EAI Endorsed Transactions
on Context-aware Systems and Applications Research Article **EAI.EU**

A Scalable IoT Video Data Analytics for Smart Cities

Mien Phuoc Doan[1], Vu The Tran[2], Hung Huu Huynh[3], Hiep Xuan Huynh[4,*]

[1]Tra Vinh University. 126 Nguyen Thien Thanh Street, Ward 5, Tra Vinh City, 87000, Vietnam
[2]Institute for Research and Executive Education, The University of Da Nang. 158A Le Loi Street, Hai Chau District, Da Nang City, 550000, Vietnam
[3]Da Nang University of Science and Technology, The University of Da Nang. 54 Nguyen Luong Bang Stress, Lien Chieu District, Da Nang City, 550000, Vietnam
[4]College of Information and Communication Technology, Can Tho University. Campus2, 3/2 Street, Ninh Kieu District, Can Tho City, 900000, Vietnam

Abstract

The smart city is a comprehensive application of information resources and a high degree of information technology integration. With the technical support from IoT (Internet of things), smart city need to have three features of being instrumented, interconnected and intelligent. IoT provides the ability to manage, remotely monitor and control devices from massive streams of real-time data.Our model offers a scalable IoT video data analytics applications for Smart cities to end users, who can exploit scalability in both data storage and processing power to execute analysis on large or complex datasets. This model provides data analytics programming suites and environments in which developers and researchers can design scalable analytics services and applications. A cloud/edge-based automated video analysis system to process large numbers of video streams, where the underlying infrastructure is able to scale based on the number of camera devices and easy to integrate analytic application. The system automates the video analysis process and reduces manual intervention. The design of our model is developed to be easily extended for new kinds of IoT devices, message routing and queueing, and data analytics, to permit specific application to be programmed via the paradigm to be flexible yet simple.

Received on 14 October 2019; accepted on 13 November 2019; published on 29 November 2019

Keywords: Scalable, video data analytics, smart city

doi:10.4108/eai.13-7-2018.163136

1. Introduction

IoT applications offer more value when incorporating video analytics, since the technology allows to consider a wider range of inputs and make more sophisticated decisions. In other words, The video-analytics applications are now capable of gathering and analyzing video footage from multiple sources, thereby generating more detailed insights.

Big video data refers to massive, heterogeneous that is difficult to process using traditional data management tools and techniques. Interoperability is a main issue in large-scale applications that use resources such as data and computing nodes. Standard formats and models are needed to support interoperability and ease cooperation among teams using different data formats and tools. Several video data analytics platforms like Wang [1], You [2], Afzal [3], Charalampidis [4] have been proposed addressing issues like scalability, privacy,..., etc. The evolution of these platforms, however, created isolated systems with limited interoperability. Coping with and gaining value from big video data requires novel architecture and innovative analytics techniques. Our work in this paper is motivated by the lack of a right platform that enables a systematic method to collect traffic data though various sources and more nuanced video analyses for developing countries. The proposed system provides

[*]Corresponding author. Email: hxhiep@ctu.edu.vn

scalable and automated classification of objects in a large number of video streams.

The remainder of the paper is organized as follows. Section 3 introduces the motivating scenario and challenges. Section 4 presents the conceptual architecture of our platform. Section 6 presents our prototype and validates our platform through various experiments. Related work is presented in Section 2. Finally, in Section 7 we conclude the paper and outline future work.

2. Related Work

2.1. IoT Video platforms

A number of platforms have been introduced for real-time video analytics. Yu [2] proposes a framework integrating the complete chain of processing from the initial capture of video to the visual content search. Wang [1] presents a scalable, privacy-aware architecture for large camera networks. Jain [5] promises to allow cameras that were originally deployed for a single application to be simultaneously used for other analytics applications, this work demonstrates the broader uses of existing cameras, and how they can be enabled by automated camera control. Sabokrou [6] designs localized video representations that enable anomaly detection in crowded scenes. Despite recent advances, video-analytics platform for surveillance still have limitations in terms of systematic accessing to large data sets and more nuanced video analyses.

The author Jain [7], Gómez [8], Bhomble [9], Charalampidis [4], Afzal [3] and Ray [10] proposed a Smart Surveillance Monitoring System for motion detection and a remote camera for video recording.

Terroso-Saenz [11] developed tools that are able to provide personalized real time feedback to change behaviors. In study of Kim [12] and Yang [13] focused on smart city including smart technology, smart industry, smart services, smart management and smart life. Humayun [14] mentioned the concept of smart city. A smart city is a multidimensional term including a smart economy, smart mobility, smart living, smart environment, smart people, and smart governance. In research [15] proposing a solution based on a new protocol to address the stated issues by extending the recent research presented by Lewis protocol et al. The proposed protocol is aimed to provide good data privacy, reduces the data security attacks, improving data management and the computation efficiency at all channels.

2.2. Video-analytics Applications for Transport Services

For transport services, the video-analytics applications can notify you of traffic jams or optimize your driving path. It can detect who drove on a red light and only capture and save that exact moment. The transition from smart to intelligent cameras makes the analytics select only the necessary data to be saved for later investigation. It can also recognize a license plate.

Vehicle recognize and traffic density estimation algorithms in traffic are of interest to many researchers such as Rabbouch [16] using cluster analysis for automatic vehicles counting and recognizing. Sanchez [17] proposed algorithm's accuracy is usually determined by comparing, on each video-sequence, the visual inspection count with the automatic count that each algorithm provides.

A virtual loop-based method is proposed to improve the quality of vehicle counting method [18]. Background subtraction is used to detect the moving vehicles and an improved Gaussian mixture model. [19] presents an approach in real-time vehicle detection and tracking using Raspberry Pi. [20] proposed vehicle detection and counting method by way sub-sample video frames using particles to reduce the number of pixels that must be processed.

Some studies of pedestrian detection should mention works of the author [21–23]. In which, Kocak [21] used real-time directional algorithms implemented by compute unified device architecture to detect and count people. [22] used adaptive modeling of detected blobs, where the background model around stationary blobs is not updated. Besides, Tay [23] presents research issue about camera public, private and human behavior using machines learning for behavior detection, behavioral change, behavioral development and contexts of behaviors.

3. Motivation

The smart city integrates information and communication technology, and the IoT to optimize the efficiency of city operations and services to citizens. Smart cities need smart transport services. Proper movement of people, goods and services accelerate the growth and development of a region. A well planned and efficiently managed transport system is a must for any smart city system.

A distinguishing characteristic between transport services in Vietnam to those in developed countries is two-wheeled vehicles comprised a high percentage of the road traffic system. Vietnam is motorcycle-dominated traffic flow, which is very much different from car traffic flow due to the motorcycles' distinguishing characteristics. Most road traffic accidents in Motorcycle-dominated traffic flow countries are caused by motorcycles. Motorcyclists are also classified as vulnerable users along with pedestrians and bicyclists because safety equipment for motorcyclists is not as adequate as equipment for car drivers[24]. Issues related with motorcycle-dominated traffic are as follows:

- Drivers just obey the lane separation when there are policemen;
- Lane separating by type of vehicle increases the number of conflict at intersection areas, thus increasing the risk of local congestion at intersection areas. Lane separation according to vehicle type could be used in cases of low traffic volumes;
- Increases in the amount of conflict at intersections increase waiting time and travel time at intersections;
- High densities of connections with main routes decrease the effect of lane separation;
- Unbalanced traffic volume distribution on lanes during peak hours sometimes “forces” road users to make lane violations.

According to the National traffic safety committee Vietnam report, in the year 2017, Vietnam has more than 20,000 traffic accidents, in which 8,000 died and 17,000 wounded. The progress of transportation systems increase is caused a dramatic increase in demand for smart systems capable of monitoring traffic and street safety.

Da Nang is Vietnam’s third largest city, and is located on the Eastern Sea coast, midway between Hanoi and Ho Chi Minh City and the largest city of Central Vietnam. The city of 1.1 million people, has roughly 600,000 cars and about 800,000 motorbikes and scooters in the year 2017. The explosion of new vehicles is stretching the city’s infrastructure, and Da Nang’s population is forecast to reach 2.5 million by 2030. The city lacks a feasible solution to curb congestion on the current road layout as the result of rapid urbanization.

Despite the magnitude of this problem from the city’s transport department, the data collection system for monitoring and handling traffic is inconsistent and incomplete in Danang city, though various sources of realtime data that can be utilized for solving traffic problems such as GPS signals from buses, taxis, videos from surveillance cameras, etc. are available. New protocols, architectures, and services are in dire needs to respond for these challenges.

4. System architecture

We have developed an architecture for the IoT Video Data Analytics, named as HAIVAN-CVA, shown in Fig. 1. The architecture allows running large-scale distributed workflows on heterogeneous platforms along with software components developed using different programming languages and tools. The core of the architecture is a set of services, including IoT networks/ edge systems, middle ware in clouds/edge systems, and analytics in clouds/edge systems.

4.1. Cameras as IoT Data Services

An IoT Data Service manages various IoT devices (camera). The IoT Data Service has a set of REST APIs which allows to return the data or the URI of the data based on request. A provider might offer different IoT Data Service. Each Data Service might manage many IoT Devices, thus the service provides various types data.

When an IoT Data Service returns the URI of data, it will put the URI into a queue through which the client (e.g., video analytics) will receive the URI. The URI of the data indicates the location of the data. Note that the location of the data might be within the IoT Data Service or any place where the IoT Data Service can push the data for the client.

The IoT Data Service publishes IoT datapoints to HINC [25], from HINC, one can search many service providers

4.2. Data Discovery, Provisioning and Management

rsiHub is used for data discovery. The purpose of using rsiHub is managing diverse resources in IoT systems at large scale extensible enough to accommodate information from various IoT resources with different data models. Our system aims to support programmers and software-agents who need a better way to program and control the IoT resources. An example is we need to (i) query and change capabilities of a set of devices, (ii) create a slice of network functions to sustain the data produced by input device due to the change in (i), and (iii) increase cloud storage to process the video data. At the client, users build a template of data point, which can include extended models, e.g. the video to capture the video rate. Users then need to define the communication channel by using the rsiHub, from which they can create and send a query and to obtain a list of data points. To send a control back to the resources, a control point (which is available and associated with the data point) wraps the change rate operation and transfers it to underlying management services to execute. Similarly, the code shows how the user can query network and cloud services.

There are two ways that users can be collected data from IoT Data Providers when they request it. The client can get/subscribe data from IoT Data Provider (IDP), when IDP accepts the request, then returning data/url for client. The client has storage mechanism data/url to serve for analytics in cloud/edge systems. Message will be stored in Advanced Message Queue Protocol.

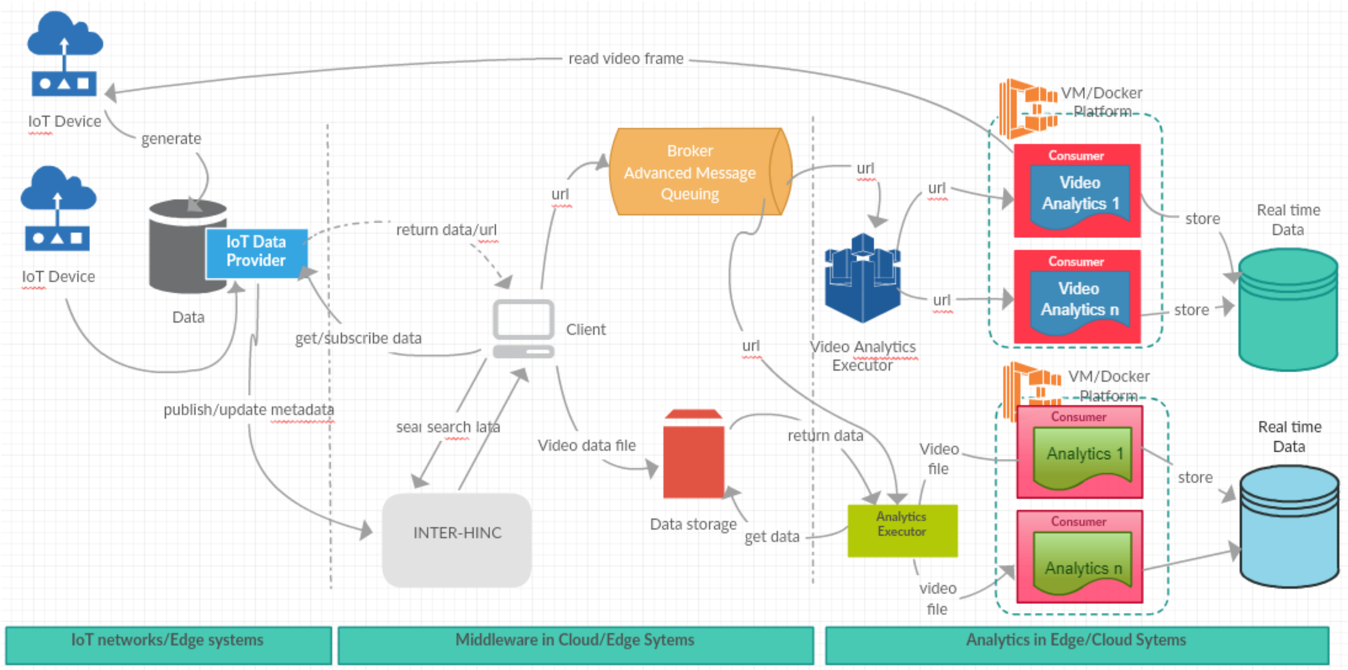


Figure 1. System architecture

4.3. Video Analytics in Clouds/Edge Systems

Video analytics component listens a queue (AMQP) to receive the URI of video. When receiving an URI, the component will download the video, if needed (e.g., when the URI indicates the data remotely). Then the component calls a plug-in which performs the analytics. The result is stored in a MongoDB/Redis for real time analytics. This component will be deployed on demand in many virtual machines/containers. We have n instances of the component running and if needed we can create more or remove instances.

5. Scalable Video Analytics

The result in the generation of thousands of high resolution videos streams are due to increasing availability and deployment of video cameras. The input data our project usually video data, so it is often so large that manually inspecting it for useful content can be time consuming and error prone, thereby it requires automated analysis to extract useful information and metadata. Such videos can be subdivided into a number of frames of interest. We using intelligent approaches to extract Various types of information due to the optimized processing of these video frames from the term video data analytics

Existing video analysis systems lack of scalability, integration and automation. A cloud/edge-based automated video analysis system to process large numbers of video streams, where the underlying infrastructure is able to scale based on the number of camera devices and easy to integrate analytic application. The system automates the video analysis process and reduces manual intervention.

A big data video analytics system must be able to support very large datasets created now and in the future. All the components in big video data systems must be capable of scaling to address the ever-growing size of complex datasets.

We propose a paradigm presented in Fig. 2 that encompasses all the steps of data analytics, from data access and filtering to data mining and sharing product knowledge, which consist of complex graphs of many concurrent tasks, offering a flexible programming model, distributed task interoperability, and execution scalability that reduces data analytic completion time to address the complexity of scientific and business applications.

6. Prototype and Evaluation

6.1. Prototype implementation

For demonstration purpose, a prototype of the architecture described in Section 4 has been implemented. A sequence communication message diagram of the prototype is described in Fig. 6

IoT devices. Big Data of videos have also arrived in the form of large and a growing number of public cameras that is, cameras directed toward public spaces or publicly open events (e.g., APEC). Within the city of

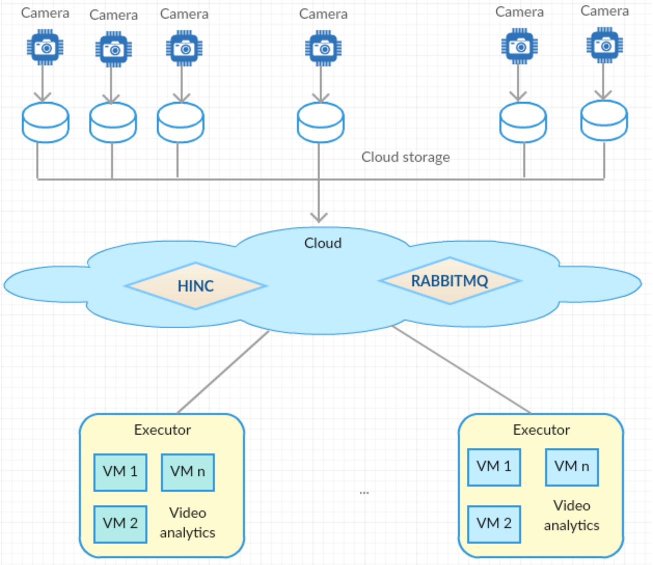


Figure 2. Video analytics paradigm

Da Nang alone, there are an estimated more than 100 surveillance cameras (see http://camera.0511.vn/). To prepare Asia-Pacific Economic Cooperation, Da Nang city set up 300 surveillance cameras.

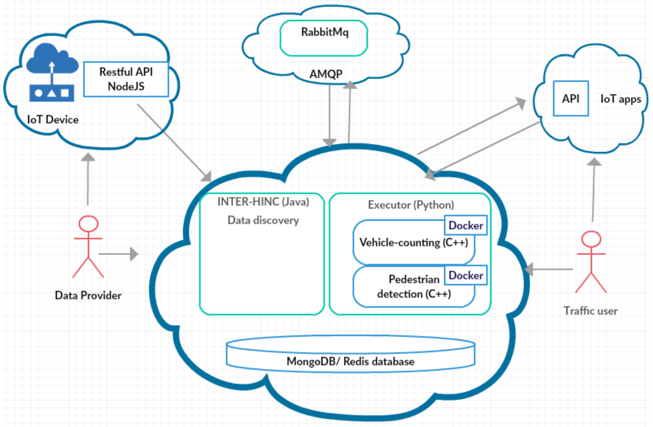
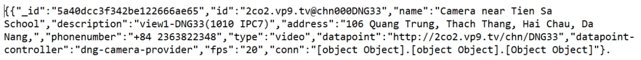


Figure 3. Haivan prototype

Data provider. A NodeJS server is developed to serve like a data adapter for camera devices query. The result of querying a camera datapoint is showin in Table 1. The functionality of the data provider is as follows:



```
{{"_id":"5a40dcc3f342be122666ae65","id":"2co2.vp9.tv@chn000DNG33","name":"Camera near Tien Sa
School","description":"view1-DNG33(1010 IPC7)","address":"106 Quang Trung, Thach Thang, Hai Chau, Da
Nang,","phonenumber":"+84 2363822348","type":"video","datapoint":"http://2co2.vp9.tv/chn/DNG33","datapoint-
controller":"dng-camera-provider","fps":"20","conn":"[object Object].[object Object].[Object Object]"}.
```

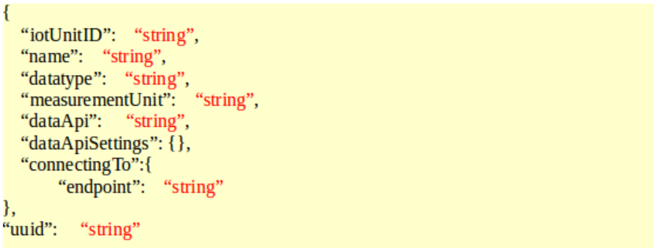
Figure 4. Provider Return Camera List

My host expects video name in the body, should include the file type extension this call will return a google cloud storage download link that expires after 3 days.

Table 1. public-camera-metadata

Field	Type	Description
_Id	string	primary key
Id	string	Use to get video frame of data point
Name	string	Camera name
Description	string	Describe about Camera
Address	string	Address which to put the camera
Phonenumber	string	Hotline
Type	string	Data type (video)
Datapoint	string	Address input data (usually file .ts)
Datapoint-controller	string	Camera provider

Data discovery. rsiHub uses to search camera source of response class are and then convert it Response Class. Example value iotUnitID, name, datatype, measurementUnit etc. HINC Global API The list of the data points available from a single or multiple sites, look at address http://35.185.26.121:9000/global-management-service-1.0. When querying a data point, the data structure will return as in the Fig. 5.



```
{
  "iotUnitID":   "string",
  "name":   "string",
  "datatype":   "string",
  "measurementUnit":   "string",
  "dataApi":   "string",
  "dataApiSettings": {},
  "connectingTo":{
      "endpoint":   "string"
},
"uuid":   "string"
```

Figure 5. rsiHub Global API the list of data points

The video streams are first fetched from cloud storage or frame URI and are decoded to extract individual video frames. Each frame is then processed individually for object detection and recognition. Each frame can be processed independently of each other. This approach enables the processing of individual frames on cloud resources, leading to high throughput and scalability.

Advance Message Queue Protocol (AMQP). When client gets video frame data point, it then will be published url/data to broker AMQP. This paper uses RabbitMQ cloud for AMQP to enable the development and industry wide use of standardised messaging middleware technology that will lower the cost of enterprise and systems integration and provide industrial grade integration services to a broad audience. There are three main types of component, which are connected into processing chains: (1) The "exchange" receives messages from publisher applications and routes these to "message queues", based on arbitrary criteria, usually message properties or content. The "message queue" stores messages until they can be safely processed by a consuming client application (or multiple applications).

The "binding" defines the relationship between a message queue and an exchange and provides the message routing criteria.

Analytics application. Our Analytics executor executes number of instances of two applications: Vehicle-counting and Pedestrian detection.

Output data storage. Output data from the analytics application is stored in MongoDB.

6.2. Evaluation

Evaluation criteria. This system is evaluated by ability extended and integrated. The scalability allow programmer using multiple programming language for a system. Beside, we easy integrate other module into system. To prove, then the system is built such as configuration server, client and registry cloud account. We have about 5 minutes to integrate the first module is vehicle counting into the system, and then, the same time to integrate the second module is pedestrian detection into this system.

Video-analytics cases. Two applications that developed for testing are as follows:

- Vehicle counting - The application is carried out in order to reduce the traffic congestion by calculating the traffic density in a particular direction of the road. The system starts with an video url or video frame. Then one frame per second continuously extracts from the live video and processed each frame by converting it into grayscale. the targeted area is selected, the area where the vehicles are present and filtered out unnecessary surrounding information. Next phase, determines the presence of objects in live video by taking the absolute difference of each extracted frame with the reference image. Then the presence of objects is enhanced by binarization of the difference image. Then the final step is to calculate the traffic density in the desired target area by counting the number of vehicles in that region. To perform this, first, the vehicles are marked in the targeted region by scanning all the connected objects, and filtering out smaller and overlapping objects.

- Pedestrian detection - The application presents pedestrian counting, capturing of pedestrian running red lights, and another behavior analysis. Pedestrian detection is to detect pedestrians in each frame of a video and to sequentially store them into a container. Our bodies all have the same basic structure regardless of gender, race, or ethnicity.

  In general, a person's body parts include the head, arms, body and legs. We can exploit this semi-rigid structure and extract features to quantify the human body. Machine learning can be used to train these features to detect and track humans in images and video streams.

  Data CENTRIST [26] to detect humans because it succinctly encodes the crucial sign information, and does not require preprocessing or post-processing.This pedestrian detect module use 108-by-36 as the detection window size, and divide the image patch into 9 × 4 blocks (108 pixels for each block). Next [27], we treat super-block (any adjacent 2×2 blocks) and extract a CENTRIST descriptor from each super-block. There are 8 × 3 = 24 super-blocks, thus the feature vector has 256 × 24 = 6144 dimensions for a candidate image patch. A one-pixel-wide border of each super-block is not included when computing the CENTRIST descriptor because the Census Transform requires a 3 × 3 region.

6.3. Experiments and Discussion

Experiments. This section provides the details of our experimental setup used to evaluate the proposed system. The parameters used to evaluate the performance of the system are scalability, simplification of integration and processing time of each video frame.

The configuration of the client is as follows: Intel Core i5 CPU M 520 @ 2.40GHz x 4, memory 8GB, SSD Hard Drive have Capacity 120GB. The software is operating system Ubuntu 32 bit. For Server, there are three cloud servers be used operating system Ubuntu, speed process 1GB, capacity 897GB. In addition, there five Amazon Web Services EC2 (AWS EC2) to run vehicle counting analytics and five 5 Amazon Web Services EC2 for pedestrian detection analytics.

The total video data used for the experimentation consists of one year of video streams from 100 cameras in metro of Danang city. Each video stream has a duration of 120 seconds. The video streams are encoded with H.264 format. The frame rate for each video stream is 18fps. The data rate and bit rate for each video stream are 6,156 kbps respectively. The decoding of each video stream generates a frame set of 3000 video frames. Each video frame holds a data size of 342,7kb.

Discussion. The result of the experiments are show in Fig. 11 an Fig.12. We using some client to run Vehicle module and send message to RabbitMQ every 10 seconds loop. Though we have been quite successfully with the current HAIVAN-CVA prototype system, evidently, we only stand at the beginning of solving the problem with the degree of reliability necessary to actually deploy such a system. The above experiment

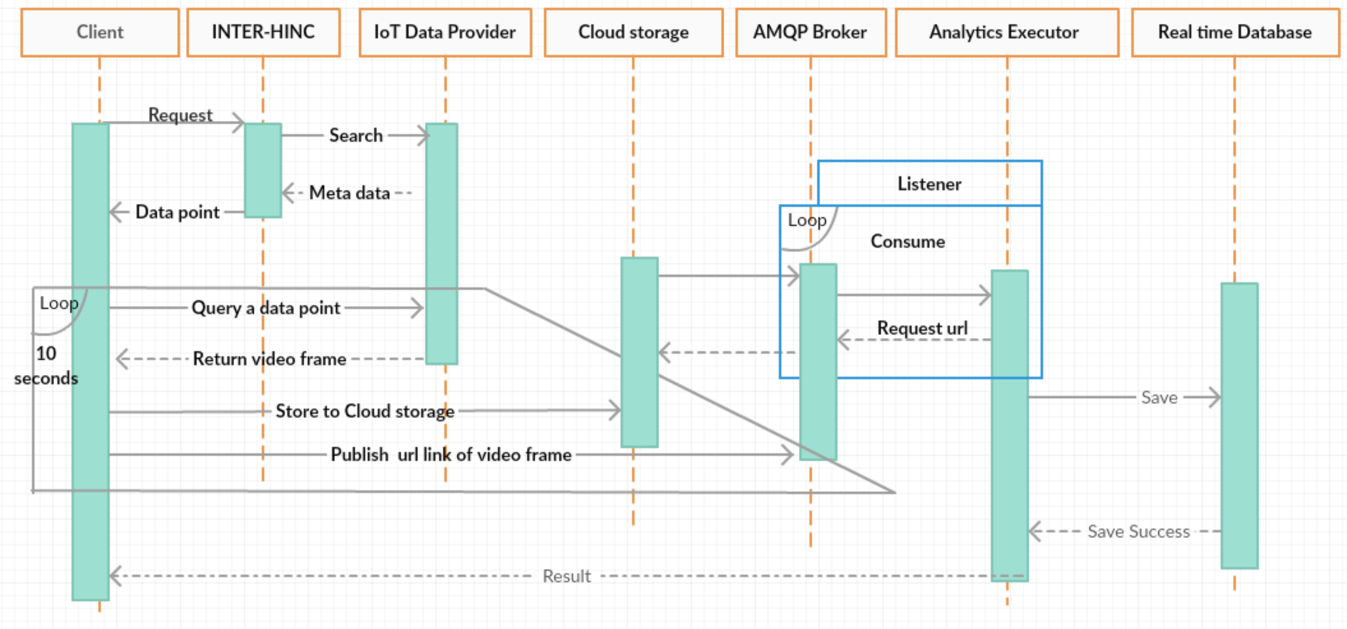


Figure 6. Haivan Sequence Project

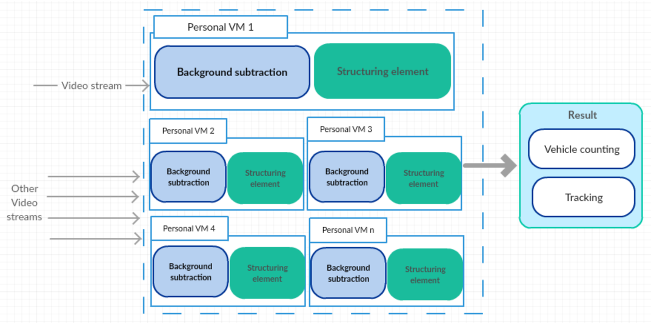


Figure 7. Vehicle Counting process

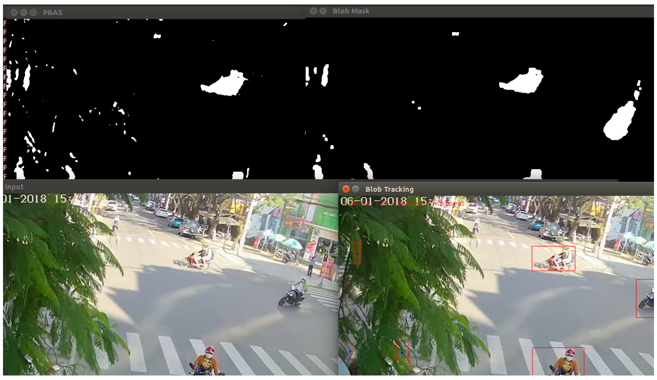


Figure 8. Vehicle Counting

of the real model, the real-time capability and the data heterogeneity are supported by our platform although the real-time capability of video streaming data is more than expected. Our platform has also the capacity to measure the value of data in real-time. we focus on

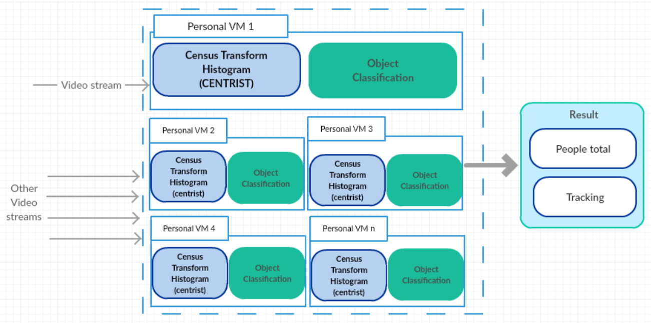


Figure 9. Pedestrian detection process

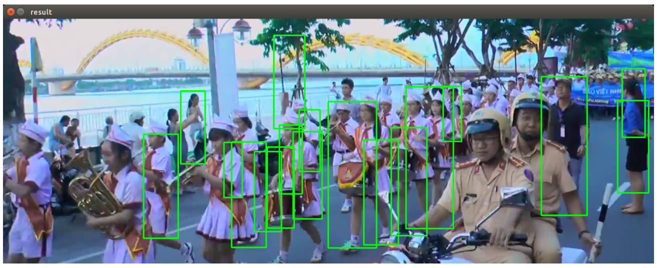


Figure 10. Pedestrian detection

quality of data analysis in this experiment, by scalable integrating other applications.

7. Conclusions

This paper presents the design and implementation of HAIVAN-CVA, a scalable IoT video data analytics for smart cities. A prototype has been implemented and experiment have been executed to evaluate the performance of the proposed HAIVAN-CVA platform. The design of our model is developed to be easily extended for new kinds of IoT devices, message routing and queueing, and data analytics, to permit specific application to be programmed via the paradigm to be flexible yet simple. The future will focus on developing more useful analytics application plug-in for the system.

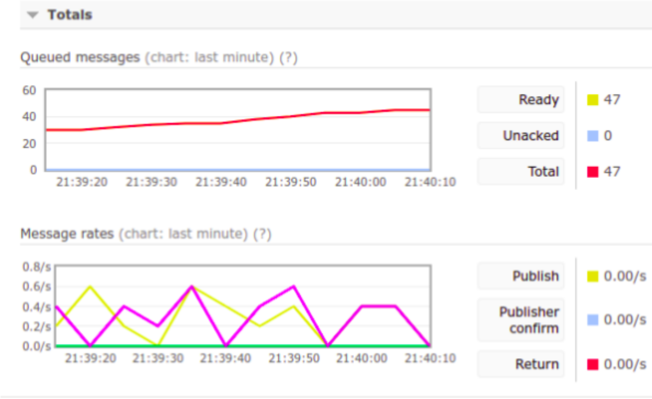
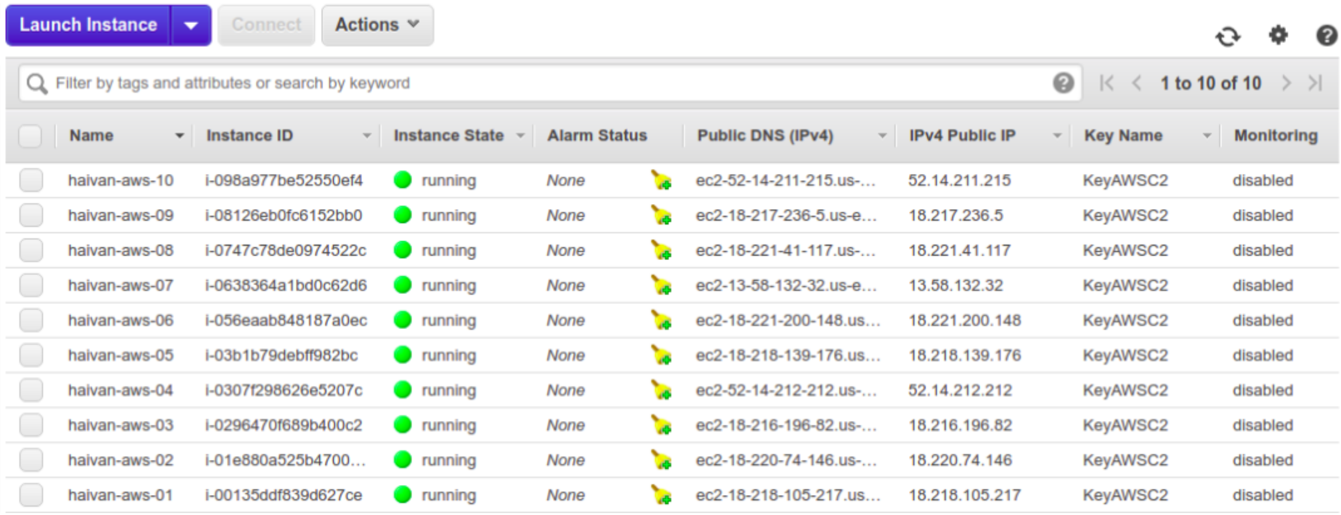


Figure 11. Message queue



	Name	Instance ID	Instance State	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	Key Name	Monitoring
□	haivan-aws-10	i-098a977be52550ef4	● running	None	ec2-52-14-211-215.us-...	52.14.211.215	KeyAWSC2	disabled
□	haivan-aws-09	i-08126eb0fc6152bb0	● running	None	ec2-18-217-236-5.us-e...	18.217.236.5	KeyAWSC2	disabled
□	haivan-aws-08	i-0747c78de0974522c	● running	None	ec2-18-221-41-117.us-...	18.221.41.117	KeyAWSC2	disabled
□	haivan-aws-07	i-0638364a1bd0c62d6	● running	None	ec2-13-58-132-32.us-e...	13.58.132.32	KeyAWSC2	disabled
□	haivan-aws-06	i-056eaab848187a0ec	● running	None	ec2-18-221-200-148.us...	18.221.200.148	KeyAWSC2	disabled
□	haivan-aws-05	i-03b1b79debff982bc	● running	None	ec2-18-218-139-176.us...	18.218.139.176	KeyAWSC2	disabled
□	haivan-aws-04	i-0307f298626e5207c	● running	None	ec2-52-14-212-212.us-...	52.14.212.212	KeyAWSC2	disabled
□	haivan-aws-03	i-0296470f689b400c2	● running	None	ec2-18-216-196-82.us-...	18.216.196.82	KeyAWSC2	disabled
□	haivan-aws-02	i-01e880a525b4700...	● running	None	ec2-18-220-74-146.us-...	18.220.74.146	KeyAWSC2	disabled
□	haivan-aws-01	i-00135ddf839d627ce	● running	None	ec2-18-218-105-217.us...	18.218.105.217	KeyAWSC2	disabled

Figure 12. AWSC2 Instances

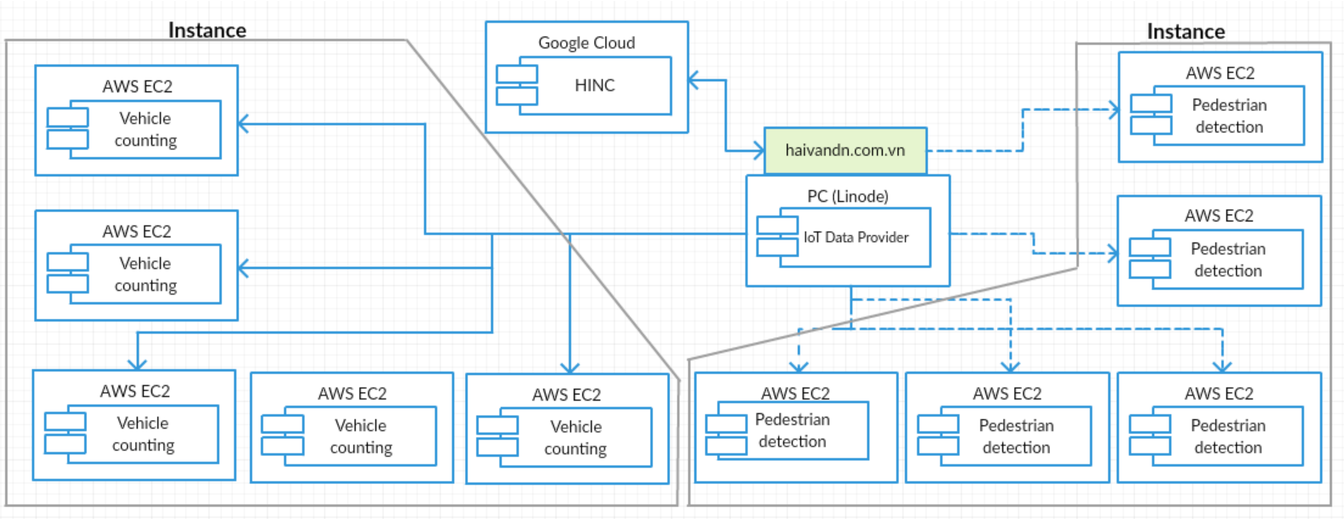


Figure 13. General resource settings for experiments -Draft

References

[1] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, “A scalable and privacy-aware iot service for live video analytics,” in *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 2017, pp. 38–49.

[2] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving video database with scalable annotation tooling,” in *arXiv*, 2018.

[3] B. Afzal, M. Umair, G. A. Shah, and E. Ahmed, “Enabling iot platforms for social iot applications: Vision, feature mapping, and challenges,” *Future Generation Computer Systems*, 2017.

[4] P. Charalampidis, E. Tragos, and A. Fragkiadakis, “A fog-enabled iot platform for efficient management and data collection,” in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2017 IEEE 22nd International Workshop on*. IEEE, 2017, pp. 1–6.

[5] S. Jain, V. Nguyen, M. Gruteser, and P. Bahl, “Panoptes: servicing multiple applications simultaneously using steerable cameras.” in *IPSN*, 2017, pp. 119–130.

[6] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, “Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes,” in *Elsevier*, vol. 172. Elsevier, 2018, pp. 88–97.

[7] A. Jain, S. Basantwani, O. Kazi, and Y. Bang, “Smart surveillance monitoring system,” in *Data Management, Analytics and Innovation (ICDMAI), 2017 International Conference on*. IEEE, 2017, pp. 269–273.

[8] J. E. Gómez, F. R. Marcillo, F. L. Triana, V. T. Gallo, B. W. Oviedo, and V. L. Hernández, “Iot for environmental variables in urban areas,” *Procedia Computer Science*, vol. 109, pp. 67–74, 2017.

[9] B. Bhomble, S. Katkar, D. Dhere, G. Arage, S. Bagwan, and M. Jadhav, “Iot based smart sniper,” in *I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on*. IEEE, 2017, pp. 557–561.

[10] P. P. Ray, “A survey of iot cloud platforms,” *Future Computing and Informatics Journal*, 2017.

[11] F. Terroso-Saenz, A. González-Vidal, A. P. Ramallo-González, and A. F. Skarmeta, “An open iot platform for the management and analysis of energy data,” *Future Generation Computer Systems*, 2017.

[12] T.-h. Kim, C. Ramos, and S. Mohammed, “Smart city and iot,” 2017.

[13] J. Yang, Y. Han, Y. Wang, B. Jiang, Z. Lv, and H. Song, “Optimization of real-time traffic network assignment based on iot data using dbn and clustering model in smart city,” *Future Generation Computer Systems*, 2017.

[14] M. Humayun, N. Jhanjhi, M. Z. Alamri, and A. Khan, “Smart cities and digital governance,” in *Employing Recent Technologies for Improved Digital Governance*. IGI Global, 2020, pp. 87–106.

[15] Z. A. Almusaylim, N. Zaman, and L. T. Jung, “Proposing a data privacy aware protocol for roadside accident video reporting service using 5g in vehicular cloud networks environment,” in *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, 2018, pp. 1–5.

[16] H. Rabbouch, F. Saâdaoui, and R. Mraihi, “Unsupervised video summarization using cluster analysis for automatic vehicles counting and recognizing,” *Neurocomputing*, 2017.

[17] A. Sanchez, P. D. Suarez, A. Conci, and E. Nunes, “Video-based distance traffic analysis: application to vehicle tracking and counting,” *Computing in Science & Engineering*, vol. 13, no. 3, pp. 38–45, 2011.

[18] Y. Xia, X. Shi, G. Song, Q. Geng, and Y. Liu, “Towards improving quality of video-based vehicle counting method for traffic flow estimation,” *Signal Processing*, vol. 120, pp. 672–681, 2016.

[19] M. Anandhalli and V. P. Baligar, “A novel approach in real-time vehicle detection and tracking using raspberry pi,” *Alexandria Engineering Journal*, 2017.

[20] P. Barcellos, C. Bouvié, F. L. Escouto, and J. Scharcanski, “A novel video based system for detecting and counting vehicles at user-defined virtual loops,” *Expert Systems with Applications*, vol. 42, no. 4, pp. 1845–1856, 2015.

[21] Y. P. Kocak and S. Sevgen, “Detecting and counting people using real-time directional algorithms implemented by compute unified device architecture,” *Neurocomputing*, vol. 248, pp. 105–111, 2017.

[22] S. Yoshinaga, A. Shimada, and R.-i. Taniguchi, “Real-time people counting using blob descriptor,” *Procedia-Social and Behavioral Sciences*, vol. 2, no. 1, pp. 143–152, 2010.

[23] L. Tay, A. T. Jebb, and S. E. Woo, “Video capture of human behaviors: toward a big data approach,” *Current Opinion in Behavioral Sciences*, vol. 18, pp. 17–22, 2017.

[24] K. V. Hung *et al.*, “Education influence in traffic safety: A case study in vietnam,” *IATSS research*, vol. 34, no. 2, pp. 87–93, 2011.

[25] D. Le, N. C. Narendra, and H. L. Truong, “HINC - harmonizing diverse resource information across iot, network functions, and clouds,” in *4th IEEE International Conference on Future Internet of Things and Cloud, FiCloud 2016, Vienna, Austria, August 22-24, 2016*, M. Younas, I. Awan, and W. Seah, Eds. IEEE Computer Society, 2016, pp. 317–324. [Online]. Available: https://doi.org/10.1109/FiCloud.2016.52

[26] J. Wu and J. M. Rehg, “Centrist: A visual descriptor for scene categorization,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.

[27] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.