Flexible Thermal Camera Solution for Smart City People Detection and Counting

Enrico Collini, Luciano Alessandro Ipsaro Palesi, Paolo Nesi(*), Gianni Pantaleo, William Zhao

Distributed Systems and Internet Technologies Lab, Dept. of Information Engineering, University of Florence, Italy, Https://www.disit.org, https://www.snap4city.org

Enrico.collini@unifi.it, paolo.nesi@unifi.it, Gianni.pantaleo@unifi.it, lucianoalessandro.ipsaropalesi@unifi.it, william.zhao@stud.unifi.it

Abstract. Tourism management assumes an important role in the context of Smart Cities. In this work, we used thermal cameras for the development of an Object Detection solution in pedestrian areas. The solution is capable to classify people, bikes, and strollers, and count people in Real-Time, in hot squares where relevant number of people pass by, by using telephoto and wide-angle thermal cameras. To this end, the solution exploited the FASTER-R-CNN and the YOLOv5 with a set of tuning approaches for improving the precision and the flexibility with respect to solutions at the state-of-the-art. Both top-down and bottom-up training adaptation approaches have been assessed by demonstrating that the bottom-up approach can provide the best results. The results overcome the state-of-the-art in terms of performance for relevant number of people in the scene (removing the limitation of the state-of-the-art solutions that were limited to provide good precision up to 10 people) and flexibility for different camera lens and resolutions. The resulting model is also capable to produce results in Real-Time on industrial PC of mid-level, and it has been enforced to work directly on thermal cameras. The proposed solution has been developed and validated in the context of Herit-Data EC project and using the Snap4City platform for the final collection of data results and publication of monitoring dashboards.

Keywords: smart city, tourism management, multiclass object detection, crowd people counting, tracking, thermal cameras, YOLO, Faster-R-CNN.

I. Introduction

Tourism is, without doubt, a vital component for many cities, however, managing it is a difficult task and many problems such as overcrowded situations can get in the way and lead to reduce the appreciation of the touristic site experience. Thanks to the development of modern cities there are plenty of possible solutions in the context of Smart Cities based on the use of Big Data and IoT Devices (Internet of Things) to acquire useful information on the city conditions/context, and tourists' behaviour in the city. People detection and counting are among the

most interesting features for monitoring tourists and security in hot destinations and sites of interest around the world, as in malls, stadiums, theatres, etc., to provide support at decision makers. The people counting can be strongly useful for detecting critical conditions (early warning), for security and cleaning activities. The detection and counting of people is a value for the cultural sites management, and the adopted solutions have to respect the GDPR (General Data Protection Regulation) [1] and privacy in general.

Italy is one of the countries in which a large number of hot destinations are present, such as: Florence, Venice, Rome, Milan, etc. In some cases, the municipalities have adopted limitations on the number of people presences in the major city squares and areas, in order to maintain acceptable, the quality of experience and services for all the categories of city users, and thus also for the tourists. The limitations are typically imposed by tripod gates, tickets, and other physical/invasive solutions which reduce the free people flows and may provoke difficulties for evacuations. In order to detect and count the number of people in specific areas of interest there are nonintrusive solutions based on different technologies. For example, IoT sensors-based solutions as PaxCounters (Wi-Fi sniffers [2], Laser counting, infrared counters, etc. [3]) are widely used and for some scenarios, like the major squares, they could be of difficult usage since they have limited range capabilities. In controlled conditions, such as fairs, festivals, museums, etc., the usage of wearable tags, which can be assigned to a significant number of the attendees, can be a viable solution to understand how they move, how much they stay in each room, etc. On the other hands, these solutions are invasive and quite expensive.

Alternative solutions are based on video cameras which allow to detect, classify, count and track people [4] by Computer Vision and artificial intelligence, AI. This is a field in which AI analyze visual data and provide support to make decisions providing hints on scene understanding of the environment and the situation [5]. Governments and companies are investing in security networks hundreds of millions more surveillance cameras are watching the world according to the report from industry researcher IHS Markit [6]. Most of these solutions have strong applications in the context of security and surveillance in which the GDPR aspects are relaxed. On the contrary, in the context of on-street people counting, RGB

cameras are not appreciated by the municipality for their difficulties in passing GDPR compliance assessment.

An emerging compromise consists in the usage of thermal cameras for the detection and counting people. Thermal cameras are much more acceptable for on-road counting since they do not allow to perform face recognition at resolutions in which the RGB cameras can do it. Thermal cameras are typically more expensive than RGB cameras, while for the people counting, they have the advantage to be (i) non-invasive privacy compliant, and (ii) capable to work well in lack of illumination. For thermal cameras, there are object detection algorithms that could be used to elaborate the video stream images to detect the presence of particular classes of objects, and their position in the image [7], [8], [9]. These algorithms can be tuned to detect particular classes of objects, while they present some limitations regarding the capability of counting as discussed in next subsection.

For these reasons, we focused on thermal camera usage for people detection and counting.

I.A *Related works*

In the context of people flow analysis for tourism management, the Multiclass Object Detection and People Counting are fundamental tasks to provide support to decision makers. In most cases, the task is solved by using Computer Vision techniques by using colour images. They are problematic for the privacy, and the procedure for their GDPR approval is not trivial. The related work discussed on this section is grounded on thermal camera and it is summarized in Table I.

The Multiclass Object Detection aims to determine the bounding boxes of the elements in an image and their classification. For this problem, Computer Vision solutions at the state-of-the-art are primarily based on RGB images [10], [11], [12], [13], or on thermal images, and in some cases using both colour and thermal data [14].

Authors	Task	Image Type	Dataset	Model	Results	Range
Jia et al., 2021 [7]	Pedestrian Detection	RGB and Thermal	LLVIP	YOLOv5	mAP_0.5: 0.9650	< 10 people per image
Krišto et al., 2020 [8]	Person Detection	Thermal	UNIRIT ID	YOLOv3	mAP_0.5: 0,9793	< 10 people per image
Kowalski et al., 2021 [9]	Boat and People Object Detection	Thermal	Elblag and Bug rivers in Poland	Faster R-CNN with ResNet101	DR_0.7: 0.83	1 or 2 people per image
Goel et al., 2021 [15]	Pedestrian Detection	Thermal	Thermal OSU Pedestrian dataset	Faster R-CNN	Accuracy: 0.9238 Precision: 0.8932 Racall: 0.9124	< 10 people per image
Dai et al., 2021 [16]	Multiclass Object Detection	Thermal	CTIR, KAIST	TIRNet	CTIR: mAP_0.5: 0.7485 KAIST: mAP_0.5: 0.5993	CTIR: 31035 people in 11938 images KAIST: 86.2K people in 95K images
Kera et al., 2022 [17]	Object Detection	Thermal	FLIR-ADAS	EfficientNet + BiFPN	FLIR-ADAS mAP_0.5: 0.773	FLIR-ADAS: 28151 people in 10288 images
Munir et al., 2021 [18]	Multiclass Object Detection	RGB and Thermal	FLIR-ADAS, KAIST	SSTN	FLIR-ADAS: mAP_0.5: 0.7757 KAIST: mAP_0.5: 0.7322	FLIR-ADAS: 28151 people in 10288 images KAIST: 86.2K people in 95K images
Li et al., 2021 [19]	Multiclass Object Detection	Thermal	FLIR-ADAS, KAIST	YOLOv5	FLIR-ADAS: mAP_0.5: 0.835 KAIST: mAP_0.5: 0.983	FLIR-ADAS: 28151 people in 10288 images KAIST: 86.2K people in 95K images

Table I Related works comparison for Object Detection via ***thermal*** cameras.

In more details, Jia, Zhu et al., presented the LLVIP dataset in [7] which contains street images in RGB and thermal formats that can be used for different purposes. In [7], the authors reported a section relevant to the problem of Pedestrian

Detection. They fine-tuned the pre-trained YOLOv5 model [20] on COCO dataset [21] using the thermal data from LLVIP. The solution achieved a mean Average Precision (mAP) at the Intersection over Union (IoU) threshold of 0.5 (mAP_0.5) of 0.965 on the thermal images compared to the 0.908 of the corresponding RGB images. For person detection and surveillance, thermal cameras have been proposed by Krišto et al. [8], taking into account also the effect of weather conditions. In that context, a custom dataset has been created with video acquired during the winter in different weather conditions (clear weather, rain, fog), during the night, and for different distances from the camera (ranging from 30 m to 215 m, using YOLOv3). This solution achieved a mAP_0.5 of 0.87. In [22], a YOLOv3 model was trained to detect both Human and NonHuman objects (e.g., dogs) in thermal images. In this case, they achieved a mAP_0.5 score of 0.9798, confirming the possibility of using the solution for the automatic monitoring of protected objects and areas. Kowalsky et at., in [9], compared different state-of-the-art Object Detection models to detect people and inflatable boats from a distance of 50-200 meters using thermal images. The best model in terms of performance was Faster R-CNN (Region based Convolutional Neural Network) with ResNet101. On the other hand, in terms of processing time, YOLOv3 was significantly faster and achieved a Detection Rate (DR) with an IoU threshold set at 0.7 (DR_0.7) of 65%. Goel et al., focused on the problem of pedestrian detection [15]. The dataset used was the Thermal OSU Pedestrian Dataset from OTCBVS Benchmark Database [23]. The best results achieved were obtained by using a Faster R-CNN, demonstrating the validity of the solution in multiple illumination conditions depending on the weather (Dense Cloudy, Light Rain, Partly Cloudy, Haze, Sunny). Multiclass Object Detection has an important role in advanced driver assistance systems (ADAS) and autonomous driving applications. In [16], Dai et al., proposed TIRNet a deep neural network architecture based on convolutional layers to detect cars, pedestrians, cyclists, buses and trucks. Overall, the mAP_0.5 over all the classes for the proposed dataset achieved 0.7485 and, considering only the pedestrian class, mAP_0.5 = 0.8047. On the KAIST dataset [24] the TIRNet achieved a mAP_0.5 = 0.5993. Other works on Multiclass Object Detection in autonomous driving are Kera et al., [17] and Munir et al., [18], in which the authors used a self-supervised technique to learn enhanced feature representation using unlabelled data and a multi-scale encoder-decoder

transformer network that used these enhanced features embedding to develop a robust thermal image object detector. In this latter case, the proposed approach achieved over all the classes on the KAIST dataset a mAP_0.5 = 0.7322 and on the FLIR-ADAS dataset [25] mAP_0.5 = 0.7757. Kera et al., [17], proposed an EfficientNet solution with a weighted bidirectional feature pyramid network, achieving a mAP_0.5 = 0.773 on the FLIR-ADAS. Li et al., [19], based their solution on YOLOv5, thus improving the state-of-the-art performance for the problem of Object Detection in the two datasets FLIR-ADAS and KAIST, achieving a mAP_0.5 of 0.835 and 0.983, respectively.

When the goal is just counting people in the scene there are many approaches at the state-of-the-art ([26], [27], [28], [29]) but primarily detection-based approaches are widely used. These systems first detect people on the images and then count their numbers as in [30]. These systems are in most cases based on classifiers trained on the whole body or on a part of the body (for example the head, which resulted to be in most cases less precise than body detection). An example, is based on YOLOv3 classifier as presented in [31], obtaining a classification accuracy of 96.1% on the INRIA dataset [32] (uncrowded urban contexts) and 82.1% on the ShanghaiTech dataset [33] (urban contexts with some crowded scenes). Detection-based approaches can be used for people counting and are also widely used at the state-of-the-art for tracking systems as in [34], [35], [36], [37], [38].

I.B *Article aims and contributions*

In this paper, we focused on the problems of people detection and counting in cultural heritage locations that are crowded with tourists such as: Florence in Italy, Valencia in Spain, Pont du Gard in France, Dubrovnik in Croatia, etc. They are locations in which specific city squares (located in strictly pedestrian areas) attract high number of tourists almost at any time of the day. Therefore, the proposed solution addressed the problems of people detection (classification) and counting, comparing and overcome the state-of-the-art solutions on three main goals and providing higher:

- precision for detection and counting in crowded conditions, detection to identify/count: people, bikes/motorbikes, strollers/carts;

- flexibility in terms of counting range in which the relevant precision can be obtained, over the 10 people which is a limitation of the state-of-the-art solutions as highlighted in the paper;

- flexibility in terms of counting precision by using different kinds of lenses for thermal cameras: from telephoto to wide-angle.

The proposed solution exploited the YOLOv5 [20] and LLVIP [7] with a set of tuning approaches for improving the precision and the flexibility of the previous solutions at the state-of-the-art. To this end, we explored both top-down and bottom-up training adaptation approaches, demonstrating that the bottom-up approach can provide the best results according to the above-mentioned objectives of performance and flexibility.

In addition, the solution has been implemented to obtain Real-Time execution on (i) mid-level industrial PC capable to perform multiple Python stream processing (which allows to use the solution connected to any RTSP stream of thermal cameras), (ii) board of AXIS thermal cameras. The results from people detection processes can be used to track the number of objects of the specific classes of interest and can be integrated in monitoring dashboards that can be a useful tool for decision-makers. The proposed solution has been tested and validated for detecting people (pedestrians, bikes/motorbikes, strollers/carts) in strictly pedestrian areas which is the typical case in cultural cities in Europe. And particular in Piazza Della Signoria in Florence, Italy, the city hall square of Florence, Italy, which is a city that attracts about 15 million of tourists per year. The solutions have been developed and validated in the context of the Herit-Data Interreg European Commission project [39] which aims to identify innovative solutions to monitor and manage the impact of tourism on cultural and natural heritage sites, with the support of new Big Data technologies. The solution has been implemented by exploiting the Snap4City framework and platform which is 100% open-source solution [https://www.snap4city.org], [40], [41].

I.C *Article Structure*

The paper is structured as follows. Section II describes the problem and data for the operating conditions, and thus the data which can be obtained by thermal cameras of different kinds. On these bases, an assessment of the state-of-the-art solutions (based on YOLO and Faster-R-CNN) has been performed to put in evidence the limitations identified which make them unsuitable to the goals identified in counting and classification. In Section III, the solution identified to enforce flexibility and capability of high precision counting in presence of high number of people is presented. Section IV presents the usage of a bottom-up training adaptation approach which additionally improved the precision and flexibility of the early YOLO with LLVIP training. In Section V, the deployment architecture, which can be used to adopt the solution for Real-Time detection and counting of people, is described. Conclusions are drawn in Section VI.

II. Problem and data definition

As mentioned in the introduction, the main goal of the presented research was to detect and count people in real-time for tourism context. And, in particular, to detect and manage situations in which we may have up to 60-70 people in a single image. The state-of-the-art of thermal camera datasets and solutions have not yet addressed such a condition. For example, considering the most diffuse dataset of thermal images: LLVIP dataset [7] is limited to max 10 people, KAIST dataset provides a mean number of objects of 0.90 per image [41], CTIR dataset provides a mean number of objects of 2.59 per image [16], and FLIR-ADAS dataset a mean number of objects per image of 2.73 [25].

Critical tourism conditions may present much higher numbers and, thus the counting solutions have to work with relevant precision in the range from 0 to 70. For example, in Piazza della Signoria square in Florence, Italy, we manually counted hundreds of people in total and 70 under the view of each single camera are a symptom of a crowded condition. While in the views reported in Figure 1, they are the most relevant portions of the square, we may have reasonably up to 70 people. Over that number of people, a critical condition may be warned. The

squares need to be physically monitored for security reasons and for cleaning and assistance. In the specific case, the inception of large amount of people in the square may unexpectedly arrive from the two main directions observed (coming from Ponte Vecchio and Uffizi, respectively, for example). In Figure 2, the images of CAM51 and CAM52 labelled in Figure 1 are reported.

Fig. 1 Cameras' views of Piazza Della Signoria, Florence, Italy.


(a)


(b)

Fig. 2 Views of CAM51 (a) and CAM52 (b) Piazza Della Signoria, Florence, Italy.

The main goals have been to produce a solution which can: (i) perform people detection and counting with high precision in the range from 0 to 70, (ii) be applied to different thermal cameras without retraining (which would increase the cost too much), (iii) be adopted in real time on RTSP stream as well as directly on board of the camera (to produce detections and counting, providing the bounding box via MQTT messages).

According to the above requirements, the first experiment has been to assess the best solution from the state-of-the-art on CAM51 and CAM52 scenarios and data. To this end, we taken into account

(i) YOLOv5 based solution pre-trained with COCO dataset and fine-tuned with LLVIP dataset has been implemented as in [7]. The implementation has been based on the Ultralytics code [20] using the architecture YOLOv5s to compromise detection capability speed of execution and model weight in view of being installed on edge devices.

(ii) Faster R-CNN pre-trained with ImageNet dataset [42] and fine-tuned with LLVIP dataset performed by us and presented in this paper (named FRCNN-LLVIP-Model). The Faster-R-CNN (in the following FRCNN) has been realized using the Detectron2 framework [43] specifically using the architecture X101-FPN.

Therefore, the resulting models (named as YOLO-LLVIP-Model and FRCNN-LLVIP-Model, respectively) have been assessed with respect to the original LLVIP validation dataset and also with respect to test datasets created from the videos acquired from CAM51 and CAM52. The results are reported in Table II. Please note that all the datasets adopted in this first case presented maximum 10 people (since the LLVIP provides images with maximum 20 people a selection has been performed for the comparison) (a similar limitation for registered on COCO and ImageNet). More details on the used standard metrics for the comparison are reported in Section IV.A. From this early analysis, it resulted evident the reduction of performance of YOLO-LLVIP-Model and FRCNN-LLVIP-Model in terms of mAP_0.5, and precision, passing from LLVIP

validation with respect to the two real cases of CAM51 and CAM52 (see Figure 1).

Trained Model and datasets		**precison**	**recall**	**mAP_0.5**	**mAP_0.5: 0.95**
YOLO-LLVIP	LLVIP Validation <=10	0.953	0.930	0.959	**0.698**
FRCNN--LLVIP	LLVIP Validation <=10	**0.971**	**0.944**	**0.964**	0.606
YOLO-LLVIP	CAM51 Test <=10	0.931	0.870	0.908	0.471
	CAM52 Test <=10	0.944	0.515	0.737	0.383
FRCNN--LLVIP	CAM51 Test <=10	0.841	0.751	0.775	0.338
	CAM52 Test <=10	0.858	0.626	0.701	0.323

Table II People detection results on YOLO-LLVIP-Model, FRCNN-LLVIP-Model

All solutions at the state-of-the-art are providing results for less than 10 people [7], [8], [9], [15], and this is also evident from the data sets as described at the beginning of Section III. Therefore, a second experiment was conducted to assess the quality of the above presented solutions (pretrained and fine-tuned with the same LLVIP) as a function of the people which are present in the image using CAM51 and CAM52 test data sets. The results for the second experiment are reported in Table III. Thus, we observed that, when the number of people is greater than 10, growing and relevant errors are experienced. Both YOLO-LLVIP-Model and FRCNN-LLVIP-Model provide a worst mAP_0.5 when the number of people per image is greater than 50 wrt to <=10 case. Both models seem to work better for CAM51, rather than for CAM52 test set (see their description in the following).

YOLO-LLVIP-Model **Test dataset**		**precison**	**recall**	**mAP_0.5**	**mAP_0.5: 0.95**
CAM51	<=10	0.931	0.870	0.908	0.471
	>10&<=25	0.933	0.731	0.842	0.439
	>25&<=50	0.887	0.492	0.706	0.387
	>50&<=75	0.836	0.348	0.602	0.299
	> 75&<=97	0.820	0.274	0.544	0.230
CAM52	<=10	0.944	0.515	0.737	0.383
	>10&<=25	0.771	0.255	0.508	0.264

	>25&<=50	0.501	0.099	0.291	0.114
	>50&<=75	0.388	0.061	0.201	0.075
	> 75&<=79	0.296	0.031	0.158	0.048
FRCNN-LLVIP-Model **Test dataset**		**precison**	**recall**	**mAP_0.5**	**mAP_0.5: 0.95**
CAM51	<=10	0.841	0.751	0.775	0.338
	>10&<=25	0.905	0.789	0.821	0.389
	>25&<=50	0.884	0.642	0.696	0.308
	>50&<=75	0.854	0.534	0.593	0.236
	> 75&<=97	0.882	0.504	0.601	0.239
CAM52	<=10	0.858	0.626	0.701	0.323
	>10&<=25	0.698	0.414	0.429	0.165
	>25&<=50	0.462	0.192	0.182	0.053
	>50&<=75	0.374	0.114	0.116	0.031
	> 75&<=79	0.285	0.068	0.614	0.014

Table III People detection results on YOLO-LLVIP-Model, FRCNN-LLVIP-Model

The above-presented experiments demonstrated the limited capability of YOLO-LLVIP-Model and FRCNN-LLVIP-Model in addressing the problems of people detection (classification) and counting with the aim of providing:

- high precision for detection and counting in crowded conditions;
- high flexibility in terms of counting range in which a reasonable precision can be obtained;
- high flexibility in terms of counting precision using different thermal cameras from telephoto to wide-angle without retraining the model.

For this reason, the approach of transfer learning was not viable to solve the problem and we decided to perform an additional fine tuning and training. Thus, to create a new data set for training and validation to overcome the problems detected and verified as described above.

II.A – Flexible people detection dataset of thermal images

In order to generate data for training, a number of video sequences have been taken and manually classified. For the labelling, the Yolo_Label tool [44] has been used. It creates a .txt file for each image containing for each object in the image: <object-class><x_center> <y_center> <width> <height>. The <object-class> for the object detection/ classification has been assigned as follows: 0-green for people, 1-blue for bikes/motorbikes and 2-red for strollers/carts (please note that the sum of these classes is the number of counted people in the image). See the example of Figure 3 for CAM51, in which 71 people, 1 bike and 2 strollers have been labeled. Regarding CAM52, the frames were rectified by removing the wide-angle lens distortion. Subsequently, the images were labelled.

Fig 3. Example of Object Detection, CAM51.

In order to build the thermal image training data sets, and to evaluate the machine learning models and solutions as a function of the number of detected/classified people, the images have been labeled according to the number of people and their classification. Typically, the images of the data set have been also grouped according to the number of people included: <=10, >10 and <=25, >25 and <=50, >50 and <=75, >75. The maximum number of people in the scene is for CAM51, 97 people and 79 for CAM52. Table IV shows the number of images and the number of people, bikes and strollers within the different datasets. All the images

are positive examples, in the sense that all images contain at least one object. In addition, to train the multi-category detection "person", "bike" and "stroller" a minimum of 7% of images containing each category were selected. Regarding the test datasets the images from the two thermal cameras have been labeled considering only the people in the scenes, for the cases where on the bike there was a person it was considered and also on the strollers.

Thermal Dataset		**# images**	**# objects with class specification**	**# tot labels**
LLVIP	training	12025	people 33648	33648
LLVIP	validation	3463	people 7931	7931
CAM51	training	178	people 5210, strollers 78, bikes 115	5403
CAM51	validation	44	people 1175, strollers 13, bikes 35	1223
CAM52	training	175	people 4472, strollers 59, bikes 39	4750
CAM52	validation	44	people 1357, strollers 13, bikes 20	1390
CAM51	test <=10	25	People, strollers, bikes	112
	test >10&<=25	25	People, strollers, bikes	474
	test >25&<=50	25	People, strollers, bikes	957
	test >50&<=75	25	People, strollers, bikes	1649
	test > 75 &<=97	25	People, strollers, bikes	2057
CAM52	test <=10	25	People, strollers, bikes	130
	test >10&<=25	25	People, strollers, bikes	329
	test >25&<=50	25	People, strollers, bikes	780
	test >50&<=75	25	People, strollers, bikes	1348
	test > 75 &<=79	25	People, strollers, bikes	1911

Table IV Dataset Description

II.B Metrics

In order to evaluate the results of the trained models, the following metrics have been used. IoU metric in Object Detection evaluates the degree of overlap between the ground truth (gt) and prediction (pd):

$$IoU = \frac{area(gt \cap pd)}{area(gt \cup pd)}$$

Fixed α IoU threshold and defining

- TP = True Positive
- FP = False Positive

- FN = False Negative
- TN= True Negative

Other metrics are:

- $Precision = \frac{TP}{TP+FF}$
- $Recall = \frac{TP}{TP+FN}$
- AP is the area under precision-recall curve $(p(r))$ evaluated by using $^{\alpha}$ IoU threshold.

$$AP_{\alpha} = \int_0^1 p(r)dr$$

- mAP, mean Average Precision is the average of AP values over all classes.

$$mAP_{\alpha} = \frac{1}{|n\_classes|} \sum_{i=1}^{n\_classes} AP_{\alpha\ i}$$

III. Enforcing flexibility in the model

This section reports the process to realize a model addressing the problems of people detection (classification) and counting providing requirements of:

R1) high precision for detection to identify people, bikes/motorbikes, strollers/carts and counting in crowded conditions;

R2) high flexibility in terms of counting range in which a relevant precision can be obtained;

R3) high flexibility in terms of counting precision using different thermal cameras from telephoto to wide-angle without retraining the model.

R4) real time computation capabilities on stream and on board of TV Cameras.

Based on the results of the related works in the state-of-the-art the single-stage object detection YOLO architecture has been compared to the multi-stage object detection Faster-R-CNN with FPN architecture both in detection capability and execution speed.

III.A. Case(i) for multiclass object detection

In this Case (i), the selected architectures YOLO pretrained on COCO dataset and FRCNN pretrained on ImageNet dataset [42], have been fine-tuned for the problem of multiclass object detection of people, bikes and stroller using the

training datasets from CAM51 and CAM52 with the aim of choosing the best solution for the requirements R1, R2, R3.

Starting from the YOLO architecture pretrained with COCO [7] has been fine-tuned with the training sets of CAM51 and CAM52 (see Table IV), respectively, thus obtaining the so called: YOLO-CAM51-Model and YOLO-CAM52-Model. The training processes used an early stopping with patience set to 100 on the mAP_0.5 of the validation set.

Also, the FASTER-R-CNN architecture pretrained with ImageNet [42] has been fine-tuned with the training sets of CAM51 and CAM52 (see Table IV), respectively, thus obtaining the so called: FRCNN-CAM51-Model and FRCNN-CAM52-Model. The training processes used an early stopping with patience set to 500 on the mAP_0.5 of the validation set.

The results in terms of precision, recall, mAP_0.5, mAP_0.5:0.95 are reported in Table V. The results show that the models based on YOLO achieved better results compared to those based on FRCNN and overall, the validation dataset CAM51 achieved better performance compared to the CAM52 in both the architectures tested. Please note that the CAM51 and CAM52 validation data sets include images for multiclass detection having a number of people in most cases higher than 10.

model vs validation set	**precision**	**recall**	**mAP_0.5**	**mAP_0.5:0.95**
YOLO-CAM51-Model vs CAM51 Validation set	**0.988**	**0.960**	**0.975**	**0.605**
YOLO-CAM52-Model vs CAM52 validation set	**0.925**	**0.879**	**0.865**	**0.398**
FRCNN-CAM51- Model vs CAM51 validation set	0,7872	0,7769	0,7438	0,3367
FRCNN-CAM52-Model vs CAM52 validation set	0,8113	0,8078	0.7794	0,3650

Table V Multiclass object detection results in validation – YOLO vs FRCNN models. In bold the best results for each specific case.

To better understand the results of the assessed models the confusion matrixes for the interested classes persons, bikes and strollers (and background) can be analyzed. In Figures 4 and 5, the confusion matrixes for the validations of CAM51/CAM52, for YOLO and FRCNN are reported, respectively. In both

validation cases, especially for the True Positives on the class Person, the YOLO based model architectures outperformed the FRCNN based models.

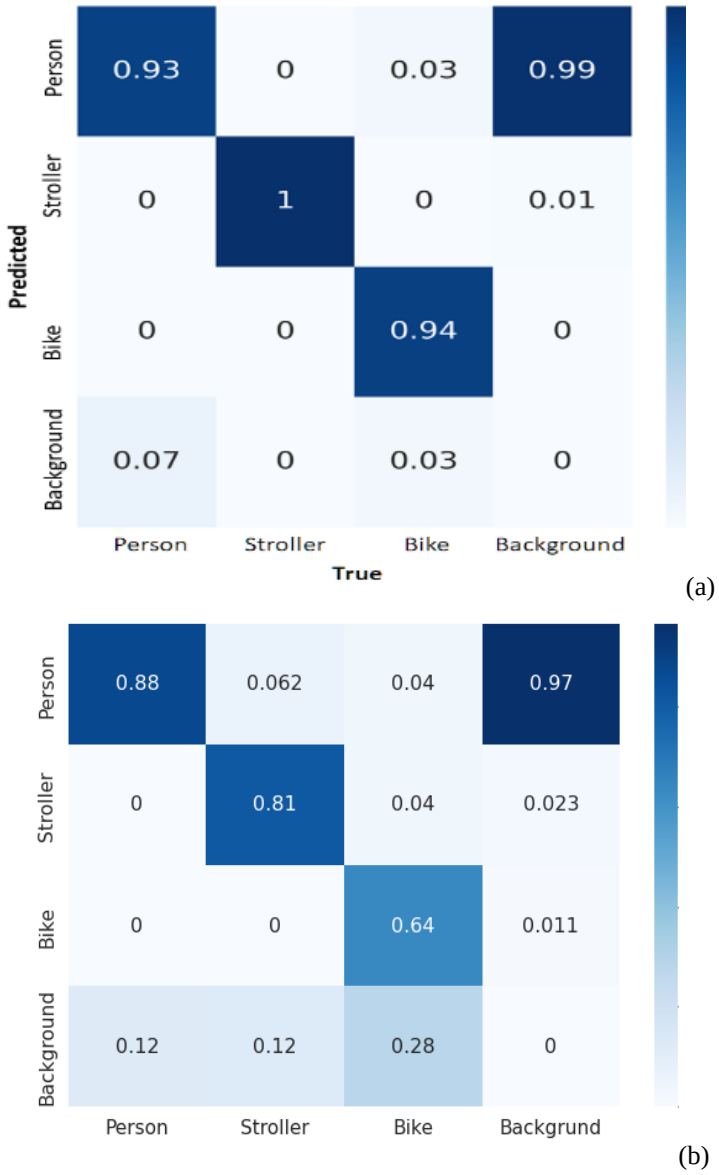

Fig. 4. Confusion Matrixes on the validation dataset of CAM51: (a) YOLO-CAM51-MODEL, (b) FRCNN-CAM51-MODEL

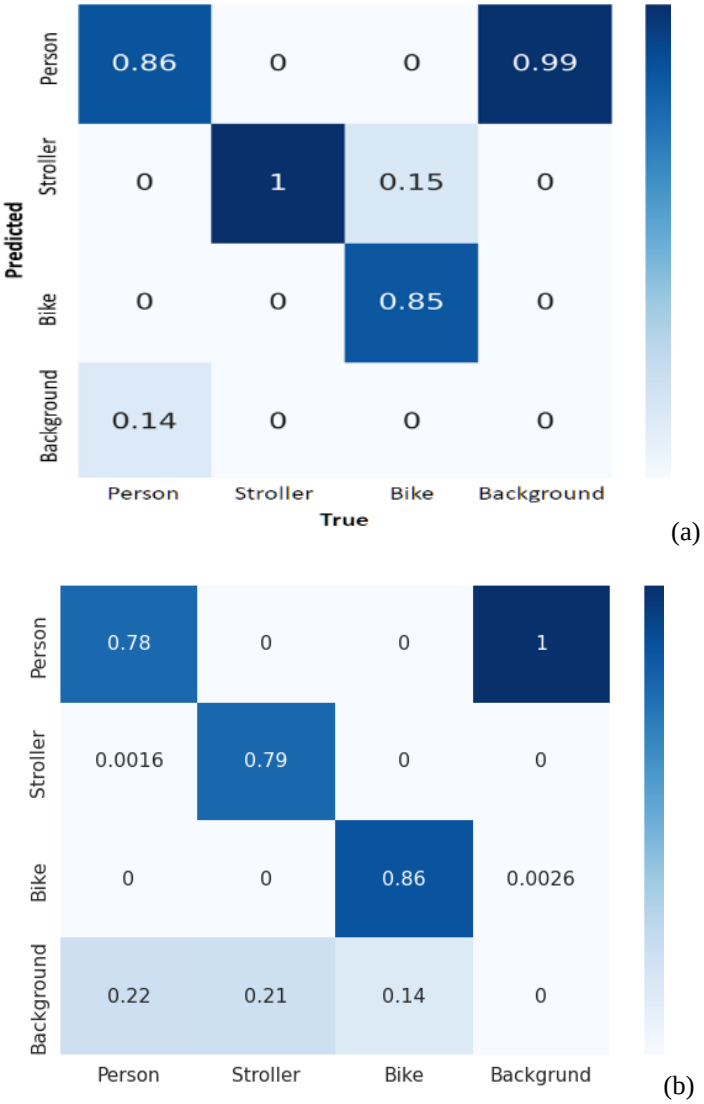

Fig 5. confusion Matrixes on the validation dataset of CAM52: (a) YOLO-CAM52-MODEL, (b) FRCNN-CAM52-MODEL

III.B. Case (i) for MonoClass object detection

One of the key requirements of which this work is focused, R1, was to understand the performance of the developed models varying the number of people in the scenes. For this purpose, the models YOLO-CAM51-model, YOLO-CAM52-model, FRCNN-CAM51-Model, FRCNN-CAM52-Model have been compared to assess the precision, as a function of a number of people detected with respect to the test sets reported in Table IV for CAM51/52.

The results are reported in Table VI for the models fine-tuned with the CAM51 training set, and in Table VII, for the models developed with the CAM52training set. As a result, the 4 models fine-tuned for the specific cameras achieve better results compared to those obtained by the YOLO-LLVIP-Model as reported in

Table III, as a function of the number of people in the scene. Please note that, the LLVIP dataset (used in models of Table III) contains images with a few people (<=10) and in this category LLVIP-Model achieved for the CAM51 test <=10 dataset an mAP_0.5 of 0.908 (see Table III) compared wrt the 0.970 of YOLO-CAM51-Model and 0.9232 for the FRCNN-CAM51-Model. Regarding the CAM52 test set <=10 the YOLO-LLVIP-Model achieved a mAP_0.5 of 0.737 compared to the 0.975 of FRCNN-CAM52-Model and 0.962 of YOLO-CAM52-Model.

When considering scenes with more than 10 people (also in Tables VI and VII) the performances the new fine-tuned models outperform the YOLO-LLVIP-Model of Table III. Therefore, the relevant level of flexibility has been enforced into the fine-tuned models with respect to the formed YOLO-LLVIP-Model and FRCNN-LLVIP-Model, respectively. In more details, according to Table VI, the most suitable architecture for flexible detection capabilities in terms of counting range for the CAM51 test sets is the YOLO-CAM51-Model. On the other hand, for CAM52 test set none of the model identified is capable to overcome the other in all cases. YOLO-CAM52-Model resulted to be better on 3 of 5 ranges of people counts, and FRCNN-CAM52-Model. As a final consideration, the YOLO-CAM51-Model resulted to be the best compromise resulting the best model on 7 over 10 cases of both CAM51 and CAM52 test sets.
The mean value of mAP_0.5 over all cases for YOLO-CAM51-Model resulted to be 0,9271, while for FRCNN-CAM51-Model we recorded 0,8882.

	Test dataset		**precision**	**recall**	**mAP_0.5**	**mAP_0.5: 0.95**
YOLO-CAM51-MODEL	CAM51	<=10	0.972	0.938	**0.970**	0.891
		>10&<=25	0.969	0.932	**0.964**	0.881
		>25&<=50	0.961	0.934	**0.963**	0.863
		>50&<=75	0.966	0.885	**0.939**	0.832
		> 75&<=97	0.971	0.861	**0.927**	0.819
	CAM52	<=10	0.984	0.923	0.961	0.682
		>10&<=25	0.946	0.906	0.950	0.665
		>25&<=50	0.915	0.847	0.909	0.596
		>50&<=75	0.868	0.868	**0.890**	0.565
		> 75&<=79	0.858	0.706	**0.798**	0.462
		Mean			**0,9271**	
FRCNN-	C	<=10	0,933	0,883	0,923	0,619

CAM51-MODEL	AM51	>10&<=25	0,940	0,922	0,958	0,628
		>25&<=50	0,929	0,892	0,914	0,576
		>50&<=75	0,898	0,780	0,861	0,484
		> 75&<=97	0,904	0,772	0,814	0,430
	CAM52	<=10	0,968	0,961	0,967	0,574
		>10&<=25	0,966	0,943	0,952	0,558
		>25&<=50	0,914	0,876	0,896	0,509
		>50&<=75	0,909	0,835	0,881	0,502
		> 75&<=79	0,885	0,712	0,716	0,377
		mean			0,8882	

Table VI People detection results for YOLO-CAM51-MODEL and FRCNN-CAM51- Model

	Test dataset		**precision**	**recall**	**mAP_0.5**	**mAP_0.5: 0.95**
YOLO-CAM52-MODEL	CAM51	<=10	0.988	0.878	0.931	0.667
		>10&<=25	0.966	0.831	0.907	0.664
		>25&<=50	0.960	0.703	0.840	0.595
		>50&<=75	0.957	0.610	0.789	0.517
		> 75&<=97	0.951	0.579	0.770	0.466
	CAM52	<=10	0.976	0.938	0.962	0.853
		>10&<=25	0.977	0.924	0.958	0.821
		>25&<=50	0.975	0.841	**0.914**	0.781
		>50&<=75	0.976	0.770	0.879	0.744
		> 75&<=79	0.963	0.601	0.786	0.656
		mean			0,8736	
FRCNN-CAM52-MODEL	CAM51	<=10	0,917	0,893	0,920	0,536
		>10&<=25	0,880	0,874	0,944	0,627
		>25&<=50	0,871	0,778	0,817	0,403
		>50&<=75	0,834	0,644	0,645	0,272
		> 75&<=97	0,703	0,489	0,533	0,183
	CAM52	<=10	0,984	0,977	**0,975**	0,500
		>10&<=25	0,935	0,924	**0,969**	0,536
		>25&<=50	0,881	0,860	0,879	0,425
		>50&<=75	0,867	0,810	0,819	0,384
		> 75&<=79	0,884	0,653	0,688	0,317
		mean			0,8189	

Table VII YOLO-CAM52MODEL and FRCNN-CAM52- Model

Moreover, according to R4 we performed an assessment on the basis of the execution time (see Table VIII). The CPU computations have been performed on 8 core XEON at 2.3 GHz, while the deep learning solution have been executed on GPU as NVIDIA Quadro GV100 with 32GByte Ram, which has 5120 CUDA

Cores, FP64 perf as 7.4 TFLOPS. The YOLO-CAM51-Model is capable of real-time detections up to 112 frames per second using the GPU in the worst case evaluated (>75&<97 detection test set) compared to the 8 frames of the FRCNN-CAM51-MODEL.

Models assessment in execution		**CAM51: <= 10**		**CAM51: >75 & <= 97**	
		tot 25 execution (s)	**Mean 1 frame (s)**	**tot 25 execution (s)**	**Mean 1 frame (s)**
FRCNN-CAM51-MODEL	CPU	41,5088	1,6603	42,9870	1,7194
	GPU	2,8147	0,1125	2,8459	0,1138
YOLO-CAM51-MODEL	CPU	0,9264	0,0371	0,9676	0,0387
	GPU	0,2084	0,0083	0,2225	0,0089

Table VIII Performance analysis

According to the above reported requirements R1-R4 the most suitable architecture resulted to be YOLO with its derived models for fine tuning. For this reason, they have been further developed by combining the training sets available and tested with also a bottom-up layer wise domain adaptation as reported in the following section.

III.C. Case (ii) for MonoClass object detection

With the aim of adding mode flexibility to the model, according to the above results, two new approaches have been produced: Case (ii)a, and Case (ii)b.
The Case (ii)a has been obtained by starting from YOLO-Model and performing a fine-tuning using both CAM51-training set and CAM52-training set (see Table IV). The results are reported in Table IX in which the produced YOLO-CAM51-52-Model has been assessed against the test datasets of both cameras. According to the mAP the combination of training sets did not improve the results obtained for CAM51, while it improved those resulted for CAM52. Especially the mAP_05 improved for all the ranges except the range of (50-75] people with a mAP_0.5 of 0.878 compared with the CAM52 model of 0.879. The mean value of mAP_05 over all ranges resulted to be 0.91.

Test dataset		**precision**	**recall**	**mAP_0.5**	**mAP_0.5:0.95**
CAM51	<=10	0.916	0.786	0.876	0.702
	>10&<=25	0.934	0.907	0.946	0.745
	>25&<=50	0.945	0.875	0.929	0.719
	>50&<=75	0.949	0.816	0.897	0.674

	> 75&<=97	0.937	0.824	0.898	0.660
CAM52	<=10	0.984	0.977	0.985	0.696
	>10&<=25	0.975	0.954	0.971	0.653
	>25&<=50	0.909	0.874	0.915	0.608
	>50&<=75	0.885	0.843	0.878	0.560
	> 75&<=79	0.891	0.692	0.805	0.495
	mean			0,910	

Table IX Case (II)A: People detection results on YOLO-CAM51-52- Model

The Case (ii)b has been created by starting from YOLO-Model and performing a fine-tuning by using three training sets: CAM51-Training, CAM52-Training, and LLVIP-Training (see Table V). The results are reported in Table X in which the produced YOLO-CAM51-52-LLVIP-Model has been assessed against the test datasets for each class of people counting. As a result, the YOLO-CAM51-52-LLVIP-Model provided a valid solution to the problem of people detection and achieved comparable results as the specific model trained on the LLVIP achieving/confirming a mAP_0.5 of 0.96, a non-reduction of performance on the LLVIP case. The mean value of mAP_0.5 on all cases resulted to be 0.9278, while it has been of 0,9247 taking into account only the results of test set coming from the CAM51/52.

Test dataset		**precision**	**recall**	**mAP_0.5**	**mAP_0.05:0.95**
LLVIP	<=10	0.959	0.928	0.959	0.690
CAM51	<=10	0.944	0.902	0.964	0.742
	>10&<=25	0.923	0.956	0.969	0.751
	>25&<=50	0.927	0.907	0.943	0.706
	>50&<=75	0.926	0.846	0.910	0.655
	> 75&<=97	0.918	0.864	0.915	0.626
CAM52	<=10	0.984	0.954	0.974	0.686
	>10&<=25	0.99	0.927	0.963	0.687
	>25&<=50	0.946	0.852	0.910	0.628
	>50&<=75	0.903	0.846	0.895	0.588
	> 75&<=79	0.906	0.678	0.804	0.515
	Mean of cam51/51			0,9247	
	Mean on all			0.9278	

Table X Case (II)b: People detection results on YOLO-CAM51-52-LLVIP- Model

On the test datasets of CAM51 and CAM52 the results are comparable and, in some cases, also better than the specific model as in the case with <=10 people on CAM52 the combined model achieves a mAP_0.5 of 0.974 and the specific one

0.962. The results with a crowded situation in our case study are also valid and especially when there are more than 75 people in Piazza Della Signoria the combined model achieves better mAP_0.5 scores for the CAM52 (with a mAP_0.5 of 0.804 with respect to the 0.786 of the specific model). Thus, some improvements in the flexibility among different cameras have been obtained with results of Table X, and at the expense of the best results obtained for CAM51 case, Table VI.

IV. Bottom-up layerwise domain adaptation

With the aim of providing a solution that can preserve both flexibility in terms of range of people and camera kinds, an additional solution has been proposed. Kiew et al., in [45] performed an extensive assessment comparing top-down and bottom-up domain adaptation strategies on thermal images and proposed a bottom-up layer wise domain adaptation on YOLOv3 architecture, outperforming the best performing single-modality pedestrian detection results on the KAIST and the FLIR-ADAS dataset.

In [45], domain adaptation attempted to exploit learned knowledge from a source domain (RGB images) in a new target related domain (thermal images). In our case, we performed fine-tuning process starting from a pre-trained model on COCO dataset (RGB images) using thermal images. Therefore, our approach of fine-tuning is a top-down domain adaptation on the thermal domain via back-propagation where the supervision signal comes from the loss at the top of the network down to the new input distribution.

On the other hand, **the bottom-up layerwise domain adaptation** is based on the hypothesis that fine-tuning slowly from the bottom of the network should preserve more knowledge from the original domain.

Differently from [45], we applied the approach **bottom-up layerwise domain adaptation** fully on thermal data and on the YOLOv5 architecture, in which it has been realized considering the epoch i during the training process, freezing the layers 3i+1:N, where N is the total number of layers of the considered architecture (for YOLOv5s is 177). Therefore, at the starting epochs of the training process, the base layers are trainable, and the other upper part of the network is frozen. After every epoch, the 3i+1 layers are unfrozen until the entire network is fine-

tuned. We applied this process on the YOLOv5 small architecture, which is made up of three main components:

- The model backbone: CSPDarknet [46] that extracts features from the input image and is composed by Simplified Cross Stage Partial Bottleneck blocks C3 and a cascaded faster version of Spatial Pyramid Pooling Layer SPPF [47].
- The model neck: PANet [48] that elaborates feature pyramids to generalize objects in different scales.
- The model head: YOLO-head that performs the final detection.

The YOLOv5s is made up of 27 blocks with a total of 177 (N) layers and a total of 7.2M parameters. The YOLOv5s architecture is reported in Figure 6 and the bottom-up layerwise process is summarized graphically in Figure 7.

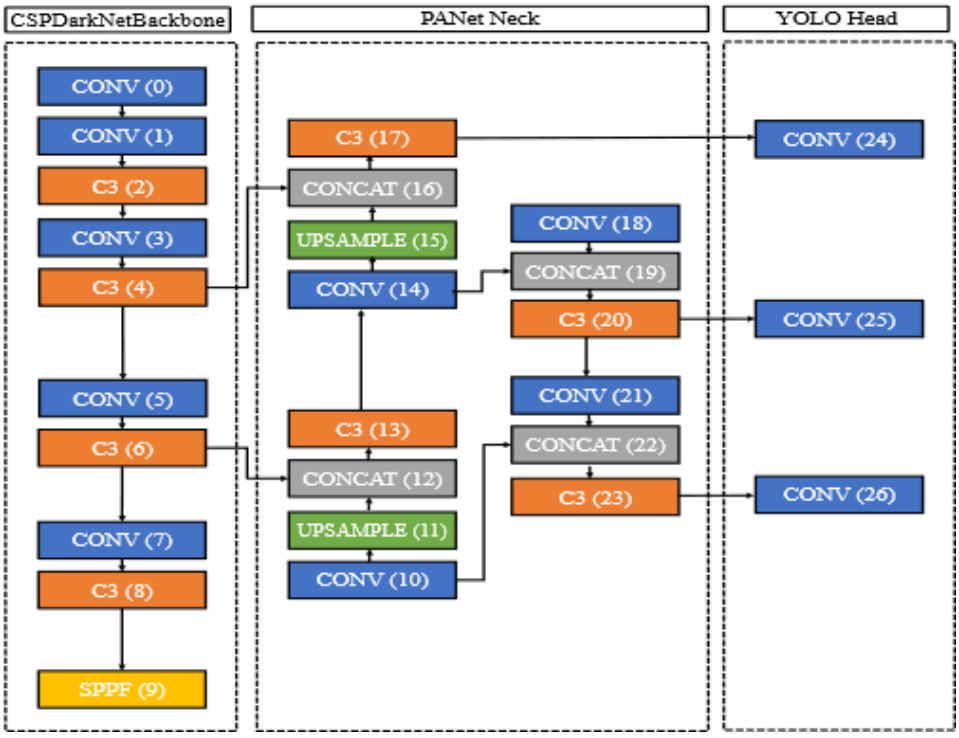

Fig 6. Yolov5s architecture.

We applied the bottom-up layerwise domain adaptation starting from the YOLOv5s pre-trained on COCO dataset first using only the single camera datasets, then using the combination of both camera datasets but the results did not improve the top-down fine-tuning reported in Section IV. Using the bottom-up layerwise domain adaptation on the union of LLVIP, CAM51 and CAM52 training datasets we improved with respect of the top-down strategy. The results

for the multi-class object detection on the validation dataset (LLVIP + CAM51 + CAM52) are reported in Table XI, and the resulting bottom-up model called YOLO-CAM51-52-LLVIP-BLDA-Model. This model achieves a mAP_0.5 on the validation dataset made up of the union of LLVIP CAM51 CAM52 validation datasets of 0.966.

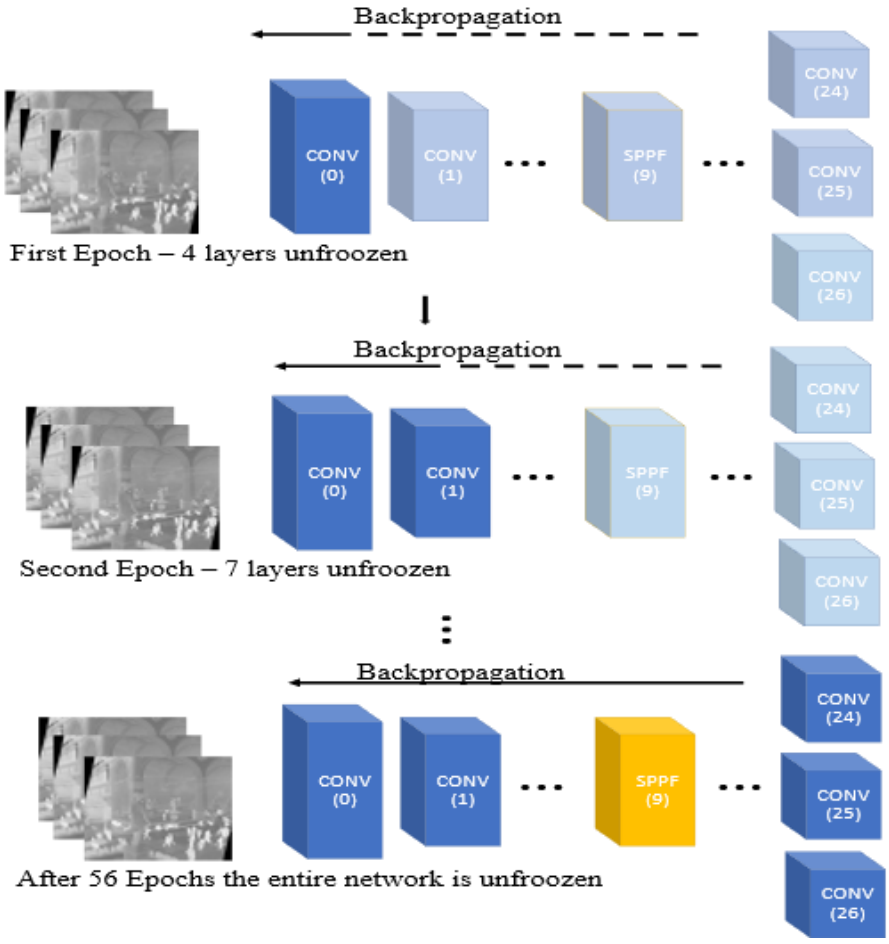

Fig 7. Bottom-up layerwise domain adaptation.

model	precision	recall	mAP_0.5	mAP_0.5:0.95
YOLO-CAM51-52-LLVIP	0.929	0.942	0.963	0.582
YOLO-CAM51-52-LLVIP-BLDA	0.943	0.930	0.966	0.563

Table XI Comparison of Multiclass object detection results in YOLO-CAM51-52-LLVIP-Model, and YOLO-CAM51-52-LLVIP-BLDA-Model

To compare these results with respect to those of the Multiclass Object Detection reported in Section IV, the YOLO-CAM51-52-LLVIP-BLDA-Model has been

validated on the union of the validation datasets CAM51 and CAM52. The results in terms of confusion matrix are reported in Figure 8.

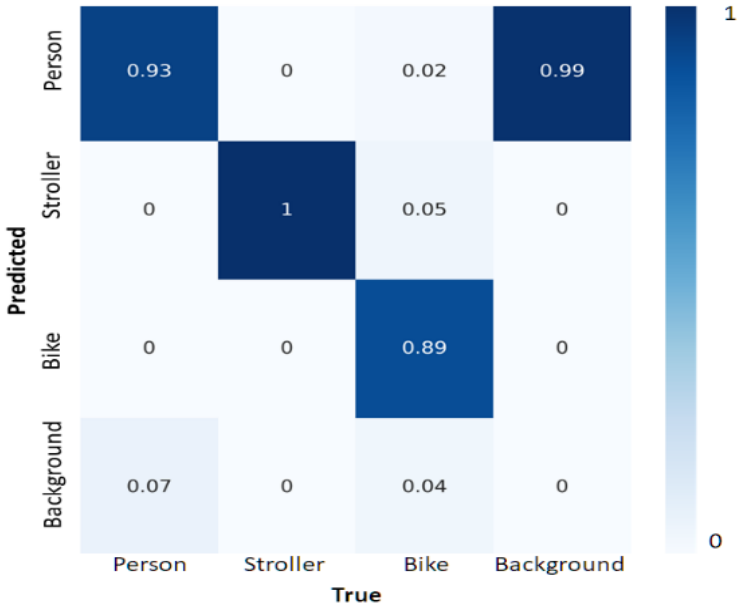

Fig 8. Confusion Matrix validation YOLO-CAM51-52-LLVIP-BLDA-Model

The YOLO-CAM51-52-LLVIP-BLDA-Model for the Multiclass Object Detection task achieves for the person class a percentage of True Positive detected objects of 0.93, for the bike class 0.89 and detected all the strollers presented in the validation sets. Comparing these results using the specifics CAM51-Model and CAM52-Model on the respective validation datasets reported in Section IV, the percentage of True Positive detected objects for the person class improves with respect to the 0,895 of the specific models, for the stroller class achieves the same totality of detection on the validation datasets, and for the bike class achieves a slight less value of 0.89 compared to the 0.895 of using the specific camera models. So based on these results this one bottom-up model provides high flexibility for detection of people, bikes, strollers on different image types from telephoto to wide-angle cameras.

When considering the analysis on the datasets with number of people in the image, the results in terms of precision, recall, mAP_0.5, mAP_0.5:0.95 are reported in Table XII. Considering the mAP_0.5 on the LLVIP validation dataset YOLO-CAM51-52-LLVIP-BLDA-Model achieves a value of 0.959 confirming the results of the YOLO-LLVIP-Model. When considering the test datasets of

CAM51 in the categories with number of people <=10 and >10&<=25 the YOLO-CAM51-52-LLVIP-BLDA-Model achieves the best results over all the models proposed with respectively mAP_0.5 values of 0.979 and 0.974. Regarding the test datasets of CAM52 in all the categories from people <=10 up to 79 this model achieves the best results over all the models considered in this work.

Test dataset		**precison**	**recall**	**mAP_0.5**	**mAP_0.05: 0.95**
LLVIP	<=10	0.960	0.917	**0.959**	0.683
CAM 51	<=10	0.881	0.991	**0.979**	0.768
	>10&<=25	0.957	0.930	**0.974**	0.757
	>25&<=50	0.915	0.920	**0.956**	0.714
	>50&<=75	0.930	0.824	**0.916**	0.636
	> 75&<=97	0.892	0.849	**0.923**	0.603
CAM 52	<=10	0.992	0.977	**0.993**	0.682
	>10&<=25	0.964	0.966	**0.980**	0.670
	>25&<=50	0.950	0.865	**0.950**	0.621
	>50&<=75	0.884	0.877	**0.921**	0.567
	> 75&<=79	0.879	0.754	**0.851**	0.493
	Mean of cam51/52			**0.9443**	
	Mean on all			**0.9456**	

Table XII People detection results on YOLO-CAM51-52-LLVIP-BLDA-Model. In bold the best results wrt Table IX of CAM51-52-LLVIP-Model

Therefore, on the basis of these results, the YOLO-CAM51-52-LLVIP-BLDA-Model provides better results with respect to the YOLO-CAM51-52-LLVIP-Model of Table XII, thus providing high flexibility with respect to the different number of people in the scenes, and also at the change of the camera resolution and lenses. The mean value of mAP_0.5 on all cases resulted to be 0.9456, while it has been of 0,9443 taking into account only the results of test set coming form the CAM51/52.

V. Deployment architecture

In compliance with GDPR rules, the system uses two thermal cameras CAM51 (which is an AXIS Q1951-E), and CAM52 (AXIS Q1952-E). The Q1951-E has a telephoto 35mm camera lens with a horizontal field of view of 10.5° and F1.14 with 768x576 pixels images. The Q1952-E has a wide-angle 10mm camera lens with a horizontal field of view of 63° and F1.17 with 640x480 pixels images but it has been positioned vertically. In order to process the images of the Q1952-E has

been applied a wide-angle correction using the undistortImage function of the Fisheye camera model of the OpenCV library [49]. The deployment of the solution can be performed in two manners, as reported in Figure 9.

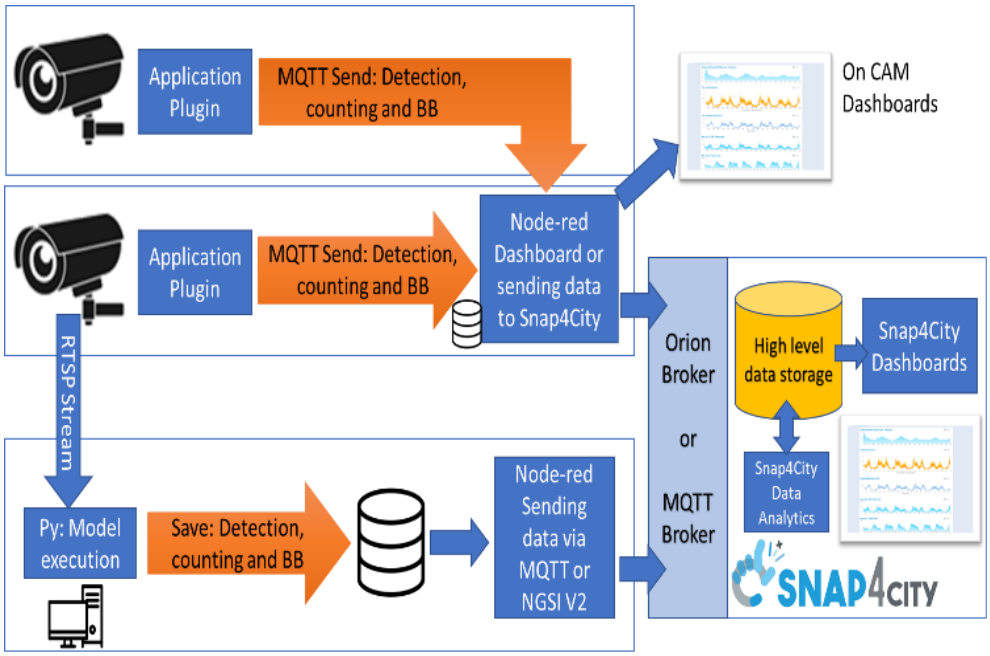

Fig 9. System Architecture for the two modalities of deployment: above on board at the TV CAM, below on an industrial PC connected to the camera via RTSP. Combinations of these cases are also viable.

Firstly, the AXIS cameras are ARM7 architecture for which we developed an Application Plugin in C++ to execute the trained models. Therefore, in this case, we can state that the model execution is performed on Edge into the Application Plugin. The Application Plugin (see Figure 10) can show the results (image and related bounding boxes, and data below the image with the bounding box and the classification) on the web interface of the camera and produce MQTT messages with bounding box of detected people and their classifications. The MQTT messages can be sent outside to some MQTT broker as well on local Node-RED (with an internal Aedes MQTT broker) installed in the camera, on which Snap4City Library can be also installed to send data in protected manner to some server and create dashboards. In addition, via Node-RED it is possible to create some on CAM dashboard or collect data coming from multiple cameras to perform data aggregation, reasoning and providing higher level results to be sent on cloud via MQTT or other protocol.

The second possibility is based on a process in Python that receives the RTSP stream from one camera, executes the model for people detection, and save the detected bounding boxes directly on some local database (eventually it could even send the information via some protocol or rest Call). The execution is performed on Python program executing the training model on an NVIDIA T1000 4gb GPU. In this case, the model execution can process 8 frames per second, which is the number of frames produced by the thermal camera. On this appliance (industrial PC), a NODE-RED can be installed to get the data from the database and send them to Snap4City framework infrastructure via MQTT or NGSI V2 messages.

Fig 10. Results Page of the Axis native App.

In both cases, the data arrive on Snap4City, on which a number of IoT Devices have been used to receive data and visualize results in dashboards in Real-Time. Thus, Snap4City dashboards have been used to show the results as in Figure 11, which reports the trends of the number of people for a week for both cameras, and on which the drill down on time trend can be performed.

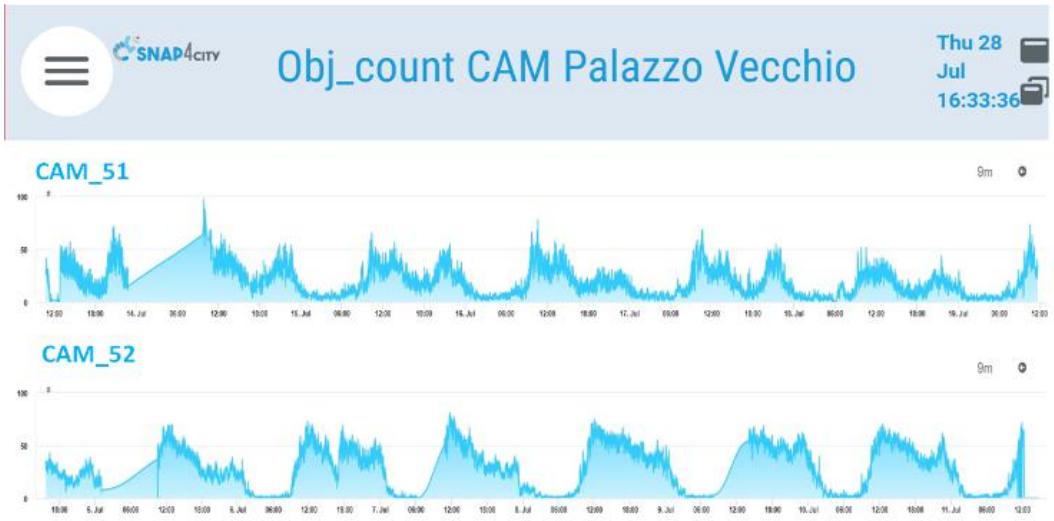

Fig 11. Monitoring Dashboard of people counting in Piazza Della Signoria, Florence

V.A. Edge-device execution performance analysis

Regarding the Application Plugin installed on the CAM51 the execution time has been assessed and tuned. In fact, the execution time also depends on the number of people detected. In Figure 12, the execution time as a function of the number of bounding boxes is reported. The analysis has been performed by processing the image data acquired from the 14/07/2022 to 19/07/2022 when the interval of boxes/people detected has been from 0 to 60. Based on these execution times starting from the base scenario (0 boxes) with an execution time of 9.174 seconds, the time increment for detecting a box is of about 0.077 seconds.

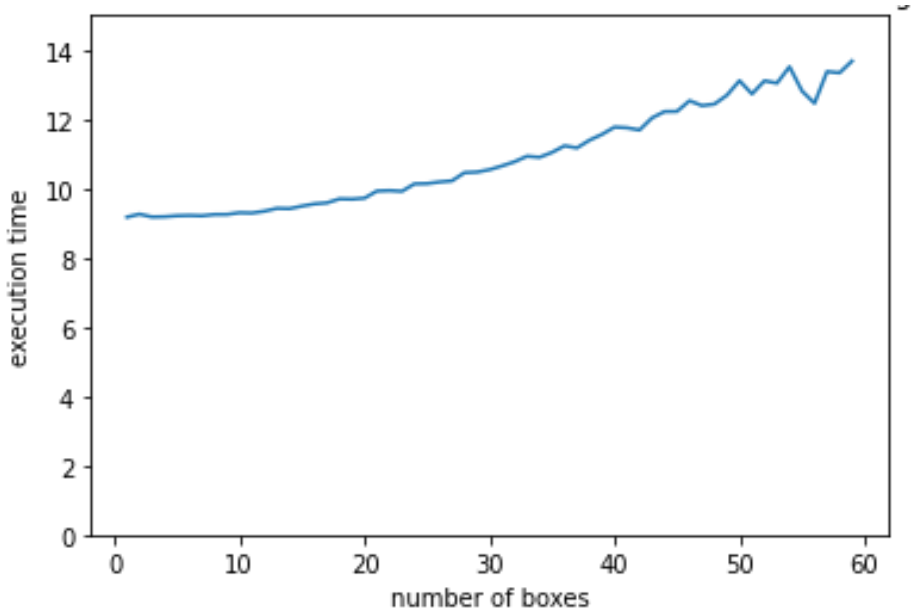

Fig 12. Mean Execution time based on the number of boxes detected on the native app installed on CAM51.

6. Conclusions

An important role for Smart Cities is played by Tourism management applications especially to study solutions for the quality of experience of tourists in crowded sites. In this work, we proposed the use of thermal cameras which are not invasive and respect the privacy in compliance with GDPR. In this paper, it has been proposed an approach that starts from the elaboration of the videos using state-of-the-art Computer Vision Algorithms for multiclass object detection of people, bikes and strollers, and uses the results to create a monitoring dashboard. For the multiclass object detection task, YOLO has been used which performs single-shot object detection providing at the same time low computational time performances and good detection accuracy over all the considered classes. YOLO has been compared with and Faster-R-CNN and both pretrained and then fine-tuned with LLVIP with the aim of obtaining good flexibility of people detection in a range of people numbers. Since the results were not satisfactory, specific training sets have been produced for fine tuning obtaining betters results for YOLO wrt Faster-R-CNN. We investigated a solution that could be flexible for cameras with different lenses from wide-angle to telephoto, taking also into account the number of objects in the scenes. We tested and compared a set of tuning approaches for improving the precision and flexibility of the previous solutions at the state-of-the-art. To this end, we explored both top-down and bottom-up training adaptation

approaches, demonstrating that the bottom-up approach can provide the best results according to the above-mentioned objectives of performance and flexibility. Tuning the YOLOv5 architecture based on a bottom-up layerwise domain adaptation responded to the need for low computational time while achieving a mean mAP_0.5 on the object of the scene on the test datasets of 0.986 for scenarios with less than 10 objects, and 0.9456 with mixed scenes with up to 97 objects (see Table XII). Moreover, the solution has been tested in two possible deployment configurations: (i) an industrial PC with GPU that could provide Real-Time processing results and, (ii) a direct installation on the thermal camera (that is on edge) that can elaborate in the worst condition 2 frames per minute. The solution has been massively tested on Piazza Della Signoria, Florence, Italy, sending data to Snap4City platform and Dashboards.

Acknowledgements. The authors would like to thank the Herit-Data (https://herit-data.interreg-med.eu/) for partially founding the reported research, and AXIS for their support for thermal cameras. Km4City and Snap4City (https://www.snap4city.org) are open technologies and research of DISIT Lab.

Data Availability. The data that support the findings of this study are available from Https://www.snap4city.org/ but restrictions apply to the availability of the original source data, which were used under licence for the current study.

Conflict of Interest: The authors declare that they have no conflict of interest

References

[1] GDPR: General Data Protection Regulation, https://gdpr.eu/

[2] Badii, Claudio, et al. "Microservices suite for smart city applications." *Sensors* 19.21 (2019): 4798.

[3] Wu, Hefeng, et al. "Multipoint infrared laser-based detection and tracking for people counting." *Neural Computing and Applications* 29.5 (2018): 1405-1416.

[4] Li, Shengye, et al. "Supervised people counting using an overhead fisheye camera." *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019.

[5] I. Udrea, C. G. Alionte, G. Ionaşcu, and T. C. Apostolescu, “New research on People Counting and Human Detection,” in 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2021, pp. 1–6.

[6] https://www.wsj.com/articles/a-billion-surveillance-cameras-forecast-to-be-watching-within-two-years-11575565402

[7] X. Jia, C. Zhu, M. Li, W. Tang, and W. Zhou, LLVIP: A Visible-infrared Paired Dataset for Low-light Vision. arXiv, 2021. doi: 10.48550/ARXIV.2108.10831.

[8] M. Krišto, M. Ivasic-Kos and M. Pobar, "Thermal Object Detection in Difficult Weather Conditions Using YOLO," in *IEEE Access*, vol. 8, pp. 125459-125476, 2020, doi: 10.1109/ACCESS.2020.3007481.

[9] M. Ł. Kowalski et al., "Detection of Inflatable Boats and People in Thermal Infrared with Deep Learning Methods," Sensors, vol. 21, no. 16, 2021, doi: 10.3390/s21165330.

[10] K. Yin et al., "Multi-scale Object Detection Algorithm in Smart City Based on Mixed Dilated Convolution Pyramid," 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), 2021, pp. 590-597, doi: 10.1109/SWC50871.2021.00088.

[11] L. R. Barba Guamán, J. Naranjo, A. Ortiz, and J. Pinzon Gonzalez, "Object Detection in Rural Roads Through SSD and YOLO Framework," 2021, pp. 176–185. doi: 10.1007/978-3-030-72657-7_17.

[12] A. Khalfaoui, A. Badri and I. E. Mourabit, "Comparative study of YOLOv3 and YOLOv5's performances for real-time person detection," 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2022, pp. 1-5, doi: 10.1109/IRASET52964.2022.9737924.

[13] M. Karthi, V. Muthulakshmi, R. Priscilla, P. Praveen and K. Vanisri, "Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset," 2021 Int. Conf. on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-6.

[14] K. My, A. Bagdanov, M. Bertini, and A. Bimbo, "Task-Conditioned Domain Adaptation for Pedestrian Detection in Thermal Imagery," 2020, pp. 546–562.

[15] R. Goel, A. Sharma and R. Kapoor, "Deep Learning Based Thermal Object Recognition under Different Illumination Conditions," *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2021, pp. 1227-1233.

[16] Dai, X., Yuan, X. & Wei, X. TIRNet: Object detection in thermal infrared images for autonomous driving. *Appl Intell* **51,** 1244–1261 (2021).

[17] Kera SB, Tadepalli A, Ranjani JJ. A paced multi-stage block-wise approach for object detection in thermal images. Vis Comput. 2022 Apr 7:1-17. doi: 10.1007/s00371-022-02445-x. Epub ahead of print. PMID: 35411122; PMCID: PMC8987521.

[18] F. Munir, S. Azam and M. Jeon, "SSTN: Self-Supervised Domain Adaptation Thermal Object Detection for Autonomous Driving," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 206-213.

[19] S. Li, Y. Li, Y. Li, M. Li and X. Xu, "YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection," in IEEE Access, vol. 9, pp. 141861-141875, 2021, doi: 10.1109/ACCESS.2021.3120870.

[20] Jocher, Glenn, et al. "ultralytics/yolov5." Github Repository, YOLOv5 (2020).

[21] T.-Y. Lin et al., Microsoft COCO: Common Objects in Context. arXiv, 2014. doi: 10.48550/ARXIV.1405.0312.

[22] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[23] J. W. Davis, M. A. Keck, “A two-stage template approach to person detection in thermal imagery.” In Proc. 7th IEEE Workshops Appl. Comput. Vis. (WACV/MOTION), vol. 1, 364–369, 2005.

[24] Yukyung Choi, “KAIST Multi-spectral Day/Night Dataset for Autonomous and Assisted Driving,” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), 2018.

[25] FLIR Thermal Dataset. [Online] Available: https://www.flir.it/oem/adas/adas-dataset-form/

[26] L. Liu, J. Chen, H. Wu, G. Li, C. Li, and L. Lin, “Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4823–4833.

[27] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, “Locality-constrained spatial transformer network for video crowd counting,” in 2019 IEEE International Conference on Multimedia and Expo (ICME), 2019, pp. 814–819.

[28] M. Xu, “An Efficient Crowd Estimation Method Using Convolutional Neural Network with Thermal Images,” in 2019 IEEE Int. Conf. on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1–6.

[29] J. Fu, H. Yang, P. Liu, and Y. Hu, “A CNN-RNN neural network join long short-term memory for crowd counting and density estimation,” in 2018 IEEE Int. Conf. on Advanced Manufacturing (ICAM), 2018, pp. 471–474.

[30] S. Sharath, V. Biradar, M. Prajwal, and B. Ashwini, “Crowd Counting in High Dense Images using Deep Convolutional Neural Network,” in 2021 IEEE Int. Conf. on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 2021, pp. 30–34.

[31] A. Menon, B. Omman, and S. Asha, “Pedestrian Counting Using Yolo V3,” in 2021 Int. Conf. on Innovative Trends in Information Technology (ICITIIT), 2021, pp. 1–9.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[33] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In IEEE Conf. on Computer Vision and Pattern Recognition, 2016.

[34] M. I. H. Azhar, F. H. K. Zaman, N. M. Tahir, and H. Hashim, “People tracking system using DeepSORT,” in 2020 10th IEEE international conference on control system, computing and engineering (ICCSCE), 2020, pp. 137–141.

[35] A. Belmouhcine, J. Simon, L. Courtrai, and S. Lefèvre, “Robust Deep Simple Online Real-Time Tracking,” in 2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA), 2021, pp. 138–144. doi: 10.1109/ISPA52656.2021.9552062.

[36] Y. Zhang, Z. Chen, and B. Wei, “A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement,” in 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1312–1316.

[37] J. Stovall, A. Harris, A. O’Grady, and M. Sartipi, “Scalable Object Tracking in Smart Cities,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3813–3819. doi: 10.1109/BigData47090.2019.9005472.

[38] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes Challenge: A Retrospective,” International Journal of Computer Vision, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[39] Herit-Data Interreg project. Innovative Solutions to Better Manage Tourism Flow Impact on Cultural and Natural Heritage Sites Through Technologie and Big Data, https://herit-data.interreg-med.eu/

[40] Garau C., Nesi P., Paoli I., Paolucci M., Zamperlin P. A Big Data Platform for Smart and Sustainable Cities: Environmental Monitoring Case Studies in Europe (2020) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12255 LNCS, pp. 393 – 406.

[41] Han Q., Nesi P., Pantaleo G., Paoli I., Smart City Dashboards: Design, Development, and Evaluation, (2020) Proceedings of the 2020 IEEE International Conference on Human-Machine Systems, ICHMS 2020, art. no. 9209493,

[42] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).

[43] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, & Ross Girshick. (2019). Detectron2.

[44] https://github.com/developer0hye/Yolo_Label.

[45] Kieu, M & Bagdanov, Andrew & My, Kieu & Bertini, Marco. (2020). Bottom-up and Layer-wise Domain Adaptation for Pedestrian Detection in Thermal Images. ACM Transactions on Multimedia Computing Communications and Applications. 10.1145/3418213.

[46] Y. Yin, H. Li, and W. Fu, “Faster-YOLO: An accurate and faster object detection method,” Digital Signal Processing, vol. 102, p. 102756, 2020, doi: https://doi.org/10.1016/j.dsp.2020.102756.

[47] https://github.com/ultralytics/yolov5/pull/4420/files

[48] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.

[49] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.