On-the-Edge Inference Enabled Vision System for Smart Cities

Carmelo Scribano[1,2,3], Ignacio Sanudo Olmedo[2], Micaela Verucchi[2], Danda Pani Paudel[1], Marko Bertogna [2], Luc Van Gool[1]
[1] INSAIT, Sofia University "St. Kliment Ohridski", Bulgaria [2] HIPERT, University of Modena and Reggio Emilia, Italy
[3] Institute of Informatics and Telematics, National Research Council, Italy
e-mail: carmelo.scribano@unimore.it, ignacio.sanudo@hipert.it, micaela.verucchi@hipert.it, danda.paudel@insait.ai, marko.bertogna@unimore.it, luc.vangool@insait.ai

***Abstract*—This work introduces a next-generation smart city platform using a novel embedded vision system. i.e., HAura. The HAura system integrates a dual camera and other sensors with a powerful embedded computing unit. The powerful perception stack, based on robust deep learning and computer vision techniques, provides a perfect baseline for implementing a variety of security, traffic management and urban planning policies. Choosing to process images directly on the device and transmit only metadata ensures compliance with privacy and security, as well as bandwidth efficiency. The next evolution of the vision stack will finally improve capabilities by introducing a new multi-task perception model.**

Keywords-smart-city; edge inference; safety; privacy-preserving AI.

I. Introduction

Increasing urbanization brings several challenges regarding public safety, congestion control, and the search for more efficient infrastructure. More urban centers are turning to camera-based smart-city solutions, which use computer vision, machine learning, and built-in data anonymization techniques to monitor public areas in real time, detecting and recognizing vehicles and pedestrians. These technologies are potentially the way to enable a new level of situational awareness and decision-making. In this paper, we present a comprehensive solution for smart-city infrastructure implementation, within the mentioned context. The designed system includes cameras and other sensors, in addition to communications systems. A powerful but efficient embedded computing unit performs local image analysis, leveraging a modern computer vision stack based on deep learning techniques. Running the vision stack on the edge device, instead of on a central compute server, provides all the flexibility and scalability of a fully decentralized architecture. In addition, the images are never transmitted, which makes the system effective in protecting the privacy of citizens. The next chapter discusses the technical hardware and software details of the developed system. In the following one, the details of the current vision stacks are presented. In the end, the development of a next-generation vision stack is introduced, which will largely improve the abilities of the system without overburdening the computational capabilities. The remainder of this paper is structured as follows:

- In Section II, we discuss the smart city application scenario in which we operate the proposed HAura system.
- In Section III, we detail the hardware and software technical characteristics of the HAura system, including the execution stack and data representation format.
- In Section IV, we delve in the details of the proposed preception stack, based on powerful and efficient deep models. We also anticipate ongoing work on the development of a next-generation perception model.
- Section V summarizes this overview and provides additional insights.

II. Outline

A. Motivation

In accordance with projections, it is anticipated that 68% of the global population is expected to live in urban areas by 2050 [1]. Consequently, there is an imperative need for improved city management, particularly in terms of security and safety measures. An urban monitoring system is implemented using a connected camera infrastructure, with several technological and non-technological challenges involved. As pointed out by [2], human monitoring operators are easily overwhelmed by simultaneous monitoring of multiple screens. Therefore, there is a prevailing need for automated and accurate monitoring systems. Current computer vision systems are already used to implement sophisticated systems for traffic monitoring [3][4], road safety, emergency detection [4][5] and urban planning [6]. However, the algorithmic scenario is constrained by the ability to comply with privacy regulations and technological limitations dictated by available economic resources.

The HAura system processes footage on the local edge computing unit, sending only the resulting metadata over a dedicated low-latency network (bypassing the public Internet), achieving an end-to-end latency below 150 ms. In contrast, conventional IP camera setups rely on the Internet, introducing hundreds of milliseconds—or even seconds—of delay. Low-latency is essential to enable the interaction between the smart city and connected vehicles, enhancing vehicle perception by providing critical information.

B. HAura embedded system

The HAura embedded kit, shown in Figure 1, is composed of a dual camera, computing board, and software, and enables real-time identification, geolocalization, and tracking of vehicles, pedestrians, and various road users. HAura computes all the data onboard. The metadata produced by the device is

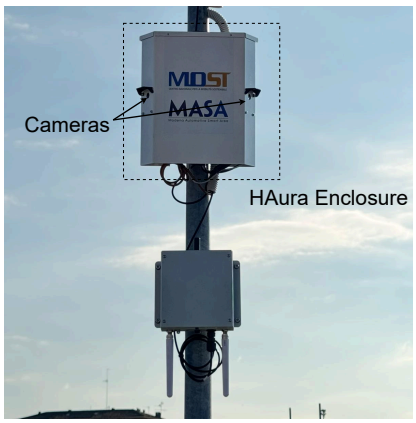


Figure 1. Haura hardware installed at the Modena Automotive Smart Area (MASA) [7]. Modena, Italy.

seamlessly transmitted to a server. Depending on the municipality or the private entity, the server, leveraging the metadata produced by the HAura infrastructure, implements different applications to monitor road users and execute smart urban strategies. At the time of writing, HAura is being implemented in several Italian cities to improve public safety and optimize traffic flow, including Modena, Reggio Emilia, and Torino.

III. Technical Description

A. HAura Technology stack

The proposed device, named HAura, is a smart road side unit designed for safety management and data analysis in smart cities and industrial contexts. Specifically, the system processes data from two cameras continuously, with an image transmission frequency of 10Hz. The metadata produced is sent to a server that can implement any urban monitoring policy utilizing the produced data.

a) Hardware Description: Enclosed in a rugged waterproof case, the HAura's computing heart is based on the Nvidia Orin Nano embedded platform. This choice is popular in the embedded computer vision domain because of the performance of the Nvidia Graphics Processing Unit (GPU) included in the Orin System on Chip (SoC).

- Computing: Specifically, the Orin Nano SoC is based on a 6-core Arm Cortex A78A Central Processing Unit (CPU), an Ampere-based Nvidia GPU with 1024 Cuda cores and 32 Tensor cores. It is also equipped with 8Gb of unified Low-Power Double Data Rate 5 (LPDDR5) memory.
- Sensors: The sensor set comprises two Red-Green-Blue (RGB) cameras. These cameras offer a wide 120° field of view, ensuring comprehensive coverage of the surveillance area. The system supports a resolution from 640x480 to 1920x1080
- Connectivity: Mainly the device is designed for low-latency 5G connectivity. The system also supports Wi-Fi (2.4GHz and 5GHz) and Ethernet. To complement this, it is equipped with a Global Positioning System (GPS) antenna, which is useful for automating the post-installation operations, ensuring accurate localization.

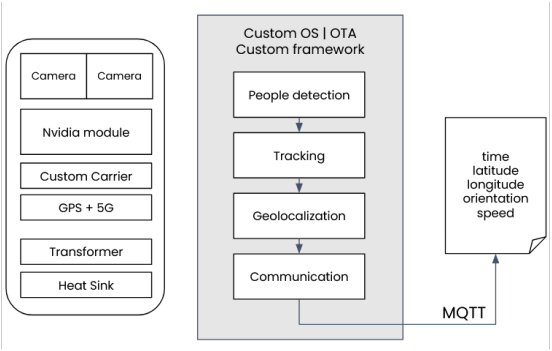


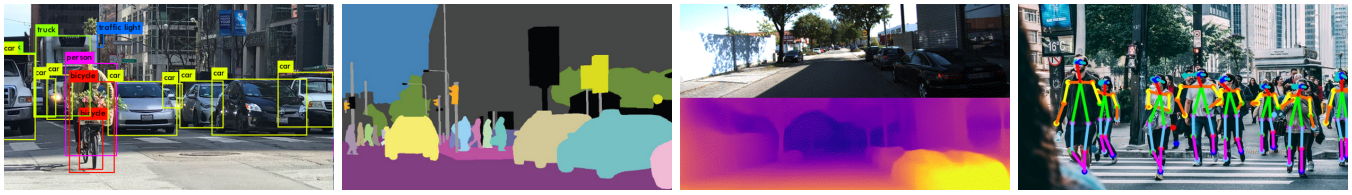
Figure 2. Diagram of the HAura's hardware, processing pipeline, and data produced.

b) Software Description: The full software suite is ready to power a full-fledged smart city infrastructure. At the base of the software stack, we have a custom Linux-based operating system (OS), boasting essential capabilities of remote performance assessment and over-the-air (OTA) updates. A key component, the HAura's perception stack depicted in Figure 2, facilitates real-time object detection and tracking over time. The upcoming frames from both cameras are processed in parallel using the powerful computer vision infrastructure detailed at Section IV-A. The obtained detections include pedestrians and different vehicles (cars, bicycles, vans, buses and motorcycles). The output is processed to recover GPS coordinates of detected objects and perform tracking of detections over time. The resulting JSON, exemplified in Figure 3, includes categorized information represented by numeric IDs (e.g., 0 for a person), latitude, longitude, tracking ID, device ID, and detection timestamp.

```
{
  "camIdx": 0,
  "nObjects": 1,
  "objects": [
    {
      "latitude": 45.06582260131836,
      "longitude": 7.662070274353027,
      "speed": 0.0,
      "orientation": 0,
      "id": 1089,
      "cl": 2
    }
  ]
}
```

Figure 3. Example json snippet with object data.

Only this metadata is transmitted, leveraging the Message Queuing Telemetry Transport (MQTT) protocol. No images of any kind are included (faces, license plates etc.). This choice is an important factor in preserving citizens' privacy and assuring compatibility with the strictest regulations.



(a) Object Detection (b) Panoptic Segmentation (c) Depth Estimation (d) Keypoint Detection

Figure 4. Visual representation of the different tasks implemented.

IV. Computer Vision Stack

A. HAura perception stack

The current version of the HAura perception stack follows a conservative approach based on known techniques that have been proven to stand the test of time The core of the vision system is based on the object detection task, which involves identifying and localizing objects of interest.

Overall, the current perception stack is structured as follows:

- Self Diagnostic: A small Resnet-18 [8] model, trained on a specialized proprietary dataset, is capable of classifying incoming images to detect abnormal conditions such as intense dirt or occlusion sources. This model is run sporadically (every several minutes) and is therefore not relevant to the overall latency.
- Object Detection: The core of the vision system is based on the YoLo-V4 [9] object detector, trained on the 80-classes MS COCO dataset [10]. YoLo-V4 is preferred to newer models because of its good balance between performance and low inference cost. Of the 80 classes, we select 6 of interest (person, car, bike, bicycle, truck, bus).
- Mullti-object tracking: We use an extremely efficient tracker based on ByteTrack [11]. This tracker works by associating the detections of successive frames and does not require additional deep models, ensuring excellent execution performance.
- Mapping to GPS: Using calibrated camera extrinsic, the object detections are mapped to GPS coordinates using the inverse perspective mapping technique.

The vision stack runs entirely on NVidia embedded hardware. Model inference is accelerated using the proprietary TensorRT framework, currently version 8.6.1.

TABLE I. PERFORMANCE METRICS FOR DIFFERENT NUMBERS OF CAMERAS. THE REPORTED LATENCY (MILLISECONDS) OBTAINED BY AVERAGING OVER 1000 FRAMES.

Stage	1 Camera	2 Cameras	4 Cameras
Detection	19.98	32.41	64.85
GeoTracking	0.77	0.78	0.85
End2End	21.15	33.63	66.96

In Table I, we report an analysis of inference performance. Detection only replays the inference time of YoLo-V4 network. GeoTracking refers to the combination of Bytetrack tracker and Inverse Perspective Mapping (IPM) in GPS coordinates. End2End latency in the end includes the complete execution cycle, including decoding the image and processing of the results in the desired format. From a performance standpoint, for precise identification, the system guarantees the following recognition ranges: 40 meters for pedestrians, 45 meters for cyclists, and 50 meters for cars.

B. Future Multi-Task perception model

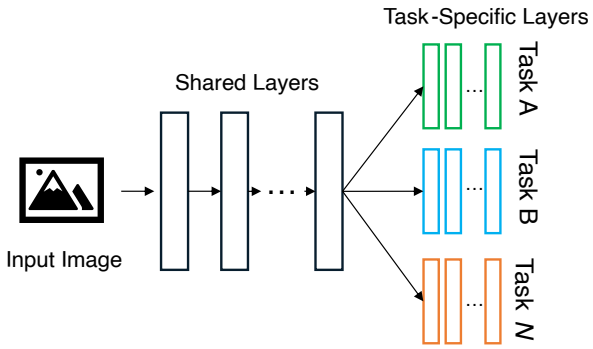


Figure 5. Outline of a multi-task learning architecture.

The current development effort is focused on the next generation of the perception stack. The underlying deep learning model is based on a multi-task learning paradigm [12]. A multi-task approach involves a single model being able to produce detection for multiple distinct tasks in a single forward pass. Compared with a classical approach, in which a specific and separate model is trained for each task, the multi-task approach has several advantages. Typically, as exemplified in Figure 5, a single backbone is used for feature extraction from the image. Only a shallow decoder is added for each task. This implies that the computational cost introduced by each additional task is marginal to the overall computational cost. In addition, in a positive-transfer effect, simultaneous learning of related tasks introduces a regularization effect that can potentially boost performance and generalization ability compared with single-task learning.

We are currently developing the model to include the following tasks, depicted in Figure 4, that presents an immediate application for numerous downstream applications.

- **Object Detection**: This is the same task underlying the current stack. Accurate prediction of bounding boxes remains a crucial element. The new enhanced model will lead to improved detection performance.

- **Panoptic Segmentation**: This task extends beyond object detection by assigning a semantic class label to each pixel while simultaneously assigning a unique label to each object instance (e.g., individual vehicles or pedestrians). Panoptic segmentation will enable a more comprehensive understanding of the urban scene.
- **Depth Estimation**: This task involves predicting per-pixel depth values to reconstruct the three-dimensional structure of the scene. This task is essential for estimating distances to objects and understanding their spatial relationship in the real world. Accurate depth data will enhance functionalities such as collision detection, and more precise GPS localization.
- **Keypoint estimation**: This task focuses on identifying and localizing critical points on objects, such as human body joints for pedestrians. This task enables fine-grained analysis of movement patterns of pedestrian, essential for advanced techniques of for behavior prediction.

The tasks of object detection, segmentation and keypoint estimation are all trainable on the MS COCO dataset, which provides the required annotations. The depth estimation ground-truth is not included, although there are dedicated datasets such as NYU Depth V2 [13] or KITTI [14], training a multi-task dataset on heterogeneous datasets is non-trivial. To overcome this limitation, we are considering leveraging pseudo labels for COCO images obtained using a powerful foundation model like DepthAnything [15].

C. Foundation Backbone

A second innovation, in addition to the multi-task paradigm, is to base the feature extractor of the new model on a powerful foundation model. In vision, a backbone foundation is obtained by pre-training the model with special techniques on a large scale, millions or even billions of images. A prominent example is Dino-V2 [16]. This backbone has been trained on a large dataset of 142 million images using a self-supervised learning approach derived from [17]. With the large-scale pre-training, the foundation models learn strong feature extraction ability, when fine-tuning on downstream tasks therefore the final model will show exceptional performances and strong generalization ability. The main disadvantage is that it is not possible to replicate pretraining on a large scale because of the huge costs and lack of proprietary training data. Therefore, we must start from the pretrained models released by the authors and keep the same model architecture. In particular, Dino-V2 is based on the Vision Transformer (ViT) family of models [18], which are generally considered expensive in terms of computational resources. For this reason, a crucial phase of the work is focused on reducing the computational cost of ViT models while maintaining compatibility with the pretrained weights of Dino-V2.

D. Computing cost reduction

Reducing computational cost, hence inference time, without degrading performance is a key goal for inference on edge devices. The TensorRT inference framework provided by NVIDIA already implements a large set of generic techniques to accelerate inference: the proprietary TensorRT compiler is capable of optimizing the inference graph, performing complex fusion of operations and carefully selecting inference kernels to maximize performance. In addition, different tools are provided to implement techniques such as quantization and pruning [19]. In addition, there is extensive scientific literature of techniques to mitigate the inefficiencies of specific categories of models. Our current work includes developing a specific novel technique to further accelerate the inference of the ViT models on which DinoV2 is based that we use as the backbone for the multi-task model.

V. Conclusion

In this paper, we introduced the HAura hardware and software stack, a generic platform for smart city infrastructure. Our design leverages on-edge inference, ensuring privacy protection by transmitting only data that complies with existing regulations, thereby reducing the risk of exposing sensitive information. The platform allows adopters to develop front-end applications that utilize aggregated metadata for a variety of purposes, from real-time traffic monitoring to long-term urban planning. While our initial evaluations are promising, we recognize that further work is needed to thoroughly assess privacy guarantees and regulatory compliance in diverse settings. Future efforts will focus on developing a comprehensive front-end platform with the most requested functionalities and on evaluating the use of Large Language Models to simplify aggregate data querying. Although specialized systems are available for individual applications, to the best of our knowledge our platform represents a unique step toward a universal, upgradeable, and reconfigurable solution for smart city infrastructure.

Acknowledgement

This research was partially funded by the dAIedge project (HORIZON-CL4-2022-HUMAN-02-02, Grant Agreement Number: 101120726) and the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure). C. Scribano work was partly funded by the Partenariato Esteso PE00000013 - “FAIR", funded by the European Commission under the NextGeneration EU program, PNRR - M4C2 - Investimento 1.3.

References

[1] *World Urbanization Prospects: The 2018 Revision*. UN, Aug. 2019, ISBN: 9789210043144. DOI: 10.18356/b9e995fe-en.

[2] J. Aldridge, *CCTV operational requirements manual: Who will be the first to test your CCTV security or safety system?* 1994.

[3] G. M. Lingani, D. B. Rawat, and M. Garuba, “Smart traffic management system using deep learning for smart city applications,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0101–0106. DOI: 10.1109/CCWC.2019.8666539.

[4] N. T. Ha *et al.*, “Leveraging deep learning model for emergency situations detection on urban road using images from cctv cameras,” in *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, IEEE, 2022, pp. 1–5.

[5] R. Nouisser, S. K. Jarraya, and M. Hammami, "A review of vision-based abnormal human activity analysis for elderly emergency detection," in *2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, IEEE, 2023, pp. 1–6.

[6] N. Naik, S. D. Kominers, R. Raskar, E. L. Glaeser, and C. A. Hidalgo, "Computer vision uncovers predictors of physical urban change," *Proceedings of the National Academy of Sciences*, vol. 114, no. 29, pp. 7571–7576, 2017, [Accessed 16-02-2025]. DOI: 10.1073/pnas.1619003114. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1619003114.

[7] *MASA – a living lab for automated driving – Modena Automotive Smart Area — automotivesmartarea.it*, https://www.automotivesmartarea.it/, [Accessed 16-02-2025].

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, 2020. arXiv: 2004.10934 [cs.CV].

[10] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[11] Y. Zhang *et al.*, "Bytetrack: Multi-object tracking by associating every detection box," in *European conference on computer vision*, Springer, 2022, pp. 1–21.

[12] S. Vandenhende *et al.*, "Multi-task learning for dense prediction tasks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3614–3633, 2021.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, Springer, 2012, pp. 746–760.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[15] L. Yang *et al.*, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 371–10 381.

[16] M. Oquab *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[17] J.-B. Grill *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[18] D. Alexey, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv: 2010.11929*, 2020.

[19] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.