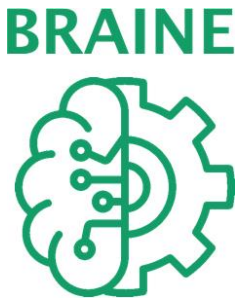


BRAINE - Big data Processing and Artificial Intelligence at the Network Edge

Project Title:	**BRAINE - Big data Processing and Artificial Intelligence at the Network Edge**
Contract No:	876967 – BRAINE
Instrument:	ECSEL Research and Innovation Action
Call:	H2020-ECSEL-2019-2-RIA
Start of project:	1 May 2020
Duration:	43 months

Deliverable No: D5.7

BRAINE smart cities use case

Due date of deliverable:	30 November 2023
Actual submission date:	25 November 2023
Version:	1.0

Project funded by the European Community under the
H2020 Programme for Research and Innovation.

Project ref. number	876967
Project title	BRAINE - Big data Processing and Artificial Intelligence at the Network Edge

Deliverable title	BRAINE smart cities use case
Deliverable number	D5.7
Deliverable version	Version 1.0
Contractual date of delivery	30 November 2023
Actual date of delivery	25 November 2023
Deliverable filename	D5.7 – BRAINE smart cities use case
Nature of deliverable	Report
Dissemination level	PU
Number of pages	33
Work package	WP5
Task(s)	T5.4
Partner responsible	NEC, ITL, SMA, MAI,

Author(s)	Antonino Albanese (ITL), Marco Beccari (ITL), Federico Civerchia (SMA), Luca Maggiani (SMA), Marcus Nordström (MAI), Roberto Bifulco (NEC)
Editor	Roberto Bifulco (NEC)

Abstract	This technical report delivers the detailed information about the development of a smart city platform designed to integrate and utilize the features available on the BRAINE platform to improve provide a feature-rich multi-tenant supporting environment to provide video transcoding and processing of smart city video cameras to be analysed by tenants using AI.

Keywords	Edge computing, Multi-tenancy, AI

Copyright

Deliverable history

Version	Date	Reason	Revised by
0.1	03.2023	Table of Contents	Sean Ahearne
0.2	10.2023	Partners contributions	Roberto Bifulco
1.0	11.2023	Finalization	Roberto Bifulco

List of abbreviations and Acronyms

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
EMDC	Edge Mobile Data Center
EU	European Union
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
PoC	Proof of Concept
TBC	To Be Confirmed
TBD	To Be Defined

Table of Contents

1. Executive summary 9

2. Use case overview 11

2.1. Background 11

2.2. Motivation 11

2.3. Objective 12

2.4. Goals (KPI’s) 13

3. Implementation and Integration 14

3.1. Use case implementation 16

3.1.1. VTU 16

3.1.2. Pedestrian Face Blurring 17

3.1.3. Video Object Detection 18

3.1.4. AI Audio Analysis 18

3.1.5. i.MTX 19

3.1.6. AI SOL Optimization 20

3.2. Integration with the BRAINE platform 21

4. Results 25

4.1. Video Transcoding Engine 25

4.2. Video Object Detection 26

4.3. Pedestrian face blurring 26

4.4. AI Audio Analysis 27

4.5. AI SOL Optimization 27

5. Impact 29

5.1. Comparison to existing systems 29

5.2. Advantages of the BRAINE platform 30

5.3. Business solutions and economic advantages of BRAINE for Telco operators and municipalities 30

5.3.1. BRAINE for Smart Cities 30

5.3.2. BRAINE for Telco 31

6. Conclusion 33

List of Figures

Figure 1.1 Example Diagram.....10

1. Executive summary

UC2 explores the deployment of AI applications in smart city and campus settings, focusing on using sensors for video and audio capture to monitor traffic, crowds, and emergencies, to mention just a few potential use cases. Traditional cloud-based processing methods face challenges such as high data transfer costs and latency issues. To overcome these, UC2 leverages the BRAINE Edge Micro Datacenter (EMDC), a compact, integrated system designed for local data processing. EMDC reduces the need for large-scale data transfer to remote datacenters, thereby minimizing associated costs and latency, and supports multi-tenancy to accommodate various services concurrently, such as 5G functions and AI analytics.

The project is driven by the need for efficient local processing in urban environments, where handling substantial video and audio data with deep learning algorithms demands significant computational resources. EMDC offers a solution by replacing traditional servers, thus addressing power, space, and carbon footprint concerns. This innovative approach also allows for the consolidation of ICT infrastructure within a city or campus, facilitating easier deployment and avoiding logistical challenges. In these settings, both real estate and power reources are critical.

UC2's objective is to leverage the EMDC for multi-tenant audio-video analytics services, covering the entire processing pipeline. This initiative includes developing components for video/audio analysis, data protection, pre-processing services, and data management frameworks, alongside hardware/software acceleration for enhanced computing capabilities. The project involves multiple actors, including edge, application, cloud, and service providers, as well as service consumers and subjects.

In summary, UC2 aims to demonstrate a scalable, efficient, and secure Edge as a Service platform for distributed video and audio analytics in smart city applications. This approach is not only more efficient in terms of infrastructure utilization but also ensures enhanced data protection and privacy. With its focus on reducing energy consumption and improving processing accuracy and performance, UC2 showcases the benefits the BRAINE EMDC can provide in smart city technology, facilitating better traffic analysis, surveillance, and emergency response services among others.

Figure 1.1 shows the high-level overview of the BRAINE components, on top of which UC2 is developed and deployed.

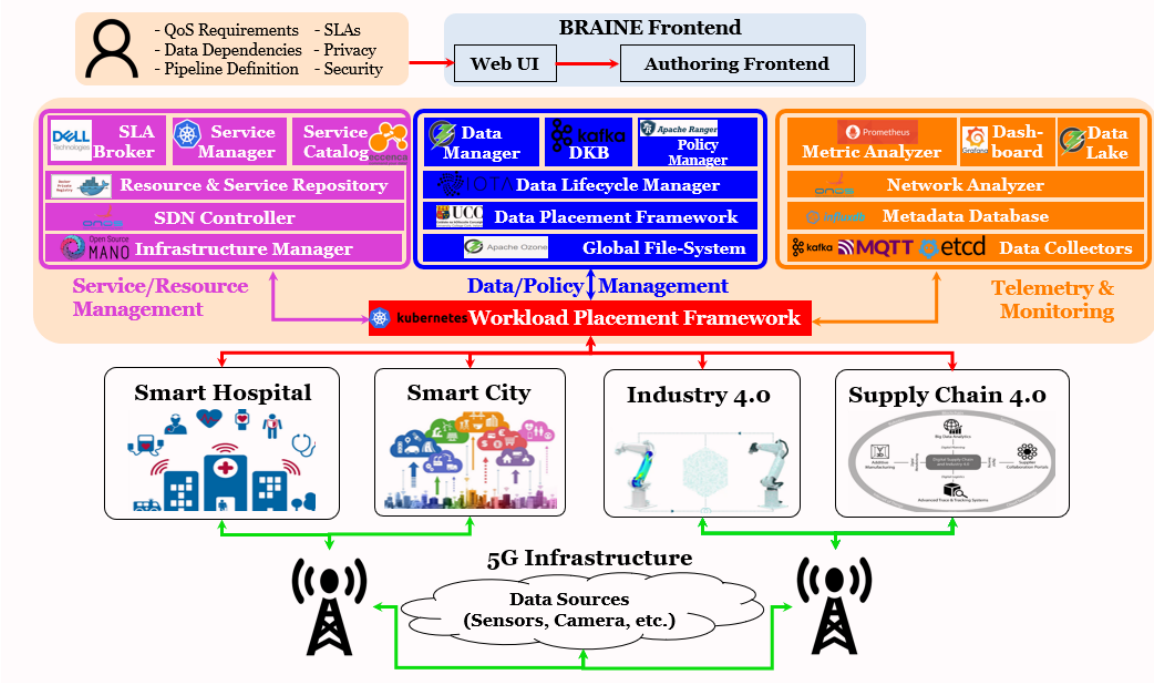


Figure 1.1 High-level overview of the BRAINE concept

2. Use case overview

UC2 deals with AI applications in the context of a smart city/smart campus scenario. These applications include monitoring using different types of sensors for video and audio capture, such as road traffic monitoring, crowd monitoring, emergency detection, etc. The rest of this section explains the background assumptions and motivations related to this use case.

2.1. Background

Smart cities and campuses are one of the main edge infrastructures in which AI applications are expected to be deployed in large scale. However, since many applications involve the processing of many video feeds, it becomes quickly demanding for the local ICT infrastructure to deal with the large amount of data that might require timely data processing. In fact, data processing requires multiple steps, such as transcoding and analysis. While these steps could be applied in a remote cloud datacenter, the cost of moving data remotely might be prohibitive, especially when considering the heterogeneous environment of a smart city, where the same camera feed might be processed by multiple services. That is, a single video feed might need to be delivered (replicated) to multiple services, and a service might need multiple video feeds to be executed. Furthermore, some applications, such as road traffic analysis, require timely response, which might need to be provided also in cases of interrupted remote connectivity. Therefore, reliance on a remote datacenter might be not possible for either reliability or communication latency constraints.

Conversely, local processing may address most of these challenges, by avoiding the need to send data to a remote datacenter, thereby reducing data transfer volumes and latency. For example, transcoding and analysis could happen localy and distribute remotely only the results of the analysis.

2.2. Motivation

Providing local processing capabiltiies in a city or campus setting is far from trivial. Processing large amounts of video/audio data, likely leveraging deep learning algorithms, is an expensive workload in terms of computation power. Relying on regular servers for this workload might not be possible, and could lead to issues in terms of power requirements, real estate space and carbon footprint.

In BRAINE, these issues motivate the development of a new system, the Edge Micro Datacenter (EMDC), which can replace several servers in a much smaller and integrated form factor, which powers advanced software that can leverage the high integration levels of the system.

This enables easier deployment of the EMDC (e.g., on top of building roofs), avoiding any complex logistic issue (e.g., power and cooling provisioning). At the same time, the EMDC can provide multi-tenancy to support a multitude of infrastructure and third-party services, thereby consolidating the overall ICT infrastructure of the city/campus. For example, the EMDC can support 5G virtual network functions, while running in parallel the transcoding and AI analytics services mentioned earlier.

The multi-tenancy gives at the same time opportunities for further optimizations. For example, services requiring to operate on the same video stream might directly share the pre-processing of such stream within the context of the same EMDC. Likewise, services can be composed to leverage their outputs to deliver more complex features.

Finally, the EMDC provides a vantage point to enforce data protection and privacy. For example, video streams can be processed close to the source to remove relevant personal identifiable information (e.g., faces from a video), before sending the data to a third-party service or to any remote location.

2.3. Objective

The goal of UC2 is to leverage the EMDC platform to provide a multi-tenant service platform for audio-video analytics, supporting all the steps of the processing pipeline. To achieve this objective, the UC2 develops test scenarios and validates them. This includes the development of the following components:

- Video/Audio analysis applications
- Data protection filters/services
- Video/audio pre-processing services
- Video/audio acquisition services
- Data management framework (for data distribution among services)
- Hardware/software acceleration for compute services (e.g., AI)

Currently, there is no shared, efficient and secure Edge as a service platform, where multiple data collectors and service providers could run their application. Overall, Use Case 2 addresses the case of smart city applications that perform distributed video and audio analytics. Instead of single service provides with own infrastructures, UC2

demonstrates a scalable, heterogeneous and multi-tenant service infrastructure for traffic analysis, surveillance, smart transportation, emergency response.

In this sense, the Use Case 2 assumes the presence of the following actors involved in the proposed scenarios:

- Edge provider: edge node owner, manages the edge node
- Application provider: provides the software that implements the use case, may be a “tenant” on the edge node
- Cloud provider: provides the remote infrastructure, if any
- Service provider: provides the end-to-end service to implement the use case, combining the services from application, edge and cloud providers
- Service consumer: buys the service from the service provider, and uses it
- Service subjects: stakeholders passively involved in the service, e.g., people appearing in the monitored videos

2.4. Goals (KPI’s)

We define the KPIs for this use case considering both application-specific and general performance metrics, as listed next:

- Efficiency: this criteria refers to the energy consumption of the edge node when running end-to-end processing pipelines. The objective is to improve the processing while reducing the energy consumption by 20% with respect to a state-of-the-art datacenter infrastructure, considering an equivalent computing power to run the service.
- Accuracy: algorithm performance in vehicles and pedestrians flow identification (Accuracy = true positive tests/ total tests). At least 85% of accuracy is our objective
- Performance: UC2 aims at demonstrating faster AI/ML Computing functions than state of the art Deep Learning execution engines (measured in terms of execution time). Moreover, we target for GPU-assisted video transcoding at least a 5x performance improvement compared to a CPU-only solution.

3. Implementation and Integration

Use case 2 implements a live stream analytics pipeline composed of a few high-level blocks, as depicted in Figure 3.1.

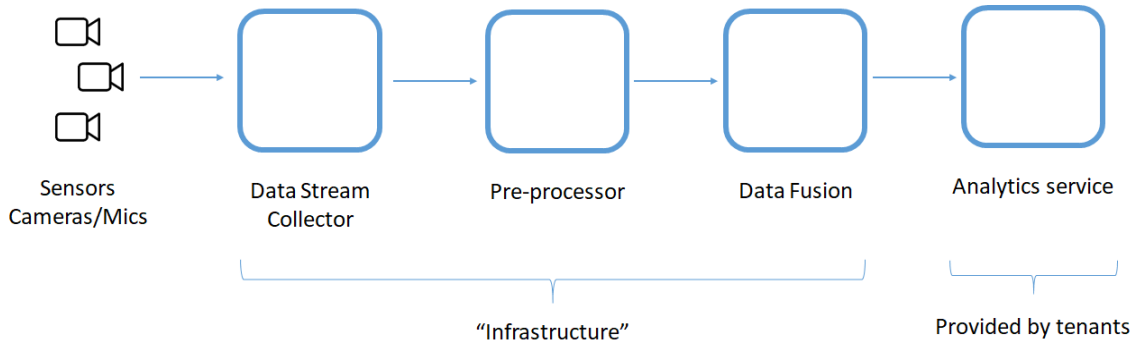


Figure 3.1: High-level computation pipeline for UC2

The pipeline is designed to work on live captured traffic, although it might eventually include storage components at different levels. For instance, one of the analytics services could provide saving of results in a database.

The different processing blocks have heterogeneous requirements in terms of hardware, with the Pre-processor and Analytics Service being the most demanding components. In particular, the pre-processor is generally in charge of video/audio transcoding and transformation. This is a generally expensive computation, which can benefit from widely parallel processor architectures, such as GPUs. Likewise, the Analytics components can employ Deep Learning algorithms, which in turn benefit from GPUs and other accelerators.

The overall pipeline works therefore as follows: data stream collectors are connected to the data stream sources, e.g., cameras, and work as an adaptation layer. This mostly I/O workload has to do with protocol conversion and with the handling of connectivity issues. The standard streams generated by the stream collector are then provided to the pre-processor. As mentioned, this component performs transcoding, but also adaptation of video rate and quality. In fact, the pre-processor offers an external interface that can be leveraged by the platform orchestrator to adapt the video quality according to the contextual resources' constraints. The Data Fusion element that follows is a sort of "data-hub and policy enforcement point". It provides routing of the raw data streams to the analytics service. This requires potential duplication of the streams, but also chaining of the stream through multiple functions, before finally delivering it to the specific analytics component. For instance, we show a case in which we perform face blurring on a video stream before providing the video to an analytics service, to provide privacy protection.

The last component, the analytics services, finally receive the modified stream and implement the custom analysis required by the specific application use case.

The implementation of Use Case 2 must consider the requirements of building a solution that supports the multi-tenancy concept, when running AI functions designed for video analytics. This allows multiple, independent AI video analytics services to be simultaneously operated by different tenants on the same data set (e.g., the same video stream).

The functional structure of the fully integrated solution is depicted in Figure 3.2, while Figure 3.3 shows the basic concept of the Application Platform, implemented to fulfill the multi-tenancy requirements of UC2.

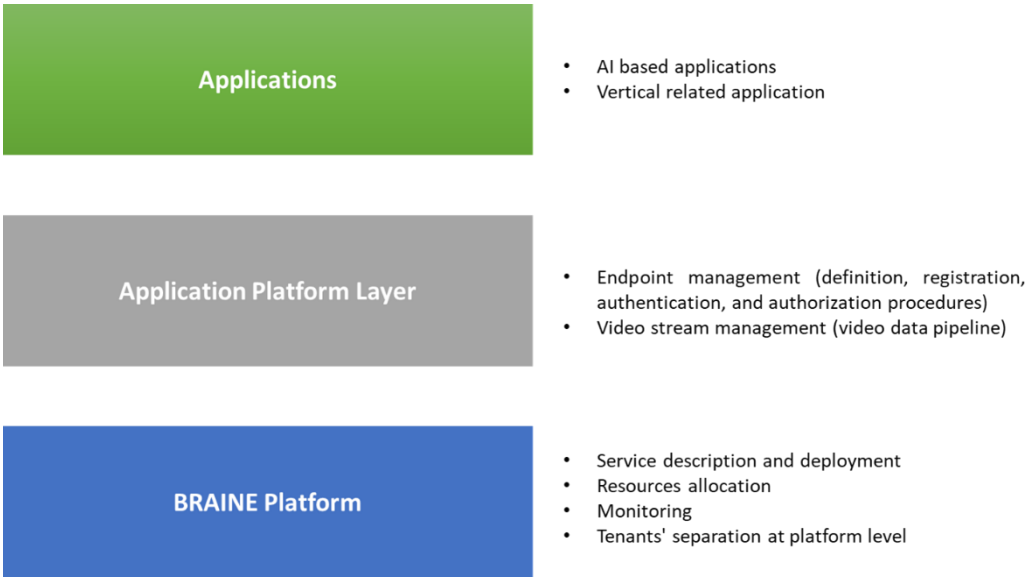


Figure 3.2: Functional structure of the integrated solution

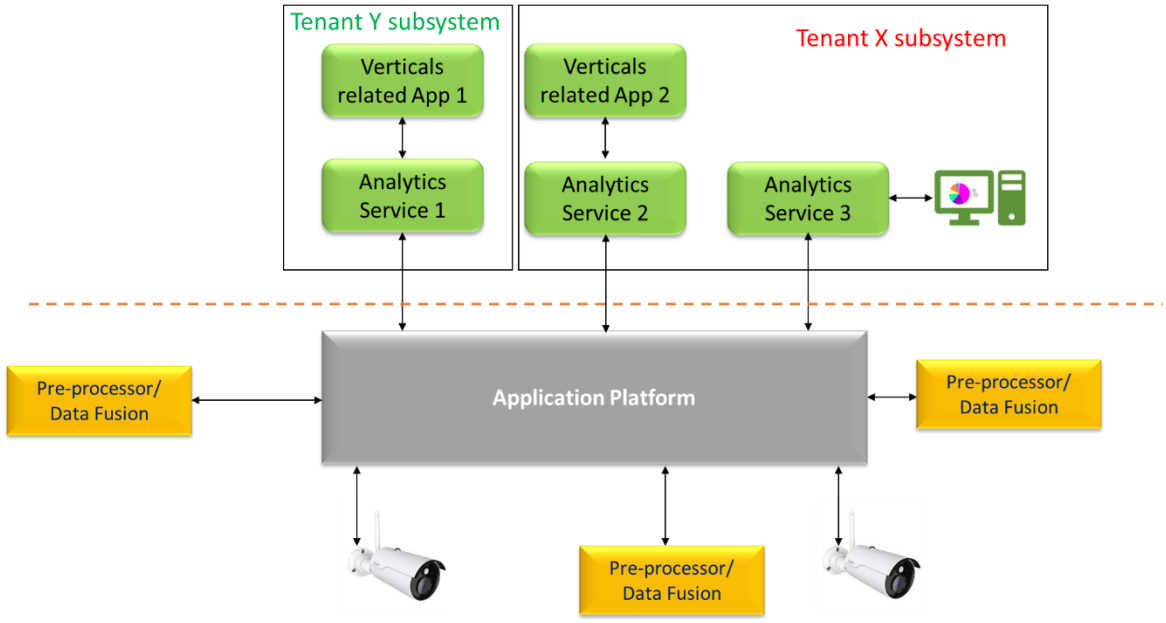


Figure 3.3: Application Platform usage scenario

3.1. Use case implementation

Use case 2 includes several components:

- Transcoding – VTU
- Face Blurring – Pedestrian Face Blurring
- Analytics Services:
  - Video Object Detection
  - AI Audio Analysis
- Application Platform - i.MTX.
- AI SOL optimization.

A brief description of the components will be provided in the following sections.

3.1.1. VTU

The VTU acts as data stream collector and pre-processor in the data processing pipeline of Use Case 2. It can convert audio and video streams from one format to another. The source stream can originate from a file within the local storage system, or maybe a packetized network stream. The requested transcoding service can be monodirectional, as in video streaming, or bi-directional, like in videoconferencing. The transcoding capabilities of the VTU are provided by Libav (https://github.com/libav/libav). Libav is an

open-source library, which can handle a wide variety of audio and video coding standards. For the most computationally intensive video encoding tasks, the VTU relies on Graphical Processing Unit (GPU) resources.

The main features of VTU are the following:

- Real-time video streaming management from different sources
- Video processing: transcoding, transrating, transizing
- Capability to share video among sources/users/applications
- Protocol support: rtmp, rtsp, http, hls, rtp, websocket.

3.1.2. Pedestrian Face Blurring

Face blurring represents the first element of the video processing chain that has been deployed in the BRAINE EMDC and dedicated to UC2. It is an important technique used to protect individuals' privacy in images or videos. For BRAINE purposes, TensorFlow, an open-source deep learning framework, has been identified as the AI platform for implementing face blurring algorithms thanks to its ease of use and an extensive library of tools.

The TensorFlow algorithm for face blurring involves the following steps:

- Face Detection: The first step is to accurately detect faces in the input images or video frames. A Faster R-CNN model is utilized to locate faces within the given input.
- Face Region Extraction: Once the faces are detected, the algorithm extracts the region of interest (ROI) containing the face. This step ensures that only the identified faces undergo the blurring process, while the rest of the image remains unchanged.
- Blurring Technique: this step involves the use of another framework: OpenCV. The blurring method in OpenCV is used to blur an image using the normalized box filter. The function smooths an image using the kernel which is represented as:

$$K = \frac{1}{ksize_{width} * ksize_{height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ \dots & & & & & \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

- Generation of RTSP blurred stream: after the algorithm is developed and implemented, the blurred stream feeds the Video Object detection algorithm to protect the user privacy.

3.1.3. Video Object Detection

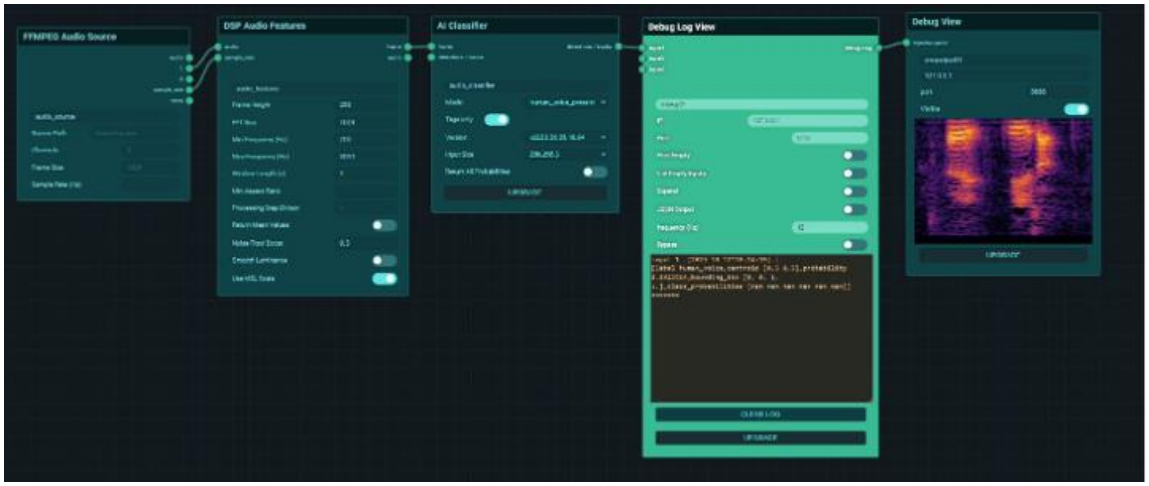
Video object detection is a part of the pipeline processing chain implemented in BRAINE for UC2. To meet the accuracy and reliability requirements defined for the smart city use case, DeepSORT algorithm has been selected as the best candidate. Such algorithm is a deep learning-based object detection network with a Kalman filter-based tracking framework. The algorithm starts by detecting objects in a video stream using a deep learning-based object detection network (i.e., YOLOv4). The object detection network produces a set of bounding boxes and corresponding object detection scores for each object in each frame of the video stream. The bounding boxes and detection scores are then passed to the tracking module, which uses a Kalman filter to estimate the position, velocity, and acceleration of each object over time. The Kalman filter uses the detection scores to update the state of the filter and predict the future state of the object in subsequent frames.

3.1.4. AI Audio Analysis

AI Audio Analysis is based on the Agile Focused Signal Analytics platform by MarshallAI. The software provides the user the possibility to define and configure an AI Application for processing an audio stream or recording and automatically take action based on inferred results. The AI Application uses a pre-trained AI Model to analyse the received audio stream. All stages of the analaysis pipeline, including AI Model training and data collection / annotation can be performed on the same platform running on the EMDC - enabling full control of the analysis process for each specific use case.

A common audio analysis AI-Program running on the AFSA consists of the following stages:

- Audio source (e.g. audio over RTSP)
- Audio feature extraction (e.g. FFT, Frequency binning, averaging)
- ML Classifier model (e.g. Convolutional Neural Networks)
- Logical decision tree (e.g. Thresholding, Mapping, If/Else)
- Output action (e.g. HTTP POST / REST, TCP, MQQT, SMS)



The program flow can be extended as needed to gain higher accuracy results in complex cases by chaining multiple AI Models to perform multiple stages of inference and filtering.

Audio feature creation on the AFSA platform can be performed using a variety of different options. The baseline features consist of a defined number of frequency bins depicted as vertical columns in a feature image where the intensity value of each bin represents a black to white color value given as an unsigned 8bit integer (0-255). As audio is read forward the vertical bars representing frequency intensities present in the incoming audio stream are stacked horizontally forming a sliding feature from left to right giving a sense of time being represented on the horizontal axis. The single values for each frequency bin are then mapped to an RGB color spectrum to better highlight minor details and changes in audio to the application developer when looking at the generated feature images through a computer monitor during annotation / labeling. For specific use cases it might be required to have a higher dynamic range present to distinquish smaller differencies between samples. For these scenarios the AFSA platform has built in digital filters to perform bandpass filtering, resampling and quantization of the incoming raw audio signal.

Features generated by the feature creation / extraction stage can then be used to train an artificial neural network to perform inference on multiple audio streams simultaneously.

Computation of the neural network's backward and forward passes utilize CUDA and Tensor -core hardware acceleration by default.

3.1.5. i.MTX

i.MTX enables the slicing and multi-tenant concepts at the application and data level among different tenants. i.MTX is developed as a Kubernetes Pod, in particular, one instance can manage one data processing pipeline.

A data processing pipeline that needs to be managed by a single i-MTX instance can be described and configured using the i.MTX builder, a web-based dashboard. The output of the builder is a configuration file to be used to configure the parameters of the application platform and a “yaml” file to be used for deploying the application Pod using Kubernetes.

A possible implementation of data pipelines using the i.MTX is depicted in Figure 3.4. Data pipeline 1 is sent to two different tenants' subsystems assuring privacy and domain separation.

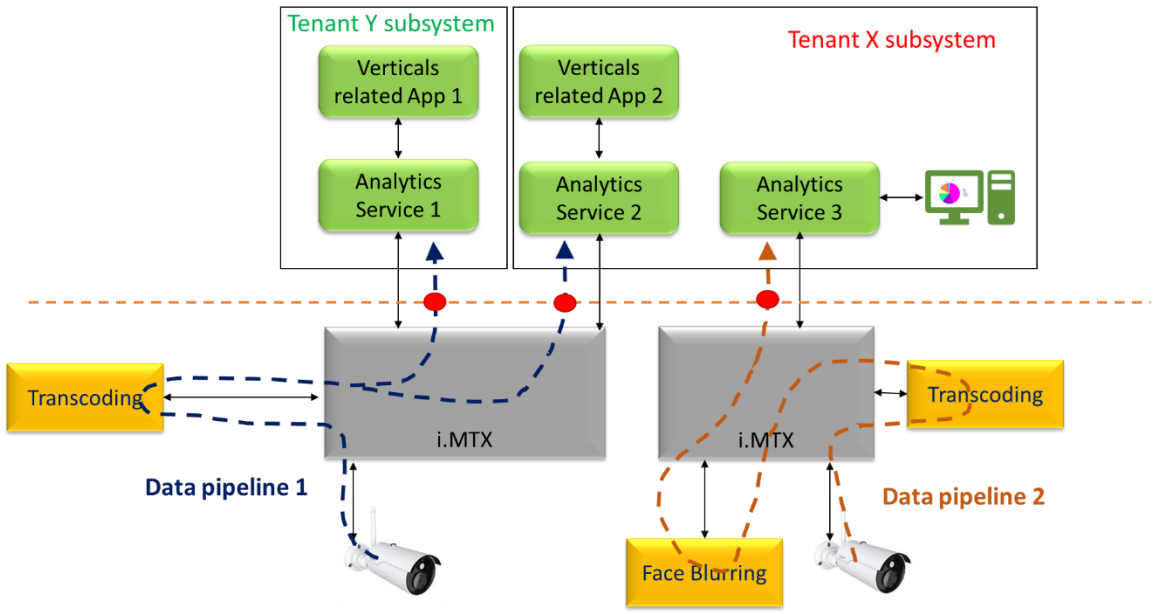


Figure 3.4: Data pipeline implementation using i.MTX

3.1.6. AI SOL Optimization

SOL is a deep learning compiler and runtime system, as such, it becomes part of the application stack, although being transparent to the developers. Developers use regular open source frameworks, such as Pytorch, to develop their applications. SOL is integrated in such applications as an additional software library, which then optimizes the software execution during the first cold-start of the application on the deployed system. Potentially, this step could also be preemtively run to ensure fast boot at any time.

Under the hood, SOL reads the input program and infers the sequence of operations applied to the program’s input. For instance, SOL reconstructs the sequence of mathematical operations that finally implement a neural network algorithm. Since these operations are described at the logical level by domain experts, there is often a large space to reconsider them in their entirety to introduce optimizations. For example, it is

possible to remove operations that finally are redundant or repeated, or some operators can be “merged” to reduce memory copy (something generally referred to as “operator fusion”). SOL applies these and other optimizations on top of an itermediate representations that allows easy manipulation of the operators. Then, the resulting representation is compiled into a new version of the source code that gets automatically integrated in the orginal program and transparently called within in. The SOL runtime ensures that memory copy and handling is also transparently integrated.

The final result is that deep learning programs are optimized and run in a fraction of the original time, with speed-ups in the range 30% to over 3 times faster.

3.2. Integration with the BRAINE platform

All the SW components developer for implementing the UC2 demo are available as Docker container images and can be deployed over the EMDC Kubernetes cluster using the BRAINE platform. Figure 3.5 shows the data pipeline used for the demo.

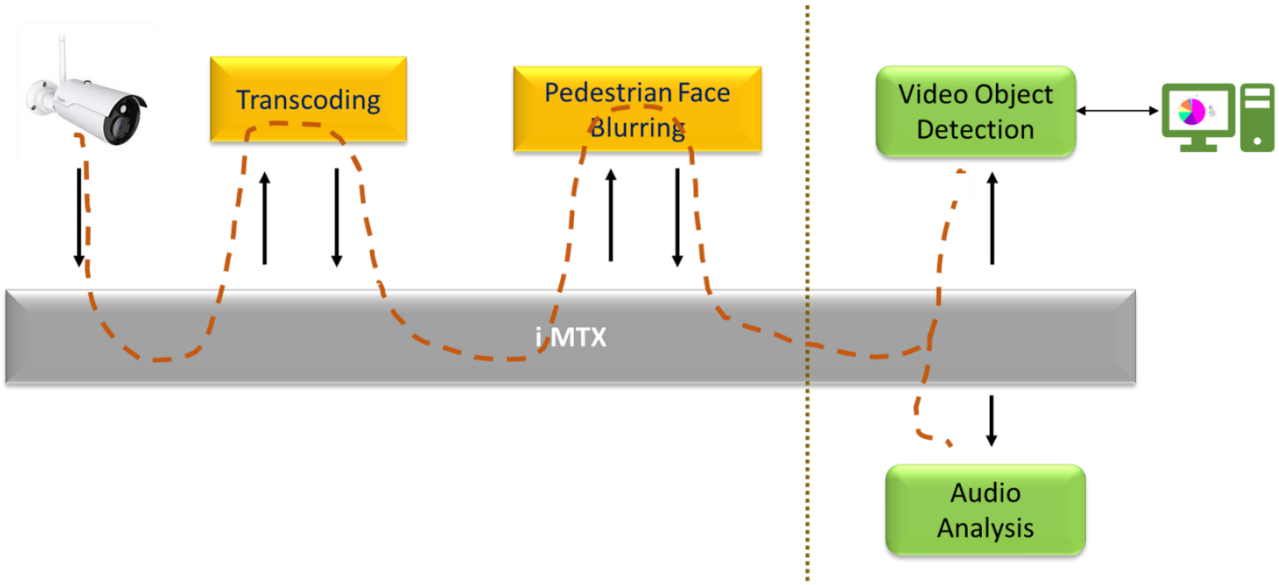


Figure 3.5: Data pipeline for use case 2 demo

The i.MTX builder filled in to implement the data pipeline is reported in Figure 3.6. Figure 3.7 shows the output configuration file and, finally Figure 3.8 shows the “yaml” file related to the i.MTX.

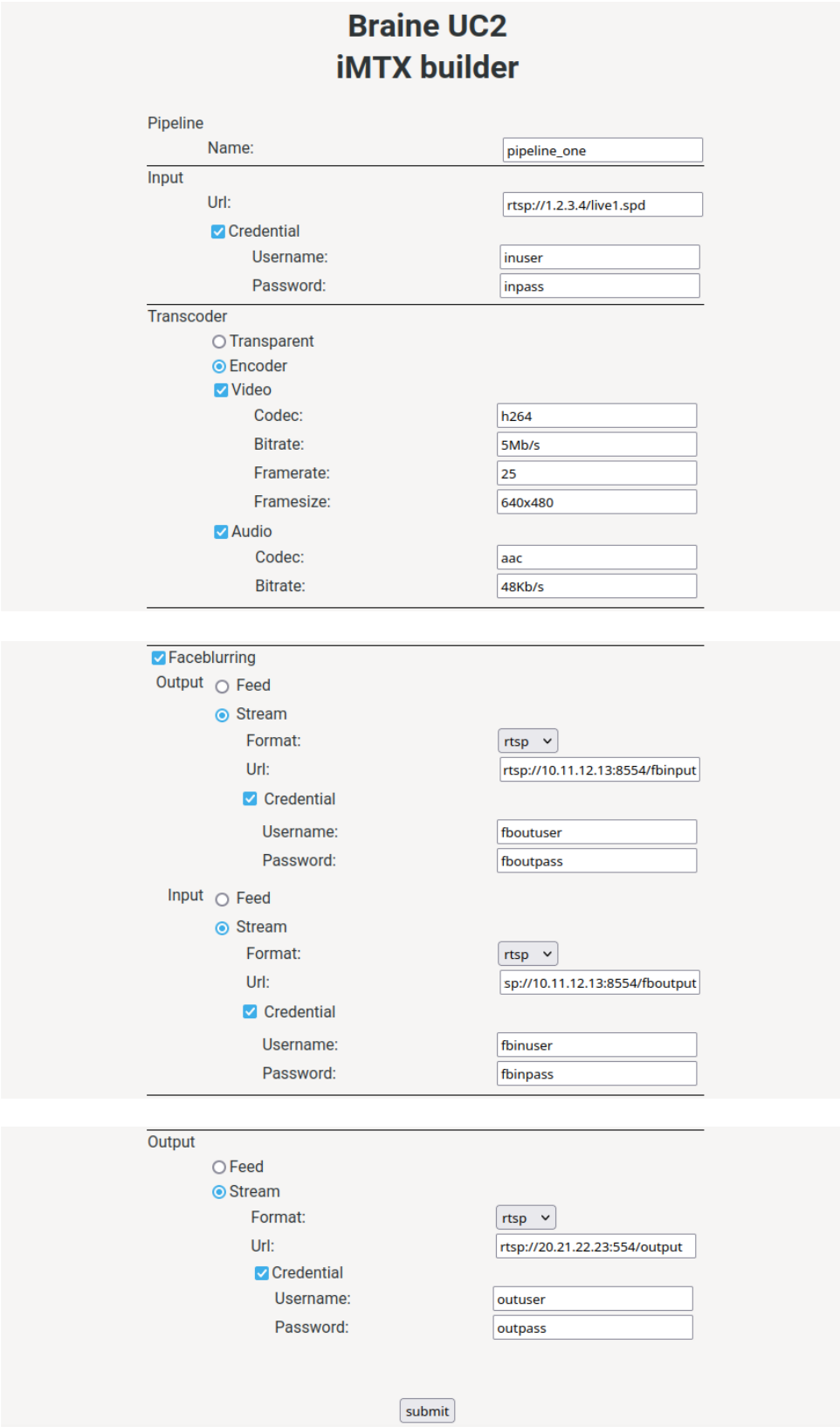


Figure 3.6: Data pipeline configuration using the builder

Figure 3.7: Data pipeline configuration file

iMTX pod yaml file

```
apiVersion: v1
kind: Service
metadata:
  name: imtx-pipeline_one
  namespace: uc2-itl
  labels:
    app: imtx-pipeline_one
spec:
  type: NodePort
  ports:
  - port: 8554
    protocol: TCP
    name: rtsp
  - port: 1935
    protocol: TCP
    name: rtmp
  - port: 9997
    protocol: TCP
    name: api
  - port: 22
    protocol: TCP
    name: ssh
  selector:
    app: imtx-pipeline_one
---
```

```
---
apiVersion: v1
kind: Pod
metadata:
  name: imtx-pipeline_one
  namespace: uc2-itl
  labels:
    app: imtx-pipeline_one
spec:
  containers:
    - name: imtx-pipeline_one
      image: 172.30.101.1:5000/uc2-itl/imtx
      imagePullPolicy: Always
      volumeMounts:
      - name: config-volume
        mountPath: /app/config
      ports:
      - containerPort: 8554
      - containerPort: 1935
      - containerPort: 9997
      - containerPort: 22
  volumes:
    - name: config-volume
      configMap:
        name: imtx-pipeline_one
  nodeSelector:
    kubernetes.io/hostname: uc2-work01
    kubernetes.io/hostname: uc2-work02
  restartPolicy: Always
```

Figure 3.8: i.MTX yaml file

4. Results

Show results for relevant KPI's here

4.1. Video Transcoding Engine

A performance evaluation was carried out on the testbed hosted at CNIT premises using the BRAINE platform. Figure 4.1 shows the test scenario: a recorded video, locally stored, is transcoded using first the x86 CPU and then a NVIDIA GPU hosted on the PCIe bus. In particular, the output parameter used for the transcoding task are:

- Codec: H.264:
  - SW, using CPU - codec libx264
  - HW, using GPU – codec h264_nvenc
- Bitrate: 5Mb/s
- Video Size: full HD
- Frame size: 24

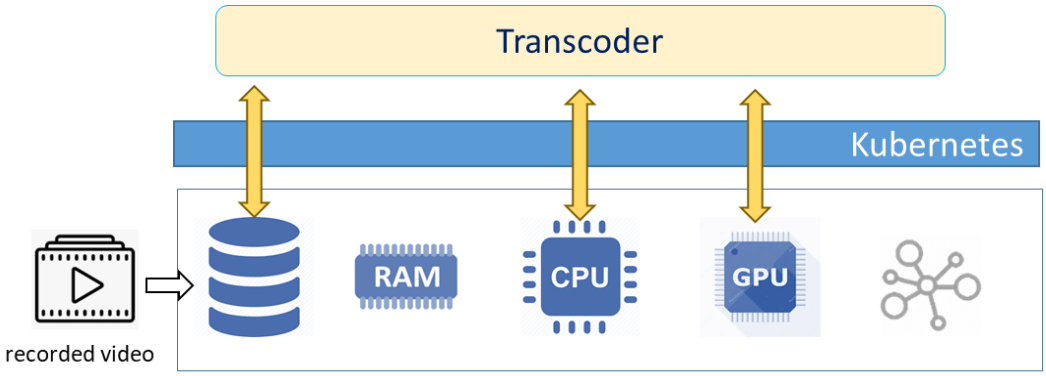


Figure 4.1: Performance evaluation scenario.

```
Output #0, mp4, to 'out.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.29.100
    Stream #0:0(und): Video: h264 (libx264) (avc1 / 0x31637661), yuv420p, 1920x1080 [SAR 1:1 DAR 16:9], q=-1--1, 5000 kb/s, 24 fps, 12288 tbn, 24 tbc (default)
    Metadata:
      handler_name    : VideoHandler
      encoder         : Lavc58.54.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/5000000 buffer size: 0 vbv_delay: -1
frame=  720 fps= 54 q=-1.0 Lsize=   19963kB time=00:00:29.87 bitrate=5473.9kbits/s speed=2.24x
video:19954kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.044424%
```

```
Output #0, mp4, to 'out.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf58.29.100
    Stream #0:0(und): Video: h264 (h264_nvenc) (Main) (avc1 / 0x31637661), yuv420p, 1920x1080 [SAR 1:1 DAR 16:9], q=-1--1, 5000 kb/s, 24 fps, 12288 tbn, 24 tbc (default)
    Metadata:
      handler_name    : VideoHandler
      encoder         : Lavc58.54.100 h264_nvenc
    Side data:
      cpb: bitrate max/min/avg: 0/0/5000000 buffer size: 10000000 vbv_delay: -1
frame=  720 fps=280 q=27.0 Lsize=   18822kB time=00:00:29.95 bitrate=5146.8kbits/s speed=11.6x
video:18818kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.020737%
```

Figure 4.2: Results on dashboard

HW used	Frame per second (max)	Speed
CPU	54	2.24x
GPU	280	11.6x

Table 4.6.1: CPU vs. GPU transcoding performance

As reported in Figure 4.2 and summarized in Table 4.6.1, using a GPU allows for a 5.2x increase in transcoding performance, in alignment with our target KPI.

4.2. Video Object Detection

In the following, measured KPIs are listed with corresponding results for Video Object Detection algorithm:

- Accuracy: this is a crucial KPI for object tracking in smart cities as it measures how well the algorithm can detect and track objects. The BRAINE Video Object Detection algorithm achieved an accuracy of 95.6% for pedestrian tracking in a smart city environment.
- Processing Time: processing time is an important KPI for real-time applications related to smart cities scenarios. The BRAINE Video Object Detection algorithm achieved a processing time of 10 frames per second (fps) on a standard CPU, and 25 FPS with GPU support (NVIDIA Tesla T4), which is suitable for real-time applications.
- Robustness: robustness is an important KPI as it measures how well the algorithm can handle challenging scenarios such as occlusions and cluttered scenes. BRAINE Video Object Detection algorithm achieved a robustness score of 94.2% for pedestrian tracking in lab environment.
- False Positive Rate: the false positive rate measures the number of false detections made by the algorithm. In BRAINE, a false positive rate of 3.5% for pedestrian tracking in UC2 has been achieved.

4.3. Pedestrian face blurring

In the following, measured KPIs are listed with corresponding results for pedestrian face blurring algorithm:

- Face Detection Accuracy: Face detection accuracy measures how well the AI algorithm can accurately detect faces in the input images or video frames. A high face detection accuracy is essential for ensuring that the blurring process covers all identifiable faces. The measured face detection accuracy for UC2 is around 95%, indicating that the algorithm can reliably identify and locate faces.
- Blurring Efficiency: Blurring efficiency measures the speed and computational efficiency of the face blurring algorithm. It is crucial for real-time or near-real-time processing of video streams in smart cities. The blurring algorithm is able to process frames at more than 25 frames per second (fps) without considering the GPU.
- False Positive Rate: False positive rate measures the occurrence of false blurring, where non-face regions are erroneously blurred. A low false positive rate indicates that the algorithm accurately identifies and blurs only the faces without affecting other parts of the image. The measured false positive rate should is below 5%, ensuring minimal unnecessary blurring and preserving the contextual information of the scene.

4.4. AI Audio Analysis

The MarshallAI AFSA running on the BRAINE platform enables the usage of multi-stage chained deep learning algorithms. This enables to pre-filter interesting events in real time with low compute requirements and makes it possible to forward data to higher compute availability once a certain threshold for interesting data is exceeded. Using this multi-stage detection, tracking and classification setup the AFSA can achieve accuracies reaching 99% in lab environments within hours of semi-unsupervised fine-tuning methods.

The modular architecture of the EMDC allows for splitting up processing chains on the AFSA to have payloads forwarded over the network for further processing.

Robustness for all solutions is achieved through environment specific fine-tuning and can be done using the "autotune" mechanism bundled into all AFSA -based solutions. The "autotune" algorithm incorporates mechanisms to internally determine a networks accuracy in live -scenarios and automatic sample collection for quickly fine tuning deep neural networks.

4.5. AI SOL Optimization

Deep learning computations are arguably one of the main sources of overhead in the audio/video analysis pipeline. One of the problems of these computations is that the

definition of a deep learning network archtiectures happens at the conceptual level dealt with by a data scientist. This enables quick iteration and fast development of new funcitons, but introduces several sources of efficiencies. In fact, the layers of a neural network are executed as independent atomic elements of computation, introducing synthetic synchronization boundaries in the computation graph, which lead to unecessary data movements within processors and across processors and memory. To address the problem in BRAINE we extended a compiler system, SOL, which is capable of understanding the intent of a deep learning computation, and synthetize a new program that specializes for the execution of such neural network, thereby removing most of the sources of overhead. To demonstrate the capability of such system, we compare to the performance of state-of-the-art frameworks (PyTorch) when running a state-of-the-art and highly optimized neural network architecture (ResNet), in a typical inference setting (with small batch size=1).

Neural Network execution method	Time (ms)
PyTorch 2.0 Eager	79.11827
PyTorch 2.0 torch.jit.trace	105.45527
PyTorch 2.0 torch.compile	56.92193
SOL sol.optimize	31.41416

The results show that SOL optimization is capable of providing a 2.5x faster execution over PyTorch out of the box, when running on an Intel CPU. This is a shift in performance from 12 frames per second to over 30 frames per second on CPU!

Compared to the latest compile feature of PyTorch introduced experimentally in the latest PyTorch version, SOL is still faster by 1.8x. This overachieves our objective of improving by at least 20% the performance of complex data analysis applications.

5. Impact

Discuss the use case results in comparison to the state of the art, and also discuss the potential economic impact of the use case and overall BRAINE platform here.

5.1. Comparison to existing systems

BRAINE architecture has been designed to be scalable and dinamic to support several applications in the Smart Cities environment. One notable improvement compared with state-of-the-art system is the integration of multiple AI video processing algorithms to process the data in real-time manner and close to the end-user. One of this algorithm ensures that individuals captured in video footage remain anonymous and their privacy is protected. By automatically detecting and blurring faces in real time, we adhere to ethical considerations and address concerns surrounding the identification and misuse of personal data. This improvement promotes a safer and more privacy-conscious environment within smart cities. In addition to face blurring, the system now incorporates advanced people detection capabilities. Using video processing algorithms, we can accurately identify and track individuals within the captured footage. This feature enables various applications, such as crowd management, traffic monitoring, and public safety. With real-time people detection, authorities can promptly respond to potential security threats, mitigate congestion, and optimize resource allocation in a dynamic and efficient manner.

Regarding the edge computing and virtualization technologies, we have decentralized video processing tasks, bringing several advantages. Firstly, video analysis is performed at the edge of the network, closer to the source of the data. This localization minimizes latency and reduces the load on the central server, resulting in faster and more efficient processing. Moreover, the system benefits from the scalability and flexibility offered by virtualization platform. It dynamically allocates computing resources based on demand, ensuring optimal performance during peak periods and minimizing resource wastage during off-peak times. Furthermore, the integration of edge computing with the system enhances its compatibility with existing smart city infrastructure. By leveraging the distributed nature of edge computing, the system seamlessly integrates with surveillance cameras and other network components. This integration minimizes the need for significant modifications to the infrastructure, making it easier to deploy and scale the system across multiple locations within the smart city.

5.2. Advantages of the BRAINE platform

The BRAINE platform provides an easily scalable hardware architecture to be used on the edge. The possibility to quickly configure the platform to be suited for the desired compute needs (GPU vs CPU vs FPGA) is a big advantage in comparison to alternative platforms. The multi-tenancy support for deploying multiple different applications by various providers in a secure way onto a uniform hardware stack ensures superior scalability and maximises the utilisation of the compute resources.

The passive cooling and robust design of the hardware supports both indoor and outdoor deployments. The low latency data transfer from sensors to the nearby EMDC for data processing enables low latency inference at scale on the edge. Utilisation of standardised containerisation (Docker & Kubernetes) ensures low modification requirements for adding existing software on to the BRAINE-platform.

5.3. Business solutions and economic advantages of BRAINE for Telco operators and municipalities

5.3.1. BRAINE for Smart Cities

Smart Cities are complex technological ecosystems composed by several diverse technologies and interconnected devices. Manage the huge amount of data generated by the smart city is a technological challenge where edge computing is part of an enabling framework for efficient data processing and Artificial Intelligence contributes by providing the ability to extract meaningful information from the smart city data.

BRAINE is based on the heterogeneous Edge Micro Data Center (EMDC) optimized in terms of computing capacity and energy efficiency, and on a software system that integrates Artificial Intelligence technologies, able to process Big Data at the edge of the network, guaranteeing its security, privacy and sovereignty. Furthermore, BRAINE is based on technologies that optimize the use of available resources, providing new methodologies for allocation of workloads at the edges of the network that take into account different parameters including optimization of data management and processing to ensure, where necessary, low latency applications and real-time management in the case of mission-critical applications.

Use case 2 highlights how BRAINE can guarantee the realization of services in the large-scale Smart City area, even in case of low-latency applications and with stringent requirements in terms of bandwidth, which also need to manage large quantities of input

data in real-time. The solution creates services using distributed audio and video analysis techniques, based on a scalable, heterogeneous and multi-tenant infrastructure. The application scenarios cover traffic analysis, active surveillance, intelligent transport and emergency response.

Use case 2 demonstrates how the use of cameras, distributed throughout the city, can allow the creation of different types of services of interest to Smart City. In fact, cameras become one of the most versatile sensors if supported by artificial intelligence techniques as they allow the extrapolation of different types of information from the analysis of audio and video flows. In this context, the use of BRAINE Platform enables the ability to process large volumes of audio and video flows to obtain a large amount of heterogeneous information that can be used for the implementation of different services such as: Traffic management, logistics planning, urban space planning, assessment of pollution levels, active management of security and emergency response issues, crowd management and maintenance of city infrastructure.

5.3.2. BRAINE for Telco

The implementation of Use Case 2 (UC2) with the Edge Micro Datacenter (EMDC) platform offers several significant advantages for the telecommunications sector, particularly in connection with 5G deployments and other applications:

Enhanced 5G Network Capabilities: The integration of EMDC in smart city and campus environments is highly synergistic with the deployment of 5G networks. 5G technology, known for its high speed and low latency, benefits immensely from local data processing. EMDC can facilitate faster processing and analysis of data generated by 5G-enabled devices and applications, leading to more efficient network performance and enhanced user experiences.

Reduced Latency: For telecommunications companies, latency is a critical factor, especially for applications requiring real-time processing such as augmented reality (AR), virtual reality (VR), and autonomous vehicle navigation. By processing data locally with EMDC, the latency caused by data transfer to and from remote cloud servers is significantly reduced, making these real-time applications more viable and effective.

Cost-Effective Data Management: With EMDC's local processing capabilities, telecom operators can manage data more cost-effectively. By reducing the need to transfer large volumes of data to remote servers, companies can save on data transmission costs. This

is particularly relevant in the context of 5G, where the volume of data generated by end-users is expected to be substantially higher than previous generations.

Improved Data Security and Privacy: Data security and privacy are paramount concerns in the telecommunications sector. EMDC's ability to process data locally means sensitive information, like personal user data, does not have to be transmitted over long distances, reducing the risk of data breaches. This is crucial for maintaining user trust and complying with data protection regulations.

Energy Efficiency and Sustainability: EMDC's focus on reducing energy consumption aligns with the growing need for sustainability in the telecom sector. By optimizing processing power and reducing the carbon footprint associated with large data centers, telecom companies can achieve more sustainable operations, which is increasingly becoming a competitive advantage.

Scalability and Flexibility for Diverse Applications: The multi-tenant nature of EMDC allows telecom operators to support a wide range of services and applications on the same infrastructure. This flexibility is essential for 5G networks, which are expected to support diverse applications, from IoT (Internet of Things) devices to high-definition video streaming.

Edge Computing Integration: The rise of edge computing, where data processing happens closer to the data source, is a natural fit for 5G networks. EMDC's local processing capabilities are in line with the edge computing paradigm, enabling telecom companies to leverage this technology more effectively.

Innovative Service Offerings: With EMDC, telecom operators can develop and offer new, innovative services that require high-speed data processing and low latency, such as smart city services, traffic management solutions, and emergency response systems. This can lead to new revenue streams and enhanced service portfolios.

In summary, the integration of UC2 with EMDC in the telecommunications sector, especially in the context of 5G deployments, offers substantial benefits including enhanced network performance, reduced latency, cost-effective data management, improved security and privacy, energy efficiency, scalability, and the potential for innovative service offerings. These advantages position telecom companies to better meet the demands of modern digital services and infrastructure needs.

6. Conclusion

We presented the implementation of smart city/campus use case using tthe BRAINE platform and its technological components. BRAINE enables the deployment of micro datacenters that support complex multi-modal analysis pipelines, including several sources of data (microphones and cameras), multiple analysis elements and algorithms, and infrastructural components to support connectivity and secure data handling. The showcased application features a multi-tenant setting with privacy preserving filters applied before the video analysis pipeline, and the results of the video analysis combined with a paralell audio analysis pipeline. The combination of these elements is made possible by the joint work of the BRAINE platform that provides resources to the application components and the use case implementation platoform that orchestrates the data flow among them. Together with both hardware and software acceleration enhancements, the overall concept described here provides a promising starting point to impact the Smart City and Telecom operators markets, where edge deployments and the supported applications already play an important role for security and 5G connectivity use cases.