Edge Coordinated Query Configuration for Low-Latency and Accurate Video Analytics

Peng Yang, *Member, IEEE*, Feng Lyu, *Member, IEEE*, Wen Wu, *Member, IEEE*,
Ning Zhang, *Senior Member, IEEE*, Li Yu, *Member, IEEE*,
and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—To develop smart city and intelligent manufacturing, video cameras are being increasingly deployed. In order to achieve fast and accurate response to live video queries (e.g., license plate recording and object tracking), the real-time high-volume video streams should be delivered and analyzed efficiently. In this article, we introduce an end-edge-cloud coordination framework for low-latency and accurate live video analytics. Considering the locality of video queries, edge platform is designated as the system coordinator. It accepts live video queries and configures the related end cameras to generate video frames that meet quality requirements. By taking into account the latency constraint, edge computing resources are subtly distributed to process the live video frames from different sources such that the analytic accuracy of the accepted video queries can be maximized. Since the amount of required edge computing resource and video quality to accurately address different video queries are unknown in advance, we propose an online video quality and computing resource configuration algorithm to gradually learn the optimal configuration strategy. Extensive simulation results show that as compared to other benchmarks, the proposed configuration algorithm can effectively improve the analytic accuracy, while providing low-latency response.**

***Index Terms*—Edge computing, gradient method, neural networks, video analytics.**

I. Introduction

With the evolution of communication infrastructure and embedded systems, an increasing number of video cameras have been deployed to obtain rich environment information for various purposes, including traffic monitoring, object tracking, e-healthcare, and intelligent industrial robotics [1]–[3]. To serve those purposes, the captured video frames need to be delivered to a data center with abundant computing resources, where the contents are then processed to obtain the required scene information. Yet, the bandwidth and computing resources required to achieve fast delivery and accurate analysis of video streams are prohibitive. For example, when a camera with 1280×1024 pixels records at 20 frames per second, it generates 50 GB of data per day [4], which poses great pressure on both video transcoding and data delivery. Video analytics is also computation intensive. For instance, some of the deep neural network (DNN) based algorithms require a 30-GigaFlop processor for accurate response to an object recognition query [5]. Considering the amount of deployed cameras and the significant data volume of captured frames, video analytics at scale is bandwidth consuming and computation intensive [6]. Consequently, cost-efficient solution for low-latency video delivery and high-accuracy analytics is of paramount importance.

Edge computing holds great potential in enabling video delivery and analytics at scale. It provides computing resources in the proximity of mobile devices, the advantages of which have been manifested in supporting Internet-of-Things [7] and industrial applications [8], video streaming [9]–[11], and caching [12], [13], mobile augmented reality [14], and facial recognition [15]. Consider the locality of video queries, the corresponding frames can be processed on the edge platform, instead of being streamed to the cloud center. The benefits are two fold. First, real-time video queries (e.g., vehicle license plate reader and object tracking) are highly related to a specific location [16]. Addressing those queries on the local edge, which is only one hop away, helps to promote context awareness and reduce service latency, compared to delivering all the contents to the cloud center. Second, if more video queries can be addressed locally, significant bandwidth resources from edge to the cloud can be saved. Unfortunately, the computing capacity at the edge node (EN) varies significantly from several cores to hundreds of cores. Oftentimes, the edge resources are insufficient to address all the queries [17], [18].

To realize large-scale video analytics, the cloud and edge, as well as their interoperation are crucial building blocks in achieving low-latency and accurate video analytics [16]. This is also validated by the industrial service providers. It is argued that

Manuscript received June 30, 2019; revised September 27, 2019; accepted October 20, 2019. Date of publication October 24, 2019; date of current version March 17, 2020. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the National Natural Science Foundation of China under Grant 61871437, and in part by the Natural Science Foundation of Hubei Province of China under Grant 2019CFA022. Paper no. TII-19-2921. *(Corresponding author: Feng Lyu.)*

P. Yang, F. Lyu, W. Wu, and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: p38yang@uwaterloo.ca; f2lyu@uwaterloo.ca; w77wu@uwaterloo.ca; sshen@uwaterloo.ca).

N. Zhang is with the Department of Computing Sciences, Texas A & M University-Corpus Christi, Corpus Christi, TX 78412 USA (e-mail: ning.zhang@tamucc.edu).

L. Yu is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: hustlyu@hust.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2949347

hierarchical architecture consisting of end cameras, edge platform, and cloud server is the only feasible solution to large-scale video analytics [19]. Based on a combination of private clusters and public cloud, Hung *et al.* revealed the tradeoff between resource demands and the accuracy of object detection queries, as well as its dependency on the choice of DNN detectors [6]. However, there is no well-established model to characterize the relationship between resource demands and accuracy, as it depends additionally on the content of video source (e.g., background, weather, object size). As a result, offline accuracy-resource profiling is the common solution to online video query planning, based on which configurations of video quality, computing resource allocation, and DNN implementations are determined. Instead of making decisions based on an offline profiler, it is of great interest to explore low-latency video analytics at the edge by harnessing the synergies of end camera and the cloud.

In this article, we propose an end-edge-cloud coordination framework for high analytic accuracy, while satisfying the latency requirement of video queries. Generally, higher quality of video frames contribute to better analytic results. In specific, the video analytic accuracy depends the content type, the resolution, and frame rate, the available computing capacity, etc. Yet, delivering and processing high-quality video frames also incur longer delay and require more computing resources [5]. Furthermore, the core video analyzing components have a variety of implementations. Different implementations have distinct resource demands that lead to varying analytic accuracy, but not any one of them is the least resource demanding and accurate across all scenarios [6]. As a result, it is difficult to build a unified video analytic accuracy model to characterize the impact of various configuration parameters.

Given that there is no established video analytic accuracy function, we make the following simplifications: 1) each query can be addressed based on the video frames from one single camera; 2) the implementation of core video analyzing components is given. Then, the video content corresponds to each query is fixed. The problem of improving the analytic accuracy narrows down to configuring only the source video quality and edge computing resources. We formulate an accuracy maximization problem that accounts for both video quality control and edge resource allocation. The challenge of this problem is obtaining the gradient of the unknown analytic function for adaptive configuration. To address this challenge, we define a smoothed version of the analytic function, which enables us to obtain the gradient information directly from the function value. An adaptive configuration algorithm is thus proposed, by which the direction of accuracy-ascending configuration is acquired. The performance of the algorithm is theoretically analyzed, which turns out to be asymptotically optimal. The contributions of this article are summarized as follows.

1) We design a camera-edge-cloud collaborative framework for accurate video analytics at scale. Specifically, the edge platform accepts video queries and coordinates with the end cameras to obtain quality video frames, so that accurate response to the video query can be produced.
2) We formulate the dynamic configuration problem as an optimization problem. To address the uncertainty of the analytic function with regard to video quality and computing resources, we develop an online algorithm to estimate the function gradient and tune the configuration of video quality and computing resource.
3) We provide theoretical analysis of the achieved analytic accuracy of the proposed algorithm. Extensive simulation results are also carried out to demonstrate the advantages of the proposed algorithm against other benchmarks.

The remainder of this article is organized as follows. Section II reviews related works. Section III describes the collaborative video analytic framework and problem formulation. The proposed online coordination algorithm and theoretical performance analysis are presented in Section IV, followed by performance evaluation and discussions in Section V. Finally, Section VI concludes this article.

II. Related Works

There are significant works on edge-assisted video streaming and cost-efficient video analytics. Since users may have different preferences to the experienced video quality metrics (e.g., frame rate, resolution, and rebuffering occurrence), Yang *et al.* developed a learning algorithm to adjust the allocated edge resources to mobile users based on their feedback, which is proved to be asymptotically optimal in maximizing user quality of experience (QoE) [9]. In addition to user QoE, Mehrabi *et al.* also considered user fairness and proposed to periodically perform user-to-edge mapping and per-user bitrate selection, based on the domain knowledge from both radio access and application level at the edge coordinator [10]. Wang *et al.* devised a video transcoding framework, where transcoding servers are deployed close to base stations [11]. In this way, video contents can be transcoded at a finer granularity in accordance with the channel condition. Consequently, both the bandwidth utilization and the user QoE can be improved. For video analytics, however, the quality of frames should be tuned to facilitate the rapid acquisition of accurate analytical results, instead of better user QoE.

To obtain accurate results at low latency, video analytics generally requires high-performance computing facilities. Therefore, improving the computing resource utilization becomes the main focus of most related works. For general video queries, Hsieh *et al.* designed and implemented a system named Focus, which deconstructs video analytics into two phases, i.e., video ingest and video query [20]. During the ingest phase, each video frame is processed by a cheap convolutional neural network (CNN) classifier to obtain coarse object indexes. After receiving the video queries, those indexes are used to provide fast responses. If the accuracy requirement cannot be satisfied, additional expensive CNN classifiers will be invoked. By tuning the share of computing resources of both phases, Focus achieves effective and flexible tradeoff of latency and accuracy of video analytics. For live video analytics, Zhang *et al.* proposed VideoStorm, which consists of an offline video profiler and an online scheduler [5]. Based on the profiled query resource-quality relationship, online configuration is optimized to strike the balance between analytic accuracy, processing latency, and resource consumption. Another line of research

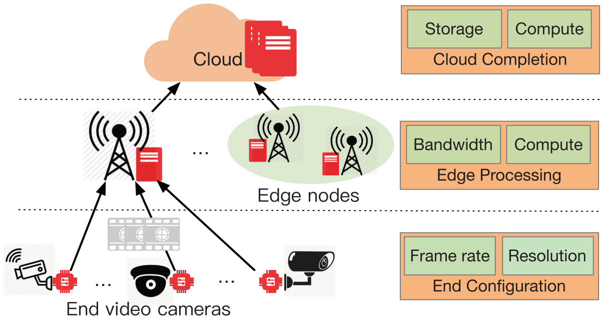


Fig. 1. End-edge-cloud collaborative video analytic framework.

focuses on optimizing the configuration of neural networks, including Chameleon [21] and MCDNN [22], to reduce resource consumption, while maintaining accuracy.

The closest work to ours is the VideoEdge system [6], which incorporates user-specified cost budget as an optimization constraint. It also deals with dynamic bandwidth via continuous probing. Our article differs from these works in the following major ways. First, instead of concentrating on video analytics on a single processing unit, we propose an end-edge-cloud orchestration framework, in which EN plays a vital control role on the video continuum from end cameras to the cloud. Second, we consider practical live video analytics, where queries arrive asynchronously and cannot be merged. Thus the camera coordination and resource allocation need to be performed on a per-query basis. At last, we adopt gradient learning techniques to configure video frames at their optimal quality level to combat system bandwidth dynamics.

III. System Model and Problem Formulation

In this section, we present the camera-edge-cloud collaborative video analytics framework. As shown in Fig. 1, the collaborative analytic framework consists of three tiers: The end cameras, the EN, and the cloud server. With the increasing penetration of the fifth generation cellular systems, video cameras can be readily connected wirelessly to base stations and WiFi access points, both of which are ideal bearers of edge deployment [7], [9]. Compared with the cloud server, ENs are limited in storage and computing capacity. Only a subset of video queries can be accepted and processed at the edge. Particularly, EN makes prior decisions on query acceptance, based on the latency requirement, the estimated computing demand, and the edge capacity. For accepted video queries, EN configures the video quality generated by the corresponding source camera, and optimizes the resource allocation to all the accepted queries. Other video queries are sent to the cloud server at additional transmission cost and processing delay. Such coordination enables flexible and scalable processing of video queries, which helps to readily extend the framework to other online monitoring applications, such as traffic monitoring and autonomous driving. In what follows, we introduce the model of video query, source camera configuration, edge resource allocation, and the problem formulation. Important notations are summarized in Table I.

TABLE I
Summary of Notations

Notation	Description
$\mathcal{T}$	Set of time slots
$\mathcal{Q}$, $\mathcal{Q}_t$	Set of video queries, set of accepted queries in time slot t
τ_q	Latency requirement of query q
r_q	Uplink data rate of the camera corresponds to query q
F, F'	Upper and lower bound of the video quality (in data volume)
C	Computing capacity of the edge node
f_q, $f_{q,t}$	Quality of video frames for query q, in time slot t
c_q, $c_{q,t}$	Computing resource allocated to query q, in time slot t
$\Lambda(f_q, c_q)$	Mapping function of video quality and computing resource to the analytic accuracy of query q
ϵ_q	A query specific scaling constant
$\mathcal{B}(f_q, c_q)$	Bernoulli parameter of query q given by $\Lambda(f_q, c_q)$ and ϵ_q
X_q	Bernoulli random variable with parameter $\mathcal{B}(f_q, c_q)$
ξ	Required processing per unit data volume (in flop/bit)
$\mathbf{x}$	Vector form of QC pair $[f, c]$
$\mathbb{F}_q$	Feasible region of QC configuration of query q
$\mathcal{R}_q(T)$	Cumulative regret of analytic accuracy of query q

A. Video Query

Consider a slotted system in which time is divided into a sequence of slots $\mathcal{T} = \{1, 2, \ldots, T\}$ with equal duration. There is a set $\mathcal{Q}$ of video queries to be answered, which are assumed to be associated with only one camera, including facial recognition, vehicle license plate reader, traffic monitoring, and industry process inspection. Video queries that require information from multiple cameras, including target tracking and object localization, deserve further investigation and are not the focus of this article. The pattern of query arrival is assumed to be stable over the time slots. At the beginning of each slot $t \in \mathcal{T}$, only a subset $\mathcal{Q}_t \in \mathcal{Q}$ of queries are accepted by the EN due to the limited edge processing capacity. Unaccepted queries will be directed to the cloud or wait until the next time slot.

The accepted video queries are first parsed by the EN, by which the particular camera that covers the interested area of the query will be identified.[1] Afterward, EN makes two decisions for each query $q \in \mathcal{Q}_t$: the requested quality of the captured video frames and the amount of edge computing resources. Once the corresponding video frames arrive at the EN, video analytics corresponds to each accepted query is performed by the allocated computing resources. The generated analytical result will be fed back to the source of video query. Each query $q \in \mathcal{Q}$ has a latency requirement τ_q, which constrains both the video streaming delay and edge processing delay. Without loss of generality, we assume, $\forall q \in \mathcal{Q}$, τ_q is less than the time slot duration, and the processing of all the accepted queries is finished within the corresponding slot.

B. Source Video Configuration

High-performance video cameras that are capable of ultrawide angle capturing, video processing, and wireless communications, are key components of surveillance systems. We assume that those cameras are connected to the EN via a wireless interface. After accepting video queries in each time slot, the EN

[1] Without loss of generality, we assume the computing resource required for query parsing is negligible, compared to that required for video processing.

sends a quality control information to all the relevant cameras. As a response, the corresponding video frames will be generated and streamed to the EN. Note that the scope of each camera is fixed; hence, the relevant location and background information can be made available to the EN, so that queries can be indexed to cameras.

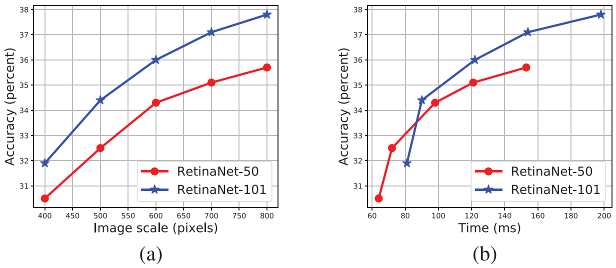
Generally, high video quality contributes to better analytic results, but it also demands more resources for encoding, transmission, decoding, and analyzing. For a received video clip, denote by f the quality (in data volume) of the captured video frames. Due to the limitation of both camera lens and supported frame rate, it is reasonable to assume that the video quality is upper bounded by $f \leq F$. Meanwhile, to make effective attempt on addressing the video queries, a minimum amount of information should be provided, i.e., $f \geq F'$. Note that the input for video analytics is the decoded frames, instead of the coded frames from the source camera. For simplicity, we assume that for any two accepted queries $q_1,\ q_2 \in \mathcal{Q}_t$, the corresponding cameras are different. In case queries are concerning the same area covered by a certain camera, the analytic results derived from the same video stream can be used to answer those queries simultaneously.

C. Edge Resource Model

1) Bandwidth Dynamics: Video cameras are connected to the EN via shared bandwidth resources with other mobile devices. Due to the mobility of mobile users and the uncertainty of traffic demands, the available resource for delivering video frames from each camera is time varying and unknown in advance. Denote by r_q the uplink data rate of the camera corresponding to query q. As our objective is improving the analytics accuracy, the uplink data rate only affects the video streaming delay. Hence, we assume that the camera data rate corresponding to different types of queries remain constant in each time slot, which can be predetermined by the EN solely based on the total edge spectrum resource.

2) Computing Resource Allocation: For a specific video query, allocating more computing resource increases the probability that accurate analytic results can be obtained by the EN. Yet, the video quality should be matched with the allocated computing resource, otherwise, allocating excessive resource to a quality limited video clip is wasteful. As a result, our objective is to jointly determine the requested video quality level and the corresponding computing resource, so that the amount of accurate analytic results can be maximized by the limited edge computing budget. Let $c_{q,t}$ be a positive value indicating the allocated computing resource (in floating point operations per second, Flop/s) for query q in the tth time slot, then we have

$$\sum_{q \in \mathcal{Q}_t} c_{q,t} \leq C \quad \forall t \in \mathcal{T} \tag{1}$$



(a) (b)

Fig. 2. Impact of quality and computing capacity on the accuracy of objection detection [24]. (a) Quality (pixels of the shorter side) versus accuracy. (b) Execution time (computing capacity) versus accuracy.

where C represents the computing capacity at the EN. Let $X_q \in \{0, 1\}$ be a binary variable that equals 1 if query q is accurately answered, and equals 0 otherwise. X_q depends on both the corresponding video quality f_q and the allocated computing resource c_q. The larger the value of f_q and c_q, the better chance that query q is accurately addressed. Formally, X_q can be considered as a sample from a Bernoulli distribution [23] with parameter

$$\mathcal{B}\left(f_q, c_q\right) = \min\left\{1, \Lambda\left(f_q, c_q\right)/\epsilon_q\right\} \tag{2}$$

where $\Lambda(f_q, c_q)$ is a function that maps the video quality and the computing resource to the accuracy of the analytic results of query q, and $\epsilon_q \in (0, \infty)$ is a query specific scaling constant. From (2), the probability of $X_q = 1$ grows linearly with $\Lambda(f_q, c_q)$ until it is equal to 1, after which allocating additional computing resource or requesting higher quality of video frames is worthless. $\epsilon_q \in (0, \infty)$ can be regarded as a measure of difficulty of addressing video query q, and $\Lambda(f_q, c_q)$ represents the analytic capability enabled by the configured quality-computing (QC) pair (f_q, c_q). Ideally, in order to obtain more accurate results in a cost-effective manner, the QC pair should be configured such that the corresponding analytic capability is close to, but less than ϵ_q.

For a given video clip, the accuracy of general video queries, such as object detection, grows with both the computing capacity and video quality. In practice, however, it is difficult to establish a general model of $\Lambda(f_q, c_q)$ that characterize the above-mentioned relation, which additionally depends on the video contents, detectors, learning rate, etc. We plot the breakdown of execution time (analogous to computing capacity), quality (image scale specified by the number of pixels of the shorter side), and the accuracy (average precision) results of two detectors (RetinaNet-50 and RetinaNet-101) in [24], which is shown in Fig. 2. The curves indicate that the accuracy grows sublinearly with both quality and computing capacity. In particular, as shown in Fig. 2(a), increasing the video quality after certain point does not help to increase the analytic accuracy at the same pace. Meanwhile, excessively high video quality demands substantial uplink bandwidth and edge computation resource. In this way, the video quality should be optimally configured, in order to balance the analytic accuracy and resource demand.

Without loss of generality, we only make the following assumptions on $\Lambda(f_q, c_q)$: It is differentiable and concave, i.e., $\partial\Lambda/\partial f_q$ and $\partial\Lambda/\partial c_q$ decrease with f_q and c_q, respectively, and it satisfies the following continuous condition.

Definition 1: There exists $L > 0$ such that for any two QC pairs (f, c) and (f', c'), it holds that

$$|\Lambda(f, c) - \Lambda(f', c')| \leq L\|(f, c) - (f', c')\| \tag{3}$$

where $\|\cdot\|$ is the Euclidean norm.

The rationale behind such continuous assumption is that, similar QC configurations lead to close analytical accuracy.

D. Problem Formulation

By continuously accepting video queries, configuring the source video quality, and allocating computing resources, the objective of the EN is maximizing the amount of queries that are accurately responded in the long term. Formally, we formulate the joint video quality and computing resource allocation problem as follows:

$$\begin{aligned} \max_{\{f_{q,t}, c_{q,t}\}\forall q,t} \quad & \sum_{t\in\mathcal{T}} \sum_{q\in\mathcal{Q}_t} X_{q,t} \\ \text{s.t.} \quad & X_{q,t} \sim \mathcal{B}(f_{q,t}, c_{q,t}) \quad \forall t \in \mathcal{T}, q \in \mathcal{Q}_t, \\ & \frac{f_{q,t}}{r_q} + \frac{\xi f_{q,t}}{c_{q,t}} \leq \tau_q \quad \forall t \in \mathcal{T}, q \in \mathcal{Q}_t, \\ & F' \leq f_{q,t} \leq F, \sum_{q\in\mathcal{Q}_t} c_{q,t} \leq C \quad \forall t \in \mathcal{T}, q \in \mathcal{Q}_t. \end{aligned} \tag{4}$$

In the second inequality, $f_{q,t}/r_q$ and $\xi f_{q,t}/c_{q,t}$ denote the video streaming time and processing time, respectively. ξ is a scaling factor that specifies the required processing capacity of unit data volume in flop/bit. While meeting the constraints on computing capacity, source video quality, and delay requirement, the optimization (4) maximizes the sum accuracy of all the accepted queries over the long term. However, this problem cannot be directly solved due to the following reasons. First, the delay posed on the video quality and computing capacity is nonlinear with decision variables. Second, the objective is a sum of a number of random variables following different distributions across the time span. At last, for a certain type of query, little information on the function $\Lambda(f_q, c_q)$ is known in advance, except that the the accuracy grows sublinearly with both decision parameters. Meanwhile, after $\Lambda(f_q, c_q)$ reaches ϵ_q, the increase of both video quality f_q and computing capacity c_q will not further increase the accuracy of the analytical result. Such cutoff feature requires repeatedly exploration around the cutoff point, which is costly at exceedingly high values of f_q and c_q.

Since we focus on the long-term accuracy maximization, the replenishment of resources in each time slot allows us to characterize the feature of analytic capability $\Lambda(f_q, c_q)$ of each query in an online manner. In what follows, within the proposed end-edge-cloud collaborative framework, we develop an online algorithm that gradually learns and maximizes the analytic capability in the long run.

IV. Algorithm Design and Performance Analysis

In this section, we attack the original problem (4) using the divided and conquer strategy. Observing that the key challenge of optimization problem (4) is the uncertain analytic capability function $\Lambda(f_q, c_q)$, we first assume that it is known in advance for all the types of queries, based on which an offline algorithm is designed to determine the optimal QC pair. Afterward, we propose an online solution to the original problem based on the offline algorithm.

A. Offline Solution

When $\Lambda(f_q, c_q)$ is known in advance, to improve the probability of obtaining accurate analytic results with minimum resource consumption, it suffices to solve $\Lambda(f_{q,t}, c_{q,t}) = \epsilon_q$ within the feasible region of the pair $(f_{q,t}, c_{q,t})$ for all the $t \in \mathcal{T}$ and $q \in \mathcal{Q}_t$ independently. For notational simplicity, the time slot index is omitted in this subsection.

First, the delay constraint is rewritten as

$$c_q \geq \frac{\xi r_q f_q}{\tau_q r_q - f_q}. \tag{5}$$

To support the delivery of videos that are of the highest quality within the delay constraint, it is necessary to provide sufficient bandwidth to delivery frames that are of different quality levels, i.e., it holds that $\tau_q r_q \geq F \,\forall q$. Otherwise the delay requirement of the video query cannot be guaranteed and it will be rejected. Second, denote by $c_q = \Upsilon_q(f_q)$ the QC curve obtained by solving $\Lambda(f_q, c_q) = \epsilon_q$. By combining $\Upsilon_q(f_q)$ and the equality of (5), we can obtain the unconstrained QC pair (f_q^*, c_q^*). According to the value of f_q^*, without considering the computing constraint (1), we have the following three cases.

1) $f_q^* \leq F'$. In this case, the unconstrained optimal point lies outside the feasible quality region. It can be inferred from (5) that, the minimum computing capacity increases with video quality. To minimize the demand on computing capacity, $(F', \frac{\xi r_q F'}{\tau_q r_q - F'})$ should be chosen as the optimal QC pair. Note that, such choice is over provision, as both quality and computing are higher than that required to achieve the analytic capability threshold, i.e., $\Lambda(F', \frac{\xi r_q F'}{\tau_q r_q - F'}) \geq \epsilon_q$.
2) $F' < f_q^* \leq F$. In this case, as the unconstrained optimal point lies inside the feasible quality region, (f_q^*, c_q^*) is the optimal QC pair, and $\Lambda(f_q^*, c_q^*) = \epsilon_q$.
3) $f_q^* > F$. In this case, as the optimal video quality is not support by the camera, we have to reduce the video quality and increase the allocated computing capacity to compensate. The optimal QC pair should be $(F, \Upsilon_q(F))$, and it holds that $\Lambda(F, \Upsilon_q(F)) = \epsilon_q$.

In the offline scenario, if the computing resource is insufficient, the treatment of queries will favor the less computation-intensive ones, as more queries can be accepted with guaranteed accuracy. Hence, before allocating resources, the queries are sorted according to c_q in ascending order, afterward resources will be allocated sequentially. Since $\Lambda(f_q, c_q) \geq \epsilon_q$ holds for all the accepted queries by the offline algorithm, from (2), the Bernoulli parameter $\mathcal{B}(f_q, c_q)$ equals 1. As a result, $X_q = 1$ and all the accepted queries are accurately answered. The corresponding algorithm is sketched in Algorithm 1. When the algorithm terminates, if $\mathcal{Q} \neq \emptyset$, those queries will be rejected

Algorithm 1: Offline Algorithm for QC Pair Determination.

```
Require: $\xi$, $C$, $F'$, $F$, $\forall q \in \mathcal{Q}$, the $\Lambda(f_q, c_q)$,
    upload data rate $r_q$, delay requirement $\tau_q$, and $\epsilon_q$
Ensure: QC-pair $(f_q, c_q)$, $\forall q \in \mathcal{Q}$
 1: solve $\Lambda(f_q, c_q) = \epsilon_q$ and obtain $(f_q^*, c_q^*)$, $\forall q \in \mathcal{Q}$
 2: set $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 \leftarrow \emptyset$, residual computing resource
    $\bar{C} = C$
 3: for $q \in \mathcal{Q}$ do
 4:   if $f_q^* \le F'$: $(f_q, c_q) = \left(F', \frac{\xi r_q F'}{\tau_q r_q - F'}\right)$,
      $\mathcal{S}_1 \leftarrow \mathcal{S}_1 \cup \{q\}$
 5:   if $F' < f_q^* \le F$: $(f_q, c_q) = (f_q^*, c_q^*)$, $\mathcal{S}_1 \leftarrow \mathcal{S}_2 \cup \{q\}$
 6:   else $(f_q, c_q) = (F, \Upsilon_q(F))$, $\mathcal{S}_1 \leftarrow \mathcal{S}_3 \cup \{q\}$
 7:   end if
 8: end for
 9: while $\bar{C} > 0$ do
10:   find $q' = \arg\min_{q \in \mathcal{Q}} c_q$
11:   if $\bar{C} \ge c_{q'}$: allocate $c_{q'}$ to $q'$
12:   else break
13:   end if
14:   update $\bar{C} \leftarrow \bar{C} - c_{q'}$, and $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{q'\}$
15: end while
```

due to the scarcity of edge computing resource. Such requests are either directed to the cloud or wait until the next time slot. The complexity of the algorithm lies in the sorting of $|\mathcal{Q}|$ queries in the while loop (line 10), which is of polynomial time complexity $O(|\mathcal{Q}| \log |\mathcal{Q}|)$ using typical algorithm, such as quick sort.

B. Online Quality-Computing Configuration Algorithm

In the online scenario, however, the queries will be treated with differentiation. Let $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ be the set of queries of the three cases in the previous subsection, respectively. Queries in $\mathcal{S}_2 \cup \mathcal{S}_3$ have higher priority, as it holds that $\Lambda(f_q, c_q) = \epsilon_q$. In contrast, queries in $\mathcal{S}_1$ are less favored, as the computing resources are over provisioned, and meanwhile they provide little information on the learning of analytic capability function. If computing resources are insufficient, queries in $\mathcal{S}_2 \cup \mathcal{S}_3$ should be prioritized from lightweight ones to computation-intensive ones. Then, remaining computing resources will be allocated to queries in $\mathcal{S}_1$ in the same way. In what follows, we present the details of the online QC configuration algorithm. Notice that the analytic capability function $\Lambda(f_q, c_q)$ is concave, we can find the global maximum if the gradient $\nabla \Lambda$ is known in advance. Unfortunately, neither $\Lambda(f_q, c_q)$ or its first-order derivative is known in practice.

In an online setting, however, it is possible to approximate the gradient by subtly designed QC pairs. Let $\mathbf{x}_q = [f_q, c_q]$ be the QC pair in vector form for notational simplicity, thus $\Lambda(f_q, c_q)$ can be written as $\Lambda(\mathbf{x}_q)$. Define the unit disk $\mathbb{D}$ and the unit circle $\mathbb{C}$ around the origin in two-dimensional space, i.e., $\mathbb{D} = \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\| \le 1\}$ and $\mathbb{C} = \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\| = 1\}$. The disk and circle of radius ζ are $\zeta\mathbb{D}$ and $\zeta\mathbb{C}$, respectively. Define the *smoothed* version of Λ as $\tilde{\Lambda}(\mathbf{x}) = \mathbf{E}_{\mathbf{v} \in \mathbb{D}}[\Lambda(\mathbf{x} + \delta\mathbf{v})]$, then the gradient of $\Lambda(\mathbf{x})$ can be approximated by its smoothed counterpart, i.e., the gradient $\nabla\Lambda(\mathbf{x})$ is approximated by $\nabla\tilde{\Lambda}(\mathbf{x})$ [25]. The following lemma gives the solution to the gradient of the smoothed function.

Lemma 1: For a constant $\delta > 0$, the gradient of $\tilde{\Lambda}(\mathbf{x})$ can be approximated as

$$\nabla\tilde{\Lambda}(\mathbf{x}) = \frac{2}{\delta}\mathbf{E}_{\mathbf{v} \in \mathbb{C}}\left[\Lambda(\mathbf{x} + \delta\mathbf{v})\mathbf{v}\right]. \tag{6}$$

From Lemma 1, the gradient of the smoothed version of $\Lambda(\mathbf{x})$ at $\mathbf{x}$ is proportional to the expectation term in (6), i.e., $\nabla\tilde{\Lambda}(\mathbf{x})$ can be obtained via the function value of $\Lambda(\mathbf{x})$ at randomly perturbed point $\mathbf{x} + \delta\mathbf{v}$, where $\mathbf{v}$ is a unit vector randomly sampled from the circle $\mathbb{C}$. Essentially, the smoothed version $\tilde{\Lambda}(\mathbf{x})$ bridges the gradient of $\Lambda(\mathbf{x})$ and its perturbed function value, enabling us to obtain the approximate gradient of $\Lambda(\mathbf{x})$ directly from the function value. Approximately, the ascending direction of function $\Lambda(\cdot)$ at $\mathbf{x}$ is given by the expectation of $\Lambda(\mathbf{x} + \delta\mathbf{v})\mathbf{v}$. With the approximated gradient, it is possible to reach the maximum of the concave function $\Lambda(\mathbf{x})$ in the limit. Now we are ready to present the online QC pair configuration algorithm.

The core idea of the algorithm is to continuously perform adjustment of QC configuration on a per-query basis according to the approximated gradient. Before reaching the optimum, the gap of accumulative analytic accuracy between the online configuration and the optimal configuration increases. To evaluate the gap in the long run, for a certain type of query q, define the cumulative regret of the analytic accuracy of an online QC configuration algorithm as

$$\mathcal{R}_q(T) = \max_{\mathbf{x} \in \mathbb{F}_q} \sum_{t=1}^{T} \Lambda_q(\mathbf{x}) - \mathbf{E}\left[\sum_{t=1}^{T} \Lambda_q(\mathbf{x}_t)\right] \tag{7}$$

where $\mathbb{F}_q$ is the convex feasible region of QC configuration of query q given by $\mathbb{F}_q = \{(f, c) : F' \le f \le F, \frac{\xi r_q f}{\tau_q r_q - f} \le c \le C\}$, and $\mathbf{x}_t \in \mathbb{F}_q$ is the algorithm configured QC pair during time slot t. In what follows, we present a lemma to characterize the performance of the expected gradient based online QC configuration across the time span.

Lemma 2: Assume a sequence of QC pairs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \in \mathbb{F}_q$ satisfy $\mathbf{x}_{t+1} = \mathbf{P}_{\mathbb{F}_q}(\mathbf{x}_t + \eta\Lambda(\mathbf{x}_t + \delta\mathbf{v})\mathbf{v})$, where $\eta > 0$ is the learning rate, $\mathbf{v}$ is a unit vector randomly sampled from the circle $\mathbb{C}$, and $\mathbf{P}_{\mathbb{F}_q}(\mathbf{x}) = \arg\min_{\mathbf{x}' \in \mathbb{F}_q} \|\mathbf{x}' - \mathbf{x}\|$ is the projection of $\mathbf{x}$ inside the feasible region $\mathbb{F}_q$, then

$$\mathcal{R}_q(T) \le G\sqrt{(F^2 + C^2)T} \tag{8}$$

where $G \ge \|\nabla\Lambda_q(\mathbf{x})\| \; \forall \mathbf{x} \in \mathbb{F}_q$.

Proof: By constructing the time-varying counterpart of the analytic capability function $\Lambda_q(\cdot)$ as

$$\Lambda'_{q,t}(\mathbf{y}) = \Lambda_q(\mathbf{y}) + \mathbf{y}\left[\Lambda(\mathbf{x}_t + \delta\mathbf{v})\mathbf{v} - \nabla\Lambda(\mathbf{x}_t)\right] \tag{9}$$

we have $\mathbf{E}[\Lambda'_{q,t}(\mathbf{x}_t)] = \mathbf{E}[\Lambda_q(\mathbf{x}_t)]$ as the gradient expectation holds in (6), and $\nabla\Lambda'_{q,t}(\mathbf{x}_t) = \Lambda(\mathbf{x}_t + \delta\mathbf{v})\mathbf{v}$, which suggests that the gradient of $\Lambda'_{q,t}(\mathbf{x})$ is deterministic. Denote by $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathbb{F}_q} \Lambda'_{q,t}(\mathbf{x})$ the optimal QC pair. Let $\mathbf{g}_t = \nabla\Lambda'_{q,t}(\mathbf{x}_t)$ and the iteration of QC pair goes as $\mathbf{x}_{t+1} = \mathbf{P}_{\mathbb{F}_q}(\mathbf{x}_t + \eta\mathbf{g}_t)$.

Since $\mathbb{F}_q$ is convex, $\|\mathbf{x}^* - \mathbf{P}_{\mathbb{F}_q}(\mathbf{x})\| \leq \|\mathbf{x}^* - \mathbf{x}\|$. Then

$$\begin{aligned}\|\mathbf{x}^* - \mathbf{x}_{t+1}\|^2 &= \|\mathbf{x}^* - \mathbf{P}_{\mathbb{F}_q}\left(\mathbf{x}_t + \eta \mathbf{g}_t\right)\|^2 \\ &\leq \|\mathbf{x}^* - \mathbf{x}_t - \eta \mathbf{g}_t\|^2 \\ &\leq \|\mathbf{x}^* - \mathbf{x}_t\|^2 + \eta^2 \|\mathbf{g}_t\|^2 - 2\eta \mathbf{g}_t \left(\mathbf{x}^* - \mathbf{x}_t\right)^\mathsf{T}. \end{aligned} \tag{10}$$

Due to the concavity of $\Lambda'_{q,t}(\mathbf{x})$, we have

$$\begin{aligned}&\Lambda'_{q,t}(\mathbf{x}^*) - \Lambda'_{q,t}(\mathbf{x}_t) \\ &\leq \mathbf{g}_t \left(\mathbf{x}^* - \mathbf{x}_t\right)^\mathsf{T} \leq \frac{\|\mathbf{x}^* - \mathbf{x}_t\|^2 - \|\mathbf{x}^* - \mathbf{x}_{t+1}\|^2 + \eta^2 \|\mathbf{g}_t\|^2}{2\eta}.\end{aligned} \tag{11}$$

Summing (11) over T slots over the time span, we have

$$\begin{aligned}\sum_{t=1}^{T} \Lambda'_{q,t}(\mathbf{x}^*) - \sum_{t=1}^{T} \Lambda'_{q,t}(\mathbf{x}_t) &\leq \frac{\|\mathbf{x}^* - \mathbf{x}_1\|^2}{2\eta} + \frac{\eta T G^2}{2} \\ &\leq \frac{R^2}{2\eta} + \frac{\eta T G^2}{2}\end{aligned} \tag{12}$$

where $R^2 = F^2 + C^2$ holds due to the definition of $\mathbb{F}_q$, G is the gradient upper bound satisfying $\|\mathbf{g}_t\| \leq G \ \forall t$, due to Definition 1. By setting $\eta = R/(G\sqrt{T})$, the above-mentioned accumulative regret can be further bounded as $RG\sqrt{T}$. Since $\mathbf{E}[\Lambda'_{q,t}(\mathbf{x}_t)] = \mathbf{E}[\Lambda_q(\mathbf{x}_t)]$, taking expectations on both sides of (12) gives the lemma. ■

Lemma 2 suggests that QC configuration of a certain query based on the expected gradient achieves asymptotic optimal performance, i.e., $\lim_{T\to\infty} \mathcal{R}_q(T)/T \to 0$. Based on the above-mentioned lemma, the following three steps will be repeatedly executed during each time slot: 1) expected gradient-based QC configuration; 2) feasibility analysis and query admission; 3) observation and update. The process is sketched in Algorithm 2, with detailed description as follows.

1) Expected Gradient-Based QC Configuration: The EN maintains the information of QC configuration and resultant feedback accuracy of each type of query during the latest time slot that it was accepted. For each query $q \in \mathcal{Q}$ during time slot t, randomly select a unit vector $\mathbf{v}_{q,t} \in \mathbb{C}$, and set the QC pair of query q as $\mathbf{x}_{q,t} = \mathbf{y}_{q,t-1} + \delta \mathbf{v}_{q,t}$, where $\mathbf{y}_{q,t-1}$ is an intermediate vector updated in the last round. Next, the feasibility of configuration $\mathbf{x}_{q,t} = [f_{q,t}, c_{q,t}]$ will be tested.

2) Feasibility Analysis and Query Admission: As the total computing resource is limited, the EN cannot accept all the video queries. By sorting all the queries according to $c_{q,t}$ in ascending order, computing resources will be allocated sequentially, until the residual resource is insufficient to meet the requirements of any of the queries. Queries that are not allocated resources will be rejected by the EN. They need to either wait till the next time slot or be directed to the cloud. Notethat information of queries that are outside the set $\mathcal{Q}_t$ will not be updated in the currently time slot.

3) Observation and Update: For all the accepted queries in $\mathcal{Q}_t$, the corresponding configuration $\mathbf{x}_{q,t} = [f_{q,t}, c_{q,t}]$ will be enforced to both the end camera and the EN. Upon the completion of analytics, the accuracy $\Lambda_q(\mathbf{x}_{q,t})$ will be feedback to the EN, which will be used to update the intermediate vector according to $\mathbf{y}_{q,t} = \mathbf{P}_{\mathbb{F}_q}(\mathbf{y}_{q,t-1} + \eta \Lambda_q(\mathbf{x}_{q,t}) \mathbf{v}_{q,t})$.

Algorithm 2: Online Algorithm for Edge Coordinated QC Pair Determination.

```
Require: $\xi$, $C$, $F'$, $F$, $\forall q \in \mathcal{Q}$, upload data rate $r_q$, delay requirement $\tau_q$, and $\epsilon_q$
Ensure: accepted queries $\mathcal{Q}_t$, and the corresponding QC configuration $(f_{q,t}, c_{q,t})$, $\forall t \in \mathcal{T}$, $\forall q \in \mathcal{Q}_t$
for $t = 1, 2, \ldots, T$ do
    for $q \in \mathcal{Q}$ do
        EN randomly select $\mathbf{v}_{q,t} \in \mathbb{C}$
        set $\mathbf{x}_{q,t} = \mathbf{y}_{q,t-1} + \delta \mathbf{v}_{q,t}$
    end for
    sort the queries in ascending order according to $c_{q,t}$, obtain $\{q_1, q_2, \cdots\}$
    identify $K = \arg\max_k \sum_{i=1}^{k} c_{q_i,t} \leq C$
    accept $\mathcal{Q}_t = \{q_1, q_2, \ldots, q_K\}$, and enforce the corresponding QC configurations $\mathbf{x}_{q,t} = [f_{q,t}, c_{q,t}]$, $\forall q \in \mathcal{Q}_t$
    for $q \in \mathcal{Q}_t$ do
        observe $\Lambda_q(\mathbf{x}_{q,t})$
        update $\mathbf{y}_{q,t} = \mathbf{P}_{\mathbb{F}_q}(\mathbf{y}_{q,t-1} + \eta \Lambda_q(\mathbf{x}_{q,t}) \mathbf{v}_{q,t})$
    end for
    $\forall q \in \mathcal{Q} \setminus \mathcal{Q}_t$, update $\mathbf{y}_{q,t} = \mathbf{y}_{q,t-1}$
end for
```

According to Lemma 2, continuously exploring the optimal QC pair does not increase the regret, and the gap between Algorithm 2 and the optimal solution diminishes in the limit. Meanwhile, Algorithm 2 executes iteratively, the most consuming operation in each round is sorting, which can be efficiently solved by typical sorting algorithm with $O(|\mathcal{Q}| \log |\mathcal{Q}|)$ complexity.

V. Performance Evaluation

To demonstrate the advantages of the proposed online QC configuration algorithm, simulation results are provided in this section. In particular, we focus on the video analytic accuracy of the accepted queries across the time span, and show how the performance varies with the edge computing capacity, as well as the performance gain against other noncooperative benchmark algorithms.

A. Simulation Setting

1) System Configuration: The system identifies $|\mathcal{Q}| = 20$ types of video queries, each of which has an analytic capability function $\Lambda_q(\cdot)$ unknown to the EN. Since an established capability function is unavailable, we adopt the experimental results of RetinaNet-50 from [24] to fit the underlying capability function. In particular, the accuracy results (average precision) of the corresponding setting of input video quality and execution time (as shown in Table II) are fed into a nonlinear concave function for curve fitting, where the logarithmic

TABLE II
RetinaNet-50 Dataset Analytic Result: Image Scale, Execution Time, and Analytic Accuracy ([24, Table 1e])

Image scale (pixels)	400	500	600	700	800
Execution time (ms)	64	72	98	121	153
Analytic accuracy (%)	30.5	32.5	34.3	35.1	35.7

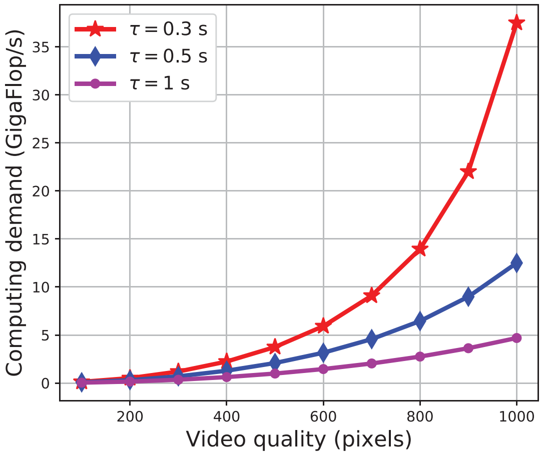


Fig. 3. Computing demand with video quality at different delay requirements.

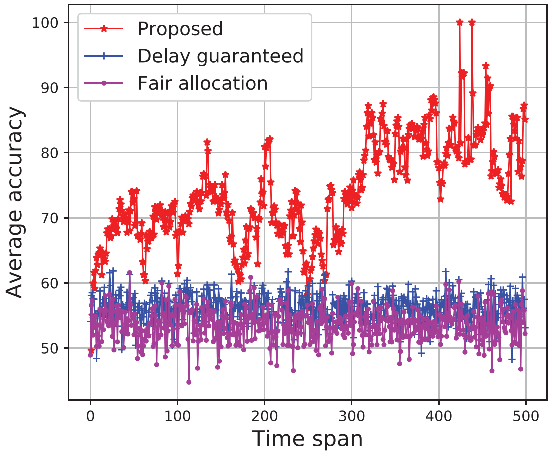


Fig. 4. Comparison of average accuracy (percentage) across the time span (slots) when edge computing capacity C equals 100 GigaFlop/s.

$\Lambda(f, c) = w_1 \log f + w_2 \log c + w_3$ is adopted. w_1 and w_2 are, respectively, the corresponding weight of the impact of the video quality and the computing resource on the analytic accuracy, while w_3 is a constant. The weights w_1, w_2, and w_3 are then randomly perturbed to emulate the underlying analytic capability function of different types of queries. It is worth noting that the logarithmic function is only one form of function that fits the data source, similar curve fitting results and evaluation can be conducted based on other concave functions, such as square root. The frame rates of the end cameras are considered to be fixed, while the frame resolution acts as the knob for quality configuration.

For end camera capability, the pixels (quality) of captured frames can vary from $F' = 100$ pixels to $F = 1000$ pixels [24]. The QC scaling factor $\xi = 3.75 \times 10^3$ flop/bit, indicating that a video clip of 1 Mb requires 3.75 GigaFlop processing capacity. The delay requirement of all the queries are uniformly and randomly sampled from $[0.1, 1]$ s, and the uplink data rate r_q of the camera corresponds to each query is sampled from 1 to 80 Mb/s.

2) Comparison Benchmarks: We consider the following two benchmark algorithms.

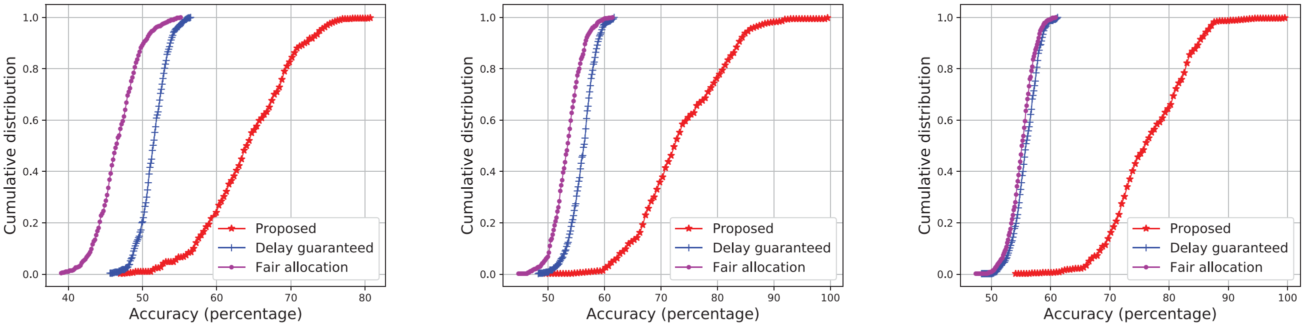
1) *Delay guaranteed:* Without coordination, each end camera streams video frames at a certain quality level. Upon the arrival at the EN, the minimum computing resources that required to satisfy the delay constraint will be first determined and sorted in ascending order. Queries will be accepted sequentially based on the required computing resource, the remaining ones will be rejected by the EN.
2) *Fair allocation:* Similarly, EN will receive uncoordinated video frames from related source cameras. Afterward, EN computing resources will be evenly distributed among all the queries. As a result, the delay requirements of certain queries may not be satisfied, and computing resources on those queries will be wasted.

Those benchmark algorithms are noncooperative, end cameras, and the EN make decisions sequentially; hence, they are not optimized in video analytic accuracy.

B. Simulation Results

Fig. 3 shows the delay-guaranteed computing demand when video quality varies from 100 to 1000 pixels. The uplink data rate is set to $r_q = 5$ Mb/s. From (5), in order to guarantee the delay requirement, the computing capacity c_q should not be less than $\xi r_q f_q / (\tau_q r_q - f_q)$, i.e., the computing demand c_q grows exponentially with video quality f_q. It is also costly to meet the stringent delay requirement when video quality is high, as it requires significantly more computing resource to process the frames. For instance, when requesting video quality at 1000 pixels, the required computing capacity to satisfy delay requirement $\tau_q = 1$, 0.5, and 0.3 s are 4.7, 12.5, and 37.5 GigaFlop/s, respectively. It can be seen that the extra computing resources required to reduce the delay from 0.5 to 0.3 s is 3× more than that of reducing the delay from 1 to 0.5 s. Such observations are useful to practical edge resource deployment.

To demonstrate the advantages of edge coordinated QC determination algorithm, we show the accuracy (averaged over the accepted queries) in each time slot across the time span in Fig. 4. It can be seen that, as the algorithm gradually explores the optimal QC pair, the accuracy of the proposed algorithm grows since more information on the analytic capability function is accumulated. In contrast, the achieved accuracy of both the delay guarantee and fair allocation algorithm does not vary with time, as they are making time-independent decisions without end-edge coordination. To further illuminate the accuracy comparison results of all the three algorithms, we plot the cumulative density function of the average accuracy across 500 time slots in Fig. 5. It is shown that the proposed algorithm always provides the highest average accuracy. When the edge computing resource increases from 50, 100, to 300 GigaFlop/s, the corresponding median accuracy achieved by the proposed algorithm increases from 64%, 73%, to 76%, respectively. As a comparison, the



(a)

(b)

(c)

Fig. 5. Comparison of cumulative density distribution of mean accuracy with varying edge computing capacity C. (a) $C = 50$ GigaFlop/s. (b) $C = 100$ GigaFlop/s. (c) $C = 300$ GigaFlop/s.

highest accuracy achieved by the benchmark algorithms is 60%, which stops increasing after the computing resource is higher than 100 GigaFlop/s, it is reasonable as the streamed video quality is uncoordinated and does not match with the available computing resources.

C. Discussions

When the delay requirement is guaranteed, there is clear tradeoff between the query acceptance ratio and the analytic accuracy. As the edge platform explores different QC configurations, the video quality and demand on computing resources increases. In turn, the query acceptance ratio drops due to the limited computing resource. Meanwhile, more bandwidth resources are required to stream the high quality video frames. In case computing and bandwidth resources are of different costs, EN can strike the balance between bandwidth and computing demands during the exploration.

Also note that the analytic capability function may vary with time, as it is determined by many other time-varying factors, in addition to the video quality and the computing resource. The simplifications we made at the beginning enable us to narrow the focus to the impact of video quality and computing resource. The impact of time-varying factors, such as the available camera uplink bandwidth, the changing camera direction, on the analytic accuracy of video queries deserves further investigation.

VI. Conclusion

In this article, we had investigated the end-edge-cloud collaboration for live video analytics. To respond to video queries with low latency at the edge, a dynamic edge configuration algorithm was developed, which adjusts the quality of generated video frames at the end cameras, as well as the allocated edge resources for each video query. Theoretical analysis demonstrated that the proposed algorithm achieves near-optimal utility, while satisfying the latency requirements. The performance was further validated by extensive simulation results. The proposed algorithm provides a low-latency solution to accurate live video analytics, which can be readily applied to practical surveillance systems. For the future work, in addition to video streaming and analytics at the edge, we will take into account the video transcoding schemes to further improve the system performance.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3113–3122, May 2019.

[3] W. Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 579–590, Mar. 2019.

[4] Seagate, "Video surveillance storage: How much is enough?" Accessed: Jun. 2019 [Online]. Available: https://www.seagate.com/ca/en/tech-insights/how-much-video-surveillance-storage-is-enough-master-ti/

[5] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and delay-tolerance," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, Boston, MA, USA, 2017, pp. 377–392.

[6] C.-C. Hung *et al.*, "VideoEdge: Processing camera streams using hierarchical clusters," in *Proc. IEEE/ACM Symp. Edge Comput.*, Bellevue, WA, USA, 2018, pp. 115–131.

[7] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, Sep./Oct. 2017.

[8] C.-F. Lai, W.-C. Chien, L. T. Yang, and W. Qiang, "LSTM and edge computing for big data feature recognition of industrial electrical equipment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2469–2477, Apr. 2019.

[9] P. Yang, N. Zhang, S. Zhang, F. Lyu, L. Yu, and X. Shen, "Asymptotic optimal edge resource allocation for video streaming via user preference prediction," in *Proc. IEEE Int. Conf. Commun.*, Shanghai, China, 2019, pp. 1–6.

[10] A. Mehrabi, M. Siekkinen, and A. Yla-Jaaski, "Edge computing assisted adaptive mobile video streaming," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 787–800, Apr. 2019.

[11] D. Wang *et al.*, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Trans. Serv. Comput.*, vol. 12, no. 5, pp. 685–697, Sep./Oct. 2019.

[12] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 915–929, Apr. 2019.

[13] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug. 2018.

[14] Q. Liu, S. Huang, J. Opadere, and T. Han, "An edge network orchestrator for mobile augmented reality," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, 2018, pp. 1–9.

[15] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing-based face identification and resolution scheme in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1910–1920, Aug. 2017.

[16] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14–20, Sep./Oct. 2017.

[17] X. Peng, J. Ren, L. She, D. Zhang, J. Li, and Y. Zhang, "BOAT: A block-streaming app execution scheme for lightweight IoT devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1816–1829, Jun. 2018.

[18] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2019.2903240.

[19] Microsoft Azure, "The future of computing: Intelligent cloud and intelligent edge." Accessed: Jun. 2019. [Online]. Available: https://azure.microsoft.com/en-us/overview/future-of-cloud/

[20] K. Hsieh *et al.*, "Focus: Querying large video datasets with low latency and low cost," in *Proc. 13th USENIX Symp. Operating Syst. Des. Implementation*, Carlsbad, CA, USA, 2018, pp. 269–286.

[21] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: Scalable adaptation of video analytics," in *Proc. ACM SIGCOMM*, Budapest, Hungary, 2018, pp. 253–266.

[22] S. Han *et al.*, "MCDNN: An approximation-based execution framework for deep stream processing under resource constraints," in *Proc. ACM MobiSys*, Singapore, 2016, pp. 123–136.

[23] T. Lattimore, K. Crammer, and C. Szepesvári, "Linear multi-resource allocation with semi-bandit feedback," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, 2015, pp. 964–972.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, Venice, Italy, 2017, pp. 2980–2988.

[25] A. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, Vancouver, BC, Canada, 2005, pp. 385–394.

Peng Yang (S'16–M'18) received the B.E. and the Ph.D. degrees from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2013 and 2018, respectively.

He is currently a Postdoctoral Fellow with the Broadband Communications Research (BBCR) Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, where he was a visiting Ph.D. student from September 2015 to September 2017. His research interests include software-defined networking, mobile edge computing, and video streaming.

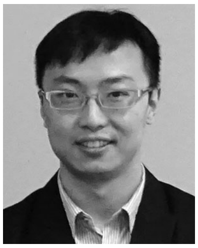
Feng Lyu (S'16–M'18) received the B.E. degree in software engineering from Central South University, Changsha, China, in 2013, and the Ph.D degree in computer science and engineering from the Department of Computer Science and Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018.

Since September 2018, he has been working as a Postdoctoral Fellow with BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include vehicular ad hoc networks, space-air-ground integrated network, cloud/edge computing, and big data driven application design.

Dr. Lyu is a Member of the IEEE Computer Society and the IEEE Communications Society.

Wen Wu (S'13–M'19) received the B.E. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2012, the M.E. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2015, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2019.

Since 2019, he is currently working as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include millimeter-wave networks and artificial intelligence (AI)-empowered wireless networks.



Ning Zhang (M'15–SM'18) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2015.

After that, he was a Postdoctoral Research Fellow with the University of Waterloo and University of Toronto, Toronto, ON. Since 2017, he has been an Assistant Professor with the Department of Computing Sciences, Texas A&M University Corpus Christi, TX, USA. His research interests include wireless communication and networking, mobile edge computing, machine learning, and physical-layer security.

Dr. Zhang serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE ACCESS, *Vehicular Communications* (Elsevier), and *IET Communications*, and an Area Editor for *Encyclopedia of Wireless Networks* (Springer, 2021).

Li Yu (M'08) received the B.Sc. degree in information engineering and the Ph.D. degree in communication and information system from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 1992 and 1999, respectively.

She joined the Huazhong University of Science and Technology, in 1999, where she is currently a Professor with the School of Electronic Information and Communications. Her research interests include image and video coding, multimedia communications, social networks, and wireless networking.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks.

Dr. Shen was the recipient of the R.A. Fessenden Award in 2019 from IEEE Canada, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society. He was also the recipient of the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004, 2007, 2010, and 2014, from the University of Waterloo and the Premier's Research Excellence Award (PREA), in 2003, from the Province of Ontario, Canada. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, the IEEE Infocom'14, the IEEE Vehicular Technology Conference'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE International Conference on Communications'10, the Tutorial Chair for the IEEE Vehicular Technology Conference'11 Spring, the Chair for the IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He is the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL and the Vice President on Publications of the IEEE Communications Society.