ALFA BK UNIVERSITY
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
FACULTY O F COMPUTER SCIENCE

ALFATECH

Proceedings of Conference

UDK: 004.89:71
711.45:004.7
COBISS.SR-ID 148894985
DOI: 10.5281/zenodo.12594418
Original scientific paper

COMPUTER VISION GATEWAY FOR REAL-TIME ON-SITE ANALYTICS IN SMART CITIES AND PERI-URBAN AGRICULTURE

Branko Brkljač[99]; Milan Brkljač[100]; Boris Antić[101]

Abstract

Current state of edge computing technology is characterized by high level of integration and increasing device availability for different application domains. In the design of computer vision systems for real-time analytics in smart cities there are various components that require significant engineering effort in order to be efficiently deployed and managed by system operators. This is especially true in the case of edge devices performing on-site processing or being deployed as generic computing platforms for different application scenarios. In such cases, continuous development cycles can be affected by the need to integrate various vendor specific software frameworks and the lack of unified middleware. In this paper we propose an architecture of computer vision gateway based on open-source software capable of simultaneously performing processing of different input data streams, artificial intelligence (AI) based inference, and generating output data streams in the form of processed video or extracted analytics. System architecture is demonstrated on heterogeneous computing edge device with hardware acceleration capabilities in both video coding and AI domains. Through proposed unification of data acquisition, information processing and data distribution, gateway allows easier design of novel vision-based applications in settings corresponding to the concepts of smart cities and peri-urban agriculture.

Keywords: *edge devices, computer vision, real-time processing, video stream analytics, middleware for the edge (middlewedge)*

Introduction

Camera sensors are ubiquitously present in modern cities [1], including their neighbourhoods that are characterized as peri-urban environments [2]. In the context of smart cities and data

[99] Branko Brkljač, 1986, PhD, Assoc. prof., Univeristy of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Republic of Serbia, brkljacb@uns.ac.rs https://orcid.org/0000-0001-7932-6676
[100] Milan Brkljač, 1988, PhD, Asst. Prof., Alfa BK Univeristy, Faculty of Finance, Banking and Auditing, Bulevar maršala Tolbuhina 8, 11070 Novi Beograd, Republic of Serbia, milan.brkljac@alfa.edu.rs https://orcid.org/0000-0002-0617-973X
[101] Boris Antić, PhD, 1973, assoc. professor, University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Republic of Serbia, antic@uns.ac.rs https://orcid.org/0000-0003-2504-9479

ISBN: 978-86-6461-074-2

driven management of infrastructure, real-time information from visual domain play important role for timely decisions, monitoring and urban planning activities [3]. Vision based analytics are also exploited in different agricultural applications [4], especially in automated crop management and field surveying using ground based [5], airborne [4] or satellite [6] imaging platforms. In all cases, besides effective task specific operations, design requirements are consistently being pushed towards capabilities providing on-site processing [7-8], low power consumption [9], small form factor design [10] and easier integration into the broader cloud and edge infrastructure [11]. In addition, development of novel applications [12-14] and continuous development cycles require rapid and unified prototyping approaches [15], with modular design [16] and lower entrance barrier [17-18]. As the result, in the market are offered various software solutions and frameworks that are abstracting many complexities involving hardware and software integration for both cloud based analytics, as well as embedded vision on the edge devices. The range of software tools and abstraction levels is very broad and goes from generic open source libraries to no-code graphical user interfaces based on proprietary solutions and cloud infrastructure. In such setting, the sweet spot for many system designers is to have generic testbed that integrates the best from both worlds, by relying on generic open source software frameworks for signal acquisition, processing and distribution, and at the same time generic support for efficient deployment of advanced analytics and artificial intelligence (AI) on the proprietary hardware platforms. Such vision system, although consisting of both hardware and software components can be easily transferred and adapted to different tasks and working environments, thanks to open software standards and AI model architectures. Combined by hardware acceleration capabilities, which are usually platform specific and proprietary technologies, such edge processing systems equipped with appropriate communication links can be described as computer vision gateways (CVGs) with advanced on-site video stream analytics and media server capabilities, Fig. 1. In a narrow sense, such systems refer to specific software frameworks and libraries providing advanced streaming and analytics capabilities regardless of specific hardware platform, Fig. 1.

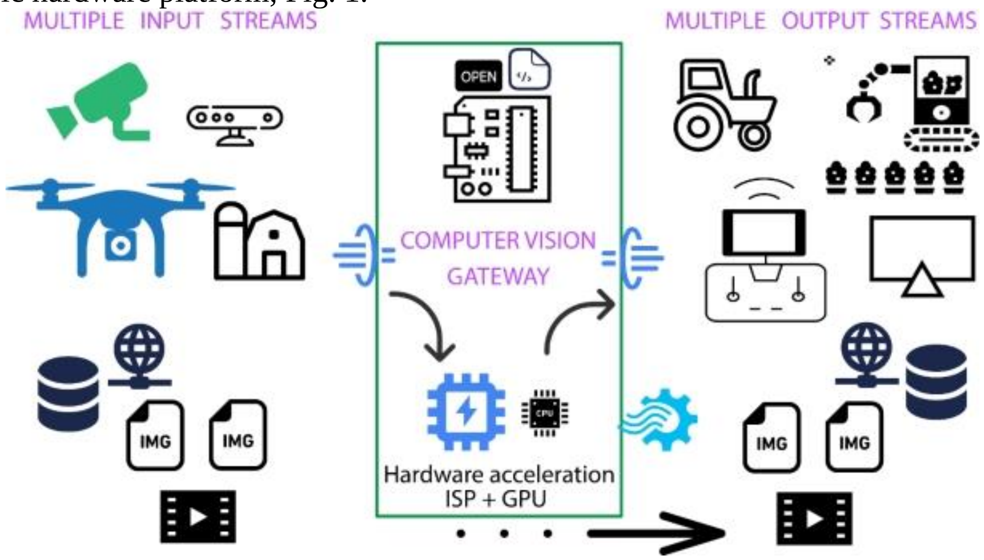


Figure 1: Designed computer vision gateway for real-time on-site video analytics

In this paper we present design and implementation of one novel CVG solution based on several open source software frameworks, adapted to specific requirements of the chosen

ISBN: 978-86-6461-074-2

system on chip (SoC) hardware platform and video capturing device. Implemented functionalities support various application scenarios in smart cities (SC) and peri-urban agriculture (PA), and can be deployed on both fixed and mobile operation platforms (equipment mounted on fixed hosts like street poles or buildings, or mobile platforms like road vehicles, agricultural machines or unmanned aerial vehicles, UAVs). The rest of the paper is organized as follows. In Section 2 we provide a brief overview of SC and PA tasks that can benefit from the proposed CVG, and also refer to technical systems with similar design and discuss their operations. Functional elements of the proposed CVG along the hardware specific details are discussed and presented in Section 3, while in Section 4 an overview of different experimental setups tested with developed CVG system are presented. Finally, in Section 5 we indicate future research directions and space for further improvements.

2. Related work

Technological advances in embedded vision hardware and camera devices have brought increased interest for different applications of vision technologies (ViTech) in many diverse fields, including smart cities (SC), e.g. [8], and peri-urban agriculture (PA), e.g. [5]. Such systems are usually installed or run on-site and provide near real-time information about observed processes and phenomena of interest. Depending on the level of integration and device capabilities, information processing is performed at the same place where visual cues are collected [10], or at the nearest edge server node of the broader data infrastructure [16]. In terms of data engineering and processing pipelines, raw visual data in forms of multiple live image or video streams need to be processed and analyzed on-the-fly, or transmitted to the nearest cloud infrastructure before further analyses. When the video analytics is performed locally, at the same video capturing device or on-site, it results in distribution of significantly smaller amount of information to the downstream tasks. Such information can consist only of data logs and specific insights coming from the locally executed inference engine, significantly reducing the required data rates and data management in distributed system. Similarly, data privacy issues can be more easily resolved by controlling the amount of private information coming from the smart device capable of providing advanced video stream analytics.
When it comes to collecting visual data in SC, dominant infrastructure platforms are still the fixed mounted systems, with the classical networked video surveillance design [1]. However, there is a general trend towards more flexible data collection platforms, enabled by novel imaging systems and embedded devices. In the future, mobile platforms equipped with intelligent edge devices, like UAVs [12] and vehicles [5, 14], will be a significantly more present. Such platforms provide high temporal and spatial resolution of observations, and are considered as part of the perception layer [12] of the urban spatial management. In the realm of digitized agriculture, in-field robotics heavily relies on visual perception provided by similar edge devices mounted on agricultural machines and equipment. There are also numerous examples of unmanned aerial systems [4] equipped with camera sensors that are intermittently or periodically used for conducting specific field survey operations in PA.
The applications of the systems like the proposed computer vision gateway can be broadly divided into three categories: 1) efficient collecting of raw sensor data, without significant on-device inference, which is postponed for later off-line analyses or left to downstream

ISBN: 978-86-6461-074-2

infrastructure nodes with more resources; 2) on-device processing of acquired video data with the goal of providing real-time on-site data analytics, 3) performing complex visual perception tasks that are part of machine automation or robot control. The last two application categories mostly differ in the goal of performed on-device inference, while the first one emphasizes efficient management of acquired and processed image and video streams. This is also illustrated in Fig. 1, where the tasks characteristic for efficient visual data management and network integration, shown in the lower part of the image, are foundation for more advanced tasks related to application specific AI based inference. In that sense, out of the box hardware accelerated video codec support for mainstream standards, like [19], or future [20], is a usual prerequisite for low power consumption and efficient utilization of computing resources in scenarios where CVG has to transcode or transmit multiple input and output video streams on-device, besides the main inference stage.

In order to reduce costs, many of the existing SoC edge devices with hardware accelerated processing capabilities are made with generic functionalities that can be adapted to specific tasks. Heterogeneous computing environments are thus easily utilized in user defined instances of processing pipelines enabled by generic software frameworks like [15, 17-18, 21-22]. Proposed CVG solution follows similar design practice, where the inference stage can be implemented by invoking hardware optimized AI model instances, through hardware specific neural inference engine. Such approach provides modular design, easily adaptable to different use case scenarios, including the ones with CVG mounted on an UAV platform or a vehicle.

Thus, in contrast to e.g. [23], where the archived aerial and street level-imagery are used to generate insights off-line, or e.g. [24] where the social media-based landscape studies utilizing crowdsourced imagery are reported, CVG on board of smart mobile platform can be used to directly collect or update volatile information of interest to some data catalogue or a service, without the need for transmitting and storing all video data or other sensor readings.

According to [12], on-line data transmission include UAV flight instructions, flight status, sensor data, but also in-flight data processing results, denoted as edge intelligence. In that sense, UAV experiment involving proposed CVG solution is similar to [25], which utilizes the same hardware platform [7] as the multi-purpose on board computer. However, in comparison to [25], which avoids the transcoding step on board UAV and utilizes device primarily for UAV authentication and establishing of WebRTC communication link, the proposed CVG simultaneously performs both transcoding of live camera feed and AI inference, on the same embedded hardware. In the case of multiple CVGs working together in some collaborative setting, described distributed processing capabilities, performed on individual CVG edge devices, can significantly reduce the computational load on the network nodes (edge servers) that are higher in the infrastructure hierarchy. E.g. the task offloading described in the design of [16] assumes that all input video streams are forwarded to the edge server, including the computer vision algorithm execution. Similarly, in [11] the individual components of the CVG (camera stream control, edge gateway control and the computer vision applications) are in general physically distributed over one or more edge servers connected to the cloud via the internet. On the other hand, in the proposed CVG these operations are envisioned to be performed simultaneously on the same device, enabling the fully decentralized edge intelligence on the fixed or mobile physical platform. Proposed design is also similar to [8], since it employs the same inference optimization framework, but without containerization of the execution environment.

ISBN: 978-86-6461-074-2

Type of SC insights that can be provided by the systems relying on CVGs include monitoring of the existing building, transportation and power infrastructure [3, 13], commuting practices, crowd analysis, anomalies detection [8], planning and management of emergency responses in cases of urban flooding, earthquakes or other natural disasters. Important applications are also landscape studies [23-24, 26], where visual information play key role in better understanding of complex urban environments. In the context of PA [27-29], CVG systems can provide visual information about the land use and land cover spatiotemporal dynamics [27], farm operations, crop phenological stages [5], but also facilitate low-level vision tasks, like depth estimation [31-32], in field robotics.

3. System design and implementation

The main objective of CVG design was to provide highly customizable and flexible solution for efficient performing of computer vision tasks in outdoor environments. Implementation requirements were to use commercially available hardware that could be easily acquired from global suppliers and mounted on drones or other acquisition platforms in urban and peri-urban environment. It should also provide high performance computing capabilities and low power consumption. The first phase of system design was to identify the most desirable features in terms of signal acquisition, processing and on-device inference. It also included a strategy for mitigating possible technical issues during envisioned field tests. As the result, functional requirements illustrated in Fig. 2 were proposed. These also include out-of-the-box support for custom stream processing pipelines, as well as deployment of advanced AI models for outdoor visual perception.

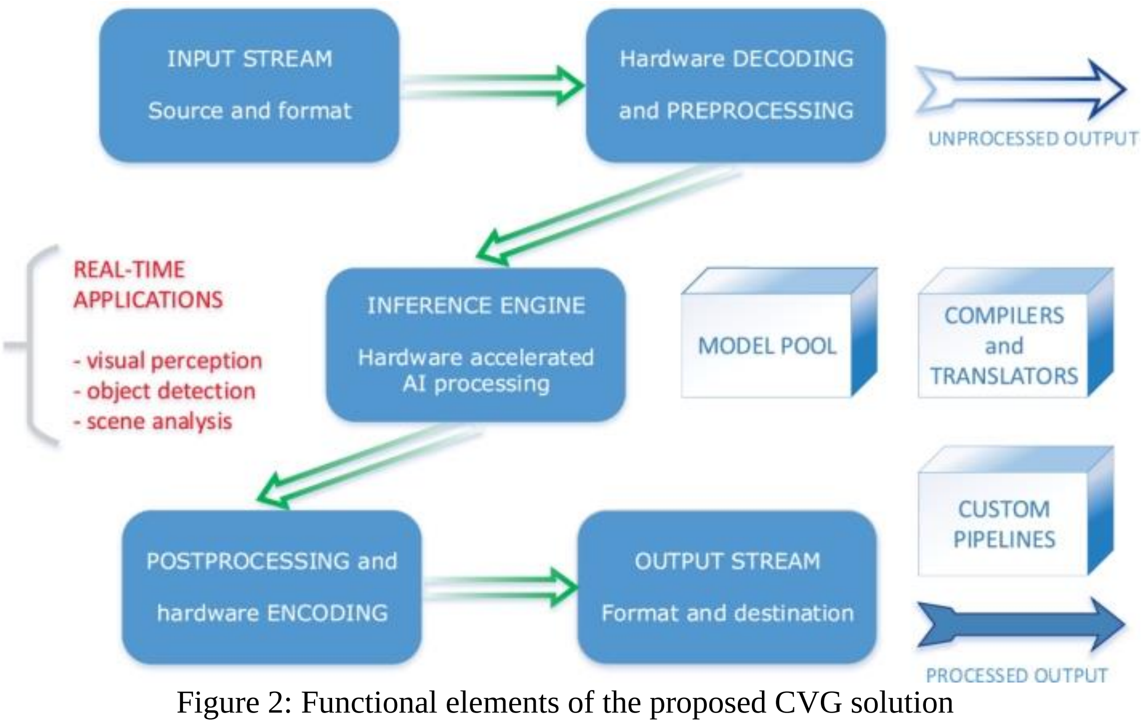


Figure 2: Functional elements of the proposed CVG solution

Generic solution provides support for different image and video sources, data formats and communication protocols. It also anticipates different use case scenarios and possible applications in which CVG would play the central role, as an information source, for data

ISBN: 978-86-6461-074-2

acquisition platforms like UAVs, road vehicles or agricultural machinery. Thus, proposed CVG supports multiple types of input sources: live camera feeds, remote image and video data (transmitted over local or wide area networks by standard communication protocols, like RTP, Real-time Transport Protocol), or data coming from the local storage devices. In all cases, system design allows high level of flexibility in terms of data formats (hardware accelerated image and video decoding, pre-processing and support of parallel input streams running in separate computing threads or processes). These functionalities, implemented as several front-end converters and software adapters for aligning input data formats with capabilities of developed inference solution, are depicted by the first two blocks in Fig. 2. In simple use case scenarios, with task offloading to nearby edge infrastructure, this is the only operation performed by CVG. It assumes streaming of unprocessed video or image signals at rates and in configurations suitable for other data processing subsystems (mounted on the same platform, e.g. drone or vehicle, or present somewhere else at remote location). In such case, input streams are decoded only in the case that additional modifications or adaptations of the input signal are to be performed, e.g. like spatial or temporal interpolation, denoising, or frame rate reduction, or when the input signal is just repacked into the output stream of different type (e.g. received bitstream is packetized and sent to different port and address). However, when the input video stream decoding is required or the output video stream is made of encoded input signal, hardware accelerated codec implementations are necessary. Similarly, live camera feeds are being pre-processed using hardware acceleration, by invoking specialized ISP (Image Signal Processor) and VIC (Video Interface Chip) elements. Described operations are thus performed without over-utilization of available CPU and GPU resources, which are left for later downstream operations depicted in Fig. 2.

As possible test applications of CVG were identified several challenging low-level and high-level vision tasks. This was done in order to design and test functionality of the central functional block in Fig. 2, named “inference engine”. In order to reduce time of AI model development and deployment, existing software frameworks for efficient computational graph (neural network) design, training, description, exchange and deployment were adopted. In general, many AI-based solutions require retraining and fine-tuning when exposed to new operational conditions. Therefore, it is important that all elements support short development cycles. This also includes easy deployment and field testing, which was one of the main goals of implemented CVG. In that sense, adopted inference engine design included using of Open Neural Network Exchange (ONNX) language [21], combined with hardware specific software frameworks for exploiting heterogeneous computing capabilities [22].

In the concrete instance, as the testbed for inference engine was adopted Pytorch library with additional model deployment optimization based on TensorRT framework for high-performance deep learning inference on NVIDIA GPUs. In usual scenario, AI models are developed offline, converted to generic description like ONNX, and then converted to specific execution target, like NVIDIA Tegra chip supporting TensorRT inference engine. Modular CVG design enables that besides hardware accelerated AI processing, system also support adaptive model configuration, by using pool of pretrained computational graphs that can be loaded and executed in the same manner (using developed procedures and tools provided in the gateway). Thus, the same CVG system can have different roles, depending on specific application scenarios, making it versatile tool for SC and PA operations.

CVG was implemented as generic solution based on open source software. In that sense it can be deployed on different hardware platforms supporting hardware accelerated operations. In

ISBN: 978-86-6461-074-2

particular, proposed CVG implementation is mostly based on GStreamer library [15], an open source multimedia framework that can be compiled against different targets, including the NVIDIA SoC systems like Jetson [7-8].
For prototype implementations were utilized Jetson Nano devices with 4GiB of shared CPU and GPU RAM memory, Quad-core ARM Cortex-A57 MPCore processor, 128-core Nvidia Maxwell architecture GPU, 0.4 TFLOPS of computational power, as well as several additional ISP, VIC and codec chips performing input and output video and/or image streams processing. These included NVdec and NVenc chips running at 716.8 MHz, NVjpg and VIC03 running at 627.2, and ISP at 793.6 MHz. Given hardware platform was also extended by custom camera interface board supporting industrial cameras with coaxial interface, which was incorporated into design requirements in order to enable higher flexibility in camera deployment on drones and other platforms.
As the main camera sensor was utilized low light 5MP Sony Starvis IMX335 image sensor [33], integrated onto custom camera board with parallel to serial interface conversion. The same camera interface conversion was also implemented on the host device, Jetson Nano. Besides this specific imaging sensor, Fig. 3 (b), different cameras mounted on drones like DJI Phantom 4 or DJI Matrice 600 were also utilized in some of the experiments.

(a) (b)

Figure 3: ViTech experimental setup performing vision task and transmitting output video to drone operator console (a); CVG mounted on UAV platform (b)

As part of conducted laboratory tests corresponding to different gateway use cases, the peak power consumptions of implemented CVG solution was determined, and accordingly corresponding battery supply for deployment of CVG on board of UAV platform was selected. In general, the peak power consumption of the implemented CVG (including attached video acquisition subsystem) varied between 4.5 W and 18 W, depending on the number and complexity of simultaneously processed input and output image and video streams, and performed on-device AI inference. The peak power consumption was in the case of high utilization of GPU by AI model running in the designed inference engine with TensorRT optimization, and simultaneously encoding input camera feed and output video streams transmitted over network. Although designed system allows adaptive configuration of all functional elements shown in Fig 2, it should be mentioned that some of changes in system configuration require more time, depending on the type of change. E.g. re-initialization of camera feed or redefinition of input and output streams can be relatively fast,

ISBN: 978-86-6461-074-2

while loading of precomputed inference engines or their production can last significantly longer. However, due to modular design, it is easy to move engine production out of the system and perform only AI model deployment, which significantly reduces necessary production time. An important part of CVG implementation was demonstration of easy integration with the existing commercial drone platforms. Therefore, for the design of ViTech testbed was used DJI Matrice 600, which was additionally modified in order to support mounting of Jetson Nano, camera and accompanying battery power supply, Fig. 3.
An instance of the gateway can be run as separate software module (component) independently of other instances, allowing for parallelization, multi-thread processing and scalability. Thus, the main task for the user interested in exploring gateway capabilities is to plan number, format and type of input and output streams, and in-between them stream processing operations utilizing hardware accelerated AI inference engine.

4. Experiments and results

One of the main targets during testing was demonstration of several low-level and high-level vision tasks in SC and PA application scenarios, exploiting AI inference engine in the gateway. The goal of these demonstrations was to highlight the real-time design of both input and output stream processing components, as well as the AI model deployment module implementing inference engine initialization, loading and execution.
CVG implementation consists of three main modules, which can have different elements depending on specified configuration. Thus, before gateway initialization, the user needs to perform certain choice of options, by selecting particular control flags and providing necessary parameters, after which the whole pipeline consisting of: input stream(s), inference engine and output stream(s), is adaptively built according to specified user requirements, initialized and launched.
In this pipeline, the central role belongs to the main processing procedure, which can utilize described AI inference optimization. Besides AI model execution, this central processing module also performs necessary pre- and post-processing of model output, in cases that such additional steps are necessary. The main prerequisite for efficient execution in this module is that developed AI model can be translated into adequate computational graph representation, which is suitable for the given inference engine framework. Thus, designed CVG also has some additional software components and compilers that allow translation between original model representation and optimized computational graph, like conversion from Pytorch model to ONNX, and from ONNX to TensorRT. However, it is not suggested to perform these transformations on the resource constrained platforms like Jetson Nano, but rather off-line and instead just provide the precomputed TensorRT models into the model pool, Fig. 2. We also note that there is a wide variety of model optimization approaches, including number format precision and complexity reduction strategies, which can result in significantly faster inference, e.g. on the same PA image classification task, Table 1.

Table 1: Model execution times against the number format precisions

16 bit floating point precision	32 bit floating point precision
TensorRT model running on GPU: resnet50_pytorch_bsize1__fp16_engine.trt Size of the input batch: (1, 3, 224, 224)	TensorRT model running on GPU: resnet50_pytorch_bsize1__fp32_engine.trt Size of the input batch: (1, 3, 224, 224)

ISBN: 978-86-6461-074-2

Average execution time on GPU: **0.037588 seconds** **** Top 5 prediction scores **** Class ID: 866, Class name: tractor Class ID: 595, Class name: harvester Class ID: 856, Class name: thresher Class ID: 730, Class name: plow Class ID: 561, Class name: forklift Class ID | Likelihood score:[(866, 16.7), (595, 14.82), (856, 13.31), (730, 12.32), (561, 10.445)]	Average execution time on GPU: **0.150072 seconds** **** Top 5 prediction scores **** Class ID: 866, Class name: tractor Class ID: 595, Class name: harvester Class ID: 856, Class name: thresher Class ID: 730, Class name: plow Class ID: 561, Class name: forklift Class ID | Likelihood score: [(866, 16.69704), (595, 14.820633), (856, 13.319504), (730, 12.332409), (561, 10.417375)]

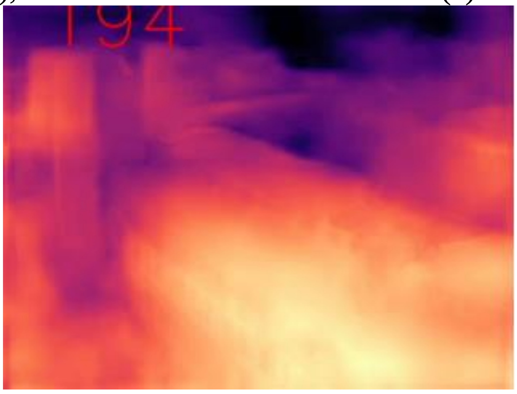
CVG allows multiple settings of camera acquisition modes and pre-processing parameters, which can be easily applied to any camera input. GStreamer library and Video4Linux components are the main interfaces for utilizing proprietary system elements, like hardware processing chips on board of Jetson Nano. Similarly, hardware encoding, which is known as the most expensive video streaming operation, can be performed with full flexibility in the choice of codec configuration (e.g. quantizer settings and predictive coding elements) and encoding strategy (e.g. variable bitrate encoding) for the particular purpose. This also holds for the stream processing of JPEG encoded images, which can be efficiently acquired or emitted from the gateway, without other system resources. Several examples of conducted experiments using implemented CVG, Fig. 2, and the ViTech UAV testbed, Fig. 3, are shown in Figures 4-6.

(a) (b)

(c)

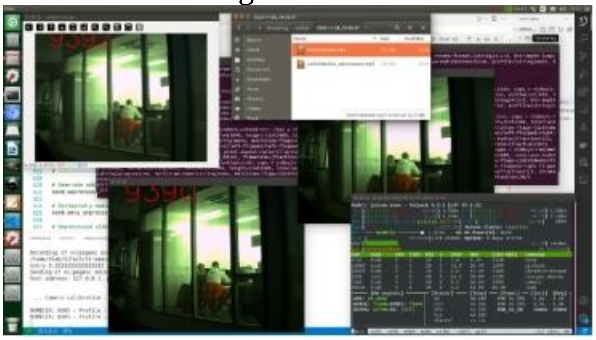
ISBN: 978-86-6461-074-2

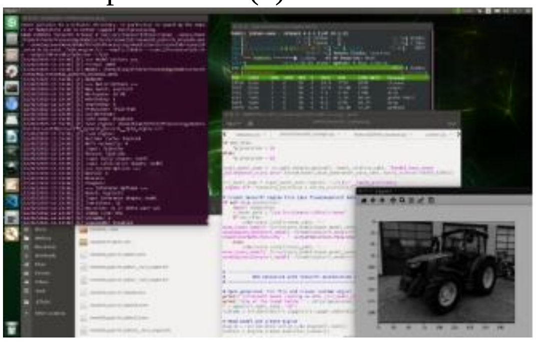
Figure 4: An illustration of semantic segmentation performed by CVG: laboratory tests (a), field tests using UAV mounted camera (b), or video streamed over network (c)



(a) (b)

Figure 5: An illustration of monocular depth estimation [32] running in real-time on UAV CVG inference engine and simultaneous transmission of output stream (b) over network RTP





(a) (b)

Figure 6: Testing of multiple stream processing and low-light camera acquisition (a); PA image classification task resulting in high GPU load and low CPU utilization (b)

5. Conclusions

Proposed CVG has demonstrated that generic and modular software frameworks on top of embedded SoC hardware can provide a versatile tool for advanced video analytics in SC and PA environments. By allowing adaptive processing pipelines and customization, such solution reduces development cycles and facilitates easier integration into larger edge infrastructure. Moreover, it is well suited for both low-level and high-level outdoor vision tasks involving mobile physical platforms like road vehicles, UAVs or agricultural machines. It also reflects the general trend of developing hardware agnostic vision solutions, but does not exclude the use of the existing frameworks, like [17], offering richer set of capabilities. However, seamless integration of similar CVG devices into the edge-cloud continuum still represents a challenging research topic that will require significant effort in the future.

Acknowledgements

The authors would like to acknowledge previous support of H2020 research project Smart4All, financed under the grant agreement No. 872614.

References

[1] Nilsson, F., *Intelligent network video: Understanding modern video surveillance systems*, CRC Press, Boca Raton, USA, 2024, DOI: 10.4324/9781003412205.

ISBN: 978-86-6461-074-2

[2] Sahana, M., et al., Where is the peri-urban? A systematic review of peri-urban research and approaches for its identification and demarcation worldwide, *Remote Sensing*, *15*, (2023), 5, pp. 1-30, DOI: 10.3390/rs15051316.

[3] Nematollahi, M. A., Shahbazi, S., Nabian, N. *Computer vision and audition in urban analysis using the remorph framework*, Springer, 2019,DOI:10.1007/978-981-13-3543-3.

[4] Latif, M. A., An agricultural perspective on flying sensors: State of the art, challenges, and future directions. *IEEE Geoscience and Remote Sensing Magazine*, *6*, (2018), 4, pp. 10-22, DOI: 10.1109/MGRS.2018.2865815.

[5] d'Andrimont, R., Yordanov, M., Martinez-Sanchez, L., Van der Velde, M, Monitoring crop phenology with street-level imagery using computer vision. *Computers and Electronics in Agriculture*, *196*, (2022), 106866, pp. 1-14, DOI: 10.1016/j.compag.2022. 106866.

[6] Nduati, E., Sofue, Y., Matniyaz, A., Park, J. G., Yang, W., Kondoh, A., Cropland mapping using fusion of multi-sensor data in a complex urban / peri-urban area. *Remote Sensing*, *11*, (2019), 2, pp. 1-24, DOI: 10.3390/rs11020207.

[7] Jetson Nano: NVIDIA Maxwell architecture SoC for embedded applications and AI IoT, https://developer.nvidia.com/embedded/jetson-nano [accessed on 03/15/2024].

[8] Pham, H. V., Tran, T. G., Le, C. D., Le, A. D., Vo, H. B., Benchmarking anomaly detection system on various Jetson edge devices, *Advances in Information and Communication*. FICC, 2024, pp 358–374, DOI: 10.1007/978-3-031-53963-3_25.

[9] Magnus, J., Göhringer, D., Millet, P. Eds., *Towards ubiquitous low-power image processing platforms (TULIPP)*, Springer, Cham, Switzerland, 2021, DOI: 10.1007/978-3-030-53532-2.

[10] Baller, S. P., Jindal, A., Chadha, M., Gerndt, M., DeepEdgeBench: Benchmarking deep neural networks on edge devices. *IEEE International Conference on Cloud Engineering*, IC2E, San Francisco, USA, 2021, pp. 20-30, DOI: 10.1109/IC2E52221.2021.00016.

[11] Luu, S., Ravindran, A., Pazho, A. D., Tabkhi, H., VEI: A multicloud edge gateway for computer vision in IoT, *Proceedings of the 1st Workshop on Middleware for the Edge*, Quebec City, Canada, 2022, pp. 6-11, DOI: 10.1145/3565385.3565877.

[12] Xu, H., Wang, L., Han, W., Yang, Y., Li, J., Lu, Y., Li, J., A survey on UAV applications in smart city management: Challenges, advances, and opportunities, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, (2023), pp. 8982-9010, DOI: 10.1109/JSTARS.2023.3317500.

[13] Luckey, D., Fritz, H., Legatiuk, D., Dragos, K., Smarsly, K., Artificial intelligence techniques for smart city applications, *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*, (Santos, E., Scheer, Eds.), ICCCBE 2020, São Paulo, Brazil, 2020, pp. 3-15, DOI: 10.1007/978-3-030-51295-8.

[14] Verma, B., Zhang, L., Stockwell, D., *Roadside video data analysis: Deep learning*, Springer, Singapore, 2017, DOI: 10.1007/978-981-10-4539-4.

[15] GStreamer: Open-source multimedia framework [https://gstreamer.freedesktop.org/, accessed on 03/15/2024].

[16] Liu, P., Qi, B., Banerjee, S. EdgeEye: An edge service framework for real-time intelligent video analytics. *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, EdgeSys, München, Germany, 2018, pp. 1-6, DOI: 10.1145/3213344.3213345.

ISBN: 978-86-6461-074-2

[17] DeepStream: Streaming analytics toolkit based on GStreamer for AI-based multi-sensor processing [https://developer.nvidia.com/deepstream-sdk/, accessed on 03/15/2024].

[18] Ham, M., Moon, J., Lim, G., et al., NNStreamer: Efficient and agile development of on-device AI systems, *IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP, 2021, pp. 198-207, DOI: 10.1109/ICSE-SEIP52600.2021.00029.

[19] Miguélez-Tercero, R., Jiménez-Ruiz, A., Ruiz-Coll, D., Fernández-Escribano, G., Cuenca, P., Analysis of the capabilities of embedded systems in intraprediction video coding, *IEEE Consumer Electronics Magazine*, *11*, (2022), 5, pp. 25-40, DOI: 10.1109/MCE.2021.3084817.

[20] Hamidouche, W., Biatek, T., Abdoli, M., François, E., Pescador, F., Radosavljević, M., Menard, D., Raulet, M., Versatile video coding standard: A review from coding tools to consumers deployment. *IEEE Consumer Electronics Magazine*, *11*, (2022), 5, pp. 10-24, DOI: 10.1109/MCE.2022.3144545.

[21] ONNX: Open Neural Network Exchange, https://onnx.ai/ [accessed on 03/01/2024].

[22] Zhou, Y., Guo, Z., Dong, Z., Yang, K., TensorRT Implementations of model quantization on edge SoC, *IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, MCSoC, Singapore, 2023, pp. 486-493, DOI: 10.1109/MCSoC60832.2023.00078.

[23] Wegner, J. D., Branson, S., Hall, D., Schindler, K., Perona, P. Cataloging public objects using aerial and street-level images, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, 2016, pp. 6014-6023, DOI: 10.1109/CVPR.2016.647.

[24] Ma, R., Furuya, K. Social media image and computer vision method application in landscape studies: A systematic literature review, *Land*, *13*, (2024), 181, pp. 1-22, DOI: 10.3390/land13020181.

[25] Sacoto-Martins, R., Madeira, J., Matos-Carvalho, J. P., Azevedo, F., Campos, L. M., Multi-purpose low latency streaming using unmanned aerial vehicles. *12th International Symposium on Communication Systems, Networks and Digital Signal Processing*, CSNDSP, Porto, Portugal, 2020, pp. 1-6, DOI: 10.1109/CSNDSP49049.2020.9249562.

[26] Meng, Y., Xing, H., Yuan, Y., Wong, M. S., Fan, K, Sensing urban poverty: From the perspective of human perception-based greenery and open-space landscapes, *Computers, Environment and Urban Systems*, *84*, (2020), 101544, pp. 1-14, DOI: 10.1016/j.compenvurbsys.2020.101544.

[27] Follmann, A., Willkomm, M., Dannenberg, P., As the city grows, what do farmers do? A systematic review of urban and peri-urban agriculture under rapid urban growth across the global south, *Landscape and Urban Planning*, *215*, (2021), 104186, pp. 1-12, DOI: 10.1016/j.landurbplan.2021.104186.

[28] Orsini, F., Pennisi, G., Michelon, N., Minelli, A., Bazzocchi, G., Sanyé-Mengual, E., Gianquinto, G. Features and functions of multifunctional urban agriculture in the global north: A review, *Frontiers in Sustainable Food Systems*, *4*, (2020), 562513, pp. 1-27, DOI: 10.3389/fsufs.2020.562513.

[29] Drescher, A. W., Isendahl, C., Cruz, M. C., Karg, H., Menakanit, A., Urban and peri-urban agriculture in the global south, in *Urban ecology in the global south*, (Shackleton, E., et al, Eds), Springer, Cham, Switzerland, 2021, pp. 293-324, DOI: 10.1007/978-3-030-67650-6.

ISBN: 978-86-6461-074-2

[30] Maloku, D., Balogh, P., Bai, A., Gabnai, Z., & Lengyel, P., Trends in scientific research on precision farming in agriculture using science mapping method, *International Review of Applied Sciences and Engineering*, *11*, (2020), 3, pp. 232-242, DOI: 10.1556/1848.2020.00086.

[31] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V., Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*, (2020), 3, pp. 1623-1637, DOI: 10.1109/TPAMI.2020.3019967.

[32] Rudolph, M., Dawoud, Y., Güldenring, R., Nalpantidis, L., Belagiannis, V., Lightweight monocular depth estimation through guided decoding. *IEEE International Conference on Robotics and Automation*, ICRA, Paris, France, 2022, pp. 2344-2350, DOI: 10.1109/ICRA46639.2022.9812220.

[33] Sony IMX 335 Low light CMOS image sensor, https://www.sony-semicon.com/files/62/flyer_security/IMX335LQN_Flyer.pdf [accessed on 03/01/2024].

ISBN: 978-86-6461-074-2