Benchmark for anonymous video analytics

Ricardo Sanchez-Matilla* and Andrea Cavallaro*

*Correspondence:
ricardo.sanchezmatilla@qmul.
ac.uk; a.cavallaro@qmul.ac.uk
Centre for Intelligent Sensing,
Queen Mary University
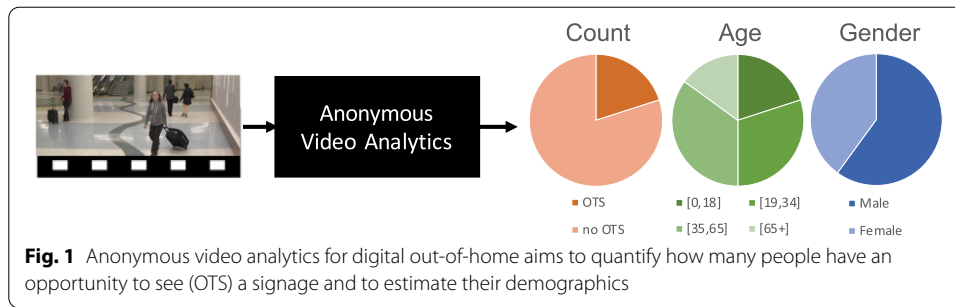of London, London, UK

Abstract

Out-of-home audience measurement aims to count and characterize the people exposed to advertising content in the physical world. While audience measurement solutions based on computer vision are of increasing interest, no commonly accepted benchmark exists to evaluate and compare their performance. In this paper, we propose the first benchmark for digital out-of-home audience measurement that evaluates the vision-based tasks of audience localization and counting, and audience demographics. The benchmark is composed of a novel, dataset captured at multiple locations and a set of performance measures. Using the benchmark, we present an in-depth comparison of eight open-source algorithms on four hardware platforms with GPU and CPU-optimized inferences and of two commercial off-the-shelf solutions for localization, count, age, and gender estimation. This benchmark and related open-source codes are available at http://ava.eecs.qmul.ac.uk.

Keywords: Anonymous video analytics, Benchmark, Audience measurement, Performance evaluation

1 Introduction

Digital out-of-home advertisement is rapidly growing thanks to the availability of affordable, internet-connected smart screens. Anonymous video analytics (AVA) aims to enable real-time understanding of audiences exposed to advertisements in order to estimate the reach and effectiveness of each advertisement. AVA ensures the preservation of the privacy of audience members by performing inferences and aggregating them directly on edge systems, without recording or streaming raw data (Fig. 1).

AVA relies on person detectors or trackers to localize people and to enable the estimation of audience attributes, such as their demographics. AVA should produce accurate results, be robust to environmental variations and varying illumination. While well-established performance measures exist to evaluate generic computer vision algorithms [1–4], these measures do not take into account the desirable features of AVA for out-of-home advertisement, such as the opportunity for a person to see the advertisement. In fact, multiple datasets exist to benchmark detection [5, 6], tracking [7–11] and re-identification algorithms [12]. Further datasets include KITTI [5], which focuses on autonomous driving, and Common Objects in Context (COCO) [6], which covers object detection, segmentation, and captioning. Moreover, the robust vision challenge [13] evaluates scene reconstruction, optical flow, semantic and instance segmentation, and

Fig. 1 Anonymous video analytics for digital out-of-home aims to quantify how many people have an opportunity to see (OTS) a signage and to estimate their demographics

depth prediction tasks, the visual object tracking benchmark [10, 11] compares single-object tracking algorithms and the multiple-object tracking benchmark [7–9] compares multiple-object trackers. These benchmarks and datasets are designed for scenarios that differ from those for digital out-of-home advertisement and their annotations lack relevant information, such as demographics and attention of the audience, for assessing AVA algorithms.
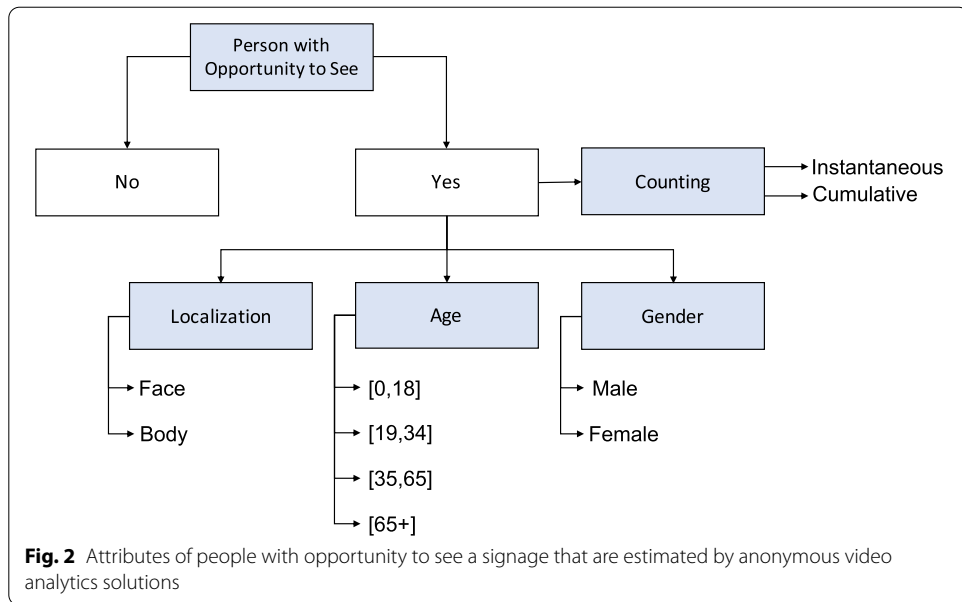
Because of the growing importance of digital out-of-home advertisement and the lack of a standard evaluation protocol, in this paper we present the benchmark for anonymous video analytics[1]. This work is the first publicly available benchmark specifically designed to evaluate AVA solutions. The benchmark includes a set of performance measures specifically designed for audience measurement, an online evaluation tool, a novel fully annotated dataset for digital out-of-home AVA, and open-source baseline algorithms and evaluation codes. The dataset annotations include over a million localization bounding boxes, and age, gender, attention, pose, and occlusion information. We also benchmark eight baseline algorithms: two face detectors, two person trackers, two age estimators, two gender estimators; and two commercial off-the-shelf solutions.

The paper is structured as follows. Section 1.1 introduces the main definitions of the work and describes the proposed analytics for AVA; Sect. 1.2 presents the performance measures used for benchmarking; Sect. 2 describes the detection, tracking, age, and gender estimation algorithms; Sect. 3 introduces the proposed dataset and its annotation; Sect. 4 presents the benchmarking results; and Sect. 5 summarizes the findings of the work.

1.1 Analytics

Let a digital signage be equipped with a system and a camera. The system is a computer that manages the advertisement playback and processes the video of the surroundings of the signage captured by the camera. Let the video $\mathcal{V} = \{I_t\}_{t=1}^{T}$ be composed of T frames I_t, each with frame index t. We consider the following attributes about the people in the video: count, age, and gender (see Fig. 2). To enable the estimation of the above attributes, we also consider the localization of people in $\mathcal{V}$, namely their position and dimensions in I_t. We consider a person in I_t to have opportunity to see (OTS) the signage when their face is visible from its left profile to its right profile, and the person is not heading

---

[1] Benchmark website http://ava.eecs.qmul.ac.uk.

Fig. 2 Attributes of people with opportunity to see a signage that are estimated by anonymous video analytics solutions

Fig. 3 A person is said to have opportunity to see (OTS) the signage when their face is visible (in any pose from the left to the right profile) and their heading direction is *not* opposite to the signage

opposite to the location of the camera, as shown in Fig. 3. We consider only the attributes of people with OTS.

Let the estimated location and dimensions of person $j \in \mathbb{N}$ with OTS in I_t be represented with a bounding box $\hat{\mathbf{d}}_t^j = [x, y, w, h]$, where $\mathbb{N}$ is the set of the natural numbers. The bounding box is defined by the horizontal, x, and vertical, y, image coordinates of its top-left corner, and by its width, w, and height, h. The location of person j, $\hat{\mathbf{d}}_t^j$, may be represented by their face or their body, and can be estimated with a detection or tracking algorithm. While with a detector, the index j may change over time (i.e., the index j is not related to the identity of the person), trackers aim to maintain the index j consistent over time.

Localization algorithms enable the estimation of the number of people with OTS (counting) at time t, $n_t \in \mathbb{N}$, and trackers enable the estimation of the number of unique[2] cumulative people with OTS within a time window between t_1 and t_2, $n_{t_1:t_2} \in \mathbb{N}$.

If $\mathcal{A}$ is a set of age ranges, an age estimation algorithm is expected to determine the age, $a_t^j \in \mathcal{A}$, of a person with OTS, $\hat{\mathbf{d}}_t^j$, with $\mathcal{A}$ defined as:

---

[2] Unique means that the algorithm does not repeatedly count the same person over time.

$$\mathcal{A} = \{[0, 18], [19, 34], [35, 65], [65+], \text{unknown}\}. \tag{1}$$

These age ranges have been selected as they are commonly used in audience analytics.

A gender estimation algorithm determines the gender, $g_t^j \in \mathcal{G}$, of each detected person with OTS, $\hat{\mathbf{d}}_t^j$, where

$$\mathcal{G} = \{\text{male}, \text{female}, \text{unknown}\} \tag{2}$$

is the set of possible classes.

In summary, for each person j with OTS, an AVA solution is expected to produce at each time t: j, the person index (for trackers, the tracking identity consistent throughout $\mathcal{V}$); $\hat{\mathbf{d}}_t^j$, the estimated location of the face and/or body; $\hat{a}_t^j \in \mathcal{A}$, the estimated age; and $\hat{g}_t^j \in \mathcal{G}$, the estimated gender.

1.2 Performance measures

We introduce a set of performance measures for assessing the accuracy of localization, counting, age and gender estimation. These measures, which are concise and easy to understand by a broad community, enable the evaluation and comparison of AVA algorithms.

Localization

We evaluate the localization performance based on precision (P), recall (R), and F1-score (F) [4], which are defined based on true positives (TP), false positives (FP), and false negatives (FN). Precision (P) is the ratio between correct estimations and the total number of estimations. Recall (R) is the ratio between correct estimations and the total number of actual occurrences. F1-score (F) is the harmonic mean of P and R.

For localization, we define TP, FP, and FN based on the intersection over union (IOU) operator between estimations, $\hat{\mathbf{d}}_t^j$, and annotations, $\mathbf{d}_t^j$:
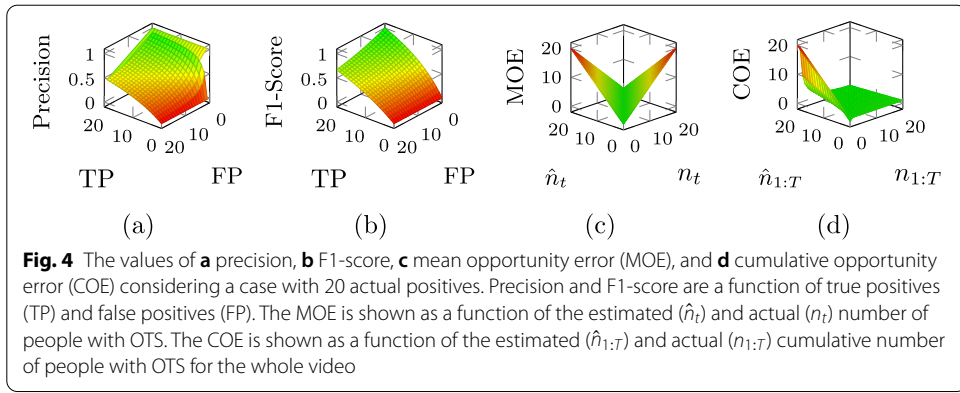
$$\text{IOU}(\hat{\mathbf{d}}_t^j, \mathbf{d}_t^j) = \frac{\hat{\mathbf{d}}_t^j \cap \mathbf{d}_t^j}{\hat{\mathbf{d}}_t^j \cup \mathbf{d}_t^j}, \tag{3}$$

where $\cap$ and $\cup$ are the intersection and the union operators, and IOU $\in [0, 1]$. An example for face localization, with highlighted TP, FP and FN detections is given in Fig. 5.

We consider the variation of R for different person–signage distances. Let $A(\cdot)$ be a function that computes the area in pixels for bounding box $\mathbf{d}_t^j$, and p_{dist}^a be the ath percentile of all the bounding boxes in the video. We define two bands for the person–signage distance, namely close when $A(\mathbf{d}_t^j) \geq p_{\text{dist}}^{50}$ and far when $A(\mathbf{d}_t^j) < p_{\text{dist}}^{50}$. We assume that people closer to the camera are annotated by a larger bounding box than for those who are farther.

We also consider the variation of R in presence of occlusions. Let $o_t^j \in \{\text{non-occluded}, \text{partially occluded}, \text{heavily occluded}\}$ be the annotated occlusion. The three occlusion bands are non-occluded when the annotation is not occluded; partially occluded when the annotated area is occluded less than 50%; and heavily occluded when the annotated area is occluded more or equal than 50%.

Note that we only report R, and not P, for person–signage distance and occlusion as false positives (necessary to compute P) cannot be unequivocally estimated when the

Fig. 4 The values of **a** precision, **b** F1-score, **c** mean opportunity error (MOE), and **d** cumulative opportunity error (COE) considering a case with 20 actual positives. Precision and F1-score are a function of true positives (TP) and false positives (FP). The MOE is shown as a function of the estimated ($\hat{n}_t$) and actual (n_t) number of people with OTS. The COE is shown as a function of the estimated ($\hat{n}_{1:T}$) and actual ($n_{1:T}$) cumulative number of people with OTS for the whole video

annotations are divided in regions such as far/close (for distance) or non-occluded/ partially occluded/heavily occluded (for occlusion). For instance, a far/close estimation might not match with any far/close annotation but this does not necessarily imply to be a false positive as it might be matching with a close/far annotation.

Counting

We quantify the performance of the localization algorithms for the task of people counting with the following performance measures: mean opportunity error (MOE), cumulative opportunity error (COE), and temporal cumulative opportunity error (TCOE).

The mean opportunity error (MOE) quantifies the ability of an algorithm to count people with OTS at a specific time t, $\hat{n}_t$, and it is calculated with respect to the actual number of people with OTS at t, n_t:

$$\text{MOE} = \frac{1}{T} \sum_{t=1}^{T} |\hat{n}_t - n_t|. \tag{4}$$

$\text{MOE} \geq 0$ (Fig. 4c) and its optimal value is $\text{MOE} = 0$. We analyze how MOE varies with the person–signage distance, as in localization, and when the input video frame rate is reduced. We show a visual sample of MOE in Fig. 5.

The cumulative opportunity error (COE) quantifies the ability of an algorithm to count unique people with OTS and it is calculated with respect to the actual cumulative number of people with OTS for the whole video, $n_{1:T}$:

$$\text{COE} = \frac{|\hat{n}_{1:T} - n_{1:T}|}{\max(n_{1:T}, 1)}, \tag{5}$$

where $\max(\cdot)$ is the max operation. $\text{COE} \geq 0$ and its optimal value is $\text{COE} = 0$. This performance measure is normalized with respect to the actual cumulative number of people; therefore, COE indicates the ratio of error with respect to the actual cumulative number of people.

The temporal COE (TCOE) quantifies the ability of an algorithm to count unique people with OTS over temporal segments of generic duration (e.g., 10-s duration), and it is calculated with respect to the cumulative number of unique people with OTS:
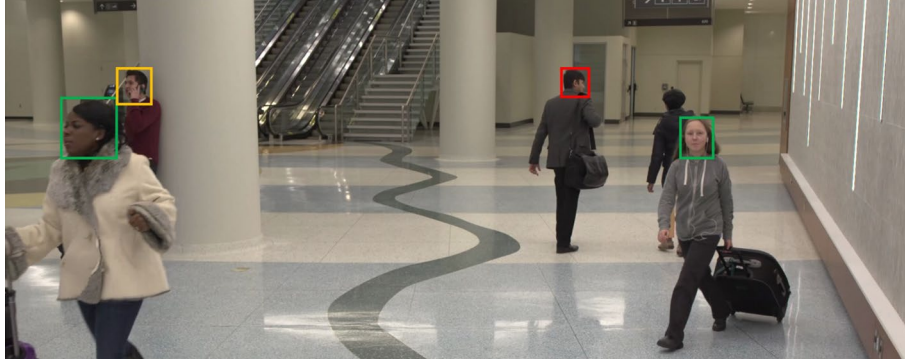
Fig. 5 Sample result and evaluation for people counting. Green, yellow, and red bounding boxes indicate true positives, false negatives, and false positive, respectively. Assuming that the frame is the first in a video, $t = 1$, the actual number of people is five ($p_1 = 5$), from which only three have OTS ($n_1 = 3$). The measures are MOE $= 0$, MPE $= 2$; and COE $= 0$. Note that the person marked with a red bounding, even that his face is visible, we consider he has no OTS as he is walking in a direction opposite to the location of the camera. The frame is cropped

Fig. 6 Sample temporal cumulative opportunity error (TCOE) computation for a video of 70 seconds, with an example of 10-second segment, $D = 10\gamma$, where γ is the frame rate of the video in frames per second. When computing TCOE, all segments of $D = \{10, 20, 30, 60, 90, 120\}\gamma$ frames (i.e., segments of 10, 20, 30, 60, 90, 120-second durations) within the video are used

$$\text{TCOE}_{D,T} = \frac{1}{|\mathcal{T}_{D,T}|} \sum_{\forall t \in \mathcal{T}_{D,T}} |\hat{n}_{t:t+D} - n_{t:t+D}|, \tag{6}$$

where $\mathcal{T}_{D,T} = \{1, 2, 3, \ldots, T - D\}$ is the set of the initial frame of each segment, $D < T$ is the duration in frames of the segments, and $|\mathcal{T}_{D,T}|$ is the total number of segments. We consider values that correspond to the typical duration of digital out-of-home advertisements $D = \{10, 20, 30, 60, 90, 120\}\gamma$, where γ is the video frame rate in fps (i.e., segments of 10, 20, 30, 60, 90, 120-s durations). TCOE considers all possible D-frame segments within the video. We show an example for 10-second segments in Fig. 6. $\text{TCOE}_{D,\mathcal{T}} \geq 0$ and its optimal value is $\text{TCOE}_{D,\mathcal{T}} = 0$. Note that when $D = T$, TCOE equals COE.

To quantify whether algorithms estimate all people in the field of view but fail to only estimate the ones with OTS, we define two accessory measures, mean people error (MPE) and cumulative person error (CPE). If p_t is the number of people at t and $p_{1:T}$ is the number of people in the whole video, then

$$\text{MPE} = \frac{1}{T} \sum_{t=1}^{T} |\hat{n}_t - p_t|, \tag{7}$$

and

Table 1 Example of classification results for a given attribute (i.e., gender) using three different algorithms. Note that an algorithm that can output unknown produces higher performance measures than one that commits the same amount of errors (e.g., algorithm B vs algorithm A). For gender estimation and for the class *female*, a true positive is a female estimated as female; a false negative is a female estimated as male; and a false positive is a male estimated as female

	Algorithm A	Algorithm B	Algorithm C
True positives	100	100	110
False negatives	51	51	51
False positives	24	14	14
Unknown	0	10	0
Precision	0.81	0.88	0.89
Recall	0.66	0.66	0.68
F1-score	0.73	0.76	0.77

Fig. 7 Age ranges and their overlap of $\pm$ 2 years for error calculation

$$\text{CPE} = \frac{|\hat{n}_{1:T} - p_{1:T}|}{\max(p_{1:T}, 1)}. \tag{8}$$

Attributes

We compute the per-class precision, recall, and F1-score for age and gender estimation. We consider the variation of F for different person–signage distances and for different occlusion levels.

We give now a few examples of the definitions of TP, FP, and FN for each attribute. For age estimation and the class [19,34], a TP is a correct age estimation. To relax the hardness of the age ranges, we consider overlapping age ranges with $\pm$ 2 years, as shown in Fig. 7 (e.g., an estimation of a person of 17 years will be a true positive if the actual age of the person is in [0,18], or [19,34]); a FP is an incorrect age estimation for a person from another age-class; and a FN is an incorrect estimation of the age of a person that belongs to the class. For gender estimation and the class *female*, a TP is a female estimated as female; a FP is a male estimated as female; and a FN is a female estimated as male.

For algorithms able to output unknown as a possible class, the corresponding estimations will contribute neither as TP, FP nor FN. We show an example of attribute evaluation in Table 1.

2 Methods

Algorithms must be accurate and perform in real-time for being suitable to generate reliable and useful AVA. Therefore, we select algorithms for benchmarking that obtain close to state-of-the-art results, that are causal (i.e., they only need past and present information), and that are able to perform close to real-time in the defined settings. We select algorithms that are compatible with GPU and OpenVINO[3] optimization for fast CPU computation.

We use RetinaFace [14] and Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks (MTCNN) [15] as detection algorithms; Simple Online Real-time Tracking with a Deep Association Metric (DeepSORT) [16], and Towards Real-Time Multi-Object Tracking (TRMOT) [17] as trackers; and FaceLib [18] and Deep EXpectation of apparent age from a single image (DEX) [19] as age and gender estimators. In addition to the above baseline algorithms, we benchmark two commercial solutions, Commercial-1 (C1) and Commercial-2 (C2), which we maintain anonymous. Localization, age, and gender estimation algorithms for AVA are described next and summarized in Table 2.

Algorithm 1 (A1), RetinaFace [14], is a face detector with a single-stage pixel-wise dense localization at multiple scales that uses joint extra-supervised and self-supervised multi-task learning. The algorithm predicts a face score, face box, five facial landmarks, and their relative 3D position, using input images resized to a resolution of 640×640 pixels. The algorithm is trained on the WIDER FACE dataset [20].

Algorithm 2 (A2), MTCNN [15], is a face detector with a cascaded structure and three stages of deep convolutional networks that use the correlation between the face bounding box and the landmark localization to perform both tasks in a coarse-to-fine manner. The first stage is a shallow network that generates candidate windows. The second stage rejects false positive candidate windows. The third stage is a deeper network that outputs the locations and facial landmarks. The learning process uses an online hard sample mining strategy that unsupervisedly improves the performance. The algorithm is trained on the WIDER FACE dataset [20].

Algorithm 3 (A3), DeepSORT [16], is a multi-object tracker that combines the detector YOLOv3 [21] and the tracker SORT [22]. YOLOv3 detects the body of people with a single neural network that divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. The predictions are informed by the global context of the image. With respect to the tracking module, DeepSORT uses Kalman filter and the Hungarian algorithm for performing association of detections over time. In addition, DeepSORT employs a convolutional neural network, trained to discriminate people, that combines appearance and motion information.

Algorithm 4 (A4), TRMOT [17], is a multi-object tracker based on the Joint Detection and Embedding (JDE) framework. JDE is a single-shot shared deep neural network that simultaneously learns detection and appearance features of the predictions. The algorithm is based on Feature Pyramid Network [23] that makes predictions at multiple

---

[3] OpenVINO framework is available at https://docs.openvinotoolkit.org/.

Table 2 Anonymous video analytics algorithms for localization, age, and gender estimation

Alias	Ref.	Name	Person part	Localization		Age	Gender	Implementation		Count performance			Speed
				Detection	Tracking			GPU	CPU	Instantaneous		Cumulative	
										People	OTS		
A1	[14]	RetinaFace	Face	✓				✓	✓	+	+++	+	+++
A2	[15]	MTCNN	Face	✓				✓	✓	+	+++	+	+++
A3	[16]	DeepSORT	Person		✓			✓	✓	+++	+	+++	++
A4	[17]	TRMOT	Person		✓			✓	✓	+++	+	+++	++
A5	[18]	FaceLib	Face	–	–	✓		✓	✓	–	–	–	+++
A6	[19]	DEX	Face	–		✓	✓	✓	✓	–	–	–	+++
C1	Commercial-1		Face		✓	✓	✓		✓	+	+++	+++	++
C2	Commercial-2		Face		✓	✓	✓		✓	+++	+++	+++	++

Algorithms count performance and speed are assessed, on a scale from 1 to 3 (the higher, the better) and regardless of the system used, based on generic algorithmic concepts (e.g., tracking outperforms detection for cumulative counts) and not on experimental results. OTS: opportunity to see. A4 was not compatible with OpenVINO optimization at submission time, thus we run A4 on CPU without optimization

scales. Then, embedding features are up-sampled and fused with the feature map from higher feature maps levels by using skip connections to improve the tracking accuracy for people far from the camera (i.e., small bounding boxes). The JDE framework learns to generate predictions and features simultaneously.

Algorithm 5 (A5), FaceLib [18], is an open-source repository for face detection, facial expression recognition, and age and gender estimation. The age and gender estimation modules use as input the true positive detections generated by RetinaFace (A1), and they use a ShuffleNet V2 with 1.0x output channels [24] as architecture. FaceLib is trained on the UTKFace dataset [25].

Algorithm 6 (A6), DEX [19], estimates the apparent age and gender using as input the true positive detections generated by RetinaFace (A1). DEX uses an ensemble of 20 networks on the detected face and it does not require the use of facial landmarks. This work introduces and uses the IMDB-WIKI, a large public dataset of face images with age and gender annotations.

Commercial off-the-shelf 1 (C1) is composed of a frontal face detector based on a cascade detector that uses a variety of gray-scale local features, a tracker designed for frontal, eye-level images; and a gender and age classifier based on regression trees.

Commercial off-the-shelf 2 (C2) uses a tracker based on a Kalman Filter for tracking people specifically designed to be robust to occlusions; and a gender and age classifier based on deep learning.

We employ A1–4 and C1-2 to estimate the instantaneous and cumulative number of people with OTS. The instantaneous number of people at t is estimated as the number of detections/tracks at the current time. The cumulative number of people between any two instants, t_1 and t_2, is estimated as the number of unique identities j during the considered time segment. Detectors (A1–2) simply assign a new identity to each detection without considering temporal relationships, thus producing an overcount. As trackers require detectors and a temporal association of detections mechanism to prevent multiple counts of the same person over time, in general trackers require more computational resources than detectors. Besides, trackers (A3–4) use person detectors (as opposed to face detectors), therefore trackers may not have skills to differentiate people with OTS from those who do not have OTS.
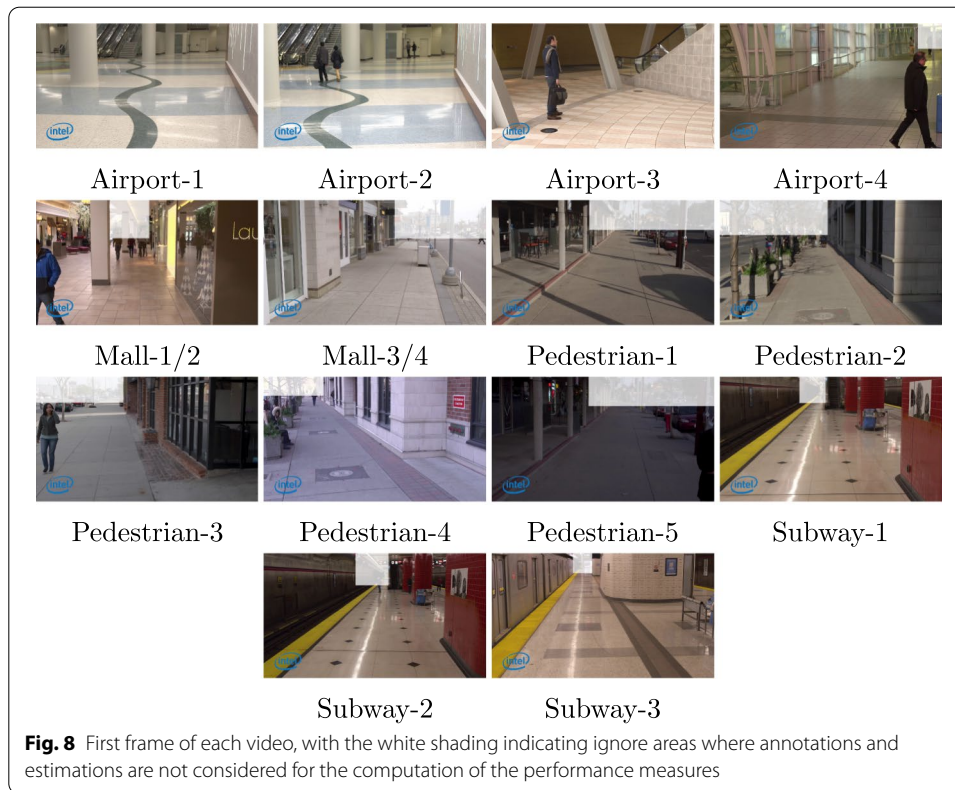
We make the codes and pre-trained model weights of all baseline algorithms available at the project website. Without altering the core of the original algorithms, we integrate changes to ensure that every algorithm processes the videos and generates the outputs following the same procedure. The codes for all baseline algorithms have been modified to enable both GPU (NVIDIA inference) and CPU (OpenVINO[4] inference) computation in PyTorch (Table 2).

3 Dataset

3.1 Videos

The dataset was collected in settings that mimic real-world signage–camera setups used for AVA. The dataset is composed of 16 videos recorded at different locations such as

---

[4] At submission time, some operations required by A4 (i.e., ScatterND) are not supported by OpenVINO framework; hence, no OpenVINO optimization is used with this algorithm.

Fig. 8 First frame of each video, with the white shading indicating ignore areas where annotations and estimations are not considered for the computation of the performance measures

airports, malls, subway stations, and pedestrian areas. Outdoor videos are recorded at different times of the day such as morning, afternoon, and evening. The dataset is recorded with Internet Protocol or USB fixed cameras with wide and narrow lenses to mimic real-world use cases. Videos are recorded at 1920×1080 resolution and 30 fps. The dataset includes videos of duration between 2 minutes and 30 seconds, and 6 minutes and 26 seconds, totaling over 78 minutes, with over 141,000 frames. The videos feature 34 professional actors with multiple ethnicities, with ages from 10 to 80, and including male and female genders. People have been recorded with varied emotions while looking at the signage. A sample frame of each location is shown in Fig. 8. We show sample frames with a reduced number of people for facilitating the visualization of the background. For the mall location, two videos are at different times: indoors (Mall-1/2) and outdoors (Mall-3/4).

3.2 Annotations

A professional team of annotators used the Intel Computer Vision Annotation Tool [26] from OpenVINO ecosystem to fully annotate all videos with the following attributes (Fig. 9): bounding boxes for face and body of the people, identity, age, gender, attention, pose, orientation, and occlusions. Annotations were generated for every key-frame. The key-frames were selected depending on the behavior and the location with respect to the camera of the person to be annotated. People closer to the camera (or moving faster) were annotated more often than people that are farther away (or moving slower). Inter-key-frames annotations were generated using linear interpolation. The annotations were

Fig. 9 Attributes used in the annotations of the dataset
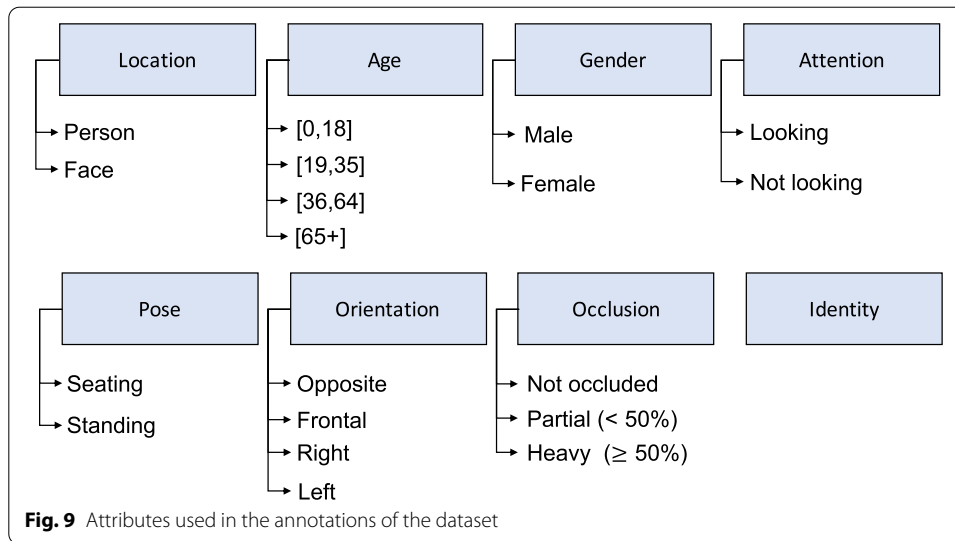
Table 3 The dataset

Video		Length		Unique people*		Number of localization annotations			
Name	Illumin. [Lux]	Time [min:s]	Frames	All	OTS	People	Per-frame	Faces	Per-frame
Airport-1	500	5:21	9629	37	29	22062	2.4 ± 1.0	12832	1.4 ± 1.1
Airport-2	500	5:34	10008	35	29	23600	2.7 ± 1.4	14214	1.6 ± 1.2
Airport-3	500	6:26	11578	47	44	26704	2.4 ± 1.2	17849	1.6 ± 1.0
Airport-4	500	5:08	9247	61	56	43685	4.7 ± 2.0	17792	1.9 ± 1.2
Mall-1	300	4:38	8344	158	111	106852	12.8 ± 2.3	45835	5.5 ± 1.8
Mall-2	300	3:41	6626	145	105	95417	14.4 ± 3.7	42779	6.5 ± 2.3
Mall-3	800	5:25	9740	33	30	37120	3.8 ± 1.5	18906	1.9 ± 1.2
Mall-4	800	6:04	10931	53	50	47113	4.3 ± 1.6	32038	2.9 ± 1.3
Pedestrian-1	60000	5:40	10202	18	17	39680	4.0 ± 1.7	19859	2.0 ± 1.4
Pedestrian-2	40000	6:15	11262	56	40	58477	5.2 ± 1.7	25042	2.2 ± 1.6
Pedestrian-3	7000	5:41	10220	27	25	22738	2.3 ± 1.2	13915	1.4 ± 1.0
Pedestrian-4	5500	4:32	8166	27	25	33031	4.0 ± 1.4	16248	2.0 ± 1.0
Pedestrian-5	250	2:58	5350	11	11	24476	4.6 ± 1.6	13504	2.5 ± 1.8
Subway-1	180	3:13	5795	17	17	36828	6.5 ± 3.1	25884	4.6 ± 2.7
Subway-2	180	2:32	4549	29	28	45125	9.9 ± 2.8	24248	5.3 ± 2.4
Subway-3	200	5:45	10342	31	29	85358	8.5 ± 2.9	35460	3.6 ± 1.7
Overall	[180,60000]	78:53	141989	785	646	748266	5.4 ± 3.9	376405	2.7 ± 2.1

*People re-entering the field of view, after exiting it for longer than 10 seconds, are considered as a new (unique) person

validated by expert annotators that checked the consistency of person-face groups, person identity within a video and across videos, and an individual visual inspection of each of the annotated attributes. For preventing the analytics to focus on very small (far from signage) people, who are likely to not have OTS, and to simplify the annotation process, we define a region in some scenarios where people are omitted, and thus not annotated. We refer to these regions as ignore area, shown as a white shading in Fig. 8. Estimations within the ignore areas are also omitted. The annotations maintain the identity of each person throughout the same video, even if the person exits and re-enters into the field

Table 4 Main characteristics of the systems (S) used in the experiments

Short name	S1	S2	S3	S4
CPU	Intel i5	AMD R.5	Intel Atom x7	ARM
GPU	GTX1660Ti	GTX1660Ti	Intel Gen9 HD	NV. Pascal
Memory [GB]	32	32	8	8
Storage [GB]	512	512	64	32
Consump. [W]	450	450	18	7.5
NVIDIA inf.	✓	✓		✓
OpenVINO inf.	✓		✓	

inf. inference, *NV.* NVIDIA

of view, and even across videos. However, for the purpose of AVA benchmarking, when an actor exits and re-enters into the field of view of a camera within the same video after more than 10 seconds, we consider the same actor to have a new identity. Each video includes a range between 11 and 158 *unique* people. The dataset annotation includes a total of 785 unique people, and over a million annotated bounding boxes. Most of the people present in the dataset have OTS and roughly 10% of them looked directly at the camera. The main characteristics of the dataset are summarized in Table 3.

4 Benchmark—results and discussion

4.1 Experimental setup

We evaluate the performance of the algorithms on four systems that enable on-the-edge AVA processing. The systems' properties are summarized in Table 4. The algorithms are executed in both GPU and CPU (one core), separately. GPU inference is used for systems with an integrated NVIDIA GPU. CPU inference is used for all systems, and it can be native (i.e., without optimization) or optimized using OpenVINO. The use of OpenVINO optimization depends on the system and the algorithm. Systems must be equipped with an Intel processor, and algorithms must be compatible with the Open-VINO optimization. For the baseline algorithms (A1–6), all algorithms but A4 are Open-VINO compatible.

We define as real-time processing the capability of a system-algorithm pair to complete the analytics for I_t before the new data frame I_{t+1} is available. When a system–algorithm pair does not achieve real-time processing (e.g., input data are at 30 fps and the processing speed is at 1 fps), one can reduce the frame rate and/or resolution of the videos. We decide to maintain the resolution of the videos and intentionally reduce the

Table 5 Input frame rate for each localization algorithm and system when using CPU (GPU) inference

Algorithm	System 1	System 2	System 3	System 4
A1	30 (30)	3 (30)	3	(3)
A2	30 (30)	3 (30)	3	(3)
A3	30 (30)	3 (30)	3	(3)
A4	30 (30)	1 (30)	1	(1)
A5-6	30 (30)	3 (30)	3	(3)

The selected frame rate allows the algorithm–system combination to achieve (near) real-time processing speed

frame rate (i.e., drop frames) of the input videos to ensure that every system-algorithm pair performs (near) real time. Table 5 shows the input frame rate of the videos that we use for all experiments for every system-algorithm pair. For System 1, all algorithms run with 30 fps videos with both GPU and CPU inference. For System 2, all algorithms run with 30 fps videos with GPU and 3 fps videos with CPU, except for A4 with CPU that runs with 1 fps videos. For Systems 3 and 4, all algorithms run with 3 fps videos, except for A4 that runs with 1 fps videos.

Most of the results shown next are represented by box plots. The horizontal line within the box shows the median; the lower and upper edges of the box are the 25-percentile and 75-percentile; and, the bottom and top edges show the minimum and maximum values.

We provide an online evaluation tool that allows one to effortlessly assess the performance of AVA algorithms in the proposed dataset. Further information regarding the requested data format and use of the evaluation tool is available at the project website.

4.2 Localization

Figure 10 shows the evaluation results of A1–2 on face localization, and A3–4 on person localization. Regarding the detectors, A2 obtains a higher precision but lower recall than A1. This indicates that A1 generates a larger amount of false positives than A2. However, the median results for F1-score show that A1 outperforms A2 by a small amount. Regarding the trackers, A3–4 obtain comparable results across the systems. For instance, in System 1 with GPU, A1 obtains a F1-score 0.17 higher than A2, and A3 and A4 obtain a median F1-score of 0.80 and 0.81, respectively. When considering the distance between people–signage, results show that all algorithms are able to localize a larger amount of people (i.e., higher recall) when people are closer to the camera. For instance in System 1 with GPU, A1 obtains a median recall of 0.94/0.68 for closer/farther faces; and A3–4 trackers obtain a median recall of 0.91 for closer people, and 0.55 and 0.63 for farther people, respectively. When analyzing the performance measures as a function of the occlusion levels, results indicate that the recall drops when faces/bodies are partially or heavily occluded. For instance in System 1 with GPU, while the median recall for *non-occluded* people and algorithms A1, A3, and A4 is above 0.8; the median recall drops to values below 0.6 for partial occlusions and below 0.5 for heavy occlusions.
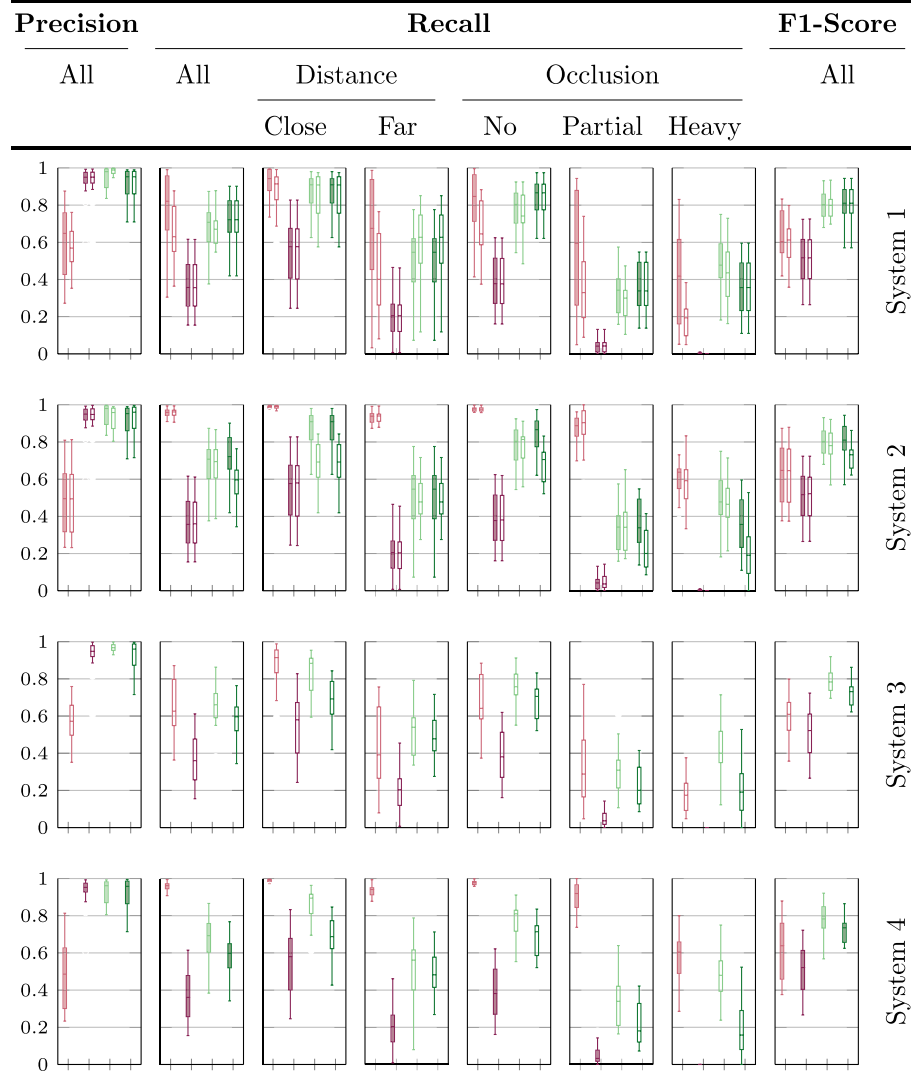
Fig. 10 Localization results in terms of precision, recall, recall for different person–signage distances, recall for different occlusion (occ.) levels, and F1-score for Algorithm 1 (A1: red box), Algorithm 2 (A2: dark red box), Algorithm 3 (A3: light green box), and Algorithm 4 (A4: dark green box). When compatible with the system, the results are reported with inference in GPU (filled boxes) and CPU (non-filled boxes). Note that above 90% of the people are not occluded. Results indicate that trackers (A3–4) perform better than detectors (A1–2), and that the localization performance substantially worsen when people are far or occluded

4.3 Counting

Figure 11 shows the counting evaluation in terms of MOE, MPE, and COE. Regarding mean opportunity error (MOE), A1–2 (i.e., detectors) have errors up to 10 people for some videos considering all systems. Detector obtains a large range of errors, indicating a non-uniform performance throughout different videos of the dataset. On the contrary, A3–4 (i.e., trackers) have a smaller median error as well as a smaller range of errors, which are always under 4 people. As observed in localization, algorithms commit fewer errors with people who are closer to the camera/signage, than with people who are farther. Algorithms that consider the body of people, instead of their

Fig. 11 People counting results in terms of mean opportunity error (MOE), MOE for different person–signage distances, mean people error (MPE), and cumulative opportunity error (COE) for Algorithm 1 (A1: red box), Algorithm 2 (A2: dark red box), Algorithm 3 (A3: light green box), and Algorithm 4 (A4: dark green box). When compatible with the system, the results are reported with inference in GPU (filled boxes) and CPU (non-filled boxes). The actual cumulative number of people with OTS across all videos is 646. Results indicate that trackers (A3–4) perform better than detectors (A1–2) for instantaneous (MOE) and, especially, for cumulative (COE) counting where the error obtained by trackers is at least an order of magnitude smaller than that obtained by detectors. COE results suggest that processing a smaller number of frames (e.g., Systems 3 and 4) can benefit the performance for cumulative counting

faces (i.e., A3–4) obtain a lower mean people error (MPE). This is expected, as these tracking algorithms have no skill in determining the OTS of people; thus, when considering all people regardless of their OTS, a lower error is obtained. Face detection algorithms (A1–2) obtain median MPE above 2 people, whereas body person tracking algorithms (A3–4) obtain median MPE below 2 people. Similar performance is obtained across systems and inference units (i.e., GPU vs CPU) except for A1, which has a smaller error range when executed in CPU.

When the task is to count the cumulative number of unique people with OTS, cumulative opportunity error (COE) indicates that tracking algorithms obtain more accurate count than detection algorithms. With System 1, while the tracking algorithm A3–4 obtains a median COE under 1.02 and 4.20, respectively; detection

Fig. 12 People counting results in terms of temporal cumulative opportunity error (TCOE) for Algorithm 1 (red line), Algorithm 2 (dark red line), Algorithm 3 (light green line), and Algorithm 4 (dark green line). The shaded area indicates the standard deviation of the error, and the line the mean error. The top chart indicates the actual number of people for each segment duration as reference. Note that *y*-axes are at different scales and *x*-axes are in logarithmic scale. When compatible with the system, the results are reported with inference in GPU (solid lines) and CPU (dashed lines). KEY: #, number of. Trackers (A3–4) obtain lower TCOE than detectors (A1–2). The lowest TCOE is obtained when a reduced number of frames is used as input (e.g., with Systems 3 and 4, or Systems 2 with CPU)

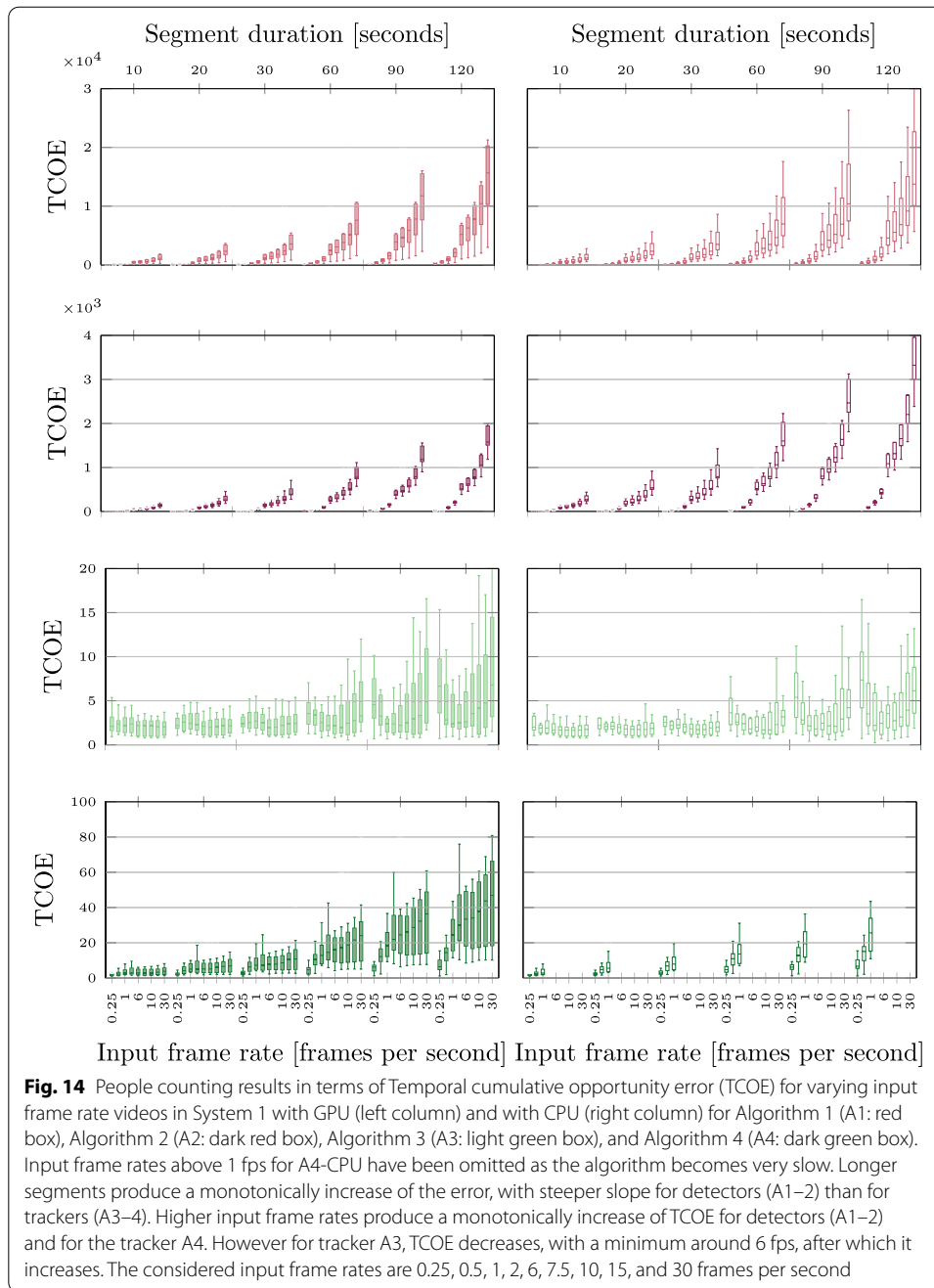algorithms A1–2 obtain a median COE of at least two orders of magnitude higher. In this case, algorithms that are using the GPU for inferring obtain in general higher error than when using CPU. This is due to the fact that algorithms that process more frames (i.e., GPU) are more prone to overcount the cumulative number of people than when processing fewer frames (i.e., CPU). Algorithms using CPU inference use

Fig. 13 Error vs input frame rate. People counting results with varying input video frame rate videos in terms of mean opportunity error (MOE), mean people error (MPE), and cumulative opportunity error (COE) for Algorithm 1 (red line), Algorithm 2 (dark red line), Algorithm 3 (light green line), and Algorithm 4 (dark green line). Note that the *x*-axes are in logarithmic scale and that the *y*-axes are at different scales. We separate COE for detectors (A1–2) and trackers (A3–4) for easing the visualization. The shaded area indicates the standard deviation of the error, and the line the mean error. Results obtained with System 1 with GPU

as input videos with lower frame rate than when using GPU inference (e.g., System 2 uses 3 fps videos with CPU inference and 30 fps videos with GPU inference), thus the overcount is likely to be reduced as the number of processed frames is reduced. This effect can also be seen in the results with System 1, where CPU and GPU use the same video frame rate, therefore, obtain very similar results regardless of the inference type. When the task is to estimate the cumulative number of people over a specific segment of time, temporal cumulative opportunity error (TCOE) results show that the error increases monotonically when the duration of the segment (D) increases (Fig. 12). Also, it can be observed that detectors (A1–2) obtain several orders of magnitude higher TCOE than trackers (A3–4). With System 1 GPU, trackers obtain TCOE of 1.5-19 (A3) and 6-70 (A4). The most accurate algorithm for this task is A3, which can estimate the cumulative number of people with OTS for 10 (120)-s segment video with a median TCOE of ± 2.20 (± 18.81), with System 1 and GPU.

4.4 Error vs input frame rate

Results indicate that the input frame rate of the videos has an important effect on the performance of the algorithms. Thus, we analyze the trade-off between the error obtained by the algorithms and the input frame rate of the videos. This analysis is divided into two experiments. In the first experiment, we intentionally modify the input frame rate of the videos to $\gamma = \{0.25, 0.5, 1, 2, 6, 7.5, 10, 15, 30\}$ fps, and then, we compare the performance of the localization algorithms in terms of MOE, MPE, and COE. Figure 13 shows the results. For MOE and MPE, in the first row, the errors are mostly flat. This indicates that those performance measures are not affected by the input frame rate of the videos. This is expected as these two measures are performed on a frame-by-frame

Fig. 14 People counting results in terms of Temporal cumulative opportunity error (TCOE) for varying input frame rate videos in System 1 with GPU (left column) and with CPU (right column) for Algorithm 1 (A1: red box), Algorithm 2 (A2: dark red box), Algorithm 3 (A3: light green box), and Algorithm 4 (A4: dark green box). Input frame rates above 1 fps for A4-CPU have been omitted as the algorithm becomes very slow. Longer segments produce a monotonically increase of the error, with steeper slope for detectors (A1–2) than for trackers (A3–4). Higher input frame rates produce a monotonically increase of TCOE for detectors (A1–2) and for the tracker A4. However for tracker A3, TCOE decreases, with a minimum around 6 fps, after which it increases. The considered input frame rates are 0.25, 0.5, 1, 2, 6, 7.5, 10, 15, and 30 frames per second

basis. Therefore, when reducing the frame rate of the input videos, the number of frames that the algorithms compute is reduced but this reduction does not affect to the performance of the algorithms, as non-seen frames are used neither for inference nor evaluation. However, when computing cumulative counts across time, the performance of the algorithms can be affected. In fact, the second row of Fig. 13, we observe that COE varies for A3–4 when the input frame rate of the videos change. COE monotonically increases when the frame rate increases for detection algorithms (A1–2). The performance obtained by A3 does not change substantially for different input frame rate, and
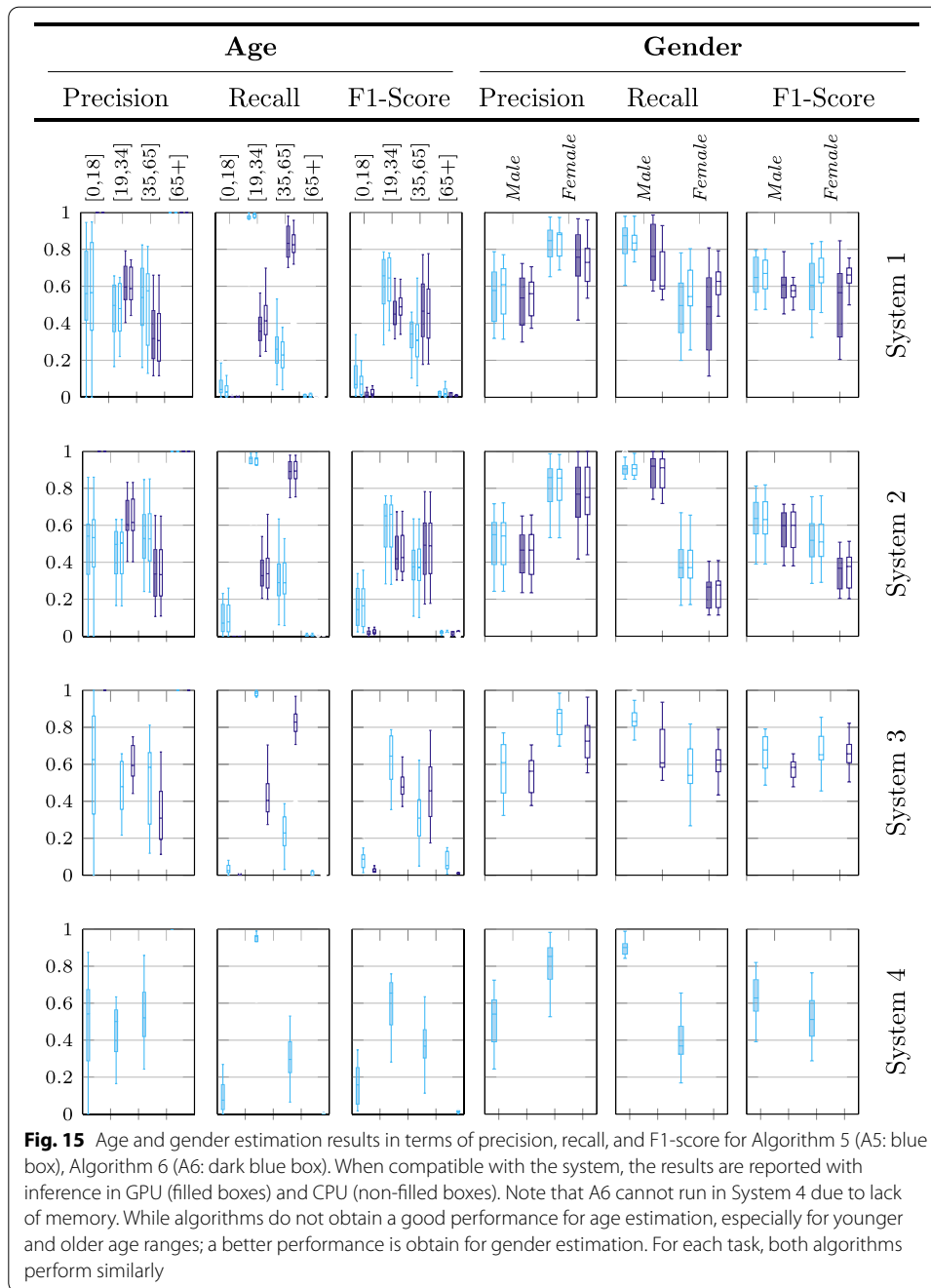
its minimum occurs around 6 fps. Surprisingly, COE monotonically increases for A4 when the frame rate increases. However, when considering absolute values, the error is considerably lower than the one obtained by detection algorithms. This indicates that higher frame rates do not necessarily ensure higher counting performance. In the second experiment, we apply the same frame rate reduction than in the previous experiment, and we compute TCOE for different segment duration of $D = \{10, 20, 30, 60, 90, 120\}\, \gamma$ frames (i.e., 10, 20, 30, 60, 90, 120 seconds). The results are in Fig. 14. Results in the first two rows indicate that TCOE monotonically increases for detection algorithms (A1–2) when the frame rate increases, and also when the duration of the segment increases. As mentioned before, this occurs as detectors have no skill for counting the cumulative number of people. When the segment increases its duration, the overcount accumulates further, thus TCOE also increases. Unlike detectors, trackers (A3–4), shown in the last two rows of Fig. 14, obtain a more stable TCOE for any segment duration and frame rate. Segments with larger duration produce a slight increment in the TCOE. Note that as A4 cannot be optimized using OpenVINO for CPU computation, we limit the input frame rate of the videos up to 1 fps as higher frame-rate videos make the algorithm very slow. In this experiment, A3 is the most accurate baseline algorithm under comparison. Visual and detailed per-video results obtained by each algorithm are available at the project website.

4.5 Age and gender estimation

Age and gender estimation algorithms require as input a crop with the person's face. We employ the face detector that obtains the highest recall (A1) as detector. To fairly compare the age and gender estimation algorithms and disregard the localization performance, we compute the age for correctly detected (i.e., true positive) faces only. The results with A5–6 for age estimation are shown in Fig. 15 (first three columns), and for gender estimation in Fig. 15 (last three columns).

Age estimation results indicate that this task is challenging as the highest median F1-score obtained is below 0.7. While the highest results are obtained for classes [19,34] and [35,65] with median F1-score between 0.3 and 0.7, even lower F1-scores are obtained for [0,18] and [65+] classes where algorithms obtain median F1-score under 0.1. This suggests that the baselines algorithms, A5-6, are not skilled in determining the age of younger and older people, where both algorithms obtain a very low recall for these classes. Similar results are obtained regardless of the system and inference type (i.e., CPU and GPU).

Gender estimation results indicate that A5-6 algorithms obtain a similar F1-score performance for both classes (male, female). For the male class, both algorithms obtain a precision between 0.4 and 0.8 with a recall between 0.6 and 1.0. The opposite behavior happens for the class female, obtaining higher precision than recall. For this task, results indicate that all systems and inference types behave similarly. Note that A6 is not able to run in System 4 due to memory limitations.
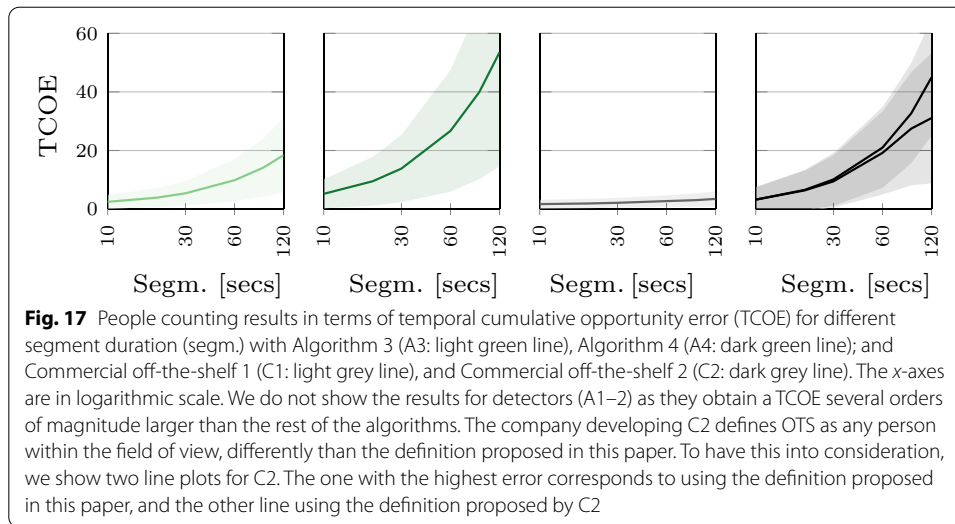
Fig. 15 Age and gender estimation results in terms of precision, recall, and F1-score for Algorithm 5 (A5: blue box), Algorithm 6 (A6: dark blue box). When compatible with the system, the results are reported with inference in GPU (filled boxes) and CPU (non-filled boxes). Note that A6 cannot run in System 4 due to lack of memory. While algorithms do not obtain a good performance for age estimation, especially for younger and older age ranges; a better performance is obtain for gender estimation. For each task, both algorithms perform similarly

4.6 Commercial solutions

As we did not have access to the source codes, the commercial solutions (C1-2) are executed in two external systems equipped with an Intel i7 CPU. To preserve the integrity of the complete dataset, until submission time, we only shared a subset of the dataset with the creators of C1-2. Thus, for this comparison, we use a subset of the dataset that is composed of the following videos Airport-1, Airport-2, Mall-3, Mall-4, Pedestrian-2, and Pedestrian-3. As a reference, we show next the results obtained by the baseline algorithms (A1–6) with System 1 GPU in the same subset of videos.
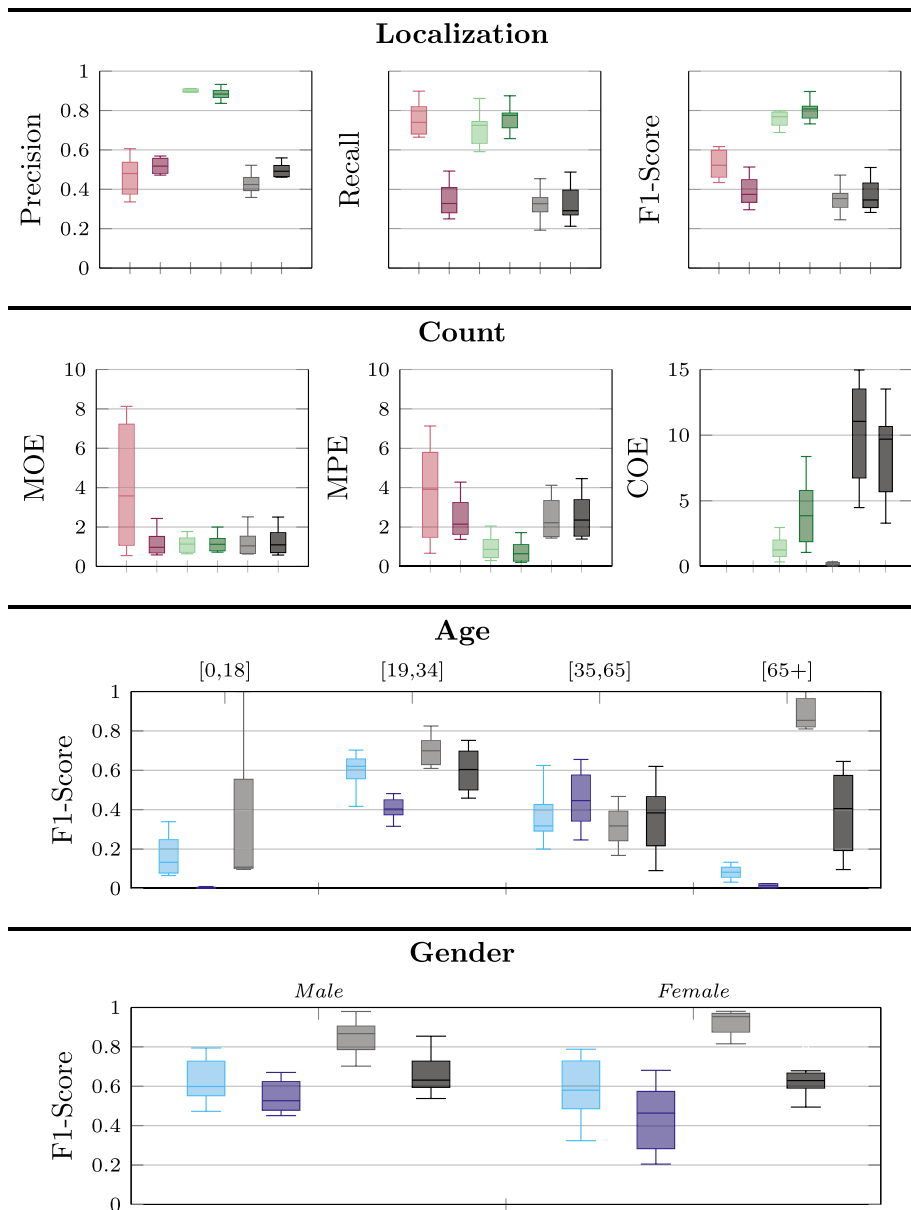
Fig. 16 Anonymous video analytics comparison of Algorithm 1 (A1: red box), Algorithm 2 (A2: dark red box), Algorithm 3 (A3: light green box), Algorithm 4 (A4: dark green box); Algorithm 5 (A5: blue box), Algorithm 6 (A6: dark blue box); and commercial solutions Commercial off-the-shelf 1 (C1: light grey box), and Commercial off-the-shelf 2 (C2: dark grey box), accounting for localization in terms of precision, recall, and F1-score; people counting in terms of mean opportunity error (MOE), Mean People Error (MPE), and cumulative opportunity error (COE); and age and gender estimation in terms of F1-score. All algorithms have been executed with the same video inputs (at 30 frames per second). A1–6 are run in System 1 with GPU, and C1-2 are run in CPU. Note that A1–2 obtain a COE several orders of magnitude larger than the rest of the algorithms; thus, these results are out of the chart ranges for easing the visualization. The company developing C2 defines OTS as any person within the field of view, differently than the definition proposed in this paper. To have this into consideration, we show two bar plots for C2. The one with the highest error corresponds to using the definition proposed in this paper, and the other one using the definition proposed by C2 (i.e., CPE, Eq. 8, is reported)

Fig. 17 People counting results in terms of temporal cumulative opportunity error (TCOE) for different segment duration (segm.) with Algorithm 3 (A3: light green line), Algorithm 4 (A4: dark green line); and Commercial off-the-shelf 1 (C1: light grey line), and Commercial off-the-shelf 2 (C2: dark grey line). The *x*-axes are in logarithmic scale. We do not show the results for detectors (A1–2) as they obtain a TCOE several orders of magnitude larger than the rest of the algorithms. The company developing C2 defines OTS as any person within the field of view, differently than the definition proposed in this paper. To have this into consideration, we show two line plots for C2. The one with the highest error corresponds to using the definition proposed in this paper, and the other line using the definition proposed by C2

The results of this experiment are reported in Fig. 16. For localization, A1–2 and C1-2 are evaluated for face localization, and A3–4 are evaluated for person localization. Results show that both commercial solutions (C1-2) perform similarly than A2 for all performance measures, while A1 outperforms other face detectors obtaining a recall higher than 0.4 and similar precision. Person detectors (A3–4) obtain higher detection performance than face detectors with F1-score around 0.8 and precision over 0.85. Count results show that all baselines and the commercial solutions obtain very similar MOE with median errors around 1 and always under 2.5, except for A1 that obtains a median MOE of 3.5 with values up to 8. Commercial solutions obtain similar MPE errors than A2 with a median error of around 2 people. A3–4 obtain lower MPE indicating that these algorithms obtain a lower error (below 1 in its median) when counting instantaneous people regardless of their OTS. A1 obtains a higher error than other algorithms with median MPE around 4. Regarding COE, the commercial solution C1 obtains the lowest error with a median of only 0.20 and a small range of errors. This indicates that the algorithm can count the cumulative number of people in each complete video with an error below ± 1 person. The authors of C2 have a different definition of OTS than the one described in this paper. C2 define OTS as any person within the field of view, regardless of their face visibility. Therefore, we also compute, for C2 only, the CPE, where all people visible on the field of view of the video are considered to have OTS. The results are on the last COE bar.
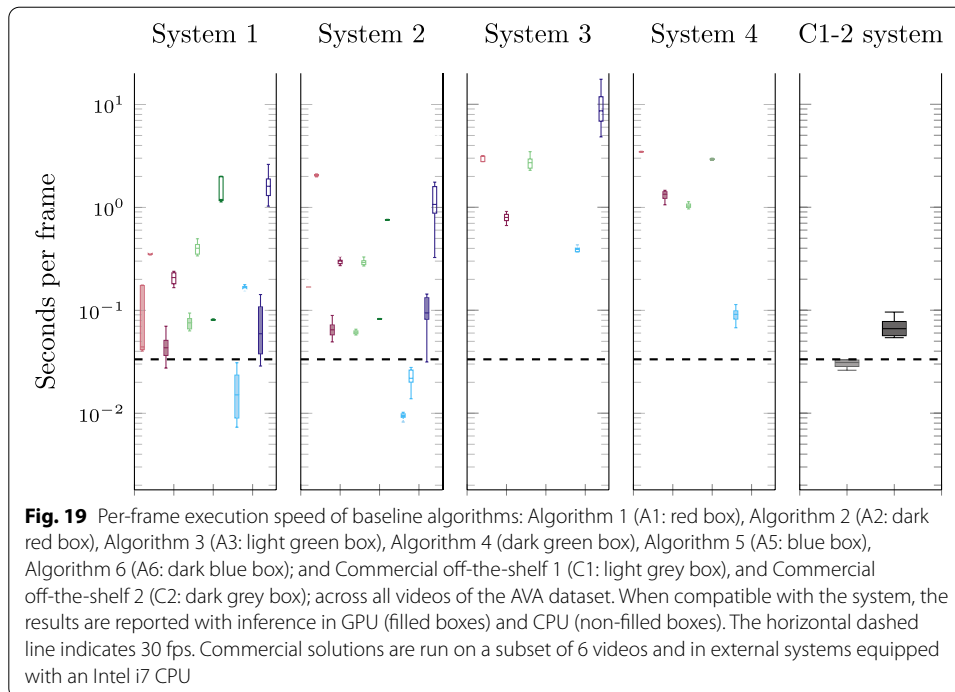
When we consider the ability of the algorithms for counting the cumulative number of people for different segment duration, algorithms obtain an average TCOE between 1 and 50 (Fig. 17). C1 is again the algorithm that obtains the lowest error for all segment duration, achieving TCOE between 2 and 4.

In addition to showing the results using the proposed performance measures, we show the raw estimated and annotated counts for a set of selected videos in Fig. 18. We can observe that for instantaneous count most of the algorithms often estimate the number
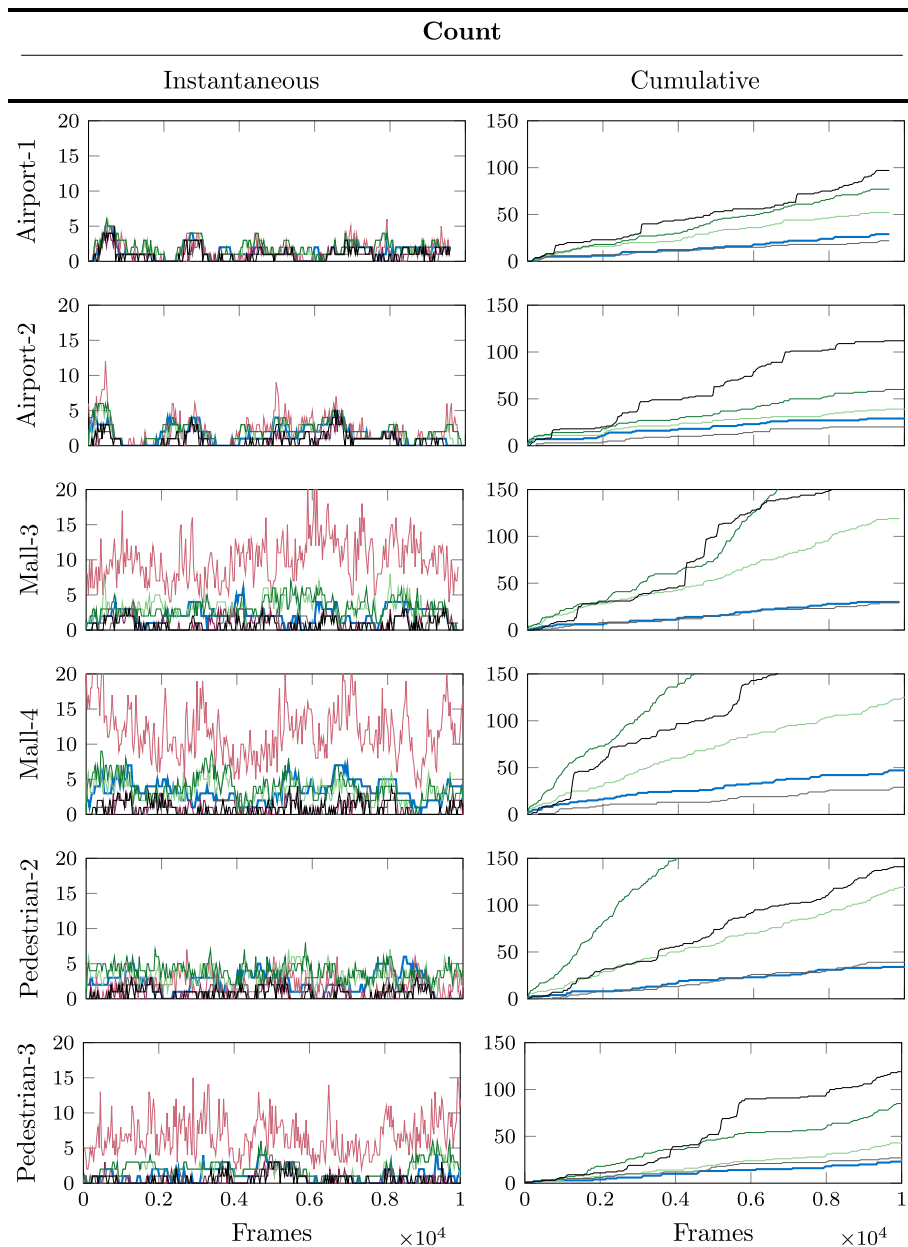
Fig. 18 Per-frame count of instantaneous and cumulative people with OTS performance in selected videos for Algorithm 1 (A1: red line), Algorithm 2 (A2: dark red line), Algorithm 3 (A3: light green line), Algorithm 4 (A4: dark green line); and Commercial off-the-shelf 1 (C1: light grey line), and Commercial off-the-shelf 2 (C2: dark grey line). The annotation count is shown in (blue line). All algorithms have been executed with the same video inputs (at 30 frames per second). A1–4 are run in System 1 with GPU. C1-2 are run in CPU. For easing the visualization, the cumulative count of A1–2 are not shown (they are several orders of magnitude larger than the rest of the algorithms); and, only one sample per second (instead of 30 samples per second) is shown. Note that the y-axes are limited to 20 (instantaneous) and 150 (cumulative), and that x-axes are limited to 10000 frames

of people in the surrounding of the annotated value. An exception here is A1, which often generates multiple false positives, thus overestimating the actual number of people with OTS for a given instant. Regarding the cumulative count (second column of Fig. 18), and as done for COE, we show the estimated cumulative count of unique people

Fig. 19 Per-frame execution speed of baseline algorithms: Algorithm 1 (A1: red box), Algorithm 2 (A2: dark red box), Algorithm 3 (A3: light green box), Algorithm 4 (dark green box), Algorithm 5 (A5: blue box), Algorithm 6 (A6: dark blue box); and Commercial off-the-shelf 1 (C1: light grey box), and Commercial off-the-shelf 2 (C2: dark grey box); across all videos of the AVA dataset. When compatible with the system, the results are reported with inference in GPU (filled boxes) and CPU (non-filled boxes). The horizontal dashed line indicates 30 fps. Commercial solutions are run on a subset of 6 videos and in external systems equipped with an Intel i7 CPU

with OTS from the initial frame. The charts indicate that the most accurate algorithms in this task are C1 and A3; and that the cumulative count is a challenging task and even state-of-the-art trackers (i.e., A4) and commercial solutions (i.e., C2) might drift from the actual number of people.

The detailed per-video results are available at the project website.

4.7 Execution speed

Figure 19 shows the statistics of the execution speed of the baseline algorithms in the four systems with both CPU and GPU, and of the commercial solutions. Considering the baseline algorithms (A1–6), the fastest algorithms are the age and gender estimation algorithms (A5–6), followed by the detection algorithms (A1–2). The tracking algorithms (A3–4) are the slowest ones. All six baseline algorithms run close to real-time (i.e., 30 fps), indicated in the charts by the black dashed line, when considering Systems 1-2 with GPU. In general, the chart confirms that GPU inference is the fastest, followed by CPU inference with OpenVINO optimization, and the slowest is non-optimized CPU inference. For instance, A1 with OpenVINO optimization increases its average execution time by 3.6 times (from 0.10 s per frame, in GPU, to 0.36 s per frame, in CPU) in System 1. In the same system, A4 (without OpenVINO optimization) increases its average execution time by 19 times (from 0.08 s per frame, in GPU, to 1.52 s per frame, in CPU).

5 Conclusion

We proposed an open-source benchmark for the evaluation of anonymous video analytics (AVA) for audience measurement and released to the research community the first fully annotated dataset that enables the evaluation of AVA algorithms. Using this

benchmark, we conducted a set of experiments with eight baseline algorithms and two commercial off-the-shelf solutions for the tasks of localization, counting, age, and gender estimation. All the tasks are evaluated in four systems, with CPU and GPU. Results showed that trackers perform better than detectors in all scenarios, that localization algorithms should improve when objects are far/occluded (Figs. 10 and 11), and that the use of higher input frame rate videos do not ensure a better performance. Further efforts should be made towards the design of holistic tracking solutions that synergistically consider body and face to account for robustness and attention attributes. The performance of age estimation algorithms is limited and results suggest that the performance in estimating the age for younger, [0,18], and older people, [65+], is degraded. This might be due to an existing bias in the datasets used for training age estimation algorithms. Based on the outcomes of the benchmark, future work could explore the design of improved AVA algorithms for age estimation and cumulative count based on multiple-object tracking, as well as for attention to evaluate the audience responsiveness to an advertisement.

Abbreviations
AVA: Anonymous video analytics; COCO: Common objects in context; COE: Cumulative opportunity error; CPE: Cumulative person error; CPU: Central processing unit; FN: False negative; FP: False positive; GPU: Graphics processing unit; IOU: Intersection over union; MOE: Mean opportunity error; MPE: Mean people error; OTS: Opportunity to see; TCOE: Temporal cumulative opportunity error; TP: True positive.

Declarations

Competing interests
Intel Corporation[5] provided funding and hardware to support the design and development of the research presented in this manuscript.

References
1. K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP J. Adv. in Signal Process. **2008**, 1–10 (2008)
2. H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2547–2554 (2013)
3. E. Bondi, L. Seidenari, A.D. Bagdanov, A. Del Bimbo, Real-time people counting from depth imagery of crowded environments. In: Proc. IEEE Conf. Advanced Video and Signal Based Surveillance, pp. 337–342 (2014)
4. L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: A survey. Int. J. Comput. Vis. **128**(2), 261–318 (2020)

[Intel] is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel's products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

5.   A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
6.   T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, Zitnick, C.L.: Microsoft COCO, , Common Objects in Context. Proc. Eur. Conf. Comput. Vis. , 740–755 (2014)
7.   L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler, MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint: 1504.01942 (2015)
8.   A. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler, MOT16: A benchmark for multi-object tracking. arXiv preprint: 1603.00831 (2016)
9.   P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, L. Leal-Taixé, MOT20: A benchmark for multi object tracking in crowded scenes. arXiv preprint: 2003.09003 (2020)
10. M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder et al., The sixth visual object tracking VOT2018 challenge results. Proc. Eur. Conf. Comput. Vis. Workshops , 3–53 (2018)
11. M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Cehovin Zajc, O. Drbohlav, A. Lukezic, A. Berg et al., The seventh visual object tracking VOT2019 challenge results. Proc. IEEE Int. Conf. Comput. Vis. Workshops (2019)
12. M. Gou, S. Karanam, W. Liu, O. Camps, R.J. Radke, DukeMTMC4ReID: A large-scale multi-camera person re-identification dataset. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops , 1425–1434 (2017)
13. Robust Vision Challenge. http://www.robustvision.net/. Accessed: 29th September 2020
14. J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou, RetinaFace: Single-shot multi-level face localisation in the wild. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 5203–5212 (2020)
15. K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (2016). https://doi.org/10.1109/LSP.2016.2603342
16. N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric. Proc. IEEE Int. Conf. Image Process. , 3645–3649 (2017)
17. Z. Wang, L. Zheng, Y. Liu, S. Wang, Towards real-time multi-object tracking. arXiv preprint: 1909.12605 (2019)
18. S. Ayoubi, FaceLib. GitHub (2020)
19. R. Rothe, R. Timofte, L.V. Gool, DEX, , Deep expectation of apparent age from a single image. Proc. IEEE Int. Conf. Comput. Vis. Workshops (2015)
20. S. Yang, P. Luo, C.C. Loy, X. Tang, WIDER FACE: A face detection benchmark. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 5525–5533 (2016)
21. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement. arXiv preprint: 1804.02767 (2018)
22. A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking. Proc. IEEE Int. Conf. Image Process. , 3464–3468 (2016)
23. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection. In: Proc. IEEE Conf. Pattern Recognit., pp. 2117–2125 (2017)
24. N. Ma, X. Zhang, H.-T. Zheng, J. Sun, ShuffleNet V2: Practical guidelines for efficient CNN architecture design. Proc. Eur. Conf. Comput. Vis. , 122–138 (2018)
25. Z. Zhang, Y. Song, H. Qi, Age progression/regression by conditional adversarial autoencoder. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 5810–5818 (2017)
26. B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov, D. Kalinin: Openvinotoolkit/cvat: V1.3.0. https://github.com/openvinotoolkit/cvat

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.