IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 22, 2019, accepted April 21, 2019, date of publication May 3, 2019, date of current version May 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914613

Distributed Architectures for Intensive Urban Computing: A Case Study on Smart Lighting for Sustainable Cities

HIGINIO MORA[1], JESÚS PERAL[2], ANTONIO FERRÁNDEZ[2], DAVID GIL[1], AND JULIAN SZYMANSKI[3]

[1]Department of Computer Science Technology and Computation, University of Alicante, Alicante, Spain
[2]Language and Information Systems Department, University of Alicante, Alicante, Spain
[3]Department of Computer Systems Architecture, Gdansk University of Technology, 80-233 Gdańsk, Poland

Corresponding author: Higinio Mora (hmora@ua.es)

This work was supported in part by the Spanish Research Agency (AEI) and the European Regional Development Fund (FEDER) through the project CloudDriver4Industry under Grant TIN2017-89266-R, in part by the Spanish Ministry of Science, Innovation and Universities through the Project ECLIPSE-UA under Grant RTI2018-094283-B-C32, and in part by the Conselleria de Educación, Investigación, Cultura y Deporte of the Community of Valencia, Spain, within the Program of Support for Research under Project AICO/2017/134 and Project PROMETEO/2018/089.

ABSTRACT New information and communication technologies have contributed to the development of the smart city concept. On a physical level, this paradigm is characterized by deploying a substantial number of different devices that can sense their surroundings and generate a large amount of data. The most typical case is image and video acquisition sensors. Recently, these types of sensors are found in abundance in urban spaces and are responsible for producing a large volume of multimedia data. The advanced computer vision methods for this type of multimedia information means that many aspects can be dynamically monitored, which can help implement value-added applications in the city. However, obtaining more elaborate semantic information from these data poses significant challenges related to a large amount of data generated and the processing capabilities required. This paper aims to address these issues by using a combination of cloud computing technologies and mobile computing techniques to design a three-layer distributed architecture for intensive urban computing. The approach consists of distributing the processing tasks among a city's multimedia acquisition devices, a middle computing layer, known as a cloudlet, and a cloud-computing infrastructure. As a result, each part of the architecture can now focus on a small number of tasks for which they are specially designed, and data transmission communication needs are significantly reduced. To this end, the cloud server can hold and centralize the multimedia analysis of the processed results from the lower layers. Finally, a case study on smart lighting is described to illustrate the benefits of using the proposed model in smart city environments.

INDEX TERMS Mobile cloud computing, data processing, distributed architectures, smart city, urban computing.

I. INTRODUCTION

Embedded systems have extended to new application areas such as healthcare, the automotive industry, robotics, home automation and smart cities, consequently leading to the development of the Internet of Things (IoT). This new paradigm consists of networking any device with processing, sensing and computation capabilities ready to understand the environment. With the growing presence of wireless communication technologies, such as wireless local area networks (WLAN), long-term evolution (LTE) communications and radio-frequency identification (RFID), devices can be connected to the Internet and remote monitoring and management performed through cloud applications. Ubiquitous IoT-enabled features allow us to measure, infer and understand environmental indicators in many application areas. Combining the IoT paradigm and the cloud offers new possibilities for upgrading service quality [1]. Indeed, edge

The associate editor coordinating the review of this manuscript and approving it for publication was Miltiadis Lytras.

computing possibilities aim to handle the new processing needs of IoT applications. This paradigm is near the data sources and brings cloud resources and services to the edge of the network in order to minimize latency minimization and improve service quality [2].

One of the most common IoT scenarios is the smart city [3], [4]. It is characterized by two main features: firstly, a collection of distributed sensors that sense the city by capturing and measuring a set of magnitudes on what is occurring in any place, at any time. Secondly, suitable tools and applications analyses data for generating value-added information. This information helps improve decision-making, anticipate problems to resolve them proactively and coordinate resources in order to act efficiently. In this way, smart cities are increasingly being equipped with sensing devices to capture data. Examples of such are smart metering to track utilities consumption, citizen traceability, transportation facilities, etc. These data provide a wealth of value-added services to citizens [5].

Edge computing provides competitive advantages in terms of latency for processing the diverse data acquired by city sensors [6]. This paradigm plays a key role in the development of citizen-centric applications since edge computing can be deployed on several layers at the edge of the network, e.g. fog, mobile edge computing (MEC), ad hoc clouds, and cloudlets [7].

Sensing devices can acquire a very large volume of data. In addition, new big data capabilities enable data-driven smart city transformation [8] and implementing sustainable initiatives for the city [9].

Sensors, especially image and video acquisition sensors are becoming increasingly common in urban spaces. Their purpose is to control traffic and improve public safety by reducing and preventing crime. Latterly, video cameras have been mass deployed in many cities, with cases like London, Chicago and Vancouver where thousands of cameras are installed [10]. However, in most cases, these cameras are underused. The multimedia data captured by these devices offer many more possibilities than the functions for which they were installed. Analyzing the multimedia data captured could effectively contribute to transforming the city into a smart city.

There is much research on image and video analysis. By applying computer vision methods and techniques to the multimedia data collected by the cameras installed in the urban environment, many aspects can be dynamically monitored, which can help implement value-added applications in the city.

Context awareness applied in smart city environments introduces new ways of deploying user-centric services. For example, detecting people walking down a street to activate intelligent lighting systems, assessing the traffic flow level to adjust traffic lights, estimating the number of people in a street to ensure adequate safety services, identifying obstacles on the road that may hinder traffic, detecting rubbish or other items to manage as well as improve urban cleaning, etc. These possibilities can significantly improve an IoT paradigm's development. However, there are significant challenges associated with the idea of using a set of cameras in an urban area to obtain more detailed semantic data, among which, the following are related to this research:

- Running complex algorithms for image processing and computer vision has a high computational cost. In most cases, acquisition devices are not powerful enough to compute the required processes. In addition, not every edge computing option is capable of providing a satisfactory performance.
- Transferring raw multimedia data acquired by a large number of sensors can saturate communication networks and cloud storage systems.
- Analyzing a large amount of multimedia data from multiple cameras can cause a bottleneck in the remote servers.
- The computing power available to the centralized processing servers bypasses the dynamic operations of many video or image acquisition systems that capture information only when motion occurs. The fact that computing power is provided for the peak work may mean that resources are underused most of the time.

This study's **main objective** is aimed at researching how to leverage cloud computing technologies and mobile computing techniques and provide flexibility for supporting multimedia data processing in IoT environments. The ultimate objective is to use the images and videos captured by cameras deployed in the city to provide information and knowledge that can help design high-level applications for smart cities.

Accordingly, the technological developments of recent times are providing acquisition sensors with Internet connectivity and processing capabilities. The concept of IoT is now evolving towards the development of complex systems with smarter *things* [11], [12].

Our study's **key contributions** can be summarized as follows:

- We provide a constructive review to reach the knowledge border both in terms of video and image processing techniques and the distributed architectures for processing them. We raise some useful findings to guide proposal design to address intensive multimedia computing in urban environments.
- We offer a distributed architecture that can benefit from existing computing capabilities in acquisition devices and cloud infrastructure and exploits the opportunities of deploying additional middle-platforms along the network to reinforce the performance so that it can handle big multimedia data processing.
- We put the proposed architecture into practice, showing how advanced services based on analyzing the image and video data acquired from the city can be implemented.

The main **novelty** of this work lies in extending the IoT paradigm with mobile cloud computing techniques to build a distributed architecture, enabling intensive urban computing.

The rest of the paper is organized as follows: Section II reviews multimedia data processing techniques and architectures. Section III describes the proposed distributed architecture. Section IV introduces a case study to better demonstrate the proposal and its advantages for intensive multimedia processing. Finally, Section V provides some conclusions and future directions.

II. BACKGROUND AND LITERATURE REVIEW

This section addresses the research conducted on multimedia data processing. Firstly, we review image and video processing techniques, followed by the architectures that support them. A final subsection summarizes the contributions to previous studies.

A. IMAGE AND VIDEO PROCESSING TECHNIQUES

There is a significant amount of research on this topic. Only some of the recent and most representative works are discussed.

The healthcare sector has traditionally been the subject of intensive research on computer vision, image analysis and pattern recognition [13], [14]. These areas have substantially progressed during the past several decades, enabling the development of applications for advanced predictive analytics and therapy [15].

Video-based and image-based human detection has been an important problem for decades, given its relevance to a wide range of applications in robotics, intelligent transportation, collision prediction, driver assistance, car safety, road scene understanding, surveillance systems, demographic recognition, etc.

Classic pedestrian detection methods involve first extracting image features and then applying classifiers such as support vector machines, adaptive boosting, decision trees, among others to classify the features [16]. Dalal and Triggs proposed one of the most widely used pedestrian detection algorithms, which was characterized by the histogram of oriented gradient (HOG) [17]. HOG describes the distribution of the intensity and direction of the gradient of the local image regions. Based on HOG, Felzenszwalb et al. defined the deformable part model (DPM) [18], which studied HOG relationships at different scales in the image pyramid. In order to increase computational efficiency, the aggregate channel features (ACF) algorithm is designed to first estimate the HOG's large-scale effectiveness and then omit the useless parts in small-scale images.

Large amounts of training data and increased computing power have led to recent deep architecture advances (typically convolutional neural networks) on diverse computer vision tasks: large-scale classification and detection, semantic labeling, etc. [19]. These results have inspired applying deep architectures to human tasks [20], [21].

Deep learning is part of the family of machine learning methods and is the most promising method in modern image recognition and semantic segmentation [22]. Because of the computing power's rapid development, due to the graphics processing unit's (GPU) parallel processing, deep learning has recently made some breakthrough results within the machine-learning field [23]. In short, many novel techniques have been developed due to technology advances and because mining techniques have matured. In addition, the use of mobile apps contributes to the proliferation of these methods [24].

Deep learning consists of multiple processing layers that are capable of learning representations of data with multiple levels of abstraction. These methods are responsible for huge improvements in the aforementioned applications, speech and facial recognition, natural language processing, bioinformatics, etc.

In 2014, Benenson et al. conducted a study on more than 40 pedestrian detection methods included in the Caltech Pedestrian Detection Benchmark [25]. They determined three main families of approaches: (1) DPM-based [26], (2) deep neural networks (DNN) [27] and (3) decision forests [28]. They concluded that overall, DPM variants, deep networks and (boosted) decision forests all reached top performance in pedestrian detection (around 37% miss-rate on Caltech-USA).

Two main practical requirements are associated with pedestrian detection methods: high accuracy and real-time performance. Pedestrian detectors need to be accurate and fast enough to run on systems with limited computing power. As mentioned above, pedestrian detection methods have employed a variety of techniques and features. Some have focused on increasing detection performance [29], where others centered on accuracy [30]. Recently, the novel range of DNN-based methods have shown impressive improved accuracy [19]. However, DNN models are known to be very slow, especially when used as sliding-window classifiers.

Regarding the better features, the most popular approach for improving detection quality is to increase or diversify the features computed over the input image [16]. By having richer and higher dimensional representations, classification becomes somewhat easier, enabling improved results. To date, a large set of feature types have been explored: edge information, color information, texture information, local shape information, covariance features, among others. More and more diverse features have proven to systematically improve performance. However, DNNs possess one great advantage: they extract features directly from raw image values. Benenson et al. conclude that during the last decade, improved features have been a constant driver for enhancing detection quality, and it seems that this will be the case in the coming years. Most of this improvement was obtained by extensive trial and error. The next scientific step would be to develop a more profound understanding of what makes good features good and how to design even better ones (including deep learning).

Apart from isolated multimedia processing, information centralization and the cloud computing evolution enable big data analysis and advanced result management [31], [32]. In this regard, storage and communication problems are added. Some proposals suggest that one solution would be storing or transmitting a descriptive summary of the original video. For example, Zhang *et al.* [33] propose a summarization technique that enforces video stability and preserves well-aesthetic frames, which is based on multitasking feature selection to efficiently discover the semantically important features. Xia *et al.* [34] propose a different solution that uses intelligent monitoring and recording systems, which include the front-end image acquisition system and the back-end data processing platform. After the image data is recorded, the front-end processing system analyses the image data and automatically extracts some data from the passing vehicles (time, location, direction, car color, registration plate number, etc.). The information is then sent to the back-end data processing platform for deeper analysis, looking at factors such as vehicle trajectory tracking and traffic state estimation.

As a conclusion, researchers agree that there is a need for cleaning, filtering, feature extraction and simplification, which is computationally expensive. For example, Ang and Seng [35] summarized recent developments for big sensor data systems in various representative studies on urban environments, including air pollution monitoring, assisted living, disaster management systems and intelligent transportation. To deal with the high volume of data, the authors discuss an intelligent data forwarder that is embedded in each data source with context-aware capability. The key idea is to reduce the data at each stage of the data collection/generation process.

B. MULTIMEDIA ARCHITECTURES

This subsection reviews how architectures have evolved to face the problems described in the previous subsection. These proposals can be used to address similar problems related to other IoT applications. Table 1 presents a summary of the proposals.

Traditionally, multimedia data processing systems have been equipped with elements that supplement specialized local processing and provide the power required by applications. To this end, dedicated very-large-scale integration implementations of image and video processing algorithms have been designed for several platforms.

Some proposals are based on reconfigurable hardware to make the architectures more versatile [36], [37]. These designs are implemented in reconfigurable Field-Programmable Gate Array (FPGA) cards. Other alternatives propose using systems composed of digital signal processing (DSP) elements known for providing high performance and low response times in multimedia processing [38], [39]. DSP and reconfigurable systems were also proposed to combine these two aspects and build versatile systems with high-throughput architectures that integrate into the same system [40].

Graphics processor units (GPU) have evolved immensely and the ease to which they can be programmed due to the popularity of languages, such as computer unified device architecture (CUDA), has promoted research into a large

TABLE 1. Architecture proposals for multimedia processing.

Architecture	Proposal
Reconfigurable architecture	Dynamically Reconfigurable [36]
	FPGA implementation [37]
DSP architecture	Image Processing [38]
	Ultrafast DSP [39]
Reconfigurable + DSP	Image Processing Platform [40]
GPU architecture	Image Enhancement [41]
	Hardware accelerator [42]
	Video analysis [43]
Reconfigurable + GPU	Image processing [44]
Multicore CPU + GPU	Video Summarisation Approach [45]
	Real-time parallel image processing [46]
Cloud computing	Storage and analysis of very-large images [54]
	Massive Remote Sensing Framework [55]
Parallel architectures	Hadoop image processing [48]
	Scalable image analysis using Hadoop [49]
	MapReduce in multimedia big data [50]
	Distributed image classification pipeline [51]
Cloud parallel architecture	Remote sensing image analysis [52]
	Scalable image processing toolbox [53]
Mobile cloud computing	Enable rich multimedia app. [58]
	Human-centric multimedia [59]
	Service image placement [60]
	Workload scheduling [61]
	Multimedia streaming [62]
	Mobile streaming QoS approach [63]
Cloudlet	Scalable Cloudlet [72]
	Cloud-assisted complex event monitoring [73]

number of proposals based on these devices [41]–[43]. These architectures provide intensive multimedia processing due to their high level of parallelism. Other proposals aim to take advantage of these processing units with other elements to improve performance, such as combining GPUs with reconfigurable architectures [44] and multicore central processing units [45], [46].

Research addressing multimedia big data processing challenges [47], mainly recommends architectures and parallel computing methods that take advantage of the high degree of algorithm parallelization.

Hadoop and MapReduce have become the most frequently used image processing platforms [48]–[51]. These parallel processing techniques favor the use of computing resources on the cloud for processing multimedia data, offering very competitive proposals for massive parallel processing [52], [53]. Additionally, the cloud computing paradigm is emerging as a solution to supply specific computing resources for applications with massive computing needs. Computing as a utility has overcome many of the barriers to designing processing intensive applications without requiring a large proprietary infrastructure. Cloud-based multimedia data processing has also been taking advantage of this trend, especially in those cases where a large multimedia database is stored on the cloud [54], [55].

The great challenge associated with cloud-based multimedia data processing is transmitting the information to the remote server where it will be processed. Image transmission techniques can mainly be divided into two categories: (1) improving transmission protocol design [56] and (2) image data encoding and compression [57]. This can be problematic for applications that process a lot of multimedia information, due to a large number of sensors (e.g. cameras) or the large volume of data collected (e.g. video streaming). In these cases, transmitting raw information for remote processing is not feasible and processing techniques are required for the acquisition devices.

Mobile cloud computing (MCC) is a computational scheme that distributes the processing load between acquisition devices and cloud infrastructure. With this paradigm, processes can be offloaded from the cloud to improve running applications' performance. There are some proposals that apply these techniques to multimedia processing tasks to improve their flexibility and performance [58], [59]. Some of the most common applications include solutions that broadcast both image [60], [61] and video data [62], [63] among acquisition devices and cloud servers.

A middleware layer can manage the running of tasks between acquisition devices and cloud resources by deciding how processes are distributed based on different aspects, such as energy consumption, response times or priority [64].

The cloudlet infrastructure is a step forward in bringing cloud resources closer to IoT devices. Cloudlet computers should have the same general architecture as cloud servers, but they are less powerful, smaller and less expensive [65], [66]. Thus, the cloudlet's physical proximity simplifies the challenge of meeting the required bandwidth and provides better response times [67]. In addition, if the number of user devices increases, the cloudlet can move this extra processing load to core cloud systems in order to meet the quality-of-service (QoS) requirements. This cloudlet infrastructure is usually deployed within a local area network (LAN) and can be accessed by wireless network connections [68]. WLAN bandwidth is typically two orders of magnitude higher than the wireless Internet bandwidth available on a mobile device [65]. Cloudlet deployment is specifically designed to provide flexibility to the cloud provisioning. Consequently, it can be used for improving multimedia applications' QoS in citizen-centric environments such as smart cities [69]. With this feature, those companies interested in boosting their applications' performance can do so by placing a cloudlet close to their users. For example, inside a shopping center, a hospital, an airport or a city district.

Cloudlets would be decentralized, widely dispersed and self-managing [70]. To this end, they are not as efficient as core cloud computing because the resources are very sparse in a wide area (for example, each city district may have its own cloudlet) [71]. With respect to security and privacy issues, cloudlet nodes are closer to the users and companies providing the service own the platforms. As such, computing the data in the cloudlet poses less of a risk than in a remote server.

Finally, recent proposals recommend using cloudlets to reduce power consumption and the network delay on multimedia applications [72], [73]. In these proposals, cloudlets are a widely distributed computing infrastructures that are basically leveraged by nearby mobile devices.

C. FINDINGS

After reviewing the literature, some findings can be drawn that justify and summaries our contributions to the previous works:

- Reviewing architectures and video and image processing in distributed environments proved that software and technology have seen huge progress in recent years.
- Multimedia data processing and advanced data mining techniques from cameras deployed in cities present a critical complexity. This issue needs to be better addressed by discovering novel paradigms and architectures.
- Researchers agree that there is a need for pre-processing before multimedia data is transmitted to remote cloud servers in order to stop communication channels and storage systems being saturated. However, these processes are computationally expensive to be run only on acquisition devices.
- The trend for upgrading the overall performance of complex applications deployed in distributed environments (such as IoT, mobile apps, etc.) is finding ways to use the networks and Internet to provide additional computing power to mobile devices and 'things'. Proposals are intended to add computational resources to the network and bring more computing capabilities near to the datasources and users.
- MCC techniques become a computational scheme that partially overcome the previous problems, by allowing processing load distribution between acquisition devices and cloud infrastructure.

The research conducted in this paper focuses on addressing the main challenge associated with intensively processing multimedia data acquired on the fly in real IoT environments: the difficulty in transmitting and processing the huge amount of generated data.

The **key idea** is based on using the network and cloud infrastructure to provide several computing layers and further capabilities for intensive processing. Accordingly, the proposal consists of gradually computing the tasks across a distributed three-layer architecture, which includes the networked 'things', cloudlets and cloud infrastructure. The idea of introducing more computing power through cloudlets is not new, however, what is relevant is integrating deployed 'things' and the cloud resources to build a comprehensive architecture to enhance IoT applications and enable intensive multimedia data processing.

As a **technical contribution**, distributed architecture can implement advanced multimedia applications and multimedia big data analysis from the information acquired by the many sensors in a smart city environment. This architecture's potential to handle the intensive data processing in urban environments is illustrated through a case study on a smart lighting application.

III. ARCHITECTURE FOR MULTIMEDIA PROCESSING

This section is divided into three subsections that this research proposal expands on. The first subsection describes the problem and the potential computing resources' deployment for intensive urban computing. The second subsection proposes a flexible architecture and a computing model to address distributed processing of multimedia data acquired in smart city environments.

This section is based on an application that detects human presence in multimedia data taken by surveillance cameras [74]. The application context is an urban area where several video cameras and street lamps have been deployed, as shown in Fig. 1.

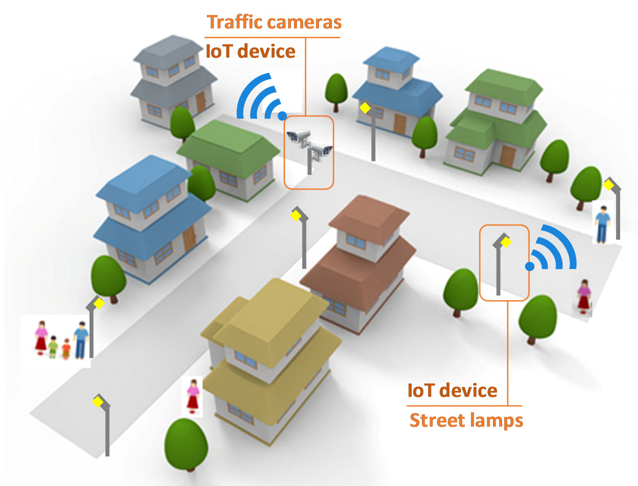

FIGURE 1. Application context deployment.

In this case, video cameras are context-aware sensors and acquire knowledge about the environment, and street lamps are actuators, in that they render the smart lighting service. Both of them are considered IoT devices with communication capabilities. This scenario is part of an IoT system and represents an example where the convergence of information and communication technologies [75] can help develop smart city services. In addition, this configuration is quite common in today's cities where the cameras are already deployed to monitor traffic and provide surveillance. For example, Fig. 2 shows the cameras deployed in the center of Vancouver, where more than 2,000 video cameras are installed. The information acquired can help make decisions about traffic management and security issues in real time. However, this large amount of information can be further exploited to advance in smart city service development, for example, intelligent systems based on live street analysis that enables sustainable city implementation.

A. DISTRIBUTED MULTIMEDIA PROCESSING

The vast amount of data acquired by image and video sensors and the complexity of multimedia big data applications may require powerful datacenters and communication networks with rich computing resources and very high bandwidths.

To address this challenge, the proposed architecture performs the processing as close to where the data are acquired as possible. To this end, communication needs are reduced and the central server's computing resources can perform analysis and big data processing rather than the non-value-added stages of multimedia methods. This approach aims to produce the best response times and application performance. Once the data has been processed (or partially processed), the results can be sent to the central system for a combined analysis.

Instead of transmitting raw multimedia data captured by cameras, the example application relays the presence or absence of people in images and videos. As can be easily observed, this drastically reduces the amount of information sent to the cloud server.

However, computing complex multimedia applications in data capture devices would not be feasible. Implementing complex multimedia algorithms requires powerful computational resources. In many cases, the infrastructure employed by the cameras and acquisition sensors deployed does not have or has very little processing capability. In this situation, device hardware has to adapt to the application needs, incurring an economic cost, which is unaffordable in many cases. Moreover, in most scenarios, installing reconfigurable coprocessors, GPUs or DSP in the acquisition devices to process the algorithms is not physically possible, so their computing power is limited.

Furthermore, moving the processing to a central server in the cloud does not seem to be the best choice for multimedia data. Scenarios with several acquisition sensors require a lot of bandwidth to transmit raw multimedia data. In addition, this configuration requires a very powerful centralized cloud infrastructure capable of simultaneously processing the computational load of all the data with acceptable response times.

B. PROPOSED DISTRIBUTED ARCHITECTURE

To address the above issue, this research aims to design a flexible architecture where processing resources are deployed at various levels depending on the acquisition sensors' capabilities, installation possibilities and the cloud infrastructure. With this model, multimedia data applications can be partially processed at each level according to the operating conditions and the execution context for the best performance. The proposed architecture is based on our previous works on distributed computing for IoT and cyber-physical systems [76], [77]. In addition, the findings from the review

FIGURE 2. Video cameras in Vancouver city center (~ 5 km^2). Source: Vancouver public space network (http://vancouverpublicspace.ca/).

of related work led us to propose a distributed architecture, as it would take advantage of the flexibility and possibilities associated with providing several computing layers for multimedia processing.

The proposed architecture configuration adds an intermediate processing level between the acquisition sensors and cloud infrastructure. This layer, known as a cloudlet, has advantages over processing in the remote cloud server [65], as it can bring the computing closer to where the data is generated, distribute the processing cost between intermediate nodes and reduce the need to communicate with remote cloud servers. In fact, this cloudlet layer is a mini cloud server that can be equipped with specialized processing units (such as GPUs) and can be deployed by city managers where needed. Because of this, it is the most suitable edge computing option to handle the huge computing needs associated with image and video processing in a city context.

This distributed architecture's objective is to share multimedia processing between acquisition devices, the intermediate cloudlet layer and the cloud infrastructure. This design enables cloud servers to focus on the application operation and high-level analysis of the multimedia algorithms' results without having to allocate resources to process them.

The intermediate results do not need to be temporarily stored in the cloudlet or other middle layers. The results can be sent to the server on the fly since it is responsible for the multimedia application's timing.

Fig. 3 shows a flow chart of the proposed architecture. As the figure shows, the acquisition devices are connected to a network of second-level cloudlet elements that contribute to computation. Lastly, a final layer consisting of cloud servers

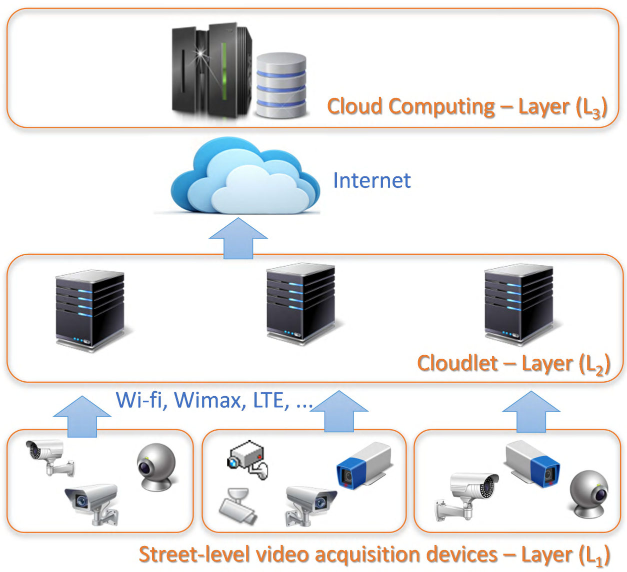

FIGURE 3. General scheme of the infrastructure of the proposed architecture.

and databases accessible via the Internet performs the deep analysis.

Communication technology evolution has been one of the key features that has enabled the smart city paradigm. Currently, several alternatives allow 'things' to interact with each other while ensuring network connectivity [78], [79], [80]. Acquisition sensors can be connected to a local area network (LAN) via wireless technology such as Bluetooth low energy (BLE), WiMAX or Wi-Fi. In this regard, a novel

Wi-Fi technology based on IEEE 802.11ah has been designed for smart city contexts [81]. It has low power consumption and long-range sensors and controllers. In addition, mobile telecommunication technology such as LTE can be also a good choice to connect the devices. In this case, the available bandwidth is typically two orders of magnitude lower than WLAN bandwidth [82].

The cloudlet layer can be on the same WLAN as the acquisition sensors with the objective of reducing latency and jitter. This proposal supports multiple design configurations. In the example described, the cloudlets can be installed on every street, in a group of streets or city district according to the devices' connectivity options and the processing capabilities. Efficiency criteria must be met before deciding to deploy the cloudlet infrastructure. The results generated by all the cloudlets are sent to the cloud server, where they are analyzed.

Based on the above, the available computing infrastructure can be defined as follows:

$$\text{Computing Infrastructure} = \{L_1, L_2, L_3\} \tag{1}$$

The acquisition devices (Layer L_1) can be heterogeneous with different data processing capabilities. This hardware consists of fixed cameras for traffic and security applications or mobile cameras installed on vehicles and/or worn by people.

The second computing level (Layer L_2) introduces flexibility to the model for multimedia application processing. This level performs part of the processing and reduces the amount of data transmitted to the cloud server (Layer L_3). This model is in line with the MCC paradigm, in which the devices can offload some of the processing to the cloud. In this case, the distributed architecture allows the 'things' to outsource their workload to the cloudlets for partial processing and, as a result, reduces the communication costs with the cloud server.

The operation mode that outsources the workloads among the layers is usually based on a client-server methodology [66]. This method defines how to link the things with the cloudlets, and the latter with the cloud server. Client-server communication requires the deployment of specific services to carry out the offloading and perform the interactions. Thus, the network infrastructures must include the application code as a service to be invoked by the architecture's lower layers [83].

The proposed architecture is one of many possible architectures for distributed data processing. However, this approach significantly reduces the need to communicate data to a central element. This feature favors the execution of big data applications in the cloud system, processing the results transmitted by the lower levels and provides a platform that is ideal for handling the increasing amount of acquired data in urban environments.

In addition, this proposal can be considered from a service-provision point of view. In consequence, Figure 4 shows the service architecture of this distributed approach where services are divided between different architecture layers. From this point of view, we can infer three layers in general terms: the sensor/actuator layer, the networking layer and the application layer. Each layer has specific features and provides its own services.

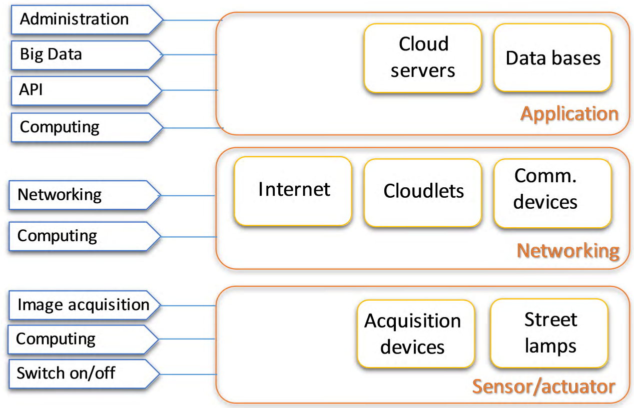

FIGURE 4. Service architecture.

Our proposal's main contribution to this service architecture is that all layers provide computing services, and this helps in developing intensive urban computing of multimedia data applications since the advanced and complex calculations involved can be divided among all infrastructure layers.

This distributed architecture is particularly suitable for intensive data processing because it provides configuration flexibility and a scalable application execution through the computing layers. In contrast, other proposals concentrate the processing power in local computers or in remote systems on the cloud.

The proposal's key feature lies in deploying this multilevel scheme in urban environments to exploit the increasing number of image and video acquisition devices being installed, and where a cloudlet infrastructure can be easily deployed in order to scale advanced smart city applications. This architecture allows a combination of network computing resources: from local processing in acquisition devices (where available) to near and remote cloud computing.

The proposed architecture goes further than the MCC paradigm as the cloudlets are ideally installed near the data acquisition and could configure LANs to provide high communication speeds.

After a general overview of the architecture, we shall outline the main aspects involved in distributed computing for massive multimedia processing.

Firstly, multimedia applications are considered to consist of a list of tasks that are executed sequentially. The general inputs' application is a multimedia flow acquired by video and/or image sensors, and a task's results are the inputs for the next task:

$$\text{Application} \equiv \{t_1, t_2, \ldots, t_n\} \tag{2}$$

These tasks (t_i) can be processed at different processing and platform levels (L_i), for example, on the acquisition device itself, on the cloudlet infrastructure or on the cloud computing server. The intermediate platforms and the cloud infrastructure collect data from various lower level elements. Each layer has a set of computing platforms, which can be heterogeneous and have different processing capacity according to their characteristics. Furthermore, there may be platforms that are specifically capable of performing certain task types (multimedia, cryptographic, etc.). To this end, the intermediate layers, such as cloudlets, can be equipped with specific computing capabilities depending on the type of workload they usually run.

Secondly, the time it takes to compute the application ($\mathrm{Computing\_time_{App}}$) is the time needed to compute each task. The computing time of a certain task (t_i) at a layer (L_j) can be formalized as follows:

$$\mathrm{Computing\_time}\ (t_i)^* \equiv \mathrm{Cmp}_{L_j}(t_i) \quad {}^*\text{in time units} \tag{3}$$

In addition to computational cost, the network communication cost must be also taken into account in the MCC paradigm, i.e., moving the tasks between the platforms as well as the data they need. Therefore, there is a second component in the overall time cost of computing a certain task (t_i) at a layer (L_j):

$$\mathrm{Network\_time}(t_i)^* \equiv \mathrm{Net}_{L_j}(t_i) \quad {}^*\text{in time units} \tag{4}$$

Of course, the cost will be null when the task does not move between layers.

The volume of each task's input data is associated with this network time cost. It is formulated in the following expression:

$$\mathrm{Input\_Data}(t_i)^* \equiv \mathrm{Data}(t_i) \quad {}^*\text{in data volume units} \tag{5}$$

where $\mathrm{Data}(t_1) = $ multimedia input.

Usually, for video/image analysis algorithms, the required data tends to decrease as the algorithm progresses. That is,

$$\mathrm{Data}(t_{i+1}) \leq \mathrm{Data}(t_i) \tag{6}$$

This is one of the middle layers' main advantages. By computing part of the workload in the cloudlet, communication costs are dramatically reduced. Thus,

$$\mathrm{Data}(L_{i+1}) \leq \mathrm{Data}(L_i) \tag{7}$$

where $\mathrm{Data}(L_i)$ is the data input arriving to Layer i.

From the previous formula, the overall time cost of the Application can be expressed as follows:

$$\mathrm{Computing\_time}^*_{App} = \sum_i [\mathrm{Cmp}_{L_j}(t_i) + \mathrm{Net}_{L_j}(t_i)] \quad {}^*\text{in time units} \tag{8}$$

where j is the layer where each t_i is processed.

According to the scheme in Fig. 3, the data flow and platforms' connectivity define a tree of connected platforms in which a set of acquisition devices is connected to a cloudlet platform and a set of cloudlets is connected to the cloud computing infrastructure. In this architecture, the tasks may move depending on the offloading configuration. The possible options enable flexible multimedia application implementation in order to optimize any of the system parameters, such as minimizing the response time, reducing the data flow through the communication network, minimizing the acquisition devices' energy consumption and monetary costs associated with using cloud services, increasing the cloud system's processing time, etc. Accordingly, the proposed distributed architecture means that numerous task execution configurations can be designed, depending on the application type, execution restrictions or operating conditions, taking into account the aspects above. In any case, a suitable scheduling function is needed to decide where to offload each of the application tasks [64], [77]. However, this architecture is intended to move the processing flow from the acquisition devices to the cloud infrastructure. This is how advanced smart city applications' increasing requirements could be met, especially those associated with analyzing data acquired by multimedia sensors.

Although possible, a data stream back from the cloud to the cloudlet or the sensors is not common. Hence, the expected ideal configurations are oriented towards incrementally distributing the application processing at each level with the objective of reducing the communication needs and moving the specialist multimedia analysis tasks to the cloud processing infrastructure. Fig. 5 illustrates some possible configurations depending on the characteristics of the infrastructure deployed.

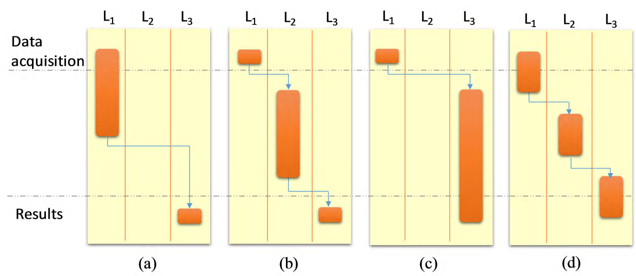

FIGURE 5. Distributed processing configurations. (a) Device processing. (b) Cloudlet processing. (c) Cloud processing. (d) Shared processing.

In Fig. 5, the (a) configuration corresponds to an on-device processing scheme. It requires powerful acquisition devices to perform most of the processing and transmit the results to the cloud system. This scenario could occur in applications where data are collected by smart mobile devices with enough computing power. The (c) configuration corresponds to a scenario in which the acquisition devices lack processing power, are limited to data acquisition and transmit the data to a central element for processing. The configurations (b) and (d) distribute the tasks among the three levels to different degrees.

IV. CASE STUDY

A. APPLICATION DEFINITION

In order to illustrate how the proposed distributed architecture functions, this section describes a case study in which the proposed architecture is used to process the multimedia data acquired by cameras deployed in a city context similar to the

TABLE 2. Comparison of smart lighting implementation methods.

Method	Advantages	Disadvantages
Presence detection based on wireless sensors in street lamps [86]	High precision.	Having to install presence sensors on each lamp.
Presence detection based on wireless sensors deployed along the street [84, 85]	High precision.	Having to install presence sensors on the street.
Presence detection based on Wi-Fi connection [90, 91]	No additional installation is needed in Wi-Fi areas.	Low precision. The presence is related with the position of the antennas.
Adaptive smart lighting system [92, 93]	Personalised lighting control	Only available for indoor environments.
Presence detection based on video analysis (this proposal)	Video sensors can cover a wide area and obtain other interesting information about the city.	High volume of multimedia data needs to be processed

one shown in Fig. 1. The case study is a simplified version of a smart lighting application for smart cities. This application aims to optimize street lamps' power consumption and provide a better service to citizens. In this application, the urban street lamps light up when humans are in the street.

Smart lighting applications are a recent trend that has arisen as a result of the new global awareness of the dangers of climate change. The final objective is to use energy in the most rational manner possible by providing night light to the streets only when necessary. This would make cities more sustainable and comfortable since it guarantees optimal control of light pollution. Currently, as a first step, many cities are replacing old sodium street light bulbs with LEDs, which are much longer-lasting and consume less energy. The next step is turning the light on only when needed.

The night luminosity can be graduated in three levels with different power consumptions: {*no_luminosity* (0%), *low_luminosity* (20%), *normal_luminosity* (100%)}. During the day, the luminosity level is *no_luminosity*, at night, when there are pedestrians on the street, it is set to *normal_luminosity* and, at night, when there are no pedestrians on the street, it is *low_luminosity*. This level means that the city streets are not in complete darkness, but the street lamps only consume 20% of energy.

The existing proposals for smart lighting are mainly based on the use of presence detectors installed along the street [84], [85] or on each street lamp [86]. These devices basically consist of an infrared sensor that detects pedestrians [87], [88]. This type of technology is used by the most common commercial solutions around the world [89].

Other recent methods use Wi-Fi connections to detect human presence [90], [91]. They are based on smartphones and other devices with Wi-Fi connection having a high penetration rate among the population. Finally, novel proposals aim to learn from the individual behaviors and context conditions to adapt to the environment luminosity accordingly. At present, these proposals are being developed for indoor scenarios [92], [93].

The sensor devices and streets lamps need to be connected in order to send the presence detection to the street lamps or from one street lamp to another. In all cases, these proposals involve adapting the existing lamps with new hardware and communication capabilities.

In this case study, our proposal consists of using standard unmodified street lamps and video cameras deployed for traffic and surveillance, to detect human presence. All of the 'things' involved can communicate through new machine-to-machine wireless communication technologies at a low cost [94].

This case study does not specifically focus on solving this issue. However, using video cameras to provide value-added services to the user in city environments offers several advantages when compared to other methods. Table 2 shows a simple comparison in regard to some aspects related to precision and installation.

The proposed architecture makes handling the multimedia data possible and can enrich the applications based on the data acquired from the city's video cameras. As mentioned, in most cases, cameras are already deployed for traffic and surveillance purposes.

We have described a simplified version of the application to ensure that we do not discuss details that are not relevant for the purpose of this paper. It can be decomposed on some common tasks related to processing multimedia data from digital signals in a possible smart city context. Based on well-known methods, the smart lighting application taskset used in this study is shown in Figure 6.

This case study will focus on analyzing the feasibility of the distributed architecture in handling intensive data processing from heterogeneous acquisition devices. We analyses the proposal's ability to perform incremental processing techniques that reduce the data volume as well as the communication needs.

Other performance aspects such as the processing cost and power consumption are not critical since the distributed architecture provides sufficient computing resources at cloudlet and cloud levels to meet the computation needs.

A brief description of the tasks is as follows:

t_1 – *capture:* data from streaming video cameras are taken. The amount of data depends on the frames per second (fps) and frame resolution in pixels. Typical values for video surveillance cameras are: 30 fps and 720 × 480 pixels [95]. These data are usually compressed, for example, with MPEG-4 video compression standard, and between 1–2 Megabits per second (Mbps) are produced. In a context with thousands of cameras (e.g. Vancouver city center),

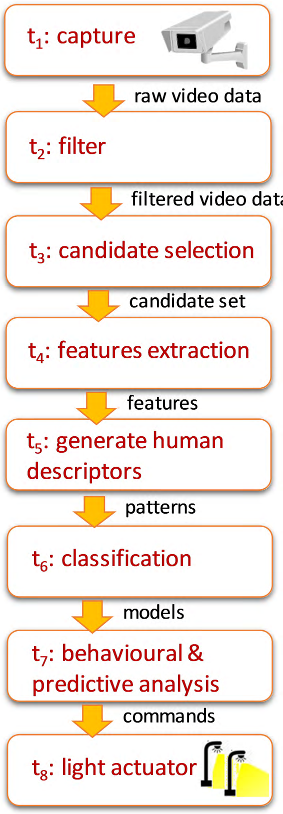

FIGURE 6. **Smart lighting application decomposition.**

communication networks of several Gbps bandwidth are required to transmit these data to a central server to be processed.

t_2 – *filter:* a filtering step is performed to improve the contrast and luminosity in shaded areas or evening environments. The data flow resulting from this stage is equal to the previous one.

t_3 – *candidate selection:* some areas are selected in the image, which are candidates for containing a human figure. This is a simple procedure carried out by movement detection techniques, included in many modern cameras. As a result, a set of image areas that could contain human forms is obtained. These areas can represent up to 10% of the image in those cases where people are walking along the streets. For the purpose of this case study, we consider pedestrian presence at all times of the day.

t_4 – *features extraction:* characteristics of each area of the candidate image are extracted. The features can be computed from low-level information: such as edge, texture or color. At this point, visual information does not need to be transmitted. Only feature sets are required. The features are composed of a data structure for each area depicting the captured form. They may be formed by binary contours, the nearest distances to the edge pixels, histograms of oriented gradients, etc. These data structures are much smaller than the area of the image they represent. It is set to 5KB size for the candidate area's features.

t_5 – *generation of human descriptors:* the descriptors are generated by combining the features. We consider that this information is half the size of the previous input.

t_6 – *classification:* classifiers and learning algorithms. Once the human descriptors are extracted from the candidate areas, this step classifies them as human or non-human. At this point, we know if the image contains one or more humans. The data produced in this stage correspond to a bit vector related to the presence or absence of humans in the image as well as attributes associated with them, such as size, location, etc.

t_7 - *behavioural & predictive analysis:* The presence or absence of a human in the images provides enough information to increase or decrease an area's luminosity. However, for comfort and convenience, turning street lamps on and off as someone walks past them is not ideal. As such, street lamps should be turned on and kept on to ensure that the person's path is lit up. For this purpose, pedestrian behavioural analysis techniques can be applied to predict the person's movements and anticipate lighting the lamps. This stage requires deeper analysis to be run in the cloud level, so only the information extracted to predict pedestrian behavior is transmitted to the architecture's upper level. As a result, selective lamp lighting proposals are produced.

t_8 – *light actuator:* the light level to be provided is indicated to each lamp or set of lamps.

Methods for conducting the previous tasks are not part of this research. Well-known methods from state-of-the-art developments can be used in each case.

In this example, the distribution of tasks among the processing levels is dynamic and depends on the number of cameras there are for each cloudlet element and the cloudlet availability. Additionally, a centralized cloud infrastructure to address processing is available.

For the proposed case study, both the cloudlet system and the cloud infrastructure can run a wide range of tasks. A typical configuration example could be the following:

- Tasks run by the camera (L_1): *usually*{1..3}
- Tasks run by the cloudlet (L_2): *usually*{4..5} + *unusually*{2..3}
- Tasks run by the cloud computing infrastructure (L_3): *usually*{6..8} + *unusually*{2..5}

The *usually* tasks are those that should be run by each part of the architecture. The *unusually* tasks correspond to tasks that could be performed if the lower level were to find it difficult or is not able to. For example, if higher performance cameras are combined with others already deployed without processing capacity, the tasks assigned to the lower performance cameras should be run by the cloudlet and, if it is not capable of doing it either, they should be run by the cloud infrastructure. Fig. 7 illustrates the distribution of tasks for the case study based on the configuration model shown in Fig. 5.c.

In this regard, the proposed architecture presents a flexible model that combines a heterogeneous acquisition device infrastructure. If the technology's deployment scheme

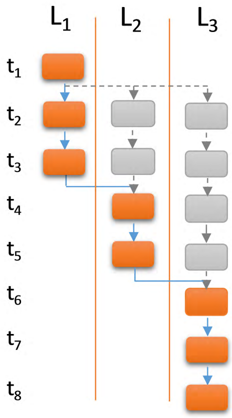

FIGURE 7. Task configuration.

FIGURE 8. Urban camera distribution.

allows it, the proposed model enables big multimedia data processing at various levels, because the generated data volume is reduced as the processing progresses, and the challenges associated with its massive processing are reduced.

This approach drastically reduces the volume of data flowing through the communication network, since each task produces results that are smaller than their inputs. With the proposed task distribution, the amount of information transmitted to the cloudlet element reduces the flow of data captured by each camera, and the processing sent to the cloud infrastructure by each cloudlet, in turn, is drastically lower. With this scheme, the cloud layer can absorb massive multimedia processing applications and can focus on performing intelligence or big data tasks on the information gathered by a multitude of sensors, rather than be distracted by low value-added processes.

This architecture is highly scalable, as it can extend the area of action by simply incorporating new acquisition and cloudlet elements where needed without altering the overall scheme of operation.

B. SIMULATION RESULTS

This simulation's planned configuration consists of a set of cameras installed at various crossroads in the city. In addition, each city district has a cloudlet that provides a service to a set of six to 20 cameras and the whole city can access a cloud processing service. The cloud infrastructure size depends on the area being monitored, the services that can be deployed in the cloudlets and the cameras themselves.

Fig. 8 shows an example of this infrastructure for a city district that consists of 15 blocks, 12 streets and a square occupying a total area of 8 ha. For this example, we have 11 cameras with different characteristics and processing capacities. These cameras may have been initially deployed for traffic management. They monitor a total street length of around 10 km. Fig. 8 shows a possible location of the installed cameras.

The 11 cameras shown in the previous figure are of three different types, showing how the proposed architecture can manage the heterogeneity of "things":: *yellow* cameras have no processing power and only capture data $\{t_1\}$; *orange* cameras, with reduced processing capacity, capture the data and apply a filter to them $\{t_1, t_2\}$; *red* cameras are advanced and can identify image areas with motion $\{t_1, t_2, t_3\}$.

In this example, street cameras have been simulated through public webcams. As such, video streaming was taken from an EarthCam public webcam (https://www.earthcam.com/) which links video streams from thousands of web cameras around the world. We used the "Abbey Road Crossing" webcam (https://www.earthcam.com/world/england/london/abbeyroad/?cam=abbeyroad_uk) (Figure 9a) installed in London (UK) because it is directed at a known place, where some street lamps are easily identified (Figure 9b). Of course, this is a very busy area due to its popularity. As such, the people flow is higher than in other standard areas of the city. This camera has an average dataflow of 2 Mbps. This experiment's advanced cameras (orange and red) were simulated by a Raspberry Pi 3 B + device. The cloudlet and cloud server were simulated by a laptop and a workstation respectively. Table 3 shows the main features of the infrastructure used to simulate the architecture.

This experiment's implementation uses public software to take the frames from the video streaming source [96] and select human figures (task t_3) [97].

Human presence was detected every second from the video streaming source provided by the webcam (Figure 9c). In this way, data heterogeneity in this application is reduced. Analyzing the video streaming from webcams becomes an image analysis process. This frequency is enough to maintain the quality, taking into account the normal speed pedestrians walk along the street (1.5 m/s). The frames only need to be stored while they are processed and can be subsequently deleted.

Simulated smart cameras' performance (yellow and orange) enables them to perform their tasks (frame capturing, filtering and human candidate selection-motion detection) in less than 250 ms in the worst case. The cloudlet performs the feature extraction from each frame in 300 ms and the cloud server detects human presence in 350 ms.

In this experiment, the cameras and cloudlets were connected to the same WLAN by means of our Wi-Fi connection (IEEE 802.11ac), while the cloud server was deployed

(a) (b) (c)

FIGURE 9. "Abbey Road Crossing" webcam. (a) Picture taken 10th December 2018. (b) Number of visible street lamps (including some Belisha beacons). (c) Pedestrian identification.

TABLE 3. Infrastructure of the distributed architecture.

Feature	Camera	Cloudlet	Cloud Server
Device type	Raspberry Pi 3	Laptop	Workstation
Operating system	Raspbian (Nov.'18)[1]	Ubuntu 16.04.5 LTS (desktop)[2]	Ubuntu 16.04.5 LTS (server)[2]
CPU	ARMv8 Cortex-A53	i7-3632	i7-4790
GPU	Broadcom VideoCore IV	Intel HD Graphics 4000	NVIDIA GTX 960
RAM	1 GB	6 GB	8 GB

[1] Downloaded from https://www.raspberrypi.org/downloads/raspbian/
[2]Downloaded from http://releases.ubuntu.com/16.04/

outside this LAN and was accessible via the Internet. An urban environment like the one depicted in Figure 8 should use the new IEEE 802.11ah protocol in order to minimize the communication infrastructure and Wi-Fi hotspots.

The method to perform the code offloading in this experiment is via the client-server communication mode. As such, the acquisition devices are thin clients, the cloudlet is configured both as client and server and the cloud machine always works as server.

On the other hand, perceived delay and jitter in our experiments have remained low (<25 ms). These results are in line with other studies on IEEE 802.11ac, which prove that a high data rate is compatible with a low mean delay and mean jitter [98]. In addition, these features are not critical for this kind of application where the response time in accessing the server (cloudlet or cloud) does not affect the outcomes achieved.

Following these results, the periodicity established (1 frame/second) is enough to perform all the calculations in the frame during this time. Nevertheless, a higher period can be established (1 frame/10 seconds) once the motion has been detected in the video stream since the tasks involved in human detection are more computationally intensive. In this way, the infrastructure described in Table 3 could handle the set of cameras deployed in the urban scenario shown in Figure 8.

During the experiment, we analyzed live webcam video streaming for a whole night (15 h 32 min). Human presence was detected in 8,845 frames, i.e., for 2 h, 27 min and 25 sec, representing 15.81% of the total time.

In this example, the cloudlet infrastructure crucially collaborates in processing and reducing the storage and communication requirements. Thanks to this multiple-layer architecture, the data flow received by the cloud infrastructure are too small for the set of cameras, meaning that the example can be easily extrapolated to all areas of a medium-sized city.

Taking this data as a reference for the 11 cameras used in the simulation, the amount of data generated as a result of each task and the data input arriving towards the district cloudlet and the cloud server is shown in Table 4.

For video streaming scenarios, previous studies on the IEEE 802.11ah protocol show that there is an inverse proportion between the number of acquisition devices and the maximum attainable data rate [99]. Therefore, a data-flow of 2 Mbps for each camera can easily saturate the network bandwidth in a city environment. In this situation, the smart cameras and the cloudlet layer play a vital role by reducing communication needs. As can be observed in Table 3, the data communicated to the cloud can be reduced from 0.59 TB to 475 MB per day.

With regard to the results of the smart lighting application for the study area, considering the defined luminosity levels and an average number of pedestrians at night-time of around 15% of the time, the system can produce energy savings of 68%.

Based on the simulation, the area should be provided with artificial light for a period of 15 hours and 32 min. It is important to take into account that some neighborhoods are busier at night (city center, tourist areas, etc.) than others (residential, commercial, etc.). In addition, fewer artificial light hours would be needed during the summer. Considering the annual average, we use12 h of night/day as a reference.

TABLE 4. Amount of data generated. (Data per second/data per night).

Application Stage	Data generated by application tasks Data(t_{i+1})	Data input arriving to district cloudlet Data(L_2)	Data input arriving to cloud computing infrastructure Data(L_3)
t_1 – capture	22 MB*1 / 0.59 TB*2		
t_2 – filter	22 MB / 0.59 TB	6 MB*3 / 327*4 GB	
t_3 – candidate selection	3.47 MB*5 / 189.9*6 GB	1.57 MB*7 / 85*8 GB	
t_4 – features extraction	-- / 475*9 MB		-- / 475*9 MB

*1: 2 Mbps x 11 cameras; *2: 2 Mbps x 11 cameras x 15 h 32 min
*3: 2 Mbps x 3 yellow cameras; *4: 2 Mbps x 3 yellow cameras x 15 h 32 min
*5: 2 Mbps x 11 cameras x 15.81% avg; *6: 2 Mbps x 11 cameras x 15h 32 min x 15.81% avg.
*7: 2 Mbps x (3 yellow cameras + 2 orange cameras) * 15.81% avg; *8: 2 Mbps x (3 yellow cameras + 2 orange cameras) * 15.81% avg x 15 h 32 min
*9: 8,845 frames * 5KB/frame * 11 cameras;

We identified 10 street lamps (Figure 9b) in the analyzed scene shown in Figure 9. We suppose a standard power consumption of 250 W for each one and a combined output of 2.5 kWh. For a standard area, such as that shown in Figure 8, we suppose that 300 street lamps have a combined output of 75 kWh. In these small-scale cases, the smart lighting system can achieve savings of 1.7 kWh and 51 kWh respectively every day.

V. CONCLUSIONS

Designing modern applications for cyber-physical systems and IoT environments aims for architectures able to address the high computation and communication requirements of multimedia data collected. We have presented a review of the proposals for video and image processing in distributed environments in order to identify the main challenges and current research lines.

In this work, we have described a distributed architecture that can perform data-intensive application processing. The proposal overcomes the main computation issues found and can simultaneously process multimedia data collected from many images and video sources. The combination of the network's computing resources, from locally processing the things to near and remote cloud computing, enables the deployment of IoT environments where data-intensive processing is needed.

This distributed multilevel scheme is a novel MCC paradigm approach. The architecture consists of three computing layers that execute complex multimedia analysis tasks: the acquisition devices, the middle computing cloudlets and the cloud computing infrastructure. The first layer takes advantage of the new data acquisition devices' increasing computing capabilities. For example, smartphones and other mobile devices could be used in this scheme. The second layer explores the idea of introducing a computing infrastructure close to where the data is taken in order to receive results from a few acquisition devices. This infrastructure is known as a cloudlet. The third layer is the cloud computing server where all the previous partial results are transmitted and the final analysis is conducted. The specifications provided are useful for designing optimal distributed configurations.

With this architecture, intensive, workload-dependent data applications are distributed through three computing layers and the communication needs are increasingly reduced at each layer. This means that raw data do not need to be transmitted to the central cloud server to be processed. To this end, IoT applications take greater advantage of the cloud infrastructure's potential because they can focus on high-level services, such as analyzing the big data associated with pre-processed results received.

The case study shows how an advanced application for smart city environments can handle multimedia data from many deployed acquisition devices using the proposed architecture. The example describes the distributed processing through the architecture layers and the data reduction that takes place. Implementing the smart lighting application using multimedia data from video cameras was possible using the proposed architecture. As a result, the proposal adds versatility to the IoT deployments to perform value-added applications and progress in developing the smart city concept. Furthermore, it provides a scalable solution for deploying such applications in urban environments.

For future research, we plan to study other distributed multimedia applications, particularly big multimedia analysis applications. In addition, we aim to extend the architecture to n computing layers in order to increase flexibility. Other aspects of the proposals need further research, such as the cloudlets' locations. Furthermore, we plan to build a prototype of the proposed smart lighting system and put it into practice in a real city area.

REFERENCES

[1] A. Amairah, B. N. Al-Tamimi, M. Anbar, and K. Aloufi, "Cloud computing and Internet of Things integration systems: A review," in *Recent Trends in Data Science and Soft Computing*, vol. 843. Cham, Switzerland: Springer, 2019, pp. 406–414. doi: 10.1007/978-3-319-99007-1_39.

[2] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: Open issues, challenges and future perspectives," *J. Cloud Comput.*, vol. 6, p. 30. Dec. 2017. doi: 10.1186/s13677-017-0097-9.

[3] T.-H. Kim, C. Ramos, and S. Mohammed, "Smart city and IoT," *Future Gener. Comput. Syst.*, vol. 76, pp. 159–162, Nov. 2017. doi: 10.1016/j.future.2017.03.034.
[4] A. H. Alavi, P. Jiao, W. G. Buttlar, and N. Lajnef, "Internet of Things-enabled smart cities: State-of-the-art and future trends," *Measurement*, vol. 129, pp. 589–606, Dec. 2018. doi: 10.1016/j.measurement.2018.07.067.
[5] M. D. Lytras and A. Visvizi, "Who uses smart city services and what to make of it: Toward interdisciplinary smart cities research," *Sustainability*, vol. 10, no. 6, pp. 1–16, 2018. doi: 10.3390/su10061998.
[6] S. K. A. Hossain, M. A. Rahman, and M. A. Hossain, "Edge computing framework for enabling situation awareness in IoT based smart city," *J. Parallel Distrib. Comput.*, vol. 122, pp. 226–237, Dec. 2018. doi: 10.1016/j.jpdc.2018.08.009.
[7] H. Mora, F. J. M. Gimeno, M. T. Signes-Pont, and B. Volckaert, "Multi-layer architecture model for mobile cloud computing paradigm," *Complexity*, vol. 2019, Feb. 2019, Art. no. 3951495. doi: 10.1155/2019/3951495.
[8] C. Lim, K. J. Kim, and P. P. Maglio, "Smart cities with big data: Reference models, challenges, and considerations," *Cities*, vol. 82, pp. 86–99, Dec. 2018. doi: 10.1016/j.cities.2018.04.011.
[9] R. Pérez-Delhoyo, C. García-Mayor, H. Mora, V. Gilart-Iglesias, and M. D. Andújar-Montoya, "Improving urban accessibility: A methodology for urban dynamics analysis in smart, sustainable and inclusive cities," *Int. J. Sustain. Develop. Planning*, vol. 12, no. 3, pp. 357–367, 2017. doi: 10.2495/SDP-V12-N3-357-367.
[10] U. Ramachandran *et al.*, "Large-scale situation awareness with camera networks and multimodal sensing," *Proc. IEEE*, vol. 100, no. 4, pp. 878–892, Apr. 2012. doi: 10.1109/JPROC.2011.2182093.
[11] C. Diwaker, P. Tomar, and A. Sharma, "Future aspects and challenges of the Internet of Things for the smart generation," in *Proc. Int. Conf. Commun. Cyber Phys. Eng.*, in Lecture Notes in Electrical Engineering, vol. 500. Singapore: Springer, 2019, pp. 599–606. doi: 10.1007/978-981-13-0212-1_61.
[12] L. Farhan and R. Kharel, "Internet of Things: Vision, future directions and opportunities," in *Modern Sensing Technologies* (Smart Sensors, Measurement and Instrumentation), vol. 29. Cham, Switzerland: Springer, 2019, pp. 331–347. doi: 10.1007/978-3-319-99540-3_17.
[13] J. Gao, Y. Yang, P. Lin, and D. S. Park, "Computer vision in healthcare applications," *J. Healthcare Eng.*, vol. 2018, Mar. 2018: Art. no. 5157020. doi: 10.1155/2018/5157020.
[14] R. W. Liu, Q. Ma, S. C. H. Yu, K. T. Chui, and N. Xiong, "Variational regularized tree-structured wavelet sparsity for CS-SENSE parallel imaging," *IEEE Access*, vol. 6, pp. 61050–61064, 2018. doi: 10.1109/ACCESS.2018.2874382.
[15] M. Zreik *et al.*, "Deep learning analysis of the myocardium in coronary CT angiography for identification of patients with functionally significant coronary artery stenosis," *Med. Image Anal.*, vol. 44, pp. 72–85, Feb. 2018. doi: 10.1016/j.media.2017.11.008.
[16] W. Sun, S. Zhu, X. Ju, and D. Wang, "Deep learning based pedestrian detection," in *Proc. Chin. Control Decis. Conf. (CCDC)*, 2018, pp. 1007–1011. doi: 10.1109/CCDC.2018.8407277.
[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893. doi: 10.1109/CVPR.2005.177.
[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. doi: 10.1109/TPAMI.2009.167.
[19] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia Comput. Sci.*, vol. 132, pp. 377–384, Apr. 2018. doi: 10.1016/j.procs.2018.05.198.
[20] L. Wang, Y. Xu, J. Cheng, H. Xia, J. Yin, and J. Wu, "Human action recognition by learning spatio-temporal features with deep neural networks," *IEEE Access*, vol. 6, pp. 17913–17922, 2018. doi: 10.1109/ACCESS.2018.2817253.
[21] B. Yang, J. Cao, R. Ni, and Y. Zhang, "Facial expression recognition using weighted mixture deep neural network based on double-channel facial images," *IEEE Access*, vol. 6, pp. 4630–4640, 2018. doi: 10.1109/ACCESS.2017.2784096.
[22] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39 no. 4, pp. 640–651, Apr. 2017. doi: 10.1109/TPAMI.2016.2572683.
[23] X. Liu, H. A. Ounifi, A. Gherbi, Y. Lemieux, and W. Li, "A hybrid GPU-FPGA-based computing platform for machine learning," *Procedia Comput. Sci.*, vol. 141, pp. 104–111, Nov. 2018. doi: 10.1016/j.procs.2018.10.155.
[24] P. Wlodarczak, J. Soar, and M. Ally, "Multimedia data mining using deep learning," in *Proc. 5th Int. Conf. Digit. Inf. Process. Commun. (ICDIPC)*, 2015, pp. 190–196. doi: 10.1109/ICDIPC.2015.7323027.
[25] P. Dollár, P. Wojek, C. Schiele, and B. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 304–311. doi: 10.1109/CVPR.2009.5206631.
[26] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li, "Robust multi-resolution pedestrian detection in traffic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3033–3040. doi: 10.1109/CVPR.2013.390.
[27] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 899–906. doi: 10.1109/CVPR.2014.120.
[28] J. Marin, D. Vázquez, A. M. López, J. Amores, and B. Leibe, "Random forests of local experts for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2592–2599. doi: 10.1109/ICCV.2013.322.
[29] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 973–986, Apr. 2018. doi: 10.1109/TPAMI.2017.2700460.
[30] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Strengthening the effectiveness of pedestrian detection with spatially pooled features," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2014, pp. 546–561. doi: 10.1007/978-3-319-10593-2_36.
[31] K. K. Mohbey, "An efficient framework for smart city using big data technologies and Internet of Things," in *Progress in Advanced Computing and Intelligent Engineering* (Advances in Intelligent Systems and Computing), vol. 714. Singapore: Springer, 2019, pp. 319–328. doi: 10.1007/978-981-13-0224-4_29.
[32] M. D. Lytras, V. Raghavan, and E. Damiani, "Big data and data analytics research: From metaphors to value space for collective wisdom in human decision making and smart machines," *Int. J. Semantic Web Inf. Syst.*, vol. 13, no. 1, pp. 1–10, 2017. doi: 10.4018/IJSWIS.2017010101.
[33] L. Zhang, Y. Xia, K. Mao, H. Ma, and Z. Shan, "An effective video summarization framework toward handheld devices," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1309–1316, Feb. 2015. doi: 10.1109/TIE.2014.2336639.
[34] Y. Xia, J. Chen, X. Lu, C. Wang, and C. Xu, "Big traffic data processing framework for intelligent monitoring and recording systems," *Neurocomputing*, vol. 181, pp. 139–146, Mar. 2016. doi: 10.1016/j.neucom.2015.07.140.
[35] L.-M. Ang and K. P. Seng, "Big sensor data applications in urban environments," *Big Data Res.*, vol. 4, pp. 1–12, Jun. 2016. doi: 10.1016/j.bdr.2015.12.003.
[36] D. Llamocca and M. Pattichis, "A dynamically reconfigurable pixel processor system based on power/energy-performance-accuracy optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 488–502, Mar. 2013. doi: 10.1109/TCSVT.2012.2210664.
[37] J. H. Luo and C. H. Lin, "Pure FPGA implementation of an hog based real-time pedestrian detection system," *Sensors*, vol. 18, no. 4, p. 1174, 2018. doi: 10.3390/s18041174.
[38] Q. Shi, D. Zhang, S. Cheng, H. Luo, and N. Yuan, "An ultrafast digital signal processor for millimeter wave real-time imaging radar," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Sep. 2015, pp. 1–4. doi: 10.1109/ICSPCC.2015.7338819.
[39] Y. Liu, H. Yan, X. He, and Z. Wei, "A digital image processing system based on DSP," in *Proc. Int. Conf. Inf. Eng. Comput. Sci.*, 2010, pp. 1–4. doi: 10.1109/ICIECS.2010.5678329.
[40] Z. Zhan, W. Hao, Y. Tian, D. Yao, and X. Wang, "A design of versatile image processing platform based on the dual multi-core DSP and FPGA," in *Proc. Int. Symp. Comput. Intell. Design*, 2012, pp. 236–239. doi: 10.1109/ISCID.2012.210.
[41] M. D. Saha, M. K. Darji, N. Patel, and D. Thakore, "Implementation of image enhancement algorithms and recursive ray tracing using CUDA," *Procedia Comput. Sci.*, vol. 79, pp. 516–524, Feb. 2016. doi: 10.1016/j.procs.2016.03.066.
[42] A. HajiRassouliha, A. J. Taberner, M. P. Nash, and P. M. F. Nielsen, "Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms," *Signal Process., Image Commun.*, vol. 68, pp. 101–119, Oct. 2018. doi: 10.1016/j.image.2018.07.007.

[43] X. Wang, Q. Wu, X. Lin, Z. Zhuo, and L. Huang, "Pedestrian identification based on fusion of multiple features and multiple classifiers," *Neurocomputing*, vol. 188, pp. 151–159, May 2016. doi: 10.1016/j.neucom.2014.10.114.
[44] C. González, S. Sánchez, A. Paz, J. Resano, D. Mozos, and A. Plaza, "Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing," *Integr., VLSI J.*, vol. 46, no. 2, pp. 89–103, Mar. 2013. doi: 10.1016/j.vlsi.2012.04.002.
[45] S. S. de Almeida, A. C. de Nazaré, Jr., A. de Albuquerque Araúujo, G. Cámara-Chávez, and D. Menotti, "Speeding up a video summarization approach using GPUs and multicore CPUs," *Procedia Comput. Sci.*, vol. 29, pp. 159–171, Jun. 2014. doi: 10.1016/j.procs.2014.05.015.
[46] H. Atasoy, E. Yildirim, S. Yildirim, and Y. Kutlu, "A real-time parallel image processing approach on regular PCs with multi-core CPUs," *Elektronika ir Elektrotechnika*, vol. 23, no. 6, pp. 64–71, 2017. doi: 10.5755/j01.eie.23.6.19696.
[47] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar, "Multimedia big data analytics: A survey," *ACM Comput. Surv.*, vol. 51, no. 1, 2018, Art. no. 10. doi: 10.1145/3150226.
[48] S. M. Banaei and H. K. Moghaddam, "Hadoop and its role in modern image processing," *Open J. Mar. Sci.*, vol. 4, pp. 239–245, Oct. 2014. doi: 10.4236/ojms.2014.44022.
[49] J. S. Hare, S. Samangooei, and P. H. Lewis, "Practical scalable image analysis and indexing using Hadoop," *Multimedia Tools Appl.*, vol. 71, no. 3, pp. 1215–1248. 2014. doi: 10.1007/s11042-012-1256-0.
[50] T. Fan, "Research and implementation of user clustering based on MapReduce in multimedia big data," *Multimedia Tools Appl.*, vol. 77, pp. 10017–10031, Apr. 2018. doi: 10.1007/s11042-017-4825-4.
[51] S. Guo, Y. Zhang, Q. Wu, L. Niu, W. Zhang, and S. Li, "The performance evaluation of a distributed image classification pipeline based on Hadoop and MapReduce with initial application to medical images," *J. Med. Imag. Health Inform.*, vol. 8, no. 1, pp. 78–83, 2018. doi: 10.1166/jmihi.2018.2236.
[52] M. H. Almeer, "Cloud Hadoop map reduce for remote sensing image analysis," *J. Emerg. Trends Comput. Inf. Sci.*, vol. 3, no. 4, pp. 637–644, 2012.
[53] S. Chen *et al.*, "Galaxy + Hadoop: Toward a collaborative and scalable image processing toolbox in cloud," in *Service-Oriented Computing—ICSOC 2013 Workshops* (Lecture Notes in Computer Science), vol. 8377. Cham, Switzerland: Springer, 2014, pp. 339–351. doi: 10.1007/978-3-319-06859-6_30.
[54] R. Alonso-Calvo, J. Crespo, M. Garc'ia-Remesal, A. Anguita, and V. Maojo, "On distributing load in cloud computing: A real application for very-large image datasets," *Procedia Comput. Sci.*, vol. 1, no. 1, pp. 2669–2677, 2010. doi: 10.1016/j.procs.2010.04.300.
[55] F.-C. Lin, L.-K. Chung, W.-Y. Ku, L.-R. Chu, and T.-Y. Chou, "The framework of cloud computing platform for massive remote sensing images," in *Proc. IEEE 27th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2013, pp. 621–628. doi: 10.1109/AINA.2013.94.
[56] D. Gao, D. Liu, Y. Feng, Q. An, and F. Yu, "A robust image transmission scheme for wireless channels based on compressive sensing," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence* (Lecture Notes in Computer Science), vol. 6216. Berlin, Germany: Springer, 2010, pp. 334–341. doi: 10.1007/978-3-642-14932-0_42.
[57] J. M. Pascual, H. M. Mora, A. F. Guilló, and J. A. López, "Adjustable compression method for still JPEG images," *Signal Process., Image Commun.*, vol. 32, pp. 16–32, Mar. 2015. doi: 10.1016/j.image.2015.01.004.
[58] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 870–883, Jun. 2013.
[59] E. Cerqueira, E. Lee, J.-T. Weng, J.-H. Lim, J. Joy, and M. Gerla, "Recent advances and challenges in human-centric multimedia mobile cloud computing," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, 2014, pp. 242–246.
[60] T.-D. Nguyen, M. Van Nguyen, and E.-N. Huh, "Service image placement for thin client in mobile cloud computing," in *Proc. IEEE Int. Conf. Cloud Comput.*, Jun. 2012, pp. 416–422.
[61] K. Khorramnejad, L. Ferdouse, L. Guan, and A. Anpalagan, "Performance of integrated workload scheduling and pre-fetching in multimedia mobile cloud computing," *J. Cloud Comput.*, vol. 7, p. 13, Dec. 2018.
[62] G. Schembra and G. Incarbone, "A business model for multimedia streaming in mobile clouds," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 3, pp. 376–389. Sep. 2014.
[63] S. Sankari, K. Ashwini, and P. Varalakshmi, "Mobile streaming for QoS approach in media cloud," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, 2015, pp. 403–407.
[64] H. Mora, D. Gil, J. F. C. López, and M. T. S. Pont, "Flexible framework for real-time embedded systems based on mobile cloud computing paradigm," *Mobile Inf. Syst.*, vol. 2015, Jun. 2015, Art. no. 652462.
[65] H.-S. Lee and J.-W. Lee, "Task offloading in heterogeneous mobile cloud computing: Modeling, analysis, and cloudlet deployment," *IEEE Access*, vol. 6, pp. 14908–14925, 2018. doi: 10.1109/ACCESS.2018.2812144.
[66] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, pp. 84–106, Jan. 2013.
[67] B. Li, Y. Pei, H. Wu, and B. Shen, "Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds," *J. Supercomput.*, vol. 71, no. 8, pp. 3009–3036, 2015.
[68] U. Shaukat, E. Ahmed, Z. Anwar, and F. Xia, "Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges," *J. Netw. Comput. Appl.*, vol. 62, pp. 18–40, Feb. 2016.
[69] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in wireless metropolitan area networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Oct. 2015, pp. 570–578.
[70] H. Raei and N. Yazdani, "Performability analysis of cloudlet in mobile cloud computing," *Inf. Sci.*, vols. 388–389, pp. 99–117, May 2017. doi: 10.1016/j.ins.2017.01.030.
[71] M. Z. Nayyer, I. Raza, and S. A. Hussain, "A survey of cloudlet-based mobile augmentation approaches for resource optimization," *ACM Comput. Surv.*, vol. 51, no. 5, 2018, Art. no. 107. doi: 10.1145/3241738.
[72] Y. Jararweh, L. Tawalbeh, F. Ababneh, A. Khreishah, and F. Dosari, "Scalable cloudlet-based mobile computing model," *Procedia Comput. Sci.*, vol. 34, pp. 434–441, Aug. 2014. doi: 10.1016/j.procs.2014.07.051.
[73] H. Shen, G. Bai, D. Ma, L. Zhao, and Z. Tang, "C2EM: Cloud-assisted complex event monitoring in wireless multimedia sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, p. 124, Dec. 2015.
[74] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognit.*, vol. 51, pp. 148–175, Mar. 2016. doi: 10.1016/j.patcog.2015.08.027.
[75] A. Das, S. C. M. Sharma, and B. K. Ratha, "The new era of smart cities, from the perspective of the Internet of Things," in *Smart Cities Cybersecurity and Privacy*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 1–9. doi: 10.1016/B978-0-12-815032-0.00001-9.
[76] H. Mora, J. F. Colom, D. Gil, and A. Jimeno-Morenilla, "Distributed computational model for shared processing on cyber-physical system environments," *Comput. Commun.*, vol. 111, pp. 68–83, Oct. 2017. doi: 10.1016/j.comcom.2017.07.009.
[77] H. Mora, M. T. Signes-Pont, D. Gil, and M. Johnsson, "Collaborative working architecture for IoT-based applications," *Sensors*, vol. 18, no. 6, p. 1676, 2018. doi: 10.3390/s18061676.
[78] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, "Enabling communication technologies for smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 112–120, Jan. 2017. doi: 10.1109/MCOM.2017.1600232CM.
[79] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli, "Low-power wireless for the Internet of Things: Standards and applications," *IEEE Access*, vol. 6, pp. 67893–67926, 2018. doi: 10.1109/ACCESS.2018.2879189.
[80] P. Marques *et al.*, "An IoT-based smart cities infrastructure architecture applied to a waste management scenario," *Ad Hoc Netw.*, vol. 87, pp. 200–208, May 2019. doi: 10.1016/j.adhoc.2018.12.009.
[81] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on IEEE 802.11ah: An enabling networking technology for smart cities," *Comput. Commun.*, vol. 58, pp. 53–69, Mar. 2015. doi: 10.1016/j.comcom.2014.08.008.
[82] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009. doi: 10.1109/MPRV.2009.82.
[83] T. Noor, S. Zeadally, A. Alfazic, and Q. Z. Sheng, "Mobile Cloud computing: Challenges and future research directions," *J. Netw. Comput. Appl.*, vol. 115, pp. 70–85, Aug. 2018. doi: 10.1016/j.aci.2016.11.002.
[84] F. Leccese, M. Cagnetti, and D. Trinca, "A smart city application: A fully controlled street lighting isle based on Raspberry-Pi card, a ZigBee sensor network and WiMAX," *Sensors*, vol. 14, no. 12, pp. 24408–24424, 2014. doi: 10.3390/s141224408.

[85] A. Abdullah, S. H. Yusoff, S. A. Zaini, N. S. Midi, and S. Y. Mohamad, "Smart street light using intensity controller," in *Proc. Int. Conf. Comput. Commun. Eng. (ICCCE)*, 2018, pp. 1–5. doi: 10.1109/ICCCE.2018.8539321.

[86] N. Khatavkar, A. A. Naik, and B. Kadam, "Energy efficient street light controller for smart cities," in *Proc. Int. Conf. Microelectron. Devices, Circuits Syst. (ICMDCS)*, 2017, pp. 1–6. doi: 10.1109/ICMDCS.2017.8211714.

[87] A. D. Shetty, Disha, B. Shubha, and K. Suryanarayana, "Detection and tracking of a human using the infrared thermopile array sensor—'Grid-EYE,'" in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, 2017, pp. 1490–1495. doi: 10.1109/ICICICT1.2017.8342790.

[88] B. Yang, Q. Wei, and M. Zhang, "Multiple human location in a distributed binary pyroelectric infrared sensor network," *Infr. Phys. Technol.*, vol. 85, pp. 216–224, Sep. 2017. doi: 10.1016/j.infrared.2017.06.007.

[89] *Intelligent Classic Luminaires, Tvilight*. Accessed: Feb. 22, 2019. [Online]. Available: https://www.tvilight.com/2016/10/18/case-study-intelligent-classic-luminaires/

[90] K. Oguchi, S. Maruta, and D. Hanawa, "Human positioning estimation method using received signal strength indicator (RSSI) in a wireless sensor network," *Procedia Comput. Sci.*, vol. 34, pp. 126–132, Aug. 2014. doi: 10.1016/j.procs.2014.07.066.

[91] H. Zou, Y. Zhou, H. Jiang, S.-C. Chien, L. Xie, and C. J. Spanos, "WinLight: A WiFi-based occupancy-driven lighting control system for smart building," *Energy Build.*, vol. 158, pp. 924–938, Jan. 2018. doi: 10.1016/j.enbuild.2017.09.001.

[92] N. K. Kandasamy, G. Karunagaran, C. Spanos, K. J. Tseng, and B.-H. Soong, "Smart lighting system using ANN-IMC for personalized lighting control and daylight harvesting," *Building Environ.*, vol. 139, pp. 170–180, Jul. 2018. doi: 10.1016/j.buildenv.2018.05.005.

[93] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy, "*LightLearn*: An adaptive and occupant centered controller for lighting based on reinforcement learning," *Building Environ.*, vol. 147, pp. 397–414, Jan. 2019. doi: 10.1016/j.buildenv.2018.10.028.

[94] F. Montori, L. Bedogni, M. Di Felice, and L. Bononi, "Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues," *Pervasive Mobile Comput.*, vol. 50, pp. 56–81, Oct. 2018. doi: 10.1016/j.pmcj.2018.08.002.

[95] J. W. King. (2009). Cisco IP Video Surveillance Design Guide. Cisco Systems, Inc. Accessed: Nov. 10, 2018. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/IPVS/IPVS_DG/IPVS-DesignGuide.pdf

[96] A. Attard. (2015). *Take a Screenshot of Web Page, Java Creed*. Accessed: Sep. 12, 2018. [Online]. Available: http://www.javacreed.com/take-a-screenshot-of-web-page/

[97] (2017). *Pedestrian Detection, GitHub*. Accessed: Sep. 12, 2018. [Online]. Available: https://gist.github.com/bigsnarfdude/52f96f1cf345492e93d3f520a09e67a0

[98] A. B. Amewuda, F. A. Katsriku, and J.-D. Abdulai, "Implementation and evaluation of WLAN 802.11ac for residential networks in NS-3," *J. Comput. Netw. Commun.*, vol. 2018, Mar. 2018, Art. no. 3518352. doi: 10.1155/2018/3518352.

[99] A. Šljivo *et al.*, "Performance evaluation of IEEE 802.11ah networks with high-throughput bidirectional traffic," *Sensors*, vol. 18, p. 325, Jan. 2018. doi: 10.3390/s18020325.

HIGINIO MORA received the B.S. degree in computer science engineering, the B.S. degree in business studies, and the Ph.D. degree in computer science from the University of Alicante, Spain, in 1996, 1997, and 2003, respectively, where he has been a member of the Faculty at the Computer Technology and Computation Department, since 2002, and is currently an Associate Professor and a Researcher of the Specialized Processors Architecture Laboratory. He has participated in many conferences, and most of his work has been published in international journals and conferences, with more than 60 published papers. His research interests include computer modeling, computer architectures, high-performance computing, embedded systems, the Internet of Things, and cloud computing paradigm.

JESÚS PERAL received the Ph.D. degree in computer science from the University of Alicante, in 2001, where he is currently an Assistant Professor with the Department of Software and Computing Systems. He has participated in numerous national and international projects, agreements with private companies and public organizations related to his research topics. He has published many papers (more than 40 papers) in journals and conferences related to his research interests. His main research interests include natural language processing, information extraction, information retrieval, question answering, data warehouses, and business intelligence applications.

ANTONIO FERRÁNDEZ received the Ph.D. degree in computer science from the University of Alicante, in 1998, where he is currently an Assistant Professor with the Department of Software and Computing Systems. He has participated in numerous national and international projects, agreements with private companies and public organizations related to his research topics. He has participated in many conferences, and most of his work has been published in international journals and conferences, with more than 70 published papers. His research interests include information extraction, information retrieval, question answering, natural language processing, and ellipsis and anaphora resolution.

DAVID GIL is currently an Assistant Professor with the Department of Computing Technology and Data Processing, University of Alicante. He has participated in numerous national and international projects, agreements with private companies and public organizations related to his research topics. He has participated in many conferences, and most of his work has been published in international journals and conferences, with more than 50 published papers. His main research interests include artificial intelligence applications, data mining, open data, big data, and decision support systems in medical and cognitive sciences.

JULIAN SZYMANSKI is currently an Assistant Professor with the Gdansk University of Technology, Poland. He is currently working on the methods of semantic document categorization. As part of his research, he is developing algorithms to more accurately extract information from text by using natural language processing tools and computational intelligence methods. In his work, cognitive theories of human knowledge serves as an inspiration for the implementation of the algorithms to process the data in a way similar to how it would be done by humans. The results of the research are used to organize knowledge in Wikipedia. His research interests include the domain of cognitive science and information retrieval.