Experimentally Assessing Deployment Tradeoffs for AI-enabled Video Analytics Services in the 5G Compute Continuum

Pavlos Basaras[1], Emmanuel Vasilopoulos[1], Stratos Magklaris[1], Konstantinos V. Katsaros[1], and Angelos J. Amditis[1]

[1]Institute of Communication and Computer Systems (ICCS)
Email: {pavlos.basaras, emmanuel.vasilopoulos, stratos.magklaris, k.katsaros, a.amditis}@iccs.gr

***Abstract*—This article investigates the performance of video analytics services in a real industrial scenario, namely, a Port, using commercial grade cellular networks (5G, LTE-A, LTE) and a private cloud infrastructure. We create a virtual platform incorporating cloud and extreme/far-edge devices to host the workload of AI services (e.g., object detection), and experimentally investigate deployment trade-offs in the 5G compute continuum, based on the criteria of service latency and bandwidth usage, inference accuracy, inference time and power consumption. Our experimental results demonstrate that private cloud computing benefits low-latency, high-performance apps, whereas far-edge processing (local offloading) can be used for bandwidth/power-efficiency.**

***Index Terms*—Cellular networks, video analytics, compute continuum**

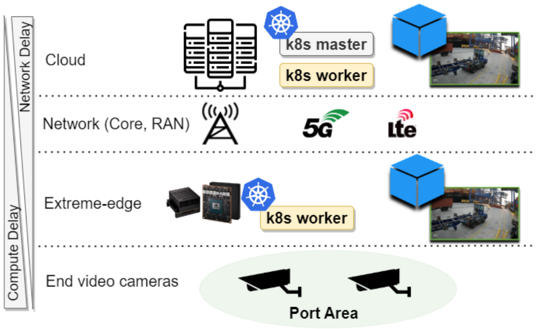


Fig. 1: Private cloud and extreme-edge system at PCT. EI placement close to the data sources alleviates the network delay at the expense of computing resources, and, vice versa.

I. Introduction

The development of advanced communication technologies such as 5G, laid the foundation for facilitating the seamless transfer of vast amounts of video data in (near) real-time. Such sensors can be embedded in machinery, equipment, infrastructure, drones, or even wearable devices, capturing rich visual information regarding the operational environment at hand. Of particular importance is the use of artificial intelligence (AI) and machine learning (ML) for processing video data, i.e., *video inferencing*, for various goals including object detection and tracking, e-health and public safety, traffic monitoring, smart manufacturing, intelligent industrial robotics [1]–[4].

Yet, video analytics services induce a heavy computational load. Cloud data centers can provide the necessary computing resources, but, at the cost of transmission delays and significant network bandwidth for sending data to remote locations [5]. On the other hand, edge (or extreme/far-edge) computing services, placed closer to the data sources, can deliver fast and accurate responses to live video queries, but, with limited computation power [6]. To facilitate this trade-off, the compute continuum paradigm introduces various placement options of edge intelligence (EI) within the 5G network infrastructure [7], [8].

Particularly, in 5G (and beyond) networks, the cloud and edge, and their interplay (Figure 1), will play a crucial role in the provisioning of efficient AI-assisted video analytics services, at scale [4], [9]. A multitude of new and exciting use cases for the industry domain are possible within this joint ecosystem focusing on security, safety, operational efficiency, quality control, predictive maintenance, energy savings, real-time decision making, and data-driven insights [3], [10]. However the efficient utilization of such distributed computing resources across various tiers, from (extreme/far-)edge devices to centralized cloud servers, coupled with the network, service, and AI specific requirements and limitations, is no trivial task. For instance, sending high quality video frames (e.g., 4K) can enhance the inference accuracy (i.e., how confident the inferences are) for the analytics tasks. However, this option also increases transmission delays and requires more computing resources and energy. On the opposite side, sending with a lower resolution and frame rate could reduce the service latency and the energy consumption, but, at the cost of lower inference confidence [6].

These considerations are further complicated by the various application needs and their operating environment. For instance, augmented reality (AR) applications, or time critical services (e.g., AI-enabled collision avoidance) require reliable connectivity, have strict latency constraints for processing (and transmitting) critical frames, and their efficient operation is highly dependent on the inference accuracy [11]. For such cases, and depending on the service, inaccurate (or slow) inferences will have a different impact, e.g., negatively affecting the quality of the AR streaming experience, or, leading to potential injuries (or fatalities) in cases of absent collision warnings. On the other hand, other types of applications require large bandwidth and processing capacity for transmitting and analyzing data, but are less sensitive to latency [12].

This paper *presents a comprehensive set of findings and assessments on the performance and resource requirements of AI-enabled video analytics services over a commercial 5G/4G network in real industrial operating conditions*, i.e., the Port of Piraeus (PCT)[1], one of the leading container terminals in the Mediterranean region founded in Greece. We experimentally explore the emerging trade-offs for various configuration options of video analytics services, and focus on object classification tasks, i.e., people detection. Our target is to engage in a reality check regarding the capability of current (commercial grade) 5G deployments to support demanding AI-enabled video analytics services. This includes a direct comparison with existing 4G deployments, in an effort to quantify the expected advances and contribute to the assessment of the motivation for a network technology upgrade, from a service operational perspective. To further demonstrate the potential of a programmable (and private) compute continuum system, we deploy a virtual environment controlled via a kubernetes (k8s) cluster, to enable efficient and flexible resource management and deployment of containerized AI application across different computing environments and various port assets (extreme/far-edge and cloud). Considering the application needs (e.g., latency critical, or bandwidth demanding), we provide data driven insights that capture the emerging trade-offs, and further discuss how this broader range of system parameters impacts the service requirements at a scale that is pertinent for an industrial port setting.

II. Related Work

Placing AI services across the various tiers of the compute continuum is a research topic of considerable interest that focuses on the following trade-offs; achieving the desired quality of service (QoS) (such as the end-to-end service latency and accuracy) while minimizing the costs (e.g., in energy, bandwidth, or computing resources). Towards this direction, researchers have found various factors that influence the QoS in AI-assisted video analytics. Particularly, in [13] the authors jointly control client side parameters such as the frame rate and video resolution, together with the edge server configuration (e.g., computation resources and CNN model) to efficiently offload mobile AR applications at the edge. [11] proposes a measurement-driven framework that determines the optimal AR service offloading strategy contemplating (among others) how video compression, CNN size, video resolution and battery usage affect the offloading decisions (local or remote) and the respective QoS. [14] proposes a joint accuracy and latency aware deep network structure decoupling solution, that finds the optimal partition of the neural network across the edge devices and central cloud. Similarly, other studies focusing on reducing resource consumption (e.g., energy or compute) with little degradation in accuracy include those reported in [4], [12] or [15].

The majority of the aforementioned studies, provide their valuable insights either through extensive simulations results [5], or via in lab (small scale) experimentation facilities and testbeds [6]. Unlike previous works, we perform a comprehensive assessment of video analytics tasks in a real experimental setting, i.e., the port of Piraeus, using a commercial 5G network, and real port assets that facilitate daily port operations. Our work employs state-of-the-art opensource orchestration tools (such as kubernetes), where we exploit Port infrastructure (extreme/far-edge and private cloud) as a service (PIaaS), to facilitate the orchestration of cloud native video analytics services. We provide data driven insights regarding the network performance and power consumption, as well as inference time and accuracy of object detection tasks in realistic network conditions, that can be further exploited for driving the orchestration decisions of (e.g., latency sensitive, or throughput intensive) 5G&AI-assisted analytics services.

III. Experimentation Platform

This section includes all relevant details for our experimental setup at PCT, and the evaluation criteria for the AI assisted video analytics services. Details regarding the software and hardware components of our system are presented in Table I.

A. Experimental Setup

Platform. Our experimentation platform (Figure 1) is located at PCT, where a commercial, private, 5G non StandAlone (NSA) network is deployed by the local mobile network operator, Vodafone, covering a subset of the port piers. We set up several extreme-edge computing devices in this area, to evaluate various video analytics services in daily port activities. These devices consist of three main components: (i) a 5G interface, namely Teltonica's RUTX50 industrial 5G router that facilitates the cellular connectivity; (ii) an NVIDIA Jetson AGX Xavier (JAX) device [16] for GPU based processing connected to the 5G modem via a gigabit Ethernet connection; (iii) and a 4K camera also connected via gigabit Ethernet to the cellular interface. In addition, a private cloud server (commercial-off-the-shelf Intel x86 system) is deployed at the back-end system of PCT (residing beyond the NSA core), equipped with a GPU NVIDIA RTX 3090. Additionally, we create a virtual platform managed via a k8s system (Microk8s [1]), where the extreme-edge and cloud infrastructure nodes are added as k8s worker nodes that host the workload of containerized AI services.

Convolutional neural networks (CNNs). We exploit the YOLOv5 [17] model, a deep CNN-based object detector model. YOLO faces object detection as a regression problem to predict both the coordinates and the class of multiple-objects [18] in images. The 5th version of the model takes advantage of specially designed layers to improve the performance of the model, in terms of speed, accuracy and lower training times. For example, the spatial-pyramid pooling layer enhances the receptive field of the network [19] and

[1] https://www.pct.com.gr/

allows feeding the network with images of variable sizes [20]. Furthermore, the network is available in 5 different sizes (nano, small, medium, large & extra-large) in terms of network depth and width, the former referring to the number of layers and the latter to the number of parameters per layer. Each model variation is pre-trained on the COCO dataset (see Table I and [17]) and prepared as docker images (i.e., container network functions, CNFs).

Network setup. Regarding the conducted experiments based on 4G connectivity we exploit two LTE configurations: single carrier LTE, operating in frequency band B7 with a 20Mhz channel bandwidth, and LTE-A (advanced) where the device is configured in dual carrier aggregation mode combining B3 and B7 frequency bands, with a 20Mhz channel bandwidth, each. For the 5G experimentation, we used B20 and N78 frequency bands for control (LTE anchor) and data plane (NR user plane) functions, respectively, with a 100Mhz channel. It's important to note that our experimentation results are obtained over a private network infrastructure (i.e., no interference with public networks) and a private cloud system exploited solely for the daily port operations at PCT.

Cluster clock sync. In the context of AI-assisted video analytics, a CNN model makes decisions by processing frames. When focusing on mission critical services with tight delay constraints (e.g., collision warning systems), it is of paramount importance to accurately measure the delay of critical decisions, and thus the transmission and processing delay of critical frames. In a typical setup, a Network Time Protocol (NTP) is used to synchronize the clocks of computer systems over a network. However, the accuracy of an NTP server distribution model, can result in several tens of milliseconds clock difference across the distributed devices. This depends on how symmetric are network routes between the servers and client, how stable is the network delay and client's clock, and how accurate are the servers themselves[2]. To alleviate this drawback, we connect each k8s compute node (extreme-edge and cloud) with a GPS receiver (connected via a serial port) creating stratum-1 devices, which also provide a pulse per second (PPS) signal to more accurately sync the local device clocks with the satellite system. Figure 2 depicts the achieved accuracy, i.e., the local clock offset from the satellite clocks as obtained from chrony. We observe a clock difference of only a few microseconds. In the following, we exploit this negligible offset to accurately measure the one-way transmission delay of packets and frames.

B. Evaluation Setup & Performance Metrics

For the evaluation, we use live video streams in 4K resolution from the high definition cameras installed at PCT. The video is encoded with H.264 encoding at 20fps. The scene is a part of the port with moving and loaded trucks, as well as quay side cranes performing loading/unloading operations on vessels. In addition, 3 to 10 professional individuals were located 500 meters from the camera's position, and moving around (Figure 3). Unless otherwise stated, for the evaluation that follows we used a video of about half an hour long, resulting in about 30 thousand frames.

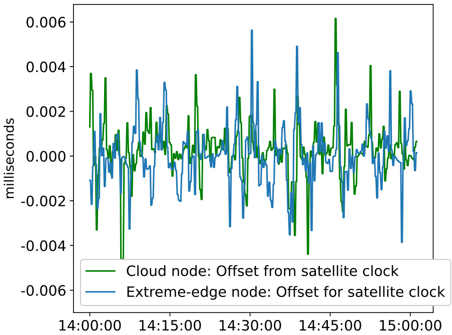


Fig. 2: Time offset between local clock and satellite clock.

Power Consumption. To measure the average power consumption of cloud and extreme-edge CNFs, we exploit NVIDIA's native tools, namely, *tegrastats* for the JAX device and *nvidia-smi* for the GPU RTX 3090, that isolate the power consumption used by the GPU for processing video frames. Hence, we measure the energy footprint of the AI services focusing on the video analytics tasks, i.e., object detection.

Accuracy. The mean average precision (mAP) is the standard performance metric for evaluating the accuracy of a multi-class object detection model, where greater mAP values indicate higher performance [5], [6], [17]. We randomly sampled 3000 images spanning the entire duration of the video and created high quality annotations, i.e. bounding boxes of "person" objects. The resulting dataset was used for the models' performance validation.

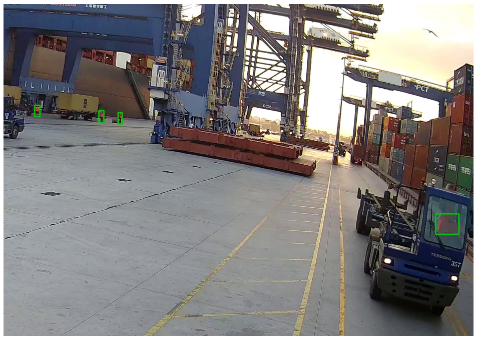


Fig. 3: Sample inferenced video frame from PCT.

Per frame Inference time. Inference time refers to the time it takes for the various YOLOv5 models to generate predictions or make decisions on new, unseen data points. We provide average results for the 30 thousand video frames obtained from the high definition cameras installed at PCT.

Per frame transmission delay. We measure the transmission delay of the 4K frames over the LTE, LTE-A and 5G interfaces. To measure the per frame network delay we employ

[2] https://chrony.tuxfamily.org/index.html

GStreamer[3] tool with the Real-time Transport Protocol (RTP)[4] where we create a 4K video streaming session. On the server side we use *tcpdump* to capture and timestamp RTP packets as they are observed by the network interface card at send time, and similarly for RTP packets at reception (client side). To eliminate clock drift (and deviation) of the devices we employ the stratum-1 clock (GPS/PPS) setup as explained in Section III-A. Lastly, by using the Mark-field of the RTP header [21] we can distinguish all packets that create a video frame, and thus calculate the video frame transmission delay over the various network configurations.

Service delay. This metric aggregates the frame transmission delay and inference time, along with the *network response time*, i.e., network latency, for transmitting the inference result, e.g., an alert, to the end device/applications. Hence, we capture the end-to-end service latency.

Hardware	Description
5G New Radio	Huawei: 5G RRU AAU 5639w
5G Core	NSA: 3GPP Release 15
4K Camera	Dahua: IPC-HFW3841T-ZAS
5G modem	Teltonica RUTX50 industrial 5G router
Extreme-edge node	NVIDIA: Jetson AGX Xavier (Arm64)
Cloud node	Intel x86 system, NVIDIA RTX 3090
Software	**Version**
[Microk8s, Docker]	[1.22.8, 20.10.11]
OS of IoT Device	NVIDIA, Linux Kernel 4.9.253
OS of k8s master node	Ubuntu 20.04 (Focal Fossa)
Object detection models	YOLOv5(n, s, m, l, x)
[CUDA, PyTorch]	[10.2, 1.8]
Parameters	**Description**

Video resolution	SD(640x480), HD(1280x720)
	FHD(1920x1080), 4K(3840x2160)

Network (band, bandwidth)	LTE (B7, 20Mhz) LTE-A (B3&B7, 20Mhz) 5G-NSA (B20&N78,100 Mhz)

TABLE I: Experimentation platform settings and parameters.

IV. Results

1) Network configuration, throughput, latency, and frame transmission delay: The results presented in Figure 4 focus on network metrics. Our objective is to measure the frame transmission delay caused by the different networks when sending (critital) frames for inference at the cloud, as well as their capacity to support massive data (video) flows in real time. The first observation is related to the higher datarate achieved in downlink and uplink measurements for 5G, compared to LTE-A and LTE. Evidently, the additional spectrum resources allow for higher bandwidth availability, enabling higher data rates. We observe (on average) about 480Mbps downlink for 5G, 190Mbps in LTE-A and about 100Mbps for LTE, whereas in uplink we observe about 120, 90 and 30Mbps, respectively.

For video analytics services, the uplink capacity of the system is more critical when data need to be uploaded to the cloud for inference. Figure 4d reports our datarate measurements from one of the cameras[5] (about 9.5Mbps on average) across a working shift of 7 hours (s1 to s7, x-axis). At Piraeus port, daily port operations take place over an area spanning approximately 3Km^2. Based on our observations, to upload the 4K digital footprint of this massive area (i.e., the input for the video analytics services), a significant number of basestation (or radio) units need to be installed (maintained, updated, etc.) increasing operational and capital expenditures. For delay tolerant applications, and to save bandwidth, an intelligent scheduler can assist by orchestrating AI services to the different levels of the compute continuum, considering also the service needs for quick and (or) accurate inferencing (c.f. Section IV-2).

When time critical services are considered (e.g., collision warning systems), a decision whether to offload inferencing at the cloud or execute locally (extreme-edge) needs to be considered. This decision is driven based on how fast the network can transmit critical frames, and on how accurately and fast the AI service can infer, compared to a local execution. Figure 4e shows our measurements corresponding to the transmission delay of 4K frames for the different networks. We observe an average frame latency of about 35ms for 5G, 50ms for LTE-A and 70ms for LTE. Evidently, the 5G network provides the faster medium for delivering high resolution video frames which is pertinent for applications with real time constraints.

Considering network latency, we provide our measurements for packet round-trip times (RTT) in all network configurations measured via ping. For the LTE configurations (no significant difference is observed between LTE and LTE-A) we recorded about 28ms RTT time (on average), whereas in 5G we measured latency of about 18ms. These values represent the *response time* of the network for delivering the results to the end device, e.g., a collision alert. In the following, we couple the network dependency (i.e., frame transmission delay and response time) and AI dependency (fast inference and accurate inference) and elaborate on the final service delay that will drive the orchestration decisions, i.e., cloud or extreme-edge.

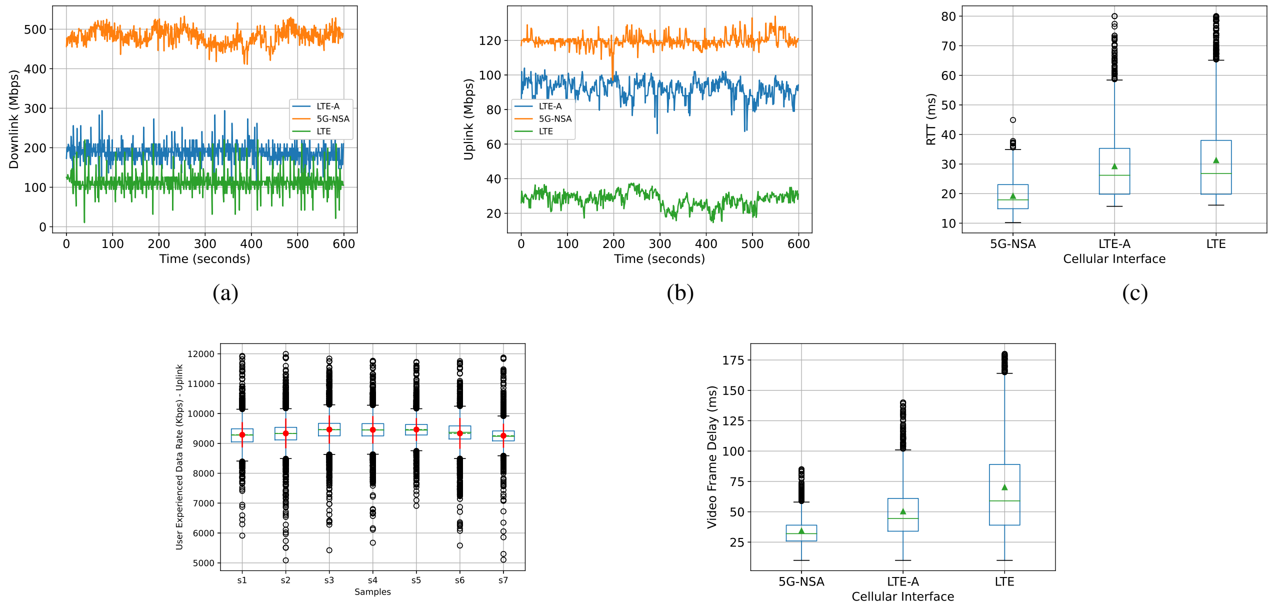
2) Video frame size, inference time, mAP, and power consumption: Figure 5 illustrates the effect of the video frame size (x-axis) and CNN model size (y-axis) on the inference time, power consumption, and mAP for extreme-edge and cloud deployments. Typically, higher resolution video frames contain more pixels, which leads to a larger input for the CNN models. This involves a higher number of computations required for convolution and pooling operations, which increases the inference complexity, and thus the processing time per frame. Similarly, given a constant video frame size, a larger model has more layers and parameters, which indicates more computations needed to obtain a result from the model. However, there is trade-off between inference time and accuracy, when either frame or model size is increased (or both).

The heat-maps of Figure 5 capture these features, where we observe that inference time increases when either a higher video resolution or a larger CNN model is used. Our ob-

[3] GStreamer is an open-source multimedia framework that provides a pipeline-based architecture for creating multimedia services.

[4] RTP is a network protocol used for the delivery of real-time media data over IP networks [21].

[5] Different settings (encoding, video quality or fps) will affect the datarate.



(a)

(b)

(c)

(d)

(e)

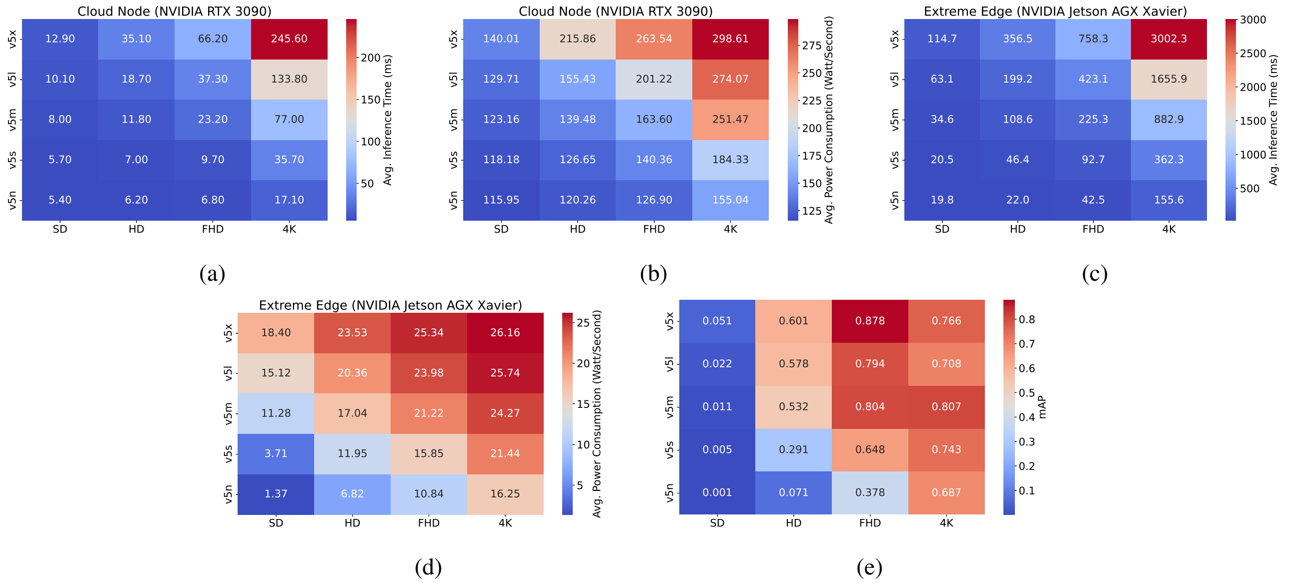
Fig. 4: (a-c) Cellular networks (LTE, LTE-A and 5G) benchmarking based on ping and iperf3 tools for latency and throughput measurements. (d) 4K (uplink) datarate streaming measurements. (e) Per frame transmission delay on various network settings.

servations are inline with other similar studies e.g., [5], [6]. Indicatively, for an HD frame and YOLOv5s model, we observed 7ms inference time per frame (or 142fps of inferenced video streaming) at the cloud, and 46ms (or 21fps of inferenced video streaming) for the extreme-edge case, on average. Evidently, the higher available compute resources at the cloud node allow for much faster processing time of video frames, in contrast to the extreme-edge, however, at the expense of frame transmission delays. Nonetheless, before coupling the network dependency and AI dependency with the service orchestration decisions, we need to further provide our conclusions on the accuracy of the various models.

Figure 5e shows the mAP values for the various AI models under the same configurations. The frame size is the main contributor to a model's accuracy, which is expected considering the distance of the individuals located in the scene. Starting from the left column with the smallest image resolution and moving to the right columns the average precision of every model increases, up to FHD. Long distance objects own small areas of an image in terms of pixels. When the size of an image is scaled down, its quality is reduced as well. Small objects are now described by even less pixels and an object detection model fails to detect them (more false negatives, less true positives). Moving from FHD to 4K, the performance of the nano and small models is better. The medium model's performance improvement is trivial, which indicates moving to a 4K resolution is redundant. On the other hand, the large and extra-large models' performance with 4K resolution images is worse than FHD images. In 4K less objects were detected correctly (true positives) along with more objects that did not exist (false positives), thus reducing the average precision. We assume this behaviour can be explained because all YOLOv5 models were trained [17] with images of 640x480 and 1280x720 and/or the training dataset did not include relative scenes, capturing features of a port environment.

Given the presented results in section IV-1 which benchmark the AI service dependency on the network (i.e., frame transmission delay and network response time), we observe that an accuracy close to 80% for the inference results is obtained with YOLOv5m and FHD frames. This threshold is chosen taking into account a collision warning service's strong need for accuracy. For this configuration, the extreme/far-edge device service delay (i.e., no network dependency) requires about 225ms for inferencing (Figure 5c), whereas if the service is offloaded to the cloud, we observed (on average) 23ms of frame processing time, 35ms for frame transmission delay (Figure 4e), and 9ms (i.e., half RTT, Figure 4c) for the network response time, aggregating a total *service delay* of about 67ms. Hence, based on our experimental driven results, offloading time critical services to the cloud, prevails.

In Figures 5 (b and d) we focus on power consumption. Particularly, we observe similar qualitative results, i.e., the consumed watts/second increase, as the video frame or CNN model size grows. We focus our second remark on the significantly different energy footprint that the AI services cause to the different devices. The JAX node (extreme-edge) is built around the Volta architecture, which is optimized for AI and deep learning workloads while prioritizing power efficiency [16]. On the other hand, RTX 3090 (cloud), a high end desktop GPU, is based on the Ampere architecture designed primarily for high-performance computing applications (e.g., gaming), prioritizing raw performance over power efficiency. In addition, a significantly higher number of compute unified device architecture (CUDA) cores are available at RTX 3090



(a) (b) (c)

(d) (e)

Fig. 5: (a-d) Cloud and extreme/far-edge experimental results on inference time and power consumption, and (e) inference accuracy (mAP), for various frame resolution and CNN model size configurations.

GPU (i.e., 10496 vs 512), making it more powerful but also more power-hungry. Note that the absolute measurements will change, e.g., if a more energy-prudent cloud/extreme-edge GPU is used, nonetheless, we expect similar qualitative results. Hence, for delay tolerant services, extreme-edge devices are preferred for minimizing the energy consumption.

V. Conclusion

5G is expected to pioneer the dynamic landscape of the industrial sector, for companies that face growing pressures to optimize operational efficiency, enhance safety/security protocols and minimize costs. With AI-enabled video analytics at the forefront of these innovations, the 5G compute continuum offers a programmable playground, to address the service requirements and make agile decisions at a scale that is relevant for an industrial ecosystem. In this context, the current study presented a holistic assessment of AI-enabled video analytics services over commercial grade cellular networks. The evaluation considered the service requirements, network capabilities, power consumption and AI-related parameters, and provided data-driven insights for the performance and limitation of AI-enabled video analytics in a real port setting.

Acknowledgement: This paper is co-funded by the EU under the H2020 Research and Innovation Programme and the 5G-LOGINNOV project (grant agreement No 957400).

References

[1] M. Volk and J. Sterle, "5g experimentation for public safety: Technologies, facilities and use cases," *IEEE Access*, vol. 9, 2021.

[2] W. Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Transactions on Multimedia*, vol. 21, no. 3, 2019.

[3] A. Lagorio, C. Cimini, *et al.*, "5g in logistics 4.0: potential applications and challenges," *Procedia Computer Science*, vol. 217, 2023. 4th International Conference on Industry 4.0 and Smart Manufacturing.

[4] C.-C. Hung, G. Ananthanarayanan, *et al.*, "Videoedge: Processing camera streams using hierarchical clusters," in *IEEE/ACM Symposium on Edge Computing*, 2018.

[5] P. Yang, F. Lyu, W. Wu, N. Zhang, L. Yu, *et al.*, "Edge coordinated query configuration for low-latency and accurate video analytics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, 2020.

[6] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Edgebol: Automating energy-savings for mobile edge ai," in *Proc. of ACM CoNEXT*, 2021.

[7] N. Makris, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas, "Experimental evaluation of functional splits for 5g cloud-rans," in *IEEE International Conference on Communications (ICC)*, 2017.

[8] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead," *Computer Networks*, vol. 182, 2020.

[9] F. van Lingen, M. Yannuzzi, Jain, *et al.*, "The unavoidable convergence of nfv, 5g, and fog: A model-driven approach to bridge cloud and edge," *IEEE Communications Magazine*, vol. 55, no. 8, 2017.

[10] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, 2019.

[11] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "Deepdecision: A mobile deep learning framework for edge video analytics," in *IEEE Conference on Computer Communications*, 2018.

[12] Y. Li, Y. Chen, *et al.*, "Mobiqor: Pushing the envelope of mobile edge computing via quality-of-result optimization," in *IEEE 37th International Conference on Distributed Computing Systems*, 2017.

[13] Q. Liu and T. Han, "Dare: Dynamic adaptive mobile augmented reality with edge computing," in *IEEE 26th International Conference on Network Protocols (ICNP)*, 2018.

[14] H. Li, C. Hu, J. Jiang, *et al.*, "Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution," in *IEEE 24th International Conference on Parallel and Distributed Systems*, 2018.

[15] J. Jiang *et al.*, "Chameleon: Scalable adaptation of video analytics," in *Proc. of the Conference of the ACM Special Interest Group on Data Communication*, Association for Computing Machinery, 2018.

[16] NVIDIA, "https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit,"

[17] G. Jocher, "Yolov5 by ultralytics (version 7.0)," vol. https://doi.org/10.5281/zenodo.3908559, 2020.

[18] R. Joseph, D. Santosh, G. Ross, and F. Ali, "You only look once: Unified, real-time object detection," *https://arxiv.org/pdf/1506.02640.pdf*, 2015.

[19] B. Alexey, W. Chien-Yao, *et al.*, "Yolov4: Optimal speed and accuracy of object detection," *https://arxiv.org/pdf/2004.10934.pdf*, 2020.

[20] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *https://arxiv.org/pdf/2004.10934.pdf*, 2015.

[21] RFC3550 *RTP: A Transport Protocol for Real-Time Applications*.