CrowdVisor: an Embedded Toolset for Human Activity Monitoring in Critical Environments

Marco Di Benedetto*, Fabio Carrara*, Luca Ciampi*, Fabrizio Falchi*, Claudio Gennaro* and Giuseppe Amato*
*Institute of Information Science and Technologies - National Research Council - Pisa, Italy

I. Introduction

As evidenced during the recent COVID-19 pandemic, there are scenarios in which ensuring compliance to a set of guidelines (such as wearing medical masks and keeping a certain physical distance among people) becomes crucial to secure a safe living environment. However, human supervision could not always guarantee this task, especially in crowded scenes. This abstract presents *CrowdVisor*, an embedded modular Computer Vision-based and AI-assisted system that can carry out several tasks to help monitor individual and collective human safety rules. We strive for a real-time but low-cost system, thus complying with the compute- and storage-limited resources availability typical of off-the-shelves embedded devices, where images are captured and processed directly onboard. Our solution consists of multiple modules relying on well-researched neural network components (such as [1]), each responsible for specific functionalities that the user can easily enable and configure. In particular, by exploiting one of these modules or combining some of them, our framework makes available many capabilities. They range from the ability to estimate the so-called social distance to the estimation of the number of people present in the monitored scene, as well as the possibility to localize and classify Personal Protective Equipment (PPE) worn by people (such as helmets and face masks). To validate our solution, we test all the functionalities that our framework makes available over two novel datasets that we collected and annotated on purpose. Experiments show that our system provides a valuable asset to monitor compliance with safety rules automatically.

II. Architectural Overview

The general purpose of our system is to be embeddable on low-cost devices and, above all, to be expandable to different features in demanding situations. To this end, we designed a framework able to orchestrate a set of internal and user-defined *plugins*, each dedicated to a single task. Specifying inputs and outputs makes it possible to create a dependency graph. Each sub-module represents a node, and each pair of matching input-output represents an edge. In this way, given the desired output, a *topological sort* is executed to minimize and linearize the sequential execution of computations.

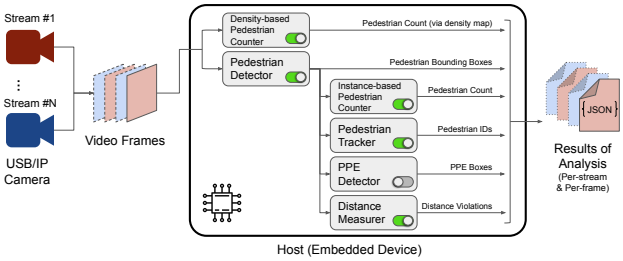

Fig. 1: Overview of our modular framework. Our system is *flexible* and *expandable*, as modules can be activated or deactivated depending on the user's needs, and novel functionalities can be introduced with additional custom modules.

An overview of our modular framework is depicted in Figure 1. Video frames are taken at regular intervals from one or more cameras and processed locally. Multiple video streams can be multiplexed and handled by a single system instance. Current modules include a) Pedestrian Detector, b) Density-based Pedestrian Counter, c) Instance-based Pedestrian Counter, d) Pedestrian Tracker, e) PPE Detector, and f) Interpersonal Distance Measurer; Figure 2 exemplifies the results of the analyses performed by each module. All the modules are toggleable; the Instance-based Pedestrian Counter, Pedestrian Tracker, Interpersonal Distance Measurer, and PPE Detector modules depend on the output of the Pedestrian Detector module and require it to be active. Results of the active modules are combined and provided in JSON format to be consumed by downstream services. Note that video frames are analyzed onboard and never stored; this enables privacy-aware solutions where captured images never leave the edge devices.

III. Experimental Evaluation

To validate our solution, we exploited two novel datasets that we collected and annotated on purpose. Specifically, we gathered *CrowdVisorPisa* a dataset of images captured by a smart camera located in a public square in the city of Pisa, Italy, representing a typical scenario for which it is crucial to check compliance with the safety rules. Moreover, we collected and annotated a second dataset (a part of which is gathered from a video game) comprising images containing pedestrians with and without PPE, such as helmets and face

This work was partially supported by the H2020 projects AI4EU (GA 825619) and AI4Media (GA 951911), and by the Tuscany POR FSE 2014-2020 project AI-MAP (CNR4C program, CUP B15J19001040004). We would like to thank Visual Engines S.r.l. - https://www.visualengines.com - for supporting this research and giving us access to their data sources collected for the CrowdVisor project.

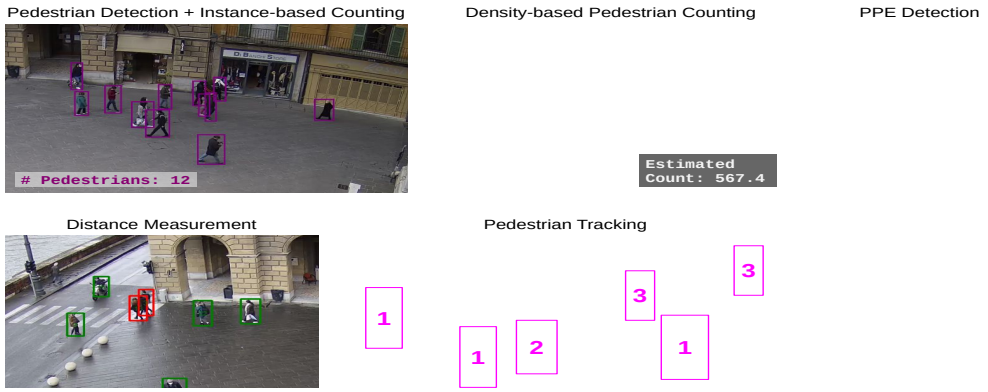

Fig. 2: Visualization of output examples of the modules currently available in our system. The outputs of each module are the following. **Pedestrian Detection & Instance-based Counting**: list of pedestrian bounding boxes and respective count. **Pedestrian Tracking**: numeric ID assigned to detected pedestrians persisting through frames. **Density-Based Pedestrian Counting**: estimated number of pedestrians (and, optionally, the density map). **Distance Measurement**: IDs of groups of pedestrians violating a predefined distance. **PPE Detection**: list of PPE bounding boxes detected per pedestrian.

DC	PPE	SysRAM	GpuRAM
7	7	2.36	0.55
3	7	2.44	0.86
7	3	2.35	2.10
3	3	2.44	2.20

TABLE I: System and GPU Memory Usage in GB. **DC** = whether the density counter module is active; **PPE** = whether the PPE detector module is active. The modular framework is assumed to always use the object detector, along with the enabled distance measure plug-in that consumes a fixed and negligible (less than 1 MB) amount of memory.

masks. Being our target a deployable monitoring system, we selected the NVIDIA Jetson TX2 embedded device as the hardware host, equipped with an external USB camera. At the time of writing, the cost of the device was less than USD 500. As detailed in Table I, memory usage is kept within 5 GB of both system and GPU RAM.

For lack of space, we report in Fig. 3 only the performance evaluation concerning the Counting by Instances functionality exploiting the *CrowdVisorPisa* dataset. For an in-depth experimental evaluation of all the modules, we refer the reader to [2]. As can be seen, we get a Mean Absolute Error (MAE), i.e., the mean of the sum of the absolute errors, close to 1 or 2, demonstrating that the module provides a reliable estimation of the number of pedestrians present in the monitored scene.

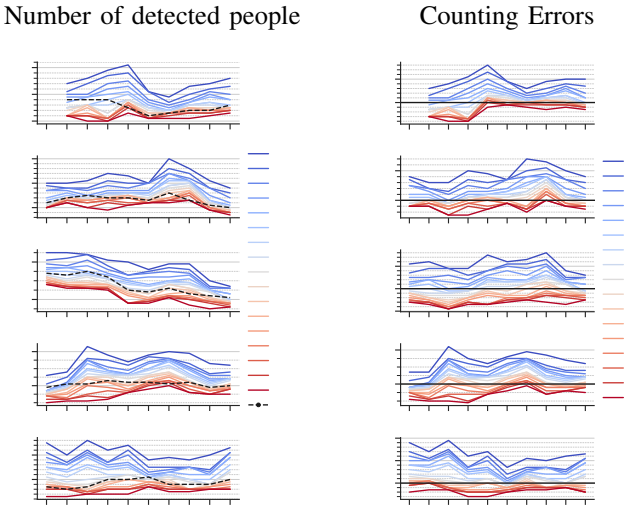

Fig. 3: Evaluation of *counting by instances* functionality of our framework, considering five test sequences of our *CrowdVisorPisa* dataset. In the first column, we report the number of people located by our detector, varying the detection thresholds. The black line (GT) indicates the actual number of pedestrians in the frame. The second column shows the counting errors and the best MAE obtained with a specific detection threshold.

IV. Conclusion

In this abstract, we presented a modular framework based on Computer Vision and AI technologies, deployed in a real use-case scenario on a low-cost off-the-shelf embedded platform and aimed at monitoring human activities in critical conditions. To test the effectiveness of our solution, we monitored a known place in Italy during the restrictions imposed from the COVID-19 pandemic, proving satisfactory accuracy in terms of detection, counting, and physical distance measurements.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, jun 2017.

[2] M. D. Benedetto, F. Carrara, L. Ciampi, F. Falchi, C. Gennaro, and G. Amato, "An embedded toolset for human activity monitoring in critical environments," *Expert Systems with Applications*, vol. 199, p. 117125, aug 2022.