A Smart Classroom based on Deep Learning and Osmotic IoT Computing

Alberto Pacheco, Pablo Cano, Ever Flores, Edgar Trujillo, Pedro Marquez
Tecnológico Nacional de México / I.T. Chihuahua, MÉXICO
Email: { apacheco, pcanof, eaflores, edgar.trujillo, pmarquez }@itchihuahua.edu.mx

***Abstract*— The biggest growth rate of network traffic in the coming years will be for smartphones and Internet-connected devices, which relentless tend to perform increasingly demanding tasks on continuously increasing amounts of data. Machine Learning and Edge Computing are emerging as effective paradigms for processing huge amounts of data supplied by the Internet of Things and Smart Cities. An osmotic computing architecture for an IoT smart classroom is used for testing a deep learning model for person recognition. A comparative performance study and analysis was made by means of selecting a single deep learning model, that it was tried to be adapted to run over the cloud, a fog microserver and a mobile edge computing device. The results obtained shown some promising results and also limitations for the edge and fog computing side that will need to be addressed in order to minimize latencies and achieve real-time responses for the present IoT application.**

Keywords— Mobile Edge Computing; Deep Learning; Cloud Computing; Internet of Things; Smart Living; Smart Buildings.

I. INTRODUCTION

The ‘Smart’ City concept: a) is profoundly complex, and has been mutated from prior adjectives such as *digital*, *creative*, *knowledge-based*, *sustainable*, *resilience-driven* or *intelligent* cities [1-4]; b) it combines different indicators, fields and technological perspectives, such as the ICT-based ‘hard’ perspective in the 1990s [5], and the recent Artificial Intelligence (AI) and the Big Data ‘soft’ perspectives; c) is scalable to ‘smart’ rooms, homes, buildings, villages, cities or nations, and can be approached from diverse paradigms, such as smart living, ambient-assisted living, or cyber-physical systems [1, 2]. Ramaprasad et al. [1] reported more than 36 smart city definitions, and proposes a unified *Smart City Ontology (SCO),* which is able to instantiate thousands of variations from six domains (Table I): outcomes, stakeholders, semiotics, structure, functions, and focus.

TABLE I. THE SMART CITY ONTOLOGY FRAMEWORK (SCO)

DOMAINS	ELEMENTS
1. *Outcomes*	Quality, sustainability, equity, livability, resilience
2. *Stakeholders*	Citizens, professionals, institutions, businesses…
3. *Semiotics*	Data, pattern, stream, information, knowledge…
4. *Structure*	Architecture, services, systems, infrastructure…
5. *Functions*	Sense, identify, monitor, control, translate, notify
6. *Focus*	Economical, environmental, social, cultural…

A properly instrumented, interconnected and smart city aims to achieve a triple sustainability (social, economic and environmental) processing large amounts of data through ICT, embedded systems, IoT and AI, in order to enhance how a city works and to improve its citizen wellness and quality of life [2, 6]. Some of its current challenges are [2, 5, 8]: to improve interoperability, standardization, scalability and decentralized control; management large amounts of data and analytics in real time; security, privacy, reliability and fault tolerance; context intelligence, autonomy and cognitive services; handle complex, dynamic, heterogeneous and unpredictable environments; to meet user and social expectations and implications; and management of energy generation and consumption.

With the aforementioned conceptual framework (Table I), scaling it down up to the smart room level, the project goal can be established as follows: implement a real-time adaptable system (*structure*) to collect and analyze *(functions)* environmental and contextual data (*semiotics & focus*), directly from all classmates (*stakeholders*) in order to automate and control several classroom facilities *(functions)* addressed for energy saving, as well as for comfort purposes (*outcomes*), inspired on the ‘smart living’ paradigm, from an osmotic IoT ‘hard’ perspective and the AI ‘soft’ perspective. Therefore, this article reports a smart lighting control prototype that uses a vision-monitored system and an osmotic IoT architecture as a test bed for running deep learning models, in this case, to identify objects, people, and handwritten text using images taken from a vision system. It is also relevant to highlight that neither a learning or teaching performance metric was considered for this smart classroom design. The focus was instead oriented to determine which osmotic computing layer (edge/fog/cloud) performs better when executes a specific deep learning model.

A. IoT and Big Data Challenges

Based on various reports [10-12], it is foreseen that in 2016-2021 period, network traffic will triple in data centers, going from 6.8 Zb to 20.6 Zb (94% are cloud services); daily network traffic will rise from 26,600 Gb/s to 105,800 Gb/s; the speed of the global network will go from 27 Mbps to 53 Mbps; smartphone IP traffic (33%) will surpass computer traffic (25%); wireless and mobile traffic will represent more than 63% of total traffic; the highest growth rate (49%) will be for mobile phones and machine-to-machine connections (M2M); there will be more than 50 billion devices connected to the Internet (IoT) increasing the traffic 10X or more; the global IoT market between 2016-2023 will rise from 16 billion to 186 billion dollars; and the global market for smart homes in 2016-2025 will grow from 46 billion to 130 billion dollars. All these forecasts implies a big problem about how to analyze this big data and internet traffic explosion sprout out from billions of

This project was partially supported by Tecnologico Nacional de Mexico.

978-1-5386-8131-2/18/$31.00 ©2018 IEEE

interconnected mobile and IoT devices able to provide real-world sensed data without people intervention using M2M protocols. Among the leading IoT challenges are [13-17]: extracting semantic and contextual information in real time by analyzing large amounts of data from dynamic and noisy environments, and designing better architectures from the point of view of safety and energy efficiency.

B. *A Cloudy, Foggy and Edgy AI*

Artificial Intelligence can profoundly change our daily lifestyle [18]. Over the past 50 years, AI has developed logic-based systems, bio-inspired systems, collaborative agents, cyber-physical intelligent systems, and ubiquitous AI or pervasive intelligence systems. The great ability of Machine Learning (ML) – a subarea of AI – is to analyze large data flows. It has been popularized and consolidated during the last decade at all levels of research, industry and entertainment [18-20]. In less than a year, very ambitious national AI initiatives have emerged in different countries [21] and inside companies such as Google, Amazon, Facebook, Microsoft, IBM, Apple, Intel, Alibaba, Baidu, Uber, among many others. These companies have invested large amounts of money in developing innovative products, applications and platforms available as cloud services capable of efficiently execute ML models, i.e. MLaaS. Nowadays, cloud computing is a cost effective and prevailing platform (Fig. 1) offering huge processing power and storage capacity for performing ML models for facial recognition, speech recognition, computer vision, automated language processing, text classification and for many IoT and smart city services and applications. However, it also has some important disadvantages (Fig. 1) like network response latency, and secondly, the security of the system and privacy concerns [20].

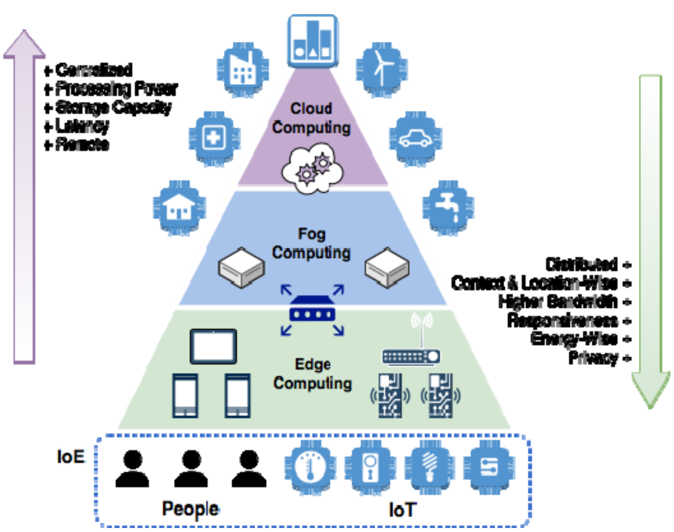

Fig. 1. Osmotic IoT Computing architecture for Smart Cities.

C. *Mobile Edge Computing (MEC)*

Most IoT control actions must be carried out in real time, so cloud processing waiting time does not work well for IoT problems [22]. Some recent and complementary paradigms that try to cover some of these limitations are Fog and Edge Computing, which claims to be capable of performing tasks in a more distributed and responsive way when IoT nodes are closer to the sensor data sources (Fig. 1), also reducing network traffic and avoiding to expose user data and privacy.

The Edge Computing (EC) paradigm is so new that it lacks a standard definition, architectures and protocols. Some attempts to define it appear in [13-15]: the ETSI proposal in 2014, NIST proposal in 2017; Open Fog consortium in 2015, and the Open Edge Computing initiative in 2015. Currently, there is a growing number of EC applications [13, 14, 16] in automation, monitoring, control and security in embedded systems and industrial-type IoT ecosystems like lighting, air conditioning and intelligent monitoring, analytical, energy saving, security for smart buildings; monitoring and remote assistance for patients, rehabilitation and elderly people; intelligent vehicles, assisted driving, V2X protocols, traffic monitoring and control; smart clothes, games, augmented reality and virtual reality, fitness and training.

The leading EC advantages are [13-17]: reduction in latency and response time suitable for real-time applications; processing, filtering, compression and encryption of data flows; reduction of data traffic and storage; greater context and location sensitivity; achieve distributed cognitive processing; and better use of the computing and communication potential present in current mobile edge computing (MEC) devices and IoT nodes. Finally, among the main obstacles to perform data processing in these MEC devices are [14, 23]: its limited energy capacity and its limited processing capability.

D. *On-Device Inference (ODI)*

A recent Edge Computing trend consists in applying neural network models acceleration and compression techniques to perform on-device inferences (ODI) over constrained devices [17-20, 25-26]. ODI-MEC devices may be useful for activity and context recognition [25], bringing also a new direction and fast evolving computer architecture topic [19], migrating from high latency, power-hungry cloud platforms towards ubiquitous and fast MEC devices equipped with a handful variety of sensors and multiple low-power processing cores that can bring ODI efficient performance over mobiles, wearables, and IoT nodes within the milliwatt or microwatt power range [20]. In 2015, DeepEar was one of the first ODI architectures able to perform a compressed voice recognition DNNs running on a very low-power DSP [25, 26]. Further dataflow and GPU-based architectures were DeepX [27], DeepSense [28], CNNdroid [29], DeepMon [30], MobiRNN [31], RSTensorFlow [32], DeepEye [33], and eBNN [35]. Some FPGA-based architectures are Caffeine [30], SPARCNet [31], and fpgaConvNet [32]. Other xPU architectures are [38]: Intel Movidius Vision (VPU), NVIDIA DLA, Apple Neural Engine, Google Tensor (TPU), Microsoft Holographic (HPU), Qualcomm Neural Engine (NPU), GraphCore Intelligence (IPU), Horizon Robotics Brain (BPU), Deephi DL (DPU), and Emoshape Emotion (EPU).

II. METHOD

In the development of the smart lighting prototype, an osmotic IoT architecture was considered (Fig. 2), which was adapted as a test bed for running deep learning models in order to determine which osmotic layer – edge, fog or cloud – has the highest overall performance, following the following stages:

1) Research stage: The first step was to find a single CNN model able to run across different devices, platforms and osmotic computing architectural layers to ensure a reliable performance comparative study.

2) Implementation stage: Deploy the osmotic computing IoT system (Fig. 2). Select and configure the corresponding cloud services (MLaaS) and fog IoT microserver (FIMS) attaching their respective IoT devices: web cameras, remote lights bulbs, sensors and actuators. In addition, install most recent version of operating system, compiler and ML framework for the mobile edge computing (MEC) device to develop the mobile app for testing the selected CNN model. A similar procedure was also considered to install the CNN accelerator to execute the CNN model inside FIMS computer.

3) Evaluation stage: Determine network latencies and processing times for DNN inferences across the diverse cloud services, fog and edge devices in order to analyze, compare and determine their respective advantages, implementation constraints and limitations.

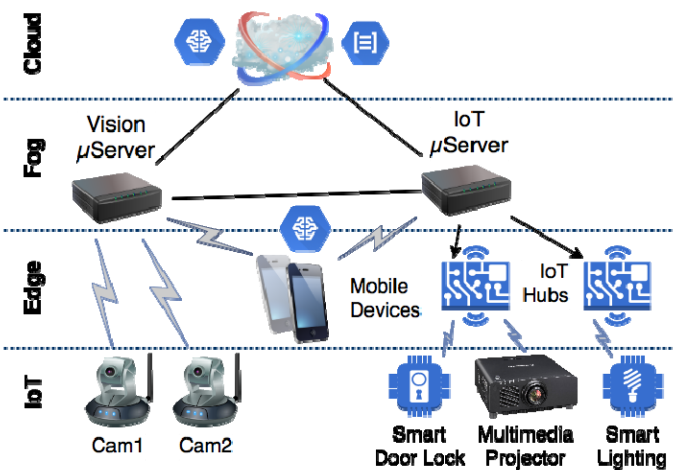

Fig. 2. Osmotic IoT architecture to perform Deep Learning models.

III. A SMART CLASSROOM TEST BED

Adopting an Osmotic Computing paradigm (Fig. 1), a hierarchical, multilayer and distributed smart classroom IoT ecosystem was implemented organized and described according the following architectural levels (Fig. 2):

a) Cloud Level: The Google Cloud Platform (GCP) with an almost unlimited availability of high-performance TPUs and its Cloud Vision API for object and text recognition was selected;

b) Fog Level: For the fog IoT microserver (FIMS) it was selected an Intel NUC7-i5BNHXF mini-PC with a dual-core Pentium i5-7260U 64-bit CPU @ 2.2 GHz and 16 GB RAM (15 watts). The FIMS installation included Ubuntu 16.04 LTS and also an open-source motion system [39] configured to auto-detect motion and capture time-stamped images from two webcams at 640 x 480 resolution and JPEG at 75% Q-factor (26-32 KB image size), and a local image repository available via SFTP. The Intel Movidius Neural Compute Stick was installed to accelerate CNN model execution in the NUC-based FIMS. A Raspberry Pi 3B+ IoT microserver was deployed with a quad-core ARMv8 64-bit CPU @ 1.4 GHz and 1 GB RAM (5W) as a Flask-based web server to control IoT devices;

c) Edge level: The mobile edge computing (MEC) device for ODI testing used was an Apple iPad 6th-Gen 9.7" tablet that has a quad-core A10 Fusion ARMv8 64-bit CPU @ 2.34 GHz and 2 GB RAM with iOS 12 (beta) installed.

d) IoT devices: Some Logitech C170 and C120 webcams were installed on FIMS (Fig. 3). The smart lighting module was implemented using Philips Hue dimmable LED bulbs and one WiFi/ZigBee bridge handled directly by a Raspberry Pi IoT microserver. Other IoT devices not included in this report were: a wireless smart lock door and a multimedia projector.

Fig. 3. Vision IoT fog microservers: Deeplens, NUC and Raspberry Pi.

Previously, different DNN models (Inception v3, MobileNet, and SqueezeNet) for object recognition were successfully tested for ODI-MEC execution [40]. The present work uses a Tiny YOLO v2 (64 Mb) trained in VOC dataset tested across all edge, fog and cloud levels.

For edge device testing, a mobile iOS app was developed using the most recent CoreML2 (beta) under two configurations: 1) to take a picture and execute the CNN model; 2) to run the CNN model over video streaming (Fig. 4).

At the cloud and fog level, the NUC-based FIMS was able: 1) to invoke Google Cloud Vision and ML API to analyze text written on a chalkboard (Fig. 5) using images taken from the webcam (Fig. 3); 2) to perform CNN inferences at fog level using the Intel Movidius Neural Compute Stick; 3) to control the smart classroom lights, multimedia projector and door lock.

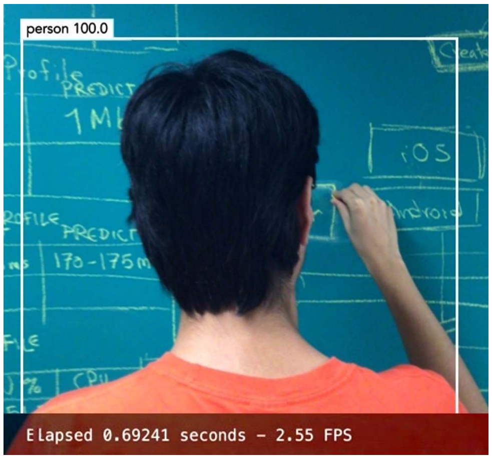

Fig. 4. Mobile edge device testing a YOLO model.

IV. Performance Results

The summarized results are included and labeled inside the UML sequence diagram of the Fig. 6. This diagram includes alternative sequence flows for each edge, fog and cloud service (*Alt* block, Fig. 6). The overall performance can include different components: 1) the DNN inference time (red bars); 2) the network latency (double lines); 3) the task's preamble and epilogue times (undetermined overhead).

The CNN inference over the edge device (label A, Fig. 6) did not require any network access. Therefore, it does not account for any network latency, as appears in Table II. Processing one image at a time in the edge device using a Tiny YOLO v2 model had an average inference time of 330 ms. The fog device assisted by the Intel Movidius, has an inference execution time slightly faster, averaging 200 ms (label B, Fig. 6). However, its overall performance (+600ms) was 2x slower than edge device, due to network latencies and some overhead.

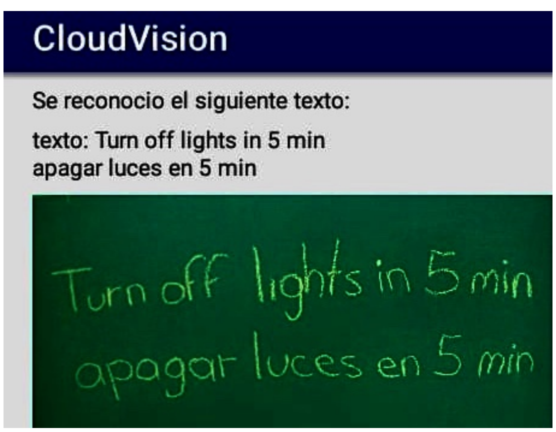

Fig. 5. RNN model for text recognition using Cloud Vision services.

Regarding video streaming ODI performance, the edge device reached up to 6 FPS vs. 5 FPS for fog device. Despite its slower edge's ODI time, it performed better than fog device, probably by its higher level of hardware/software integration and ML/Vision optimizations for video processing or maybe due to Fog/Movidius USB transfer data bottleneck.

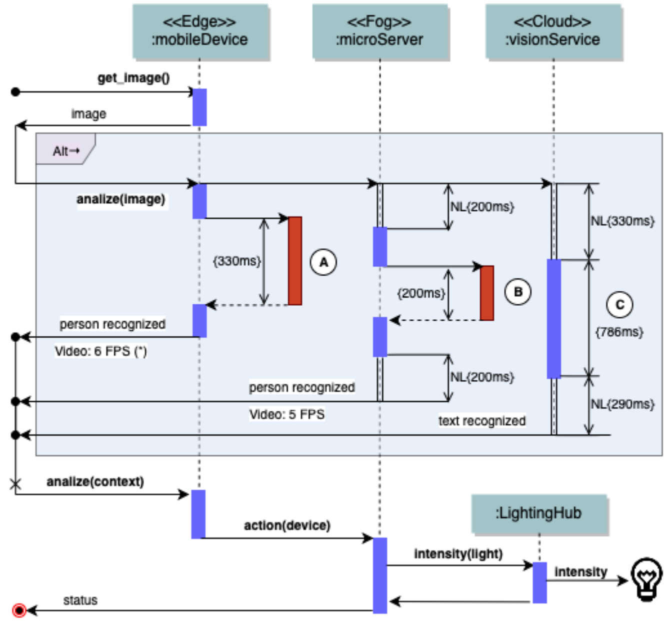

Fig. 6. UML sequence diagram for edge/fog/cloud-based smart lighting.

Unfortunately, it was not possible to run the Tiny YOLOv2 model as a GCP cloud service. Instead a recurrent neural network for text recognition was tested, yielding an average cloud processing time of 786 ms with an undetermined inference time (label C, Fig. 6) and around 620 ms for network latencies. Despite using a different DNN model, this alternative was almost 4x slower than edge device's overall performance.

TABLE II. ODI Performance Tiny YOLO v2 Results

Osmotic Level	Avg. Latency (ms)	Avg. Inference Time (ms)	Avg. FPS
Edge	0	330	6
Fog	400	200	5

V. Conclusions

Smart Cities seems to increase the amount of IoT sensors, Deep Learning is well suited for this kind of big data explosion. An osmotic computing architecture can exploit the best of the edge, fog and cloud worlds. Adopting a modular and scalable architecture design, a smart classroom prototype was implemented for testing Deep Learning inferences on each one of these layers. Overall performance results shown that on-device ODI-MEC DNN inference outperformed almost by 2x for FIMS, and 4x for a cloud service, partly due to network latencies and data transfer bottlenecks. In order to derive a more powerful osmotic computing configuration, some changes can be incorporated into the fog IoT microserver, improve network speed and protocol design, reduce transfer-and-process bottlenecks, and add multiple ODI units per FIMS, e.g. Intel Movidius and other DNN accelerators. Also, new candidates for implementing a fog IoT microserver are emerging, e.g. Amazon AWS Deeplens.

ACKNOWLEDGMENT

This research was partially supported by the Tecnológico Nacional de México under the project "Aula Inteligente con Interfaz Natural basada en Realidad Aumentada y Aprendizaje Automático" No. 6417.18-P (2018).

REFERENCES

[1] A. Ramaprasad, A. Sánchez-Ortiz, & T. Syn, "A Unified Definition of a Smart City," in Electronic Government. EGOV 2017, vol. 10428, M. Janssen, K. et al., Eds. Springer Int. Publishing, 2017, pp. 13–24.

[2] H. Song, R. Srinivasan, T. Sookoor & S. Jeschke, Smart cities: foundations, principles, and applications. John Wiley & Sons, 2017.

[3] O. Mkrtychev, Y. Starchyk, S. Yusupova, and O. Zaytceva, "Analysis of various definitions for Smart City concept," IOP Conf. Ser.: Mater. Sci. Eng., vol. 365, no. 2, p. 22065, 2018.

[4] A. Celani, S. Bellintani, G. Ciaramella, and A. Balakina, "Smart and Sustainable cities, a new definition for a Resilience-driven city," IOP Conf. Ser.: Mater. Sci. Eng., vol. 365, no. 2, p. 22062, 2018.

[5] S. Alawadhi, A. Aldama-Nalda, H. Chourabi, et al., "Building Understanding of Smart City Initiatives," Lecture Notes in Computer Science 7443 (2012) 40–53.

[6] V. Albino, U. Berardi, and R. M. Dangelico, "Smart Cities: Definitions, Dimensions, Performance, and Initiatives," *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, Jan. 2015.

[7] N. Komninos, M. Pallot, and H. Schaffers, "Special Issue on Smart Cities and the Future Internet in Europe," *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 119–134, Jun. 2013.

[8] T. Hui, R. Sherratt, & D. Sánchez, "Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies," Future Generation Computer Systems, vol. 76, pp. 358–369, nov. 2017.

[9] D. Minoli, K. Sohraby, & B. Occhiogrosso, "IoT Considerations, Requirements, and Architectures for Smart Buildings, Energy Optimization and Next-Generation Building Management Systems", IEEE Internet of Things Journal, vol. 4, núm. 1, pp. 269–283, feb. 2017.

[10] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper," San Jose, CA, White Paper 1454457600805266, 2017. Available: https://goo.gl/2aVofM

[11] WinterGreen Research, "Internet of Things (IoT) Market Shares, Strategies, and Forecasts, Worldwide, 2017 to 2023," Lexington, Massachusetts, Market Analysis Report SH26944999, 2017. Available: https://goo.gl/bDZDB7 [Accessed: 26-Jun-2018].

[12] Grand View Research, "Smart Home Automation Market Size, Industry Trend Report 2014-2025," San Francisco, CA, Market Research Report GVR-1-68038-880-0, Aug. 2017. Available: https://goo.gl/FCRteF

[13] K. Bilal, O. Khalid, A. Erbad & S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," Computer Networks, vol. 130, pp. 94–120, Jan. 2018.

[14] M. Satyanarayanan, "The Emergence of Edge Computing," Computer, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[15] A. Iorga, A. Feldman, A. Barton, A. Goren & A. Mahmoudi, "SP 800-191 (Withdrawn DRAFT), The NIST Definition of Fog Computing." Available: https://goo.gl/yDpN4e. [Accessed: 14-mar-2018].

[16] M. Mohammadi & A. Al-Fuqaha, "Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges," IEEE Communications, vol. 56, núm. 2, pp. 94–101, feb. 2018.

[17] H. Li, K. Ota, & M. Dong, "Learning IoT in Edge," IEEE Network, vol. 32, no. 1, pp. 96–101, 2018.

[18] Q. Li, Q. Xiao & Y. Liang, "Enabling high performance deep learning networks on embedded systems," in IECON 2017 - 43rd Annual Conf. IEEE Ind. Electronics Soc., 2017, pp. 8405–8410.

[19] M. Verhelst & B. Moons, "Embedded DNN Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices," IEEE Solid-State Circuits Mag, vol. 9, no. 4, pp 55–65, 2017.

[20] Dave Gershgorn, "Forget the Space Race. The Artificial Intelligence Race is Just Beginning," 2018. Available: http://shamov.info/node/652

[21] M. S. B. Kolhar, M. G. M, & A. Alameen, "Cloud Servers And Fog Or Edge Computing With Their Limitations & Challenges," *Journal of Computer Hardware Engineering*, vol. 1, núm. 1, pp. 1–7, apr. 2018.

[22] S. Singh, "Optimize cloud computations using edge computing," in Int. Conf. on Big Data, IoT and Data Science (BID), 2017, pp. 49–53.

[23] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Communications Surveys Tutorials, vol. 19, no. 3, pp. 1628–1656, 2017.

[24] MIT Technology Review Insights, "On-Device Processing and AI Go Hand-in-Hand", MIT Technology Review. [13-mar-2018]. Available: https://goo.gl/76F8Le. [Accessed: 19-mar-2018].

[25] P. Georgiev, S. Bhattacharya, N. Lane & C. Mascolo, "Low-resource Multi-task Audio Sensing for Mobile and Embedded Devices via Shared Deep Neural Network Representations," Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 1, no. 3, pp. 50:1–50:19, Sep. 2017.

[26] N. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi & F. Kawsar, "Squeezing Deep Learning into Mobile and Embedded Devices," IEEE Pervasive Computing, vol. 16, no. 3, pp. 82–88, 2017.

[27] Y. He, F. R. Yu, N. Zhao, VC Leung & H. Yin, "Software-defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," IEEE Comm. Magazine, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[28] L. N. Huynh, R. K. Balan & Y. Lee, "DeepSense: A GPU-based Deep Convolutional Neural Network Framework on Commodity Mobile Devices", in Proc. 2016 Workshop on Wearable Systems and Applications, New York, USA, 2016, pp. 25–30.

[29] S. Latifi Oskouei, H. Golestani, M. Hashemi & S. Ghiasi, "CNNdroid: GPU-Accelerated Execution of Trained Deep Convolutional Neural Networks on Android," in Proc. 2016 ACM Multimedia Conf., NY, USA, 2016, pp. 1201–1205.

[30] L. Huynh, Y. Lee & R. Balan, "DeepMon: Mobile GPU-based Deep Learning Framework for Continuous Vision Applications," on Proc. Int. Conf. on Mobile Systems, Applications & Services, 2017, pp. 82–95.

[31] Q. Cao, N. Balasubramanian & A. Balasubramanian, "MobiRNN: Efficient Recurrent Neural Network Execution on Mobile GPU," in EMDL'17, NY, USA, 2017, pp. 1–6.

[32] M. Alzantot, Y. Wang, Z. Ren & M. Srivastava, "RSTensorFlow: GPU Enabled TensorFlow for Deep Learning on Commodity Android Devices," on Proc. Intl. Work. Deep Learning for Mobile Sys. & App., NY, USA, 2017, pp. 7–12.

[33] A. Mathur, N. Lane, S. Bhattacharya, A. Boran, C. Forlivesi & F. Kawsar, "DeepEye: Resource Efficient Local Execution of Multiple Deep Vision Models Using Wearable Commodity Hardware," in Proc. Intl. Conf. on Mobile Sys., App. & Services, 2017, pp. 68–81.

[34] W. Meng, Z. Gu, M. Zhang & Z. Wu, "Two-Bit Networks for Deep Learning on Resource-Constrained Embedded Devices," arXiv: 1701.00485 preprint, Jan. 2017.

[35] C. Zhang, Z. Fang, P. Zhou, P. Pan & J. Cong, "Caffeine: towards uniformed representation and acceleration for deep convolutional neural networks," in Proc. of ICCAD '16, Austin, TX, 2016, pp. 1–8.

[36] A. Page, A. Jafari, C. Shea & T. Mohsenin, "SPARCNet: A Hardware Accelerator for Efficient Deployment of Sparse CNNs," J. Emerg. Technol. Comput. Syst., vol. 13, núm. 3, p. 31:1–32, may 2017.

[37] S. Venieris & C. Bouganis, "fpgaConvNet: A Toolflow for Mapping Diverse Convolutional Neural Networks on Embedded FPGAs," in Proc. of NIPS 2017, Long Beach, CA, 2017.

[38] Y. Xie, "A Brief Guide of xPU for AI Accelerators," ACM SIGARCH. April, 2018. Available: https://www.sigarch.org/?p=15093

[39] Motion-Project, "Motion - Open source security camera software," [Online] Available: http://motion-project.github.io [Accessed: 04-07-18]

[40] A. Pacheco, E. Flores, R. Sanchez, and S. Almanza, "Smart Classrooms aided by Deep Neural Networks inference on Mobile Devices," in Proc. 2018 IEEE International Conference on Electro Information Technology (EIT), Michigan, 2018, pp. 605–609.