ISEE: An Intelligent Scene Exploration and Evaluation Platform for Large-Scale Visual Surveillance

Da Li, Zhang Zhang, Kai Yu, Kaiqi Huang, *Senior Member, IEEE,* and Tieniu Tan, *Fellow, IEEE*

Abstract—Intelligent video surveillance (IVS) is always an interesting research topic to utilize visual analysis algorithms for exploring richly structured information from big surveillance data. However, existing IVS systems either struggle to utilize computing resources adequately to improve the efficiency of large-scale video analysis, or present a customized system for specific video analytic functions. It still lacks of a comprehensive computing architecture to enhance efficiency, extensibility and flexibility of IVS system. Moreover, it is also an open problem to study the effect of the combinations of multiple vision modules on the final performance of end applications of IVS system. Motivated by these challenges, we develop an Intelligent Scene Exploration and Evaluation (ISEE) platform based on a heterogeneous CPU-GPU cluster and some distributed computing tools, where Spark Streaming serves as the computing engine for efficient large-scale video processing and Kafka is adopted as a middle-ware message center to decouple different analysis modules flexibly. To validate the efficiency of the ISEE and study the evaluation problem on composable systems, we instantiate the ISEE for an end application on person retrieval with three visual analysis modules, including pedestrian detection with tracking, attribute recognition and re-identification. Extensive experiments are performed on a large-scale surveillance video dataset involving 25 camera scenes, totally 587 hours 720p synchronous videos, where a two-stage question-answering procedure is proposed to measure the performance of execution pipelines composed of multiple visual analysis algorithms based on millions of attribute-based and relationship-based queries. The case study of system-level evaluations may inspire researchers to improve visual analysis algorithms and combining strategies from the view of a scalable and composable system in the future.

Index Terms—Intelligent Surveillance System, Big Visual Data, Distributed System and Parallel Computing.

1 Introduction

Recent years, more and more video surveillance devices are deployed as the increasing demands on public security and smart city. By the year 2010, more than 10 million monitoring cameras have been equipped for surveillance systems in China alone [1]. With such huge number of cameras, surveillance video has become the largest source of Big Data [2]. To alleviate the labor intensive task of monitoring surveillance regions as well as explore richly valuable information from the big surveillance data, researchers seek the advanced computer vision algorithms to develop intelligent video surveillance (IVS), where the raw non-structured video can be parsed into meaningful structured information, e.g., object categories, attributes, activities, automatically. As introduced in [2] and [3], it is always an active research area for developing an IVS system to explore big video data efficiently, flexibly and accurately.

The research of IVS grows up from the beginning of this century. One of classical work "W4" [4] describes the main research issues of IVS, i.e., developing IVS system to answer the W4 problems, i.e., Who, When, Where and What, in virtue of visual analysis algorithms. As the core component of IVS system, visual analysis towards video surveillance has been an important research issue in computer vision community. In past decades, many prominent vision algorithms have been proposed, ranging from low-level background modeling, object detection [5], [6], [7], [8] to middle-level target tracking [9], [10], [11], trajectory analysis [12], [13], [14] and high-level activity recognition [15], [16], [17]. Most researches focus on a single vision task using specific data preprocessed from raw video data to validate the effectiveness of proposed algorithms. A few work on IVS system, such as IBM S3 system [18], has also been developed with the main purpose for processing large-scale video data from multi-cameras network, where several individual analysis functions with simple combinations of algorithms are presented for demonstration.

Currently, the deep learning based models have achieved breakthrough in many single vision tasks, e.g., image classification, object detection and image captioning. However, just as the review in [19], todays AI systems are monolithic which makes them hard to develop, test, and evolve. Thus, it is essential to build a composable IVS system with flexible modular execution as well as large scale

- *D. Li is with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China, and also with the CRIPAC, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. E-mail: da.li@cripac.ia.ac.cn.*
- *Z. Zhang is with the CRIPAC and NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with UCAS, Beijing 100049, China. E-mail: zzhang@nlpr.ia.ac.cn.*
- *K. Yu is with Carnegie Mellon University, Pittsburgh, PA 15213, United States. E-mail: kaiy1@andrew.cmu.edu. This work was done during his internship at CRIPAC, Institute of Automation, Chinese Academy of Sciences.*
- *K. Huang is with CRISE, Institute of Automation, Chinese Academy of Sciences, Beijing, 100090, China. E-mail: kqhuang@nlpr.ia.ac.cn.*
- *T. Tan is with CRIPAC and NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, 100090, China. E-mail: tnt@nlpr.ia.ac.cn.*

(Corresponding author: Zhang Zhang.)

computing capabilities. In summary, there are mainly three challenging issues to be concerned for a composable IVS system, i.e., *efficiency, extensibility, and evaluation*, termed as "*E3*" in abbreviation.

- *Efficiency*: To process the huge amount of surveillance videos efficiently, researchers developed various distributed computing architectures based on Message Passing Interface (MPI) [20], or open-source distributed architectures [21], [22] or cloud computing strategies [23], [24], [25]. Meanwhile, since the dependencies in different vision modules, how to schedule different analysis tasks with appropriate computing resources is very important for an efficient IVS system. However, most solutions utilized a simple fixed scheduling strategy, i.e., assigning specific tasks to fixed worker nodes, which will result in the wasting of computing resources.
- *Extensibility*: The demand for extensibility denotes the IVS system can implement different visual analysis modules conveniently. On one hand, a new vision module should be easily embedded into current system. On the other hand, for existing vision algorithms, the execution pipelines indicating the execution order of multiple vision modules should be modified flexibly. However, many current IVS systems are heavily customized for specific applications. It is difficult to add new analysis modules or to update the existing ones. Thus, how to construct a platform with good extensibility and flexibility is still a big challenging problem.
- *Evaluation*: In a composable IVS system, there are a number of visual analysis modules. Some are general analysis modules, such as motion detection, object recognition, tracking, and some are higher-level analysis modules for specific applications, e.g., people counting, activity recognition etc. Thus, the final performance of an end application depends on all vision algorithms in the execution pipeline. However, vision algorithms at different levels are often studied independently with self input-output definitions, train/test dataset and evaluation metrics. In an integration system, some vision algorithms with superior performance on single-task datasets may not be the optimal choice as a component in a whole system [26].

To alleviate the three challenges, an Intelligent Scene Exploration and Evaluation (ISEE) platform is proposed in this paper. It aims at meeting the needs for parsing large-scale surveillance video data with multiple kinds of computer vision algorithms efficiently and flexibly. Based on these parsing results, a unified system evaluation can be performed, such that different system pipelines and vision algorithms can be evaluated at system level effectively and consistently. An overview of the goal of ISEE is illustrated in Fig. 1. Here, toward a specific end application on person retrieval, we instantiate the ISEE with three visual analysis algorithms, i.e., pedestrian detection with tracking, attribute recognition and re-identification (ReID). We adopt open-source distributed framework, including, Hadoop Yarn [27] and Spark Streaming [28], and Apache Kafka [29] to construct the underlying computing architecture of ISEE platform. Then, the Neo4j [30] graph database is adopted for organizing all parsing results. By performing person query in the graph database, the system evaluation is performed.

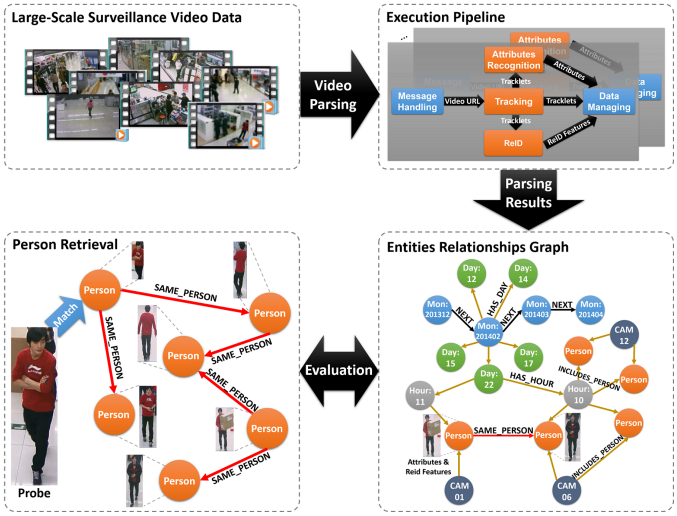


Fig. 1. An illustration of the goal of ISEE platform. The large-scale surveillance video data is parsed by a specific execution pipeline efficiently, where the meta data is organized by a semantic graph including visual entities and various relationships. Subsequently, an end application on person retrieval is conducted based on the semantic graph to evaluate the performance of current parsing pipeline.

In summary, the contributions of this paper include:

- A distributed computing platform ISEE is developed for parsing big surveillance video data, where Spark Streaming is adopted for processing large-scale video efficiently and multiple visual analysis tasks are scheduled by Hadoop Yarn for allocating computing resources automatically. Based on the proposed computing and scheduling strategies, the execution efficiency of computing resources can be improved significantly, compared to traditional IVS system.
- The middle-ware message center is designed to use in ISEE platform based on Apache Kafka. It decouples different analysis modules in the platform so that it is possible to add or replace new algorithms conveniently. Furthermore, the execution order of multiple analysis modules can be specified by user with a form of flow graph in User Interface (UI). The design of middle-ware message center and the flow graph based execution pipeline enhance the extensibility and flexibility of IVS system significantly.
- Toward an end application in video surveillance, i.e., person retrieval, we instantiate the ISEE with three visual analysis modules, i.e., pedestrian detection with tracking, attribute recognition and ReID. Based on a richly annotated pedestrian (RAP) dataset [31] and large-scale raw HD videos, we perform extensive end-to-end system evaluations from both the efficiency and the performance of person retrieval with different vision algorithms and execution pipelines. This case study on system-level evaluations may inspire researchers to improve visual analysis algorithms and combining strategies from the view of a scalable and composable system in the future.

The remainder of this paper is organized as follows: Section 2 describes related work on large-scale IVS systems. Section 3 presents the proposed ISEE platform in details. As a case study of the ISEE, i.e., the task of person retrieval, the implementation setting up and video datasets are introduced in Section 4. Section 5 shows the experimental results. Finally, we conclude this work in Section 6.

2 Related Work

Early IVS systems, e.g., Pfinder [32], Advisor [33] and Vsstar [34] etc., paid little attentions on the design of scalable computing architecture for large-scale video processing. Since our work aims to construct an efficient and extensional IVS system based on distributed architectures, and further perform system-level evaluation, the related work can be divided into two parts. One is distributed architectures for IVS, the other is IVS system evaluations.

2.1 Distributed architectures for IVS

As one pioneer in IVS, Video Surveillance and Monitoring (VSAM) [35] system proposed a series of design concepts, e.g., scalability, modularization and reusability, which are still important for modern IVS systems. As a real IVS system deploying for large-scale applications, IBM Smart Surveillance System (S3) [18] is an excellent representative in early years, who adopted distributed processing architectures to handle large-scale video data, and proposed the design patterns of middle-ware and plugin to support the resources sharing and meta-data management. However, these early systems usually adopt a fixed scheduling strategy where a specific analysis task is assigned to one or more fixed machines, so that it is difficult to meet the needs for processing big video data efficiently. Nowadays, as the increasing developments of technologies of Big Data, many IVS systems are built based on parallel programming model and cloud computing platform.

2.1.1 Parallel Programming Models Based Systems

Message Passing Interface (MPI) [36], Hadoop MapReduce [37] and Spark [38] are the three most popular parallel programming models used in the systems of large-scale image/video processing.

MPI is a communication protocol for programming parallel computers [36]. It provides rich interfaces which are used to data exchanging among different task nodes. The functions of data synchronization, sending and receiving among parallel tasks are also completed through calling these interfaces. Mubasher et al., [20] proposed a system using MPI to detect video changes based on Gaussian Mixture Model (GMM). Qi et al., [39] constructed a MPI based Visual Turing Test (VTT) system, where 93.5 hours surveillance videos are analyzed with 8 analysis modules. One of the advantages of MPI is its flexibility that no intermediate file generates during processing. However, as a programming model for communication, it requires high network bandwidth among computing nodes. As another drawback, MPI lacks of fault-tolerant mechanism, which is very important for Big Data processing.

Hadoop MapReduce is a programming model proposed by Google [37], which is an associated implementation for processing and generating big data sets in parallel. In recent years, more and more systems for video analysis are built based on Hadoop MapReduce architecture due to its advantages of easy deployment and fault-tolerance. Tan et al. [21] proposed a system for distributed video analysis, where two independent algorithms, i.e., face detection and motion detection and tracking, are tested on the system. Zhao et al., [40] provided a framework to extend Hadoop MapReduce to support video analytic applications, where the interfaces of a series of common video analysis algorithms were designed for using with Hadoop MapReduce easily. Two applications of image dehazing and object detection and tracking are tested under their framework. G. Li et al., [41] also constructed a system based on Hadoop MapReduce for large visual traffic data analysis. Although Hadoop MapReduce based systems achieve scalability and good efficiency to some extents, the huge intermediate files generated in one execution pipeline must be stored in Hadoop Distributed File System (HDFS) [42] for subsequent processing. The frequent I/O operations will greatly hinder the system from further improvement of efficiency.

Besides Hadoop MapReduce, Spark [38] is another popular MapReduce-like distributed computing architecture. It has almost all the merits of Hadoop MapReduce, due to its nature of MapReduce programming model. Moreover, benefiting from the advantage of in-memory computing, the frequency of I/O operations of Spark will be decreased significantly. For the tasks of video analysis, Yang et al. [22] proposed to use Spark for video action detection and near-duplicate video retrieval. Similarly, Wang et al. [43] developed a Spark based system for action recognition in large-scale offline videos. Compared to Hadoop MapReduce, the speeds of offline video processing in above work have been improved significantly using Spark. While for realtime video processing in IVS system, Spark Streaming [28] which divides a streaming computing into several mini-batch tasks has received more attentions due to its high throughput, real time and fault tolerance. In [44] and [45], researchers proposed to develop IVS systems for realtime video processing based on Spark Streaming and Kafka, yet only some primary experimental results were presented with the lacks of necessary analysis on efficiency and scalability.

Different from previous work, some researchers designed specific underlying parallel mechanism for large-scale image/video processing. Antonio et al. [3] proposed a scalable and flexible IVS system to separate analysis modules from underlying computing architecture, so that the execution pipelines can be performed with different analysis modules flexibly. However, the system can be only deployed to a single machine, but cannot scale to clusters. Scanner [46], is another system for productive and efficient big video data analysis. In Scanner, the source video data, intermediate outputs and processing results are stored into database as relational tables. It provides mechanisms to access data from database efficiently and flexibly in execution pipelines. Moreover, it also support to easily distribute them to heterogeneous computing resources, i.e., CPUs, GPUs and media processing ASICs. However, Scanner only supports to pre-defined sequential execution pipeline but not complex Directed Acyclic Graphs (DAGs). It may limit its flexibility and extensibility in complex video analysis

applications. As an improved version of Scanner, tScanner [47] is able to build complex DAGs through replacing the scheduler with a Tensorflow [48] graph generator. Although these above systems achieved scalability, extensibility and good performance in experiments, we prefer to construct the ISEE platform based on MapReduce or Spark whose stability, fault-tolerance and efficiency have been validated in different applications.

2.1.2 Cloud Computing Based Systems

Besides the popular distributed programming models, a number of researchers also constructed large-scale IVS systems based on some cloud computing infrastructures developed by several IT companies, such as the system based on Amazon EC2 for spatio-temporal analysis on large-scale camera networks [23], the system Optasia based on Microsoft Cosmos for large-scale video analytic [25], the system based on Google cloud [24] and the system [49] constructed by OpenStack [50], and so on.

The most advantage of IVS systems based on cloud computing is that the developers of IVS systems need only focus on developing vision algorithms and designing execution pipelines but not considering the underlying parallel computing strategies, so that the developing difficulties and development cycle can be reduced greatly. Take Optasia [25] as an example, it allows users to develop analysis modules under provided interfaces and then submit them to system to be implemented in parallel, where the degree of parallelism is adjusted automatically in terms of the amount of data need to be processed. Moreover for the multiple queries at one time, the system will merge the overlapping processes to further improve the query efficiency. However, such systems based on cloud computing heavily depend on specific cloud platforms with the lack of flexibility and extra costs for cloud services.

2.2 IVS System Evaluation

Most work in above focused on designing and developing distributed architecture to promote the capabilities of IVS systems for large-scale video processing, e.g., efficiency, extensibility or scalability. Only a few researchers studied the evaluations of IVS systems integrating multiple visual analysis algorithms. Venetianer and Deng [51] discussed some challenges involved in IVS system evaluation, e.g., environment factors, metrics, and ground-truth, and provided a fuzzy ground-truth based evaluation approach for both system-level and component-level evaluation. However, this work only showed a conceptual framework without practical test for a real IVS system.

Recently, Qi et. al., [39] developed an IVS system with the end application of restricted visual Turing test (VTT), which provides a good example for system-level evaluation to test the overall performance of system to understand long-term and wide-area surveillance scenes. The system consists of multiple video parsing modules, a query engine, a knowledge base and a Q/A evaluation server, where 93.5 hours videos captured in four different locations are analyzed by a cluster with ten workstations. They annotated 3,426 story-line queries to evaluate the performance of system. However, the computing infrastructure of the system depends on MPI programming without efficient scheduling for multiple analysis tasks.

As another application, person search first proposed by Xu et al. [52], is to study the combination of pedestrian detection and person re-identification as an end-to-end system. Recently, the performance of person search in [26] and [53] can validate the importance of system-level evaluation, rather than the evaluations at level of individual vision algorithm. However, existing work still lacks of large-scale tests with wild surveillance data. In [26], the total length of data videos including 6 cameras scenes is only 10 hours.

3 Architecture of ISEE Platform

3.1 Overview

The architecture of ISEE platform is shown in Fig. 2, where the ISEE platform consists of the following parts:

Control Center: Control center is responsible for loading initial system configuration and launching all the kernel modules and analysis modules specified by users. For all analysis modules, control center will calculate the necessary size of physical computing resources and set running parameters respectively. In addition, control center is also used to monitor the heartbeat messages of system.

Message Center: Message center is responsible for sending, receiving and caching messages of different system modules. Specially, in ISEE, it acts as middle-ware to decouple system modules from each other, so that the visual analysis modules in an execution pipeline do not depend on their antecedent modules directly. Thus, the usage of message center permits users to construct extensible pipelines, in which the analysis modules can be replaced, removed and added flexibly. Moreover, the message center also enhances the fault-tolerance of IVS system.

Analysis Modules: To realize a specific application, different analysis modules are organized to a pipeline. Multiple instances of a pipeline will be implemented in parallel on ISEE platform. Each module should define the interface to call a specific algorithm and the interface to message center. Different vision algorithms are embedded into analysis modules as plugins, so that the flexibility is further improved. Currently, for the end application of person retrieval, three analysis modules of pedestrian detection with tracking, attribute recognition and re-identification are integrated into ISEE platform.

Data storage: Two kinds of data storage models are adopted in ISEE platform, i.e., HDFS and Graph Database. The meta-data generated by analysis modules will be inserted into HDFS or Graph Database in terms of data type. Considering the stability and fault-tolerance, but high latency of HDFS, the data with larger volume are stored in it, such as raw video data, image data in the bounding boxes of tracklets, etc. While the parsing results of interesting objects (tracklets coordinates, attributes, etc.) are inserted into Neo4j database. Finally the graph database recording the parsing results will support further queries by end users and system evaluations.

Search Server & Web UI: The component is responsible for the interactions between end users and ISEE system. On one hand, users can define the execution pipelines (select analysis modules and the execution logical order) of IVS

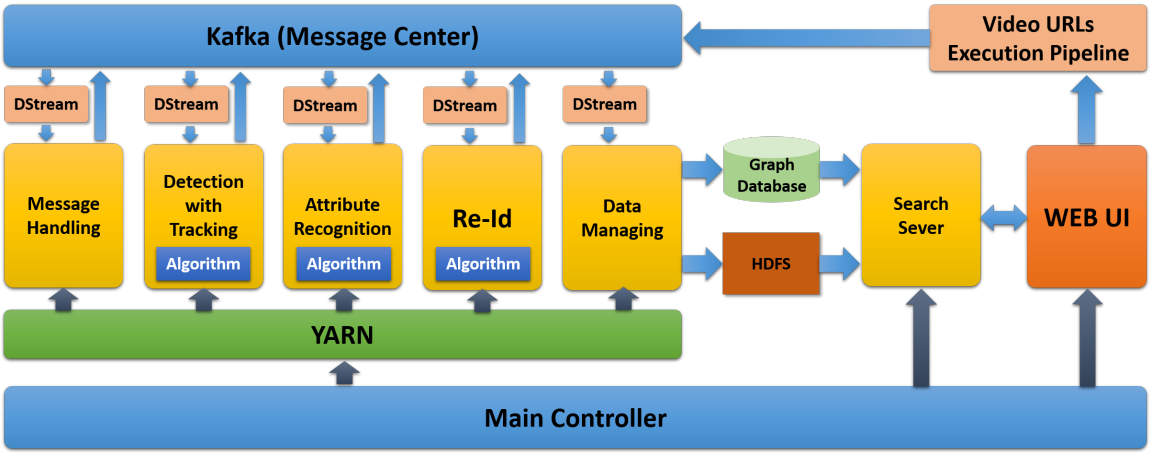


Fig. 2. Architecture of ISEE platform. It mainly includes five parts. And details of each part are presented in the text.

system. On the other hand, it receives the query conditions that users specified and feedback the retrieval results from graph database.

To construct ISEE platform for large-scale video processing, a set of advanced big data tools are adopted, e.g., Apache Hadoop YARN [27], Spark [38], [28], Apache Kafka [29], Apache HDFS [42], Neo4j graph database [30]. The functions of these tools are listed in Table 1. The detailed mechanisms are explained in the following.

TABLE 1
The Big Data Tools Used in ISEE Platform and Their Functions.

Tools	Functions
Hadoop Yarn	To schedule the computing resources and tasks.
Spark	Distributed computing engine.
Kafka	To realize the message center. (It is a distributed message consuming and producing system.)
HDFS	To store the data with the requirement on high stability but not low latency (e.g., source video data, frame data of trajectories).
Neo4j	To store the meta-data e.g., pedestrian trajectories, attributes, ReID features, and relationships.

3.2 Computing And Communication

The computing and communication are the core components in an IVS system. Here, Spark Streaming and Kafka are used as the underlying infrastructures of computing and communication components in ISEE, respectively.

As the computing engine of ISEE, Spark Streaming utilizes Spark's fast scheduling capability for large-scale video processing. When a computing application based on Spark Streaming is submitted, it will process input data in batch-grain, where the real-time data stream will be divided into a series of RDDs (Resilient Distributed Datasets) [54] termed Discredited Stream (DStream). If there is not any data need to be processed, the application keeps running in idle state until the users/system halt it. In ISEE platform, all visual analysis modules and two auxiliary modules, i.e., MessageHandling for communication and DataManaging for data storage (shown in Fig. 2), are the applications based on Spark Streaming.

As the message center of ISEE, Kafka is chosen for transmitting intermediate data among different applications of Spark Streaming. The work flow between Kafka and Spark Streaming is shown in Fig. 3, where each application in Spark Streaming consumes the messages with specified topics from Kafka. For a continuous stream input, the size of RDDs depends on the length of time window (Δt) for data collection. Each RDD will be scheduled as multiple partitions for implementing in parallel, where Spark Streaming consumes it from Kafka with a tiny time window. Thus, the number of partitions determines the degree of parallelism.

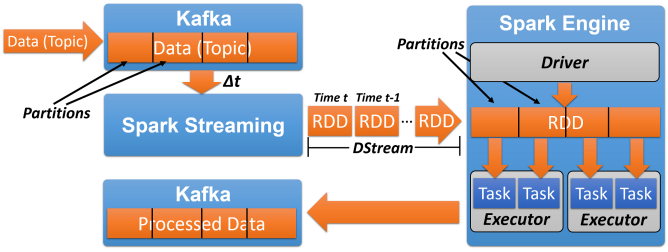


Fig. 3. Flow diagram of data processing with Kafka and Spark Streaming. The data in Kafka is divided into several partitions.

3.3 Task Scheduling And Task Flow

In ISEE, multiple kinds of video analysis tasks submitted by different users can be implemented concurrently, where each task can be further decomposed into a number of implementations of visual analysis modules. Here, Hadoop Yarn is adopted for scheduling tasks with available computing resources. When an application (App.) of Spark Streaming is submitted to Yarn, Yarn allocates computing resources, controls the creation/destruction and monitors the runtime status of each worker node for the App. The App. is an instantiation of certain visual analysis module for ISEE implementation. Thus, each task is scheduled by Yarn automatically, which increases the efficiency of ISEE.

For multiple tasks of ISEE, the implementation pipeline of vision modules, termed *Execution Plan*, can be specified by users' definitions through Web UI. Thus, each task of

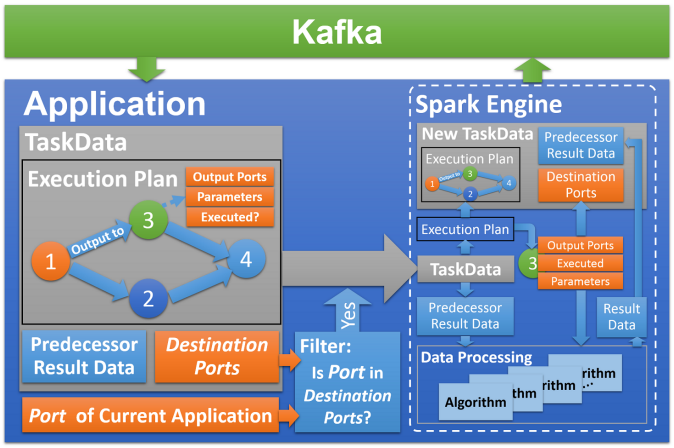


Fig. 4. The task flow in ISEE platform. The class *TaskData* defines the tasks flow among different modules. It mainly consists of *Execution Plan* (a.k.a, DAG), *Predecessor Result Data* and *Destination Ports*.

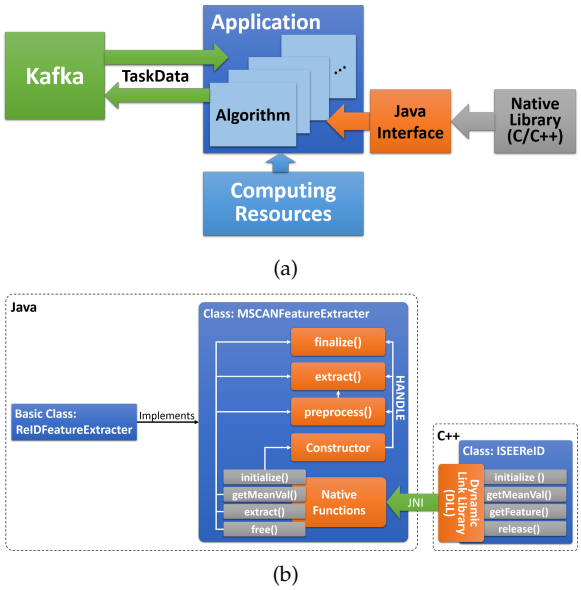


Fig. 5. Method for algorithms embedding. (a) An overview for algorithms embedding using JNI. (b) An example (the module for ReID feature extraction) shows the definitions of interfaces.

ISEE corresponds to an execution plan represented by a Directed Acyclic Graph (DAG). To distinguish different tasks concurrently executed by ISEE, the execution state of each task as well as the intermediate data is attached to the data messages in Kafka, termed as *TaskData*. Concretely, *TaskData* consists of the DAG of *Execution Plan*, *Predecessor Result Data* and *Destination Ports*, where *Destination Ports* indicates the next Apps of the message to be sent for further processing. As one App. receives the *TaskData* from Kafka's message queue with specified topic, it verifies the *Destination Ports* with the *Port* in itself that is pre-defined in creation of App. If the *Port* belongs to *Destination Ports*, the App. processes this data. After the execution of this vision module, the output result is bounded with the updated state of execution plan to form a new *TaskData* where the *Destination Ports* is updated by the next nodes in the execution plan DAG. The new *TaskData* is sent to Kafka for the subsequent processing. The process is illustrated in Fig. 4.

3.4 Encapsulation of Visual Analysis Algorithms

ISEE is developed by Java and Scala, while most computer vision models are developed by C/C++. Moreover, there are two kinds of analysis models deployed by ISEE using different computing hardware, i.e., deep learning models based on GPU implementation and traditional analysis models based on CPU implementation.

Here, we encapsulate the visual analysis algorithms into a class of analysis modules written by Java which define the methods to interact with underlying architecture (e.g., Message Center, Control Center, etc.), as shown in Fig. 5(a). The Java Native Interfaces (JNIs) are defined inside the analysis modules, which determine the interface to vision algorithms in C/C++. Then the codes in C/C++ are packaged into dynamic link libraries (DLLs) based on the predefined JNIs. Fig. 5(b) shows an example to encapsulate the algorithm of *MSCANFeatureExtraction* into the corresponding module in ISEE as a plugin. With such methods, the inside visual analysis algorithm can be replaced easily without the need for considering the underlying architecture in ISEE.

3.5 Meta-Data Management

A flexible and extensible management of multiple kinds of meta-data (analysis results) generated by different visual analysis modules is also very important for providing friendly and efficient experience of interactions with end users. In ISEE, we adopt Neo4j Graph Database to organize the meta-data obtained from large-scale videos into a huge graph for information retrieval and system evaluation.

Firstly, we construct a space-time tree to organize all visual objects detected at different time and surveillance scenes, in which the Year, Month, Day, Hour and Minute are as the temporal nodes organized hierarchically. Meanwhile, different surveillance cameras are represented as spatial nodes. An example of space-time tree is shown in Fig. 6. Then, the detected visual objects are inserted to the corresponding space-time nodes. For each visual object, its visual attributes recognized by vision modules are saved as

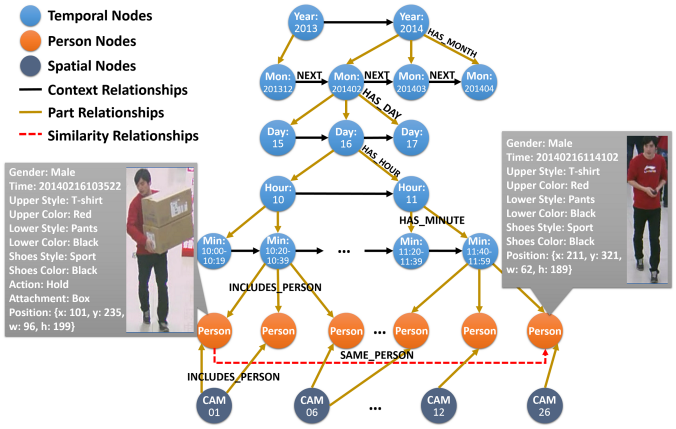


Fig. 6. Meta-Data management using Graph Database. It consists of spatio-temporal entities, visual entities, and the corresponding relationships. The spatio-temporal nodes compose of a tree-like structure to organize the detected visual entities with part or context relationships.

various properties of the corresponding node. Subsequently, the similarity relationships between different objects are added by the edges between nodes, which may be obtained by the vision module of pedestrian ReID. The graph-based representation provides an intuitive and unified approach to organizing visual entities (nodes), attributes (properties), similarity relationships and spatial-temporal context relationships (edges) detected by multiple vision modules. Meanwhile, Neo4j supports efficient queries on interesting nodes and edges for large graphs.

4 An End Application: Person Retrieval

As an end application in video surveillance, *Person Retrieval* is performed in this work to test the efficiency of ISEE for processing large-scale surveillance videos and present a case study of system-level evaluations on different execution pipelines, where two kinds of queries including attribute-based query (AQ) and relationship-based query (RQ), relating to three kinds of vision algorithms, i.e., pedestrian detection with tracking, attribute recognition, and re-identification, are tested on HD videos with duration of over 400 hours. The details on the dataset, querying types, vision algorithms and evaluation metrics are described as follows.

4.1 Dataset

The richly annotated pedestrian (RAP) dataset [31] is adopted to demonstrate the ISEE platform, which is collected from the wild indoor surveillance video data (720p) which covers 25 scenes and 587 hours. For training and test of various algorithms on attribute recognition, totally 84,928 pedestrian image samples are annotated manually with 72 visual attributes as well as the spatio-temporal positions. Furthermore, a subset of the RAP including 26,638 image samples is annotated with 2,589 person IDs which are visible in at least two camera views. The details on the RAP dataset can be found in [31].

To evaluate the efficiency for large-scale video analytics, different amounts of raw videos with the durations of 10h, 50h, 100h, 200h and 410h are feed to ISEE platform under different execution pipelines and computing resources. To measure the system-level performance on person retrieval, a set of video files with total duration of 275 hours are selected, where 17,227 person samples in the RAP are adopted to form querying conditions in AQs and RQs.

Here, as the input of ISEE, raw continuous videos are split into an amount of video files with the size of 64M (around 10 minutes) per file saved in HDFS.

4.2 Queries

Inspired by the restricted Visual Turing Test (VTT) in [55] [39], we adopt a question-answering (QA) paradigm to test the performance of person retrieval. Here, two kinds of queries on person retrieval are designed. One is the attribute-based query (AQ) which queries whether a person with specific attributes existing in a specific spatio-temporal region or not. The other is relationship-based query (RQ) to check whether two person images belong to the same person ID or not. By retrieving from the parsing results in the graph database, an answer with binary values "Yes" or "No" is returned.

Attribute-based Query:

Q: Is it a person in the green bounding box? (Define the person as P1)	***A: Yes***
Q: Is P1 female?	A: Yes
Q: Does P1 hold a bag?	A: Yes
Q: Does P1 has long hair and wear leather shoes?	A: Yes
Q: Is P1 in padded jacket and skirt?	A: No
Q: …	A: …

Relationship-based Query:

Q: Are they persons in both of the two red bounding boxes? (Define the upper one as P2, and define the lower one as P3)	***A: Yes***
Q: Are P2 and P3 the same person?	A: Yes

Fig. 7. Examples of the question-answering paradigm in person retrieval. The definition stage is to confirm the given bounding boxes to be persons or not. If the answer is "Yes" and correct, subsequent attribute or relationship queries will be performed.

For both kinds of queries, each query is executed with two stages:

- The definition stage determines whether a person or a pair of persons existing in specific sptio-temporal regions (bounding boxes in certain frame) or not.
- The recognition stage will be processed, if and only if a positive definition query is returned by "Yes". In this stage, the attributes of a single person or the ReID relationship of a pair of persons are further determined true or false.

Examples of the two kinds of queries are shown in Fig. 7.

For both kinds of queries, a large number of queries including not only positive but negative queries are generated for a sufficient and comprehensive performance measurement of ISEE. The distributions of each kinds of queries are summarized in Table 2. Since the negative samples have much larger variations than that of positive ones, the numbers of negative queries have 10 times larger than those of positive queries. Especially the number of negative queries in RQ's definition stage is over 100 times larger than that of positive queries, since each RQ needs to specify two bounding boxes of person samples.

For AQ, the positive queries in definition stage are generated by 17,227 persons' bounding boxes in the RAP, while the negative queries are generated by random sampling bounding boxes from non-person regions in test frames. Based on the positive bounding boxes, each attribute query in recognition stage corresponds to question the person

TABLE 2
The statistics of AQs and RQs.

AQ	Definition Stage		
	# Queries	# Positives	17,227
		# Negatives	179,078
	Attribute Recognition Stage		
	# Attributes	# Queries	
		# Positives	# Negatives
	1	74,188	365,191
	2	22,096	313,442
	3	16,682	654,394
	4	2,267	189,469
	Totally	115,233	1,522,496
RQ	Definition Stage		
	# Queries	# Positives	109,705
		# Negatives	11,572,437
	Relationships Recognition Stage		
	# Queries	# Positives	9,704
		# Negatives	100,001

sample existing single or multiple (1-4) attribute categories. Finally, over 1.5 millions of AQs are generated.

For RQ, a query in definition stage is to determine whether both the specified bounding boxes contain persons or not. The definition queries can be obtained by sampling pairs of bounding boxes from the above definition queries in AQ. Noted, it will be a negative query if only one bounding box contains person. Then, with the ReID annotations in RAP, the positive definition queries are further divided into positive set and negative set in the recognition stage, respectively. Totally, there are over 10 millions definition queries and 100 thousands ReID queries in RQ.

4.3 Analysis Modules and Vision Algorithms

The development of ISEE aims to meet the needs for efficiently parsing large-scale videos with flexible execution of multiple analysis algorithms. Moreover, the ISEE platform is also used to study the evaluation problem of a composable intelligent system containing multiple analysis modules. Thus, to satisfy the above two requirements, we choose an end application on person retrieval to validate the traits of ISEE, which includes three main analysis modules, i.e. pedestrian detection with tracking (D. & T.), attribute recognition (A.R.), and re-identification(ReID). Firstly, the three modules can form various kinds of pipelines including sequential executions and parallel executions, which can validate the flexibility and efficiency of the ISEE for analyzing large-scale video data. Secondly, the three analysis modules can form a well-defined person retrieval system including both attribute-based queries and image-based queries, which can support the subsequent system evaluation naturally. Moreover, based on the collected raw videos, the RAP dataset [31] is annotated that can provide sufficient ground truth labels for large-scale system evaluations on person retrieval. Thus, the aforementioned three analysis modules are selected to instantiate the ISEE. It should be noted that the pedestrian ReID in current ISEE is just to extract ReID features which are then saved as one data property of person nodes in the graph database. To construct the similarity edges between person nodes, a top-k ranking algorithm based on Euclidean distance is implemented over all person pairs detected in different camera views.

TABLE 3
The analysis modules and corresponding algorithms used in ISEE.

Analysis Modules	Algorithms	
D. & T.	Detection	Association
	GMM [56], SSD [8], Faster-RCNN (FRCNN) [7]	Nearest Neighbour (NN)
A.R.	DeepMAR [57], LSPR_attr [58]	
ReID	MSCAN [59], LSPR_reid [58]	

Concretely, the algorithms encapsulated in ISEE are presented in Table 3. For the module of D. & T., the GMM based motion detection with Nearest Neighbour (NN) tracker is performed on CPUs, while the SSD [8] and Faster-RCNN (FRCNN) [7] with NN tracker are performed on GPUs. For A.R. and ReID modules, four state-of-the-art algorithms (DeepMAR [57], LSPR_attr [58], MSCAN [59] and LSPR_reid [58]) are performed, where the deep learning based recognition models are fine-tuned with the training samples in RAP and run on GPUs as well. LSPR_attr and LSPR_reid are the champion algorithms in the competition of large-scale pedestrian retrieval [58].

4.4 Performance Metrics

To measure the performance on person retrieval, the metrics of *Precision*, *Recall* and F_1 *score* are adopted, because the current question-answering diagram can be seen as a series of binary classifications.

For each kind of queries (AQ and RQ), we calculate the overall performance over all queries in both of *definition stage* (*D*) and *recognition stage* (*R*) as weighted sum models in the following.

$$Precision = \frac{\sum_{I\in\{D,R\}} w_I N_I^{tp}}{\sum_{I\in\{D,R\}} w_I N_I^{tp} + \sum_{I\in\{D,R\}} w_I N_I^{fp}} \quad (1)$$

$$Recall = \frac{\sum_{I\in\{D,R\}} w_I N_I^{tp}}{\sum_{I\in\{D,R\}} w_I N_I^{vp}} \quad (2)$$

where the subscript $I \in \{D, R\}$ represents one of the query stages. N_I^{tp} is the number of correctly answered positive queries in the stage I (a.k.a. true positive); N_I^{fp} is the number of wrongly answered negative queries (a.k.a. false positive); N_I^{vp} is the number of valid positive queries, where N_R^{vp} means the number of practical positive queries in the recognition stage which correspond to those positive queries answered correctly in the definition stage; and w_I is the weight parameter to handle the unbalanced distributions of the queries in two querying stages, i.e., $w_I = \frac{1}{N_I^v}$ where N_I^v is the total number of valid queries in the stage I.

Based on the *precision* and *recall*, the comprehensive performance metric F_1 *score* is computed as follows.

$$F_1 = \lambda \cdot \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (3)$$

where λ is a coefficient to award the parsing results which can answer more queries, i.e., proportional to the number of invalid queries in practice. Thus, it can be calculated by Eq. 4, where N_I is the total number of queries in stage I designed in Section 4.2.

$$\lambda = \frac{\sum_{I\in\{D,R\}} w_I N_I^v}{\sum_{I\in\{D,R\}} w_I N_I} \quad (4)$$

It should be noted, for the case study, only two querying stages are involved in person retrieval, which aims to show an example of system-level evaluation for a system application including multiple vision algorithms. In future studies, the above performance metrics can be easily extended to more complicated applications with more than two querying stages, e.g., event retrieval.

5 Experimental Results

5.1 Setups

5.1.1 Assignments of Computing Resources

The ISEE platform is deployed on a cluster which is composed of one master node and five worker nodes. The hardware information of the cluster is listed in Table 4.

TABLE 4
The basic information of worker nodes used in this paper.

Worker Nodes	Node1	Node2	Node3	Node4	Node5	Totally
CPU Model Name (Intel Xeon)	E5-2620 V4				E5-2618L V3	-
CPU Frequency	2.1GHz				2.3GHz	-
#CPU Cores	16					80
Memory	128 GB					640 GB
#GPUs	4					20
GPU Model Name	GeForce GTX Titan X (Pascal)					-
Network Bandwidth	100Mb/s					-

The computing resources assigned to five modules in ISEE are listed in Table 5, where M.H., D.M., D. & T., A.R. and ReID are shorten for message handling, data managing, detection with tracking, attribute recognition and re-identification, respectively. As presented in Section 3, M.H. and D.M. are auxiliary modules and the other three are visual analysis modules. The resources inside each module are scheduled by the platform automatically.

Since the sum of the assigned resources must be not exceed the total available resources in the cluster, the settings on the size of partitions and the number of CPU cores are a trade-off between the degree of parallelism and the available resources. Here, for each module, the number of executors is set to 5 to guarantee that each worker node launches one executor. And the memory size is set empirically to avoid the error of out-of-memory.

TABLE 5
Assignment of computing resources to each module on ISEE.

Resource Items		Modules				
		M.H.	D.M.	D. & T.	A.R.	ReID
#Partition	Kafka	10	10	10	10	10
	RDD	10	10	10	10	10
#Cores	Driver	1	1	1	1	1
	Executor	2	2	2	2	2
Memory	Driver	2GB	2GB	2GB	2GB	2GB
	Executor	6GB	6GB	20GB	16GB	16GB
#Executors		5	5	5	5	5

5.1.2 Execution Pipelines

To validate the extensibility and flexibility of ISEE platform, five execution pipelines are constructed based on different execution plans and vision algorithms. Fig. 8 shows two

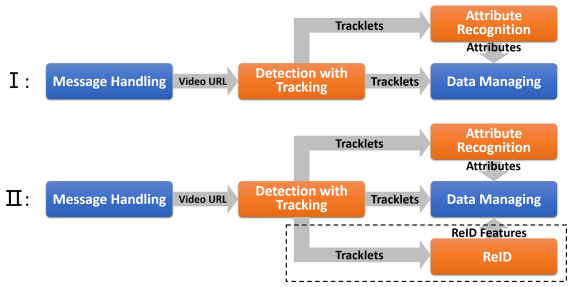


Fig. 8. Two kinds of execution plans. The blue rectangles denote auxiliary modules and the orange ones represent analysis modules.

TABLE 6
Five pipelines for running on ISEE.

Execution Pipeline	Execution Plan	Vision Algorithms of Analysis Modules		
		D. & T.	A.R.	ReID
P1	I	GMM & NN	DeepMAR	-
P2	II	GMM & NN	DeepMAR	MSCAN
P3	II	SSD & NN	DeepMAR	MSCAN
P4	II	FRCNN & NN	DeepMAR	MSCAN
P5	II	FRCNN & NN	LSPR_attr	LSPR_reid

basic types of execution plans (I and II), in which the blue rectangles (*Message Handling* and *Data Managing*) represent the auxiliary modules; while the orange ones are visual analysis modules. The plan I is a simple sequential execution of D. & T. module and A.R. module. The plan II is generated by extending I with a ReID module which is executed in parallel with the A.R. module. Five execution pipelines are further generated with different execution plans and different vision algorithms, as shown in Table 6.

5.2 Efficiency Evaluations on Message Centers

For an extensible and flexible platform, message center is an essential part of ISEE to decouple different analysis modules. Its performance on Reading/Writing (R/W) intermediate data affects the efficiency of ISEE significantly.

Apache Kafka is a highly scalable and distributed messaging system which has been successfully used for decoupling processing from data producers to data consumers in LinkedIn's data pipeline. According to its official introduction of use cases [29], in comparison to most messaging systems Kafka has better throughput, built-in partitioning, replication, and fault-tolerance which makes it a good solution for large scale message processing applications.

Thus, due to the above excellent traits, we chose Kafka as the middle-ware message center for transmitting intermediate data among different applications. Furthermore, with a specific vision application on person retrieval in this work, we also conduct an evaluation on three possible candidates of message centers, i.e., Kafka, MySql and HDFS. We studied their R/W performance along with different data sizes which range from 100KB to 2000KB. For each time, 100 items of messages are written and read through different types of message center. Table 7 shows the elapsed time of reading and writing separately. We can see that the efficiency of Kafka is indeed superior to the other methods remarkably in both reading and writing tasks. Overall, it is reasonable to adopt Kafka as the message center in ISEE platform.

TABLE 7
R/W performances of different methods for message center.

Data Size		Elapsed Time (s)		
		Kafka	MySQL	HDFS
100KB	Read	**0.666**	1.097	2.288
	Write	**0.822**	2.846	6.289
500KB	Read	**2.923**	5.708	9.528
	Write	**5.093**	15.083	17.772
1000KB	Read	**5.908**	10.877	17.411
	Write	**10.325**	28.206	26.246
1500KB	Read	**11.264**	16.165	26.385
	Write	**18.054**	37.418	36.788
2000KB	Read	**20.153**	21.705	34.873
	Write	**25.007**	47.655	48.344

5.3 Efficiency Evaluations on Video Parsing

In this section, we firstly validate the scalability of ISEE by analyzing the time costs with different computing resources. Then, the efficiencies of processing different sizes of video data are shown to demonstrate the capability of ISEE for large-scale video analysis.

Scalability is an essential characteristic of a distributed platform. It reflects the capability to improve the efficiency proportionally to the increasing of computing resources. Here, a set of video files with the total length of about 50 hours(h) are adopted as test samples. And four execution pipelines (P2∼P5) are implemented individually with different numbers of worker nodes. The speedups of elapsed time related to one node are shown in Fig. 9. We can find that the ratios of speedup equal to the ratios of increasing numbers of nodes approximately for all the four pipelines, which suggests the scalability of the ISEE platform.

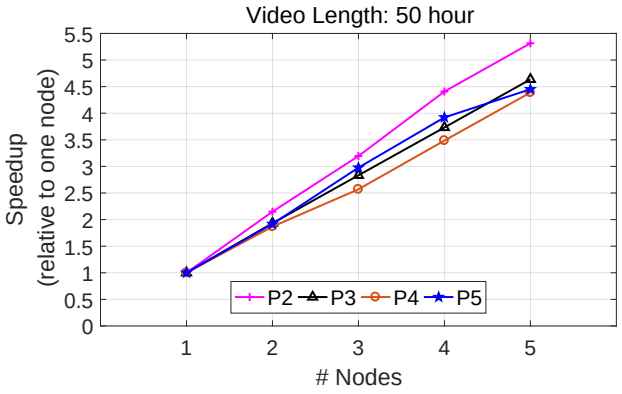


Fig. 9. The relationships of speedup to number of nodes. It reflects the speedups about total cost of each pipeline.

Subsequently, we investigate the relationships between time costs and sizes of video data. We perform five pipelines, i.e., P1∼P5, on ISEE platform with different sizes of video data from 10h to 200h. Fig. 10 and Fig. 11 show the elapsed times of P1 and P2∼P5 with different video lengths respectively. We can find that the time cost (the red lines in the figures) is approximately linear proportional to the amount of video data. And all the three analysis modules in each pipeline have similar time costs. That is because the D.&T. module is the bottleneck in current system, which serves as a preprocessing step to reduce a large amount of redundancies in raw HD video. Comparing the results of

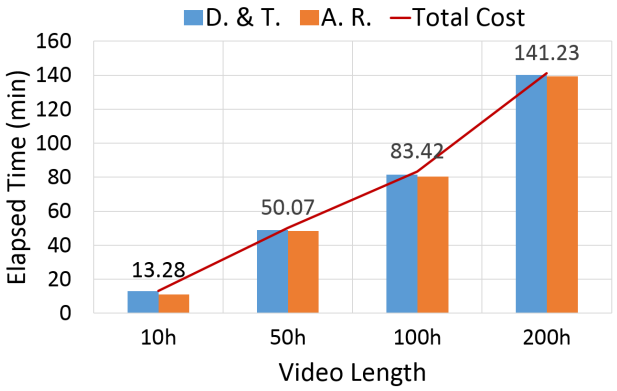
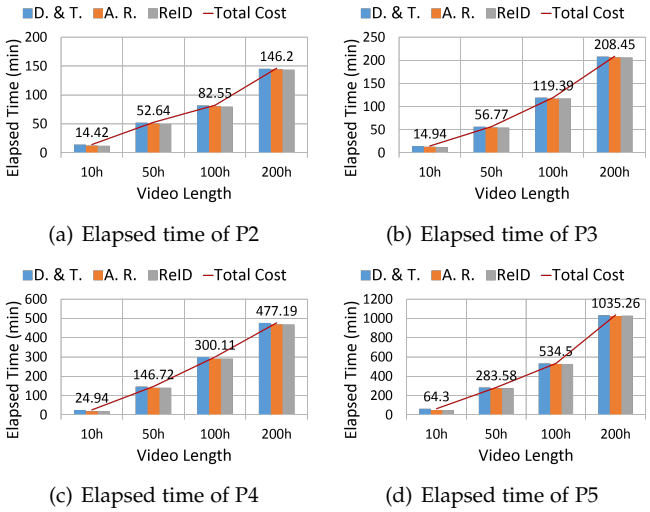


Fig. 10. The relationships of elapsed time to video length with P1 on ISEE. The red line reflects the total elapsed time of each pipeline.



(a) Elapsed time of P2 (b) Elapsed time of P3

(c) Elapsed time of P4 (d) Elapsed time of P5

Fig. 11. The relationships of elapsed time to video length with P2, P3, P4 and P5 on ISEE. The red line also reflects the total elapsed time of each pipeline.

P1 (Fig. 10) and P2 (Fig. 11(a)), they get similar efficiencies as that the same vision algorithms are used in the modules of D. & T. and A.R., and the ReID module with MSCAN can be completed very quickly within 1 second per tracklet. However, different from P1 and P2, P3 (Fig. 11(b)) and P4 (Fig. 11(c)) substitute the GMM in D.&T. with complicated deep learning methods, i.e., SSD and FRCNN respectively, which are more time consuming as the complexities of SSD and FRCNN are much higher than the simple GMM based detector. In addition, compared to P4, the algorithms of LSPR_attr and LSPR_reid in P5 (Fig. 11(d)) have higher computational complexity. The high consuming on GPU resources makes it impossible to share the same GPU with other algorithms. The reduced parallelism increases the cost of P5 significantly.

Finally, the 410h video files (19 million of image frames) are parsed by ISEE with the tasks of pipelines P2∼P5. The totally time costs are 4.4h, 6.8h, 16.6h and 34.1h respectively without frame skipping in processing. Table 8 lists the main technical characteristics of 4 large-scale platforms reported recently as well as ours for a qualitative comparison. Compared to other platforms, under the similar workload for

TABLE 8
Comparisons with other large-scale visual analysis platforms.

Platform	Cluster	Video Length	Analysis Modules	Time Consumption
VTT [39]	10 workers	93.5 hours (1080p)	Object Detection (FRCNN), Tracking, Attributes Recognition, Pose Recognition, Behavior Detection, Multi-view Fusion, etc.	2 Weeks
Optasia [25]	Microsoft's Cosmos System	100 GB	License Plate Recognition, Traffic Flow Mapping, Classification of Vehicles, Object Re-identification	68.35 h
Google [24]	Google's Cloud Platform (40 workers)	912 hours	Shot Boundary Detection	2.9 h
Scanner [46]	16 workers (256 cpu cores, 16 TitanX GPUs)	421GB (35 million frames)	Object Detection (FRCNN, at a stride of 24 frames), Tracking	10.8 h (5.5h+5.3h)
our	5 workers (80 cpu cores, 20 TitanX GPUs)	410 hours (720p, 19 million frames)	Person Detection (GMM) with Tracking, Attributes Recognition (DeepMAR), ReID (MSCAN)	4.4 h
			Person Detection (SSD) with Tracking, Attributes Recognition (DeepMAR), ReID (MSCAN)	6.8 h
			Person Detection (FRCNN) with Tracking, Attributes Recognition (DeepMAR), ReID (MSCAN)	16.6 h
			Person Detection (FRCNN) with Tracking, Attributes Recognition (LSPR_attr), ReID (LSPR_reid)	34.1 h

processing hundreds of hours videos and multiple analysis modules (FRCNN is also used in some of the comparison platforms), the ISEE can achieve a comparable performance on computing efficiency with relative limited computing resources (only 5 worker nodes).

5.4 Performance on Person Retrieval

In this section, we will show the performances of both AQ and RQ, where four execution pipelines, i.e., P2, P3, P4 and P5, are compared as an example on system-level evaluations involving multiple visual analysis modules.

5.4.1 Query Results of AQ

For each AQ, as presented in Sec. 4.2, the definition stage is firstly to check whether a person node can be retrieved from the graph database according to the spatio-temporal condition (a bounding box BB at frame t) or not, where the retrieved person should be detected in frame t and its spatial overlap with BB should be larger than a threshold, i.e., intersection-over-union (IoU) score > 0.5. Then, the recognition stage is executed to check whether the queried attributes are true or false, if the person node can be retrieved correctly. As the larger the threshold of IoU score, the less likely a person node can be found. We investigate the impact of the IoU threshold on the performance of AQ, the precision (Pre.), recall (Rec.) and F_1 score are computed over different IoU thresholds. Their changing trends of the overall results as well as the results in different stages are individually presented in Fig. 13. While the results of λ (the ratio of valid queries in Eq.4 which reflects the trend of recall in the stage of D) is shown in Fig. 12.

The first row of Fig. 13 presents the *Overall* performances in AQ. From Fig. 13(c) we can find that all the four pipelines achieve the peak value of F_1 scores at IoU threshold of 0.6. Meanwhile, P4 shows the best overall performances except for the IoU threshold of 0.9, which is mainly due to its superior values on precision (Fig. 13(a)), recall (Fig. 13(b)) and λ (Fig. 12) as the IoU thresholds ranging from 0.5 to 0.8. While the lower value on λ at IoU threshold of 0.9 results in an inferior performance. Fig. 13(c) also indicates that the pipelines (P3, P4 and P5) with deep learning methods used in D. & T. are inclined to achieve better overall performances than that with traditional method (P2). Especially, the use of FRCNN (P4 and P5) that promotes the overall performances

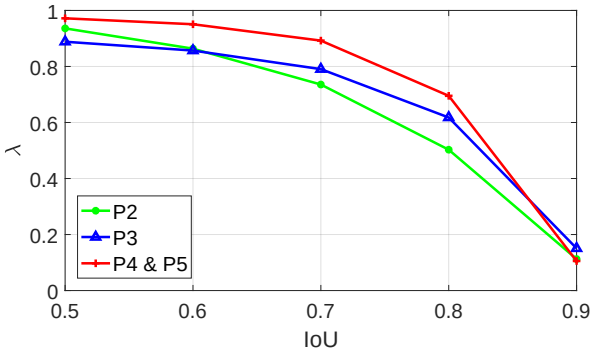


Fig. 12. The ratios of valid queries (λ) about P2, P3, P4 and P5 with various IoU thresholds in AQ.

remarkably with the IoU threshold changing from 0.5 to 0.8. In addition, compared to the results of P4 and P5, the use of LSPR_attr in A.R. drops down the overall performances of P5, which is consistent with the results in [58] evaluated on the RAP dataset.

The performances of *Stage D* and *Stage R* are displayed in the last two rows of Fig. 13. It is shown that the performances of P5 are worse in the stage R, as an inferior attribute recognition model is adopted. As shown in Fig. 13(f), P4 and P5 achieve the highest F_1 scores in the stage D with the IoU thresholds ranging from 0.5 to 0.8. While P3 gets higher values in recall than P2 with larger IoU thresholds, which reflects its superior ability to detect more persons with high-quality bounding boxes than P2. The large superiorities of deep learning methods used in D. & T. of P3, P4 and P5 make them achieve higher recognition performances in the stage R (see Fig. 13(i)). Meanwhile, the much better performances in the stage D combed with the large value of λ result in the superiority of P4 finally.

In addition, from the curves of F_1 scores of the four pipelines, we can find that the F_1 scores in the stage R (Fig. 13(i)) are improved monotonously along with the increasing of IoU threshold which results in high-quality detected bounding boxes; while the F_1 scores (Fig. 13(f)) in the stage D increase firstly and then go down with larger IoU thresholds due to the increasing of miss retrievals. The performance of stage D has a consistent trend with the overall performance (Fig. 13(c)), which displays the key role of the detection module on the performance of whole

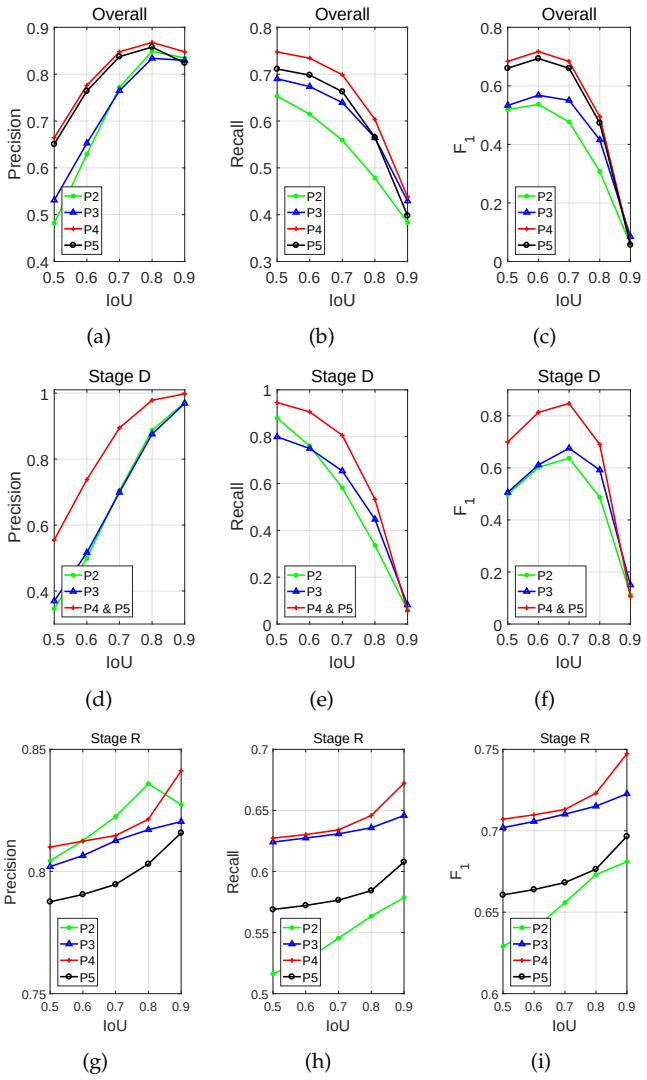


Fig. 13. Performances of P2, P3, P4 and P5 with various IoU thresholds in AQ. (a), (b) and (c) reflect the changing trends on Precision, Recall and F_1 about the overall performances in AQ; (d), (e) and (f) are about Stage D; (g), (h) and (i) are about Stage R.

system. Meanwhile, we can also find that the best overall performances of the two pipelines are inferior to the peak values of both stage D and stage R. It reflects the rationality of designed metric that the final overall performance is usually limited by the one with worst performance in a pipeline, which is known as Liebig's law of the minimum [60]. And it also indicates that is a challenging problem to integrate multiple analysis components into a composable visual intelligent system.

5.4.2 Query Results of RQ

Besides AQ on unary object retrieval in large-scale videos, we also test the ISEE platform with RQ to retrieve binary relations between two visual objects.

As presented in Sec. 4.2, in the definition stage of RQ, the two bounding boxes as querying condition are firstly verified in the parsing results. If and only if a positive definition query (both bounding boxes in the query indeed contain persons) is responded by two correct person nodes from the graph database (a.k.a., a valid query in the next recognition stage), we will further inquire whether one ReID path within certain limited steps exists between the two persons. In our experiments, we construct the similarity edges for each person node with its 10 nearest neighbors based on the ReID features. And the maximal number of steps (path length) is set to 3. In practice, we calculate the different metrics of RQ for each path length.

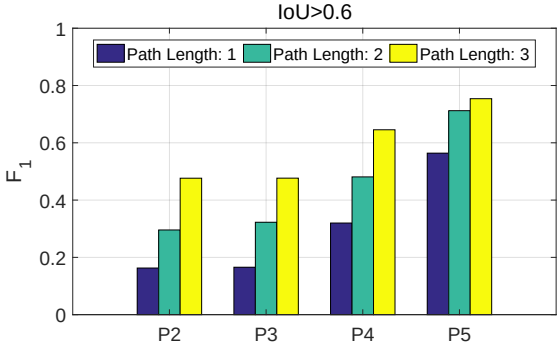


Fig. 14. Metric results of RQ with different path lengths of P2, P3, P4 and P5, where the IoU threshold is 0.6.

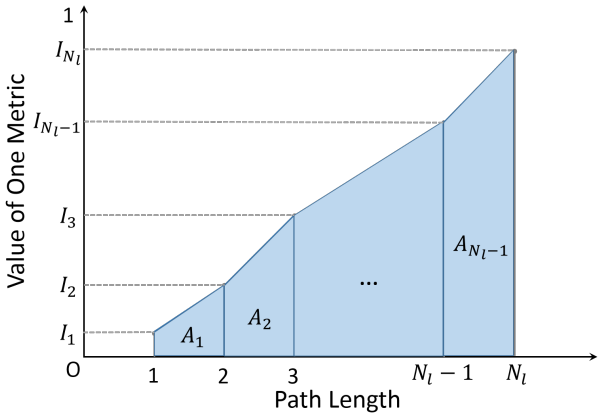


Fig. 15. An illustration on the equation to calculate the average values of different metrics used in RQ.

Fig. 14 shows us the changing trends of F_1 scores along with the path length ranging from 1 to 3 at the IoU threshold of 0.6. As expected, we can find that the F_1 scores increase monotonically along with the path length. Similar situations can also be obtained with other IoU thresholds. To measure the performances, we define the average metric with Eq. 5 which reflects the shadow area under the score curve. An illustration is shown in Fig. 15, where the I_i is the value of a metric with one specific path length; N_l is the maximal path length; and A_i denotes the area under the curve; $\frac{1}{N_l-1}$ is the normalization term. The curves shown in Fig. 17 presents the calculated average values of different metrics.

$$\begin{aligned} Average &= \frac{1}{N_l-1}\sum_{i=1}^{N_l-1} A_i \\ &= \frac{1}{2(N_l-1)}\sum_{i=1}^{N_l-1}(I_i+I_{i+1}), N_l \geq 2 \end{aligned} \tag{5}$$

The first row of Fig. 17 also presents the *Overall* performances on precision, recall and F_1 score in RQ. While the values of λ, which reflects the ratio of valid queries in RQ,

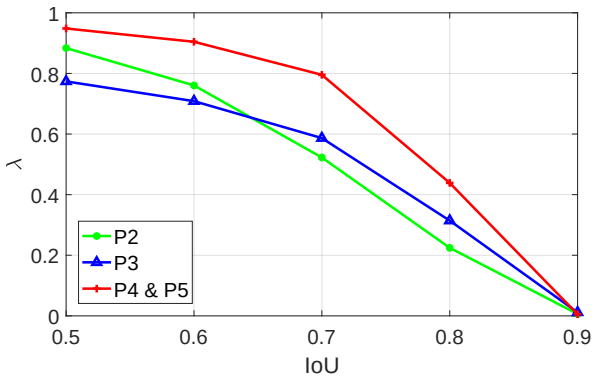


Fig. 16. The ratios of valid queries (λ) about P2, P3, P4 and P5 with various IoU thresholds in RQ.

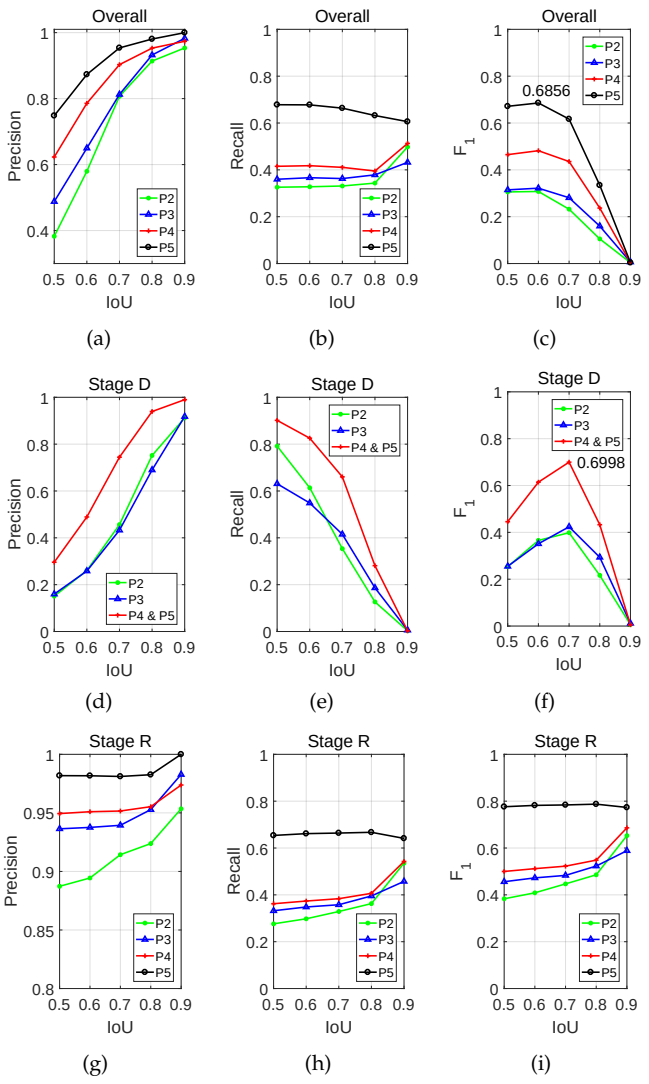


Fig. 17. Performances of P2, P3, P4 and P5 with various IoU thresholds in RQ. (a), (b) and (c) reflect the changing trends on Precision, Recall and F_1 about the overall performances in RQ; (d), (e) and (f) are about Stage D; (g), (h) and (i) are about Stage R.

are shown in Fig. 16. As shown in Fig. 17(c), we can get similar conclusion with AQ from these results that the peak performances (F_1) of the four pipelines are at IoU threshold of 0.6. And the pipelines (P3, P4 and P5) whose D.&T. is realized with deep model also present better performances than those of the traditional one (P2). However, different from AQ, P5 achieves the best results as the LSPR_reid is obtained in the ReID module whose performances are superior to the MSCAN significantly which is used in other pipelines.

We also explore the performances of *Stage D* and *Stage R* in isolation. Their results are plotted in the last two rows of Fig. 17, respectively. We can find (see Fig. 17(f), Fig. 17(d) and Fig. 17(e)) that P4 and P5 get better performances (F_1) than P2 and P3 in the stage D as the higher precision and recall are achieved. Furthermore, as the use of strong ReID algorithm, the P5 produces much higher performance in the stage R combined with the better results in the Stage D, so as to achieve the best overall performance.

The overall F_1 score as well as the F_1 scores of individual stages (D and R) of the four pipelines are plotted in Fig. 17(c), Fig. 17(f) and Fig. 17(i) respectively. For all of the four pipelines, the trends of F_1 scores of stage D & R and overall are similar to those of AQ in Fig. 13. It also reflects the dominant effect of detection algorithm in stage D on the final overall performance. While the overall performances in RQ are much lower than those of AQ except for P5. It is because that the miss detections of queried persons will lead to more serious effects on RQ than AQ, as two persons are involved in a RQ query rather than only one person in an AQ query. However, the use of strong ReID algorithm in P5 alleviates the issue to some extent which leads to its high performance in RQ. Meanwhile, the best overall performances of the four pipelines are also inferior to the peak values of both the stage D and the stage R, which again indicates the rationality of designed performance metric.

5.5 Remarks

From the above extensive experiments on both system efficiency and person retrieval performance, some main characteristics and findings are summarized as follows.

- As a large-scale visual exploration platform, the ISEE integrates multiple visual analysis modules with the de-coupling design of middle-ware message center into a Spark Streaming based computing framework, which enables the system to perform various visual analysis pipelines flexibly and efficiently. Compared to other popular large-scale visual analysis platforms (Table 8), the ISEE achieves comparable high efficiency (more than 90 times speed-up compared to raw video length) with only 5 worker nodes.
- As an example of system-level evaluation, we adopt the ISEE to perform an end-to-end application on person retrieval. From the experimental results in Sec. 5.4, we find that person detection is the bottleneck of system performance, which is the most time consuming part of whole video parsing process (Fig. 10 and Fig. 11). Moreover, the overall performance on person retrieval is mainly affected by the results of person detection (the third columns of Fig.13 and Fig.17), where the threshold of IoU in person detection plays an important role for overall performance for both query types.
- The comparisons among various pipelines with three pedestrian detection models, i.e., GMM, SSD and

FRCNN, indicate that the latter two algorithms with deep models provide more accurate detection bounding boxes, so that superior performances of subsequent person retrieval based on attributes or person images can be obtained (Fig. 13 and Fig. 17). Especially, the combination of FRCNN and strong ReID model (LSPR_reid) improve the overall performance in RQ remarkably. It indicates that the combination of superior models is inclined to obtain robust performance in a composable intelligent system. However, the efficiency of the robust models are usually more time- and resources-consuming (Table 8). How to make a compromise between the accuracy and efficiency is still an open issue in constructing such composable AI system.

- As shown in the baseline system evaluation results on person retrieval, the best overall performance of two kinds of queries are lower than 70% in the wild surveillance scenes, which are still not satisfying for real applications. Although the deep learning based algorithms have achieved high performance for many single visual recognition tasks, it is still a challenging problem to integrate multiple analysis components into a composable AI system [19].

6 Conclusion

In this paper, ISEE, a large-scale visual scene exploration and evaluation platform is established by utilizing recent advanced big data tools based on heterogeneous computing resources (CPUs and GPUs). Combining with the Spark Streaming computing framework, the decoupling design of middle-ware message center, the Yarn based scheduling strategy as well as the highly customized task flow enable ISEE to integrate multiple kinds of visual analysis modules and execute various analysis pipelines flexibly and efficiently. To validate the effectiveness of ISEE, we perform an end-to-end system application on person retrieval, where hundreds of hours of surveillance videos with millions of image frames are parsed into a semantic graph with millions of entity nodes and relationships, by multiple kinds of visual analysis modules. As a case study of ISEE system evaluation, we compare different execution plans involving multiple visual analysis modules from both efficiency and performance. The efficiency of video parsing is measured by time costs and the performance of person retrieval is evaluated by millions of person queries based on attributes and visual similarity. The system-level evaluations may inspire researchers to improve visual analysis algorithms and combining strategies from the view of a composable intelligent system.

In future work, we will further extend the ISEE with more visual analysis modules and study the combinational optimization problem to improve the efficiency and the performance of overall system. Moreover, facing the big video data rising from real open environments, we will introduce a new evaluation mechanism based on "human-in-the-loop" to adopt users' feedbacks to evaluate the system performance. In future, a module of adaptive learning from increasing meta-data and users' feedbacks will be added, so that the ISEE can be evolved as a life-long learning vision system.

Appendix

TABLE 9
Acronyms used in this paper.

Acronym	Full Name
AI	Artificial Intelligence
App.	Application
AQ	Attribute-based Query
BB	Bounding Box
D	Definition stage in the query
DAG(s)	Directed Acyclic Graph(s)
DLL(s)	Dynamic Link Library(Libraries)
D.M.	Data Managing
D. & T.	Detection with Tracking
DStream	Discredited Stream
E3	Efficiency, Extensibility, and Evaluation
fp	false positive
FRCNN	Faster-RCNN
GMM	Gaussian Mixture Model
h	hour(s)
HDFS	Hadoop Distributed File System
IoU	Intersection-over-Union
ISEE	Intelligent Scene Exploration and Evaluation
IVS	Intelligent Video Surveillance
JNI(s)	Java Native Interface(s)
M.H.	Message Handling
min	minute(s)
MPI	Message Passing Interface
NN	Nearest Neighbour
QA	Question-Answering
R	Recognition stage in the query
RDD(s)	Resilient Distributed Dataset(s)
ReID	Re-Identification
RQ	relationship-based query
R/W	Reading/Writing
S3	Smart Surveillance System
tp	true positive
UI	User Interface
VSAM	Video Surveillance and Monitoring
VTT	Visual Turing Test
W4	Who, When, Where and What

TABLE 10
Variables and parameters used in this paper.

Variable/ Parameter	Meaning	Value
N_I^{tp}	# correctly answered positive queries in the stage I	$I \in \{D, R\}$
N_I^{fp}	# wrongly answered negative queries in the stage I	$I \in \{D, R\}$
N_I^{vp}	# valid positive queries in the stage I	$I \in \{D, R\}$
N_I^{v}	# valid queries in the stage I	$I \in \{D, R\}$
N_I	total number of queries in stage I	$I \in \{D, R\}$
w_I	the weight parameter to handle the unbalanced distributions of the queries	$I \in \{D, R\}$
λ	a coefficient to award the parsing results which can answer more queries	-
N_l	the maximal path length	3 in this paper
I_i	the value of a metric with one specific path length	$i \in \{1, ..., N_l\}$
A_i	the area under the curve shown in Fig. 15	-

Acknowledgments

This work is funded by the National Key Research and Development Program of China (2016YFB1001005) and the National Natural Science Foundation of China (Grant No. 61473290).

REFERENCES

[1] K. Huang, X. Chen, Y. Kang, and T. Tan, "Intelligent visual surveillance: A review," *Chinese Journal of Computers*, vol. 38, no. 6, pp. 1093–1118, 2015.

[2] T. Huang, "Surveillance video: The biggest big data," *Computing Now*, vol. 7, no. 2, pp. 82–91, 2014.

[3] A. C., N. Jr., and W. R. Schwartz, "A scalable and flexible framework for smart video surveillance," *Computer Vision and Image Understanding*, vol. 144, pp. 258–276, 2016.

[4] I. Haritaoglu, D. Harwood, and L. S. Davis, "W/sup 4: Who? when? where? what? a real time system for detecting and tracking people," in *Proc. IEEE International Conference on Automatic Face and Gesture Recogntion*, 1998, pp. 222–227.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, 2016, pp. 21–37.

[9] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *European conference on computer vision*, 2004, pp. 28–39.

[10] X. Song, J. Cui, H. Zha, and H. Zhao, "Vision-based multiple interacting targets tracking via on-line supervised learning," in *European Conference on Computer Vision*, 2008, pp. 642–655.

[11] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5620–5629.

[12] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1832–1845, 2010.

[13] P. Ochs and T. Brox, "Higher order motion models and spectral clustering," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 614–621.

[14] Z. Zhang, K. Huang, T. Tan, P. Yang, and J. Li, "Red-sfa: Relation discovery based slow feature analysis for trajectory clustering," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2016, pp. 752–760.

[15] C.-C. Tseng, J.-C. Chen, C.-H. Fang, and J.-J. J. Lien, "Human action recognition based on graph-embedded spatio-temporal subspace," *Pattern Recognition*, vol. 45, no. 10, pp. 3611–3624, 2012.

[16] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 436–450, 2012.

[17] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor, "Glimpse clouds: Human activity recognition from unstructured feature points," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, June 2018.

[18] Y.-l. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, and C.-f. Shu, "Ibm smart surveillance system (s3): event based video surveillance system with an open and extensible framework," *Machine Vision and Applications*, vol. 19, no. 5, pp. 315–327, 2008.

[19] I. Stoica, D. Song, R. A. Popa, D. A. Patterson, M. W. Mahoney, R. H. Katz, A. D. Joseph, M. Jordan, J. M. Hellerstein, J. Gonzalez, K. Goldberg, A. Ghodsi, D. E. Culler, and P. Abbeel, "A berkeley view of systems challenges for ai," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2017-159, Oct 2017. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-159.html

[20] M. M. Mubasher, M. S. Farid, A. Khaliq, and M. M. Yousaf, "A parallel algorithm for change detection," in *Proc. IEEE International Multitopic Conference*, 2012, pp. 201–208.

[21] H. Tan and L. Chen, "An approach for fast and parallel video processing on apache hadoop clusters," in *Proc. IEEE International Conference on Multimedia and Expo*, 2014, pp. 1–6.

[22] S. Yang and B. Wu, "Large scale video data analysis based on spark," in *Proc. IEEE International Conference on Cloud Computing and Big Data*, 2015, pp. 209–212.

[23] K. Hong, M. Voelz, V. Govindaraju, B. Jayaraman, and U. Ramachandran, "A distributed framework for spatio-temporal analysis on large-scale camera networks," in *Proc. IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2013, pp. 309–314.

[24] Y. Wang, W.-T. Chen, H. Wu, A. Kokaram, and J. Schaeffer, "A cloud-based large-scale distributed video analysis system," in *Proc. IEEE International Conference on Image Processing*, 2016, pp. 1499–1503.

[25] Y. Lu, A. Chowdhery, and S. Kandula, "Optasia: A relational platform for efficient large-scale video analytics," in *Proceedings of the Seventh ACM Symposium on Cloud Computing*, 2016, pp. 57–70.

[26] L. Zheng, H. Zhang, S. Sun, M. Chandraker, and Q. Tian, "Person re-identification in the wild," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1367–1376.

[27] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth *et al.*, "Apache hadoop yarn: Yet another resource negotiator," in *Proc. of the 4th annual Symposium on Cloud Computing*, 2013.

[28] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: An efficient and fault-tolerant model for stream processing on large clusters," *HotCloud*, vol. 12, 2012.

[29] Apache kafka. [Online]. Available: http://kafka.apache.org/

[30] J. Webber, "A programmatic introduction to neo4j," in *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*, 2012, pp. 217–218.

[31] D. Li, Z. Zhang, X. Chen, and K. Huang, "A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1575–1590, 2019.

[32] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[33] N. T. Siebel and S. Maybank, "The advisor visual surveillance system," in *ECCV 2004 workshop applications of computer vision*, vol. 1, 2004.

[34] K. Huang and T. Tan, "Vs-star: A visual interpretation system for visual surveillance," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2265–2285, 2010.

[35] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt *et al.*, "A system for video surveillance and monitoring," *VSAM final report*, pp. 1–68, 2000.

[36] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A high-performance, portable implementation of the mpi message passing interface standard," *Parallel computing*, vol. 22, no. 6, pp. 789–828, 1996.

[37] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[38] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," *HotCloud*, vol. 10, no. 10-10, 2010.

[39] H. Qi, T. Wu, M.-W. Lee, and S.-C. Zhu, "A restricted visual turing test for deep scene and event understanding," *arXiv preprint arXiv:1512.01715*, 2015.

[40] X. Zhao, H. Ma, H. Zhang, Y. Tang, and Y. Kou, "Hvpi: extending hadoop to support video analytic applications," in *Proc. IEEE International Conference on Cloud Computing*, 2015, pp. 789–796.

[41] G. Li, X. Li, F. Yang, J. Teng, S. Ding, Y. F. Zheng, D. Xuan, B. Chen, and W. Zhao, "Traffic at-a-glance: Time-bounded analytics on large visual traffic data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 9, pp. 2703–2717, Sept. 2017.

[42] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proc. IEEE International Symposium Mass storage systems and technologies*, 2010, pp. 1–10.

[43] H. Wang, X. Zheng, and B. Xiao, "Large-scale human action recognition with spark," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2015, pp. 1–6.

[44] 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*, 2016, pp. 309–318.

[45] X. Guo and Y. Cao. (2016, Feb) Online security analytics on large scale video surveillance system. Spark Summit East 2016. [Online]. Available: https://spark-summit.org/east-2016/events/online-security-analytics-on-large-scale-video-surveillance-system/

[46] A. Poms, W. Crichton, P. Hanrahan, and K. Fatahalian, "Scanner: Efficient video analysis at scale," in *International Conference on Computer Graphics and Interactive Techniques*, vol. 37, no. 4, 2018, pp. 1–13.

[47] (2017) tscanner: Distributed video processing in tensorflow. [Online]. Available: https://www.andrew.cmu.edu/user/matthief/proposal.html

[48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[49] X. Yang, H. Zhang, H. Ma, W. Li, G. Fu, and Y. Tang, "Multi-resource allocation for virtual machine placement in video surveillance cloud," in *International Conference on Human Centered Computing*, 2016, pp. 544–555.

[50] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "Openstack: toward an open-source solution for cloud computing," *International Journal of Computer Applications*, vol. 55, no. 3, 2012.

[51] P. Venetianer and H. Deng, "Performance evaluation of an intelligent video surveillance system - a case study," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1292 – 1302, 2010, special issue on Embedded Vision.

[52] Y. Xu, B. Ma, R. Huang, and L. Lin, "Person search in a scene by jointly modeling people commonness and person uniqueness," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 937–940.

[53] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3415–3424.

[54] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012.

[55] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," *Proceedings of the National Academy of Sciences*, vol. 112, no. 12, pp. 3618–3623, 2015.

[56] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 246–252.

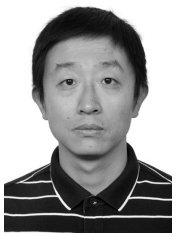
[57] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," in *2015 3rd IAPR Asian Conference on Pattern Recognition*, 2015, pp. 111–115.

[58] D. Li and Z. Zhang, "Large-scale pedestrian retrieval competition," *arXiv preprint arXiv:1903.02137*, 2019.

[59] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for pedestrain re-identification," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2017, pp. 384–393.

[60] F. B. Salisbury and C. W. Ross, Eds., *Plant Physiology*, 4th ed. Belmont, California: Wadsworth, 1992.

Da Li received the M.Eng. degree in electronics and communication engineering from Suzhou Institute of Nano-Tech and Nano-Bionics, Chinese Academy of Sciences, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Artificial Intelligence, University of Chinese Academy of Sciences, China, and the Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation, Chinese Academy of Sciences. His research interests include big visual data and video surveillance.



Zhang Zhang received the B.S. degree in computer science and technology from Hebei University of Technology, Tianjin, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China in 2008. Currently, he is an associate professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include activity recognition, video surveillance, and time series analysis. He has published a number of papers at top venues including the IEEE Transactions on Pattern Analysis and Machine Intelligence, CVPR, and ECCV.

Kai Yu received the B.S. degree in computer science from Beihang University, China, in 2016. Currently he is pursuing master degree in Carnegie Mellon University, and taking an internship at Facebook. His research interests include video surveillance, visual tracking and simultaneous localization and mapping.

Kaiqi Huang received the M.S. degree in electrical engineering from Nanjing University of Science and Technology, Nanjing, China, and the Ph.D. degree in signal and information processing from Southeast University, Nanjing. He is currently a Professor in Institute of Automation, Chinese Academy of Science (CASIA). He has published more than 150 papers on TPAMI, TIP, TCSVT, TSMCB, CVIU, Pattern Recognition and ICCV, CVPR, and ECCV. His interests include visual surveillance, image and video analysis, human vision and cognition, computer vision, etc. Dr. Huang is IEEE Senior Member and Program Committee Member of more than 50 international conferences and workshops, he also severed several AEs of journals such as IEEE Systems, Man, and Cybernetics: Systems.

Tieniu Tan received the B.Sc. degree in electronic engineering from Xian Jiaotong University, China, in 1984, and the M.Sc. and Ph.D. degrees in electronic engineering from Imperial College London, U.K., in 1986 and 1989, respectively. He is currently a Professor with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China. His research interests include biometrics, image and video understanding, information hiding, and information forensics. He is a fellow of CAS, TWAS, BAS, IAPR, and the U.K. Royal Academy of Engineering, and Past President of the IEEE Biometrics Council.