D5.6 Final AI methods for use case applications and mechanism for AI transparency interactions

Deliverable Id:	**5.6**
Deliverable Name:	**Final AI methods for use case applications and mechanism for AI transparency interactions**
Status:	**Final**
Dissemination Level:	**PU**
Due date of deliverable:	**2022-10-31 (M26)**
Actual submission date:	**2022-10-31**
Work Package:	**5 ("AI & Safe Autonomous Decision")**
Organization name of lead contractor for this deliverable:	**Modis Consulting S.r.l**
Author(s):	**Daniela Angela Parletta (MODIS)**
Partner(s) contributing:	**Cristina Ganado (PROINTEC)** **Chiara Garibotto (UNIGE)** **Luca Paganin (RULEX)** **Luca Bertaccini (ETH)** **Alfonso Gonzalez Gil (ZYLK)** **Bekim Chilku (SIEMENS)** **Florian Thaler (V2C2)** **Gianluca Brilli (UNIMORE)** **Stefano Delucchi (AITEK)** **Artur Kaufmann (BEEWEN AUTOMATION)** **Fernando Eizaguirre (IKERLAN)**
Reviewers:	**RULEX, ROT**

Abstract: Task 5.3 is concerned with the analysis and, possibly, the enhancement of the FRACTAL AI methods in the context of the proposed use cases. This deliverable reports a summary of the identified AI solutions on a use case-basis, with a special focus on the transparency issues when appropriate. This is achieved reporting a detailed description of: the AI building blocks (i.e. the algorithms) developed in each use case; the data collection and data flows processes; the related transparency mechanisms. In the context of this deliverable, transparency is mostly meant as the clarity of the data collection processes and the presence of algorithms ensuring the explainability of the building block predictions (XAI). Finally, a methodological study on iris disease recognition aimed at illustrating the use of state-of-the-art algorithms to achieve explainability in AI.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056

Co-funded by the Horizon 2020 Programme of the European Union under grant agreement No 877056.

Contents

1 History

Version	Date	Modification reason	Modified by
0.1	01-05-2022	Creation	Modis
0.2	30-05-2022	Structure Modifications. Content filling.	Partners
0.3	12-06-2022	Structure Modifications. Content filling.	Partners
0.4	30-06-2022	Structure Modifications. Content filling.	Partners
0.5	15-07-2022	Structure Modifications. Content filling.	Partners
0.6	16-09-2022	First complete version for internal review.	Modis
1.0	27-10-2022	Internal reviews feedbacks addressed.	Partners

Table 1 – Document history

2 Summary

This deliverable is related to Task 5.3 which is concerned with the analysis and, possibly, the enhancement of the FRACTAL AI methods in the context of the proposed use cases. The focus of this document is on the description of the AI solutions (named *building blocks*) developed within each use case. In particular, the analysis focusses on the transparency aspects of these building blocks.

Transparency is a broad attribute concerning AI and involves all the aspects related not only to the algorithms themselves, but to the entire design and production phases. In particular, transparency refers to:

- the clarity of the AI algorithm in terms of its goals and its I/O;
- the clarity of the data collection processes; the capability of making the output and/or the model of an AI algorithm interpretable by a human (XAI);
- the clarity of the information that the algorithm acquires about its users, and the data flows it participates to.

The structure of this deliverable reflects the main features of the transparency attribute. After the introduction (Section 3), the transparency features analyzed in the document are defined (Section 4). In the same section the controversial relations with the GDPR are also discussed. From Section 5 to Section 12 the transparency analysis is carried on use-case-wise. In Section 13, a complete methodological study on the development of a transparent AI building block is described. Finally, in Section 14 and 15 the conclusion is drawn.

To summarize, for most use cases, many transparency features play a minor role. For a subset of them, some information is not available yet due to the fact that the development of the relative building blocks has not reached yet the maturity of certain stages (mostly the data collection processes). For some others, on the other hand, transparency seems reached on multiple features ranging from the clarity of the data collection up to having explainable output.

3 Introduction

Objectives

The goal of the FRACTAL project is to create a basic platform called FRACTAL node. It is a reliable computing platform node able to build a Cognitive Edge (a network that makes predictions and diagnoses) under industry standards. The FRACTAL node will be the building block of scalable decentralized Internet of Things (ranging from Smart Low-Energy Computing Systems to High-Performance Computing Edge Nodes).

Task 5.3 aims at enhancing the AI solutions developed in the FRACTAL project in order to cope with the many context learning and situation awareness tasks arising in the use cases. The objective is two-fold. On the one hand, it is important to identify the best algorithms for all the AI sub-tasks involved in each use case: such as visual identification and tracking from cameras, image recognition and classification, data-driven control, natural language processing, rule-based classification and so on. On the other hand, this task aims at identifying solutions for non-functional requirements, mostly energy optimization and AI explainability. In order to fulfill with these objectives, the AI tools and technologies described in D5.1 are suitably combined and enhanced.

This deliverable describes the AI building block designed to fulfill the use case's needs. Transparency is analyzed through the lenses of the data collection processes and (when appropriate) on the explainability.

Links to other deliverables

This deliverable shares some contents with others deliverable as for the description of the use cases. In particular, similar descriptions of the use cases can be found (at least) in D2.1, D5.1 and D8.1: The difference is on the emphasized aspects: D5.6 is mostly concerned with the role of the developed AI building blocks within each use case. These descriptions, in this deliverable, serve mainly to provide a context for the associated AI building blocks.

On the other hand, this deliverable has important connections with D4.5, D5.1 and D5.3.

- D4.5 is concerned with the compliance of the FRACTAL solutions to the GDPR. GDPR regards European normative about the privacy disclosure risks. This is somehow connected with a broad notion of transparency in AI, which is concerned with the possibility for a human operator of understanding all the processing made by an AI model. This deliverable (D5.6) focuses on the algorithmic aspects of the AI transparency: that is, how the AI algorithms acquire the data and present their output to human operators.

- D5.1 describes the AI tools and methods considered within the scope of the FRACTAL project. These tools and methods are combined together to design complex AI building blocks fulfilling the use case needs. A description of these blocks are given within this deliverable (D5.6).

- D5.3 describes the key fundamental AI concepts defined within the scope of the FRACTAL project. Such concepts have been deeply investigated within a number of scientific publications produced during the project and summarized indeed in D5.3. The AI building blocks described in this deliverable (D5.6) are designed to implement some of these concepts.

Organization

The deliverable is organized as follows: Section 4 presents a description of the key aspects that have been analyzed for each AI building block developed. Sections 5 to 12 describe the AI building blocks developed in the context of the FRACTAL project on a use case basis. Section 13 is devoted to a methodological study on iris diseases recognition that illustrates the use of state-of-the-art approaches to the explainability of AI model. Section 14 presents a summary of the transparency mechanisms employed in each AI building block. Section 15 draws the conclusions.

4 Concepts and Definitions

This section aims at introducing the key concepts and definitions that are considered in the analysis of each AI building block within the scope of this deliverable. Transparency is a broad attribute consisting of many different features that range from a clear algorithmic description (in terms of goals and I/O) of the AI building block to a clear description of the technical transparency mechanisms designed to achieve XAI, and passing through the data collection processes. Here, it is clarified what is meant for data collection processes and what are the type of transparency meachnisms considered in this deliverable. Finally, a discussion of the relevant aspect of the GDPR is presented.

4.1 Transparency in AI

Transparency is implied by the most basic conception of accountability: if we cannot know what an organization is doing, we cannot hold it accountable, and cannot regulate it. But the demand for transparency of algorithmic systems goes beyond that simple and assumed sense. We ask for transparent algorithmic systems because they are becoming so central to our lives and economies, and yet some of them use models and algorithms the workings of which are too complex for the human mind to follow. While the 'black box' metaphor is clearly evocative of this sense of impenetrable mystery of the systems acting upon us, it is important to consider if making the box transparent, so that we can see the gears within, is truly what is needed to satisfy our concerns with these systems. Depending on which aspect of an algorithmic system is in question, that is usually not what the calls for the transparency really aim at.

There are seven broad areas of machine learning systems where transparency might be demanded [1]:

1. Data. The transparency of the data used by the algorithmic system -- in particular by machine learning and deep learning algorithms -- can refer to the raw data, to the data sources, to how the data were preprocessed, to the methods by which it was verified as unbiased and representative (including looking for features that are proxies for information about protected classes), or to the processes by which the data are updated and the system is retrained on them.

2. Algorithms. The transparency of the systems' algorithms can refer to testing its output against inputs for which we know the proper output, reducing the variables to the most significant so we can validate them, testing the system with counterfactuals to see if prejudicial data is infecting the output, a third party code review, analysis of how the algorithms work, inspection of internal and external bug reports, or assurance the software development processes are sound. Algorithmic systems can also be transparent about their goals. When a system has multiple goals, this would mean being transparent about their relative priorities. For example, the AI driving autonomous vehicles (AVs) might be aimed at reducing traffic fatalities, lowering the AVs' environmental impact, reducing serious injuries, shortening transit times, avoiding property

damage, and providing a comfortable ride. A manufacturer could be required to be transparent about those goals and their priority.

3. Outcomes. Manufacturers or operators could be required to be transparent about the outcomes of the deployment of their algorithmic systems, including the internal states of the system (how worn are the brakes of an AV? How much electricity is used?), the effects on external systems (how many accidents, or times it caused another AV to swerve?), and computer-based interactions with other algorithmic systems (what communications with other AVs, what data fed into traffic monitoring systems?).

4. Compliance. Manufacturers or operators may be required to be transparent about their overall compliance with whatever transparency requirements have been imposed upon them. In many instances, we may insist that these compliance reports are backed by data that is inspectable by regulators or the general public.

5. Influence. Just as the public has an interest in knowing if an article in a newspaper was in fact paid for by an interested party, the public may have an interest in knowing if any element of the AI process was purposefully bent to favor a particular outcome. For example, if a trusted search platform is artificially boosting some results because they were paid to, and if it is not flagging that fact to users, users can be manipulated. Regulators might want to insist that such influence be conspicuously acknowledged.

6. Usage. Users may want to know what personal data a system is using, either to personalize outcomes or as data that can train the system to refine it or update it. Knowing what personal data is used, they may then want to control that usage, perhaps to make their personalized results more accurate, or, more urgently, because they feel that usage violates their privacy, even though the data in question may already be a desired part of the system, such as a purchase or search history. There are grey areas here as well: collecting anonymized, highly detailed information about trips made by autonomous vehicles — how often the car brakes or swerves, for example — could be important to optimize traffic for safety or fuel efficiency. Regulators may face some difficult decisions as well as drawing relatively obvious lines.

Transparency is therefore not a single property to be applied blindly to every element of every algorithmic system. It should be applied differently to different systems depending upon the nature of the algorithmic system, the complex circumstances that lead to the need for governance, and the goals of that governance. In the context of this deliverable, a special focus is put on the transparency aspects of the data collection processes, the goals of the AI building blocks and the explainability of the output.

4.2 Data collection

In this deliverable, data collection is meant as the family of computational processes involved both in the data acquisition campaigns for training an AI model and the data acquisition process performed by the AI building block once it is deployed in production. An accurate understanding of these processes is important when evaluating the transparency of the AI model [1].

First, AI models heavily depend on the data used at training time. A clear description of the training data in terms of size, sources, features, training/validation splits and its statistics is fundamental in order to explain part of the AI model behaviour. Moreover, this analysis can also disclose privacy leakage issues.

Second, once an AI model is deployed, in order to solve a task, it needs to constantly acquire data. Typically, this is done by means of a multimodal sensing infrastructure (e.g., cameras and microphones). Whenever the AI system interacts with humans, it is important to understand which data about the humans the system is acquiring, if the system acquires sensible information (e.g., financial data, health records, and so on), how these data are treated, where these data are sent, and more importantly whether the humans are aware of the data collection process and are willing to make available these data. All these aspects are important in the context of the AI transparency as they contribute to determine whether a system can be classified as a *white box* or not. A white box system is one whose prediction can be fully explained and whose data processing is disclosed to its human users.

4.2.1 Training data

Training data are a fundamental part of any AI project. Essentially, they determine the learnt model through its parameters. Even changing a single data point can affect the final model and as such it is always important to have a clear specification of the training data. Moreover, for a model to be truly transparent, it is important to have the ability to replicate it starting from its training data and a specification of the training algorithm parameters (including the random seeds). If, given this information it is not possible to reproduce the model, then the AI model cannot be considered as a white box.

Depending on the underlying model (deep networks or classical machine learning algorithms) a complete specification of the training data consists of the data sources, the training size, and, optionally, the collected features. In particular:

- The training data can be a publicly available dataset used as research benchmark, or a dataset acquired by a target data collection campaign or a mix of these.
- Training set size is determined by the number of available data points and their dimension either in terms of features or in terms of some other natural quantity such as number of pixels, number of audio samples and so on.
- Classical AI models require the data to contain relevant information about the phenomenon being modelled. This information is usually expressed in terms of a set of carefully designed measurements called features. On the other hand, deep learning models can work directly with raw data and internally "engineer" a good set of features.

4.2.2 Production data

Once an AI model is trained, it is deployed in production to make predictions. Predictions are computed on the basis of the environmental data. For example, in order to estimate the age of a person, an image of his/her face needs to be acquired.

An understanding of the process of data acquisition is important in terms of transparency. This is important especially if data about human subjects are acquired. The data collection process has to be fully specified in terms of the type of data that are acquired, where the data are stored, where the data are sent and how the data are used by the system.

4.3 AI Transparency Mechanisms

Broadly speaking, transparency mechanisms are a property of an AI system to be fully understandable by its human user. Transparency applies at different stages and levels of a system. Starting from the data collection (see above) up to its predictions and including all the intermediate steps. This deliverable focusses on the transparency of data collection processes and on the interpretability of the system output. The latter being usually referred as Explainable AI (XAI) [5,6].

4.3.1 Explainable AI (XAI)

Explainable AI (XAI) refers to the principle of making the operation of an artificial intelligence and its results understandable to the user as much as possible. We describe the Explainable AI (XAI) in relation to the following aspects:

• Explainability of AI model (i.e., is the description of the AI model understandable to the end user?)

• Explainability of AI results (i.e., are the descriptions of the AI model results understandable to the end user?)

Many state-of-the-art AI algorithms achieve super-human performances on many important tasks but behave like black boxes in that it is usually hard to figure out why a model made a specific prediction or even understanding its output. This is especially true when it comes to deep networks models that with millions of parameters and dozens of layers appear as mysterious objects that compute predictions on the basis of inner, hidden, feature representations of the environment. This fact is even stressed by the lack of grounded mathematical understanding of such algorithms due to the sheer complexity of deep models.

In order to overcome this situation, a number of solutions have been proposed that can be classified into two categories.

4.3.1.1 Explainers

An explainer is an algorithm that explains the predictions of an AI model. Roughly speaking, an explainer takes as input a model M and the object X to be predicted and builds an interpretable explanation about the prediction made by the model M on the object X. Usually, the explainer is agnostic of the specific model to explain, but is application dependent. That is, there are explainers for image classification algorithms, explainer for Natural Language Processing algorithms and so on. A popular explainer is LIME [4] which identifies the regions in an image that most contributed to its prediction (see the methodological study for more details).

4.3.1.2 Interpretable models

Interpretable models are AI models that are natively understandable due to their simple structures. Examples of such models are linear models and decision trees. The former, build their prediction by taking linear combinations of the features. By inspecting the model *i*-th parameter magnitude it is possible to understand at what extent the *i*-th feature matters in a prediction, and by looking at its sign it is possible to understand whether a feature contributes positively or negatively to the prediction. Obviously, if a linear model has hundreds of thousands of parameters, then it becomes unfeasible to look at each weight individually to "understand" the model. In order to face this situation, one can regularize the models by forcing *sparsity*. That is, it is possible to force the learning algorithm to produce a linear model with few non-zero parameters, which makes the model interpretable even in presence of many parameters.

Decision tree models produce predictions by inspecting a sequence of conditions on the input features. In other words, it is possible to represent a decision tree as a set of logical rules that governs the behaviour of the model. Depending on the height of the tree, such rules may become difficult to interpret. In order to avoid this phenomenon, it is possible to prune the tree reducing its height without penalizing too much its predictive accuracy.

4.3.1.3 Applications with interpretable results

Finally, there are applications with natural interpretable outputs. One prominent example is that of *image segmentation*. In image segmentation the goal is to classify each pixel of an image in order to isolate a patch or generate a heat-map. In such cases, regardless of the complexity of the model, its predictions (i.e., the segmented image) are naturally understandable. Obviously, even such applications can be enriched with explanations depending on the level of understanding required by the practitioners.

4.4 GDPR and Transparency

Legislation about transparency is still immature and partial. Perhaps the most relevant norm concerning AI transparency is provided by Article 22 of the GDPR [2,3]:

The data subject shall have the right not to be subject to a decision based solely on an automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her.

This article however is often used in practice to enforce the *right of explanations*: meaning that it is usually invoked to ask for an interpretable explanation about a controversial system decision. As such, its applicability is usually limited to critical domains such as healthcare, laws and financial decisions. On the other hand, this article cannot be applied each time an explicit consent is asked before any automated processing is performed. The result is that in many cases transparency can be circumvented by asking consent to the algorithm user before processing. Furthermore, the Article cite the "*the right…*" a fact that is generating many debates. As a right, there is no prohibition in adopting non-transparent automated decisions, it is left up to the willing of a user whether to dispose or not of such decisions [2].

Other relevant articles are 13 to 15 which invoke for the right to have a logical information about automated decisions. However, these articles are controversial since it is not clear what is meant by a logical information. Indeed, complex algorithms, such as those used in AI, do provide logical information about their output by just making their description public. However, it is rare that an algorithmic description, although fully logical, may carry any interpretation for a human user.

Finally, Article 35 related to the *data protection impact assessment* and the Data Protection Authorities (DPA) should control the algorithmic accountability. However, the concept of algorithm accountability is still vaguely defined and as a result such norms have not encountered a practical application [1].

5 UC1: Edge computing technologies applied for engineering and maintenance works

The objective of UC1 is to adopt Computer Vision and Artificial Intelligence techniques to digitalize some aspects of the construction projects in civil engineer. In particular, the objective is to develop AI based algorithms for improving the maintenance of critical structures and the monitoring of working areas to automatically detect risks for workers and machines.

5.1 Description of the use case

Nowadays, the construction sector is one of the less digitalized. This lack of use of technology starts in the first stages of the life of an infrastructure and continues until its final stage of operation and maintenance, resulting in an increase in costs and in dangerous situations during the work on site. This not only means a loss of business opportunities, but it also reduces the profit margin of the projects, losing efficiency in the works on site, and making it difficult to share information between different phases. Projects are becoming more and more complex and hard to manage with the current state of digitalization. In addition, constructions must be oriented towards sustainability and reduction of resource use, so some traditional methods must be modified, adapted or eliminated to meet the challenges of the future. Digitalization offers new ways of working that will allow to create knowledge of the construction business with clarity and transparency.

The construction work itself is the least digitalized stage in the lifecycle of an infrastructure. On-site analysis/consulting is necessary to improve and integrate processes related to cost, time and supplier management. In this stage, an abundant quantity of information is lost due to the large number of means of communication that are used (telephone, mail, personal, etc.) both internally and externally to the work. The constant monitoring of the work and its workers is also one of the key points to study, which helps to improve the quality of the processes, reduce costs and improve health and safety during the development of the works.

UC1 was born following the current tendencies in digitalization of the processes through the technological evolution of the systems, enhancing efficiency and reducing costs. Within this UC1, two end-to-end solutions will be developed and tested, which will allow improving economic and operation efficiency and safety conditions in the construction of civil engineering works. The UC1 aims to enhance the management environment by treating the collected data on site, through the use of IoT platforms and the deployment of sensors in the construction areas. Given this context, the UC1 has been focused on two main areas: (1) the maintenance of critical structures to increase the useful life of structures, and (2) the digitalization of the workforce equipment on construction sites; both areas aiming to improve safety and operation conditions in construction works.

The AI building blocks to be developed in UC1 are the **UAV supervision of critical structures** and the **WSN for safety at construction sites**.

5.2 UAV supervision of critical structures (UC1.1)

Description

Demonstrator 1 will be focused on the supervision of critical structures as bridges or viaducts, where images of the structural status will be collected through the use of UAVs, systematizing the visual inspection in near-real-time to detect failures and cracks in the concrete surface. Within this demonstrator, PROINTEC will develop an algorithm capable of distinguishing between active and non-active cracks of a wide range of pathologies registered in concrete structures. These measurements of the structural status will allow an early detection and classification of the cracks, and which is even more important, a comparison at different times of the evolution of the crack.

The information collected from the UAVs will be processed in the fractal node, and applying artificial intelligence, the images will be treated in order to create a map of the structural status of the detected cracks.

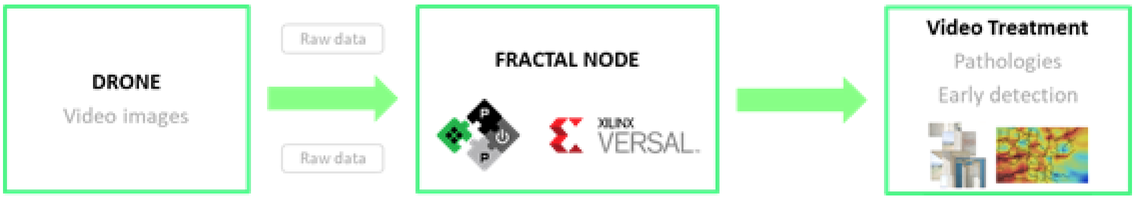

Figure 1 - UC1 demonstrator 1 scheme.

The developed tool tries to segment all the cracks contained in an image. The segmentation consists in the identification of the pixels that belong to cracks (labelled as 1) or to the background (labelled as 0). Additionally, the tool generates a report with relevant information about the images analyzed, such as number of cracks segmented, family or topology. Specifically, the building blocks used to achieve this result interact as follows:

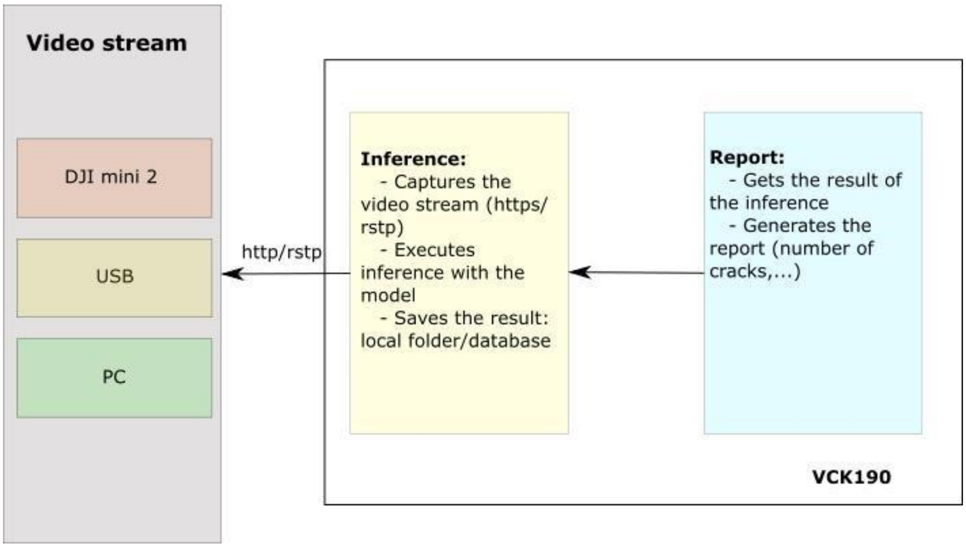

Figure 2 - UC1 demonstrator 1.

There are two main components:

- *Inference module*. This component is in charge of capturing the video stream, preprocessing every frame and adapting it to perform the inference using the trained model. As a result, the mask of 0s and 1s of each image in which the cracks are segmented will be stored in a local database to be analyzed by a human expert and the result of the inference will be passed as the input of the report module.
- *Report module*. This component creates the reports with the relevant information of detected cracks, based on the masks created by the inference module. In these reports the number of detected cracks will be specified, as well as the characterization of the crack families and topologies (e.g., number of cracks that form them, their relative position with respect to the reference, etc.).

In addition, all the information generated can be uploaded to the cloud in real time. Each of the results of the inspections should be stored in a database, along with the original images. In this way, the structure experts can easily access the data to ensure the integrity of the structure.

Data Collection

The following data sources have been used to train the model:

1. Datasets taken from the internet: In a first iteration, the DeepCrack dataset has been used to train the model [7]. As these datasets are usually very simple (flat concrete structures with cracks), data augmentation (flip, rotation, change of the brightness, superposition of textures and so on) has been implemented in order to enlarge the dataset. This dataset consists of ~20000 images of size 512x512.

2. Dataset created with images taken by the UAV at construction sites: The model trained with the datasets taken from the internet is fine-tuned with real images taken with a UAV on construction sites. Thus, the model will better adapt to the real scenario, as it will find images very similar to the ones used in this step (vegetation in the images, defects like paintings and so on). This dataset consists of 52 images of which it has been extracted ~3000 patches of size 512x512.

Once the model is deployed and more images have been taken by the UAV, new fine-tuning iterations will be carried out. The images taken in every construction inspection will be stored in a database in the cloud and, periodically, used to train the model locally.

In production, the data flow is as follows:

1. First the UAV collects images/video of the concrete structure under study.
2. These images are then collected and analyzed in the Xilinx Versal VCK190. This board should be located on the ground, next to the inspection site.
3. Once the images are analyzed in the Xilinx Versal VCK190 (the cracks are inferred), the raw original images are uploaded to the database, along with the masks (with the detected cracks) and some additional information (report of the cracks).
4. The structures expert that will assess the condition of the building under study, will access the images (raw images and masks) and the reports via an application connected to this database. Notice that the reports and masks generated in step 2 are not meant to be a definitive evaluation of the quality of the building. Instead, they are conceived as a support, so the structures experts can do their job in a safer and faster way.

Mechanism for AI transparency interactions

The model itself is a black-box in the sense that it is based on a CNN with U-Net architecture and thousands of parameters. However, the model has been conceived as a tool to ease and help in the inspection of structures, and not as a tool to replace the experts that currently carry out this type of work. Thus, in any case, the result of our segmentation will be used by an expert that will assess its quality and usefulness.

In addition, due to the nature of the problem, there is no danger of the model being biased towards some groups or suppressing information that may affect the final decision. In any case, the structures expert will make the last decision of the quality of the building under study, and the raw untreated original images of the building will be available along with the inference made by the AI system.

5.3 Wireless Sensor Network for safety at construction sites (UC1.2)

Description

Demonstrator 2 will be focused on the monitoring of both workforce and machinery within a construction area, by deploying a WSN that provides information about the status and location of the workers in real time. This information will be managed through an IoT platform, registering possible dangers and alarms, apart from establishing a protocol in case of emergency. With this solution, the risk of accidents involving machinery and workers will be reduced, improving traditional health and safety systems, focusing the action in the vision of zero injuries at construction sites.

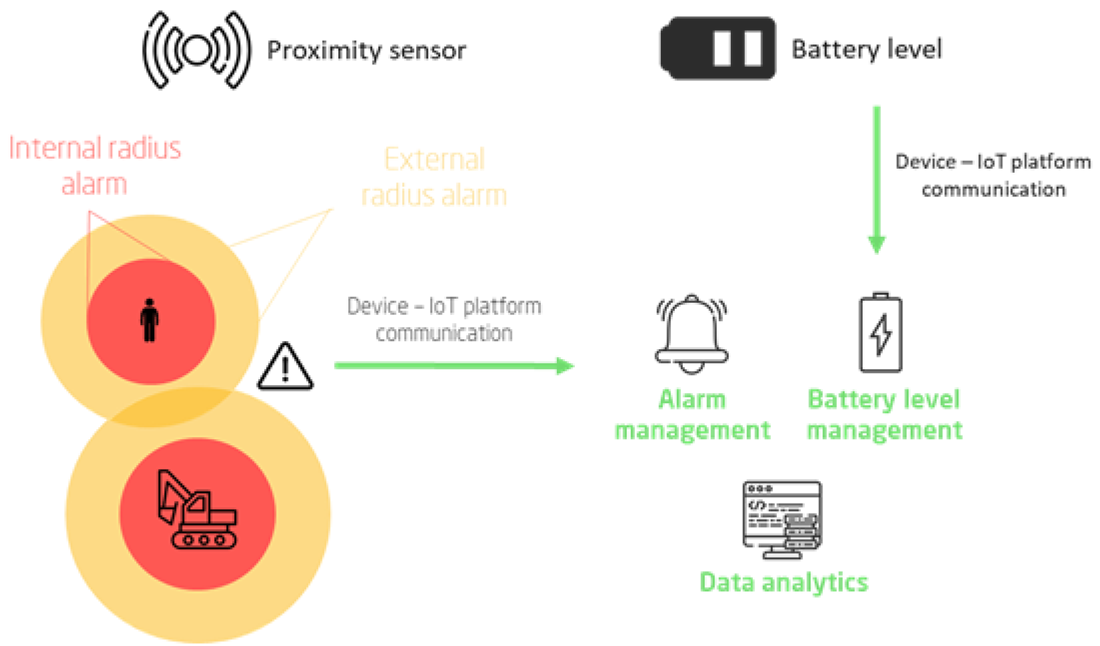

Figure 3 - UC1 demonstrator 2 relations description.

In this case, workers will carry personal sensors in their work equipment that will record their position, and will generate several alarms, both to workers and machines, when a vehicle comes too close and vice versa. All this raw information will be collected, processed in the FRACTAL node and represented in an IoT platform through a user-friendly dashboard. A report will be generated that allows us to take measures and reschedule the work on site.

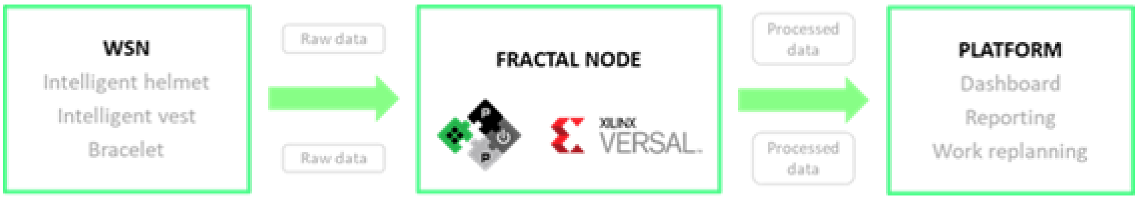

Figure 4 – UC1 demonstrator 2 schemes

All the information and data collected by the Wireless Sensor Network will be utilized to train multiple AI models, in order to obtain predictions about the potential generation of alarms, and early detection of hazardous situations involving the workers and the machinery, while also monitoring the generated alarms to predict whether they involve humans and machines.

These models will have three different functions: the Alert predictor (1), that will be able to predict whether alarms will happen or not in the near future; the Alert classifier (2), which will be able to tell the cause and type of the predicted and generated alarms (human-machine interactions, machine-machine interactions, or others); and the Anomaly detection (3) model, which will perform inference on the current situation of sensor positioning, in order to distinguish if the current positioning of workers and machinery corresponds to a regular situation or an anomalous one which could lead to hazardous situations.

These will be the three building blocks for the component, and each of them is described below in more detail:

(1) *Alert predictor*. The Alert predictor will take as input data structured time-series, where a timestamp is provided together with the relative positioning of workers and machinery. This information will be used by the model to determine whether these relative positionings constitute a hazardous situation or even if it directly results in a dangerous position of the worker with respect to the machine. This algorithm will be trained through a Long Short-Term Memory Neural Network (LSTM) which will be receiving as training data a collection of historical time-series-like data, where for each alarm detected in the historical data, the relative positioning of machines and workers is provided. This will allow the model to distinguish what situations are actually dangerous by labelling the hazardous situations with an alert flag.

(2) *Alert classifier*. This algorithm could be either Unsupervised or Semi-Supervised, depending on the necessities and the available data. The alert classifier is in charge of telling the user what kind of alarm is being dealt with. It is able to classify the alarms depending on the nature of the interaction between the sensors (worker-machine, machine-machine, several workers with the same machine, or multiple machines), and it allows the construction site manager to act in advance of the hazardous situation after an alert has been either already detected, or predicted by the Alert predictor. The inputs for this model are the filtered time-series data where the position of workers and machinery that lead to actual alerts or were predicted as potential alerts are provided. Clustering algorithms can be used for this, being trained with the historical data of recorded alerts, and trying to avoid predicted alerts because this would have a direct negative impact in the model’s accuracy by introducing outliers into the historical data.

(3) *Anomaly Detection*. Anomaly detection algorithms are used to detect what events in a time-series formatted dataset have special features that make them somehow different from the others. These "anomalous" data have statistical peculiarities that make them differ from the rest of the dataset, and these statistical differences can be computed by the model autonomously following specific rules. There are several algorithms for this, both supervised and unsupervised, and the selection of ones over the others will depend on the data labelling. For structured and labelled data (where the anomalous periods of time in the historical data are labelled) kNN , Bayesian networks, or Support Vector Machines can be used. In the case of unstructured (unlabeled) data, unsupervised ML models can be used, like K-means clustering or one-class SVMs, which learn from unstructured data and classify the data by their own. The advantage of this models is that they become better when functioning, because a semi-supervised strategy can be followed to tell the algorithm when it failed or succeeded in its predictions.

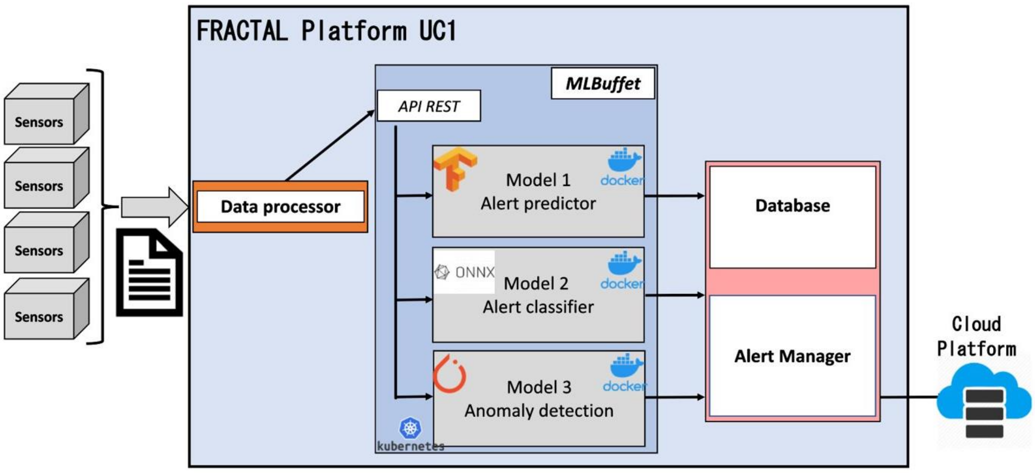

Figure 5 - UC1 demonstrator 2 architectural design of the FRACTAL platform.

The architectural design and the deployment frameworks of the models is depicted in Figure 5.

The board used to build the architecture is a Zynq UltraScale MPSoC (ZCU104), but could be also built on any ARM-64 bit architecture device. The FRACTAL Platform consists on a Data processor which could be any ETL tool (e.g., Java-based Apache NiFi or Python-based Apache Airflow, Pandas, PySpark...) that receives formatted data (XML) and pre-processes it into model-readable formats, depending on the model's architecture and input requirements.

An instance of MLBuffet is deployed on the board, orchestrated by any Kubernetes distribution, preferably Edge-oriented and light-weight ones like MicroK8S or K3S.

MLBuffet is an open-source tool developed during the course of the FRACTAL project and is based on a micro-services architecture of containers to build a Machine

Learning model server, in such a way that all the models can be managed in a decentralized manner, and MLBuffet takes care of the ML library handling and model deploying. It can also be used to re-train the models in the Edge, so that the features can also be used to re-train the algorithms when enough historical data are gathered.

Once the models have been trained and uploaded to the MLBuffet server, they are deployed and exposed through separate APIs for each model. The APIs can be sent requests from the Data processor, and the outputs of each model are then sent to a database for further storage, to the other models which can take them as inputs, and to the Alert manager.

Finally, the Alert Manager is in charge of notifying the users that something went wrong, or an alarm was predicted, and the database eventually sends its stored data to an external Cloud, where data are gathered into historical datasets, which will later be used for model-retraining and fine-tuning of parameters that improve the results.

Data Collection

Details on the training of the 3 blocks (predictor, classifier and detector) are not available yet.

The data acquired from the Wireless Sensor Network includes the position of workers with respect to the machinery that they have interacted with. Interaction here refers to being close enough that the sensors communicate with each other and register the area of the machine where the worker has entered the danger zone (areas being a numeric value depending on the zone disposition which is arbitrary).

This interaction generates a group of data, which consists of a timestamp, the type of sensor entering the danger area (beepers for wearable sensors, truck for machinery sensors, or zone delimiters), the type of sensor of the element being interacted with, the MAC address of both interacting sensors, and relative position between both (latitude and altitude).

These data are sent to the FRACTAL Platform as an XML which must then be formatted into JSONs to be sent to the MLBuffet APIs through the Data processor. Different data processing flows are used depending on the model that is performing the inference. The data are then ingested by the model and the output is stored in a database and sent to the Alert Manager in case any further action is required. Finally, as described in the AI Building Blocks, the stored data and the outputs are sent to the Cloud for historical data collection.

Mechanism for AI transparency interactions

The AI building block is not directly interacting with humans. However, the inputs of the model are the collected positions and alarms generated during the course of the construction and are in direct relation with human behaviour. These data are then processed by the models and the outputs are given to human end-users, who will use this information as an indicative and predictive response to hazardous situations that could be happening at the construction site. The human receiving the outputs from the system should be aware of how these data are processed and what they mean, by being instructed by a technical expert who can clearly explain what information is being collected (and what information is not being collected, like personal information or position tracking) and what operations are being performed on the data to reach the output result.

For privacy reason data are anonymized. Then, data are stored in local databases and sent to an external proprietary Cloud to gather historical datasets.

On the other hand, explainability is not required. The outputs will be reviewed by a human who assures their veracity.

The end user could be shown the data pre-processing outputs, but this information is not relevant because it is just a representation of the construction site positioning of workers and machinery, and this information can be accessed with the bare eye. The outputs of data pre-processing are stored in local databases which can be accessed by the end user, and the data pre-processing mechanisms are limited to data curation and formatting, with no modification of the content itself.

The results are presented by an Alert Manager which sends alarms to the end user operating the AI early alarm detection system. Monitoring of performance, accuracy and efficiency are presented through the MLBuffet metrics module.

The mechanisms that lead to AI results (the models and algorithms) are not shown to the end user through the FRACTAL platform, however, they will be public and accessible through GitHub repositories. The same applies to the mechanisms to evaluate performance, accuracy and efficiency.

6 UC2 Improving the quality of automotive cabin heating

The objective of UC2 is to adopt reinforcement learning (RL) algorithms to automate the choice of the heat source for cabin heating in electrical vehicles.

6.1 Description of the use case

Currently, the existing automotive **control strategies in Battery electric vehicles (BEVs)** are fully **reliant on model-based control techniques**. The Thermal management of the BEV is a highly nonlinear system with dead zones, hysteresis and delays. Typically, the behaviour of the system is modeled using first principles that allow physical interpretation, but it is hard to exactly capture the dynamics of the environment while they cannot be incorporated into the controller. The ability of traditional thermal management control strategies to perform self-learning through observations of specific situations is therefore minimal. The usage of these systems is **predominantly localized (at the vehicle level)**, but their development might benefit from swarm behaviour. This use case will, therefore, contribute to integrate complex environmental knowledge as a fundamental part of the system for the development of AI based thermal management systems in BEVs.

The resulting algorithms are general and lend themselves to be adapted to specific variations (e.g., uncertainties due to manufacturing tolerances or aging etc.)

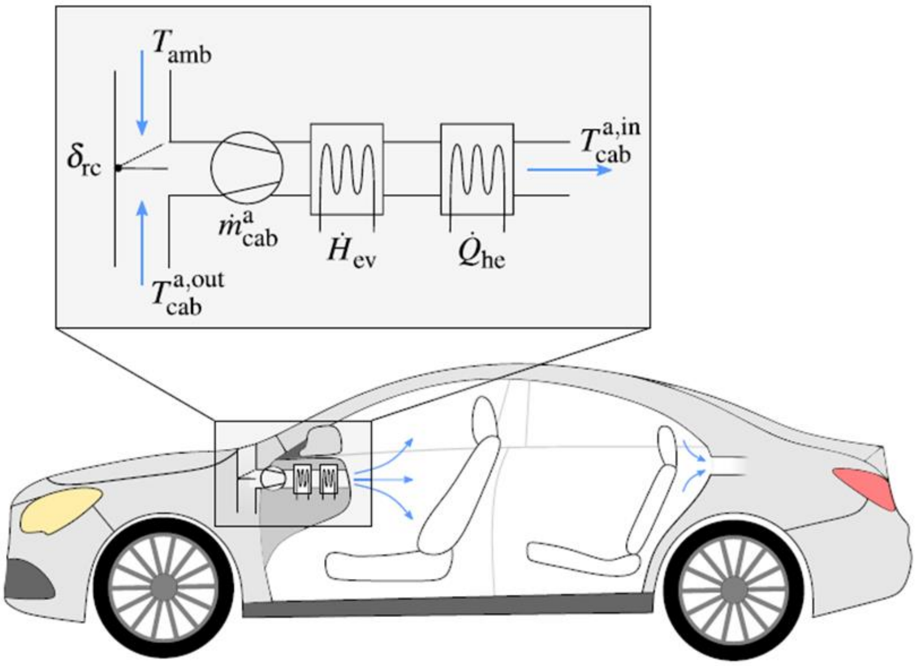

Figure 6 - Cabin Thermal Management for BEV.

In Figure 6 the high-level layout of the Thermal Management system of a BEV can be seen. The main actuators of the Air Path are the Lambda valve, and the heat exchanger. The main control variables of the system are the modes to select the temperature source for the cabin heating. The thermal management of a BEV **is a multifaceted system** consisting of components of various complexity levels. Together, they are responsible for critical functionality. Their operation has the potential to benefit from **proactive self-learning**. Such ability would help correct the component and the system shortcomings. The use case aims to demonstrate:

- Implementation of data-driven models aimed at improved energy efficiency and reduction of environmental pollutants
- AI-based self-learning and self-adaptation of the data-driven models, as well as consequential enactment of control strategies
- Identification of potential cyber-security breaches through anomaly detection

The concept of data-driven model application to thermal management has the potential to provide **higher precision in control and diagnosis**. The resulting **dependability enhancements** are possible through **simplified development with significantly lower computational effort** over existing controllers. The ability to self-learn and apply adequate improvements to the control system enable an appropriate response to changes in the environment of CPSoS. The quality of learning and predictions must be raised beyond the current level so that some aspects of the Learning Engine replace the existing model-based control. The collective learnings should improve **resilience and energy-awareness**.

We envision the development of various models within this UC. Each model makes use of in-use data (e.g., in-vehicle data for electric powertrain components). In the following we briefly characterize one potentially envisioned roadblock related to the Thermal Management of the BEV. The evaluation and online training of the Reinforcement Learning model on the real vehicle is not easy because of the involvement of huge computational overhead. In striving to balance a trade-off between **cloud** and **edge-based computation**, this use case also investigates crucial issues around massive computational power inside the system.

6.2 Data-Driven Control (UC2.1)

Description

The UC uses RL, an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. In the UC, the main target of the RL model is to select the mode from which the heat source has to be considered in order to do thermal management of the BEV based on the State of Charge of the battery

Data Collection

In this UC, different driving cycles are used to train the agents, which simulate the real-world driving conditions using various KPIs measured during real world driving.

The driving cycle data is collected either from Test Vehicles available at AVL Headquarters or from the simulators.

The already existing drive cycle data is uploaded via batch upload to FRACTAL Cloud platform, mainly to the object storage as JSON files. For the streaming data from the test vehicles, the data is being sent using the MQTT proxy to a specific Topic in the Kafka cluster, which is existing in the FRACTAL Cloud Platform.

The RL agent training is done entirely in the cloud. The data collected from streaming and batch upload are pre-processed and a drive cycle dataset is created for each driving scenario. The processed dataset is stored in the LakeFS, a dataset storage component existing in FRACTAL cloud.

Data from the electric powertrain, the battery management system, the ambient weather conditions and the cabin environment are collected for training the RL Agent. Figure 7 illustrates the data flow in the use case.

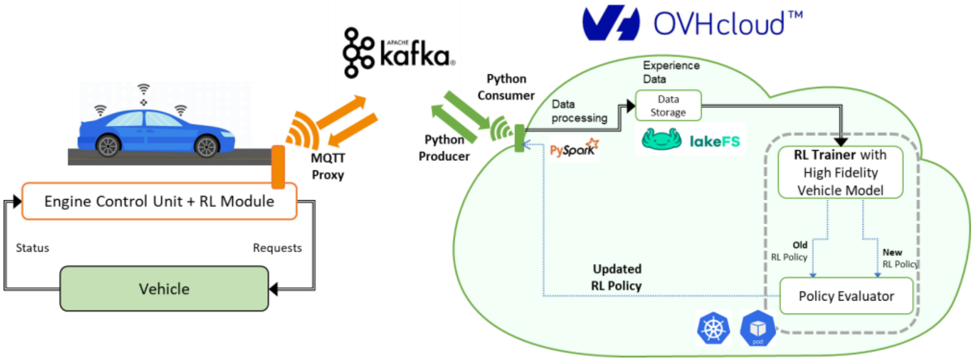

Figure 7 - Data flow in the Use Case.

Mechanism for AI transparency interactions

Transparency is not an issue in this use case, as all the processing is invisible to human users. Moreover, all processing results are kept within the system.

7 UC3: Smart meters for everyone

The objective of UC3 is to adopt Artificial Intelligence algorithms to empower utility meters with self-reading capabilities.

7.1 Description of the use case

Smart metering is a hot topic and one of the top use cases for the internet of things. The goal is to read the meters remotely by connecting them to the internet. This allows utility providers to remotely read the meters, with the benefit that they would no longer need to visit customers to physically read the meters. In order to support smart metering, the meters and its infrastructure around need to be electrified, which is often not the case. Especially legacy utility meters such as gas, and water meters often work with pure mechanical principles. Such meters lack power supply and an electronic interface for accessing the meter stand. Electrifying the infrastructure and replacing these meters with a smart device that is connected to the internet is a big investment.

A low-cost non-invasive alternative would be to put a battery-operated device equipped with a camera to take a picture of the meter and run a pattern recognition algorithm directly on the device to identify the meter stand. The extracted values can then be transmitted wirelessly over the cellular network. Such a device must have a small form factor in the range of a 3-5 cm^2 such that it can simply be tagged on a meter. It should consume as little power as possible such that it can be in the field for multiple years. Further, it needs to efficiently and reliably read the meter stand in suboptimal lighting conditions. And finally, it must be capable of transmitting the data over a wireless channel, even if the device is in a location with limited connectivity such as a basement.

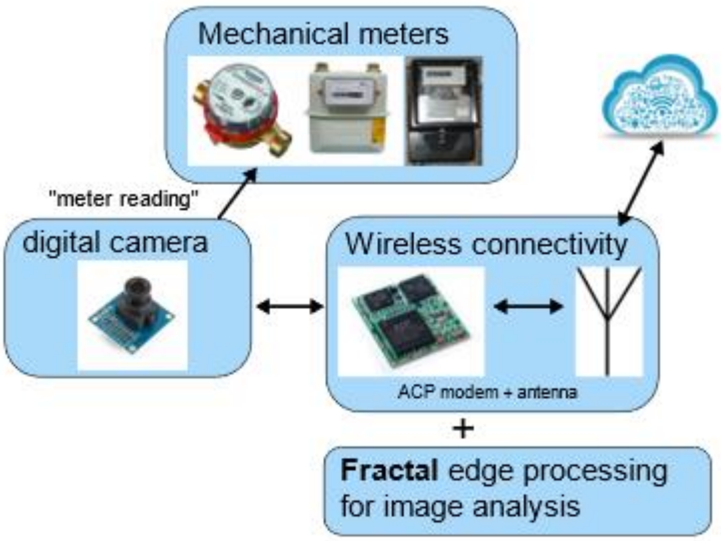

Figure 8 - Smart meter diagram.

As a final product, we envision a single chip solution that consists of a programmable platform that can extract the meter stand and run a protocol stack for wireless connectivity on the same chip. For this purpose, a powerful, but energy-efficient computer platform is required. To achieve this final product, a prototype based on the fractal platform will be developed. The following chapters will highlight how we want to achieve this goal and provide detailed platform requirements.

The AI building block to be developed in UC3 is **pattern recognition for self-reading meters**.

7.2 Pattern recognition for self-reading meters (UC3.1)

Description

The pattern recognition module simply consists of a CNN trained to recognize characters into an image. In particular, an offline-trained CNN, deploying the trained NN model on a PULP-based design will be used and will perform the inference directly on the edge.

Data Collection

Details on the training and test data are not available yet.

The data flow is very simple:

1. An ultra-low power camera acquires an image containing the meter stand.
2. A CNN performs the inference extracting the reading from the image.
3. The reading is transmitted over a cellular network.

Mechanism for AI transparency interactions

In this use case, there are no transparency or explainability issues.

8 UC4: Low-latency Object Detection as a generic building block for perception in the edge for Industry 4.0 applications

The objective of UC4 is to adopt Artificial Intelligence algorithms to build a vision-based object detection and recognition system with Low-Latency capabilities.

8.1 Description of the use case and of the AI applications

Real-time object recognition has been shown to be an important segment for many industrial applications where automation replaces manual work. This feature enhances the automation process with intelligent capability to detect and recognize objects visually. The whole process is based on machine learning approaches where the inference of a previously trained convolutional neural network (CNN) is used as an algorithm for detection and recognition of the objects from the input data.

The computation capacity provided by nowadays edge computing made it possible to run the object detection and recognition algorithms closer to the location where the object is observed, and with that to eliminate the needs for sending the input video data to the cloud services for data processing. The proximity to the source brings few crucial benefits for this type of solution. First, it eliminates the need for remote computation in the cloud and with that the need for wide bandwidth, second, it lowers the respond time that would have been imposed due to the network communication delays, and third, it increases the privacy by keeping the video data local. Running the inference locally on the node also enables the edge computing device to perform the process of detection and recognition in real time.

This Use Case has the goal to implement a vision-based object detection and recognition algorithm in the form of a Low-Latency Object Detection (LLOD) building block as part of the FRACTAL edge platform. The proposed LLOD building block will have the ability to detect the objects, localize their positions in the image, and categorize them based on pre-defined classification. Figure 9 shows the main component of this use case as well as the flow of the data. The LLOD building block takes as an input a video stream generated from the camera. The stream is handed to a device that runs algorithm for computer vision on top of it. Once the frame processing is finished the device publishes the results on the display. The output is localization of the objects in the image and their classification based on the group that they belong to. All this will be performed in real-time as the input video stream flows. The detection, localization and recognition of the objects will be done with the help of inference of a previously trained convolutional neural network model called YOLO, which is described below. The LLOD building block will be implemented in the form of a hardware accelerator as part of a larger SoC deployed on FPGA hardware platform. The flexibility of FPGA allows the LLOD building block to be configurable and adaptable for different neural network algorithms. Thus, any change in the inference will simply require reconfiguration of the accelerator in order to improve execution speed and reduce energy consumption.

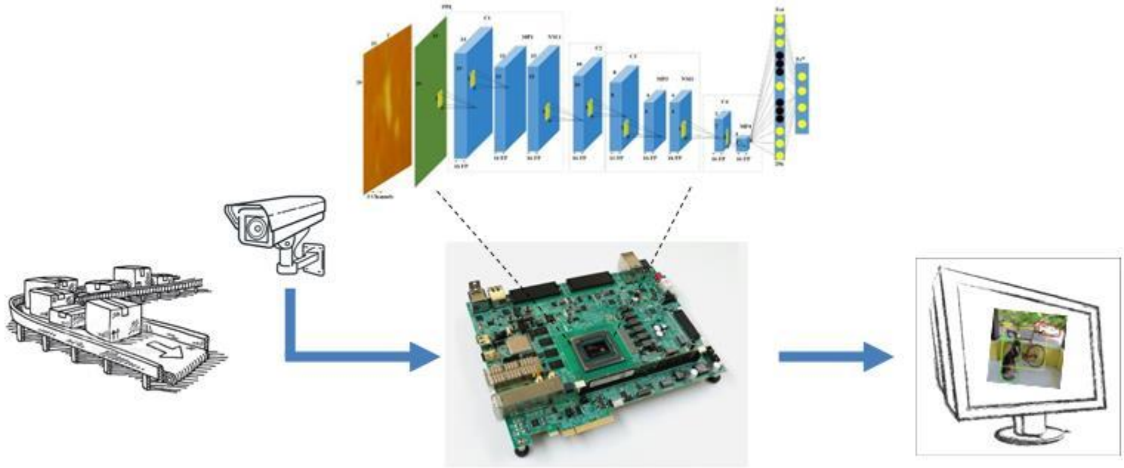

Figure 9 - VER-UC4 Object detection and recognition in industry.

This Use Case will focus on exploiting the performance and behaviour of the new proposed hardware extensions developed in WP3. The inference will be run on proposed hardware and the outcomes will be evaluated. This will allow us to have a better understanding of the impact that proposed hardware extension can have on the execution performance of the inference for visual computation.

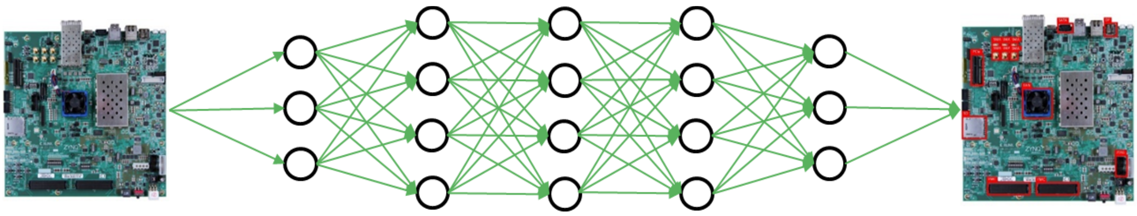

Figure 10 - Example of circuit classification.

Once the prototype is ready it will be handed over to use case Use Case 8 to be integrated as part of the SPIDER autonomous robot.

8.2 Object detection and recognition (UC4.1)

Description

Object detection and object recognition are techniques used for detecting and identifying objects within an image or a video. Object detection has the goal to localize the objects in the image, while object recognition understands the content of the image and identifies the objects on it.

The light variation of the YOLO algorithm with less convolutional layers and fewer filters is called Tiny-YOLO. This solution consists of 15 layers on which kernels of size 3x3 and 1x1 are used for convolutional layers and kernel of size 2x2 for pooling layers. The inference has a smaller size (less than 50MB), is a few times faster than the main version and achieves a higher rate of frame processing.

The reason for choosing Tiny-YOLO as inference for edge computing in UC4 are:

- It detects and identifies the objects in the images very fast,
- Has a high rate of processing frames per seconds,
- Has a small size that makes it suitable for embedded devices,
- Achieves a high accuracy on object identification,
- It analyzes the whole image at once,
- Has a low rate of background errors compared to other approaches,
- It is a mature solution for object detection and recognition,
- And it is an open source trained neural network.

Apart from these advantages, YOLO has limitation as well. From each grid cell in the image the model can only predict two boxes and can only have one class per cell. The model also struggles with objects that are small and appear in group. These constraints limit the number of objects that can be predicted within a cell.

The LLOD building block will be implemented on the Xilinx VERSAL Adaptive Compute Acceleration Platform (ACAP) and the Parallel Ultra Low Power (PULP) platform. Such an approach will give us a better understanding on the impact that different hardware designs developed in WP3 can have on the behaviour and performance of the neural network inference for computer vision. Xilinx ACAP is described in section **Error! Reference source not found.**, while the PULP platform is described in section **Error! Reference source not found.**.

The LLOD prototype to be developed for this use case consists of a camera, the software/hardware platform from the FRACTAL edge node and a display. The camera points to the production line and is used for generation of the input video streams. The frames from the video stream are processed from convolutional neural network that runs on one of the defined hardware platforms. The output results on detection and recognition of the objects are shown on the display.

In order to observe the impact of different hardware architecture on the execution of the inference we propose five solutions with diverse hardware architectures:

- The first solution will run the neural network inference on scalar processor without utilizing any hardware acceleration. For Xilinx ACAP platform this will be an ARM processor, while for the PULP platform a RISC-V processor. This will demonstrate the difference between ARM and RISC-V processors when they run inference of neural network.
- The second solution will utilize the AI engine provided from Xilinx platform. The neural network inference will be mapped on the AI engine to exploit the benefits that could be gained by using the array of AI engine tiles. This case will demonstrate how the AI engine deals with data-level parallelism and what impact the AI engine has on the performances of the neural network. The whole demonstrator will be built using the SDK provided by Xilinx for ACAP platform.
- The third solution is based on a High-Level Synthesis (HLS) hardware accelerator, which consists of a configurable array of processing elements. The accelerator will be implemented in the programmable logic part of the board for both Xilinx ACAP and PULP platform. In contrast to the second

solution where the accelerator is a hardcore solution, this one is an array of processors that is flexible, configurable, and adaptable. This case will demonstrate the impact of adaptability of accelerator on the performance of neural network. The outcome results will also be compared with Xilinx ACAP solutions that uses the AI engine.

- The fourth solution will be implemented on heterogeneous embedded system on chip (HESoC) consisted of a standard scalable processor as host and a cluster of programmable many core accelerators (PMCA). The platform is called HERO [16] and is part of the PULP solutions. The cores of the accelerators are RISC-V processing elements which are adapted to run the neural network efficiently. The software stack provided by HERO will be used for building the demonstrator and the generated results will be evaluated.
- The fifth solution will be a solution that utilizes a RISC-V [17] processor that supports Instruction Set Architecture (ISA) extension for data-level parallelism. The advantage of RISC-V processor is that it has a modular ISA, thus adding or removing a set of instructions belonging to one module will not affect the other ISA modules. Based on the specification of RISC-V there are two types of modules to deal with this form of parallelism called “P” and “V” extensions. “P” standard extension is a packet with SIMD instructions, while “V” extension covers the instructions for vector operations. “P” extension covers the SIMD instructions only for integer operations, while the float-point SIMD operations are dropped in favor of standardizing the “V” extension. Also, the size of the vector on which SIMD instructions can operate is limited. The “V” extension offers more flexibility since the number of registers that it uses to define the size of the vectors is not fixed and also the type of the elements within the vector. All parts of the code where data-level parallelism comes into expression will be executed on the vector part of the processor while the rest of the code will be mapped on the scalar part. Since inference of a neural network has properties of data-level parallelism it will by transformed from the compiler on a set of vector instructions. Extension of RISC-V processor with “V” or “P” instructions has not been considered as part of this project, but a possible implementation would be a good case to exploit the behaviour of the neural networks on such architectural solution.

All the development on Xilinx platform will be done with the help of the SDK provided from the vendor. For PULP platform we will use the 64bit version of the RISC-V processor that can run Linux operating system.

Data Collection

For training the model we have collected a small dataset of 100 labeled images with electronic components (capacitors, resistors and transistors). Details on the test data are not available yet.

Mechanism for AI transparency interactions

Transparency is not an issue for this use case as the system is not directly interacting with human users.

9 UC5: Increasing the safety of an autonomous train through AI techniques

The objective of UC5 is to apply Computer Vision and Artificial Intelligence techniques to increase the dependability and safety of the train operations by improving the following functionalities: platform detection, accurate train stop, and safe passenger transfer between train and platform.

9.1 Description of the use case

CAF Signalling, as a strategy to the future, is interested in applying CV&AI techniques to improve different *autonomous train operation functionalities*, such as precision stop, visual odometry, rolling stock coupling operation or person/obstacle detection-identification in railroads.

CAF Signalling will use the FRACTAL platform (more precisely Versal Edge Node) to execute the following functionalities developed in CV&AI field for autonomous train operation:

- **Automatic platform detection**: This functionality consists of detecting the platform area in the station (Figure 11). The application will be based on a Neural Network that identifies characteristic patterns allowing the detection of train platforms. Platform detection functionality will enable CV&AI- based automatic train approximation to accurately stop the train.

Figure 11 - Station Platforms.

- **Automatic accurate stop at door equipped platforms aligning the vehicle and platform doors**: It will perform precise positioning inside the platform area using visual patterns detection (PLATFORM END MARK PATTERN), identification, and tracking in order to reach an accurate stopping point and manage automatic train operation.

Figure 12 - Platform End Mark Pattern.

- **Safe passenger transfer**: It will manage automatic safe door enabling (ERMTS functionality) making sure the train is completely stopped in the platform area (using visual sensors) avoiding (a) door opening operation if the train and the platform doors are not precisely aligned and (b) door closing operation if any passenger is getting in/out the train.

In summary, two AI building blocks will be developed in UC5: the **Accurate Stop** application and the **Safe Passenger Transfer** application, that include both Platform Detection and Persons/Obstacles detection around train doors.

9.2 AI based Accurate Stop (UC5.1)

Description

The objective of this block is to compute an estimate of the distance from the train to the stop signals inside the train station platform. At the beginning and the end of the platform, there are specific train stop signals (station end lines) that are used by drivers to stop the train accurately. Figure 13 shows an example of the detection of the station end lines with the Yolo algorithm.

Figure 13 - Station End line detection with Yolo algorithm.

The Accurate Stop block is composed of two RGB high-resolution cameras on the front left and the front right of the train. As the train is entering the station, the technique to estimate the distance from the stop signals first locates the stop signals by both cameras in real-time and then uses an OpenCV stereo vision algorithm to triangulate the signal in both cameras and finally estimates the distance. This way, as the train approaches the stop signal, the application calculates in real-time the distance to the stop signal. Expected precision (with regard to the ground-truth of the neural network) is about +-1 meter, at an inference time of 10fps. In the future this application can evolve as an input to an automatic train stop operation (traction and brake commands, ATO-Automatic Train Operation functionality).

More in detail, the block is mainly composed of the well-known Neural Network YoloV3 for signal detection but also needs functions provided by OpenCV software package for stereovision functions. The distance estimation, which is needed to correct the aberrations introduced by the camera, is done as follows:

1. The cv::undistort function from the opencv2/calib3d library is used to correct the right and left images to correct the distortion of the lens and the origin position of the cameras, in this way we have two images from the same reference system.

2. The class OpenCV2 StereoSGBM is used to calculate disparity, i.e., it identifies a set of pixels in both images and calculates the distance between these sets in pixels. This distance in pixels is converted into the actual physical distance using the calibration parameters in an expression, specifically the focal length of the cameras and the baseline (distance between cameras). In general, if it is done for an entire image, it is computationally expensive, especially the SGBM matching because it identifies a group of n x n pixels in the right image with another group of n x n pixels in the left image, for which it needs to calculate a comparison for all the pixel combinations (n is the window size of the algorithm).
3. Finally, a simple triangulation allows calculating the distance to the stop point.

The building block mainly consists of (1) a YoloV3 TensorFlow Neural Network that has been trained off-line in a powerful server and exported into ONNX format, and (2) a set of stereo-matching functions from OpenCV to calculate the distance out from the predictions of the network. The ONNX network is imported into the Versal Platform via the Xilinx Vitis AI package. The whole Accurate Stop application is developed in C++ with the goal of reaching a 10 fps speed and an accurate distance calculation of -+1 meter.

Data Collection

This application consists mainly of a Yolo V3 Neural Network that has been trained to detect the stop signal (station end lines). For training, more than 80 hours of video from front left and front right cameras will be recorded. Frontal cameras produce video of a 1280x960 pixels resolution. From that video, more than 4.000 images will be selected and labelled (with station end lines) for the training, development and test data sets. With these images, Yolo V3 will be trained with a technique known as transfer learning. With this approach, the training starts from an existing solution (YoloV3 standard solution to detect 80 classes of objects) and the network is re-trained to detect the station end lines (train stop signals).

75% of the 80 hours are recorded from frontal cameras (1280x960 pixels resolution) for Accurate Stop application training and testing. From these videos, 4000 images will be extracted and labelled, and will be distributed for training and testing as follows.

- Training Set: 2250 images
- Development Set: 750 images
- Test Set: 1000 images

Labelling of Training and Development Sets is a mix of manual/semiautomatic mode, using CVAT (Computer Vision Annotation Tool [8]) with interpolation between frames that generates labels that will be corrected manually later.

Mechanism for AI transparency interactions

In the FRACTAL project, the output of the application, which is the distance to the stop point, will be used only as informative data for demo purposes. It is out of the scope of the project to couple this module with ATO (Automatic Train Operation) module to actually command the traction and brake of the train.

9.3 AI based Safe Passenger Transfer (UC5.2)

Description

The objective of this block is to make sure that when the doors of the train are opened or closed, (1) the train is inside the platform of the station and (2) there are neither people nor obstacles around the doors so they can be closed or opened safely. Figure 14 shows the platform of a station marked with a red line. Platform beginning signal is also seen (yellows pattern).

Figure 14 - Platform of the station marked with a red line.

The Safe Passenger Transfer block is composed of a high-resolution RGB camera in the front of the train pointing toward the rear so that the whole platform and the doors can be seen. A Yolo V3 network is in charge of detecting that the train is inside the platform and the the doors are free of people or obstacles and therefore they can open or close. Figure 15 shows the doors of a CAF unit.

Figure 15 - Train doors of a CAF unit.

In the future, this application can evolve as an input to an automatic train operation (commanding the train doors).

Safe Passenger Transfer component is a C++ application that mainly consists of a YoloV3 TensorFlow Neural Network that has been trained offline in a powerful server and exported into ONNX format. The ONNX network is imported into the Versal Platform via the Xilinx Vitis AI package. The application is expected to reach 10 fps.

Data Collection

This application consists mainly of a Yolo V3 Neural Network that has been trained to detect the stop signal (station end lines). For training, more than 80 hours of video from front left and front right cameras will be recorded. Front cameras pointing to the rear will have a resolution of HD (720p, 1.280 x 720 pixels) or Full HD (1080p, 1.920 x 1.080 pixels). From that video, more than 4.000 images will be selected and labelled (with station end lines) for the training, development, and test data sets. With these images, Yolo V3 will be trained with a technique known as transfer learning. With this approach, the training starts from an existing solution (YoloV3 standard solution to detect 80 classes of objects), and the network is re-trained to detect the station end lines (train stop signals).

25% of the 80 hours are recorded from the camera pointing to the rear (HD 720p, 1.280 x 720 pixels, or Full HD 1080p, 1.920 x 1.080 pixels) for Safe Passenger Transfer application testing. From these videos, 1000 images will be extracted for testing. In this case, the Yolo pretrained model is used, therefore all images are used for testing as no training is needed.

Mechanism for AI transparency interactions

The output of the application is an image with the detection of the platform, persons, or obstacles with the corresponding bounding boxes. It is out of the scope of the project to couple this module with ATO (Automatic Train Operation) module to actually command the opening or closing the doors. Figure 16 from [9] shows a typical detection of persons in a station platform by the Yolo network.

Figure 16 - Yolo detecting persons in the platform [9].

10 UC6: Elaborate data collected using heterogeneous technologies

The UC6 is focused on the development of a smart totem, equipped with smart sensors and actuators, collecting data and running AI based content analysis. The totem should have an impact on retail and shopping mall business, providing the customers with personalized advertisements, and product recommendations and guiding them towards products in the store.

10.1 Description of the use case

Nowadays, several emerging edge technologies are combined together within a new disruptive retail paradigm, called *Sentient Spaces*. It represents an advanced ICT-based space that has sensing capabilities, an *Artificial Intelligence* (AI) based brain to process information and data collected, and a large amount of actuation capabilities to interact with customers. It is a dynamic space able to adapt itself promoting products according to individual's preferences. Leveraging on it a double positive impact will be possible: the consumer will experience accurate guidance and product information and retailers will be much more efficient, making marketing more targeted and effective.

Figure 17 - Smart Totem.

In a sentient space, a smart totem is equipped with intelligent sensors and actuators, such as cameras, that collect data and implement AI-based content analysis providing output and actuations. It is then clear that such a totem could have a disruptive impact on retail and shopping mall business, providing personalized advertisements and product recommendations and driving customers towards products.

In such a context, the aim is to make these devices more accessible and faster to use. To this end, advanced AI approaches will be developed and deployed on the edge to process collected data to extract meaningful proximity information and a detailed understanding of its surroundings.

The inference is provided by a neural network and rule-based approaches, optimized for running on the embedded device installed in the smart totem. Data will be collected in terms of customers' gender and age range, and the effectiveness of marketing campaigns as customers' attention time for each content promoted. Not only video but also audio processing will be used to collect meaningful data that can be further elaborated to provide useful support for targeted advertisement and a personalized marketing strategy. Moreover, audio processing algorithms for in-store context awareness exploit audio signals to provide user-tailored information, contents, and services, delivering a shopping experience that meets consumer expectations. Both audio processing and video content analysis are based on innovative AI approaches that can be deployed on edge devices without requiring to upload data collected (i.e., video streams and audio signals) to a centralized cloud infrastructure.

In more detail, Figure 18 illustrates the target scenario.

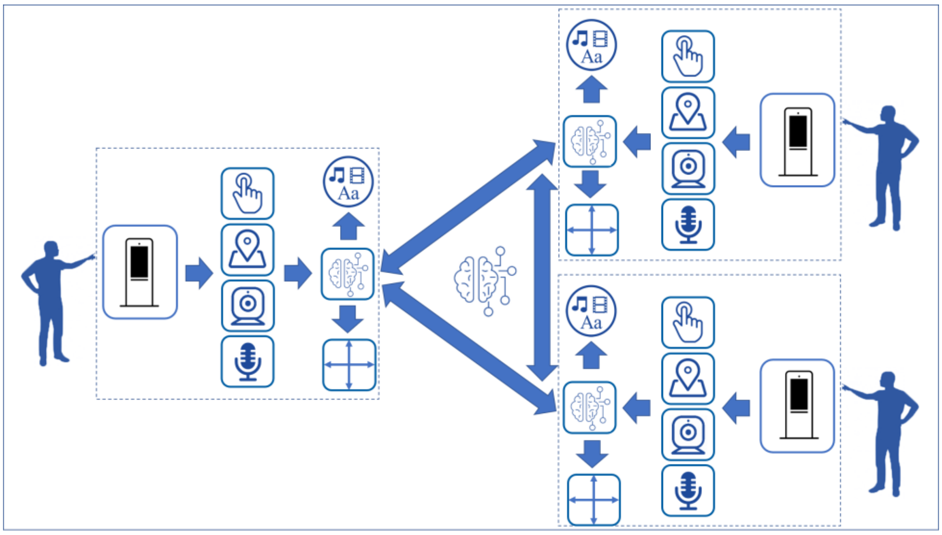

Figure 18 - Schematic representation of VAL-UC6.

In order to implement this UC, several AI-based building blocks, described in the following subsections, are needed.

10.2 People Density Estimator (UC6.1)

Description

The objective of this block is that of estimating the number of people in an area from the video scene acquired by a camera. Each node/totem will be able to estimate the number of people in view and compute a density level (LOW, MEDIUM, HIGH). In addition, this information can be transmitted through the MTTQ network to the other totems to allow for global inference of the density status in the mall. A highlevel architecture for this block is given in Figure 19.

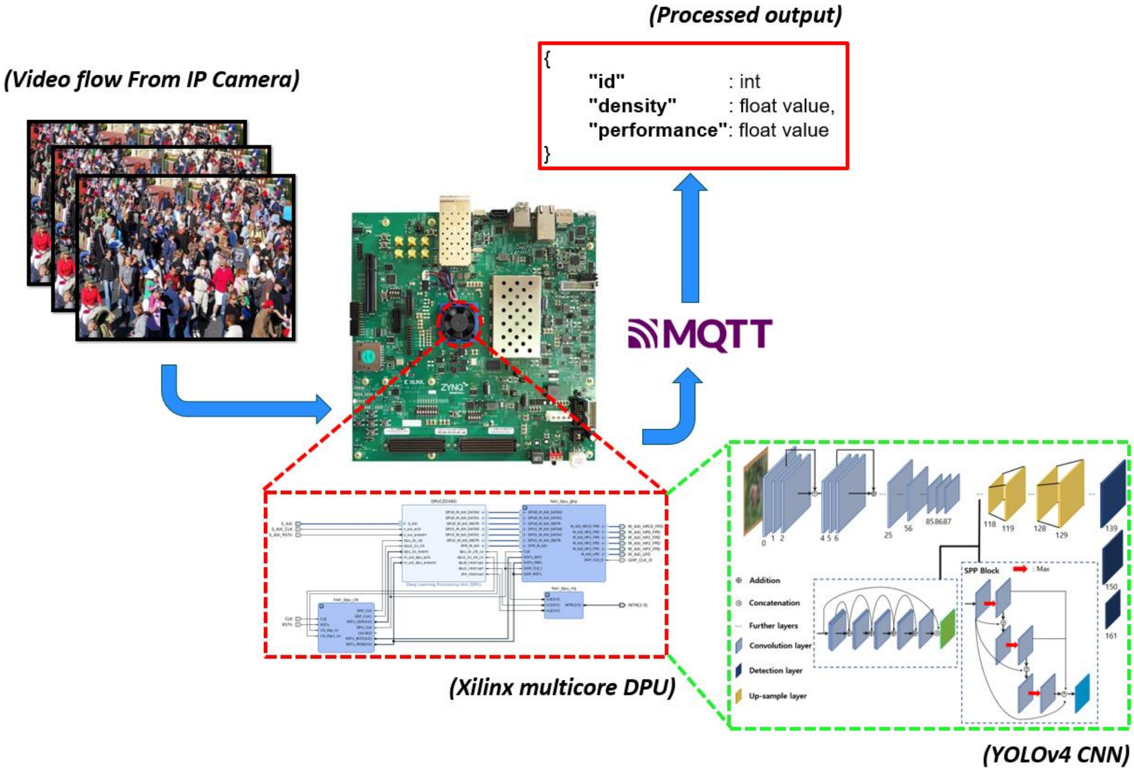

Figure 19 - High level processing pipeline for the density estimation task. Bounding boxes are computed using YOLOv4.

This task is implemented by YOLOv4, a state-of-art CNN (Convolutional Neural Network) used in computer vision (see D5.1 for details). This network performs object detection within each video frame. The results appear as bounding boxes over the video frame, containing a label for the object being detected. As illustrated in Figure 20.

Figure 20 - Yolov4 output on a video frame.

These bounding boxes are used to count the number of people in a stream video.

YOLOv4 is implemented in the Xilinx Vitis AI 1.4 framework and the block is deployed on the Xilinx Zynq UltraScale+ (ZCU102) with DPU engine.

Data Collection

The CNN used to implement the Density Estimator application is based on the YOLOV4 model, which has been trained using the standard Common Objects in Context (COCO) dataset. This dataset is split into 40000 images for training and 5000 images for the validation phase. The edge application is implemented using the C++ APIs provided by the Xilinx Vitis-AI tool suite in order to take advantage of the DPU engines deployed on the target Zynq UltraScale+. After the inference phase, where the number of people is computed by counting the bounding boxes, the raw number is passed to a stabilization phase. The aim of this second stage is to stabilize the people counting fluctuations detected by subsequent video frames. Finally, the people density information is forwarded using the MQTT protocol. When the Density Estimator application is activated, two different acquisition options are possible: i) LOCAL COMPUTATION: an IP camera flow is acquired using the Real Time Streaming Protocol (RTSP). The image flow is decompressed using OpenCV and fed to the YOLOV4 neural network; ii) REMOTE COMPUTATION: another option is to exploit the computational capacity of another FRACTAL node (People Detector). In this case, the input of the Density Estimator is pre-processed bounding boxes. Following this principle, only the stabilization phase is needed, with a consequent reduction of local computational requirements. All the computation required by the Density Estimator is local to edge nodes and does not require storing or exchanging information with the cloud.

Mechanism for AI transparency Interactions

The output of the density estimator is not directly provided to humans, so there is no need for Explainable AI.

10.3 Age and Gender Recognition (UC6.2)

Description

The objective of this block is that of estimating the age and gender of a person from the face. In particular, given a facial image, acquired through a camera, this block provides an estimate of the person's sex and an estimate of the age as a 10-years wide range (e.g. for a 45 years old man, the correct predictions would be (Male, {40, …,49}). At the heart of this block, there is a pair of VGG16 networks, one per task; those networks are identical, but the output layer. Figure 21 shows the processing pipeline of this block.

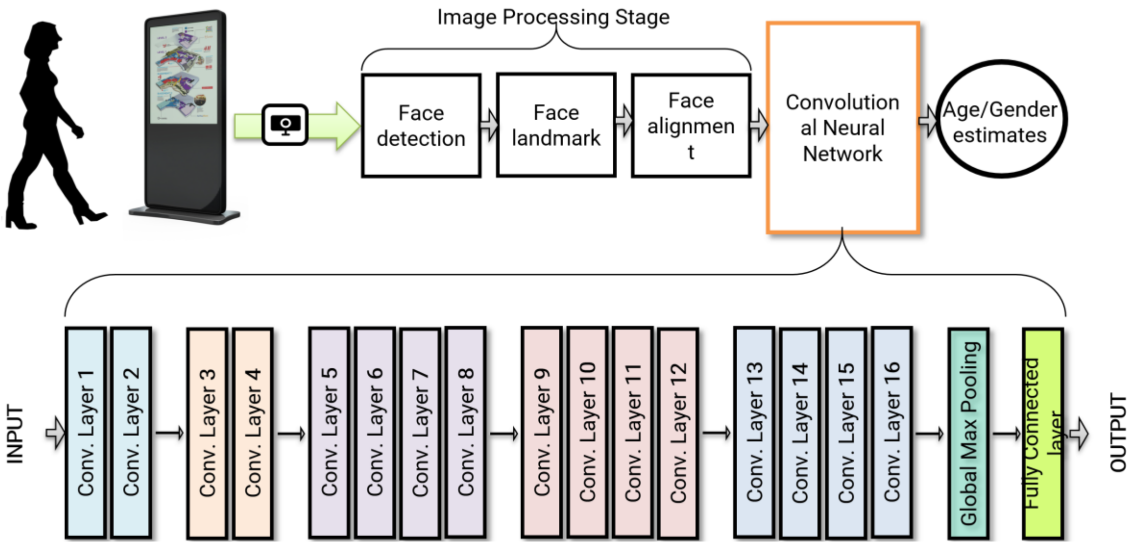

Figure 21 - Processing pipeline for the tasks of age and gender classification.

The VGG16 networks are implemented in Python by using Tensorflow and Keras and are trained *offline on a* host computer. Both networks are deployed on a Xilinx Zynq UltraScale+ (ZCU102) with DPU engine at production time. As for the deployment onboard, the framework Xilinx Vitis AI is employed. As a result, the deployed network is quantized and the effect of such quantization on the accuracy needs to be carefully evaluated.

Data Collection

The CNN at the hearth of this block, a VGG16 implemented using Python, TensorFlow, and Keras, is trained on a host computer using the MORPH II dataset, an established benchmark in computer vision research. This dataset is made of 55135 images with gender and age labels.

The dataset has been split into training, validation, and test respectively to train the model, optimize the hyper-parameters and test the performance. Table 1 reports the split sizes for the 2 tasks.

Task	Training	Validation	Test
Age	39700	4410	11025
Gender	12226	1358	3394

Table 2 - Dataset sizes for the tasks of Age (first row) and Gender (second row) Recognition.

At test time, the face image is acquired through the camera and processed as illustrated in Figure 20 above. The acquired image goes through the image detection phase, which consists of detection, landmark, and alignment. Then, the face image is fed into the CNN and the estimates are computed. All the computation is local to the node.

Mechanism for AI transparency Interactions

As described in 10.2, all the computations performed by this node are local to the totem and even more, internal to the processing nodes. Its output is not directly provided to a human, and therefore, there is no need of explainable AI for this building block.

10.4 People Detection (UC6.3)

Description

The objective of this block is that of determining the presence of one or more people in the proximity of the totem and send specific alarms based on the situation to the totem node. Figure 22 shows the results of the segmentation of the area around the totem into a *far-from* and *near-to* regions.

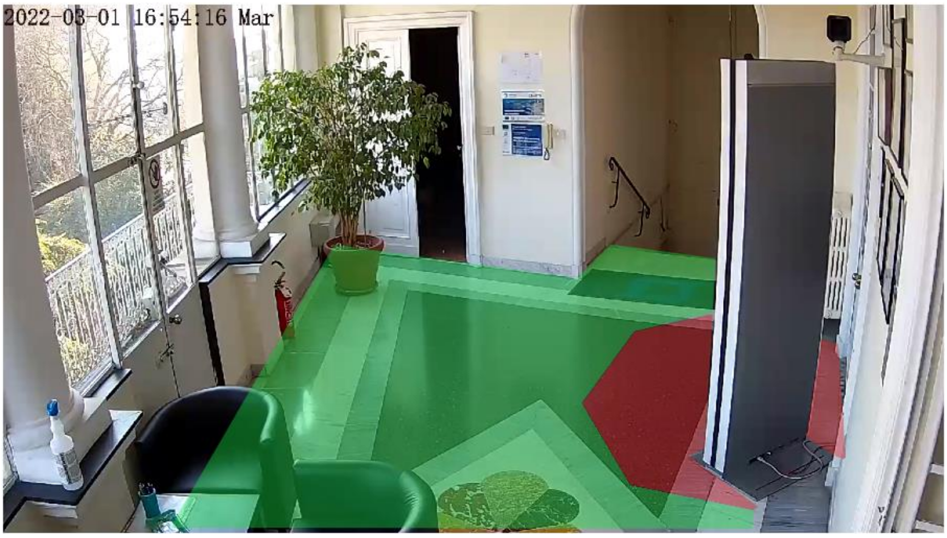

Figure 22 - Segmentation of a video frame acquired through a surveillance camera. Objects falling into these areas would trigger a proximity alarm.

This task is performed by combining a custom neural network based on YoloV5 which detects the presence of humans in a video frame, with another component that, analyzing the neural network outputs is able to determine if an object is near to or far from the totem and then generate the corresponding MQTT alarm. Figure 23 shows a high-level processing pipeline for this task.

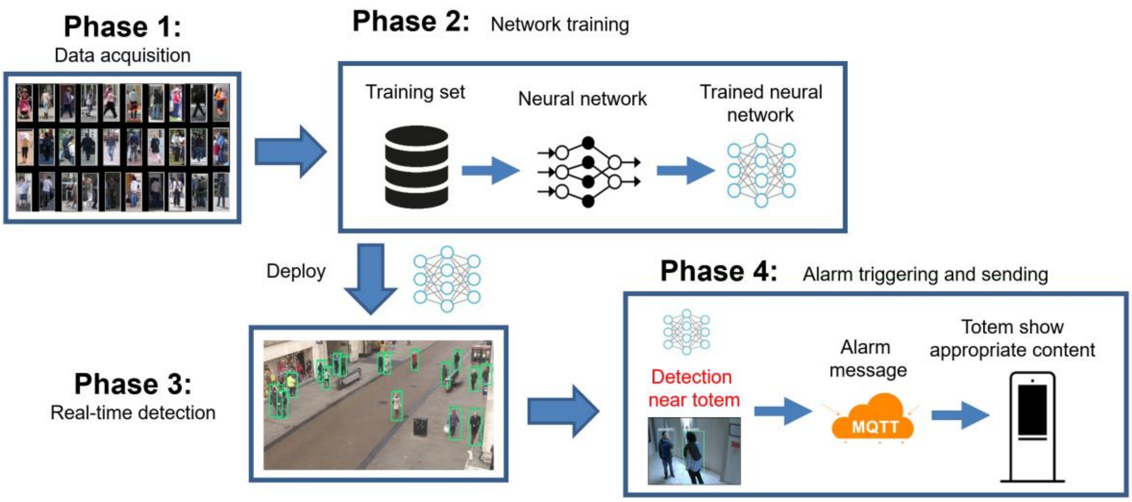

Figure 23 - People detection pipeline.

Different alarms are triggered based on the situation:

- **Alarm1**: A person enters the "*far-from*" area

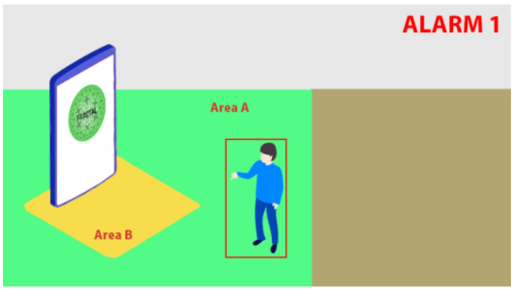

Figure 24 – Example of situation that triggers ALARM 1

- **Alarm2**: A person enters and stays for at least 2 seconds in the "*near-to*" area

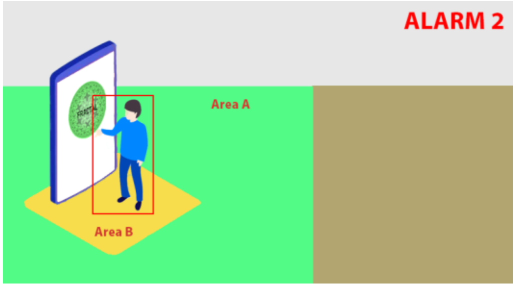

Figure 25 – Example of situation that triggers ALARM 2

- **Alarm3**: No person is detected for at least 5 second in the "*near-to*" area

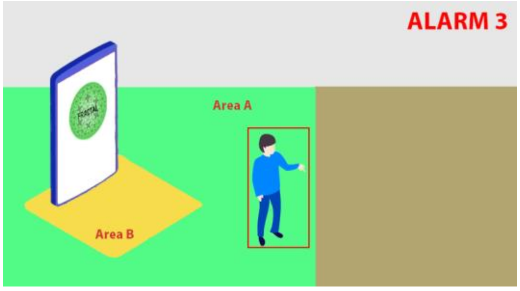

Figure 26 – Example of situation that triggers ALARM 3

- **Alarm4**: No person is detected for at least 5 second in the "*farm-from*" area

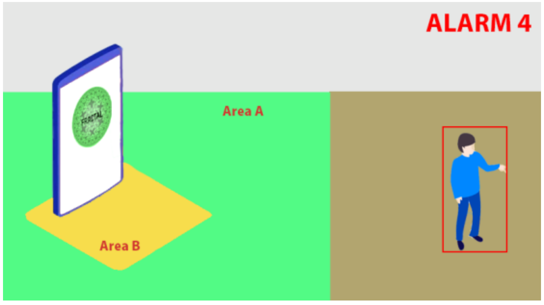

Figure 27 - Example of situation that triggers ALARM 4

Data Collection

The people detector component is based on a custom YoloV5n re-trained with transfer learning to go from the standard 80 classes to only pedestrian. The training sets are a mixture of open-source available datasets and collected data from Ip cameras: Coco dataset (~40000 datapoints); Open Images dataset (~40000 datapoints); and other additional data collected from Ip cameras (~10000 datapoints). Labeling of the collected data has been done manually, using VIA (VGG Image Annotator).

In the Test phase, the images are acquired by an Ip camera and processed in real-time by the component (Roof Node). Based on the intersection between the detected boxes and the predefined areas around the totem an alarm is triggered and a specific MQTT message is sent.

The People Detector computation is done entirely on the edge node and does not require storing or exchanging information with the cloud.

Mechanism for AI Transparency Interactions

Transparency is not an issue in this component since its computations are not directly used by human users.

10.5 Idiom Recognition (UC6.4)

Description

The Idiom Recognition (IR) module is used to perform automatic language recognition based on speech processing. The module outputs the language of the current speaker by capturing and analyzing an input audio stream.

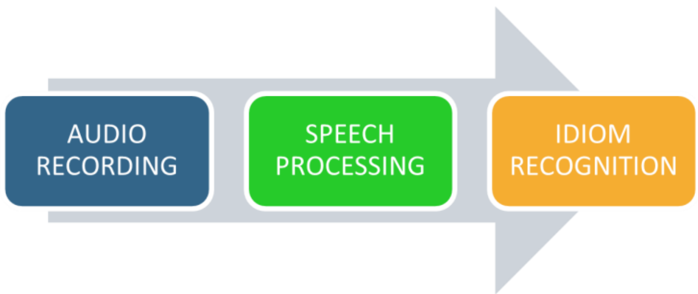

Figure 28 - The IR module process: from input to output.

The main objective of the IR module is to enable the Intelligent Totem with language recognition in order to automatically provide customized content to users in their spoken language (*e.g.*, *ENG* | *ITA* | *DEU* | *FRA* | *ESP*).

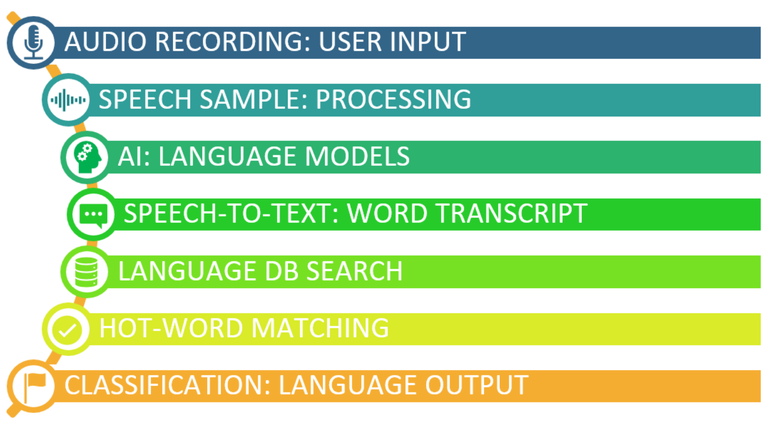

Figure 29 - Overview of the IR module workflow.

The IR module can either perform real-time recording or take as input an already stored audio speech file to process. Machine learning based Speech-to-Text (STT) solutions are then used to obtain the transcript of the audio file and a keyword search algorithm is employed to detect specific hot-words inside the speech sentence. Database search and matching are finally employed to identify the language corresponding to the recognized hot-word. Information about the detected language is used by the Intelligent Totem to provide contents to the user in his/her native language.

Data Collection

The IR module needs a speech sample to process and output the corresponding language. The specific approach leverages existing speech processing and STT libraries to obtain the transcript out of the recorded audio speech file. In particular, the Vosk speech recognition API is used to identify words inside the audio sample and ad-hoc language models are employed to obtain the speech transcript. Vosk-compatible language models are lightweight and pre-trained with thousands of hours of speech data. Once the transcript has been extracted, the recognized words are processed by a keyword detection algorithm to identify potential hot-words related to the supported languages. Pre-defined hot-words are stored in a database, which is hence queried for hot-word matching. A diagram showing data flow is reported in Figure 30.

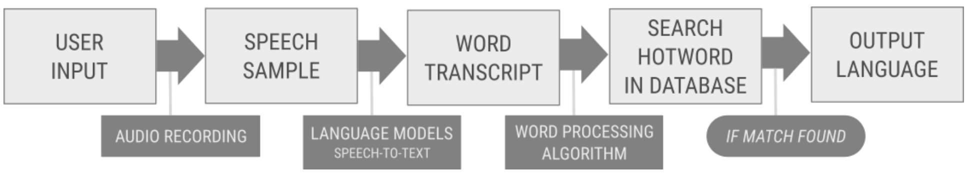

Figure 30 - IR Module data flow diagram.

The IR module output (i.e., the recognized language) is then sent to the Runtime Manager for interaction with other Intelligent Totem modules. In case of congestion, a recording of the audio speech sample is sent to the Runtime Manager for load balancing purposes.

Mechanism for AI Transparency Interactions

The IR module exploits transparent interactions with the user, employing data collected from user actions. In particular, the user is invited to speak by the Intelligent Totem, and the provided speech is recorded in order to be processed by the IR module. The outcome of data pre-processing, such as the audio transcript, are available to be presented to the user. The final output of the IR module is the recognized user spoken language, which is then used by the Intelligent Totem to provide customized contents. Hence, the AI results are available to the user in terms of language output and in case the system is not able to recognize the idiom, the user is informed and invited to speak again.

In the IR module, the interaction with the user is minimal. As described above, the output of the building block is the recognized user language. Therefore, the result is easily understandable by the user, who is able to visualize the transcript of its speech to assess if the IR module provided the correct processing and decision.

10.6 Rule-based recommendations (UC6.5)

Description

Rule-based Recommendations (RBR) has the goal of putting together information extracted by other modules providing indications about the contents to be shown by the totem. RBR uses as input the information derived from the video streams, namely age and gender, and by audio streams (idiom) in order to understand the best way to attract the attention of the customer. RBR could make use of both *a priori* rules, defined by the stakeholder, and rules derived from the data by means of machine learning algorithms. In order to combine *a priori* rules and data-driven models, a rule-based machine learning model has been selected for this task. In particular, the Logic Learning Machine model has been adopted in this task. This model makes use of past data to automatically derive the rules associating a profile (age, gender and idiom) to a suggested content. The criterion used for this goal is the level of attention. It is reasonable that the time that the customer stays in front of the totem, watching the content is an indicator of the level of interest in that content.

Data Collection

In the final system, data should be continuously gathered, providing datasets to be periodically processed to obtain new sets of rules. The update of the model is done in a cloud environment. For the first tests, given that complete data are not available, some data augmentation techniques are used to obtain a dataset of about one hundred records starting from a few cases that were available from previous experiments. This set of data is used to retrieve the first set of rules to be implemented in the system. This set will be improved and refined when real data are flowing from the system, making the recommendation system more and more accurate. The dataset includes the outcome of the interactions of some users with a video content provider.

The users, whose gender, age, and idiom were known, were undergone some videos and the attention was measured for each interaction. These tests involved 5 people with 4 videos. To increase the number of records to train the algorithms, a Synthetic Minority Oversampling Technique (SMOTE) data augmentation algorithm was used to obtain a dataset with 80 samples of about 20 users.

In particular, the information about the customer (age, gender, and idiom) is available together with an indication of the level of interest, measured as the time spent in front of the totem. Starting from this indicator, a binary value is introduced to flag interested people. In particular, for each user the content with the highest level of attention is considered the most suitable one. This variable is the output of a classification problem, while age, gender and idiom are the relative inputs.

Mechanism for AI Transparency Interactions

The recommendation is fully transparent since the mechanism that provides the suggestion is based on if-then rules.

Since RBR is providing the final decision, it is important that its output can be understood. Usually, the output is not provided to the customer, which is only the end-user, but it can be useful for the stakeholder (e.g., the responsible of the mall) in order to understand the behaviour of their customers. This is particularly important since decisions involve information that could be potential sources of biases and discrimination.

11 UC7: Autonomous robot for implementing safe movements

The objective of UC7 is to integrate the FRACTAL node on the SPIDER robot to enhance its functionalities.

11.1 Description of the use case

The "Smart Physical Demonstration and Evaluation Robot" (SPIDER), Figure 31, is an autonomous robot prototype. Within this use case, the Cognitive Edge Node developed in FRACTAL will be integrated into the autonomous robot SPIDER and evaluated against its applicability for performing computationally intensive relevant vehicle functions of variable complexity at the edge of the network (near the source of the data) while still being able to guarantee extra-functional properties (dependability, timeliness) for preserving its safety and security operational behaviour.

Figure 31 – Smart Physical Demonstration and Evaluation Robot (SPIDER).

The use case targets two main objectives:

- *O1: Co-execution of safety-relevant, security-relevant as well as AI-based tasks.*

- *O2: Implement fail-operational capabilities.*

The (user-task dependent) computationally intensive relevant vehicle functions might be task-dependent, for instance: enhanced AI-based computer vision and AI-based decision-making techniques, sensor fusion, and the creation of an occupancy grid. By performing computationally intensive data processing at the edge of the network, the SPIDER robot only sends aggregated data to the cloud, reduces communication bandwidth requirements, and thus fosters node autonomy by reducing the cloud functionality to management and control.

Two SPIDER functions will be deployed on the Cognitive Edge Node platform and use the fractality of the nodes for maintaining safety and security while providing high adaptability of the functions at the same time.

The collision avoidance function (CoA) is a safety-critical function and prevents collisions of the SPIDER with surrounding environment objects to avoid damage and most importantly human harm. The CoA uses four LIDAR sensors, visualized in Figure 32, which constantly measure the distance to environmental objects. If one of the objects gets too close to the SPIDER, an emergency brake is initiated.

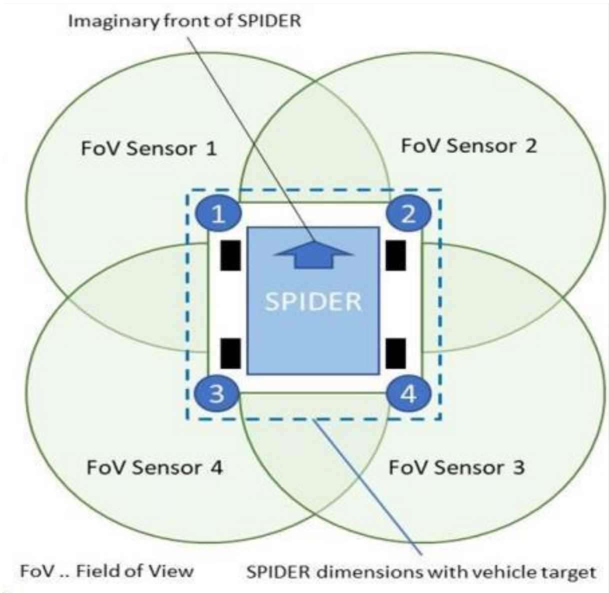

Figure 32 - Sensor setup for collision avoidance function of VAL-UC7.

The SPIDER is capable of omnidirectional driving (moving in all directions). Thus, a 360° environment perception with high accuracy of position and range is required. The SPIDER is intended to be operated in a closed environment like a proving ground, where the access of humans is prohibited. However, to ensure maximum safety, the CoA shall detect humans (or objects) approaching the SPIDER from an arbitrary angle and reduce speed or initiate an emergency brake if they come too close. Since the SPIDER can move on its own, the area in which the movement is directed is particularly safety critical. Therefore, if an environmental object is detected within this area – called the movement zone – an emergency brake shall be initiated.

The hardware platform to be used will be the medium performance node (RISC-V). VIF will run (user-task dependent) computationally intensive applications (like enhanced AI-based trajectory planning, or creation of an occupancy grid), on the FRACTAL Cognitive Edge Node's platform to demonstrate its applicability for the automotive market, where the applicability will be verified by the execution of predefined demanding tests, designed to stress the component. The separated implementation of the functions ensures that neither security issues nor erroneous decisions made by (uncertified) AI algorithms can impact the functional safety.

11.2 Safe AI decision-making (UC7.1)

Description

The goal of this task is to build a decision-making function for the autonomous navigation of the SPIDER in real-world scenarios by means of approaches from the field of Reinforcement Learning (RL). The function is trained to follow a predetermined path while avoiding stationary obstacles like trees or walls and moving obstacles like other vehicles or pedestrians along or near the path, as shown in Figure 33. For this purpose, a virtual driving environment is provided with which the (software) agent, i.e., a virtual model of the robot equipped with the current version of the controller, is interacting. During the training the agent is performing actions within the virtual environment and earns feedback – either a punishment or a reward - from the environment in terms of a numerical value.

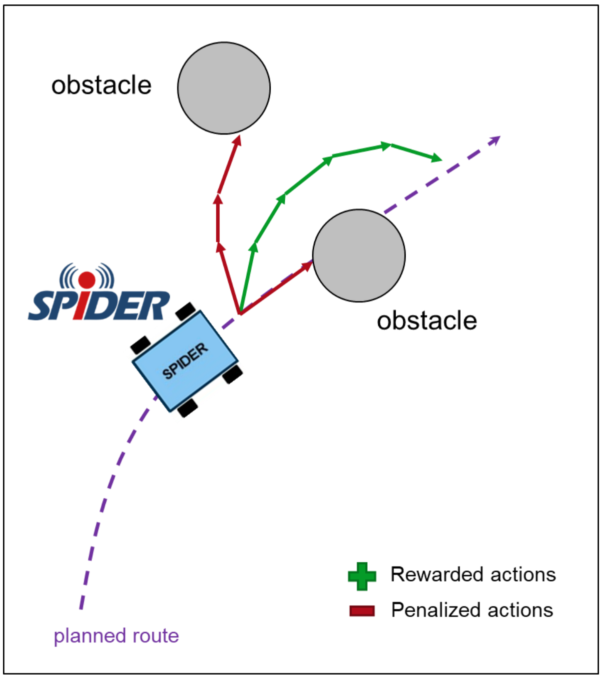

Figure 33 - Illustration of the RL-training process.

By using an RL algorithm, a control strategy maximizing the expected cumulative reward is derived. Thus, the rewarding approach must be shaped in such a way that the agent is encouraged to steer towards waypoints on the path and to keep distance from any kind of obstacles. As an output, the decision-making function gives longitudinal and lateral control values for the robot.

Data collection

The data for the training of the decision-making function is perceived by the continuous interaction of the agent with its virtual environment. The data available to the agent is composed of *costmaps* and vehicle states:

- A **costmap** is a 2D grid of cells indicating the free and occupied space in the robot's surrounding. Each cell contains a numerical value representing its state (free, occupied, unknown).[14,15]
- The **vehicle state** refers to a tuple comprising
  - the position of the robot,
  - the position of the waypoint which must be reached next,
  - the robot's linear speed,
  - the target speed of the robot, and
  - the orientation of the robot.

To reduce the complexity of the problem, costmaps are not directly fed into the ANN of the decision-making function. By means of a range-finding module based on a ray-casting approach, the distance between the robot and obstacles is extracted from a given costmap.

The vector of distances, as well as the vehicle state data, are scaled and only then fed into the ANN. Thus, the ANN is taking a 1D array of floating-point values as input.

Figure 34 schematically displays the data flow of the function.

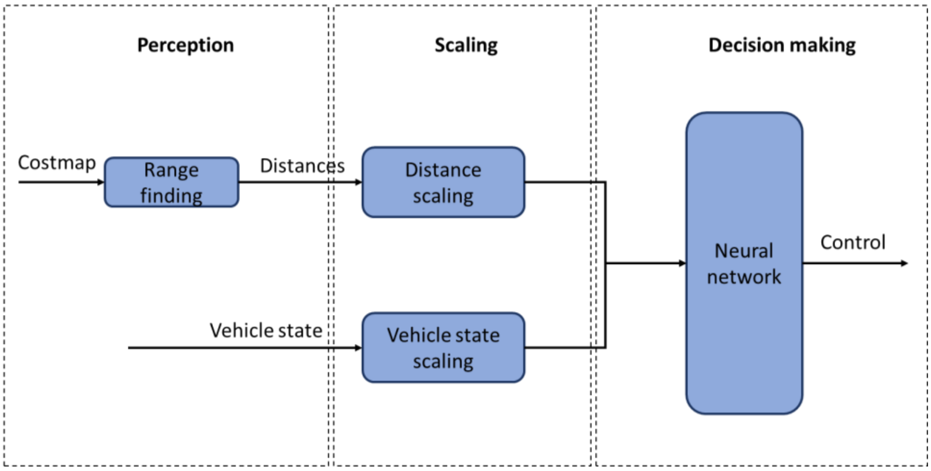

Figure 34 - Data-flow of SPIDER path tracking function

Mechanisms for AI transparency interactions

The AI block of the decision-making does not directly interact with humans. Furthermore, only data representing the state of the robot's environment is processed by means of the controller's ANN.

The results of the decision-making function are only to a certain degree explainable: if for example the robot's surrounding is free of obstacles, and the robot is driving towards the next waypoint, the behaviour of the controller aligns with the expected result. Thus, one can infer that the behaviour of the decision-making function is reasonable and transparent in this context. The same conclusion can be made, if an evasion maneuver is applied by the decision-making, upon detection of an obstacle. Apart from scenarios like these, it is difficult to explain the behaviour of the controller.

The end user cannot examine the pre-processing procedure, or information on the accuracy and efficiency of the decision-making function. For the assessment of it, during and after the training procedure tests are performed and metrics computed. These data can be used to monitor the training progress and as well to assess the AI-unit in terms of accuracy and efficiency.

Because there is no direct interaction between the AI block of the use case and humans, no analysis in regard to XAI is conducted.

12 UC8: Improve the performance of autonomous warehouse shuttles for moving goods in a warehouse

The objective of UC8 is to adopt Artificial Intelligence algorithms to the shuttle technology of an automated storage and retrieval solution and improve the overall performance regarding throughput, adaptivity, and reliability.

12.1 Description of the use case

FRACTAL components should give a new perspective on how functions and features can be solved differently and improve the three previously mentioned aspects for manufacturers and customers. The goal is to improve the warehouse throughput, considering that delays in warehouse operation are critically undesirable, since it has a domino effect on the supply chain. The handling, storage, and retrieval of warehouse goods by automated shuttles are optimized using Artificial Intelligence techniques. AI will optimally organize and analyze the masses of generated data, in order to improve the warehouse throughput.

The automated shuttle systems shall operate as agents of a swarm intelligence to improve its reliability and adaptivity. The need for a central coordinator shall be eliminated as communication failures could destabilize the system, the targeted swarm intelligence solves the bulk order fulfilment by its own but requires additional computational capabilities in the field level of automation, where these devices are located. To gain this computational capability and host AI functions, the shuttles shall utilize the FRACTAL edge nodes. Real-time information (e.g., diagnostics, battery health, task) collected during shuttle operation is registered in a local database and used for environmental statistics. Therefore, the FRACTAL edge nodes will be suitable to satisfy the computational requirement at low energy by resource optimization at the application and node level.

The shuttles will be edge-computing nodes that will process real-time information at a very high speed through integrated filters. Task handling will be shifted from a material flow computer to shuttles with local decision capabilities (e.g., routing and sequencing). The system shall minimize human interruptions resulting from faults.

The warehouse system shall utilize new data flows (via deep learning techniques) to optimize the warehouse throughput.

The following AI features are desirable:

- Establish uninterrupted communication between the shuttles by exploiting machine learning techniques on the aggregated data obtained from signal connectivity monitoring.
- Predictive maintenance: The task that previously led to failure or low performance will be optimized and corrected to improve the warehouse availability.

- Adaptive system: A shuttle system that will adapt independently to new situations within the warehouse.
- Power optimization and improved storage strategy: By optimizing the location of high-velocity goods, while spreading them out in an optimal way to minimize congestion and improve the retrieval efficiency. Machine learning will be exploited to establish the desired optimal values.
- Route optimization: Aggregated data of route-patterns and delivery efficiency will be exploited a through AI application to obtain a higher throughput for the warehouse.
- Pick-up order (Productivity): Using supervised learning techniques with inputs – accumulated pickup list to schedule an optimized system-directed picking (Output – result of the best pattern).
- Defined bulk processing of orders. Bulk information, including expected timing, is given to a SWARM that finds the solutions according the specified optimization strategies e.g., optimized throughput for recurring items or energy optimization by weight distribution for shorter routes based on probabilities.

12.2 Shuttle orchestrator (UC8.1)

Description

The meta-scheduler component from WP4 will be adapted to the application layer of the edge node and orchestrate the shuttles regarding optimized path and warehouse storage strategies. Most of the desired AI features from the Use Case description can be accomplished by that scheduler, except for the communication and the predictive maintenance part. More specifically, the component task consists in generating shuttles depending on the situation, position, and incoming orders. In order to achieve this, the orchestrator will acquire the position of each shuttle and their status (operational or disturbance). Moreover, it will also get orders from the overlaying system. The inference will be performed by a deep network (initially ANN, preferred GNN, but as of today not feasible in Versal with Vitis).

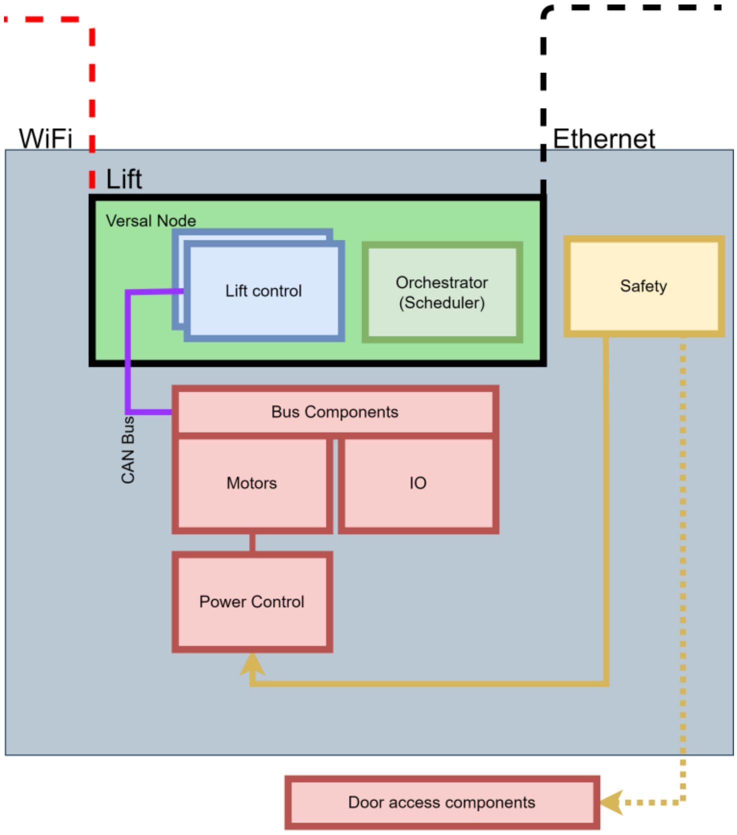

Figure 35 - Orchestrator location.

Data collection description

The model will be trained offline using data collected from the test setup, the simulation and self-generated by the scenario generator. Details on test data are not available yet.

Mechanism for AI transparency interactions

This component does not present any transparency or explainability needs.

12.3 Object detection (UC8.2)

Description

Object detection is used for the shuttle to recognize human bodies and other obstacles. This approach will provide an advanced safety concept and a more user-friendly environment for maintenance staff. The reason for this approach is the strong drop of throughput, in case of malfunction of a shuttle and the long access time for maintenance staff to fix it due to the waiting period of the safety door control. Additionally, uninterrupted communication will not be necessary anymore, as the shuttles get an intrinsically safe design with a self-sufficient solution. So, the rest of the system remains operational and will not be partially shut down to fix small issues.

Data will be pictures acquired through cameras and inference will be performed with YOLO.

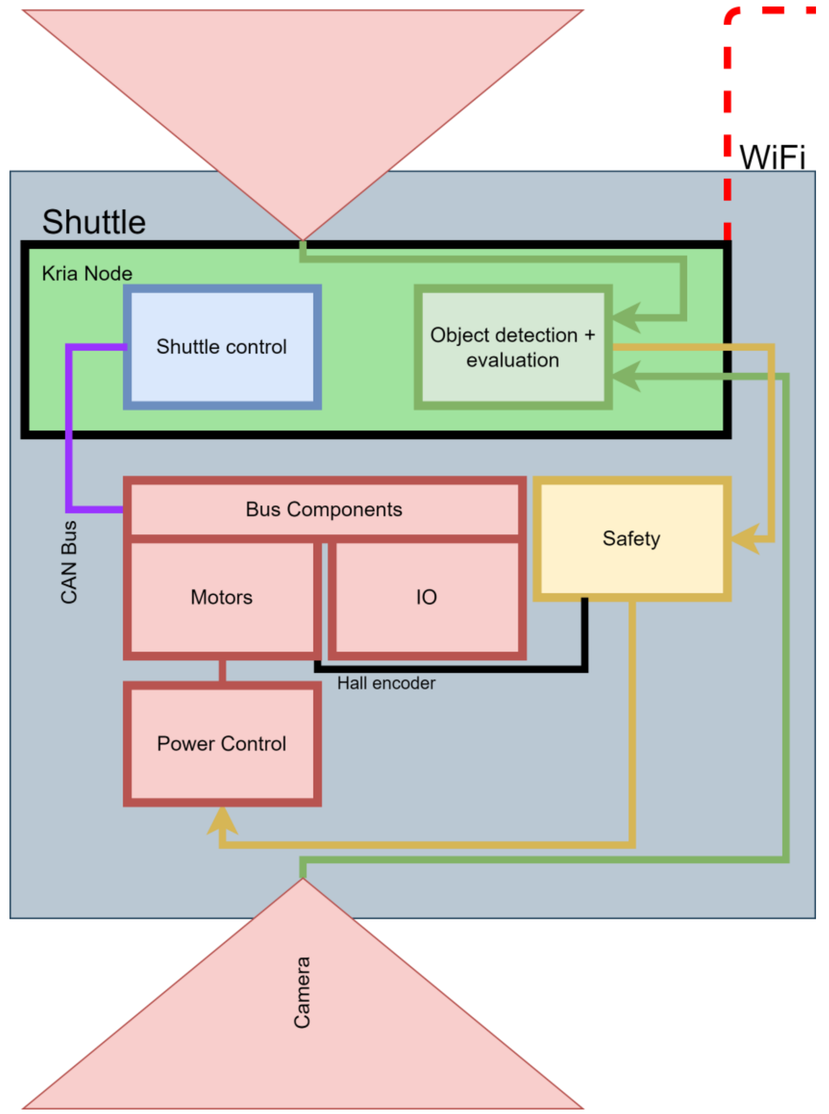

Figure 36 - Object Detector Location.

Data collection description

For human body detection, an available dataset will be taken for the first testing. The obstacle detection will be added later after more data will be recorded from the test setup as we are not expecting ready data. The model will be trained offline.

Mechanism for AI transparency interactions

This component does not present any transparency or explainability needs.

13 Methodological Study: Automatic Iris Disease Diagnosis

13.1 Description of the study (MT)

This experimental study is dedicated to the design, development, and empirical evaluation of Deep Learning (DL) algorithms for the automatic diagnosis of iris pathologies from high-definition ocular fundus images. Such algorithms will be also empowered with an *explainer*, i.e., an algorithm that will provide human-interpretable explanations about the algorithm predictions.

Introduction

Automatic diagnosis systems are becoming widespread due to the phenomenal success of AI algorithms and technologies. Such methods, and in particular *Deep Neural Networks*, proved to be capable of reaching super-human performances in many prediction tasks involving image analysis. Medical imagining, i.e., the area of study dedicated to the automatic analysis of complex medical images acquired by modern machinery during medical exams, has hugely benefited from DL algorithms. Automatic diagnosis is one of the most important tasks in modern medical imaging and roughly consists in designing algorithms capable of identifying one or more pathologies, if any, from the medical data of a patient. Recent studies proved that through DL algorithms, it is possible to effectively identify several important iris pathologies, such as (Glaucoma, Diabetic retinopathy (DR), Macular edema (ME), and Age-related macular degeneration (AMD)) from ocular fundus images. Such algorithms operate as a doctor inspecting manually such images to produce a diagnosis.

This study is focused on the identification of the DR and more specifically its level of development according to the standard retinopathy classification in 5 levels:

- 0 = None
- 1 = Mild DR
- 2 = Moderate DR
- 3 = Severe DR
- 4 = PDR (Proliferative DR).

In the context of this study, the DR identification problem has been formulated as a multi-class classification problem, where given a high-resolution eye fundus image, the algorithm outputs the associated level of retinopathy. The multi-class classifier has been implemented with an ensemble of Convolutional Neural Networks (CNN) according to the *all-pairs* reduction. With this approach, for each pair of classes (e.g. (0,1)) a binary classifier is trained to distinguish between the two. Once all the (5*4)/2 = 10 classifiers are trained, every time an image needs to be classified, all classifiers are fed with that image, and the class with the greatest number of wins is output (with ties broken according to the lexicographical order).

Implementation Details. EfficientNet B7 with the cross-entropy loss has been selected as a binary classifier. This is a state-of-art network for image classification with 66 million parameters. The networks have been implemented in Python using the Pytorch library. The learning rate has been set to 0.1 and the batch size to 2. Due to the size of the network, training and tests have been performed on Google Colab using a 16GB Nvidia Tesla P100 GPU on Google COLAB. All the preprocessing stages have also been implemented in Python.

Data collection

In this study, two real-world datasets have been considered: MESSIDOR II [11] and IDRiD [12]. MESSIDOR stands for Methods to Evaluate Segmentation and Indexing Techniques in the field of Retinal Ophthalmology (in French). The 1200 eye fundus color numerical images of the posterior pole of the Messidor database were acquired by 3 ophthalmologic departments using a color video 3CCD camera mounted on a Topcon TRC NW6 non-mydriatic retinograph with a 45-degree field of view. Images were captured using 8 bits per color plane at 1440x960, 2240x1488, or 2304x1536 pixels. 800 images were acquired with pupil dilation (one drop of Tropicamide at 0.5%) and 400 without dilation. The 1200 images are packaged in 3 sets, one per ophthalmologic department. Each set is divided into 4 zipped subsets containing 100 images each in TIFF format and an Excel file with medical diagnoses for each image. Every image is classified into one of four classes (0-3), where 0 stands for 'normal' (without DR), and the other three classes represent DR severity.

The MESSIDOR-2 is an extension of the MESSIDOR-1 [10,13] and it contains 874 examinations (1748 images with the same resolutions as MESSIDOR-1). The main difference between MESSIDOR-1 is that the images are classified into one of 5 classes (0-4):

- 0 = None
- 1 = Mild DR
- 2 = Moderate DR
- 3 = Severe DR
- 4 = PDR (Proliferative DR)

Of these 1748 images, 4 were classified as ungradable and therefore don't have the corresponding DR classification label. That said, the images that can be used with a supervised approach are 1744. A sample image from this dataset is reported in Figure 37.

The class breakdown in MESSIDOR-2 is the following:

- 1017 images labeled with class 0
- 270 images labeled with class 1
- 347 images labeled with class 2
- 75 images labeled with class 3
- 35 images labeled with class 4

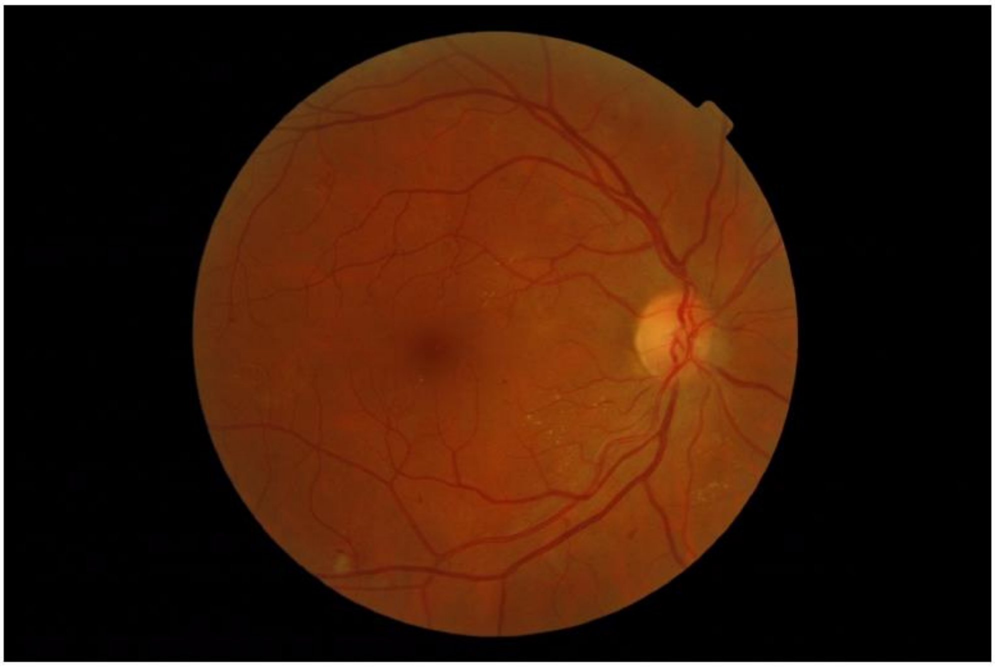

Figure 37 - Sample image from MESSIDOR-2.

The IDRiD dataset (Indian Diabetic Retinpathy Image Dataset) contains fundus images that were captured by a retinal specialist at an Eye Clinic located in Nanded, Maharashtra, India. From the thousands of examinations available, we have extracted 516 images to form our dataset. Images were acquired using a Kowa VX-10 alpha digital fundus camera with 50-degree field of view (FOV), and all are centered near the macula. The images have a resolution of 4288×2848 pixels and are stored in jpg file format. The size of each image is about 800 KB. The medical experts graded the full set of 516 images with a variety of pathological conditions of DR. Grading for all images is available in the CSV file. The diabetic retinal images were classified into separate groups according to the International Clinical Diabetic Retinopathy Scale: classes from 0 to 4 (the same as MESSIDOR-2).

The class breakdown in IDRiD is the following:

- 168 images labeled with class 0
- 25 images labeled with class 1
- 168 images labeled with class 2
- 93 images labeled with class 3
- 62 images labeled with class 4

Preprocessing: To bring all the images at the same computationally feasible resolution, we resized all the images at 456x456 resolution.

The pre-processing phase has two main steps:

1. Cropping: This is needed to bring the images to a squared shape.

- In the case of MESSIDOR2, since its images have has three different resolutions:2240x1488;
- 1440x960;
- 2304x1536.

The images are, respectively, cropped to:

- 1488x1488;
- 960x960;
- 1536x1536.

For IDRiD instead, since its image have one resolution, 4288x2848 and the images are cropped to 3475x3475. Additionally, for IDRiD images are also padded (in the upper and lower part) with black pixels using the opencv function *copyMakeBorder*.

2. Resizing: Once we have squared images, they have been reseized to the common resolution of 456x456.

Mechanism for AI transparency interactions

As part of this study, an explainer has been developed. Such explainer is based on *Local Interpretable Model-Agnostic Explanations* (LIME) [4], a well-established technique that explains the predictions of any classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction.

The algorithm ensures the following properties about explanations:

- Interpretable: explanations need to use a representation that is understandable to humans, regardless of the actual features used by the model.
- Local fidelity: it must match how the model behaves in the vicinity of the intended instance. We note that local fidelity does not imply global fidelity: characteristics that are important globally may not be important locally.

Explaining Diseases Prediction:

We propose to use theLIME methodology to generate the explanations of the predictions carried out by a previously deep trained network, providing then a fully XAI system for automatic diagnosis.

There are several pathologies that can be diagnosticated via the analysis of eye fundus images, including the diabetic retinopathy.

Such diseases has are characterized by peculiar, although not deterministic, symptoms.

The typical diagnosis pipeline starts with specialists analysing the fundus screening to assess the quantity and severity of the peculiar symptoms. In the context of DR, the most common symptoms are:

1- Microaneurysms: limited oxygen supply results in unusual dilation of retinal capillaries. These bulging of capillary endothelia linings often appear in the form of small sac-like structures named *microaneurysms*. The fragility of capillary walls is the main reason behind the development of *microaneurysms*.
2- Cotton wool spots: oxygen supply to certain retinal areas may completely close off due to blockage in arterioles. Consequently, large regions of retina become completely deprived of oxygen and result in emanation of fluffy white patches identified as cotton wool spots or soft *exudates*.
3- Hemorrhages: the blockade in arterioles may instigate a pressure build up within the vessels. A significant amount of pressure could burst the vessels and result in origination of *hemorrhages.*
4- Hard Exudates: they appear because of leakage of fat sand proteins along with water from abnormally permeable walls of retinal vessels. Mostly hard exudates appear on the outer layer of the retina individually, in the form of patches, or surrounding *microaneurysms* in the form of a crescent.

Using a CNN model, it is possible to classify different the DR levels from the eye fundus images, however, it is not always easy to interpret the produced predictions. To ensure that the final architecture is reliable and responsible, it is useful to generate explanations for the results. There are several advantages in this. First, the specialist can supply a final diagnosis from the results of LIME. Furthermore, it is also possible to improve the network by analyzing the prediction by going back to the features that led to a wrong prediction.

LIME generates the explanations for the classification of images by highlighting only the super-pixels with positive weight with respect to a specific class, since they give proof of why the model might think that the class may be present. Figure 38 shows the high-level view of the LIME operation.

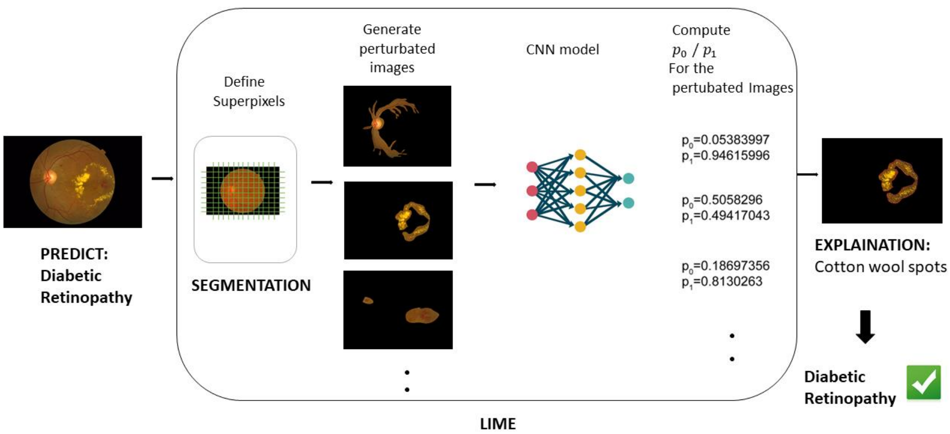

Figure 38 - Diagram of the Locally Interpretable Model-agnostic Explanations (LIME) algorithm in four steps.

The image, whose prediction you want to know, is passed as input to the LIME. Insidea segmenter has the task of defining the super-pixels of the image or creating masks that capture the main features of the image. In this example the masks obtained are blood vessels, cotton wool spots, and *hemorrhages*. Subsequently, the obtained masks are perturbed, (i.e., the pixels are altered), turning them on and off randomly. The perturbed images are passed in input to the network used for classification before the LIME. The network returns the probabilities of perturbed images with respect to the target image (p_0). These probabilities and perturbed images were presented to a regression model that estimated the positive or negative contribution of each super-pixel to the classification. At this point, it is possible to generate the explanation in two ways:

- Mask that outlines with different colors the pixels with greater and lesser weights (in this example the black background indicates low weight).
- Heatmap visualization with color-bar that shows the values of the weights.

As an example, in Figure 39, it is reported the prediction of the implemented classification model along with the explanation provided by LIME. LIME explanations come in the form of patches over the classified image. The pixel that was more important in the classification, has been included in the patches.

Figure 39 - An application of the trained explainer. In the output image, patches highlight the region of the image that most contributed to its classification.

14 AI-Human-AI transparency interactions results

This section presents a synthetic and summarized evaluation of the transparency mechanisms employed by the building block developed in the context of the FRACTAL project. The criteria considered in this evaluation are the same analyzed in sections 5 to 14.

The building blocks are evaluated on the basis of their transparency as assessed by the following criteria:

- Data collection process transparency.
    - *Training data*: description of the training data used to learn the AI model. This criterion can take three values: **preliminary**, corresponding to a preliminary description with little to no details; **partial**, corresponding to a partial description with some to almost all details; **complete**, corresponding to a complete description enabling reproducibility.
    - *Production data*: description of the data collection process at production time. This criterion can take three values: **preliminary**, corresponding to a preliminary description with little to no details; **partial**, corresponding to a partial description with some to almost all details; **complete**, corresponding to a complete description enabling reproducibility.
    - *Pre-processing*: description of the pre-processing performed by the algorithm (if applicable) before any prediction is made. This criterion can take three values: **preliminary**, corresponding to a preliminary description with little to no details; **partial**, corresponding to a partial description with some to almost all details; **complete**, corresponding to a complete description enabling a full understanding of the pre-processing pipeline; **NA**, corresponding to Not Applicable in case pre-processing is not required.
- Transparency and interpretability.
    - *Data transmission*: binary flag denoting whether the data are transmitted to other entities or not; **TDB**, if not defined yet.
    - *Interpretability*: what type of XAI technique is applied (if applicable) to make the results interpretable by a human user. This criterion can take three values: **explainer (EX)**, corresponding to the case of interpretability obtained via an explainer; **interpretable model (IM)**, corresponding to an AI model that is natively interpretable; **interpretable output (IO)**, corresponding to a task whose output, by definition, is already human-interpretable; **NA**, corresponding to Not Applicable in case interpretability is not required.

UC	AI BB	Training data	Production data	Data transmission	Preprocessing	XAI
1	UC1.1	complete	complete	Yes	NA	IO
1	UC1.2	preliminary	complete	Yes	complete	IO
2	UC2.1	partial	partial	Yes	preliminary	NA
3	UC3.1	preliminary	partial	Yes	preliminary	NA
4	UC4.1	partial	preliminary	TBD	preliminary	NA
5	UC5.1	complete	complete	No	complete	NA
5	UC5.2	complete	complete	No	complete	NA
6	UC6.1	complete	complete	No	complete	NA
6	UC6.2	complete	complete	No	complete	NA
6	UC6.3	complete	complete	No	complete	NA
6	UC6.4	complete	complete	Yes	complete	NA
6	UC6.5	preliminary	partial	Yes	preliminary	IM
7	UC7.1	complete	complete	No	complete	NA
8	UC8.1	preliminary	preliminary	No	preliminary	NA
8	UC8.2	preliminary	preliminary	No	preliminary	NA
MS	MS1	complete	complete	No	complete	EX

Table 3 – Summary of the transparency evaluation.

Results are shown in Table 2. Each row refers to a specific AI building block (AI BB). The building blocks with the most prominent transparency features are highlighted in green. These blocks have advanced transparency features in terms of their data collection processes and the interactions with human operators. Beyond a clear description of the data collection processes, most of these blocks have also XAI features. In particular:

- UC1.1 provides segmented crack images which are naturally interpretable;

- UC1.2 generates alerts whenever a risky situation is perceived. Alerts are already a quite semantically advanced output which is naturally interpretable by the end-user.
- UC6.5 is a rule-based system, which as we have discussed in section 4 is naturally interpretable.
- MS adopts LIME as an explainer for the automatic diagnosis generated from the eye fundus images.

Here it is stressed that XAI features are not a necessary condition for transparency. Indeed, in the case of UC6.4, Idiom Recognition, the model's predictions are not explainable nor is the model itself (which is generated by off-the-shelf NLP libraries). Nevertheless, the system has a fully transparent and interactive interaction with its human user as described in the Section 10.5.

On the other hand, it appears that the large majority of the developed solutions do not have specific transparency requirements to fulfill.

15 Conclusions

The objective of task 5.3 is that of designing and enhancing AI tools and solutions in order to fulfill the requirements of the FRACTAL use cases. This deliverable describes the proposed AI building blocks and investigate the details of the non-functional requirements related to the transparency and interpretability of the AI models. With this respect, it is observed that these requirements intersect only a (relatively) small subset of the proposed use cases. Transparency is obtained by means of a clear description of the data collection processes, which includes details of the training data, production data and data transmissions and through the use of pre-processing and processing mechanisms that either make the user of the system aware of the computation being performed or make the predictions of the system interpretable. As described in this deliverable, this is obtained with different approaches depending on the particular building block. Even when transparency is not the main requirement, a clear description of the data collection processes and the pre-processing performed by the system enables reproducibility of the results improving the overall quality of the proposed contributions. To conclude, this deliverable presents a description of the use cases and the associated building block from the perspective of the data collection processes and algorithmic transparency. This is somehow orthogonal to D4.5 where compliance with the legislation of the GDPR is checked. Together these deliverables provide an assessment of the transparency of the solutions developed within the context of the FRACTAL project.

16 Bibliography

[1]. European Parlimentary Research Service. *A governance framework for algorithmic accountability and transparency*. 2019.

[2]. https://gdprhub.eu/Article_22_GDPR.

[3]. Burack, J. *AI and GDPR: Data Protection and Transparency in Focus*. 2020.

[4]. Riberio, M.T., Singh S., Guestrin C. *Why should I trust you? Explaining the predictions of any classifier.* KDD, 2016.

[5]. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K. Klaus-Robert Muller. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning.* Springer 2019.

[6]. Denis Rothman. *Hands-On Explainbale AI (XAI) with Python*. Pack Publishing Limited 2020.

[7]. https://github.com/yhlleo/DeepCrack.

[8]. https://cvat.org

[9]. Abhinand, A., Mulerikkal, J., Antony, A., Aparna, P.A., Jaison, A.C. *Detection of Moving Objects in a Metro Rail CCTV Video Using YOLO Object Detection Models*. Lecture Notes on Data Engineering and Communications Technologies, 2021.

[10]. https://adcis.net/en/third-party/messidor2.

[11]. https://adcis.net/en/third-party/messidor.

[12]. https://kaggle.com/datasets/mariaherrerot/idrid-dataset.

[13]. Decencière E., Zhang X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., Charton, B., Klein, J.C. *Feedback on a publicly distributed image dataset: the MESSIDOR database*. Image Analysis & Stereology, 2014.

[14]. https://www.mathworks.com/help/driving/ref/vehiclecostmap.html

[15]. http://wiki.ros.org/costmap_2d

[16]. Andreas Kurth, Pirmin Vogel, Alessandro Cpotondi, Andrea Marongiu, Luca Benini. *Hero: Heterogeneous Embedded Research Platform for Exploring RISC-V Manycore Accelerator on FPGA*. CARRV 2017.

[17]. The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA, 20191213.

17 List of figures

List of tables

18 List of Abbreviations