Towards Edge Intelligence in Smart Spaces

Universidade Fernando Pessoa

Bruno Gomes

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

A thesis submitted for the degree of

Master of Science

2020

Abstract

After more than two decades of existence, the internet of things has been revolutionizing the way we interact with the world around us. Although, in its origins, the adoption of a cloud computing paradigm supported this ubiquitous computing model, the increasing complexity of IoT systems has led to the gradual fading of the traditional hierarchical model of cloud computing. The search for solutions to the problems of latency, scalability and privacy has, in recent years, driven the movement of data processing and storage, from the cloud, to the edge of the network (edge computing). Starting from the particular case of edge computing that keeps the focus on extending the boundaries of artificial intelligence to the edge of the network - Edge intelligence - a survey of the current state of the art is carried out, culminating into the specification of an architecture to support edge intelligence applications. In order to validate the proposed architecture, two scenarios are presented.

In the scope of waste management and energy recycling, a system for used cooking oil classification in a national domestic collection network is presented. With the local classification of the trustworthiness of each deposit, it was possible to significantly shorten the response times, with a direct impact on energy consumption levels.

Aimed at smart cities, a second application scenario, proposes an approach based on computer vision and deep learning, for local detection of pedestrians on crosswalks. In this context, an edge intelligence paradigm allowed to overcome privacy related issues, as well as reducing response times by more than 80 times, when compared to a cloud computing based solution.

Abstract

Após mais de duas décadas de existência, a internet das coisas, tem vindo a revolucionar a forma como interagimos com o mundo que nos rodeia. Apesar de, nas suas origens, a adoção de um paradigma de computação em nuvem ter servido de suporte a este modelo de computação ubíqua, a crescente complexidade dos sistemas IoT tem conduzido ao paulatino esvanecer do tradicional modelo hierárquico da computação em nuvem. A procura por soluções para os problemas de latência, escalabilidade e garantia de qualidade de serviço tem, nos últimos anos, impulsionado a deslocação do processamento e armazenamento de dados, da nuvem, para a periferia da rede (computação periférica). Partindo do caso particular de computação periférica que mantém o foco no alargar das fronteiras da inteligência artificial para a periferia da rede - Periferia inteligente - um levantamento do atual estado da arte é levado a cabo, culminando na especificação de uma arquitetura de suporte a cenários de periferia inteligente. Com vista à validação da arquitetura proposta, dois cenários são apresentados.

No âmbito da gestão de resíduos e reciclagem energética, um sistema para classificação de óleo alimentar usado, numa rede nacional de recolha doméstica é apresentado. Com classificação local da veracidade de cada depósito foi possível encurtar significativamente os tempos de resposta, com impacto direto nos níveis de consumo energético.

Direcionado às cidades inteligentes, um segundo cenário de aplicação, propõe uma abordagem baseada em visão computacional e aprendizagem profunda, para deteção local de peões em passadeiras. Neste contexto, um paradigma de periferia inteligente permitiu ultrapassar questões relativas à privacidade na transmissão de dados, assim como reduzir em mais de 80 vezes os tempos de resposta, quando comparado com uma solução de computação em nuvem.

I would like to dedicate this dissertation to my parents, family and girlfriend,
for their endless love and encouragement

Acknowledgements

I would like to acknowledge my supervisors Prof. José Torres, Ph.D and Prof. Pedro Sobral, Ph.D, whose help and advice proved to be essential in guiding me through the course of this work. Also to be acknowledged is the relevant role played by the remaining teachers of the computer engineering course, by providing me the knowledge and inspiration to make possible this dissertation.

To my family, friends and colleagues for their support and motivation.

To Hardlevel, namely Dr. Celio Carvalho, Dr. Karim Karmali and Dr. Salim Karmali for presenting me with challenging and innovative projects.

This work was funded by Project P-FECFP-HARDLEVEL-ISUS0001-2018 supported under the scope of a protocol established between Hardlevel - Energias Renováveis Lda and Fundação Ensino e Cultura Fernando Pessoa.

Contents

List of Figures

List of Tables

Acronyms

AI *Artificial Intelligence*

AIoT *Artificial Intelligence of Things*

CNN *Convolutional neural network*

DNN *Deep neural network*

ECG *Electrocardiogram*

EIASS *Edge Intelligence Architecture for Smart Spaces*

edge AI *Edge Artificial Intelligence*

FPGA *Field-programmable Gate Array*

GDPR *General Data Protection Regulation*

GPU *Graphics Processing Unit*

IoT *Internet of Things*

IR *Infrared*

KPU Nueral network processor

NFC *Near Field Communication*

OTA *Over the air*

PIR *Passive Infrared*

QoS *Quality of Service*

R-CNN *Region - Convolutional Neural Network*

RFID *Radio-Frequency Identification*

ROI Region of Interest

SoC *System on a Chip*

SVM *Support Vector Machine*

ToF Time of Flight

UML *Unified Modeling Language*

WAN *Wide Area Network*

XML *Extensible Markup Language*

YOLO *You Only Look Once*

Chapter 1

Introduction

Creating a network of connected, smart physical objects, was, from the beginning, the goal of the *Internet of Things* (Kranenburg and Bassi, 2012). By defining a concept, rather than a technology, nearly 20 years after its first appearance, the generic definition of IoT is still evolving, with an ever-growing number of technological branches converging into the term IoT. From wireless networks, to real-time analytics, many knowledge areas have been shaping the *Internet of Things*, while directly contributing to its exponential growth.

Even without noticing, our daily lives are already driven by the technological advances in smart connected devices. For instance, by simply visiting a museum, a person can be surrounded by a multitude of IoT solutions/products. The navigation system of his/her smartphone guides the person to the museum, an RFID/NFC card enables the contact-less payment, proximity sensors enrich the user experience by providing contextual information for the various exposed items, while a smart security system constantly monitors visitors for any kind of suspicious behaviour. Although its ability to fade into ambient objects somehow disguises the omnipresence of IoT, when looking at the 75 billion connected devices expected by (Statista, 2015) by the year 2025, the prevalence of IoT is made clear.

Although presenting interesting, convenient solutions for citizens and organisations, this rapid IoT growth is challenging pre existing infrastructures and regulations. From the extra network load generated by the communication of billions of devices, to the strict data privacy regulations, difficult to met in a resource constrained environment, IoT still needs to overcome many obstacles to continue its evolution. In (Kranenburg and Bassi, 2012) the authors compiled a list of the challenges faced by IoT, from which we highlight:

- Energy management - Emerging as a major technological challenge, research should be conducted on energy harvesting solutions, as well as minimising the used energy during operation;
- Scalability - With the increasing number of active connected devices, IoT can vastly outnumber the magnitude of the current Internet, therefore raising concerns regard-

ing the scalability of the standard hierarchical cloud computing paradigm for the *Internet of Things*;

- Security and Privacy - Given the low availability of resources, most of the times, advanced security techniques are not implemented by IoT devices. An approach where the computing architecture serves as the basis for the security and privacy is suggested by the authors;

- Communication - Communication is still one of the main technological challenges facing IoT. The lack of a silver bullet on a trade-off between coverage, data rate and energy consumption is still impacting communication dependent solutions.

Although disjoint from one another, the computing paradigm emerges as the common ground between the four challenges. Therefore, a disruptive change in the computing paradigm could potentially contribute to surpass all the above mentioned problems faced by IoT systems.

With most of the energy usage occurring on short bursts during communication (Gomes et al., 2020), an architecture where the device locally processes the collected data can reduce the time (and consequently, the energy) spent on communication. The scalability concerns raised by billions of connected devices, can also be mitigated by relaxing the end-to-end concept of relay all the complexity to the cloud layer (Kranenburg and Bassi, 2012), leveraging on local or nearby resources to process data where it is collected. With most of the security / privacy concerns relating to data transmission and data storage, the on-site data processing bypasses the need for complex security techniques, as well as the single point of failure presented by the communication link.

The need for offloading processing from the cloud has lead to the emergence of alternative computing paradigms, focusing on load balancing IoT systems towards the edge of the network. A comparison between cloud, fog and edge computing is laid out in chapter 2, resulting in the adoption of an edge computing paradigm to be applied in two distinct application scenarios:

A waste management system consisting on a public network of **smart cooking oil collection units**, aiming to locally classify the trustworthiness of each oil disposal. The collection unit should maintain its capabilities even if installed on remote areas with weak or no network coverage.

A **smart crosswalk** platform appears as a second application scenario. In this system, an end node should notify approaching vehicles that a pedestrian is crossing, or about to cross, the street. The detection should rely on computer vision capabilities, while ensuring data privacy and a low network load.

Considering the similarities between the two application scenarios, as well as the possibility to scale the same solutions to future projects, a generic, edge computing enabled, network architecture is specified. By leveraging on the built-in privacy and scalability, brought by the adoption of the architecture presented at 3.1, the focus could move towards the application logic.

The high complexity in manually modelling a solution for both the predictive analysis of oil disposals and the object (human) detection required by the second application scenario, justify for the use of artificial intelligence techniques in both contexts. Although from a layered network architecture stand point, artificial intelligence algorithms have always tended to be executed in the cloud (Zhou et al., 2019), a new branch of edge computing is starting to explore the possibilities of bringing artificial intelligence capabilities to resource constrained devices - edge AI / Edge Intelligence. Adopting a level 3 on the edge intelligence chart specified in (Zhou et al., 2019), both application scenarios should rely on cloud trained models to support their local classification/detection, therefore combining the inherited benefits of an edge computing paradigm, with the simplicity of a centralised cloud training.

1.1 Restrictions

Even considering the high applicability of computing paradigms across a wide variety of applications, this dissertation keeps the computing paradigm discussion always in the IoT scope. Besides this assumption, every aspect described during the following chapters assumes four pre-established project restrictions:

- Energy availability: every edge device must be battery powered, with a emphasis being put into reducing its energy footprint to a minimum. When, and if, possible energy harvesting solutions should be applied in order to achieve a self sustainable energy source.
- Unitary cost: since the solution must scale to other different scenarios, the unitary cost will always be considered when choosing between approaches and technologies.
- Data security: sensitive data should always be protected when stored, transferred or processed, in agreement with the European data protection rules (Consulting, 2020).
- Network connection: some edge units may be installed in remote areas where a scenario with no network coverage must be considered.

1.2 Contributions

Being a totally new concept for the *Internet of Things*, edge intelligence is still under strong conceptual discussion. Given the incipient development state on this computing approach, few representative solutions showcase concrete implementations of an edge intelligence paradigm. Thus, with the goal of taking a part on this movement of more complex computation tasks to the lower layers of the network an, edge AI focused, network architecture is presented. Following a top-down approach to the architecture specification, every network layer and communication channel are carefully bounded and described with two application scenarios evaluating the merits of the specified architecture. Annotated datasets will be left publicly available, which, in conjunction with the UML documentation presented in each of the application scenarios, should ease the replication of the works proposed in this dissertation. Therefore, three main contributions can be distinguished:

- Specification of a complete network architecture to cope with the requirements of edge intelligence applications
- Implementation of two distinct application implementing an edge AI paradigm
- Elaboration of a public crosswalk dataset, containing more than 200 labelled pedestrian images

1.3 Document Structure

This document is subdivided in six different chapters: Introduction, State of the art, Architecture specification, Application Scenarios, System evaluation and, at last, Conclusions and future work. The first chapter lays out the dissertation subject, along with the main motivations for developing this work and the description of the problem to be solved. The following chapter carries out an analysis of the related work on edge computing applied to the *Edge Artificial Intelligence* scenario. Chapter three focus on the specification of a generic edge computing architecture to cope with the strict requirements of edge AI applications. A theoretical and practical approach of two application scenarios is presented on the fourth and fifth chapters. Also, in chapter five, the system implementation culminates on its evaluation. The sixth and last chapter concludes this dissertation, leaving opened the possibilities for further developments on the edge intelligence subject.

Chapter 2

State of the art on edge intelligence

2.1 Cloud, Fog and Edge computing - A comprehensive overview

With its roots on nearly sixty years of history (Verma and Katti, 2014) and being defined by the US National Institute of Standards & Technology (Mell and Grance, 2011) as:

> A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing, driven by the exponential popularity growth brought by services like AWS or Google, in early 2000's, as become a standard on where to store and process data. Given its working principles, cloud computing appears as a democratised service, closing the technological gap between large and small organisations (Shaw and Singh, 2014). Besides cost reduction and greater simplicity (from the client stand-point), also scalability, reliability and efficiency are some of the other known benefits (Shaw and Singh, 2014). But, while cloud computing still presents the best solution for many applications, recent technological trends are leading the way for the appearance of new computing paradigms.

In 2015, Cisco introduced the concept of fog computing, as an extension to the classic cloud computing. Aiming to surpass the problems of latency and bandwidth aroused with the growing needs of IoT systems, the fog specifies an area close to the data sources, in which an heterogeneous network of devices is left in charge of locally produce and act on field generated data (Cisco, 2015). By carrying out an in-depth analysis on fog computing applied to the *Internet of Things*, (Bellavista et al., 2019) gathers the list of requirements that fog computing must fulfil. With inspiration in IoT applications with strict demands,

not easily achieved by the standard cloud computing paradigm, the authors suggest fog as the solution for a series of requirements, from which we highlight:

- Scalability, by being able to manage highly distributed systems with an ever growing number of devices and data generated;
- Real-time responsiveness, obtained by moving processing closer to the edge (less network hops);
- Data quality, by performing operations like data filtering, data aggregation and data normalisation, thus ensuring that only "good data" reaches the cloud;
- Location awareness, by being able to identify the location of deployment and act accordingly to the surrounding environment.

As fog computing started to gain traction in the research community, a new paradigm began to take form in an layered computing architecture - Edge computing (applied to the *Internet of Things*). Inheriting the 20 year old, original key principals proposed by (Akamai, 2020), edge computing has recently found in the IoT world a relevant position, with (Shi and Dustdar, 2016) defining it as:

> The enabling technologies that allow computation to be performed at the network edge so that computing happens near data sources. (...) In edge computing, the end device not only consumes data but also produces data. And at the network edge, devices not only request services and information from the cloud but also handle computing tasks—including processing, storage, caching, and load balancing.

Therefore, by taking on the premise of bringing data processing closer to the field, edge computing, applied to the *Internet of Things*, can not only inherit the above mention fog benefits, but also conduct them to an even higher level, by the cost of a more complex network edge.

To sum up, table 2.1 presents a qualitative comparison between the three computational paradigms based on the the study presented at (Al-Qamash et al., 2018).

Worth noting that, while exclusively comparing the same non-functional requirements as the ones displayed at (Al-Qamash et al., 2018), other less tangible features need to be considered. For instance, given its highly distributed nature, the edge might not always present the best/easiest layer to deploy business logic. Thus, instead of searching for an absolute answer on the best approach, the focus is shifted to find the one(s) that better suit a specific scenario.

Table 2.1: Non functional comparison on Cloud vs Fog vs Edge computing (Al-Qamash et al., 2018)

Criteria	Cloud Computing	Fog Computing	Edge Computing
Scalability	+	+	+
Interoperability	+-	+	++
Mobility	-	+	++
Heterogeneity	-	+	++
Geog. Distribution	-	+	++
Location Awareness	-	+	++
Performance	+-	+	++
QoS Management	+-	+	++

2.2 Edge Intelligence

In 2019, Gartner, Inc (Gartner, 2019) unveiled their predictions for the technology research trends that could change the technological market in a near future. One of the main selected trends was the move of artificial intelligence to the network's edge in the next two to five years.

The recent interest in edge intelligence made for the development of relevant solutions for AI processing at the edge. However "the edge" does not define an homogeneous environment and/or device but rather an area close to the data sources (Jie Cao, 2018). Thus, most of the already known solutions do not cope with the strict power requirements of a scenario, in which the data processing occurs right where its collected - On-device Intelligence.

To simplify the scientific review, this section is divided in two distinct subsections:

- The first subsection (2.3) focus on state-of-the-art developments on edge computing applied to the edge AI scenario.
- The second subsection (2.4) takes on the analysis of the related work regarding concrete implementations of edge AI based systems.

2.3 Edge Computing applied to Edge AI

With the motto of taking demanding computing tasks, usually performed at the cloud, to the leaves of the network, edge intelligence emerged as a way to process data at the

edge, relying on artificial intelligence algorithms (Plastiras et al., 2018). In (Zhou et al., 2019) a conceptual overview of edge AI is performed. In a standardisation effort, an edge intelligence classification chart is laid out, focusing on distinguishing between the different levels of load balancing the train and inference tasks 2.1.

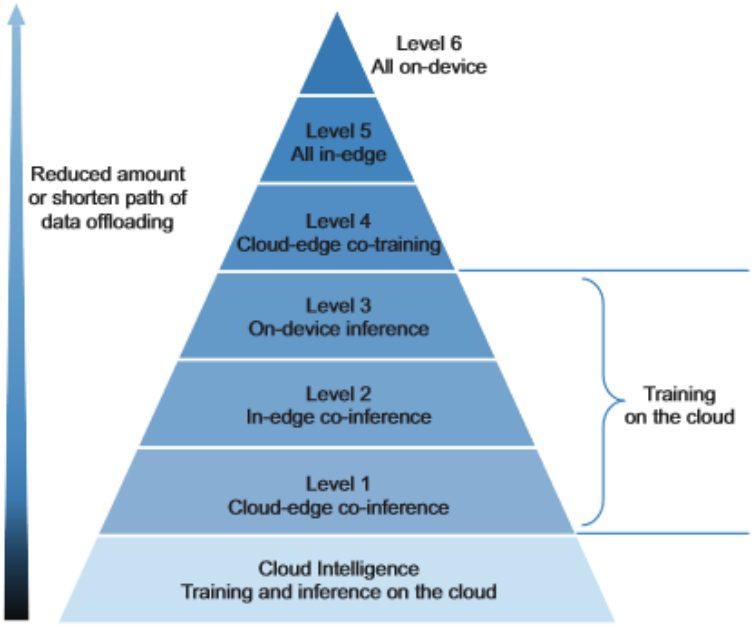


Figure 2.1: edge AI rating. As presented in (Zhou et al., 2019)

In (Calo et al., 2017) the authors described an architecture aiming to solve the problems of network latency, data volume and load balancing. In this approach, as opposed to the standard cloud computing architecture, the complex task of classifying or detecting objects on a captured frame is accomplished by the edge device. The authors highlight examples mainly applied to drone operation, where a drone searches for people to rescue on natural disasters, or for specific weeds on which to spray a weed-killer in a smart agriculture context, but no concrete implementation of any of the above mentioned solutions was evaluated.

Combining the concepts of *Internet of Things*, with the movement of artificial intelligence to the leaves of the network brought researchers to established a new branch on IoT systems - *Artificial Intelligence of Things* (AIoT). (Loh, 2020) sets the scene on the future of IoT, while exploring the applications and challenges associated with running artificial intelligence tasks on power constrained devices. Also from an hardware design point of view, an analysis on the trade-off between power efficiency and programming flexibility for different types of processors is laid out.

An in-depth analysis on the convergence of edge computing and deep learning is presented in (Wang et al., 2020). Due to the same heterogeneity referred in (Jie Cao, 2018), the blurred boundaries of edge computing justify for a differentiation between on-device

intelligence and edge intelligence. A comparison chart is presented (figure 2.2), rating on-device, edge and cloud intelligence by six parameters: privacy; latency; diversity; scalability; on-device cost and reliability.

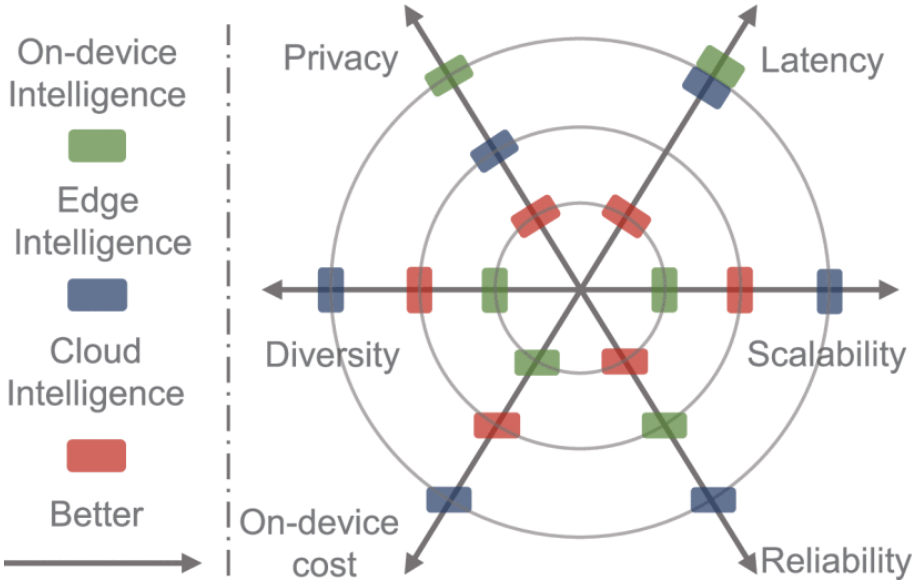


Figure 2.2: On-device vs Edge vs Cloud comparison. As presented in (Wang et al., 2020)

2.4 Related Work on Edge Intelligence

A real time human detector is presented in (Nikouei et al., 2018). The authors explored some lightweight deep neural network models evaluating their results on a raspberry pi 3 platform. Thus, although the image classification is performed totally at the edge, the all-purpose nature of the chosen microprocessor leads to a sub-optimal solution on what regards to power consumption and unitary cost. Also regarding the application of edge AI techniques to surveillance cameras, (Jinguji et al., 2019) suggests an efficient implementation of real-time object detection at the edge. Taking form basis the use of lightweight, CNN based, object detectors, the authors focused on two main ideas: run on FPGA for better performance; split each image into sub-images preventing the detector to discard small objects during image re-sizing for CNN input.

In (Chang et al., 2019) an AIoT solution is proposed. Aiming to mitigate the problem of incorrect use of medication in an ageing world population, the authors present a complete array of systems components (ST-Med-Box) from which our main interest goes to the intelligent medicine recognition device. The on-site object (drug) recognition is GPU processed by an NVIDIA Jetson TX2, splitting the detection into two main tasks: 1) a bounding box is placed around each pill after a Fast R-CNN detection, 2) each pill is then identified by an inception V3 model.

The strict latency requirements imposed by an ECG monitoring system led the authors of (Lin et al., 2019) to present an AIoT solution for real-time analysis of the signal sent by am ECG patch device. Although an efficient implementation (including a solar energy harvesting module) of a wireless ECG patch is introduced, the need to transmit the collected data to a nearby smartphone for classification purposes, makes for a sub-optimal latency value (around 500ms). On a positive note, besides the clever energy management also the mentioned ongoing trials at the National Cheng Kung University Hospital suggest a near production-ready state on the system development.

Table 2.2 lays out a qualitative comparison analysis between the related work described in section 2.4. Three distinct parameters were considered. First, energy efficiency relates to the possibility of battery power the edge device in an uninterrupted work scenario. Secondly, overall capabilities focus on the features presented by the system, as well as its sensing techniques and subsequent accuracy. At last, scalability relates to the possibility of extending the solution to other applications. Here, price, hardware availability and overall response to more demanding scenarios are all classified.

Table 2.2: Comparison of related work on edge AI

Project	**Energy efficiency**	**Overall Capabilities**	**Scalability**
(Nikouei et al., 2018)	+-	+	+-
(Jinguji et al., 2019)	+-	++	+-
(Chang et al., 2019)	-	++	+-
(Lin et al., 2019)	+	+-	+

To conclude, either cloud, fog or edge are nowadays well known and carefully specified computing paradigms, but, while cloud computing already counts with a multitude of tangible projects showcasing its capabilities, edge computing, and even fog to a certain level, are yet to be fully proven. Being edge AI a new branch of edge computing, its related work is still in an incipient state, with a scarce number of projects evaluating edge intelligence enabled solutions.

Chapter 3

Edge Intelligence Architecture for Smart Spaces

A flying drone that performs real-time human detection in natural disasters, a surveillance camera for traffic monitoring that ensures agreement with the most strict data protection rules or a computer vision based crop monitoring solution for quantifying the ripening state of a fruit/vegetable, present some instances of applications were a standard cloud computing paradigm would lead to over-complex solutions to deal with the problems of latency, privacy and scalability. Therefore, embracing the challenge of moving logic and processing closer to the data sources, this chapter takes on the specification of a computing architecture, specifically targeting the branch of edge computing that is pushing artificial intelligent operations towards the borders of the network - edge AI.

3.1 Architecture

After decades of evolution and adaptation to a wide variety of scenarios, cloud computing has shaped the connected world we live in (Shaw and Singh, 2014), presenting applications with a convenient, on-demand access to a shared pool of resources, while guaranteeing minimal management effort (Verma and Katti, 2014). Although not always considered when comparing computing paradigms (Table 2.1), this simplicity in accessing a virtually centralised pool of resources becomes especially relevant on artificial intelligence enabled applications, where a boundary is placed between the training and classification/detection (inference) tasks. As an example, while an edge intelligence paradigm supposes an inference performed at the edge of the network, the training task can still benefit from the client side simplicity brought by a cloud computing paradigm. Just like cloud computing, according to (Al-Qamash et al., 2018), fog computing has also seen its main benefits overshadowed by the movement of data processing towards the edge. While not directly contributing on either the training or inference, an additional fog layer could potentially

take advantage of its privileged position (between cloud and edge layers) to route information between its adjacent layers.

As an extension to the edge intelligence rating proposed by (Zhou et al., 2019) (Figure 2.1), table 3.1 displays the different training, inference and data migration approaches for an edge intelligent application.

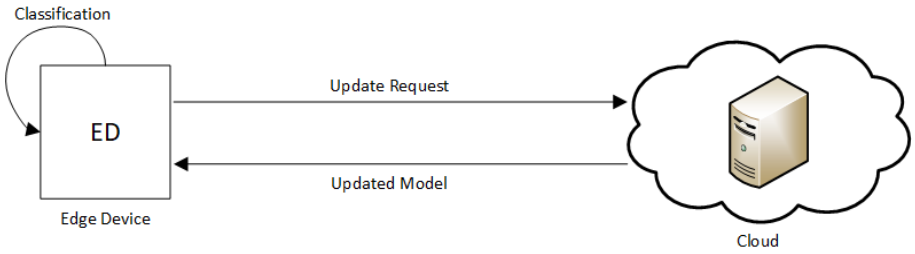
Table 3.1: Edge AI rating

Rating	**Train**	**Inference**	**Model Migration**
(1) Basic Cloud-Edge	Cloud	Edge	N/A
(2) Convergent Cloud-Edge	Cloud	Edge	Standard routing
(3) Convergent Cloud-Fog-Edge	Cloud	Edge	Fog routing 3.1
(4) All Edge	Edge	Edge	N/A

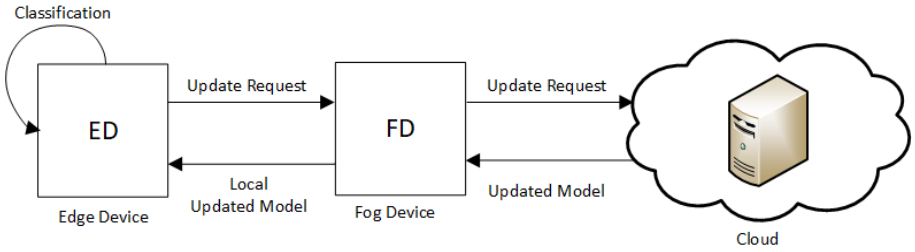
A simple, yet not scalable solution for an edge AI enabled system stands on statically cloud train a model, deploying it to the edge in conjunction with the end-node. Since this rating does not allow for future model improvements it does not comply with the requirements of a generic edge computing architecture. In order to tackle this issue, a convergent Cloud-Edge approach should support the movement of the trained model from the cloud to the edge via standard network routing (Figure 3.1a). Extending the same principles, a convergent cloud-fog-edge rated solution should also rely on a cloud trained model, but instead of forward it to the edge device in an end-to-end logic, a fog device can act as a relay agent, serving its nearby edge nodes with update models. By implementing an application layer model routing, a fog device can operate over the exchanged data, therefore contributing to a more customizable approach (Figure 3.1b). On an extreme edge intelligence scenario, both the train and classification must occur at the edge, not benefiting from any of the cloud/fog capabilities. Given the high specificity of such solution, its adoption in a generic edge architecture gets compromised.

Therefore, aiming to explore the disruptive capabilities brought by an edge computing/edge intelligence paradigm, while still counting on the well known cloud benefits, a convergent cloud-fog-edge architecture was specified - *Edge Intelligence Architecture for Smart Spaces* (EIASS). Figure 3.2 displays an high level view of the three computing layers, as well as the connections between them. From the edge layer point of view, EIASS specifies two communication channels:

> **Fog Channel** - An high bandwidth, low latency, full-duplex channel, aimed at sensitive data exchange and network demanding tasks. In order to assure the effectiveness of the architecture on a wide variety of scenarios, this layer might appear



(a) Standard Routing



(b) Fog Routing

Figure 3.1: Convergent Cloud-Edge vs Convergent Cloud-Fog-Edge

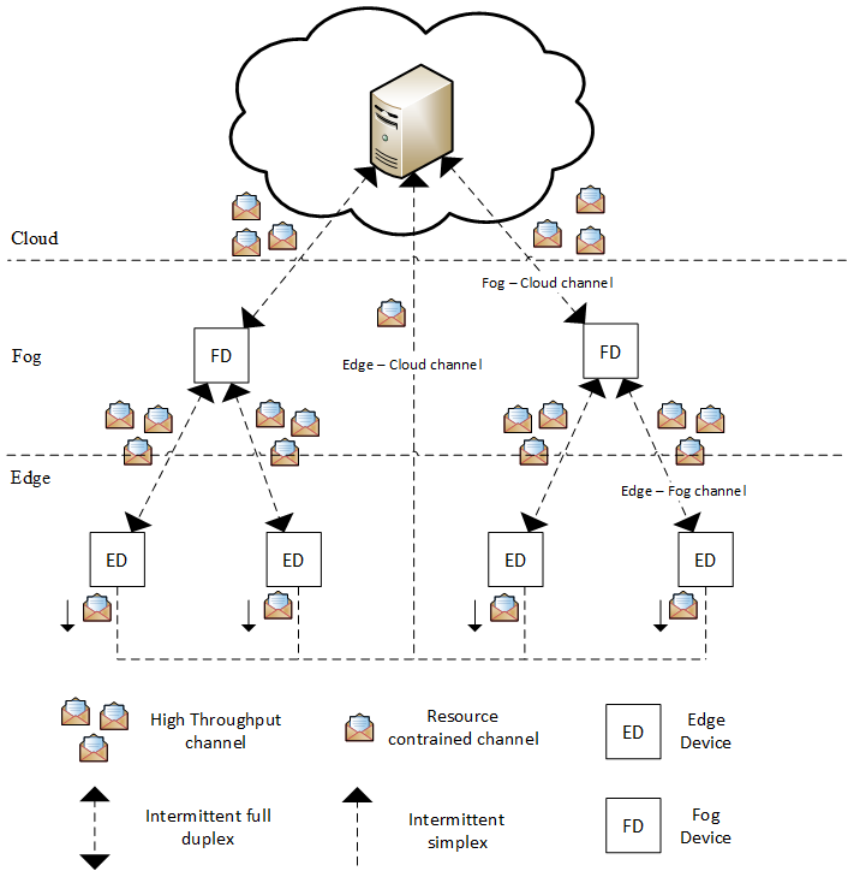


Figure 3.2: EIASS - *Edge Intelligence Architecture for Smart Spaces*

either as a fixed or mobile asset (vehicular fog), enabling a convergent edge classification by receiving updated machine learning models, via downlink channel, while feeding the cloud training script with new, field collected, data.

No concrete device nor communication protocol are specified for this layer/channel, although a microprocessor (Raspberry Pi Zero) implementing WiFi communication was the chosen device for the EIASS evaluation. Worth noting that, while useful in most edge computing/edge intelligence scenarios, EIASS does not pose the fog layer as a requirement.

Cloud Channel - A low bandwidth, low power, long range, one way channel, focusing on simple telemetry exchange. The constrained communication environment, along with the typically high energy usage during data exchange, justify the effort to reduce the messages flowing from the edge to the cloud. Looking at the cloud layer as a service rather than a concrete device, EIASS sees the cloud as a generic pool of resources capable of efficiently perform computing demanding tasks, coping with the definition proposed in (Verma and Katti, 2014).

Relying on its on-site computing capabilities, the edge device should be able to maintain a full working state, unaffected by the variable link quality of any of the above mentioned channels, even in scenarios with weak to no network coverage. Although in its first stages in the IoT realm, this concept of dealing with partial intermittent resources is not entirely new for distributed systems (Dini et al., 2004), with a great example being presented by CODA (Satyanarayanan et al., 1990), a distributed file system with a focus on implementing a resilient solution to network and server failures. Similarly to the disconnected operation proposed in CODA, figure 3.3 displays an example message exchange between the different network layers, showcasing an edge classification based on an, on-device stored, trained model. By design, the edge device should rely on the latest known model for each classification, while opportunistically retrieving updated versions from the cloud (relayed by the fog layer). Also align with the concept of partial intermittent resources, this disconnected operation should happen without any noticeable difference form the end-user stand point (Satyanarayanan et al., 1990).

3.2 Edge AI - Key Benefits

Being a result of the fusion between the paradigms of edge computing and cloud intelligence, most of the key benefits presented by an edge intelligence approach are directly inherited from its roots (Zhou et al., 2019).

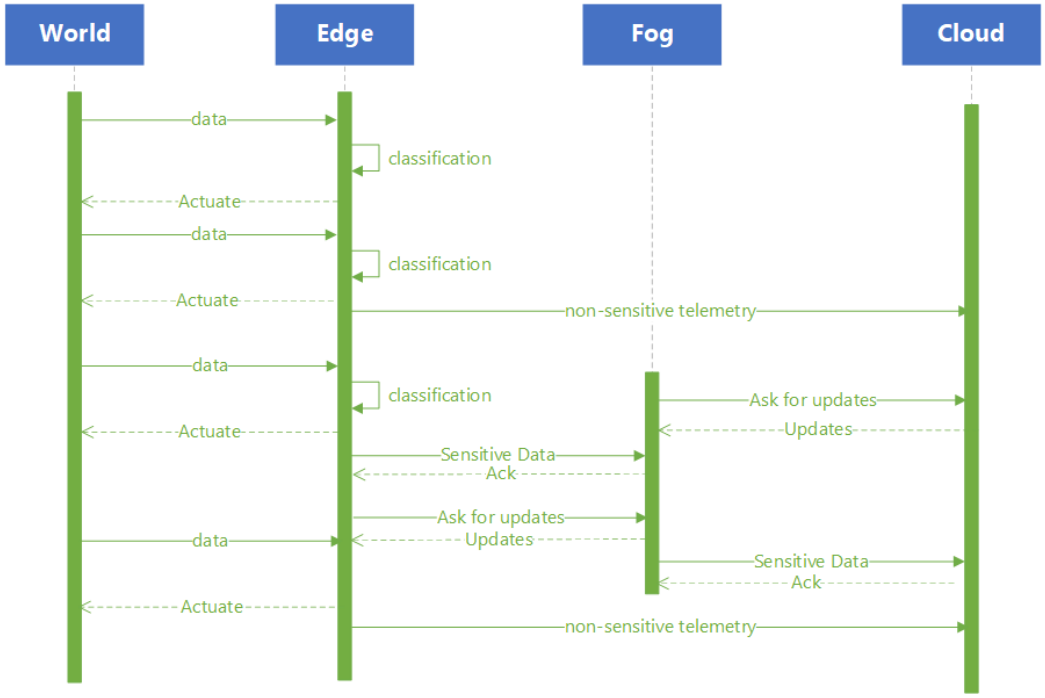


Figure 3.3: Load balancing between network layers

Network Offloading

Every wireless communication technology must share the radio frequency spectrum. Therefore, strict boundaries are defined for each protocol limiting its available bandwidth. As a shared medium, even with the advanced multiplexing techniques, the quality of service cannot be guaranteed, when relying on a best-effort approach and the number of active users exceeds the physical limits of the network.

Given its nature, IoT systems are already challenging the existing network technologies with an ever growing number of active devices. With more than 75 million connected devices expected by the year 2025 (Alam, 2018), the load over network infrastructures might scale faster than the networks themselves. This problem might even be made worse with the increasing complexity of IoT systems. In (Lin et al., 2019), autonomous driving and AI based video surveillance are presented as two applications requiring network offloading techniques - namely, Edge computing - given their capability of generating thousands of GB of data on a single day.

Therefore, edge computing, and specifically edge AI, appears as a way to tackle this issue, not by reducing the total number of active devices, but rather by shrinking exchanged messages and increasing (or bypassing) the required communication intervals.

Data Privacy

Low computational resources and strict power constraints have been negatively influencing the security levels implemented by the vast majority of IoT systems (Kranenburg and Bassi, 2012). This problematic, combined with the recent released European data protection rules, is nowadays affecting IoT applications where personal data can be (directly or indirectly) collected. For instance, on applications relying on computer vision capabilities, there's a strong chance of person capture and consequent recognition. Therefore, each one of the input image samples falls on the definition of personal data proposed in the European *General Data Protection Regulation* (GDPR) (Consulting, 2020).

> 'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;

As it states in article 5, paragraph 1 f of the European GDPR (Consulting, 2020):

> Personal data shall be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures ('integrity and confidentiality').

Even with the standard cloud computing approach coping with most of the personal data processing principles, the typical collect and transmit pattern could raise privacy concerns regarding the security level of the data transmission. This problematic is made even worse in an IoT environment, where most of the long range, low power, wireless networks, do not always implement the state of the art in what regards to communication security (Coman et al., 2019).

Therefore, bypassing the need for sensitive data storage and/or transmission, an edge computing approach cuts the Gordian knot for the security and privacy subjects.

Response Times

"A system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure" is the definition proposed in (Laplante, 1992) for a real-time system. Figure 3.4 lays out the topological differences between a standard cloud computing approach (top) and the same system powered by an EIASS end-node (bottom). By implementing remote data processing, prior to any actuation, data must be sent from

the end node and routed through an unpredictable number of network hops. In contrast, an edge computing approach allows for data processing right where it is collected, with only the edge available processing power influencing the response times.

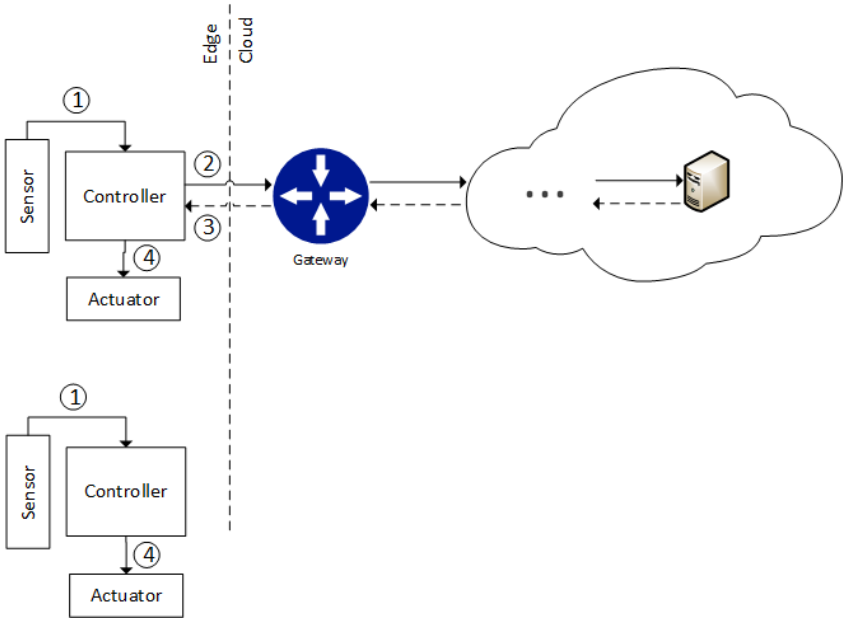


Figure 3.4: From sensing to actuation. Cloud computing (top) versus EIASS (bottom)

In order to systematically compute the cost of local versus remote computing for a given task, (Lin et al., 2019) suggests an approach where the time for local computation (Tlocal) is compared to the total time for computation offloading (TcompOffloading), with an edge computing approach being beneficial when:

$$Tlocal < Tcomm + Tremote <=> Tlocal < TcompOffloading$$

Worth noting that the communication time (Tcomm) should comprehend both the round-trip time and the time required for the communication establishment (highly dependent on the chosen communication technology).

Scalability

As well as the network constraints referred in 3.2, also the available computational resources in a cloud computing environment, will struggle to keep up with the ever-growing demands of modern IoT systems (Kranenburg and Bassi, 2012).

Even with an unlimited, unbounded transmission medium, the costs and complexity of building datacenters capable of processing an estimated 180PB of, IoT generated, data per year (Shi et al., 2016), would make for an unrealistic solution. Therefore, in (Shi et al., 2016) edge computing is proposed as a way to tackle these issues, bringing the processing

closer to the field, hence mitigating the IoT impact over pre-existing infrastructures.

As an example of the relevance of edge computing in an IoT-based society, some of the biggest cloud computing providers are already working on frameworks to offload computing demanding tasks. AWS IoT Greengrass (AWS, 2020) or Azure IoT edge (Azure, 2020) are some examples of this shift to edge computing.

Service availability

With the heterogeneity in the deployment environments raising one of the most common IoT systems concern – network coverage (Kranenburg and Bassi, 2012), the capability to locally process data allows for a resilient system, less prone to fail under network connection losses.

Without conflicting with the connected nature of IoT, EIASS specifies both fog and cloud channels as a delay tolerant communication environment, with the edge layer relying on its local knowledge to transparently maintain its working state. Similar to caching techniques commonly used in portable workstations (Satyanarayanan et al., 1990), the end-node performs accordingly to the knowledge acquired in its last successful communication, while waiting for the next connected operation to update its working parameters. Therefore, powering the future with knowledge from the past, edge computing is able to merge the distributed systems concept of partial intermittent resources into the also distributed *Internet of Things*, culminating into a disruptive approach where, the end-nodes are not just designed to communicate, but also to "decide", exchanging data only in an opportunistic manner.

Unitary Cost

Even considering the subjectivity of a "low-cost" rating, two distinct approaches might be directly compared, price-wise, to one another. In (Wang et al., 2020) an on-device intelligence receives a poor rating, justified by the increased complexity of such device. However, this rating focus mainly on the purchase price of the end-node and less on the running costs, which might not lead to a fair comparison. Moreover, the total cost of deployment not always equals the device cost, since other network components might be required (i.e. gateways) (Mekki et al., 2018).

In fact, most of the times the extra initial cost added by an intelligent edge device can rapidly be mitigated, if taken into consideration the network offloading capabilities of an edge AI approach, whenever the network appears as a paid service (usually more data transferred leads to higher running costs) or as a custom build solution, where the price would also go along with the network capabilities.

Power efficiency

The concept of bringing connectivity and computing capabilities to daily-life objects has, from its origins, raised many technical challenges, with energy efficiency being one the most demanding technological requirements for a successful IoT system (Kranenburg and Bassi, 2012). Aiming at mitigating the energy scarcity effects on a battery powered end node, cleaver hardware and software techniques are already implemented by many IoT systems. For instance, given the, usually high, energy consumption required during communication, power constrained applications tend to implement low power communication technologies (Perles et al., 2018) or keep a focus on reducing the time required for data exchange events (Gomes et al., 2020).

By enabling end nodes to locally store and efficiently process data, an edge computing/edge intelligence approach can significantly contribute for a reduction on the energy requirements, while also allowing for a simpler (smaller) energy harvesting solution.

By leveraging its operations on the locally available resources, an edge intelligence paradigm is able to bypass a multitude of problems typically faced by IoT systems. Therefore, by taking on the specified network architecture (EIASS), as well as its main benefits in an IoT scope, the focus could move to the definition of the application scenarios.

Chapter 4

Application Scenarios

As an emerging technology, edge intelligence is nowadays under strong conceptual discussion. While slowly leading to the desired convergence on the specification of definitions, patterns and boundaries, this theoretical oriented effort is yet to culminate on real-life applications, with only a few systems evaluating true edge AI solutions (2.4). Therefore, aiming to prove the key benefits of the edge intelligence approach presented in section 3.2, two distinct application scenarios where specified:

- Smart cooking oil collection unit - An IoT enabled, public oil disposal bin, capable of locally estimate the chances of a trustful disposal.
- Smart crosswalk - A computer vision based system capable of real time spot pedestrians at or near crosswalks.

4.1 Smart Cooking oil Collection Unit

It is estimated that only around 1.9% of all the used cooking oil is forwarded to recycling, while the remaining 98.1% are left untreated (Botelho et al., 2018). Given that a single litter of cooking oil can pollute up to a million litters of water (Singh et al., 2017) and knowing its virtues as an renewable energy source (Arslan and Ulusoy, 2016), an effort should be put into sensitise citizens for the harmful consequences of incorrect oil disposals.

While most of the used oil collection still relies on standard waste bins spread across a given region, some interesting, IoT enabled, solutions are already making use of sensors and microcontrollers for a collection side improved efficiency. As an example, Hardlevel (Hardlevel, 2020), a Portuguese company in the area of renewable energy sources, is already operating under efficient and environmentally friendly oil collection procedures, relying on a filling level monitoring network to optimise collection routes and understand the disposal habits of a given population.

Although from a logistics stand point, an efficient oil collection task is already reachable, the same cannot be said for the key point of a good cooking oil collection network - user engagement. In fact, the ultimate goal of increasing the amount of used oil that goes towards recycling can only be achieved with a greater number of active users willing to be a part of the recycling process. With the user engagement driving the systems outcome, and considering the already known human response to games serving as motivators (Zichermann and Cunningham, 2011), only one last technical demanding task would be required - disposal classification, ensuring that a positive game feedback is only given to a user after a trustworthy disposal. An on site, lab equivalent, automated oil analysis would probably exceed the complexity levels wanted in a public waste bin, resulting in an overpriced and oversensitive final product. Thus, instead of directly measuring oil properties, the soft classification was performed based on the meta-data acquired upon each deposit, as well as by crossing the density (p) obtained by the ratio of mass (m) over volume (v) with the expected cooking oil density of $0.92 g/cm^3$ (Factbook, 2020).

$$p = \frac{m}{v}$$

In order for the above mentioned ratio to be applied, only standardised oil bottles are accepted by the smart bin. The oil weight is easily attainable by subtracting the known weight of a standardised bottle from the sensed value, while a volume estimation is provided by a, computer vision based, solution (Canny Edge Detector) enabling the detection of discontinuities in a vertical transparent line present on the bottle.

Focusing on the simplification of the authentication task, the standardised bottles are shipped to the user's address upon registration, with an embedded passive RFID tag carrying relevant user and bottle information (e.g. userID, bottleID) Therefore, also the user engagement can benefit from the use of standardised bottles, since no extra step is required prior to an authenticated oil disposal.

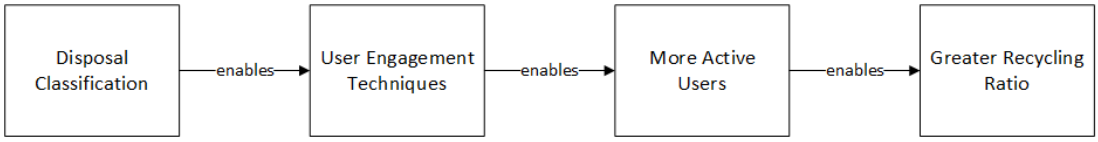


Figure 4.1: User engagement chain

Therefore, by completing the user engagement chain presented in figure 4.1, a solution for domestic cooking oil collection was specified. The collection task occurs on a country level scale with thousands of collection units available to the citizens. The solution described should monitor the trustworthiness of each deposit responding with real time feedback to the citizen. Aligned with the architecture figure 3.2 the system should work as specified even if no network coverage is available relying on its edge computing/edge AI

capabilities to ensure service availability to the end user.

4.1.1 Smart Oil Collection Unit Architecture

With a basis on EIASS, the architecture to support the smart cooking oil collection unit is presented at figure 4.2. In this specific scenario, the fog layer is represented by vehicular network, consisting on moving application layer gateways (collection trucks), capable of store, process and transport data.

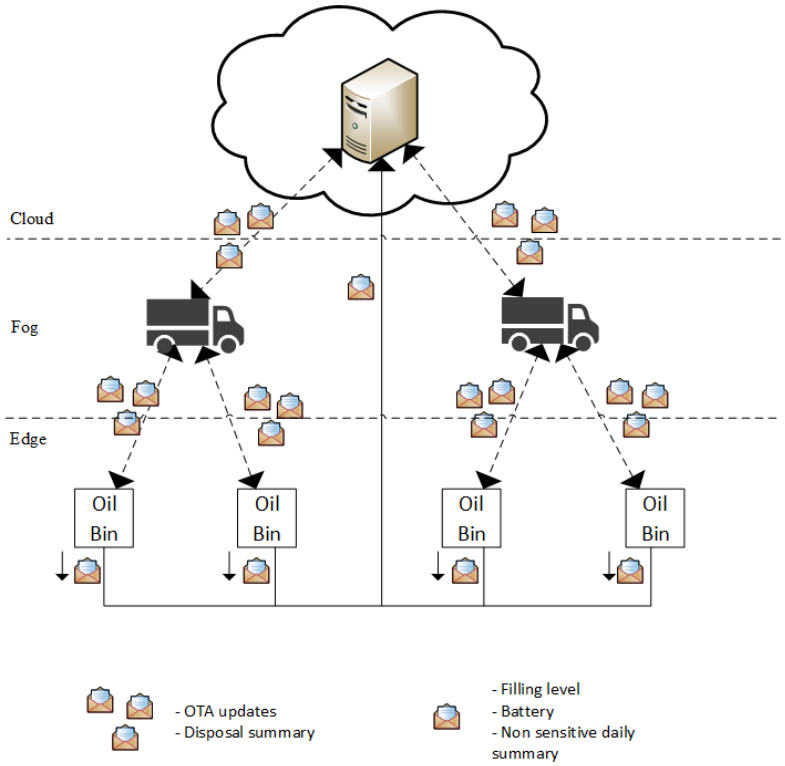


Figure 4.2: EIASS extension for the smart cooking oil collection

Service availability

Specifying a single solution to be fitted in a multitude of scenarios, from rural populations, to city centres, raises questions regarding network coverage. For instance, while city installed collection units might rely on cloud services in order to keep its full capabilities, some countryside installed units cannot depend on an unreliable, and sometimes nonexistent, network services. Thus, by decentralising the main system components from the cloud to the end nodes, an edge computing approach enables the system to bypass the single point of failure represented by the network link.

Figures 4.4 and 4.3 represent, the main architectural difference between a standard smart collection unit 4.4 and an edge intelligence enabled one 4.3. By eliminating the

need for a remote disposal classification, an edge intelligence approach allows for a user feedback to be generated upon each deposit, even in a no network scenario. On the other hand, a cloud running classifier eases the job of continuously improve the classifier output, by being able to train and adapt over centralised data. Consequently, in order to bring the above mentioned advantage to an edge AI approach, a vehicular fog layer, here represented by the oil collection trucks was included. Since these trucks travel frequently from warehouses to the collection points, they open a channel for a delayed tolerant over-the-air update distribution.

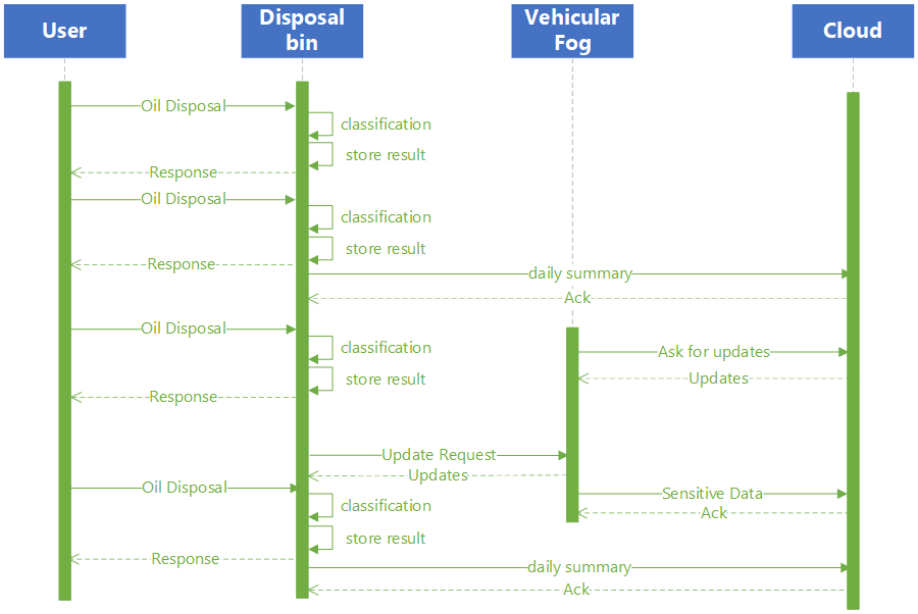


Figure 4.3: Sequence chart of an edge intelligence cooking oil bin

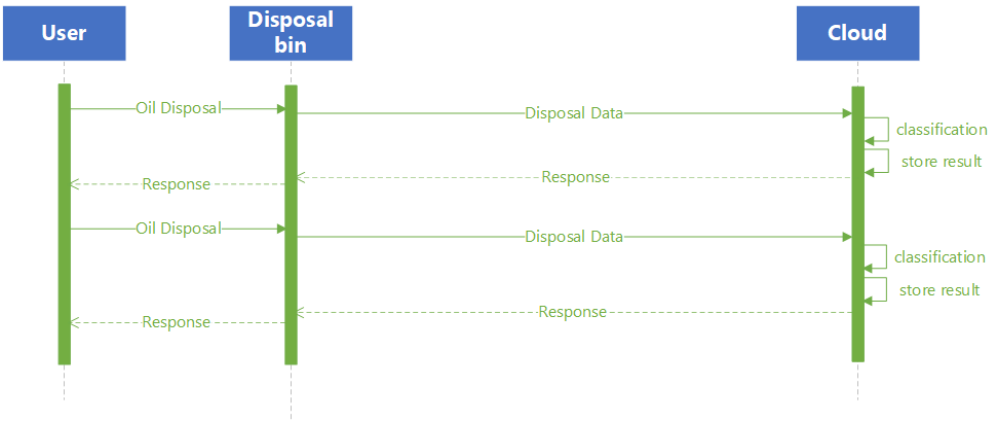


Figure 4.4: Sequence chart of a cloud intelligence cooking oil bin

Response times

Also aligned with the focus on a better user experience, the system response times from the final user perspective could dictate the difference between a failure or a success in whats regards to user engagement.

(Mekki et al., 2019) carries out an analysis on different long range, low power networks. More specifically, along with the already low data rates expected from these technologies, the authors highlight the antagonistic ideas of low power and low latency. Thus, having both energy efficiency and low response times as two of the primary goals of the proposed system, the typical cloud computing approach presented in figure 4.4, would struggle to meet the required energy and latency values for our oil collection unit.

By looking at one of the lowest latency WAN protocols (Mekki et al., 2018) in the IoT realm - NbIoT - and the worst case scenario of 10s latency for an high priority exception report (Matz et al., 2020), a time reference is set for further comparisons with an edge AI approach.

Unitary cost

(Mekki et al., 2018) presents a cost comparison table between SigFox, Lora and NBIoT. As stated in section 3.2, the overhead price of additional network infrastructures largely exceeds the cost of an additional microcontroler aimed at more demanding computational operations. Although not included in the above mentioned table, the choice for a *pay-per-use* subscription plan could also highly benefit from an edge intelligence scenario, as up and down-link messages are reduced in both size and quantity

Power efficiency

Although focusing on the benefits of a fog computing paradigm, the results obtained in (Costa et al., 2020) (Figure 4.5) showcase the behaviour of a cellular based communication and its consequences on a battery powered device.

If, instead of an on-site disposal classification, a cloud intelligence approach were to be considered, similar delay values as the ones presented in figure 4.5 would linearly relate to the number of oil disposals. Therefore, an estimation on the energy efficiency gains can be established by comparing the exact same system, with a conservative assumption of ten disposal per day, and being the computing paradigm the only variable.

- Cloud intelligence

  10 disposals = 10 (for classification) + 1 (daily summary) communications

  11 x 6 = 66 seconds of daily activity

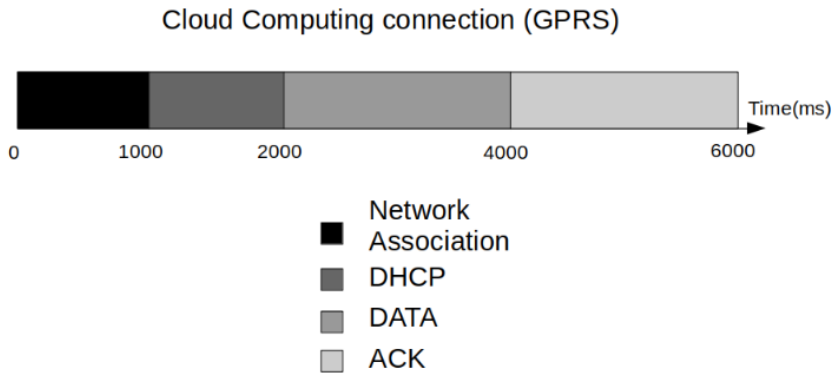


Figure 4.5: Cellular connection delay

- Edge intelligence 10 disposals = 1 (daily summary) communication

  1 x 6 = 6 seconds of daily activity

With a nearly 90% improved communication efficiency an edge AI approach could significantly increase the battery life expectancy, while also allowing the energy demands to be easily surpassed by the integration of solar panels or other energy harvesting solutions.

4.1.2 System Setup

In order to materialise the mentioned benefits brought by the adoption of EIASS, a master-slave architecture was chosen for the edge device. An ESP32 plays the role of master device, while a Sipeed Maix Bit (slave) was left in charge of the, computer vision based, oil level sensing. Although the same results could probably be attainable with just the ESP32 with an external camera, the choice for an embedded deep neural network accelerator ensures a future proof solution for oil classification. Completing the hardware array, a 50kg load cell, combined with an HX711 amplifier, was included, in order to estimate the weight of the disposed bottle, with the fill level sensing being perfomed by a VL53L0X ToF sensor.

GPRS was the chosen protocol for data communication, with a SIM800L being responsible for uploading the daily disposal summary.

4.1.3 Cloud Training

Implementing the convergent edge-fog-cloud rating proposed in EIASS, the training part of the problem was performed by taking advantage of the convenient access of resources, available in a cloud environment. While still lacking a reliable method for local oil clas-

sification, a series of meta-data and indirect disposal properties (table 4.1) are used as features for the classifier training and inference.

Table 4.1: Classifier Features

Feature	Description
User ID	Keep tracking of the user's disposals history
Age	User's registered age
Household	N^{o} of users registered within the same family account
User Points	Given by the n^{o} of registered disposals
Zone	Place of disposal
Distance	Linear distance's to the user's registered address
Time Of Day	Time of disposal
Weight	Measured disposal weight
Volume	Measured disposal volume

Considering the lack of structured data collected from real-life operation of the currently installed waste bins, a small synthetic dataset was created. 250 entries splited in into train and test sets on an approximated 2/3 1/3 ratio emulate an expected user behaviour. Nearly 10 % of the disposals were considered untrustworthy. In order to discover patterns in oil disposals, feature engineering was applied to the synthetic dataset created with the features mentioned at table 4.1. Both weight and volume columns are merged into a synthetic feature - Sensor. This feature points at the evaluation of the sensor readings, implementing the following logic:

Listing 4.1: Classification priorities

```
if sensor == False:
    # invalid disposal
else
    # classify disposal
```

Thus, an invalid sensor parameter directly translates into an untrustworthy disposal, bypassing the need for classification, while allowing for a future-proof solution, able to deal with improved oil sensing techniques.

Given the small dataset size, cross-validation was used during training for a better prediction on how the model would perform when dealing with unseen data. Then the trained classifier was compared against a test set of entries, with the results being shown at section 5.1.

4.1.4 Edge Deployment

The growing interest in bringing artificial intelligence to the edge is leading to the emergence of frameworks like tensorflow lite for microcontrollers (TensorFlow, 2020) or micromlgen (EloquentArduino, 2020). Both frameworks are intended to deploy pre-trained machine learning models to power constrained devices, therefore perfectly suiting an edge intelligence scenario. For a proof-of-concept, micromlgen was chosen to port the trained model to the ESP32 (4.6).

Figure 4.6: Classifier port. From the Cloud to the Edge

Even considering that a built-in neural network accelerator is present at the slave device, in this specific scenario, two main reasons make for a more efficient classification when performed on the master device.

- The hardware accelerator is aimed at efficiently feed-forward convolutional neural networks, being targeted at applications like image, video or audio, therefore limiting its benefits in this specific context.
- The level of simplicity at which the classification occurs is totally aligned with the capabilities of a low power microcrontoller, thus not justifying the extra energy required to keep both master and slave active during classification.

Figure 4.7 displays the set of activities performed upon each oil disposal. A master device is constantly listening (while sleeping) for a disposal event. Once a disposal is detected, the master device proceeds to its classification, involving waking up the slave device for the contact-less level sensing. The disposal iteration ends with the classification result being returned to the user.

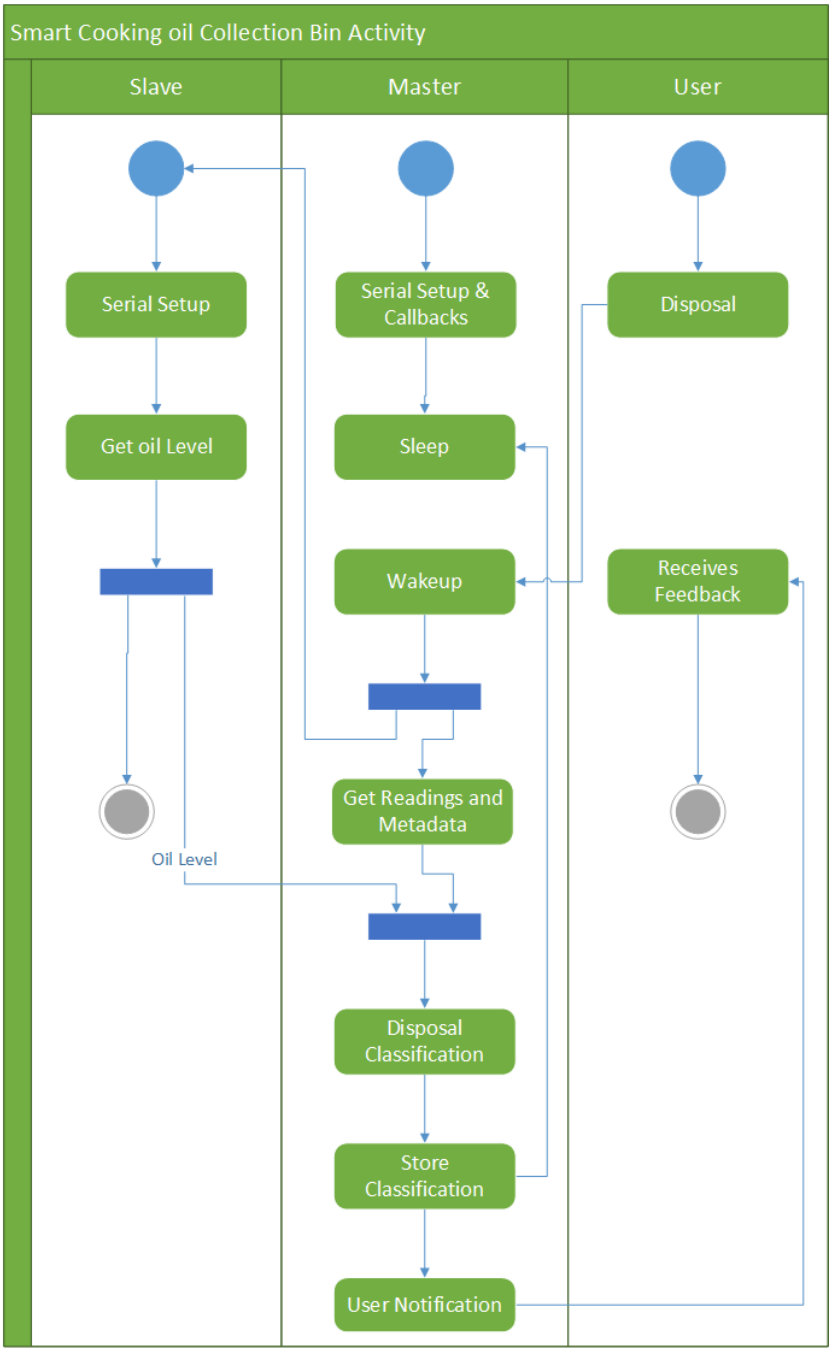


Figure 4.7: Smart cooking oil collection unit activity

4.1.5 Power Dimensioning

As mentioned in 4.1.1 by counting on local disposal classification, the smart cooking oil collection unit bypasses the need for communication upon each disposal, contributing to a simpler energy harvesting solution.

Two pre-established energy requirements were set:

1. Up to six months of continuous operation in the unlikely scenario of zero energy collected.

2. Self sustainable energy source. No grid backup available.

In order to dimensioning the the energy harvesting solution, the device daily duty-cycle was discretized into three different states:

- Sleep. Minimum current - $2x10^{-5}A$
- Awake 1. Awake state triggered by an oil disposal - approximately 5 seconds at $0.08A$
- Awake 2. Awake state triggered by the internal timer and aimed at GPRS communication - 30 seconds $1A$ (worst case scenario)

Considering a day with 10 disposals,

$$24h = 1440min = 86400s$$

$$current = 10 * \frac{5}{86400} * 0.08 + \frac{30}{86400} * 1 + \frac{86320}{86400} * 2 * 10^{-5} = 4.135 * 10^{-4}A$$

By implementing a duty cycle similar to the one presented at (Gomes et al., 2020), six months of continuous operation would require a battery capacity (C) of around 1.8Ah:

$$C = Xt \Leftrightarrow C = 4.135 * 10^{-4} * (24 * 180) \Leftrightarrow C = 1.786Ah$$

Given the low energy requirements of this application scenario, the cost of developing multiple power solutions by a function of solar irradiance, would probably exceed the cost of a single design. For instance, if considered a monthly solar irradiance, equivalent to 51 KWh/m2 (lowest irradiance registered during the course of a month in Porto, PT (Cavaco et al., 2016)), a panel of just 1.6cm2 should be enough for a self sustainable energy source:

$$51KWh/m^2 = 51000Wh/m^2 permonth$$

Get average power over area, HM = number of Hours in a Month = 720

$$51000/HM = 70.833W/m^2 = 7.083 * 10^{-3}W/cm^2$$

Considering a 20% efficiency for the solar panel (PE) and a 80% efficiency for the charging board (BE):

$$(7.083 * 10^{-3} * PE * BE = 1.113 * 10^{-3}W/cm^2$$

Knowing the energy harvested at a square centimetre, and the approximated edge required power DP of $4.2 * 4.135 * 10^{-4} = 1.737 * 10^{-3}W$, a panel size (PS) of around $1.561cm^2$ should, in theory cope with our energy harvesting requirements.

$$1.113x10^{-3}PS = 1.737 * 10^{-3} \Leftrightarrow PS = 1.561cm^2$$

$$\sqrt{1.561} = 1.249cm$$

4.2 Smart Crosswalk

Along with convenient solutions for our daily lives, the recent efforts in making cities smarter should also keep a focus in safety oriented systems. In recent years we saw the introduction of some approaches to pedestrian detection at (or near) crosswalks, leveraging on the *Internet of Things* to increase the safety of both drivers and pedestrians. From a sensing stand point, the typical systems for pedestrian detection at smart crosswalks still rely on conventional movement detectors. PIR sensors, doppler radars or even IR beam brake sensors are among the most used technologies for such task (of Transportation, 2013). While there are some strong reasons to continue to use the above mentioned sensors (i.e. costs, simplicity, known working principles) (Zhang et al., 2015) they still work under some strict assumptions that might not always be true. For instance, the human sensing is usually done just on the entry point of the crosswalk with the system losing track of the pedestrian during road crossing (Saad et al., 2020). This means that a person entering the crosswalk in an unusual manner will probably not be detected. Worth also noting that these sensors tend to implement movement detection techniques, instead of object (human) detection. This behaviour might lead to two serious limitations:

- False positives - The unit starts signalling the presence of a person while the moving object might just be an animal or simply a passing vehicle.
- False negatives - The unit fails to signal a standing still pedestrian, waiting for its turn to cross the road. Or, in another scenario, the system does not activate with an

"on road" falling pedestrian.

In order to summarise the state of the art on pedestrian detection, (Zhang et al., 2015) carries out a comparison between multiple detection techniques, with computer vision based systems displaying promising results when compared against traditional sensing techniques (Radar, laser or IR based sensors). While interesting from a detection stand point, the high computing and communication costs associated with a vision based detection leave exposed some of the weaknesses of a cloud computing paradigm. With the above mentioned in mind, a different approach was specified, taking from basis the architecture presented in 3.2. This application scenario, not only requires edge intelligence capabilities, but also the ability to perform computer vision operations at the edge. Therefore, no image data is exchanged with any remote device, at a cost of a more complex edge layer.

Conceptually, the main difference from the traditional IoT systems implementing artificial intelligent elements, lies on the load balancing between the different network layers. Instead of deploying end nodes as simple data collectors, an edge AI approach also delegates them the inference task. In an high level view, four different steps can be distinguished:

- Dataset preparation. Typically object detection models tend to take from basis an annotated dataset with bounding boxes identifying the target classes.
- Cloud training. A lightweight architecture, adequate for image processing, should be selected.
- Model conversion. In order to ensure an high (performance : power consumption) ratio, the edge device should implement, in hardware, most of the required functions for local inference. Thus, prior to deploying the trained model it should be converted according to the selected hardware.
- Local inference. The edge device starts the sensing and detection tasks. From this point, the vision capabilities combined with the local inference create what can be considered a sensor.

Figure 4.8 displays the pedestrian detection pipeline, from the the dataset collection to the actual edge detection.

4.2.1 Smart Crosswalk Architecture

While adopting EIASS, a new variable is added to the fog layer. Instead of relying on a single type of device for nearby fast data exchange, the smart crosswalk edge layer might count on an heterogeneous fog layer consisting on a wide variety of street assets, from

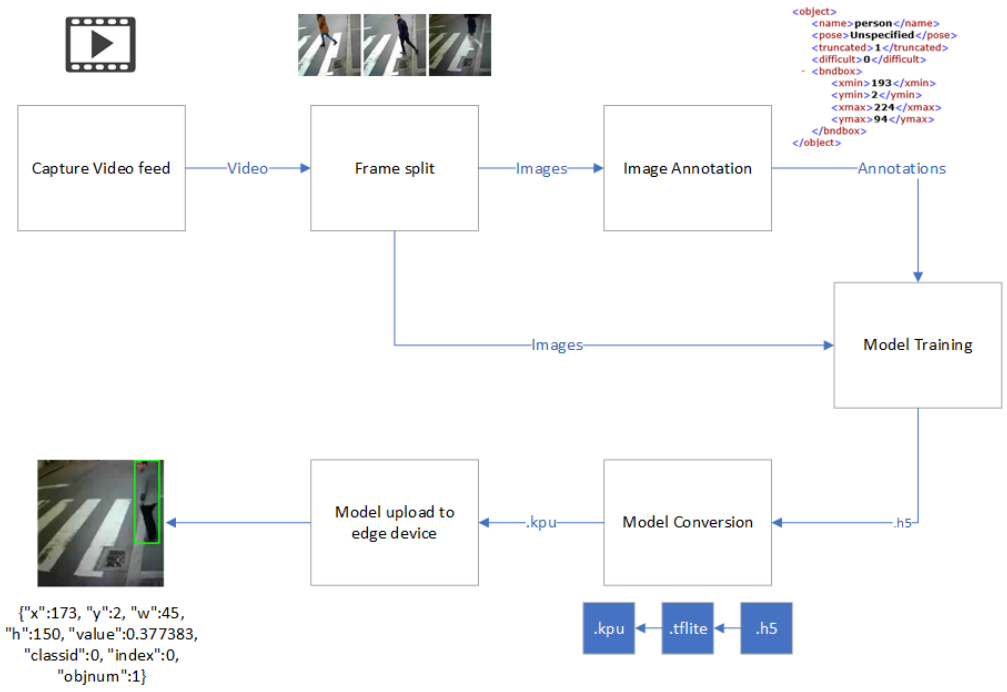


Figure 4.8: Smart crosswalk - Pedestrian detection pipeline

infra-structured light poles, to moving traffic on a vehicular network. From the edge layer stand point, the smart crosswalk scenario takes advantage of a master-slave architecture to cope with the specification of EIASS 4.10. A low power microcontroller (master) is left in charge of implementing the defined communication guidelines, while an on-site neural network accelerator enables an efficient pedestrian detection. Worth noting that the pedestrian detection task is totally delegated to the crosswalk edge device, without depending on the fog connection nor the slow, resource constrained, WAN communication. Therefore, considering the nature of the captured data (images), and the required high sampling rate, the edge intelligence, applied to the smart crosswalk problem, appears (like described in the next sections) as the ultimate example of the network offloading, data privacy and response times brought by an edge AI approach.

Network Offloading

Considering two systems with the same goal of, computer vision based, pedestrian detection:

1. Standard cloud computing approach.

2. Edge intelligence approach.

With both systems capturing RGB, 8bit depth, 224x224p 4.11, uncompressed images, the major difference stands on the cloud (1) vs on site (2) detection. In the first scenario,

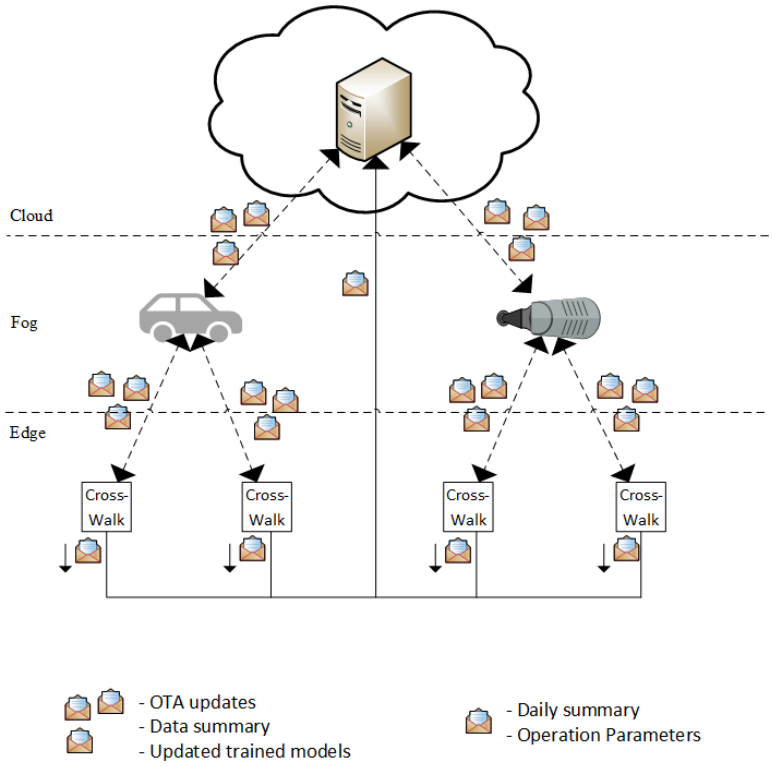


Figure 4.9: EIASS extension for the smart crosswalk

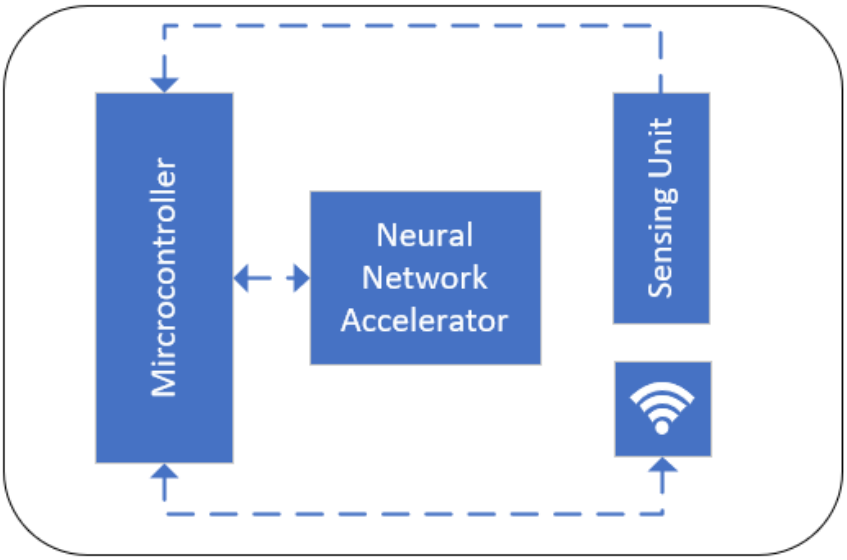


Figure 4.10: Edge device architecture

the image is captured and real-time transmitted to a remote web server, with the end node acting accordingly to the back response.

Figure 4.11: Captured image for pedestrian detection

$$224 * 224 * 3(8) = 1\,204\,224 \quad bits\,per\,frame$$

If an image feed of five frames per second would to be considered, just the payload for raw images transmission would required a bit-rate of more than 6 mbps (1204224 * 5). By performing image compression at the edge, this value could be significantly reduced (< 1mbps), but power constraints and limited network resources would still impose a problem for this approach.

On the other hand, the same system, implementing edge AI capabilities (2), could reduce the exchanged data to some (optional) sporadic telemetry communications. Therefore, by locally performing pedestrian detection, an edge intelligent approach is able to daily offload, from the network, dozens of GB for a single smart crosswalk.

Data Privacy

An edge intelligent, computer vision based, solution for pedestrian detection implies a continuous sampling (image capturing) of the public area surrounding the smart crosswalk. Thus, by tacking on the rules presented in section 3.2, the collected samples fall on the definition of personal data, as the requirement of pedestrian detection cloud potentially lead to person recognition.

If taken into consideration transmission, processing and storage as the tasks raising most of the security/privacy concerns, the choice for a cloud computing paradigm would require the implementation of complex encryption techniques for data transmission, as well as an effort to guarantee no caching or storage on the cloud side. On the other

hand, by locally sensing the presence of a pedestrian, an edge approach bypasses the problem, simply by not transmitting or storing any collected image, discarding it right after detection.

Figure 4.12 displays a qualitative comparison regarding the complexity / security risks on each layer for a cloud (upper diagram) vs edge (lower diagram) computing approach in the smart crosswalk context.

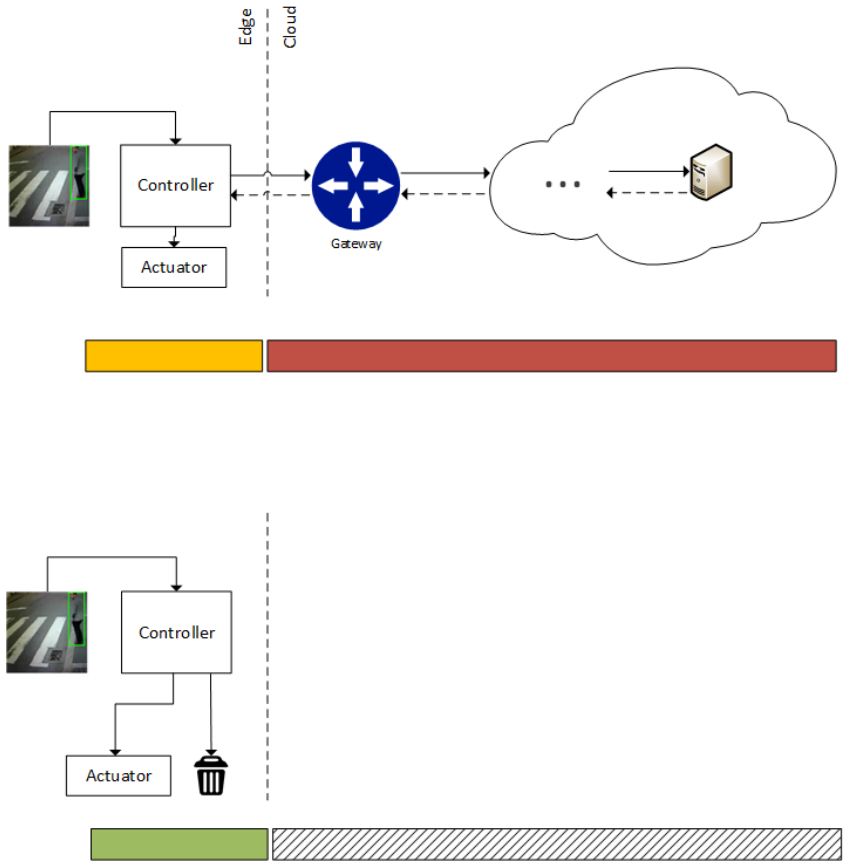


Figure 4.12: Privacy risks on an cloud (top) vs edge computing (bottom) approach

Response Times

Few fractions of a second separate a sidewalk walking pedestrian from one about to cross / crossing the road. Therefore, from the moment a pedestrian is detected at the entrance of a crosswalk, the system should immediately (<100ms) notify the flowing traffic.

These strict time intervals largely exceed the capabilities of the typical wide area networks (Mekki et al., 2019), making unfeasible an approach where the pedestrian detection occurs remotely. Even implementing advanced optimisation techniques (like the ones presented in (Costa et al., 2020)), aimed at reducing the required time for communication, the total round trip time still way above the defined 100ms maximum.

As a comparison, the AIoT board Sipeed Maix Bit announces an optimistic rate of 60 frames per second, which could translate into approximately 17 milliseconds sampling interval. In section 5.2 an evaluation on this time is performed, in order to assure that is copes with the pre-established requirements.

Scalability

On a hypothetical scenario where the network does not present a boundary for a cloud intelligent approach to the smart crosswalk problem, the resources required from the cloud would probably not scale well enough to power such an wide-spread system.

By not depending on a remote running detector, an edge intelligent crosswalk decentralises the pedestrian detection bringing built-in scalability. This means that the number of deployed crosswalks is independently related to required cloud resources.

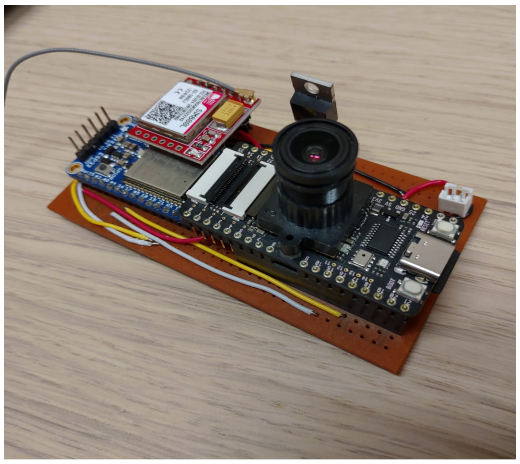
4.2.2 System Setup

With the EIASS supporting the edge intelligence paradigm, the focus cloud move to the specification of each of the systems components. For a proof of concept a sipeed maix bit, with its full MicroPython development environment was chosen as the slave node, while an ESP32 based board played the role of master device.

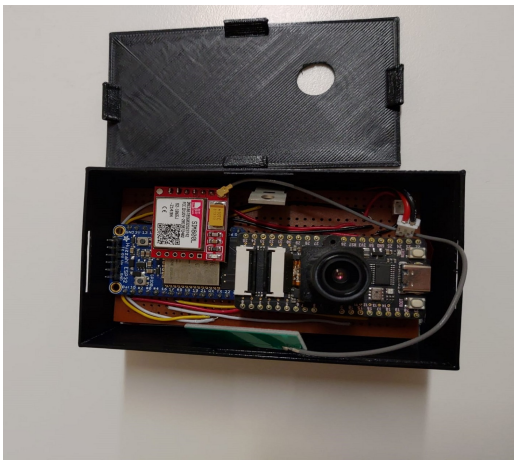
Figure 4.13 displays the edge device targeted at the smart crosswalk problem

As presented in figure 4.13, a serial communication is established between master and slave devices. This communication channel enables an half duplex data exchange between the two devices, given that an open serial communication is kept on both ends. In parallel with the serial communication, an additional connection is settled between the slave and master devices. As opposed to the above mentioned serial connection, instead of data exchanges, this one serves the purpose of system optimisation, aiming to reduce the overall power requirements, thus directly impacting the power dimesioning at 4.2.5. This power optimisation enables the slave device to keep its uninterrupted sensing capabilities while the remaining components (namely the master device) maintain a deep sleep state, mutable every time a pedestrian is detected. With no need for an additional border router, and with built-in TCP/IP stack, the SIMCOM module SIM800L was left in charge of the WAN communication. Worth noting that, as described in 3.2, this communication channel is intended for some sporadic meta-data exchanges and therefore totally interchangeable with most WAN networks.

Figure 4.14 displays activity flow maintained at the edge, from the moment the device is powered. The slave device is responsible for constantly checking for the presence of pedestrians crossing (or about to cross) the street, waking the master device every time a pedestrian is detected. The master device then performs filtering upon the received bounding boxes and actuates if a valid pedestrian detection is obtained.



(a)



(b)

(c)

Figure 4.13: Smart crosswalk edge device prototype (a), (b) and (c)

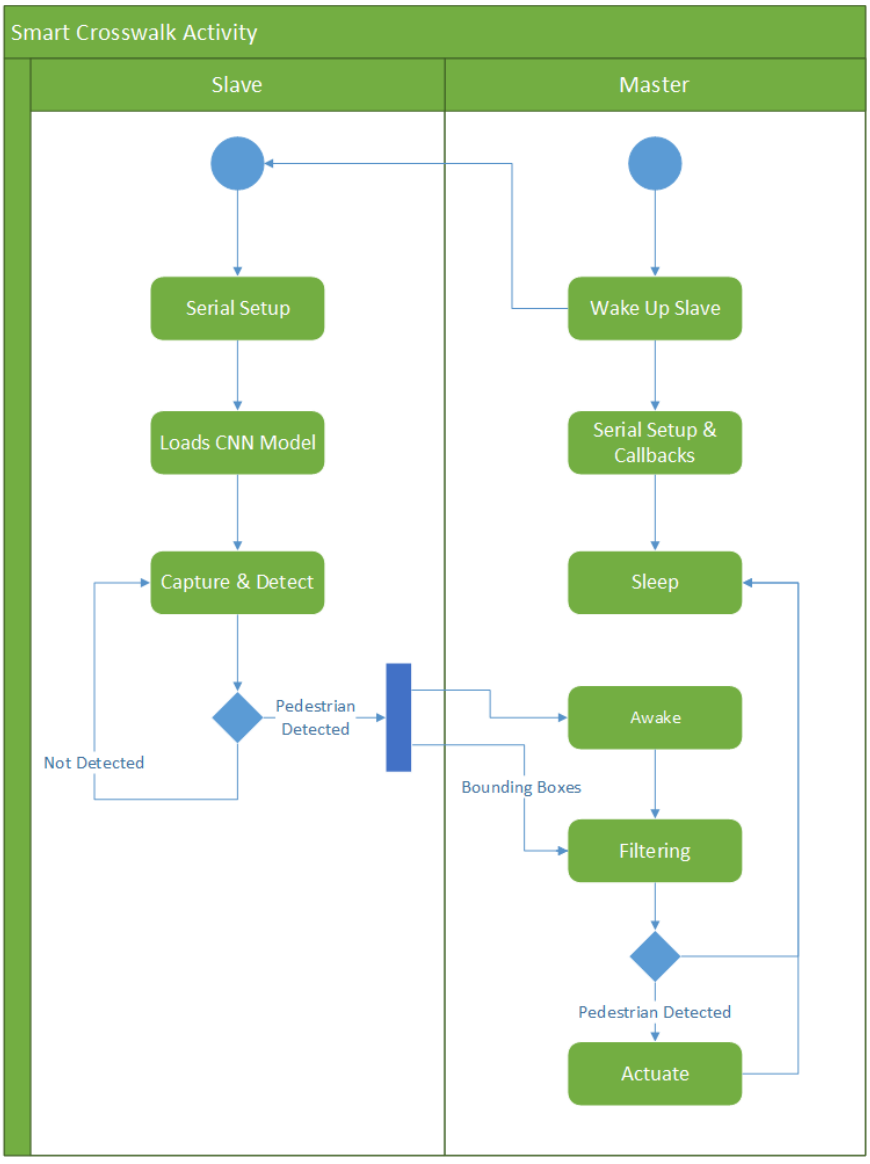
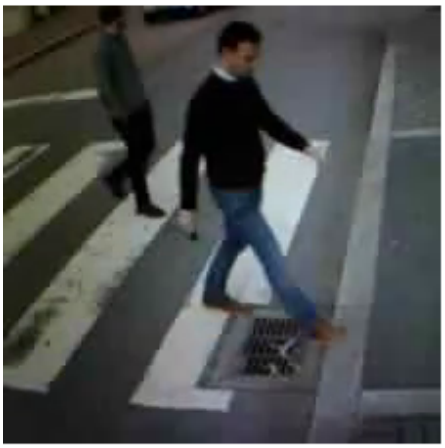


Figure 4.14: Master-Slave Activity

4.2.3 Dataset

In order to test and validate the proposed system a dataset of around 200 labelled images was collected.

The image set was created by splitting a video stream in its frames. Each one of these frames were then annotated in accordance with the pascal VOC (Everingham et al., 2010) data format (figure 4.15).



```
<object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>1</truncated>
    <difficult>0</difficult>
  - <bndbox>
        <xmin>46</xmin>
        <ymin>1</ymin>
        <xmax>79</xmax>
        <ymax>107</ymax>
    </bndbox>
</object>
<object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
  - <bndbox>
        <xmin>95</xmin>
        <ymin>7</ymin>
        <xmax>159</xmax>
        <ymax>164</ymax>
    </bndbox>
</object>
```

Figure 4.15: Pascal VOC data annotation

In order to prevent an overfitting to the dataset images, some simple data augmentation techniques were applied to the training samples, with the results behing shown at section 5.2

4.2.4 Edge Load Balancing

The task of object detection at the edge is carried out by the the slave device. In order to, efficiently, feedforward a network with almost 2 million parameters, most of the operations are hardware implemented, meaning that the board supports only a limited set of models/detectors.

Giving the suitable (and available) detectors for edge inference and their performance / accuracy ratio, the choice fell on version 2 of the real-time object detector YOLO (Redmon et al., 2016). Instead of performing the object detection in multiple steps, like

R-CNN and its variants, YOLO simplifies the detection to a regression problem, predicting classes and boundary boxes in just one run of the algorithm (Redmon et al., 2016), therefore contributing for a faster object detection.

Since the detection is supposed to run on a SoC with low computational resources, even with hardware implemented detection, the trained model would still need to be uploaded to the device RAM. Therefore, the state of the art architecture on mobile vision applications (Andrew G. Howard, 2017) - Mobilenet - as well as the lightweight version of the YOLO detector backend - Tiny YOLO - were chosen as the feature extractors for further evaluation on section 5.2.

4.2.5 Power Dimensioning

Energy management is still one of the most important features of any "off grid" IoT solution. Our lab analysis suggests that the edge device presented at 4.13 requires an approximate average of 100mA at 3.3v, during uninterrupted operation. Since the proposed system is intended for outdoor installation, renewable sources, like solar energy, are highly suitable to help powering the device.

Two pre established energy requirements were set:

1. Up to two full days of continuous operation in the unlikely scenario of zero energy collected.

2. Self sustainable energy source. No grid backup available.

With no energy being collected, only the battery is responsible to cope with the first requirement.

$$C = Xt \Leftrightarrow C = 100 * 48 = 4800mAh$$

Thus, accordingly to (Gomes et al., 2020), a minimum capacity of 4800mAh is required in order to guarantee, at least, 2 days of uninterrupted operation. This value does not take into consideration efficiency losses.

Given the heterogeneity in solar irradiance around the world, a single power setup would not be suitable for every working scenario. Thus, for an efficient deployment, the solar cell must be chosen in accordance with a specific working environment.

Taking into consideration the uneven solar distribution throughout the year, we chose to dimensioning the system to the worst case scenario. With a test prototype being installed in Porto, Portugal, the average $51KWh/m^2$ of solar irradiance registered in December (Cavaco et al., 2016), served as the starting point for our calculations.

$$51KWh/m^2 = 51000Wh/m^2 permonth$$

Get average power over area, HM = number of Hours in a Month = 720

$$51000/HM = 70.833W/m^2 = 7.083 * 10^{-3}W/cm^2$$

Considering a 20% efficiency for the solar panel (PE) and a 80% efficiency for the charging board (BE):

$$(7.083 * 10^{-3} * PE * BE = 1.113 * 10^{-3}W/cm^2$$

Knowing the energy harvested at a square centimetre, and the approximated edge required power DP of $4.2 * 0.1 = 0.420W$, a panel size (PS) of arround 20x20 centimetres (or $377.358cm^2$) should, in theory cope with our energy harvesting requirments:

$$1.113 * 10^{-3}PS = 0.420 \Leftrightarrow PS = 377.358cm^2$$

$$\sqrt{377.358} = 19.426cm$$

To summarise, two applications scenarios, implementing the proposed *Edge Intelligence Architecture for Smart Spaces*, were discussed and specified. Both smart cooking oil collection unit and smart crosswalk extended EIASS, inheriting its main benefits. Therefore, only the evaluation task is required in order to validate the concepts presented in chapters 3 and 4.

Chapter 5

Evaluation

Extending on the application scenarios specified in chapter 4, the present chapter keeps a focus on the evaluation of an edge intelligence paradigm applied to both smart crosswalk and smart cooking oil collection unit. A standard cloud computing approach to the same problems is used as a reference for a comprehensive result comparison.

5.1 Smart cooking oil collection Unit

Even highly relevant for a successful cooking oil collection bin, some of the requirements discussed in chapter 4 are not in-lab verifiable, given its subjective (e. g. unitary cost) or non-quantifiable (e. g. Service availability) nature. Therefore, considering the measurable requirements of the smart cooking oil collection unit, this section subdivides the system evaluation into three subsections:

- Training evaluation - result analysis for the cloud trained classifiers
- Classification Times - response times evaluation for the edge classification
- Energy Harvesting - Real life analysis on the energy collected / spent during a set of days of continuous operation.

5.1.1 Training Evaluation

As a convergence between the concepts of edge computing and *Artificial Intelligence*, prior to any other validation, an edge intelligent paradigm requires a trained machine learning model. In order to chose the most suitable model to cope with the requirements of the smart cooking oil collection unit table 5.1 displays the comparison on four different lightweight classifiers. Worth noting that a small footprint is required in order to port the trained classifier to the edge device, here represented by an ESP32.

Table 5.1: Classifier Comparison

Classifier	Train accuracy
Decision Tree	0.855
Naive Bayes	0.905
SVM	0.905

Looking at the results displayed in table 5.1 either the Naive Bayes classifier or the SVM seem to perform the best against the synthetically created dataset. Therefore, considering the unwanted conditional independence provided by the bayesian classifier (Shi and Liu, 2011), the SVM approach was the one to be chosen for further evaluation. Even considering its simplicity, the high chances of over-fitting (Kerdprasop and Kerdprasop, 2011) and the overall lower training score led us to discard the Decision Tree classifier. Worth noting that, the chosen classifier does not present a definitive answer for the disposal classification. With the underlying architecture (EIASS) enabling updated model distribution as well as over the air firmware updates, the edge classification is able to maintain a continuous improvement state.

5.1.2 Classification Times

With the time required for disposal feedback directly influencing the overall user experience, an effort is placed into reducing the time elapsed from the oil disposal to the classification feedback. In order to estimate the local classification performance, figure 5.1 displays the detection times required by the ESP32 to classify each one of the test dataset entries. While representative of the local inference times, these values do not take into consideration boot, sensor readings or actuation times.

With a mean time of 263 microseconds, the real-time classification achieved with an edge intelligence approach contrasts with the multiple second classification observed in a standard cloud intelligence paradigm. For instance, when comparing the 263 us to the 6 second obtained in (Costa et al., 2020), for GPRS communication, the shift to the lower layers of the network accounts for a disposal classification 22 814 times faster.

5.1.3 Energy Harvesting

Figure 5.2 displays the battery voltage levels during nearly 6 days of continuous operation under changeable weather conditions. A TP4056 Li-Po charger, interfaced an 1 Watt solar cell with the edge device and a 1000mAh battery, with the battery voltage being measured every 30 minutes via a voltage divider. In other to increase the load on the

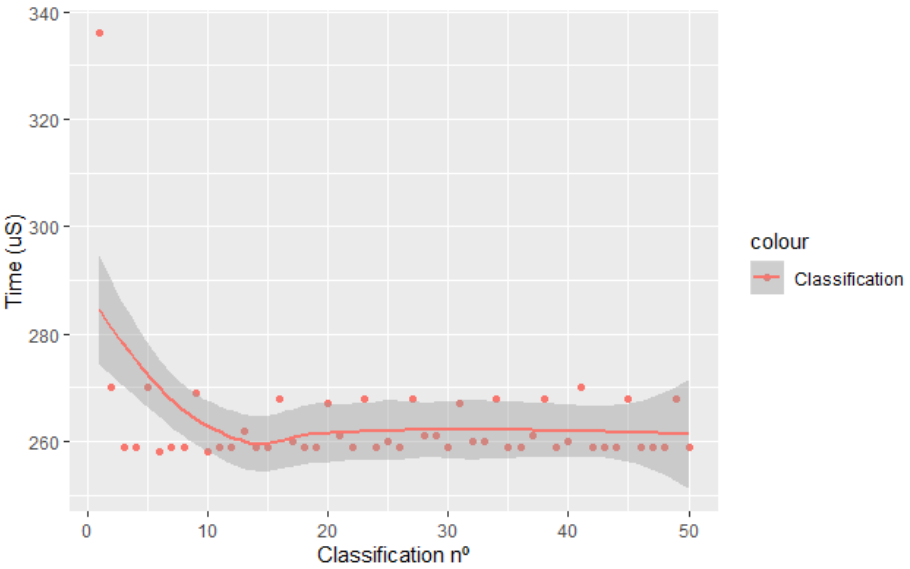


Figure 5.1: Classification times

energy harvesting solution, the voltage division was achieved by two 4.7kΩ, resulting into an additional current draw from 394uA to 447uA, depending on the battery's voltage.

$$I = \frac{V}{R}$$

At 4.2 volts

$$I = \frac{4.2}{9400} = 447uA$$

At 3.7 volts

$$I = \frac{3.7}{9400} = 394uA$$

Even with the synthetically generated extra load over the energy harvesting solution, a single day of direct sunlight is enough to keep the device powered for several months. During rainy/cloudy days (days 3 and 4 of figure 5.2) the collected energy stays close to the one spend during operation.

Although successfully validated, from a computing paradigm point-of-view, during our in-lab tests, the weight sensors did not achieve the manufacture specified accuracy. High sensitivity and overall low accuracy prevented the system from a reliable density estimation. Future works on this subject involve the choice for other products and sensing techniques to improve the effectiveness in oil classification.

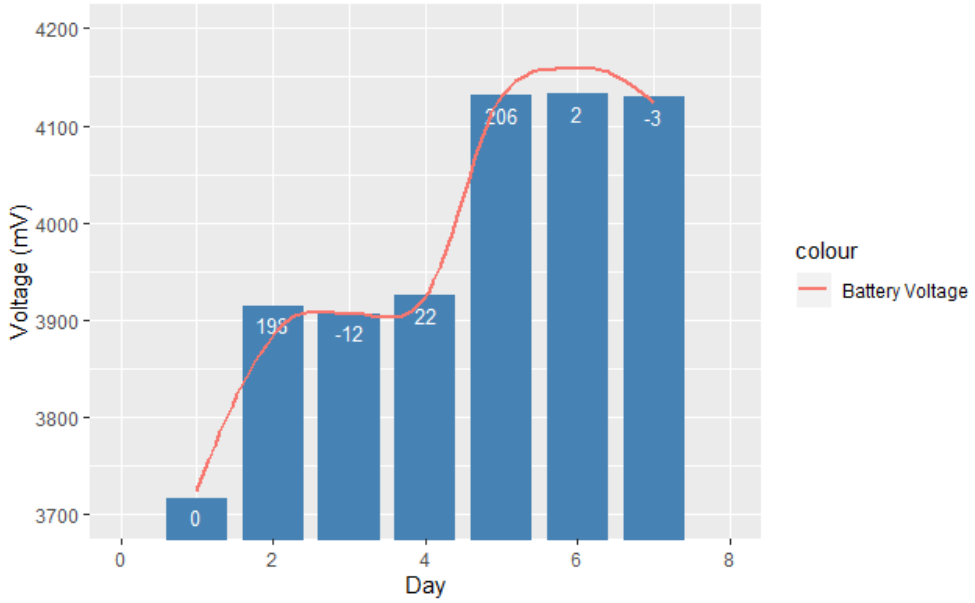


Figure 5.2: Energy harvested by the smart cooking oil collection unit

5.2 Smart crosswalk

Once implementing a computer vision based pedestrian detection, the smart crosswalk solution presents, by default, an extreme scenario for both computing and communication. In order to surpass these problems an edge intelligence paradigm was suggested in 4.2 with the evaluation taking place in this current section. In order to directly check the compliance with the established requirements this section is subdivided into three subsections

- Training Evaluation - Practical comparison between features extractors, against the dataset presented in 4.2.
- Detection Accuracy - Evaluation of the trained model on unseen data
- Response Times - Analysis on the time required for the pedestrian detection

5.2.1 Training Evaluation

Given the limited set of feature extractors supported by the sipeed maix platform, two lightweight models were compared: Tiny YOLO and Mobilenet in its versions of 1, 0.75 and 0.5 (width multiplier). Both training tasks were performed with the dataset specified in 4.2, under the configurations displayed at table 5.2.

Figure 5.3 displays the validation losses for the tested models. As expected, the higher the width multiplier the lower the loss. However, an alpha of 0.75 nearly reaches the same loss values of the full Mobilenet, confirming the results presented at (Andrew G. Howard,

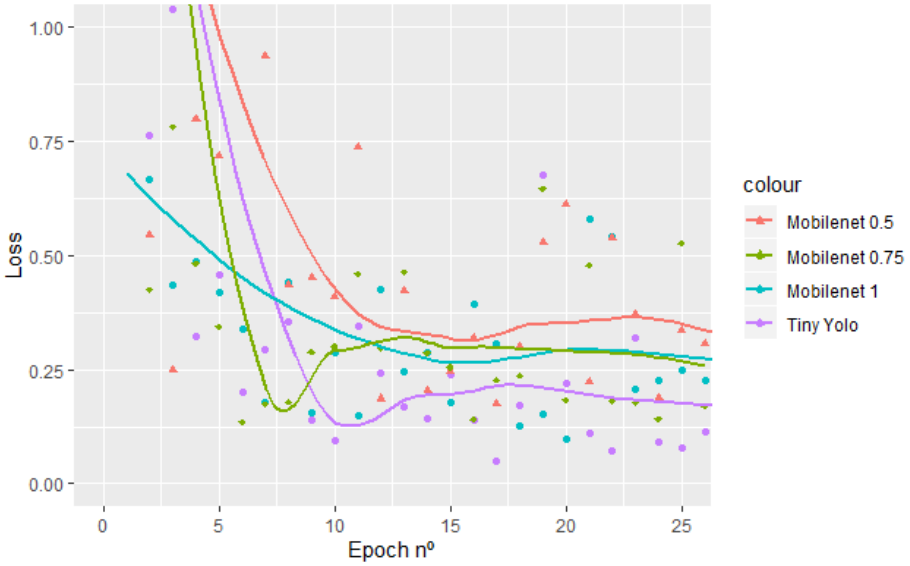


Figure 5.3: Feature extractor comparison

Table 5.2: Training setup

Architecture	**Input Size**	**Labels**	**epochs**	**train/val times**
Mobilenet 1	224	"person"	100	4
Mobilenet 0.75	224	"person"	100	4
Mobilenet 0.5	224	"person"	100	4
Tiny Yolo	224	"person"	100	4

2017). Even with a similarly small trained model, the lightweight version of YOLO backend scored the lowest loss values, therefore being chosen as the one to be ported to the edge device.

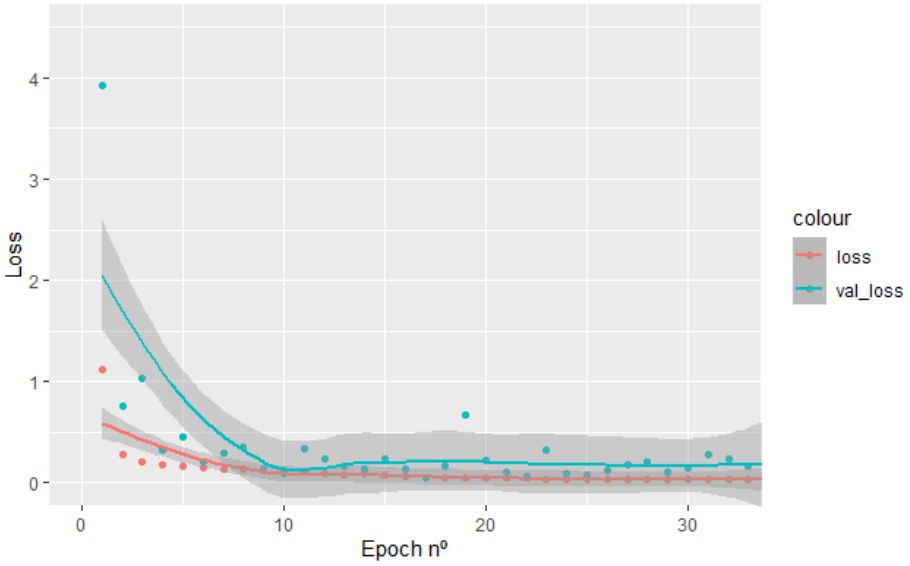


Figure 5.4: Tiny Yolo training and Validation Losses with jitter

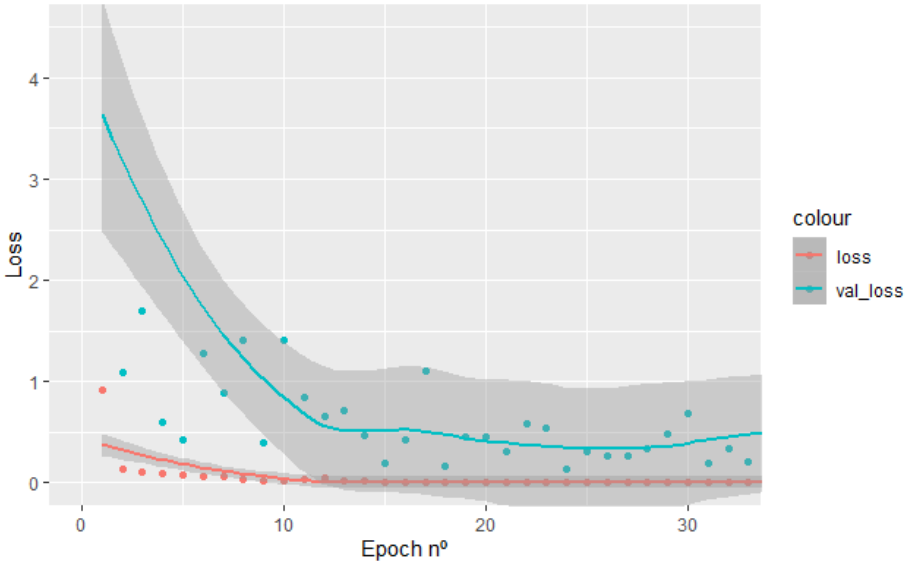


Figure 5.5: Tiny Yolo training and Validation Losses without jitter

Figures 5.4 and 5.5 display the differences between training and validation losses. On 5.5 the results acquired with no data augmentation suggest an overfitting scenario with the training losses significantly lower than the validation ones. On the other hand, the same dataset augmented by random zooms and shifts to the training samples 5.4 makes for a lower validation loss, this time closer to the losses obtained with training portion of the

dataset.

5.2.2 Detection Results

Lacking a way to estimate the detection accuracy in a regression problem, an empirical analysis was conducted on the trained detector against the test dataset. Figure 5.6 displays the model behaviour on unseen data.

(a) Detection 1

(b) Detection 2

(c) Detection 3

(d) Detection 4

(e) Detection 5

(f) Detection 6

Figure 5.6: Pedestrian detection Results

In order to further extend the encouraging results achieved in pedestrian detection, future works in this subject are targeted at improving the training dataset with a greater variety of detection scenarios. Also contemplated is the introduction of night vision,

wide-angle lenses and the definition of a pre-defined image Region of Interest for each installation.

5.2.3 Response Times

Being from the beginning one of the main functional requirements for the smart crosswalk problem, an emphasis is put into evaluate the response time gains achieved by a shift in the computing paradigm. Figure 5.7 displays the time (milliseconds) required for each iteration of the slave device.

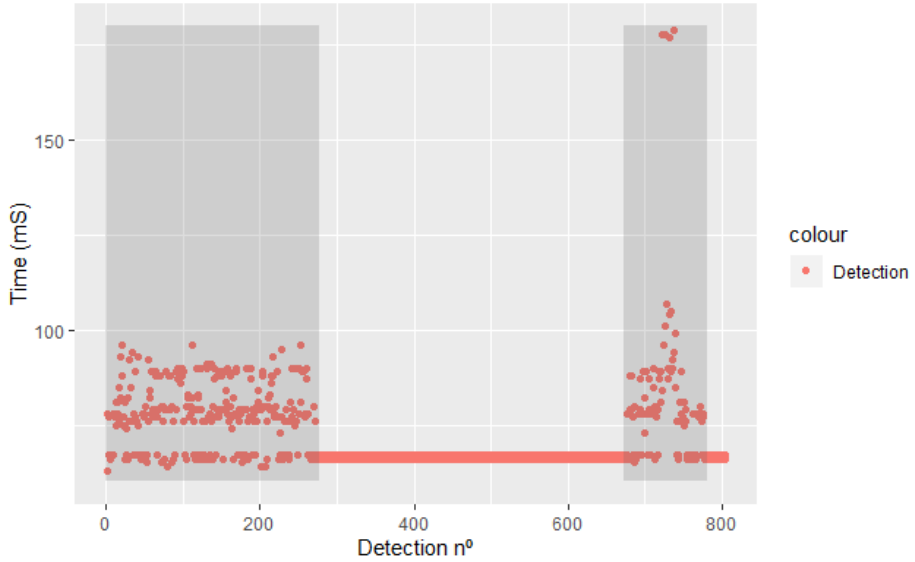


Figure 5.7: Pedestrian Detection Times

While presenting a total average of around 72ms for each detection, two different stages can be distinguished from figure 5.7. Bounded by a grey box are the time intervals with positive detection of one or more pedestrians. Proximity (size) or quantity of pedestrians are some of the variables affecting the detection times. All values were attained considering a standard KPU clock of 400MHz. When compared the average 72ms to the 6 seconds obtained in (Costa et al., 2020), an edge intelligence approach would perform the pedestrian detection more than 80 times faster than a standard cloud intelligence paradigm.

Chapter 6

Conclusion

The evolving concept of IoT is nowadays in a redefinition state. While, during most of its existence, connecting daily-life objects was the main goal of the *Internet of Things*, the new trend of load balancing computing towards the edge of the network is starting to relax this original concept. Benefiting from this urgent need to push computing closer to the data sources IoT is, once again, enlarging its scope, converging an even greater number of technologies.

By taking on the recent efforts in merging the concepts of an, edge computing enabled, *Internet of Things* with the acquired knowledge in AI, we conducted a top down approach to the edge intelligence paradigm. Aiming to distinguish between the different computing paradigms, while comparing its advantages/disadvantages we started with in-depth survey on this subject. After gathering the concepts, benefits and limitations of cloud, fog and edge computing, a generic architecture was specified - *Edge Intelligence Architecture for Smart Spaces*. Driven by the goal of creating a common ground between edge intelligence applications, EIASS carefully specifies data movements and load balancing over the pre-defined communication channels and network layers. In order to evaluate the EIASS two distinct application scenarios were presented. A smart cooking oil collection bin capable of locally performing disposal classification and a, computer vision enabled, smart crosswalk showcase the reduced response times, network load and energy consumption brought by the adoption of EIASS in expense of a cloud intelligence approach. Over-the-air updates, communication optimisation, and energy harvesting were some of the appended modules to the already complete EIASS specification.

In conclusion, the success achieved in materialising the conceptual works on edge intelligence, proved the high applicability of this disruptive paradigm in the near future of IoT.

6.1 Future Work

Although successfully validated, both application scenarios reflect an ongoing work, with finals products as the ultimate goal. Regarding the smart cooking oil collection unit, future work in these subjects is targeted at achieving a more accurate oil sensing. This resilience in untrustworthy disposal classification could potentially be implemented by the adoption of liquid contact sensing techniques, exploring the benefits of a standardised bottle. On the other hand, in the context of the smart crosswalk solution, the encouraging results obtained on pedestrian detection could be further enhanced by an improved dataset containing samples with a greater variety of scenarios. An ongoing work is being carried out in order to achieve the same detection levels during night. By providing UML documentation for both experiments, as well as the full image dataset used in the smart crosswalk scenario, is left open the possibility to replicate the obtained results.

Moreover, the specification of a generic architecture implementing an edge intelligence paradigm presents the basis on which new IoT applications can rely, hopefully contributing to inspire other researchers to bring this concepts even further.

Bibliography

Akamai. 20 years of edge computing. https://blogs.akamai.com/2020/08/20-years-of-edge-computing.html, 2020. Accessed: 2020-08-10. 6

Amal Al-Qamash, Iten Soliman, Rawan Abulibdeh, and Saleh Moutaz. Cloud, fog, and edge computing: A software engineering perspective. pages 276–284, 08 2018. doi: 10.1109/COMAPP.2018.8460443. ix, 6, 7, 11

Tanweer Alam. A reliable communication framework and its use in internet of things (iot). 3, 05 2018. 15

Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam Andrew G. Howard, Menglong Zhu. Mobilenets: Efficient convolutional neural networks for mobile visionapplications. 04 2017. 40, 45

R. Arslan and Y. Ulusoy. Utilization of waste cooking oil as an alternative fuel for turkey. In *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 149–152, 2016. doi: 10.1109/ICRERA.2016.7884526. 20

AWS. Aws iot greengrass. https://aws.amazon.com/pt/greengrass/, 2020. Accessed: 2020-07-16. 18

Azure. Azure iot edge. https://azure.microsoft.com/pt-pt/services/iot-edge/, 2020. Accessed: 2020-07-16. 18

Paolo Bellavista, Javier Berrocal, Antonio Corradi, Sajal K. Das, Luca Foschini, and Alessandro Zanni. A survey on fog computing for the internet of things. *Pervasive and Mobile Computing*, 52:71 – 99, 2019. ISSN 1574-1192. doi: https://doi.org/10.1016/j.pmcj.2018.12.007. URL http://www.sciencedirect.com/science/article/pii/S1574119218301111. 5

Cardoso Botelho, Carolina Gautier, and Carla Alexandra Monteiro da Silva. Valorização dos óleos alimentares usados com base na caracterização dos meios, pessoas e procedimentos. http://hdl.handle.net/10451/35414, 2018. Accessed: 2020-01-24. 20

S. B. Calo, M. Touna, D. C. Verma, and A. Cullen. Edge computing architecture for applying ai to iot. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3012–3016, Dec 2017. doi: 10.1109/BigData.2017.8258272. 8

Alvaro Cavaco, Hugo Silva, Paulo Canhoto, Samuel Neves, Jorge Neto, and Pereira Manuel. Radiação solar global em portugal e a sua variabilidade, mensal e anual, 2016. `http://www.ipes.pt/ipes/wp-content/uploads/2017/10/Radiaç~ao-Solar-Global-em-Portugal-e-a-sua-variabilidade.pdf`, last checked on 2020-10-26. 29, 40

W. Chang, L. Chen, C. Hsu, C. Lin, and T. Yang. A deep learning-based intelligent medicine recognition system for chronic patients. *IEEE Access*, 7:44441–44458, 2019. 9, 10

Cisco. Fog computing and the internet of things: Extend the cloud to where the things are. *White paper*, 2015. 5

F. L. Coman, K. M. Malarski, M. N. Petersen, and S. Ruepp. Security issues in internet of things: Vulnerability analysis of lorawan, sigfox and nb-iot. In *2019 Global IoT Summit (GIoTS)*, pages 1–6, 2019. 16

Intersoft Consulting. General data protection regulation - gdpr. `https://gdpr-info.eu/`, 2020. Accessed: 2020-04-23. 3, 16

Pedro Costa, Bruno Gomes, Nilsa Melo, Rafael Rodrigues, Célio Carvalho, Karim Karmali, Salim Karmali, Christophe Soares, José M. Torres, Pedro Sobral, and Rui S. Moreira. Fog computing in real time resource limited iot environments. In Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, Sandra Costanzo, Irena Orovic, and Fernando Moreira, editors, *Trends and Innovations in Information Systems and Technologies*, pages 102–112, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45691-7. 24, 35, 43, 49

Petre Dini, Pascal Lorenz, and José Neuman De Souza. *Service Assurance with Partial and Intermittent Resources: First International Workshop, SAPIR 2004, Fortaleza, Brazil, August 1-6, 2004, Proceedings (Lecture Notes in Computer Science)*. Springer-Verlag, Berlin, Heidelberg, 2004. ISBN 3540225676. 14

EloquentArduino. Micromlgen. `https://eloquentarduino.github.io/`, 2020. Accessed: 2020-09-07. 27

Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88 (2):303–338, June 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0275-4. URL `https://doi.org/10.1007/s11263-009-0275-4`. 39

The Physics Factbook. Density of cooking oil. https://hypertextbook.com/facts/2000/IngaDorfman.shtml, 2020. Accessed: 2020-10-20. 21

Gartner. 5 trends appear on the gartner hype cycle for emerging technologies, 2019. https://www.gartner.com/smarterwithgartner/5-trends-appear-on-the-gartner-hype-cycle-for-emerging-technologies-2019/, 2019. Accessed: 2020-01-24. 7

Bruno Gomes, Nilsa Melo, Rafael Rodrigues, Pedro Costa, Célio Carvalho, Karim Karmali, Salim Karmali, Christophe Soares, José M. Torres, Pedro Sobral, and Rui S. Moreira. A power efficient iot edge computing solution for cooking oil recycling. In Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, Sandra Costanzo, Irena Orovic, and Fernando Moreira, editors, *Trends and Innovations in Information Systems and Technologies*, pages 113–124, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45691-7. 2, 19, 29, 40

Hardlevel. Household collection of used cooking oil. https://www.hardlevel.pt/produtoseservicos-en, 2020. Accessed: 2020-09-11. 20

Weisong Shi Jie Cao, Quan Zhang. *Edge Computing: A Primer*. Springer, 2018. ISBN 978-3-030-02082-8. 7, 8

A. Jinguji, Y. Sada, and H. Nakahara. Real-time multi-pedestrian detection in surveillance camera using fpga. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, pages 424–425, 2019. 9, 10

Nittaya Kerdprasop and Kittisak Kerdprasop. Discrete decision tree induction to avoid overfitting on categorical data. pages 247–252, 07 2011. 43

Rob Kranenburg and Alex Bassi. Iot challenges. *Communications in Mobile Computing*, 1, 12 2012. doi: 10.1186/2192-1121-1-9. 1, 2, 16, 17, 18, 19

Phillip A. Laplante. *Real-Time Systems Design and Analysis: An Engineer's Handbook*. IEEE Press, 1992. ISBN 0780304020. 16

Li Lin, Xiaofei Liao, Hai Jin, and Peng Li. Computation offloading toward edge computing. *Proceedings of the IEEE*, 107:1584–1607, 07 2019. doi: 10.1109/JPROC.2019.2922285. 15, 17

Y. Lin, C. Chuang, C. Yen, S. Huang, J. Chen, and S. Lee. An aiot wearable ecg patch with decision tree for arrhythmia analysis. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, 2019. 10

K. L. Loh. 1.2 fertilizing aiot from roots to leaves. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 15–21, 2020. 8

Andreas Philipp Matz, Jose-Angel Fernandez-Prieto, Joaquin Cañada-Bago, and Ulrich Birkel. A systematic analysis of narrowband iot quality of service. *Sensors*, 20(6): 1636, Mar 2020. ISSN 1424-8220. doi: 10.3390/s20061636. URL http://dx.doi.org/10.3390/s20061636. 24

Kais Mekki, Eddy Bajic, Frédéric Chaxel, and Fernand Meyer. Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot. 03 2018. doi: 10.1109/PERCOMW.2018.8480255. 18, 24

Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot deployment. *ICT Express*, 5(1):1 – 7, 2019. ISSN 2405-9595. doi: https://doi.org/10.1016/j.icte.2017.12.005. URL http://www.sciencedirect.com/science/article/pii/S2405959517302953. 24, 35

Peter M. Mell and Timothy Grance. Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, USA, 2011. 5

S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B. Choi, and T. R. Faughnan. Real-time human detection as an edge service enabled by a lightweight cnn. In *2018 IEEE International Conference on Edge Computing (EDGE)*, pages 125–129, July 2018. doi: 10.1109/EDGE.2018.00025. 9, 10

US Department of Transportation. Pedsafe - pedestrian safety guide and countermeasure selection system. http://www.pedbikesafe.org/pedsafe/countermeasures_detail.cfm?CM_NUM=11, 2013. Accessed: 2020-10-21. 30

Angel Perles, Eva Pérez-Marín, Ricardo Mercado, J. Damian Segrelles, Ignacio Blanquer, Manuel Zarzo, and Fernando J. Garcia-Diego. An energy-efficient internet of things (iot) architecture for preventive conservation of cultural heritage. *Future Generation Computer Systems*, 81:566 – 581, 2018. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2017.06.030. URL http://www.sciencedirect.com/science/article/pii/S0167739X17313663. 19

George Plastiras, Maria Terzi, Christos Kyrkou, and Theocharis Theocharidcs. Edge intelligence: Challenges and opportunities of near-sensor machine learning applications. pages 1–7, 07 2018. doi: 10.1109/ASAP.2018.8445118. 8

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. pages 779–788, 06 2016. doi: 10.1109/CVPR.2016.91. 39, 40

Wasan Saad, Yasir Hashim, and Waheb Al-Areeqi. Design and implementation of portable smart wireless pedestrian crossing control system. *IEEE Access*, PP:1–1, 06 2020. doi: 10.1109/ACCESS.2020.3000014. 30

Mahadev Satyanarayanan, James Kistler, Puneet Kumar, Maria Okasaki, Ellen Siegel, and David Steere. Coda: A highly available file system for a distributed workstation environment. *Computers, IEEE Transactions on*, 39:447 – 459, 05 1990. doi: 10.1109/12.54838. 14, 18

Subhadra Shaw and A. Singh. A survey on cloud computing. pages 1–6, 03 2014. doi: 10.1109/ICGCCEE.2014.6921423. 5, 11

Hongbo Shi and Yaqin Liu. Naïve bayes vs. support vector machine: Resilience to missing data. pages 680–687, 09 2011. doi: 10.1007/978-3-642-23887-1_86. 43

W. Shi and S. Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016. 6

W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016. 17

Dimple Singh, Rean Maharaj, Nazim Mohamed, and Vitra Ramjattan-Harry. Potential of used frying oil in paving material: solution to environmental pollution problem. *Environmental science and pollution research international*, 24, 05 2017. doi: 10.1007/s11356-017-8793-z. 20

Statista. Internet of things (iot) connected devices installed base worldwide from 2015 to 2025. `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/`, 2015. Accessed: 2020-07-15. 1

TensorFlow. Tensorflow lite para microcontroladores. `https://www.tensorflow.org/lite/microcontrollers`, 2020. Accessed: 2020-09-07. 27

Jitendra Verma and C.P. Katti. Study of cloud computing and its issues: A review. *Smart Computing Review*, 4, 11 2014. doi: 10.6029/smartcr.2014.05.005. 5, 11, 14

X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 22(2):869–904, 2020. viii, 8, 9, 18

Ronghui Zhang, Fuliang Li, Jiali Zhou, Feng You, Wenxiao Zeng, and Tonghai Jiang. A review on pedestrian crossing detection and behavior analysis: In intelligent transportation system. *Recent Patents on Computer Science*, 8:2–12, 05 2015. doi: 10.2174/2213275907666141010214110. 30, 31

Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107 (8):1738–1762, Aug 2019. ISSN 1558-2256. doi: 10.1109/JPROC.2019.2918951. viii, 3, 8, 12, 14

G. Zichermann and C. Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, Inc., 2011. 21