam, and V. V. Zira, “Advancing big data for humanitarian needs,” Procedia Engineering, vol. 78, pp. 88–95, 2014.
11. B. Marr, Big Data: Using SMART big data, analytics and metrics to make better decisions and improve performance. John Wiley & Sons, 2015.
12. D. E. O’Leary, “Artificial intelligence and big data,” IEEE Intelligent Systems, vol. 28, no. 2, pp. 96–99, 2013.
13. J. J. Berman, Principles of big data: preparing, sharing, and analyzing complex information. Newnes, 2013.
14. S. John Walker, “Big data: A revolution that will transform how we live, work, and think,” 2014.
15. 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
16. G. F. Pfister, In search of clusters. Prentice Hall PTR Englewood Cliffs, 1998, vol. 2.
17. R. Buyya et al., “High performance cluster computing: Ar-chitectures and systems (volume 1),” Prentice Hall, Upper SaddleRiver, NJ, USA, vol. 1, p. 999, 1999.
18. H. Brian, T. Brunschwiler, H. Dill, H. Christ, B. Falsafi,
    1. Fischer, S. G. Grivas, C. Giovanoli, R. E. Gisi, R. Gut-mann et al., “Cloud computing,” Communications of the ACM, vol. 51, no. 7, pp. 9–11, 2008.
19. G. Aceto, A. Botta, W. De Donato, and A. Pescapè, “Cloud monitoring: A survey,” Computer Networks, vol. 57, no. 9, pp. 2093–2115, 2013.
20. N. Marz and J. Warren, Big Data: Principles and best practices of scalable realtime data systems. New York; Manning Publications Co., 2015.
21. H. Liu, S. Chen, and N. Kubota, “Intelligent video systems and analytics: A survey,” IEEE Transactions on Industrial Informatics, vol. 9, no. 3, pp. 1222–1233, 2013.
22. I. E. Olatunji and C.-H. Cheng, “Video analytics for visual surveillance and applications: An overview and survey,” in Machine Learning Paradigms. Springer, 2019, pp. 475–515.
23. W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, “A sur-vey on visual content-based video indexing and retrieval,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 41, no. 6, pp. 797–819, 2011.
24. B. Patel and B. Meshram, “Content based video retrieval systems,” arXiv preprint arXiv:1205.1641, 2012.
25. M. Haseyama, T. Ogawa, and N. Yagi, “[survey paper] a review of video retrieval based on image and video semantic understanding,” ITE Transactions on Media Technology and Applications, vol. 1, no. 1, pp. 2–9, 2013.
26. B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou,
    1. Shen, and S. Tang, “Hierarchical and networked vehicle surveillance in its: a survey,” IEEE transactions on intelligent transportation systems, vol. 16, no. 2, pp. 557–580, 2014.
27. A. B. Mabrouk and E. Zagrouba, “Abnormal behavior recog-nition for intelligent video surveillance systems: A review,” Expert Systems with Applications, vol. 91, pp. 480–491, 2018.
28. N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, M. Ali,
    1. Kamaleldin, M. Alam, M. Shiraz, and A. Gani, “Big data: survey, technologies, opportunities, and challenges,” The Scientific World Journal, vol. 2014, 2014.
29. C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, “Big data analytics: a survey,” Journal of Big data, vol. 2, no. 1, p. 21, 2015.
30. D. Agrawal, S. Das, and A. El Abbadi, “Big data and cloud computing: current state and future opportunities,” in Proceedings of the 14th International Conference on Extending Database Technology. ACM, 2011, pp. 530–533.
31. L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, “Ma-chine learning on big data: Opportunities and challenges,” Neurocomputing, vol. 237, pp. 350–361, 2017.
32. D. Che, M. Safran, and Z. Peng, “From big data to big data mining: challenges, issues, and opportunities,” in International conference on database systems for advanced applications. Springer, 2013, pp. 1–15.
33. S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. Iyen-gar, “Multimedia big data analytics: A survey,” ACM Computing Surveys (CSUR), vol. 51, no. 1, p. 10, 2018.
34. L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, “Big data analytics in intelligent transportation systems: A survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 383–398, 2018.
35. H. Zahid, T. Mahmood, A. Morshed, and T. Sellis, “Big data analytics in telecommunications: literature review and architecture recommendations,” IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 1, pp. 18–38, 2019.
36. D. Elliott, “Intelligent video solution: A definition,” Security, vol. 47, no. 6, 2010.
37. 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 frame-work,” Machine Vision and Applications, vol. 19, no. 5-6, pp. 315–327, 2008.
38. M. A. Sujana, “Zoneminder video surveillance by linux plat-form,” Ph.D. dissertation, Universiti Teknologi MARA, 2004.
39. A. Alam and Y.-K. Lee, “Tornado: intermediate results or-chestration based service-oriented data curation framework for intelligent video big data analytics in the cloud,” Sensors, vol. 20, no. 12, p. 3581, 2020.
40. J. Kreps, N. Narkhede, J. Rao et al., “Kafka: A distributed messaging system for log processing,” in Proceedings of the NetDB, 2011, pp. 1–7.
41. A. Videla and J. J. Williams, RabbitMQ in action: distributed messaging for everyone. Manning, 2012.
42. B. Snyder, D. Bosnanac, and R. Davies, ActiveMQ in action. Manning Greenwich Conn., 2011, vol. 47.
43. W. Zhang, L. Xu, P. Duan, W. Gong, Q. Lu, and S. Yang, “A video cloud platform combing online and offline cloud com-puting technologies,” Personal and Ubiquitous Computing, vol. 19, no. 7, pp. 1099–1110, 2015.
44. G. Banavar, T. Chandra, R. Strom, and D. Sturman, “A case for message oriented middleware,” in Distributed Computing,
    1. Jayanti, Ed. Berlin, Heidelberg: Springer Berlin Heidel-berg, 1999, pp. 1–17.
45. A. Ejsmont, “Web scalability for startup engineers,” 2015.

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 45 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

1. ——, Web Scalability for Startup Engineers, 1st ed. McGraw-Hill Education Group, 2015.
2. E. Gamma, Design patterns: elements of reusable

object-oriented software. Pearson Education India, 1995.

1. 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 Proceedings of the 7th Symposium on Operating Systems Design and Implementation, ser. OSDI ’06. Berkeley, CA, USA: USENIX Association, 2006, pp. 307–320. [Online]. Available: http://dl.acm.org/citation.cfm? id=1298455.1298485
2. E. B. Boyer, M. C. Broomfield, and T. A. Perrotti, “Glusterfs one storage server to rule them all,” Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2012.
3. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), May 2010, pp. 1–10.
4. P. J. Braam and R. Zahir, “Lustre: A scalable, high perfor-mance file system. cluster file systems,” Inc. technic white paper, 2002.
5. D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié, “Moose: A parallel computational framework for coupled systems of nonlinear equations,” Nuclear Engineering and Design, vol. 239, no. 10, pp. 1768 – 1778, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0029549309002635
6. M. Ovsiannikov, S. Rus, D. Reeves, P. Sutter, S. Rao, and J. Kelly, “The quantcast file system,” Proc. VLDB Endow., vol. 6, no. 11, pp. 1092–1101, Aug. 2013. [Online]. Available: http://dx.doi.org/10.14778/2536222.2536234
7. D. G. Chandra, “Base analysis of nosql database,” Future Generation Computer Systems, vol. 52, pp. 13 – 21, 2015, special Section: Cloud Computing: Security, Privacy and Practice. [Online]. Available: http://www.sciencedirect.com/ science/article/pii/S0167739X15001788
8. A. Moniruzzaman and S. A. Hossain, “Nosql database: New era of databases for big data analytics-classification, charac-teristics and comparison,” arXiv preprint arXiv:1307.0191, 2013.
9. 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
10. E. Brewer, “Pushing the cap: Strategies for consistency and availability,” Computer, vol. 45, no. 2, pp. 23–29, Feb. 2012. [Online]. Available: https://doi.org/10.1109/MC.2012.37
11. ——, “Cap twelve years later: How the "rules" have changed,” Computer, vol. 45, no. 2, pp. 23–29, Feb 2012.
12. J. L. Carlson, Redis in Action. Greenwich, CT, USA: Manning Publications Co., 2013.
13. K. Banker, MongoDB in Action. Greenwich, CT, USA: Manning Publications Co., 2011.
14. M. N. Vora, “Hadoop-hbase for large-scale data,” in Computer science and network technology (ICCSNT), 2011 international conference on, vol. 1. IEEE, 2011, pp. 601– 605.
15. 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. [Online]. Available: http://doi.acm.org/10.1145/ 1365815.1365816
16. M. Stonebraker and A. Weisberg, “The voltdb main memory dbms.” IEEE Data Eng. Bull., vol. 36, no. 2, pp. 21–27, 2013.
17. S. Ghemawat, H. Gobioff, and S.-T. Leung, “The google file system,” in Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, ser. SOSP ’03. New

York, NY, USA: ACM, 2003, pp. 29–43. [Online]. Available:

http://doi.acm.org/10.1145/945445.945450

1. L. Xie, H. Sundaram, and M. Campbell, “Event mining in multimedia streams,” Proceedings of the IEEE, vol. 96, no. 4,
   1. 623–647, 2008.
2. 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 Transactions on Multimedia, vol. 10, no. 2, pp. 252–259, 2008.
3. C. S. Regazzoni, A. Cavallaro, Y. Wu, J. Konrad, and A. Ham-papur, “Video analytics for surveillance: Theory and practice [from the guest editors],” IEEE Signal Processing Magazine, vol. 27, no. 5, pp. 16–17, 2010.
4. L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gram-fort, J. Grobler et al., “Api design for machine learning software: experiences from the scikit-learn project,” arXiv preprint arXiv:1309.0238, 2013.
5. R. R. Schultz and R. L. Stevenson, “Extraction of high-resolution frames from video sequences,” IEEE transactions on image processing, vol. 5, no. 6, pp. 996–1011, 1996.
6. C. Saravanan, “Color image to grayscale image conver-sion,” in 2010 Second International Conference on Computer Engineering and Applications, vol. 2. IEEE, 2010, pp. 196– 199.
7. J. S. Boreczky and L. A. Rowe, “Comparison of video shot boundary detection techniques,” Journal of Electronic Imaging, vol. 5, no. 2, pp. 122–129, 1996.
8. A. Hampapur, T. Weymouth, and R. Jain, “Digital video seg-mentation,” in Proceedings of the second ACM international conference on Multimedia. ACM, 1994, pp. 357–364.
9. A. Vetro, C. Christopoulos, and H. Sun, “Video transcod-ing architectures and techniques: an overview,” IEEE Signal processing magazine, vol. 20, no. 2, pp. 18–29, 2003.
10. Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, “Adap-tive key frame extraction using unsupervised clustering,” in Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269), vol. 1. IEEE, 1998, pp. 866–870.
11. Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8,
    1. 1798–1828, 2013.
12. F. Baumann, A. Ehlers, B. Rosenhahn, and J. Liao, “Computation strategies for volume local binary pat-terns applied to action recognition,” in 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2014, pp. 68–73.
13. ——, “Recognizing human actions using novel space-time volume binary patterns,” Neurocomputing, vol. 173, pp. 54– 63, 2016.
14. A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock et al., “Ibm research trecvid-2003 video retrieval system,” NIST TRECVID-2003, vol. 7, no. 8, p. 36, 2003.
15. J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel, “Fxpal experiments for trecvid 2004,” Proceedings of the TREC Video Retrieval Evaluation (TRECVID), pp. 70–81, 2004.
16. R. Yan and A. G. Hauptmann, “A review of text and image retrieval approaches for broadcast news video,” Information Retrieval, vol. 10, no. 4-5, pp. 445–484, 2007.
17. J. Sivic, M. Everingham, and A. Zisserman, “Person spotting: video shot retrieval for face sets,” in International conference on image and video retrieval. Springer, 2005, pp. 226–236.
18. R. Visser, N. Sebe, and E. Bakker, “Object recognition for video retrieval,” in International Conference on Image and Video Retrieval. Springer, 2002, pp. 262–270.
19. R. Mattivi and L. Shao, “Human action recognition us-ing lbp-top as sparse spatio-temporal feature descriptor,” in

46 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

International Conference on Computer Analysis of Images and Patterns. Springer, 2009, pp. 740–747.

1. G. Zhao and M. Pietikainen, “Dynamic texture recogni-tion using local binary patterns with an application to fa-cial expressions,” IEEE transactions on pattern analysis and machine intelligence, vol. 29, no. 6, pp. 915–928, 2007.
2. L. Nanni, S. Brahnam, and A. Lumini, “Local ternary patterns from three orthogonal planes for human action classification,” Expert Systems with Applications, vol. 38, no. 5, pp. 5125– 5128, 2011.
3. Y. Yi, Z. Zheng, and M. Lin, “Realistic action recognition with salient foreground trajectories,” Expert Systems with Applications, vol. 75, pp. 44–55, 2017.
4. M. A. 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, 2019.
5. K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Advances in neural information processing systems, 2014, pp. 568–576.
6. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with con-volutional neural networks,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
7. L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory-pooled deep-convolutional descriptors,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4305–4314.
8. X. Lu, H. Yao, S. Zhao, X. Sun, and S. Zhang, “Action recog-nition with multi-scale trajectory-pooled 3d convolutional descriptors,” Multimedia Tools and Applications, vol. 78, no. 1, pp. 507–523, 2019.
9. G. Yao, T. Lei, J. Zhong, and P. Jiang, “Learning multi-temporal-scale deep information for action recognition,” Applied Intelligence, vol. 49, no. 6, pp. 2017–2029, 2019.
10. 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, 2018.
11. R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, “Actionvlad: Learning spatio-temporal aggregation for action classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 971– 980.
12. S. Zhao, Y. Liu, Y. Han, R. Hong, Q. Hu, and Q. Tian, “Pool-ing the convolutional layers in deep convnets for video action recognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 8, pp. 1839–1849, 2017.
13. S. C. Hoi, J. Wang, P. Zhao, and R. Jin, “Online feature selection for mining big data,” in Proceedings of the 1st international workshop on big data, streams and heterogeneous source mining: Algorithms, systems, programming models and applications. ACM, 2012, pp. 93– 100.
14. H. Tan and L. Chen, “An approach for fast and parallel video processing on apache hadoop clusters,” in Multimedia and Expo (ICME), 2014 IEEE International Conference on. IEEE, 2014, pp. 1–6.
15. Y. Cong, S. Wang, B. Fan, Y. Yang, and H. Yu, “Udsfs: Un-supervised deep sparse feature selection,” Neurocomputing, vol. 196, pp. 150–158, 2016.
16. O. Dekel, “From online to batch learning with cutoff-averaging,” in Advances in neural information processing systems, 2009, pp. 377–384.
17. A. E. Abdel-Hakim and A. A. Farag, “Csift: A sift de-scriptor with color invariant characteristics,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 2. Ieee, 2006, pp. 1978–1983.
18. Y. Fang and Z. Wang, “Improving lbp features for gender classification,” in 2008 International Conference on Wavelet Analysis and Pattern Recognition, vol. 1. IEEE, 2008, pp. 373–377.
19. O. Déniz, G. Bueno, J. Salido, and F. De la Torre, “Face recognition using histograms of oriented gradients,” Pattern Recognition Letters, vol. 32, no. 12, pp. 1598–1603, 2011.
20. F. Borisyuk, A. Gordo, and V. Sivakumar, “Rosetta: Large scale system for text detection and recognition in images,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp. 71–79.
21. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, p. 436, 2015.
22. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao,
    1. Ranzato, A. Senior, P. Tucker, K. Yang et al., “Large scale distributed deep networks,” in Advances in neural information processing systems, 2012, pp. 1223–1231.
23. J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun,
    1. Kianinejad, M. Patwary, M. Ali, Y. Yang, and Y. Zhou, “Deep learning scaling is predictable, empirically,” arXiv preprint arXiv:1712.00409, 2017.
24. H. Zhang, Z. Zheng, S. Xu, W. Dai, Q. Ho, X. Liang,
    1. Hu, J. Wei, P. Xie, and E. P. Xing, “Poseidon: An efficient communication architecture for distributed deep learning on fGPUg clusters,” in 2017 fUSENIXg Annual Technical Conference (fUSENIXgfATCg 17), 2017, pp. 181–193.
25. J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan,
    1. Vinyals, R. Monga, and G. Toderici, “Beyond short snip-pets: Deep networks for video classification,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4694–4702.
26. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
27. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner et al., “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
28. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,
    1. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
29. ——, “Going deeper with convolutions,” in Computer Vision and Pattern Recognition (CVPR), 2015. [Online]. Available: http://arxiv.org/abs/1409.4842
30. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
31. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learn-ing for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
32. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean,
    1. Devin, S. Ghemawat, G. Irving, M. Isard et al., “Tensor-flow: A system for large-scale machine learning.” in OSDI, vol. 16, 2016, pp. 265–283.
33. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Gir-shick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014, pp. 675–678.
34. N. Ketkar, “Introduction to pytorch,” in Deep learning with python. Springer, 2017, pp. 195–208.
35. T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao,
    1. Xu, C. Zhang, and Z. Zhang, “Mxnet: A flexible and effi-cient machine learning library for heterogeneous distributed systems,” arXiv preprint arXiv:1512.01274, 2015.

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 47 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

1. F. Seide and A. Agarwal, “Cntk: Microsoft’s open-source deep-learning toolkit,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 2135–2135.
2. M. Mathwork, “Deep learning toolbox.”
3. S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and
   1. Wang, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” IEEE transactions on neural networks and learning systems, vol. 30, no. 2, pp. 601–614, 2018.
4. R. Mayer and H.-A. Jacobsen, “Scalable deep learning on distributed infrastructures: Challenges, techniques and tools,” arXiv preprint arXiv:1903.11314, 2019.
5. A. Deshpande and M. Kumar, Artificial Intelligence for Big Data: Complete Guide to Automating Big Data Solutions Using Artificial Intelligence Techniques. Packt Publishing Ltd, 2018.
6. Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen,
   1. Lee, J. Ngiam, Q. V. Le, Y. Wu et al., “Gpipe: Efficient training of giant neural networks using pipeline parallelism,” in Advances in Neural Information Processing Systems, 2019, pp. 103–112.
7. J. Dean and S. Ghemawat, “Mapreduce: simplified data pro-cessing on large clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.
8. M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust,
   1. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin et al., “Apache spark: a unified engine for big data process-ing,” Communications of the ACM, vol. 59, no. 11, pp. 56– 65, 2016.
9. P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, “Apache flink: Stream and batch processing in a single engine,” Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 36, no. 4, 2015.
10. A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham et al., “Storm@ twitter,” in Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014, pp. 147–156.
11. G. Wang, J. Koshy, S. Subramanian, K. Paramasivam,
    1. Zadeh, N. Narkhede, J. Rao, J. Kreps, and J. Stein, “Building a replicated logging system with apache kafka,” Proceedings of the VLDB Endowment, vol. 8, no. 12, pp. 1654–1655, 2015.
12. S. Owen and S. Owen, “Mahout in action,” 2012.
13. D. Team et al., “Deeplearning4j: Open-source distributed deep learning for the jvm,” Apache Software Foundation License, vol. 2, 2016.
14. F. Chollet et al., “Keras,” 2015.
15. J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang, Y. Wang,
    1. Jia, C. Zhang, Y. Wan, Z. Li et al., “Bigdl: A dis-tributed deep learning framework for big data,” arXiv preprint arXiv:1804.05839, 2018.
16. X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman,
    1. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen et al., “Mllib: Machine learning in apache spark,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 1235–1241, 2016.
17. G. D. F. Morales and A. Bifet, “Samoa: scalable advanced massive online analysis.” Journal of Machine Learning Research, vol. 16, no. 1, pp. 149–153, 2015.
18. J. Dean and S. Ghemawat, “Mapreduce: a flexible data pro-cessing tool,” Communications of the ACM, vol. 53, no. 1, pp. 72–77, 2010.
19. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and
    1. Stoica, “Spark: Cluster computing with working sets.” HotCloud, vol. 10, no. 10-10, p. 95, 2010.
20. M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Sto-ica, “Discretized streams: Fault-tolerant streaming compu-tation at scale,” in Proceedings of the twenty-fourth ACM

symposium on operating systems principles. ACM, 2013,

* 1. 423–438.

1. D. Cook, Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI. " O’Reilly Media, Inc.", 2016.
2. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
3. 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,” arXiv preprint arXiv:1503.01817, 2015.
4. Trecvid, “Trec video retrieval evaluation: Trecvid,” 2019,

accessed: 2019-12-07. [Online]. Available: https://trecvid. nist.gov/

1. K. Soomro, A. R. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild,” arXiv preprint arXiv:1212.0402, 2012.
2. W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vi-jayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev et al., “The kinetics human action video dataset,” arXiv preprint arXiv:1705.06950, 2017.
3. CEUR, “Multimedia benchmark workshop,” 2015, accessed: 2019-12-07. [Online]. Available: http://ceur-ws. org/Vol-1436/
4. A. Nambiar, M. Taiana, D. Figueira, J. C. Nascimento, and A. Bernardino, “A multi-camera video dataset for research on high-definition surveillance,” International Journal of Machine Intelligence and Sensory Signal Processing, vol. 1, no. 3, pp. 267–286, 2014.
5. B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 961–970.
6. H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recogni-tion,” in 2011 International Conference on Computer Vision. IEEE, 2011, pp. 2556–2563.
7. C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Suk-thankar et al., “Ava: A video dataset of spatio-temporally localized atomic visual actions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6047–6056.
8. 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,” arXiv preprint arXiv:1906.03327, 2019.
9. S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, “Youtube-8m: A large-scale video classification benchmark,” arXiv preprint arXiv:1609.08675, 2016.
10. L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra, “Video annotation and retrieval using ontologies and rule learning,” IEEE MultiMedia, vol. 17, no. 4, pp. 80–88, 2010.
11. A. F. Smeaton, P. Over, and A. R. Doherty, “Video shot boundary detection: Seven years of trecvid activity,” Computer Vision and Image Understanding, vol. 114, no. 4,
    1. 411–418, 2010.
12. 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. Springer, 2009, pp. 1–24.

[164] H. Bhaumik, S. Bhattacharyya, M. D. Nath, and S. Chakraborty, “Hybrid soft computing approaches to content based video retrieval: A brief review,” Applied Soft Computing, vol. 46, pp. 1008–1029, 2016.

48 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

1. 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,” The Visual Computer, vol. 29, no. 5, pp. 393–405, 2013.
2. N. Spolaôr, H. D. Lee, W. S. R. Takaki, L. A. Ensina,
   1. S. R. Coy, and F. C. Wu, “A systematic review on content-based video retrieval,” Engineering Applications of Artificial Intelligence, vol. 90, p. 103557, 2020.
3. L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, “Real-time large scale near-duplicate web video retrieval,” in Proceedings of the 18th ACM international conference on Multimedia, 2010, pp. 531–540.
4. H. Wang, F. Zhu, B. Xiao, L. Wang, and Y.-G. Jiang, “Gpu-based mapreduce for large-scale near-duplicate video re-trieval,” Multimedia Tools and Applications, vol. 74, no. 23, pp. 10 515–10 534, 2015.
5. K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” International journal of computer vision, vol. 60, no. 1, pp. 63–86, 2004.
6. J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in null. IEEE, 2003, p. 1470.
7. S. Ding, G. Li, Y. Li, X. Li, Q. Zhai, A. C. Champion, J. Zhu,
   1. Xuan, and Y. F. Zheng, “Survsurf: human retrieval on large surveillance video data,” Multimedia Tools and Applications, vol. 76, no. 5, pp. 6521–6549, 2017.
8. N. Zhu, W. He, Y. Hua, and Y. Chen, “Marlin: Taming the big streaming data in large scale video similarity search,” in Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015, pp. 1755–1764.
9. J. Lv, B. Wu, S. Yang, B. Jia, and P. Qiu, “Efficient large scale near-duplicate video detection base on spark,” in Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016, pp. 957–962.
10. M. N. Khan, A. Alam, and Y.-K. Lee, “Falkon: Large-scale content-based video retrieval utilizing deep-features and dis-tributed in-memory computing,” in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2020, pp. 36–43.
11. F.-C. Lin, H.-H. Ngo, and C.-R. Dow, “A cloud-based face video retrieval system with deep learning,” The Journal of Supercomputing, pp. 1–21, 2020.
12. W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, “Semantic-based surveillance video retrieval,” IEEE Transactions on image processing, vol. 16, no. 4, pp. 1168–1181, 2007.
13. J. Sivic and A. Zisserman, “Video google: Efficient vi-sual search of videos,” in Toward category-level object recognition. Springer, 2006, pp. 127–144.
14. Y. Aytar, M. Shah, and J. Luo, “Utilizing semantic word simi-larity measures for video retrieval,” in 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, pp. 1–8.
15. L. S. Kennedy, A. P. Natsev, and S.-F. Chang, “Automatic discovery of query-class-dependent models for multimodal search,” in Proceedings of the 13th annual ACM international conference on Multimedia. ACM, 2005, pp. 882–891.
16. R. Yan, J. Yang, and A. G. Hauptmann, “Learning query-

class dependent weights in automatic video retrieval,” in Proceedings of the 12th annual ACM international conference on Multimedia. ACM, 2004, pp. 548–555.

1. R. Lienhart, “A system for effortless content annotation to unfold the semantics in videos,” in cbaivl. IEEE, 2000, p. 45.
2. C. G. Snoek, B. Huurnink, L. Hollink, M. De Rijke, G. Schreiber, and M. Worring, “Adding semantics to detec-tors for video retrieval,” IEEE Transactions on multimedia, vol. 9, no. 5, pp. 975–986, 2007.
3. S.-Y. Neo, J. Zhao, M.-Y. Kan, and T.-S. Chua, “Video retrieval using high level features: Exploiting query match-ing and confidence-based weighting,” in International

Conference on Image and Video Retrieval. Springer, 2006,

* 1. 143–152.

1. L.-H. Chen, K.-H. Chin, and H.-Y. Liao, “An integrated approach to video retrieval,” in Proceedings of the nineteenth conference on Australasian database-Volume 75. Australian Computer Society, Inc., 2008, pp. 49–55.
2. F. Hopfgartner, J. Urban, R. Villa, and J. Jose, “Simulated testing of an adaptive multimedia information retrieval sys-tem,” in 2007 International Workshop on Content-Based Multimedia Indexing. IEEE, 2007, pp. 328–335.
3. R. Yan, A. G. Hauptmann, and R. Jin, “Negative pseudo-relevance feedback in content-based video retrieval,” in Proceedings of the eleventh ACM international conference on Multimedia. ACM, 2003, pp. 343–346.
4. A. G. Hauptmann, M. G. Christel, and R. Yan, “Video re-trieval based on semantic concepts,” Proceedings of the IEEE, vol. 96, no. 4, pp. 602–622, 2008.
5. C. Sun and R. Nevatia, “Discover: Discovering important segments for classification of video events and recounting,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2569–2576.
6. A. Habibian and C. G. Snoek, “Recommendations for rec-ognizing video events by concept vocabularies,” Computer Vision and Image Understanding, vol. 124, pp. 110–122, 2014.
7. H. Song, X. Wu, W. Yu, and Y. Jia, “Extracting key segments of videos for event detection by learning from web sources,” IEEE Transactions on Multimedia, vol. 20, no. 5, pp. 1088– 1100, 2017.
8. H. Wang, X. Wu, and Y. Jia, “Video annotation via image groups from the web,” IEEE Transactions on Multimedia, vol. 16, no. 5, pp. 1282–1291, 2014.
9. 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 Transactions on Multimedia, vol. 17, no. 9, pp. 1562– 1575, 2015.
10. 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 and Applications, vol. 76, no. 5, pp. 6111–6126, 2017.
11. H. J. Escalante, I. Guyon, V. Athitsos, P. Jangyodsuk, and J. Wan, “Principal motion components for one-shot gesture recognition,” Pattern Analysis and Applications, vol. 20, no. 1, pp. 167–182, 2017.
12. F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, “Multi-layered gesture recognition with kinect,” The Journal of Machine Learning Research, vol. 16, no. 1, pp. 227–254, 2015.
13. D. Wu, F. Zhu, and L. Shao, “One shot learning gesture recog-nition from rgbd images,” in 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2012, pp. 7–12.
14. C. Lu, J. Jia, and C.-K. Tang, “Range-sample depth fea-ture for action recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014,
    1. 772–779.
15. X. Yang and Y. Tian, “Super normal vector for activity recognition using depth sequences,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 804–811.
16. O. Oreifej and Z. Liu, “Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 716–723.
17. X. Yang, C. Zhang, and Y. Tian, “Recognizing actions using depth motion maps-based histograms of oriented gradients,” in Proceedings of the 20th ACM international conference on Multimedia. ACM, 2012, pp. 1057–1060.

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 49 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

1. 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 Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 1119–1122.
2. 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 Transactions on Human-Machine Systems, vol. 46, no. 4, pp. 498–509, 2015.
3. D. Wu, L. Pigou, P.-J. Kindermans, N. D.-H. Le, L. Shao, J. Dambre, and J.-M. Odobez, “Deep dynamic neural net-works for multimodal gesture segmentation and recogni-tion,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 8, pp. 1583–1597, 2016.
4. P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, “Depth pooling based large-scale 3-d action recognition with convo-lutional neural networks,” IEEE Transactions on Multimedia, vol. 20, no. 5, pp. 1051–1061, 2018.
5. V. Veeriah, N. Zhuang, and G.-J. Qi, “Differential recurrent neural networks for action recognition,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4041–4049.
6. Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1110–1118.
7. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional net-works,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489–4497.
8. S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 1, pp. 221–231, 2012.
9. P. Guler, D. Emeksiz, A. Temizel, M. Teke, and T. T. Temizel, “Real-time multi-camera video analytics system on gpu,” Journal of Real-Time Image Processing, vol. 11, no. 3, pp. 457–472, 2016.
10. V. Pham, P. Vo, V. T. Hung et al., “Gpu implementation of ex-tended gaussian mixture model for background subtraction,” in 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF). IEEE, 2010, pp. 1–4.
11. R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detect-ing objects, shadows and ghosts in video streams by ex-ploiting color and motion information,” in Proceedings 11th international conference on image analysis and processing. IEEE, 2001, pp. 360–365.
12. C. Beyan and A. Temizel, “Adaptive mean-shift for auto-mated multi object tracking,” IET computer vision, vol. 6, no. 1, pp. 1–12, 2012.
13. 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 Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2840–2849.
14. S. Shantaiya, K. Verma, and K. Mehta, “A survey on approaches of object detection,” International Journal of Computer Applications, vol. 65, no. 18, 2013.
15. B. Deori and D. M. Thounaojam, “A survey on moving ob-ject tracking in video,” International Journal on Information Theory (IJIT), vol. 3, no. 3, pp. 31–46, 2014.
16. 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,” arXiv preprint arXiv:1704.02781, 2017.
17. 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

Transactions on Internet & Information Systems, vol. 9, no. 10, 2015.

1. S. Zhang, J.-B. Huang, J. Lim, Y. Gong, J. Wang,
   1. Ahuja, and M.-H. Yang, “Tracking persons-of-interest via unsupervised representation adaptation,” arXiv preprint arXiv:1710.02139, 2017.
2. E. Hayman and J.-O. Eklundh, “Statistical background sub-traction for a mobile observer,” in null. IEEE, 2003, p. 67.
3. P. Lenz, J. Ziegler, A. Geiger, and M. Roser, “Sparse scene flow segmentation for moving object detection in urban en-vironments,” in 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011, pp. 926–932.
4. C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pent-land, “Pfinder: Real-time tracking of the human body,” IEEE Transactions on pattern analysis and machine intelligence, vol. 19, no. 7, pp. 780–785, 1997.
5. Z. Zivkovic and F. Van Der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction,” Pattern recognition letters, vol. 27, no. 7, pp. 773–780, 2006.
6. Y. Jin, L. Tao, H. Di, N. I. Rao, and G. Xu, “Background mod-eling from a free-moving camera by multi-layer homography algorithm,” in 2008 15th IEEE International Conference on Image Processing. IEEE, 2008, pp. 1572–1575.
7. Y. Wu, X. He, and T. Q. Nguyen, “Moving object detection with a freely moving camera via background motion subtrac-tion,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 2, pp. 236–248, 2015.
8. M. Braham and M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural net-works,” in 2016 international conference on systems, signals and image processing (IWSSIP). IEEE, 2016, pp. 1–4.
9. T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, “Decomposition into low-rank plus additive matri-ces for background/foreground separation: A review for a comparative evaluation with a large-scale dataset,” Computer Science Review, vol. 23, pp. 1–71, 2017.
10. M. Yazdi and T. Bouwmans, “New trends on moving object detection in video images captured by a moving camera: A survey,” Computer Science Review, vol. 28, pp. 157–177, 2018.
11. I. Elhart, M. Mikusz, C. G. Mora, M. Langheinrich, and
    1. Davies, “Audience monitor: an open source tool for tracking audience mobility in front of pervasive displays,” in Proceedings of the 6th ACM International Symposium on Pervasive Displays. ACM, 2017, p. 10.
12. G. M. Farinella, G. Farioli, S. Battiato, S. Leonardi, and
    1. Gallo, “Face re-identification for digital signage ap-plications,” in International Workshop on Video Analytics for Audience Measurement in Retail and Digital Signage. Springer, 2014, pp. 40–52.
13. T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, “Crowded scene analysis: A survey,” IEEE transactions on circuits and systems for video technology, vol. 25, no. 3, pp. 367–386, 2014.
14. X. Wang, X. Yang, X. He, Q. Teng, and M. Gao, “A high accuracy flow segmentation method in crowded scenes based on streakline,” Optik-International Journal for Light and Electron Optics, vol. 125, no. 3, pp. 924–929, 2014.
15. R. Mehran, B. E. Moore, and M. Shah, “A streakline repre-sentation of flow in crowded scenes,” in European conference on computer vision. Springer, 2010, pp. 439–452.
16. 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 Transactions on Information Forensics and security, vol. 8, no. 10, pp. 1575–1589, 2013.
17. L. Kratz and K. Nishino, “Tracking pedestrians using lo-cal spatio-temporal motion patterns in extremely crowded scenes,” IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 5, pp. 987–1002, 2011.

50 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

1. P.-M. Jodoin, Y. Benezeth, and Y. Wang, “Meta-tracking for video scene understanding,” in 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, 2013, pp. 1–6.
2. 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 gra-dients,” in Mexican Conference on Pattern Recognition. Springer, 2013, pp. 94–103.
3. B. Zhou, X. Wang, and X. Tang, “Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents,” in 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 2871– 2878.
4. W. Chongjing, Z. Xu, Z. Yi, and L. Yuncai, “Analyzing motion patterns in crowded scenes via automatic tracklets clustering,” china communications, vol. 10, no. 4, pp. 144– 154, 2013.
5. B. T. Morris and M. M. Trivedi, “A survey of vision-based trajectory learning and analysis for surveillance,” IEEE transactions on circuits and systems for video technology, vol. 18, no. 8, pp. 1114–1127, 2008.
6. C. Piciarelli, C. Micheloni, and G. L. Foresti, “Trajectory-based anomalous event detection,” IEEE Transactions on Circuits and Systems for video Technology, vol. 18, no. 11,
   1. 1544–1554, 2008.
7. S. Co¸sar, G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, and F. Brémond, “Toward abnormal trajectory and event de-tection in video surveillance,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 3, pp. 683– 695, 2016.
8. 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 Transactions on Intelligent Systems and Technology (TIST), vol. 4, no. 2,
   1. 35, 2013.
9. L. Brun, A. Saggese, and M. Vento, “Dynamic scene under-standing for behavior analysis based on string kernels,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 10, pp. 1669–1681, 2014.
10. D. Tran, J. Yuan, and D. Forsyth, “Video event detec-tion: From subvolume localization to spatiotemporal path search,” IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 2, pp. 404–416, 2013.
11. A. R. Revathi and D. Kumar, “An efficient system for anomaly detection using deep learning classifier,” Signal, Image and Video Processing, vol. 11, no. 2, pp. 291– 299, Feb 2017. [Online]. Available: https://doi.org/10.1007/ s11760-016-0935-0
12. O. P. Popoola and K. Wang, “Video-based abnormal hu-man behavior recognition—a review,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 865–878, Nov 2012.
13. Y. Yuan, Y. Feng, and X. Lu, “Statistical hypothesis detec-tor for abnormal event detection in crowded scenes,” IEEE Transactions on Cybernetics, vol. 47, no. 11, pp. 3597–3608, Nov 2017.
14. 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, 2012. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S0925231211006990

1. Y. Zhang, L. Qin, R. Ji, H. Yao, and Q. Huang, “Social attribute-aware force model: Exploiting richness of interac-tion for abnormal crowd detection,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 7,
   1. 1231–1245, July 2015.
2. S. Yi, X. Wang, C. Lu, and J. Jia, “L0 regularized stationary time estimation for crowd group analysis,” in Proceedings of

the 2014 IEEE Conference on Computer Vision and Pattern Recognition, ser. CVPR ’14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 2219–2226. [Online]. Available: https://doi.org/10.1109/CVPR.2014.284

1. K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Video anomaly detection and localization using hierarchical feature rep-resentation and gaussian process regression,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2909–2917, 2015.
2. Y. jye Lee, Y. ren Yeh, and Y. chiang Frank Wang, “Anomaly detection via online oversampling principal component anal-ysis,” IEEE Trans. Knowledge and Data Eng, 2013.
3. B. Krausz and C. Bauckhage, “Loveparade 2010: Automatic video analysis of a crowd disaster,” Computer Vision

and Image Understanding, vol. 116, no. 3, pp. 307

– 319, 2012, special issue on Semantic Understanding of Human Behaviors in Image Sequences. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S1077314211002037

1. D. Lee, H. Suk, S. Park, and S. Lee, “Motion influence map for unusual human activity detection and localization in crowded scenes,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 10, pp. 1612–1623, Oct 2015.
2. C. C. Loy, T. Xiang, and S. Gong, “Salient motion detection in crowded scenes,” in 2012 5th International Symposium on Communications, Control and Signal Processing, May 2012, pp. 1–4.
3. S. Vishwakarma and A. Agrawal, “A survey on activity recognition and behavior understanding in video surveillance,” The Visual Computer, vol. 29, no. 10, pp. 983–1009, 2013. [Online]. Available: https://doi.org/10.1007/s00371-012-0752-6
4. K. Xu, X. Jiang, and T. Sun, “Anomaly detection based on stacked sparse coding with intraframe classification strategy,” IEEE Transactions on Multimedia, vol. 20, no. 5, pp. 1062– 1074, May 2018.
5. D. Sen, B. Raman et al., “Video skimming: taxonomy and comprehensive survey,” ACM Computing Surveys (CSUR), vol. 52, no. 5, p. 106, 2019.
6. X.-n. Xie and F. Wu, “Automatic video summarization by affinity propagation clustering and semantic content mining,” in 2008 International Symposium on Electronic Commerce and Security. IEEE, 2008, pp. 203–208.
7. J. Wu, S.-h. Zhong, J. Jiang, and Y. Yang, “A novel clustering method for static video summarization,” Multimedia Tools and Applications, vol. 76, no. 7, pp. 9625–9641, 2017.
8. H. Bhaumik, S. Bhattacharyya, S. Dutta, and S. Chakraborty, “Towards redundancy reduction in storyboard representa-tion for static video summarization,” in 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2014, pp. 344–350.
9. K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Summary transfer: Exemplar-based subset selection for video summa-rization,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1059–1067.
10. M. Gygli, H. Grabner, and L. Van Gool, “Video summa-rization by learning submodular mixtures of objectives,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3090–3098.
11. J. Mohan and M. S. Nair, “Static video summarization using sparse autoencoders,” in 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2019, pp. 1–8.
12. Z. Ji, Y. Zhao, Y. Pang, and X. Li, “Cross-modal guidance based auto-encoder for multi-video summarization,” Pattern Recognition Letters, 2020.
13. K. A. Peker, I. Otsuka, and A. Divakaran, “Broadcast video program summarization using face tracks,” in 2006 IEEE

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 51 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

International Conference on Multimedia and Expo. IEEE, 2006, pp. 1053–1056.

1. C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, “Video summarization and scene detection by graph modeling,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 2, pp. 296–305, 2005.
2. 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 2008 Congress on Image and Signal Processing, vol. 4. IEEE, 2008, pp. 555–559.
3. Y. Gao, W.-B. Wang, and J.-H. Yong, “A video summa-rization tool using two-level redundancy detection for per-sonal video recorders,” IEEE Transactions on Consumer Electronics, vol. 54, no. 2, pp. 521–526, 2008.
4. B. Xu, X. Wang, and Y.-G. Jiang, “Fast summarization of user-generated videos: Exploiting semantic, emotional, and quality clues,” IEEE MultiMedia, vol. 23, no. 3, pp. 23–33, 2016.
5. Z. Ji, K. Xiong, Y. Pang, and X. Li, “Video summariza-tion with attention-based encoder-decoder networks,” IEEE Transactions on Circuits and Systems for Video Technology, 2019.
6. J. Wu, S.-h. Zhong, and Y. Liu, “Dynamic graph convo-lutional network for multi-video summarization,” Pattern Recognition, p. 107382, 2020.
7. S.-h. Zhong, J. Wu, and J. Jiang, “Video summarization via spatio-temporal deep architecture,” Neurocomputing, vol. 332, pp. 224–235, 2019.
8. A. Rav-Acha, Y. Pritch, and S. Peleg, “Making a long video short: Dynamic video synopsis,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 1. IEEE, 2006, pp. 435–441.
9. Y. He, Z. Qu, C. Gao, and N. Sang, “Fast online video synopsis based on potential collision graph,” IEEE Signal Processing Letters, vol. 24, no. 1, pp. 22–26, 2016.
10. Y. He, C. Gao, N. Sang, Z. Qu, and J. Han, “Graph coloring based surveillance video synopsis,” Neurocomputing, vol. 225, pp. 64–79, 2017.
11. K. Batuhan Baskurt and R. Same, “Improved adaptive back-ground subtraction method using pixel-based segmenter,” 2017.
12. K. B. BASKURT¸ and R. Samet, “Long-term multiobject tracking using alternative correlation filters,” Turkish Journal of Electrical Engineering and Computer Science, vol. 26, no. 5, pp. 2246–2259, 2018.
13. L. Lin, W. Lin, W. Xiao, and S. Huang, “An optimized video synopsis algorithm and its distributed processing model,” Soft computing, vol. 21, no. 4, pp. 935–947, 2017.
14. 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 Transactions on Intelligent Transportation Systems, 2019.
15. X. Zhu, J. Liu, J. Wang, and H. Lu, “Key observation selection-based effective video synopsis for camera network,” Machine vision and applications, vol. 25, no. 1, pp. 145–157, 2014.
16. A. Mahapatra, P. K. Sa, B. Majhi, and S. Padhy, “Mvs: A multi-view video synopsis framework,” Signal Processing: Image Communication, vol. 42, pp. 31–44, 2016.
17. Z. Zhang, Y. Nie, H. Sun, Q. Zhang, Q. Lai, G. Li, and M. Xiao, “Multi-view video synopsis via simultane-ous object-shifting and view-switching optimization,” IEEE Transactions on Image Processing, vol. 29, pp. 971–985, 2019.
18. J. Ferryman and A.-L. Ellis, “Performance evaluation of crowd image analysis using the pets2009 dataset,” Pattern Recognition Letters, vol. 44, pp. 3–15, 2014.
19. 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 occlu-sions,” in 2013 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS). IEEE, 2013, pp. 29–36.

1. S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papas-tathis, and M. G. Strintzis, “Knowledge-assisted semantic video object detection,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 10, pp. 1210– 1224, 2005.
2. A. García and J. Bescós, “Video object segmentation based on feedback schemes guided by a low-level scene ontol-ogy,” in International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, 2008, pp. 322–333.
3. J. Gómez-Romero, M. A. Patricio, J. García, and J. M. Molina, “Ontology-based context representation and reason-ing for object tracking and scene interpretation in video,” Expert Systems with Applications, vol. 38, no. 6, pp. 7494– 7510, 2011.
4. V. Haarslev and R. Möller, “Racer system description,” in International Joint Conference on Automated Reasoning. Springer, 2001, pp. 701–705.
5. N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs, “Detection of loitering individuals in pub-lic transportation areas,” IEEE Transactions on intelligent transportation systems, vol. 6, no. 2, pp. 167–177, 2005.
6. B. Laxton, J. Lim, and D. Kriegman, “Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video,” in 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1–8.
7. J. Wang, Z. Chen, and Y. Wu, “Action recognition with multiscale spatio-temporal contexts,” in CVPR 2011. IEEE, 2011, pp. 3185–3192.
8. A. R. Francois, R. Nevatia, J. Hobbs, R. C. Bolles, and J. R. Smith, “Verl: an ontology framework for representing and annotating video events,” IEEE multimedia, vol. 12, no. 4, pp. 76–86, 2005.
9. L. Snidaro, M. Belluz, and G. L. Foresti, “Representing and recognizing complex events in surveillance applications,” in 2007 IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, 2007, pp. 493–498.
10. J. C. SanMiguel, J. M. Martinez, and Á. Garcia, “An ontology for event detection and its application in surveillance video,” in 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, 2009, pp. 220– 225.
11. L. Greco, P. Ritrovato, A. Saggese, and M. Vento, “Abnormal event recognition: A hybrid approach using semanticweb technologies,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2016, pp. 1297–1304.
12. L. Chen and C. Nugent, “Ontology-based activity recognition in intelligent pervasive environments,” International Journal of Web Information Systems, vol. 5, no. 4, pp. 410–430, 2009.
13. D. Cavaliere, S. Senatore, M. Vento, and V. Loia, “To-wards semantic context-aware drones for aerial scenes un-derstanding,” in 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2016, pp. 115–121.
14. B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu, “I2t: Image parsing to text description,” Proceedings of the IEEE, vol. 98, no. 8, pp. 1485–1508, 2010.
15. M. Xue, S. Zheng, and C. Zhang, “Ontology-based surveil-lance video archive and retrieval system,” in 2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI). IEEE, 2012, pp. 84–89.
16. Z. Xu, L. Mei, Y. Liu, and C. Hu, “Video structural descrip-tion: a semantic based model for representing and organizing video surveillance big data,” in 2013 IEEE 16th International

52 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

Conference on Computational Science and Engineering.

IEEE, 2013, pp. 802–809.

1. Z. Xu, Y. Liu, L. Mei, C. Hu, and L. Chen, “Semantic based representing and organizing surveillance big data using video structural description technology,” Journal of Systems and Software, vol. 102, pp. 217–225, 2015.
2. M. Sah and C. Direkoglu, “Semantic annotation of surveil-lance videos for abnormal crowd behaviour search and analysis,” in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2017, pp. 1–6.
3. F. Sobhani, N. F. Kahar, and Q. Zhang, “An ontology frame-work for automated visual surveillance system,” in 2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI). IEEE, 2015, pp. 1–7.
4. A. Alam, M. N. Khan, J. Khan, and Y.-K. Lee, “Intellibvr-intelligent large-scale video retrieval for objects and events utilizing distributed deep-learning and semantic approaches,” in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2020, pp. 28–35.
5. S. O. Ajiboye, P. Birch, C. Chatwin, and R. Young, “Hier-archical video surveillance architecture: a chassis for video big data analytics and exploration,” in Video Surveillance and Transportation Imaging Applications 2015, vol. 9407. Inter-national Society for Optics and Photonics, 2015, p. 94070K.
6. C.-F. Lin, S.-M. Yuan, M.-C. Leu, and C.-T. Tsai, “A frame-work for scalable cloud video recorder system in surveil-lance environment,” in Ubiquitous intelligence & computing and 9th international conference on autonomic & trusted computing (UIC/ATC), 2012 9th international conference on. IEEE, 2012, pp. 655–660.
7. 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.
8. C. Ryu, D. Lee, M. Jang, C. Kim, and E. Seo, “Exten-sible video processing framework in apache hadoop,” in Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on, vol. 2. IEEE, 2013, pp. 305–310.
9. G. Bradski and A. Kaehler, “Opencv,” Dr. Dobb’s journal of software tools, vol. 3, 2000.
10. 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 Transactions on Cloud Computing, 2020.
11. B. White, T. Yeh, J. Lin, and L. Davis, “Web-scale com-puter vision using mapreduce for multimedia data mining,” in Proceedings of the Tenth International Workshop on Multimedia Data Mining. ACM, 2010, p. 9.
12. R. Pereira, M. Azambuja, K. Breitman, and M. Endler, “An architecture for distributed high performance video process-ing in the cloud,” in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, 2010, pp. 482–489.
13. C. Liu, K. Fan, Z. Yang, and J. Xiu, “A distributed video share system based on hadoop,” in Cloud Computing and Intelligence Systems (CCIS), 2014 IEEE 3rd International Conference on. IEEE, 2014, pp. 587–590.
14. M. A. Hossain, “Framework for a cloud-based multimedia surveillance system,” International Journal of Distributed Sensor Networks, vol. 10, no. 5, p. 135257, 2014.
15. 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,” The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 42, p. 99, 2017.
16. 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.
17. W. Zhang, L. Xu, Z. Li, Q. Lu, and Y. Liu, “A deep-

intelligence framework for online video processing,” IEEE Software, vol. 33, no. 2, pp. 44–51, 2016.

1. 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, 2019.
2. H. Zhang, J. Yan, and Y. Kou, “Efficient online surveillance video processing based on spark framework,” in International Conference on Big Data Computing and Communications. Springer, 2016, pp. 309–318.
3. M. A. Uddin, J. B. Joolee, A. Alam, and Y.-K. Lee, “Human action recognition using adaptive local motion descriptor in spark,” IEEE Access, vol. 5, pp. 21 157–21 167, 2017.
4. H. Wang, X. Zheng, and B. Xiao, “Large-scale human action recognition with spark,” in Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on. IEEE, 2015, pp. 1–6.
5. Q. Huang, P. Ang, P. Knowles, T. Nykiel, I. Tverdokhlib,
   1. Yajurvedi, P. Dapolito IV, X. Yan, M. Bykov, C. Liang et al., “Sve: Distributed video processing at facebook scale,” in Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017, pp. 87–103.
6. W. Zhang, Z. Wang, X. Liu, H. Sun, J. Zhou, Y. Liu, and
   1. Gong, “Deep learning-based real-time fine-grained pedes-trian recognition using stream processing,” IET Intelligent Transport Systems, vol. 12, no. 7, pp. 602–609, 2018.
7. Citilog, “Citilog,” 1997.
8. checkvideo, “checkvideo,” 1998, accessed: 2018-12-07.

[Online]. Available: https://www.checkvideo.com/

1. I. Vision, “Intelli vision,” 2002. [Online]. Available: https://www.intelli-vision.com/
2. G. Inc, “Video ai,” 2017. [Online]. Available: https: //cloud.google.com/video-intelligence/
3. B. LLC, “Usher in the new cognitive era,” 2015. [Online]. Available: https://bluechasm.com/
4. J. H. Julia Kornich, Craig Casey, “Media an-alytics on the media services platform,” 2010. [Online]. Available: https://docs.microsoft.com/en-gb/azure/ media-services/previous/media-services-analytics-overview
5. G. Inc, “Features,” 2017. [Online]. Available: https://cloud. google.com/video-intelligence/docs/features
6. J. G. Collier, “Ibm uses watson’s power to unlock insights from video in the cloud,” 2017. [Online]. Available: https: //www-03.ibm.com/press/us/en/pressrelease/52126.wss
7. I. Cloud, “Ibm cloud object storage,” 2017. [Online]. Available: https://www.ibm.com/cloud/object-storage
8. ——, “Cloudant,” 2017. [Online]. Available: https://www. ibm.com/cloud/cloudant
9. G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintala-pudi, 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.
10. F. Loewenherz, V. Bahl, and Y. Wang, “Video analytics towards vision zero,” Institute of Transportation Engineers. ITE Journal, vol. 87, no. 3, p. 25, 2017.
11. H. Qiu, X. Liu, S. Rallapalli, A. J. Bency, K. Chan,
    1. Urgaonkar, B. Manjunath, and R. Govindan, “Kestrel: Video analytics for augmented multi-camera vehicle track-ing,” in 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE, 2018, pp. 48–59.
12. 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,” Cybernetics and Systems: An International Journal, vol. 40, no. 7, pp. 569–588, 2009.
13. 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 Analysis & Prevention, vol. 144, p. 105628, 2020.
14. L. Fridman, D. E. Brown, M. Glazer, W. Angell, S. Dodd,
    1. Jenik, J. Terwilliger, J. Kindelsberger, L. Ding, S. Seaman

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 53 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

et al., “Mit autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and in-teraction with automation,” arXiv preprint arXiv:1711.06976, 2017.

1. W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and
   1. Shen, “Internet of vehicles in big data era,” IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 1, pp. 19–35, 2017.
2. J. Wang, G. Bebis, M. Nicolescu, M. Nicolescu, and
   1. Miller, “Improving target detection by coupling it with tracking,” Machine Vision and Applications, vol. 20, no. 4, pp. 205–223, 2009.
3. Y. J. Tsai, Z. Hu, and C. Alberti, “Detection of roadway sign condition changes using multi-scale sign image matching (m-sim),” Photogrammetric Engineering & Remote Sensing, vol. 76, no. 4, pp. 391–405, 2010.
4. 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 Processing Letters, vol. 51, no. 1, pp. 837–851, 2020.
5. A. M. Callan, R. Osu, Y. Yamagishi, D. E. Callan, and N. In-oue, “Neural correlates of resolving uncertainty in driver’s decision making,” Human brain mapping, vol. 30, no. 9, pp. 2804–2812, 2009.
6. C. Yan, F. Coenen, and B. Zhang, “Driving posture recog-nition by convolutional neural networks,” IET Computer Vision, vol. 10, no. 2, pp. 103–114, 2016.
7. S. Gite and H. Agrawal, “Early prediction of driver’s ac-tion using deep neural networks,” International Journal of Information Retrieval Research (IJIRR), vol. 9, no. 2, pp. 11– 27, 2019.
8. S. Fleck and W. Straßer, “Smart camera based monitoring system and its application to assisted living,” Proceedings of the IEEE, vol. 96, no. 10, pp. 1698–1714, 2008.
9. 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 Transactions on Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1489–1498, 2008.
10. J. Aertssen, M. Rudinac, and P. Jonker, “Fall and ac-tion detection in elderly homes,” AAATE: Maastricht, The Netherlands, 2011.
11. D. Brulin, Y. Benezeth, and E. Courtial, “Posture recognition based on fuzzy logic for home monitoring of the elderly,” IEEE transactions on information technology in biomedicine, vol. 16, no. 5, pp. 974–982, 2012.
12. G. Sacco, V. Joumier, N. Darmon, A. Dechamps, A. Der-reumaux, J.-H. Lee, J. Piano, N. Bordone, A. Konig,
    1. Teboul et al., “Detection of activities of daily living impair-ment in alzheimer’s disease and mild cognitive impairment using information and communication technology,” Clinical interventions in aging, vol. 7, p. 539, 2012.
13. M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Cham-bers, “A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment,” IEEE transactions on information technology in biomedicine, vol. 16, no. 6, pp. 1274–1286, 2012.
14. W. Choi, C. Pantofaru, and S. Savarese, “A general frame-work for tracking multiple people from a moving cam-era,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 7, pp. 1577–1591, 2012.
15. V. Pragada, “Intrusion detection and video surveillance ac-tivation and processing,” Aug. 6 2015, uS Patent App. 14/172,880.
16. Z. Zhao, H. Li, R. Zhao, and X. Wang, “Crossing-line crowd counting with two-phase deep neural networks,” in European Conference on Computer Vision. Springer, 2016, pp. 712– 726.
17. G. Xu and W. K. Cobb, “Loitering detection in a video surveillance system,” Dec. 8 2015, uS Patent 9,208,675.
18. P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, “Cnn-rnn based method for license plate recognition,” CAAI Transactions on Intelligence Technology, vol. 3, no. 3, pp. 169–175, 2018.
19. B. Krausz and R. Herpers, “Metrosurv: detecting events in subway stations,” Multimedia Tools and Applications, vol. 50, no. 1, pp. 123–147, 2010.
20. J.-L. Shih, C.-c. Han, and K.-C. Yan, “Illegal entrant detec-tion at a restricted area in open spaces using color features,” in Proceedings 40th Annual 2006 International Carnahan Conference on Security Technology. IEEE, 2006, pp. 68–74.
21. 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. IGI Global, 2018, pp. 42–59.
22. H. Xu, P. Lv, and L. Meng, “A people counting system based on head-shoulder detection and tracking in surveil-lance video,” in 2010 International Conference On Computer Design and Applications, vol. 1. IEEE, 2010, pp. V1–394.
23. I. Haritaoglu and M. Flickner, “Attentive billboards: To-wards to video based customer behavior understanding,” in Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings. IEEE, 2002, pp. 127–
24. Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, “Ac-tion detection in complex scenes with spatial and temporal ambiguities,” in 2009 IEEE 12th International Conference on Computer Vision. IEEE, 2009, pp. 128–135.
25. R. Sicre and H. Nicolas, “Human behaviour analysis and event recognition at a point of sale,” in 2010 Fourth Pacific-Rim Symposium on Image and Video Technology. IEEE, 2010, pp. 127–132.
26. A. D. Popescu, A. Balmin, V. Ercegovac, and A. Ailamaki, “Predict: towards predicting the runtime of large scale itera-tive analytics,” Proceedings of the VLDB Endowment, vol. 6, no. 14, pp. 1678–1689, 2013.
27. S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, “Power to the people: The role of humans in interactive machine learning,” AI Magazine, vol. 35, no. 4, pp. 105–120, 2014.
28. Y.-K. Kwok and I. Ahmad, “Static scheduling algorithms for allocating directed task graphs to multiprocessors,” ACM Computing Surveys (CSUR), vol. 31, no. 4, pp. 406–471, 1999.
29. X. Zhu, J. Wang, J. Wang, and X. Qin, “Analysis and de-sign of fault-tolerant scheduling for real-time tasks on earth-observation satellites,” in 2014 43rd International Conference on Parallel Processing. IEEE, 2014, pp. 491–500.
30. E. Bortnikov, A. Frank, E. Hillel, and S. Rao, “Predicting execution bottlenecks in map-reduce clusters,” in Presented as part of the, 2012.
31. Y. Zhai, Y.-S. Ong, and I. W. Tsang, “The emerging" big dimensionality",” 2014.
32. 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 Systems, vol. 23, no. 3, pp. 303–313, 2017.
33. A. Guttman, “R-trees: A dynamic index structure for spatial searching,” in Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’84. New York, NY, USA: ACM, 1984, pp. 47–
    1. [Online]. Available: http://doi.acm.org/10.1145/602259. 602266
34. P. Ciaccia, M. Patella, and P. Zezula, “M-tree: An efficient

access method for similarity search in metric spaces,” in Proceedings of the 23rd International Conference on Very Large Data Bases, ser. VLDB ’97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 426–435. [Online]. Available: http://dl.acm.org/citation.cfm? id=645923.671005

[375] K.-I. Lin, H. V. Jagadish, and C. Faloutsos, “The tv-tree:

54 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

An index structure for high-dimensional data,” The VLDB Journal, vol. 3, no. 4, pp. 517–542, 1994.

1. M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distri-butions,” in Proceedings of the twentieth annual symposium on Computational geometry. ACM, 2004, pp. 253–262.
2. P. O’Neil, E. Cheng, D. Gawlick, and E. O’Neil, “The log-structured merge-tree (lsm-tree),” Acta Informatica, vol. 33, no. 4, pp. 351–385, 1996.
3. T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis, “The case for learned index structures,” in Proceedings of the 2018 International Conference on Management of Data. ACM, 2018, pp. 489–504.
4. G. De Francisci Morales, “Samoa: A platform for mining big data streams,” in Proceedings of the 22nd International Conference on World Wide Web. ACM, 2013, pp. 777–778.
5. J. Lu, S. C. Hoi, J. Wang, P. Zhao, and Z.-Y. Liu, “Large scale online kernel learning,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 1613–1655, 2016.
6. D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari,
   1. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and
   2. Pothuhera, “Online incremental machine learning plat-form for big data-driven smart traffic management,” IEEE Transactions on Intelligent Transportation Systems, 2019.
7. J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning. Springer series in statistics New York, 2001, vol. 1, no. 10.
8. S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, “En-terprise data analysis and visualization: An interview study,” IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 12, pp. 2917–2926, 2012.
9. A. Kumar, R. McCann, J. Naughton, and J. M. Patel, “Model selection management systems: The next frontier of advanced analytics,” ACM SIGMOD Record, vol. 44, no. 4, pp. 17–22, 2016.
10. Z. Zhang, B. Cui, Y. Shao, L. Yu, J. Jiang, and X. Miao, “Ps2: Parameter server on spark,” in Proceedings of the 2019 International Conference on Management of Data, 2019, pp. 376–388.
11. J. Jiang, F. Fu, T. Yang, and B. Cui, “Sketchml: Accel-erating distributed machine learning with data sketches,” in Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 1269–1284.
12. Y. Huang, T. Jin, Y. Wu, Z. Cai, X. Yan, F. Yang, J. Li,
    1. Guo, and J. Cheng, “Flexps: Flexible parallelism control in parameter server architecture,” Proceedings of the VLDB Endowment, vol. 11, no. 5, pp. 566–579, 2018.
13. R. M. Haralick, “Performance characterization in computer vision,” in BMVC92. Springer, 1992, pp. 1–8.
14. W. Förstner, “Dagm workshop on performance characteris-tics and quality of computer vision algorithms,” Technical University of Brunswick, Germany, 1997.
15. P. J. Phillips and K. W. Bowyer, “Empirical evaluation of computer vision algorithms,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 289– 290, 1999.
16. H. Christensen and W. Foerstner, “Special issue on perfor-mance evaluation,” Machine Vision and Applications, vol. 9, no. 5, 1997.
17. A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, “Etiseo, performance evaluation for video surveillance sys-tems,” in Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on. IEEE, 2007, pp. 476–481.
18. A. G. Hauptmann and M. G. Christel, “Successful approaches in the trec video retrieval evaluations,” in Proceedings of the 12th annual ACM international conference on Multimedia. ACM, 2004, pp. 668–675.
19. H. O. S. D. Branch, “Imagery library for intelligent detection

systems (i-lids),” in 2006 IET Conference on Crime and Security. IET, 2006, pp. 445–448.

1. D. Singh and C. K. Reddy, “A survey on platforms for big data analytics,” Journal of big data, vol. 2, no. 1, p. 8, 2015.
2. 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 Transactions on Systems, Man, and Cybernetics: Systems, 2019.
3. A. Mazrekaj, I. Shabani, and B. Sejdiu, “Pricing schemes in cloud computing: an overview,” International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, pp. 80–86, 2016.
4. S. Khalid, S. Wu, A. Alam, and I. Ullah, “Real-time feedback query expansion technique for supporting scholarly search using citation network analysis,” Journal of Information Science, p. 0165551519863346, 2019.
5. D. Kang, P. Bailis, and M. Zaharia, “Challenges and oppor-tunities in dnn-based video analytics: A demonstration of the blazeit video query engine.” in CIDR, 2019.
6. Y. Lu, A. Chowdhery, and S. Kandula, “Visflow: a relational platform for efficient large-scale video analytics,” in ACM Symposium on Cloud Computing (SoCC), 2016.
7. D. Kang, P. Bailis, and M. Zaharia, “Blazeit: Optimizing declarative aggregation and limit queries for neural network-based video analytics.”
8. A. Ghosh and I. Arce, “Guest editors’ introduction: In cloud computing we trust-but should we?” IEEE security & privacy, vol. 8, no. 6, pp. 14–16, 2010.
9. S. Khusro, A. Alam, and S. Khalid, “Social question and answer sites: the story so far,” Program, vol. 51, no. 2, pp. 170–192, 2017.
10. 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 ex-perts generation,” Journal of Information Science, vol. 43, no. 2, pp. 260–274, 2017.
11. M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman et al., “Blockchain technology: Beyond bitcoin,” Applied Innovation, vol. 2, no. 6-10, p. 71, 2016.

AFTAB ALAM has received his mas-ter’s degree in computer science with a specialization in Web engineering from the Department of Computer Science, University of Peshawar, Pe-shawar, Pakistan. He is currently pur-suing his Ph.D. at the Department of Computer Science and Engineering, Kyung Hee University (Global Cam-pus), South Korea. His research inter-



ests in the areas of: Big data analytics, distributed computing, video retrieval, social computing, machine/deep learning, infor-mation service engineering, knowledge engineering, knowledge graph and graph mining.

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 55 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3017135, IEEE Access



A Alam et al.: Video Big Data Analytics in the cloud

IRFAN ULLAH has received his mas-



ter’s degree in computer science from

the National University of Sciences

and Technology (NUST), Islamabad,

Pakistan. Currently, he is serving as

a research assistant at the Department

of Computer Science and Engineering,

Kyung Hee University (Global Cam-

pus), South Korea. His research inter-

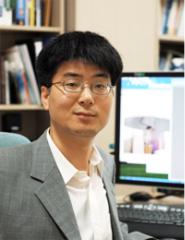
ests in the areas of: Natural language

processing, social computing, crisis informatics, machine/deep

learning, OS design and optimization on memory systems, big

data analytics, and distributed computing.

YOUNG-KOO LEE has received the



B.S., M.S., and Ph.D. degrees in com-

puter science from the Korea Advanced

Institute of Science and Technology,

Daejeon, South Korea, in 1992, 1994,

and 2002. Since 2004, he has been with

the Department of Computer Science

and Engineering, Kyung Hee Univer-

sity. His researches have been concen-

trated on data mining, online analytical

processing, and big data processing.



56 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.