Performance of video processing at the edge for crowd-monitoring applications

Camille Ballas, Mark Marsden, Dian Zhang, Noel E. O'Connor and Suzanne Little
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
{camille.ballas, mark.marsden}@insight-centre.org
{dian.zhang, noel.oconnor, suzanne.little}@dcu.ie

***Abstract*—Video analytics has a key role to play in smart cities and connected community applications such as crowd counting, activity detection, event classification, traffic counting etc. Using a cloud-centric approach where data is funnelled to a central processor presents a number of key problems such as available bandwidth, real-time responsiveness and personal data privacy issues. With the development of edge computing, a new paradigm for smart data management is emerging. Raw video feeds can be pre-processed at the point of capture while integration and deeper analytics is performed in the cloud. In this paper we explore the capacity of video processing at the edge and shown that basic image processing can be achieved in near real-time on low-powered gateway devices. We have also investigated deep learning model capabilities for crowd counting in this context showing that its performance is highly dependent on the input size and rescaling video frames can optimise processing and performance. Increased edge processing resolves a number of issues in video analytics for crowd monitoring applications.**

I. Introduction

Smarter cities utilising Internet-of-Things (IoT) technologies are required to provide a sustainable environment to accommodate the needs of the increasing urban population of tomorrow and preserve natural resources [1]. In [2] the authors define smart cities as *"an urban system that uses information and communication technology (ICT) to make both its infrastructure and its public services more interactive, more accessible and more efficient"*. A smart city goal is to improve the quality of life of citizens while providing a sustainable environment and reducing the cost of living. This objective can be achieved by deploying sensors (cameras, microphones) across the city to capture urban data and analyse them to extract knowledge and wisdom in order to improve city management.

However, because cities are big and public areas, it represents a big challenge to manage the quantity and variety of potential sensor data including video. Three major issues are: (i) optimising network bandwidth; (ii) real-time responsiveness and (iii) preserving personal data privacy. Various applications of image processing and computer vision require high to moderate resolution data that increases the volume of data that needs to be transferred from the camera to a central processing server or cloud system. Crowd monitoring using video can provide excellent real-time data on crowd density or abnormal situations but require a fast turn around of analytics. Finally the general public is increasingly aware of the potential intrusiveness of surveillance video systems and legislation is being enacted to protect personal data requiring system to avoid the capture, transmission and storage of data where individuals are visible.

978-1-4673-9944-9/18/$31.00 © 2018 IEEE

In the early stage of IoT, the majority of the infrastructures were cloud centric to exploit the computational power of shared resources and to facilitate large-scale deployment and scaling of data processing. But cloud architectures have shown some limitations in terms of bandwidth, latency and data privacy. Edge applications, also known as fog computing, have emerged as an alternative architecture to alleviate some of those issues [3]. For example, in a video-based crowd monitoring application captured frames can be transformed into meaning information (number of people in the scene, main activity) at the device level, close to point-of-capture, enhancing response-time and improving data privacy by pushing only low-bandwidth, anonymous data (numbers) to the cloud for analysis and aggregation.

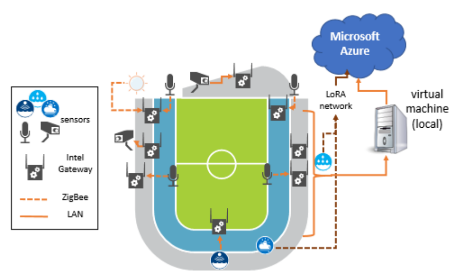

Fig. 1. Croke Park sensors infrastructure

A common challenge for practical research in IoT and smart environments is the gap between laboratory prototyping and deploying in a real-world scenario. The Smart Stadium for Smarter Living project[1] is a research collaboration between Insight Centre for Data analytics (Dublin City University), Intel, Microsoft, the GAA (Gaelic Athletic Association) and

[1] https://channel9.msdn.com/Blogs/DX-Ireland/Croke-Park-IOT-Smart-Stadium-Dublin

Croke Park (Dublin) to provide within Croke Park stadium a test bed for smart cities solutions. The stadium offers an ideal environment, small enough to trial but big enough to prove. Figure1 shows the current sensors installation in Croke Park including gateways, sound level monitoring devices and cameras.

A stadium environment is very like a city. During an event (a match or a concert), a stadium has to deal with a large crowd facing the same difficulties as a city centre area. The stadium experience can be improved by monitoring the crowd to avoid congestion and improve emergency management, or by monitoring the sound level for nearby residential areas.

Among the common scenarios to smart city and smart stadium, crowd monitoring is an interesting one to investigate. Highly congested city areas present both safety risks for people, economic performance and can have a significant impact on the quality of life. Crowd monitoring is through video analytics and many cities are already equipped with Closed-Circuit Television Video (CCTV) network. Video analytics can be computationally expensive and the data captured are highly sensitive. An area of significant and active interest is to develop smart cameras capable of running sophisticated algorithms for person detection or object counting. However, this requires equipment upgrades and generally expensive hardware to implement.

In this paper we explore the capacity of video processing at the edge of an IoT infrastructure utilising low-powered gateway devices for a crowd monitoring use case. In the following section we present a brief overview of video analytics in smart cities. Then we explore the limitations of processing at the edge comparing the performance of three different algorithms with different complexity levels, using a Dell Edge Gateway 5000. Finally we discuss the feasibility of video processing at the edge focussing on the processing speed and capacity. We have found that even state-of-the-art approaches based on computationally expensive deep learning methods can be orchestrated on gateway devices though the trade off in responsiveness is significant.

II. Video processing and smart cities

Over the last decade IoT has extended computing capacity beyond traditional desktops and servers, connecting the real and virtual together [4]. This provides a distributed environment of inter-connected devices able to communicate and exchange data with each other to enable environment awareness and data-driven decision making. The amount of data generated by IoT devices in 2014 was estimated as 233 exabytes and by 2020 this number is set to exceed 1.600 exabytes [5]. Therefore, particularly with the additional load of video data, efficient management of data is critical for successful deployment in smart cities and smart ecosystems.

Many cities are already equipped with a video surveillance network and leveraging video analytics for near real-time applications at the edge can facilitate smart cities development. However, the processing power required is very different when you move from simple statistical computations on sets of numbers to analysing images in near real-time.

Deep learning has emerged in the past 10 years as a powerful tool to handle complex decision-making and more specifically image processing, achieving human-level prediction [6], [7]. In [8] the authors present an overview of deep learning algorithms useful for video analytics in smart cities (object detection, object tracking, face recognition and image classification). Deep learning models have a key role to play in the future of smart city; one model can achieve up to three different tasks [9]. One challenge of applying this technology in realistic IoT environments is the need for specific powerful hardware in the form of GPUs to train and process the networks.

A. Smart cities applications using video processing

In this section we present two smart cities applications that have been explored by research from a video analytics perspective – smart lights and smart transportation.

1) Smart Lights: City streets are lit for over half the day to increase road safety as well as citizen comfort. Lighting systems represent one of the major sources of energy consumption and have to be targeted to preserve resources. Veena et al. [10] proposed a light system where video analytics is used to recognise movement in real-time to activate street lights during low traffic hours. They used a raspberry Pi to leverage computational capabilities at the edge. The camera frames are ingested by the Raspberry Pi and converted to a grey scale image. Contour mapping and object extraction is then applied using the OpenCV library and the result is compared to a threshold value to activate or not the street lights.

2) Smart Transportation: Another big application for video analytics in smart cities is traffic management. CCTV cameras are already used by humans to monitor and regulate the traffic but automated systems to detect, recognise, classify and count objects are the target of much research. Aryal et al. [11] proposed a framework for object recognition. However, their framework operates on the Cloud and requires high resolution images. The city of Boston, in association with IBM, is engaged in a smart city urban plan specifically including smart traffic control using CCTV cameras [12].

Bonomi et al. [13] present a way to balance edge and cloud analytics for smart traffic lights system. Traffic lights are equipped with sensors that detect the presence of pedestrians and cyclists and sensors measure the speed of incoming cars to perform instant decision making at the edge (change the colour of the traffic light) and at the same time enhance long term planning prediction in the cloud.

In [14] the authors have surveyed smart cities from a data centric point of view. Several smart city applications have demonstrated that video processing has a key role to play in the city of tomorrow. Edge computing is appealing for real-time processing but it lacks computational power, but some studies have proposed efficient framework to mine data at the edge [15].

Balancing data processing between the cloud and the edge can greatly improve performance, responsiveness and cost. Raw data that is transformed into meaningful information at the edge reduces the cost of using cloud services, while heavier processing and decision making can be performed in the cloud to aggregate data and perform stronger analytics [16]. Gateway devices have the potential to enhance data processing on the edge but limited studies have been conduct on the specific trade-offs that can be made between performance and responsiveness. We focus on image processing and deep learning to explore how far we can take real-time video processing at the edge for application to crowd monitoring in a smart environment.

B. Data processing

With the large volume of data generated by smart devices, a natural platform to handle IoT application is cloud computing. It presents many advantages as it is prone to scale up, and offers infinite resources to alleviate computational weakness of the devices at the edge [17]. However, using cloud services implies offloading some workload to a third party. This can be an issue in terms of data privacy and reliability. Video data for example are highly sensitive data and with new law enforcements like the European regulation 2016/679 [18] it will become harder to outsource sensitive data.

But one of the main drawbacks of cloud computing is dealing with real-time applications. In [19] the authors have surveyed network topologies for real-time application and state that in their current form, data centres are not suitable for real-time processing due to network lag and transfer delays. This was also identified in the Smart Stadium project during an exercise to measure crowd sound on busy match days [20] [21].

In [22], Bonomi et al. presented an alternative to cloud centric application which they named fog computing. The architecture they proposed uses the edge of the network to alleviate cloud limitation by carrying out some of the processing tasks on the devices themselves. This reduce the cost of using cloud computing and enhance more robust real-time applications. Nevertheless not all the devices are IP capable. In those case, gateways can be used to link the device to the rest of the network. Gateways are minimised computer like Raspberry Pi where the emphasis is put on the connectivity and weather robustness.

III. Crowd monitoring at the edge

Highly congested urban areas present many challenges for city planners. Monitoring crowd movement can help understanding crowd patterns to avoid congested areas and ensure people move in a safe, secure and predictable manner, for example before the game kick-off in a stadium. This has potential to improve quality of experience but also, critically, crowd safety as illustrated by recent deaths in situations of crowd panic or congestion such as Shanghai 2014 new-year eve where 34 people tragically lost their life.

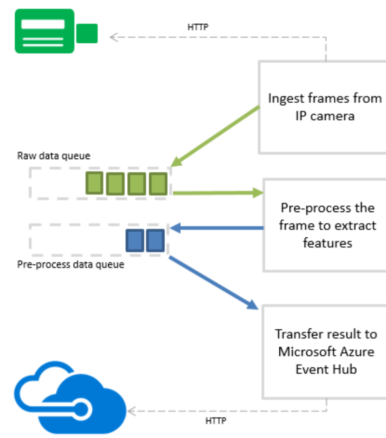

Fig. 2. Software architecture for crowd monitoring

Crowd monitoring uses image processing and the data captured are highly sensitive. Outsourcing storage and processing to the cloud is not suitable. However, when performing crowd analytics only metadata such as the crowd density, activity Or the number of people present matter. The image in itself is not meaningful. Therefore, video processing at the edge improve responsiveness while ensuring better data privacy.

A. Algorithms

To assess the capabilities of real-time video processing at the edge, three algorithms of different complexity are tested. All the algorithms are written in python. For the deep learning model, pre-trained models are used. Table I summarises the different algorithms tested.

Crowd density: The frame are ingested and converted to greyscale to extract the background of the scene and compute interesting point locations corresponding to objects in the scene. The crowd density is estimated by splitting the frame into a grid and computing the percentage of grid cells that are occupied by one or more tracked points. This algorithm has a low complexity.

ResNet50: Object detection is an integral part of image processing and known to be computationally demanding. Residual network architecture has been introduce by He et al. [23] for object recognition. It is among the best performing state-of-the-art deep learning model for image classification. It presents a medium complexity and works with low resolution images.

Crowd Counting: ResnetCrowd was developed by Marsden et al. [24] to achieve crowd counting. It is based on a ResNet18 [23] architecture and uses an heat map approach to estimate the number of people in a scene. High resolution images are divided into patches and fed to the ResnetCrowd to count the number of people. Each individual result is summed to compute total crowd counting. This model achieves state-of-the-art results and presents a high complexity with significant compute requirements.

TABLE I
CROWD MONITORING ALGORITHMS – ALL THE ALGORITHMS ARE IMPLEMENTED IN PYTHON

Name	Description	Algorithm Complexity	Deep Learning	Library	Data size (pixels)
Crowd Density Estimation	Extract background of image to estimate the density of people in the scene	Low	No	Opencv-python, Numpy	1024×728
ResNet50 from Keras	ResNet50 default Keras model application with ImageNet weights	Medium	Yes	Numpy, Keras, Tensorflow, pillow	224x224
Crowd Counting	Crowd counting from Keras ResNet50 model with custom weights	High	Yes	Numpy, Keras, Tensorflow, Opencv-python	1024×728

TABLE II
CPU SPECIFICATION

	Model	Release date	#core (#thread)	CPU	cache	TDP
Laptop	Intel Core i5-3210M	Q2 2012	2 (4)	2.50 GHz	3MB SmartCache	35W
Gateway	Intel Atom E3825	Q4 2013	2 (2)	1.33 GHz	1MB L2	6W

B. Data

The IP camera was emulated with a Python RESTful web service sending images every second using the HTTP protocol. For the Crowd Density estimation and Crowd counting algorithms, the data consisted of 10 images with high resolution (1024×728 pixels) extracted from the Shanghaitech Part_B dataset [25]. The Shanghai street scenes containing between 23 people and 476 people. For the ResNet50 model, 10 images from the ImageNet dataset [26] with low resolution (224×224 pixels) were used.

C. Technologies

1) Hardware: The performance of each algorithm was tested on two different machines, a laptop and DELL Edge gateway 5000 with respectively an Intel `I5-3210M` (medium) and Intel `Atom E3825` (low) CPU. The main differences between the two CPUs are highlighted in Table II. We intentionally choose a i5 CPU to have a fair point of comparison with the Atom CPU. There is no doubt that an i7 or a Xeon CPU would have outperform the E3825 by far. The gateway runs with Ubuntu Core 16.04 OS.

2) Software: The software was implemented in Python using a multi-threaded approach to ingest raw images from the emulated IP camera, pre-process the frame and transfer the result to the Event Hub service in Azure as shown in figure 1. The runtime was monitor with the `timeit` library. Docker was used to deploy the solution on the gateway. The software was run multiple times on the laptop to ensure consistency before to be deploy on the gateway.

D. Results

1) CPUs performance comparison: The results are presented in Table III. The crowd density estimation algorithm performs equally well on a middle and low CPU; the algorithm is not CPU-bound. Basic image pre-processing can be done at the edge to ease the cloud workflow. However when the complexity of the algorithm increase there is a significant difference between the two CPUs.

When the complexity of the algorithm is high but input data has low resolution (ResNet50), the performance is better on the middle CPU but the runtime on the gateway is acceptable (3 seconds per frame). But when the complexity of the algorithm is high and the image resolution is high (ResnetCrowd), the low CPU lacks of computational power. In both case, increasing the complexity have a runtime impact around 5 times worst on the E3825 than the i5-3210M CPU. This can be explained by the CPU architecture.

With only two cores the Atom E3825 is not capable of real multi-threading. Moreover, adjacent L2 and L3 memory size is larger in the i5-3210M which results in a better support of high performance computing. Note that I/O-bound algorithms do not require high performance computing; therefore the difference in runtime for the crowd density is explained by the dimensions of the input image. With low resolution images, the crowd density algorithm performs equally well on both CPUs. It is also important to note that we observed a 2 minute average delay between the ingestion of the data and the the Microsoft Azure cloud reception due to the choice of the HTTP protocol known to be slower than MQTT.

2) Downscaling input data: The size of the input image has an important impact on the performance. To push further, we investigated if a good trade-off between the model accuracy and runtime could be found by rescaling the input frame for the crowd counting algorithm. The images were rescaled by a factor of 5 from 100% (full-size image) up to 25% size image. The `cv2.resize()` function was used to shrink the frame with the `INTER-AREA` interpolation parameter. Figure 3 shows the different accuracy of the model using the mean square error (MSE) and mean absolute error (MAE) of the miscount number of people obtained while figure 4 pictures the speed improvement when rescaling the image.

Both the speed and accuracy presents a linear relationship with the size of the input. However, reducing the input size

TABLE III
CPU PERFORMANCE COMPARISON

	Laptop		Gateway	
Algorithm	Frames per second	Seconds per frame	Frames per second	Seconds per frame
Crowd Density	12.5	0.08	6.7	0.15
ResNet50	2	0.5	0.25	4
Crowd Counting	0.02	44	0.005	219

can have a positive impact on the accuracy. The 90% image is an interesting result. It performs 31% better than the full-size image (MSE of 36 compare to 53 and MAE of 25 compare to 31) with a 22% time decrease (167 s/frame instead of 219 s/frame).

A good trade-off can be found. The 80% image is a good candidate with a 7% decrease in accuracy and 31% decrease in runtime. Nevertheless it is still taking two and a half minutes (150 seconds) to process one frame.

IV. Discussion

Video processing at the edge has already been integrated in several smart city applications such as smart lights systems [10]. It presents many advantages mainly regarding application responsiveness and data storage and privacy. Edge computing has an important role to play for a sustainable data management. For many years It was put aside because of its computational limitation but with the progress of technology it is now possible to enhance advanced processing at the edge of an IoT network.

A fine tuning of the data pipeline where the processing is balanced between cloud and edge computing, is a key element for smart applications and several parameters need to be taken into account. When designing a solution, real-time requirement and the sensitivity of the data should be carefully considered to choose the right trade-off.

For example, from a crowd monitoring perspective most of the CCTV cameras capture only one image per second. When computing the crowd density or the crowd counting, real-time processing (approximately 25 frames per second) is not crucial. Extracting those features every half minute is sufficient. However, other applications like autonomous driving require an extreme responsiveness and therefore the computational limitation of the edge along with network latency is an issue.

But regarding data privacy, video analytics at the edge has many benefits. The raw data captured by CCTVs are highly sensitive. To perform data-driven decision making just a high level representation of those data such as the density of the crowd or the activity of the crowd is needed. Computing those informations at the edge reduced the cost of using cloud services and ensure a better anonymity of the data.

V. Conclusion

In the early stage, smart city applications were relying on a cloud centric approach. This design has shown its limitations (high costs in terms of bandwidth, latency and energy consumption) and a new paradigm has emerged where data processing is balanced between the cloud and the edge.

In this paper we have explored video analytics at the edge using a Dell Edge Gateway 5000. Basic image processing which are not CPU-bound are performing equally good on a low and middle range CPU devices. They can process more than one frame per second where most CCTVs only capture one image per second which make them suitable for near real-time crowd monitoring. However, with the algorithm complexity increasing, the size of the frame highly impacts the performance. It can increase by a factor of 5 the processing time at the edge. Carefully choosing the input size is important as a good trade-off between the model accuracy and the runtime performance can be found saving up to 30% time.

In the future work, the crowd monitoring solution will be stressed out in Croke Park. We will also explored further rescaling methods to fasten deep learning processing. Other hardware solutions exist to enhance computing at the edge.

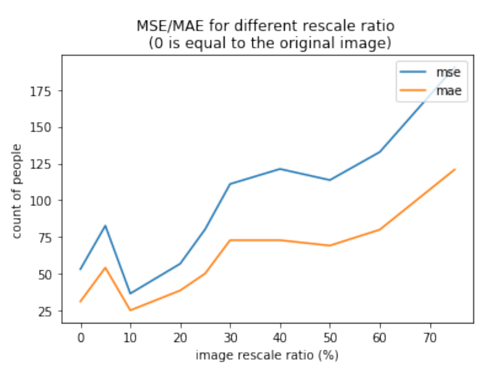

Fig. 3. MSE and MAE

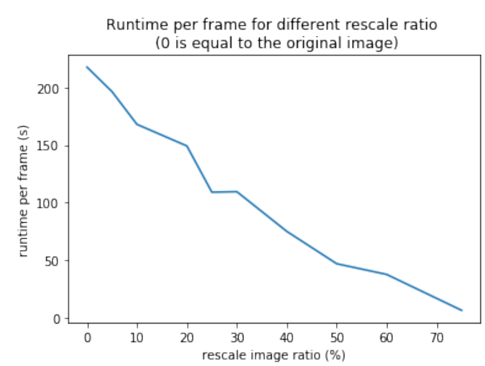

Fig. 4. Runtime

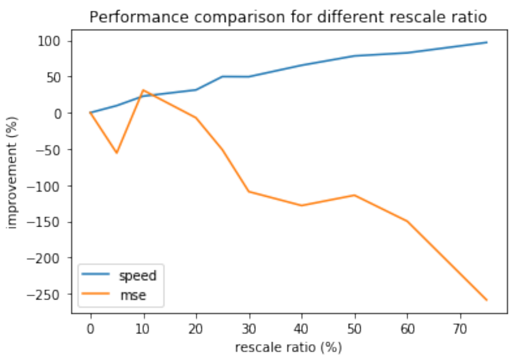

Fig. 5. Trade-off speed vs accuracy

Movidius, an Intel company, has released a computational USB stick to provide a computer vision solution for low CPU devices.

Processing video at the edge enhance real-time analytics as well as better data-privacy; sensitive images can be anonymised at capture point. In this paper we have reviewed the capacity of edge video analytics. We have shown that video analytics can be performed but the application responsiveness highly depends of the input data size. Rescaling video frames is a good way of reducing the processing time for complex algorithms. Therefore to balance processing between the cloud and the edge some parameters such as the responsiveness of the application desire and sensitivity degree of data need to be taken into account.

Acknowledgment

The authors would like to thank Intel Ireland for providing the gateway device. This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number 12/RC/2289 and 16/SP/3804 (Insight Centre for Data Analytics).

References

[1] United Nations. Department of Economic and Social Affairs. Population Division., *The World's Cities in 2016 : Data Booklet.* [New York NY?]: United Nations, Department of Economic and Social Affairs, Population Division, 2016.

[2] S. Pellicer, G. Santa, A. L. Bleda, R. Maestre, A. J. Jara, and A. G. Skarmeta, "A Global Perspective of Smart Cities: A Survey," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 7 2013, pp. 439–444. [Online]. Available: http://ieeexplore.ieee.org/document/6603712/

[3] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog Computing for Sustainable Smart Cities: A Survey," *ACM Computing Surveys*, vol. pp, no. pp, p. pp, 2017. [Online]. Available: http://arxiv.org/abs/1703.07079

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 9 2013.

[5] A. Markkanen, "Competitive Edge from Edge Intelligence IoT Analytics Today and in 2020 IoT Analytics Today and in 2020," 2015.

[6] Y. Bengio, "Learning Deep Architectures for AI," *Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks."

[8] L. Wang and D. Sng, "Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey," 12 2015. [Online]. Available: http://arxiv.org/abs/1512.03131

[9] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "ResnetCrowd: A Residual Deep Learning Architecture for Crowd Counting, Violent Behaviour Detection and Crowd Density Level Classification," 5 2017. [Online]. Available: http://arxiv.org/abs/1705.10698

[10] P. C. Veena, P. Tharakan, H. Haridas, K. Ramya, R. Joju, and T. S. Jyothis, "Smart street light system based on image processing," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 3 2016, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7530216/

[11] J. Aryal and R. Dutta, "Smart city and geospatiality: Hobart deeply learned," in *2015 31st IEEE International Conference on Data Engineering Workshops*. IEEE, 4 2015, pp. 108–109. [Online]. Available: http://ieeexplore.ieee.org/document/7129557/

[12] R. Boston, "IBM's Smarter Cities Challenge."

[13] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," *Studies in Computational Intelligence*, 2014.

[14] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, "Smart Cities: A Survey on Data Management, Security and Enabling Technologies," *IEEE Communications Surveys & Tutorials*, vol. X, no. X, pp. 1–1, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8003273/

[15] K. A. Alam, R. Ahmad, and K. Ko, "Enabling Far-Edge Analytics: Performance Profiling of Frequent Pattern Mining Algorithms," *IEEE Access*, vol. 5, pp. 8236–8249, 2017.

[16] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing," in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 12 2014, pp. 325–329. [Online]. Available: http://ieeexplore.ieee.org/document/7033259/

[17] A. Botta, W. De Donato, V. Persico, and A. Pescap??, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[18] "REGULATION (EU) 2016/ 679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL - of 27 April 2016 - on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/ 46/ EC (General Data Protection Regulation)."

[19] S. Verma, Y. Kawamoto, Z. Fadlullah, H. Nishiyama, and N. Kato, "A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues," *IEEE Communications Surveys & Tutorials*, no. c, pp. 1–1, 2017.

[20] N. Moran, D. Prendergast, D. Zhang, and S. Little, "Iot microcosm: Irelands croke park smart stadium paves the way for smarter cities," https://customers.microsoft.com/en-us/story/gaelic-athletic-association-croke-park-it-technology-azure-cloud-windows10, Nov 2016.

[21] S. Little, D. Zhang, C. Ballas, N. E. O 'connor, D. Prendergast, K. Nolan, B. Quinn, N. Moran, M. Myers, C. Dillon, and T. Meehan, "Understanding packet loss for sound monitoring in a smart stadium IoT testbed."

[22] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things." [Online]. Available: http://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 12 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[24] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Fully Convolutional Crowd Counting On Highly Congested Scenes," 12 2016. [Online]. Available: http://arxiv.org/abs/1612.00220

[25] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2016, pp. 589–597.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," 9 2014. [Online]. Available: http://arxiv.org/abs/1409.0575