Virtual Reality (2024) 28:44
https://doi.org/10.1007/s10055-023-00937-2

ORIGINAL ARTICLE

# A real-time wearable AR system for egocentric vision on the edge

**Iason Karakostas[1] · Aikaterini Valakou[2] · Despoina Gavgiotaki[2] · Zinovia Stefanidi[2] · Ioannis Pastaltzidis[1] · Grigorios Tsipouridis[1] · Nikolaos Kilis[1] · Konstantinos C. Apostolakis[2] · Stavroula Ntoa[2] · Nikolaos Dimitriou[1] · George Margetis[2] · Dimitrios Tzovaras[1]**

Received: 12 January 2023 / Accepted: 21 December 2023 / Published online: 19 February 2024


**Abstract**
Real-time performance is critical for Augmented Reality (AR) systems as it directly affects responsiveness and enables the timely rendering of virtual content superimposed on real scenes. In this context, we present the DARLENE wearable AR system, analysing its specifications, overall architecture and core algorithmic components. DARLENE comprises AR glasses and a wearable computing node responsible for several time-critical computation tasks. These include computer vision modules developed for the real-time analysis of dynamic scenes supporting functionalities for instance segmentation, tracking and pose estimation. To meet real-time requirements in limited resources, concrete algorithmic adaptations and design choices are introduced. The proposed system further supports real-time video streaming and interconnection with external IoT nodes. To improve user experience, a novel approach is proposed for the adaptive rendering of AR content by considering the user's stress level, the context of use and the environmental conditions for adjusting the level of presented information towards enhancing their situational awareness. Through extensive experiments, we evaluate the performance of individual components and end-to-end pipelines. As the proposed system targets time-critical security applications where it can be used to enhance police officers' situational awareness, further experimental results involving end users are reported with respect to overall user experience, workload and evaluation of situational awareness.

**Keywords** Intelligent user interfaces · Augmented reality · Artificial intelligence · Situational awareness

✉ Iason Karakostas
iason@iti.gr

Aikaterini Valakou
valakou@ics.forth.gr

Despoina Gavgiotaki
gavgiotaki@ics.forth.gr

Zinovia Stefanidi
zinastef@ics.forth.gr

Ioannis Pastaltzidis
gpastal@iti.gr

Grigorios Tsipouridis
tsipurid@iti.gr

Nikolaos Kilis
nikolaoskk@iti.gr

Konstantinos C. Apostolakis
kapostol@ics.forth.gr

Stavroula Ntoa
stant@ics.forth.gr

Nikolaos Dimitriou
nikdim@iti.gr

George Margetis
gmarget@ics.forth.gr

Dimitrios Tzovaras
dimitrios.tzovaras@iti.gr

[1] Information Technologies Institute, Centre for Research and Technology Hellas, GR-57001 Thessaloniki, Greece

[2] Foundation for Research and Technology Hellas, Institute of Computer Science, GR-70013 Heraklion, Greece

## 1 Introduction

Augmented reality (AR) is drastically reshaping everyday tasks in all aspects of human activity, including work, education and entertainment. There are several contributing factors that have fostered this progress, e.g. advances in hardware miniaturizing AR devices, while improving the computational capabilities of embedded systems, progress in computer vision (CV) that permits the registration of virtual content to the real-world and next-generation communications that enable connectivity with fog and cloud computation nodes. One critical requirement for all AR systems is real-time performance so that the rendered content is aligned to the real scene, with further difficulties arising in the case of wearable AR devices. In such a case highly dynamic scenes are usual since the user's viewpoint constantly changes with motion. Even minor latency results in a misalignment between the background scene and the rendered virtual content which can be quite problematic, particularly for time-critical use cases where visual artefacts can aggravate the user's stress level and hinder task execution.

AR systems have a long history with the foundations of today's AR technology putting emphasis on real-time interaction with the user. Recently, researchers have explored the potential of mobile edge computing and 5 G for AR (Siriwardhana et al. 2021), whereas a bundle of work focuses on artificial intelligence (AI) methodologies for AR that can detect particular objects of interest and superimpose relevant information on an AR device (Hoque et al. 2021; Zhang et al. 2022). There are several research works that study the potential of AR for specific applications in quite heterogeneous domains, including healthcare (Buettner et al. 2020) and education (Alvarez-Marin and Velazquez-Iturbide 2022; Pellas et al. 2019).

Computer vision can be of great assistance in police tasks, e.g. frameworks that detect specific actions such as petty crimes (Dimitriou et al. 2017), or security-oriented applications in autonomous systems that require methods achieving real-time performance on wearable computing systems (Tsiktsiris et al. 2020). In this paper, we present a real-time wearable AR system for law enforcement officers that embeds several artificial intelligence (AI) modules aiming to enhance officers' perception and improve Situational Awareness (SA). The proposed system is part of the DARLENE framework as described by Apostolakis et al. (2021) and is motivated by the requirements and needs of the police officers in the filed, concretely, their needs for enhanced Situational Awareness and rapid Decision Making. The system aims to facilitate police operations through the use of a distributed computation continuum along the edge, fog and cloud while supporting several AI functionalities and using AR to visualize and communicate analysis results to the user. In this respect, the focus of our work is on minimal latency and real-time processing as well as improved and user adaptive visualization using wearable edge nodes.

In this paper, we present the architecture, design and functionality of the DARLENE wearable node presenting several novelties to support complex computational pipelines while minimizing execution times and overall latency. In this regard, the contributions of our work can be summarized as:

- an interoperable architecture for a real-time wearable AR system with edge computing capabilities,
- lightweight instance segmentation method that can achieve real-time speed performance on the embedded system while approximating the accuracy performance of more computational costly methods,
- a framework for complementary CV tasks that satisfies running time restrictions by reducing computation requirements and exploiting synergies between different modules,
- a methodology for adaptive AR visualization that can automatically adjust the rendered content according to user status,
- extensive experiments to evaluate both system performance and user acceptance.

## 2 Related work

### 2.1 Real-time computer vision methods

**Instance segmentation:** Methods in this field, typically based on deep architectures, usually follow two strategies to segment an image: (a) "Top–down" or "Proposal-based" (Bolya et al. 2019) and (b) "Bottom–up" or "Proposal-free" (Gao et al. 2019). We focus on the faster first category, where bounding boxes are initially found for every instance, followed by an estimation of the precise shape of that instance. The "Top–down" approaches are divided into single-stage and two-stage methods, based on the underlying detection framework. Single-stage approaches do not require proposal generation or pooling operations and employ dense predictions of bounding boxes and instance masks, leading to inferior, but real-time performance for embedded systems. Bolya et al. (2019) presented a fast fully convolutional single-stage method (YOLACT). The method breaks up instance segmentation into two parallel tasks: (1) generation of spatially coherent prototype masks via convolutional layers, and (2) prediction of mask coefficients per instance-mask, able to achieve real-time performance although not on embedded systems. Lee and Park (2020) proposed CenterMask that

balances segmentation speed and model accuracy being an anchor-free, single-stage instance segmentation method. Bolya et al. (2020) introduced YOLACT++ based on YOLACT that in contrast to the base method employs deformable convolutions into the backbone network leading to performance gains. An interesting approach for Instance Segmentation was introduced by Jocher et al. (2020), incrementing the well-established YOLO object detection method (Redmon et al. 2016). This method added a variant of Spatial Pyramid Network (He et al. 2015) and the Path Aggregation Network (Liu et al. 2018) was modified to incorporate the BottleNeckCSP (Wang et al. 2020) resulting in a faster method than the previous versions of YOLO.

**Pose estimation:** The goal of pose estimation is to predict a person's position and/or orientation. This is usually achieved by predicting specific keypoints, such as the wrists, ankles, head. There are two approaches to this problem, namely Bottom-Up (Cao et al. 2019), in which all the body parts are first predicted and then they are reorganized and grouped together to their corresponding persons and Top–Down (Fang et al. 2017). In the latter, body keypoints are calculated iteratively and a human detector is required for multi-person pose estimation. The basic pipeline is as follows: (1) detect the people in an image with a human detector, (2) crop the regions where a person was detected, (3) resize the cropped images to match the model's input resolution and (4) predict the keypoints. He et al. (2017) introduced an interesting method that is performing instance segmentation and calculates the keypoints of the people at the same time. HRNet (Sun et al. 2019) contrary to most Top-Down pose estimation techniques maintains high-resolution representations during the whole process achieving competitive performance and more importantly; in theory, it achieves these results with less computational power compared to Pose-ResNet (Xiao et al. 2018). In Liu et al. (2021) the Polarized Self-Attention (PSA) block was introduced, using HRNet-48 as backbone achieving promising results. State-of-the-art methods also employ Residual nets (Xiao et al. 2018) and transformers (Mao et al. 2021). Xu et al. (2022) introduced a method that utilizes a Vision Transformer (Dosovitskiy et al. 2021) as its backbone, showcasing the ability of transformer architectures to be used in complex CV tasks. This method outperforms every existing method in the literature in terms of accuracy; however, it is not suitable for real-time applications that rely on embedded processing units.

**Object tracking:** Object target tracking can be divided to two categories, single object tracking (SOT) methods that follow a single object or multiple object tracking (MOT) methods that aim to track all the targets in a scene. SOT methods can be further divided to state-of-the-art in terms of performance, deep convolutional methods (Fu et al. 2021) and less computational heavy correlation filter-based methods (Henriques et al. 2014) that can achieve high-speed performance even on embedded devices. The MOT task could be confronted by employing multiple SOT method instances, one per target, although this is not possible for deep convolutional methods since they usually struggle to perform real time for a single target on embedded systems. Towards addressing the problem of MOT, the most common approach is to employ an Object Detector and a mechanism that can assign and update uniquely the IDs of each detection per frame (Zhang et al. 2021). The main issue with the vast majority of these methods is that they struggle to perform real time on limited computational resources, rendering the usage of MOT methods alongside an instance segmentation method impossible.

Most of the described computer vision task methods can perform real time on the edge, albeit, this speed performance is usually achievable when solely one of these tasks is executed. In Sect. 4, a unified framework for these tasks, able to perform real time on wearable devices, such as the Wearable Edge Computing Node (WECN) of DARLENE is presented, that does not consume all of the available computational resources of the system, leaving space for visualization and other tasks.

### 2.2 Context-aware adaptation for AR assistive systems

Context-awareness refers to a characteristic integrated into a piece of software that triggers it to adapt its functionality in order to remain usable whenever changes are detected in the context of use (e.g. the functional logic dictated by the current state of the environment or situation under which the software operates) (Abowd et al. 1999). A variety of approaches have been proposed in the literature regarding context modelling and reasoning frameworks (Pradeep and Krishnamoorthy 2019), both of which are incorporated in the design of various applications, and, closely related to the current work, adaptation of (graphical) user interfaces in user-facing applications.

Relevant works in context-aware user interface (UI) adaptation can be distinguished into model-based (Hussain et al. 2018) and optimisation-based approaches (Oulasvirta et al. 2020), with various methodologies and frameworks having been proposed in both topics. With regard to AR systems, context-awareness has been a subject of study in a variety of application domains (e.g. entertainment, manufacturing, education, medical and others), but, due to the complexity involved, the bibliography is inherently limited in terms of universal implementation frameworks for such applications (Yigitbas et al. 2020). The kinds of ubiquitous AR interfaces that can deliver a continuous experience adapting to the user's current situation underpin the concept of Pervasive AR (PAR) (Grubert et al. 2017), effectively defined as a

super-set of AR applications with the capacity to recognize and react to changes in the context of their use. Additional pathways towards PAR are opened when considering the integration of Internet of Things (IoT) sensors' modalities into a common implementation architecture, hence combining information from the real (vision), virtual (UI) and ambient (IoT) world (Kim et al. 2021). Particularly with respect to wearable PAR, the capacity to complement the visual modality afforded by AR with biosignals tracking the wearer's physiological state **(a)** has been shown to boost productivity when the adaptive system is meant to operate in an assistive capacity (ElKomy et al. 2017). Furthermore, the effective orchestration of digital information superimposed onto the real-world environment through intelligent level-of-detail (LOD) management **(b)** plays an important role in avoiding unwanted information overload, particularly when targeting "glanceable" visualization systems (Daskalogrigorakis et al. 2021; Köppel et al. 2021; Lavoie et al. 2021).

Therefore, in the present work we describe a wearable PAR system targeted at law enforcement, and which takes into account best practices (a, b) with regard to being usable in a real Law Enforcement Agent (LEA) operational context, while simultaneously applying the principles of the Human-Centered Design (HCD) framework, by facilitating the active participation of real end users in both the conceptualization and realization of the final solution. Hence, we outline the interplay between both AR and IoT components within a comprehensive architectural model that further incorporates machine intelligence algorithms towards elevating LEAs' situational awareness through continuous re-adaptation of the presented visuals on the wearable device. The specifics are covered in the next section.

## 3 Methods and architectural overview

### 3.1 Human-centered artificial intelligence

Technologies targeting the domain of law enforcement should aim to bridge the gap between understanding the real challenges LEAs face in their day-to-day operations, and systems designers' eagerness to build 'science-fiction' systems that might favor decoration over usability, in order to be useful to the intended end users (Silvennoinen and Jokinen 2016). For intelligent user interfaces and AI-enabled systems, the stakes are higher, going beyond usability and usefulness to issues such as fairness, explainability and ethics, demanding Human-Centered Design (HCD) methodological approaches which put humans actively in the loop, thus fostering Human-Centered AI. To this end, we fully involved target end users and stakeholders in the requirements, design and evaluation phases, so as to address real needs and requirements regarding the target use cases of police patrol and tactical units and ensure that the developed prototype successfully meets them, but also to guarantee that the decision-making algorithm is designed in an optimal way to support LEAs' situational awareness and actually achieves this, in a way that is not 'black-box' for its users (Margetis et al. 2021).

Specifically, user requirements were elicited through co-creation workshops, involving 30 target end users and police stakeholders (e.g. police officers, tacticians and trainers). Workshop activities included listing desired functionality, voting proposed functionality and analysing the top-voted functionality. A thematic analysis approach was used (Braun and Clarke 2021) for mapping workshop outcomes to 44 functional and non-functional system requirements. The elicited requirements were validated and refined employing a user survey, in which 60 end users were engaged. The deployed survey asked participants to indicate the importance of various functionalities and propose additional desirable functionalities and resulted in a total of 64 functional and 27 non-functional requirements. Further elaboration and validation of requirements were carried out through system demonstrations and training events of the target users, resulting in a final list of 78 functional and 38 non-functional requirements.

Informed by the identified requirements, Graphical User Interface (GUI) design prototypes were developed in an iterative approach, following well-established design principles, AR heuristics and research findings for enhancing GUI legibility in AR glasses (Endsley et al. 2017; Syberfeldt et al. 2017). Furthermore, three User eXperience (UX) experts were iteratively involved in the process, by assessing the developed prototypes. Overall, 10 UI widgets were designed representing the core functionality of the AR glasses, as it was identified through the co-creation workshops as crucial for enhancing LEAs' SA. Each component encompassed three Levels of Detail (LoD) to accommodate adaptivity according to user status.

End users were also involved in the design of the adaptation decision-maker, which is described in Sect. 5.2 and more specifically in identifying the priority of the component types according to the policing task at hand. To this end, feedback acquired from the co-creation workshop was used, as well as responses to a subsequent questionnaire. The questionnaire, which was handed out to 10 LEAs, described the component types and asked respondents to order them in decreasing level of importance and usefulness, according to their relevance for two policing tasks, namely incident resolution and patrolling.

An expert-based evaluation of the developed prototype was also conducted, with the participation of 5 UX and 5 LEA experts, aimed at acquiring feedback on the developed visualizations, but also at assessing the implemented decision-making with regard to which components are selected

**Table 1** Summary of functional requirements as these were elicited through co-creation workshops, organized in coarse categories

| Category | Details |
|---|---|
| Suspect and foe identification | Suspicious persons should be highlighted as suspects, whereas persons considered to be dangerous should be highlighted as foes. |
| Allies identification | The system should clearly identify allies, providing information about their health status as well |
| Inured persons identification | Injured persons and victims of malicious acts should be clearly highlighted, providing information about their health status as well |
| Object identification | The system should highlight dangerous objects (e.g. a gun) as well as suspicious objects (e.g. an abandoned suitcase at the airport) |
| Colour coding | Different types of information should be displayed with different colour codes (e.g. according to threat level) |
| Skeleton diagrams | For persons highlighted in the officer's view, skeleton diagrams should also be provided |
| Location information | Location information regarding identified persons (e.g. foes, allies, victims) should be provided in a map |
| Directions | Navigation directions should be provided in addition to a map |
| Communication with the Command and Control (C &C) | Direct messages from the C &C should be provided when appropriate. |
| Summative information | Summative information about the number of foes, allies, and victims should be provided. |
| Information prioritization | The system should support information prioritization, displaying the most crucial information at the highest priority |

to be visualized in the AR glasses, the selected LoD, as well as their placement in the agent's field of view. The resulting quantitative and qualitative findings were analyzed, leading to valuable insights for improving the developed GUI widgets and the decision-making (Stefanidi et al. 2022). The final system developed was evaluated with end users following an XR simulation approach, aimed at assessing SA, mental workload and overall UX prior to the field trials (see Sect. 6.2).

### 3.2 Co-creation workshops

To actively involve end users in the requirements elicitation phase, co-creation workshops were organized with the participation of 30 end users. In brief, co-creation is "a creative process that taps into the collective potential of groups to generate insights and innovation. Specifically, it is a process, in which teams of diverse stakeholders are actively engaged in a mutually empowering act of collective creativity with experiential and practical outcomes" (Rill and Hämäläinen 2018). One facilitator was responsible for coordinating discussions, and one more facilitator was assisting in administering the workshop activities.

All workshops had the same structure of activities. In particular, each workshop was structured in five (5) main sections: (1) Discussion of the aims and objectives of the workshop, and provision of general instructions to participants; (2) Warm-up activity, acting as an ice-breaker to stimulate discussions within the group. (3) Presentation of DARLENE and its use cases, to familiarize participants with its objectives; (4) Co-creation activities for each use case, structured in three parts, namely identification of functionality that participants would like the DARLENE technologies to have, voting on the most appealing functionality, and analysis of top-voted functionality; (5) Workshop evaluation.

The workshop outcomes were analysed manually, following a combination of deductive and inductive coding, involving two researchers (Fereday and Muir-Cochrane 2006). In particular, one code for each one of the functionalities identified in the desired functionality activity was created, following the deductive coding approach. Then, the researchers examined the data regarding functional requirements in order to assign one of the predefined codes. In the cases when the need for assigning a new code was identified, this was added to the set of codes, and all responses were re-examined, following the inductive coding approach. The examination of responses and code assignments was carried out by two individual researchers. The outcomes of the two individual analyses were compared, following a consensus-building approach to address inconsistencies in the codes assigned.

As a result, a total of 44 initial requirements were collected, describing functional and non-functional aspects of the DARLENE system. More specifically, the identified requirements are summarised in Table 1, classified in high-level categories.

Non-functional requirements pertained to the security of the device, unobtrusiveness and user-friendliness of the system, accuracy of detections, as well as compliance with the legislation.

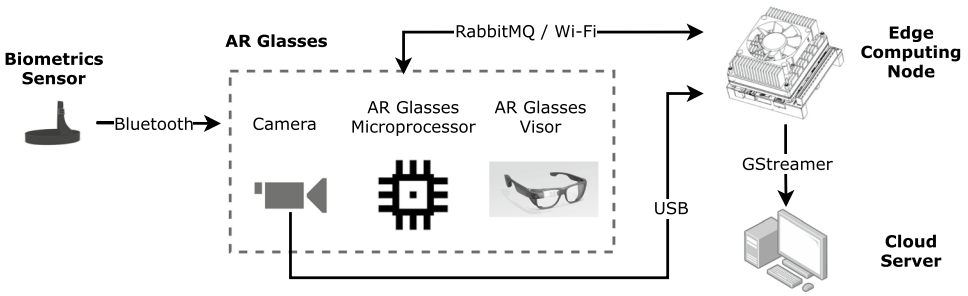



**Fig. 1** Communications between the WECN components. The wearable computing node receives video data from the AR glasses camera. The bio-signals caught by the wearable sensor are collected by the AR glasses processor and transmitted to the wearable computing node. The wearable computing node either processes the video data and provides results to the AR glasses processor for visualization or requests computational power from more powerful computing nodes

### 3.3 Architectural overview

The proposed system includes a real-time video processing pipeline, implemented on a Wearable Edge Computing Node (WECN), consisting of various sub-components. The first one being the AR visors that is in fact a standalone subsystem within the proposed wearable system, consisting of the AR visors and a microprocessor. The core processing unit of the wearable is a Jetson AGX Xavier.[1] A camera that is physically located on the AR visors, is connected via a USB connection directly to the main processing device ensuring minimal latency. The information between the main processing unit of WECN and the AR microprocessor is handled by a message broker (RabbitMQ) over a local private Wi-Fi connection. Additionally a smart band[2] transmitting bio-signals via Bluetooth is connected to the AR glasses microprocessor. Figure 1 displays the components that form DARLENE's WECN and the connections between. WECN's main processing unit is powered by a 4 S Li-Po battery and the AR glasses microprocessor and visors by a 18650 battery. The system power autonomy can exceed one hour of continuous usage both for the main processing unit and the AR glasses.

The video stream is processed by the WECN processing unit, and results are fed back to the AR Glasses structured in JavaScript Object Notation[3] (JSON) data format, so as to be rendered, enhancing the officer's perception. The processing pipeline consists of computer vision modules that run in real time on the wearable edge computing node. At the time of publishing, Instance Segmentation, 2D Pose Estimation and 2D Target Tracking are integrated on DARLENE WECN device. In order to achieve a high processing frame rate, combined with satisfactory computer vision results, instance segmentation, being computationally heavier, is employed for 1 frame per second and the segmentation masks are propagated by the output of the 2D Target Tracking. The output of the target tracking for the human target detections, is also employed by the Pose Estimation module in the intermediate frames in order to calculate the desired human skeleton. Additionally, whenever the computational load is high, affecting performance, the WECN has the ability to forward the video data to more powerful cloud computing nodes, with a GStreamer service.[4]

DARLENE takes into account several parameters before presenting any information to the AR glasses. In specific, it considers the officer's stress level, analysing the ECG signals gathered by a smart band, as well as other multilateral parameters, such as the context of use (e.g. if the LEA is trying to neutralize a perpetrator or just patrolling), the environmental conditions (e.g. whether the LEA is located in a crowded area or not) and the agent's experience. To that end, all computer vision results are filtered by an adaptation component before projection, to decide which information will eventually be displayed on the LEAs' AR glasses, where and in which LoD, as described in Sect. 5.2.

## 4 DARLENE real-time computer vision functions

In this section the computer vision modules that are incorporated in the DARLENE Computer Vision Analysis Framework (CVAF), as well as the functionality of the framework are presented. More specifically, in the following subsections the Segmentation (S), Pose estimation (P) and Tracking (T) modules are analyzed. Furthermore, the way that they

[1] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/

[2] https://plux.info/biosignalsplux-wearables/274-cardioban-820202404.html.

[3] https://www.json.org/json-en.html.

[4] https://gstreamer.freedesktop.org/

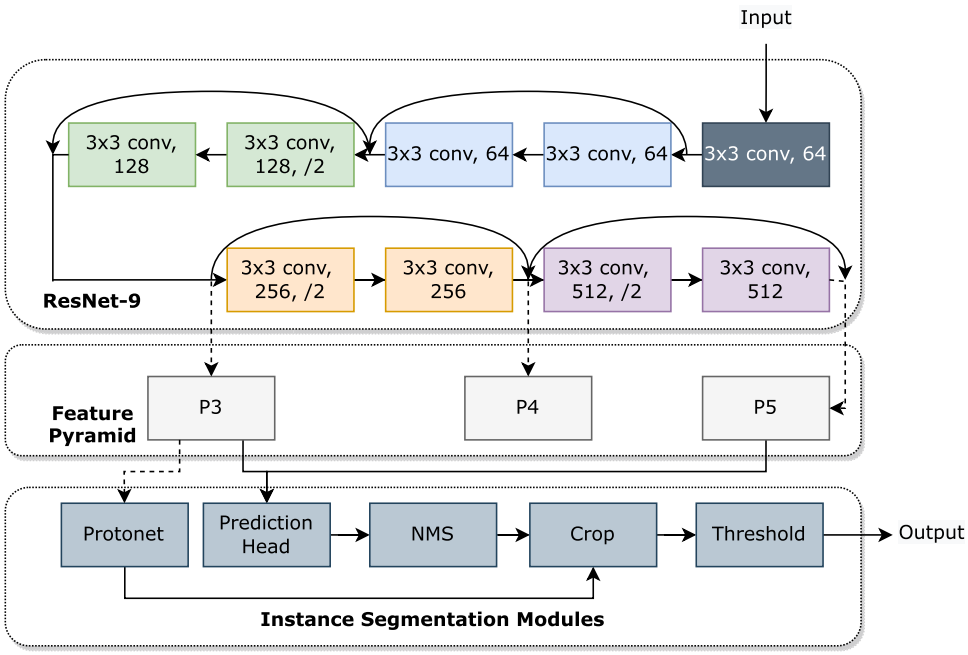



**Fig. 2** Overall instance segmentation architecture

collaboratively produce the computer vision analysis output is explained.

## 4.1 Instance segmentation

This section analyzes the Segmentation module (S) of the DARLENE CVAF. The instance segmentation module is crucial for the system functionality since it allows for correct annotation of the object of interest in terms of position on the AR glasses projectors. Based on the near real-time method (measured on the embedded platform of DARLENE) of Bolya et al. (2019), we developed a lightweight backbone network towards improving the inference speed for the instance segmentation task. The backbone network is inspired by the well established Residual Networks introduced by He et al. (2016). The proposed lightweight feature extraction network, ResNet9, is composed by 9 convolutional layers with residual connections. The rest of the instance segmentation network is composed by the Feature Pyramid Network, the Protonet and the Prediction Head of the baseline method. The overall architecture is depicted in Fig. 2.

### 4.1.1 Training of the segmentation module

For the training procedure of the Segmentation module, we constructed a YOLACT-18 network utilizing a pre-trained ResNet18[5] on the ImageNet dataset (Deng et al. 2009) as the feature extraction backbone by removing its last 2 layers. We trained YOLACT-18 for the instance segmentation task by exploiting the well-established MS-COCO dataset introduced by Lin et al. (2014). In a similar manner, the YOLACT-9 network was constructed, utilizing the proposed feature extraction network ResNet9.

The training of the YOLACT-9 network was carried out in multiple steps. In the first step we focused only on the training of the feature extraction backbone. The aim of this step was to force the ResNet9 backbone, to produce features similar to the heavier ResNet18 architecture. The lightweight segmentation method was initialized with the trained weights of YOLACT-18 apart from the backbone weights. In each training step, let $I$ be the input training image. The output from the feature extraction layers will be $X_9 = r_9(I)$ and $X_{18} = r_{18}(I)$ for the ResNet-9 and ResNet-18, respectively. The extracted features are then passed as input to the rest of the segmentation network $S$, and for each case, the final output will be given by:

$$O_9 = S(r_9(I)), \tag{1}$$

$$O_{18} = S(r_{18}(I)). \tag{2}$$

For each image training batch the training loss will be given by the mean squared error:

$$\mathcal{L}_9 = \frac{1}{N}\sum_{i=0}^{N}(O_9 - O_{18})^2, \tag{3}$$

[5] https://pytorch.org/vision/stable/models.html.

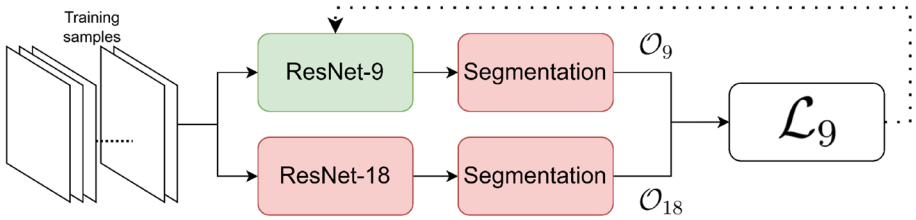



**Fig. 3** Training procedure of the proposed ResNet-9 backbone. Red blocks indicate frozen layers during the training procedure

where $N$ is the batch size. Since all of the YOLACT-9 network weights were kept frozen apart from the ResNet9 during the training, in fact $\mathcal{L}_9$ was only employed for the training of the proposed backbone. The training procedure is depicted in Fig. 3

As a second step, the YOLACT-9 network was fine tuned for the desired classes for DARLENE. Towards constructing the training dataset we used the classes person, handbag, suitcase, backpack and knife from the publicly available dataset MS-COCO. The knife class was augmented by images obtained for the DARLENE project needs, manually annotated images from the MGD dataset (Lim et al. 2021) and Open Image Dataset (Kuznetsova et al. 2020). The firearm class was constructed by manually annotating firearm images from the MGD dataset and images obtained for the DARLENE project. Additionally, specific video sequences were recorded in CERTH premises depicting scenes of interest for the DARLENE project, as suspicious/unattended objects, people attacking with knife/firearm, etc. The custom dataset (Kilis et al. 2023) is split in train and test set and is publicly available upon request.

As loss function, the Cross-Entropy function between the assembled masks and the ground truth is employed, in addition to the standard Mean Squared Error and Cross-Entropy losses for the regression of bounding box and classification for the semantic class, respectively.

#### 4.1.2 Artificial occlusions towards robust training.

A main goal for the DARLENE WECN CVAF is to detect humans under occlusions. In this context, the training dataset was augmented by constructing artificial human-to-human occlusions, similar to the augmentation method proposed by Ghiasi et al. (2021). An occluder was picked from an image with a human and pasted, possibly at a later image, so that it occludes another picked human (occluded).

In a given training set $\mathcal{S}$, let $\mathbf{X}_i \in \mathbb{R}^{H\times W\times C}$ be a training image of $\mathcal{S}$ containing $N$ targets, where $H$, $W$ are the height and width of the image, respectively, and $C$ the number of colour channels. For each target, a vector $\mathbf{r}_n$, $n = 0, \ldots, N-1$, $N \in \mathbb{Z}^+$, is available containing the top left and bottom right pixel coordinates of the groundtruth bounding box and a segmentation mask matrix $\mathbf{M}_n \in \mathbb{R}^{H\times W}$ with each cell equal to 1 for the pixels where the target lies and 0 otherwise. By exploiting $\mathbf{r}_n$ and $\mathbf{M}_n$, the image $\mathbf{T}_i^n \subset \mathbf{X}_i$ can be extracted, being of size equal to the bounding box of the target, $H_n \times W_n$.

The augmentation technique exploits two input images $\mathbf{X}_i$, $\mathbf{X}_j$, $i \neq j$. When $N > 0$ for both images, a random $r_i^n$ is selected from the first image in order to occlude another randomly selected target $r_j^n$. The first step towards the new image, is to calculate the translation vector $\mathbf{v}$, required in order to translate the central pixel of $r_i^n$ on top of the central pixel of $r_j^n$, altered by a random regularizing factor $\lambda$, proportional to the pixel dimensions of the occluded target:

$$\mathbf{v} = \begin{bmatrix} \lambda(\mathbf{t}_{f_0} - \mathbf{t}_{b_0}) \\ \lambda(\mathbf{t}_{f_1} - \mathbf{t}_{b_1}) \end{bmatrix}. \tag{4}$$

The regularization factor is applied towards creating multiple levels of occlusions. The translation vector is exploited towards translating $\mathbf{T}_i^n$ on top of $\mathbf{T}_j^n$ and with the aid of $\mathbf{M}_i^n$ only the pixel values that actually belong to the occluder object are translated.

#### 4.1.3 Compressing instance segmentation information

In order to compress the information generated (object classes, bounding boxes and segmentation masks) from our architecture, we calculate the 2D polygon surrounding each detected mask, as depicted in Fig. 4. For communicating with the other system components only the polygons need to be transferred and not mask images. This abstraction allows for a big reduction on the data that need to be transferred between various system components, since each polygon is a simple list containing the $(x, y)$ pixel coordinates that enclose a mask found.

### 4.2 Pose estimation

This section describes the Pose estimation module (P) of the DARLENE CVAF. For this task we propose a solution, based on Xiao et al. (2018). It has a ResNet-8 backbone and its input is a cropped image in accordance with the bounding box predicted by the instance segmentation module, S or propagated by T. Compared to module S, the ResNet-8 backbone does not have the last 512-channel convolutional

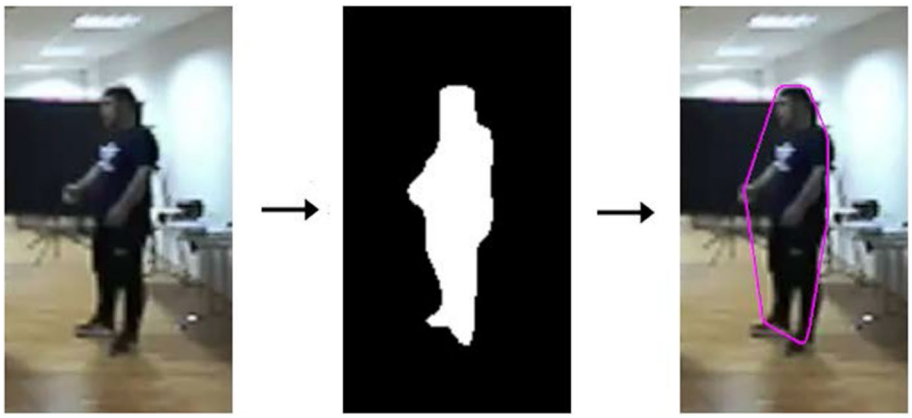


**Fig. 4** Generation of segmentation polygon from the produced segmentation map. This abstraction allows for huge data transfer reduction since a small amount of $(x, y)$ pixel pairs are sufficient to describe the target position

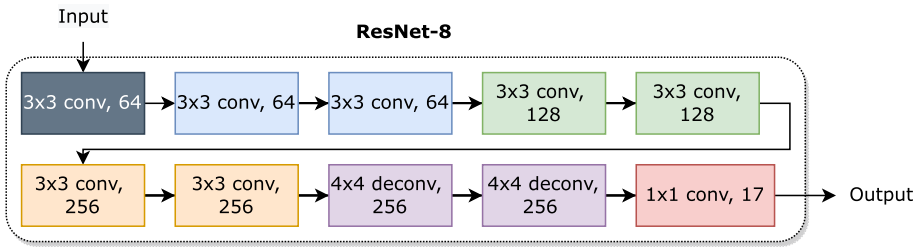



**Fig. 5** DARLENE Pose estimation model

basic block. Two deconvolutions are placed after the backbone for up-sampling purposes.

Its architecture is visualized in Fig. 5. In the training phase, Gaussian heatmaps are generated as the targets for each joint, each one having its highest value at the corresponding joint location; these heatmaps have size of $64 \times 48$ pixels, height and width, respectively. The Gaussian heatmaps for the joints can be generated as in Eq. (5), where $\mathbf{p_i}$ is the Gaussian heatmap, $(\mathbf{x_i}, \mathbf{y_i})$ is the $\mathbf{i}$-th joint location, $(\mathbf{x}, \mathbf{y})$ is the pixel location, and $\sigma$ is a constant spatial variance.

$$\mathbf{p_i}(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi\sigma^2} \exp \frac{-(\mathbf{x} - \mathbf{x_i})^2 + (\mathbf{y} - \mathbf{y_i})^2}{2\sigma^2}. \tag{5}$$

Mean-squared error (MSE) is used as the loss function, and in this context it can be written as follows:

$$\mathcal{L}_{\mathbf{mse}} = \frac{1}{N} \sum_{i=1}^{N} \parallel \hat{\mathbf{p}}_i - \mathbf{p}_i \parallel_2^2 \tag{6}$$

In the above equation $\hat{\mathbf{p}}$ represents the joint heatmap prediction from the model, $\mathbf{p}$ is the ground truth heatmap for the same joint, and $N$ is the number of joints. The pose module has 17 channels as their output, corresponding to the number of joints, and each channel is the predicted heatmap for the specific joint.

### 4.3 Visual object tracking

The DARLENE CVAF exploits a correlation filter-based single-object tracking method (T). This method expects as input during initialization ($t = 0$) a Region-Of-Interest (ROI) containing the desired target, produced in this case by the S module of the framework. In order to track an object we construct $\mathbf{x}$, a vectorized descriptor of the ROI having length equal to $N = H_t \times W_t$. The ROI contains a slightly bigger area of the image with respect to the detected bounding box. The image descriptor can be just the grayscale pixel values, Histogram of Oriented Gradients (HOG) or the output of a convolutional architecture (e.g. ResNet-9). In order to create target tracking examples, all the possible permutations of $\mathbf{x}$ are exploited by utilizing a permutation matrix that shifts the descriptor vector one element at a time. By applying this permutation matrix N times, the $\mathbf{X}$ matrix is constructed containing all the possible permutations of $\mathbf{x}$. The training goal is to find a tracking filter $\mathbf{w}$ that can regress the unaltered representation of the target to the peak of a Gaussian distribution $\mathbf{y}$ and the most distorted ones to zero. The filter is calculated by optimizing a Ridge-Regression problem (Henriques et al. 2014):

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2, \tag{7}$$

where $\lambda$ is a regularization parameter. The solution to the above problem is given by:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix. In case of multi-channel descriptors then the datamatrix can be defined as the concatenation of per channel datamatrices $\mathbf{X}_i, i = 0, ..., C-1$, where $C$ the number of channels. It is easy to conclude that calculating the result of Eq. (8) in a per-frame basis would be a significantly heavy computational task. To overcome this difficulty, the circulant properties of matrix $\mathbf{X}$ are exploited according to $\mathbf{X} = \mathcal{F}^{\mathrm{H}} diag(\mathcal{F}\mathbf{x}_1)\mathcal{F}$, where $\mathcal{F}$ is the Discrete Fourier Transform and $.^{\mathrm{H}}$ the Hermitian transpose. Applying this to Eq. (8), in the Fourier domain:

$$\hat{\mathbf{w}}^* = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}, \tag{9}$$

where $\odot$ denotes element-wise operations, the division is element-wise, $\hat{\cdot}$ and $\hat{\cdot}^*$ denote the DFT transform and its complex-conjugate, respectively. By applying $\mathbf{w}$ to $\mathbf{x}$ the expected output, namely the response map $\mathbf{R}$, should be the desired Gaussian distribution $\mathbf{y}$:

$$\mathbf{R} = \mathcal{F}^{-1}(\hat{\mathbf{w}} \odot \hat{\mathbf{x}}). \tag{10}$$

Subsequently, in the following frames ($t > 0$), the feature descriptors of the area that previously contained the desired target $\mathbf{z}$ are extracted and the trained filter $\mathbf{w}$ is applied. Examining the obtained response map $\mathbf{R}$, the target translation can be obtained by the offset of the peak value from the expected center. After obtaining the updated target position, Eq. (9) can be recalculated, producing $\mathbf{w}_t$, and the tracking filter can be updated by a predefined learning rate $l$ as: $\mathbf{w} = (1-l)\hat{\mathbf{w}} + l\hat{\mathbf{w}}_t$.

The result of Eq. (10) can also be exploited in order to detect tracking failures, by examining the peak value, statistical characteristics, or by classifying the output as successful or not as in Karakostas et al. (2020); Ma et al. (2015); Li et al. (2016). In DARLENEs CVAF the maximum value is exploited as a measure of tracking quality as well as a metric for target re-identification, further explained in following Sect. 4.4.

## 4.4 Computer vision analysis framework

All of the previously described methods are combined in the Computer Vision Analysis Framework (CVAF), able to perform real-time on embedded devices such as the WECN of DARLENE. The desired goals of this framework are to produce a polygon containing the desired targets, the pose of human targets and maintain a tracking id for each detected object as long as it is in the camera Field-of-View (FoV). Towards achieving this, the segmentation (S) and tracking (T) modules work collaboratively alongside the pose estimation module (P).

The CVAF framework takes as input a video stream set to 25 frames per second (FPS). For the first frame the S module segments the scene and produces the desired polygons for visualization in the AR glasses. For the following frames, T is employed in order to update the polygon position of each tracked object. S is employed again after 24 frames, thus runs at 1 Hz. It is trivial to understand that with the initialization of the framework, the detected objects can be assigned with a unique identity (ID) number. For the subsequent outputs of S although, the necessity of re-assigning the correct IDs to the already tracked object arises as well as identifying newly detected objects. To address this issue, after each S output, two metrics are employed for the re-identification task. The first one is the Intersection-over-Union (IoU) of the detected box/polygon with the propagated box by T module. Given two bounding boxes $A$ and $B$, IoU ($d_1$) will be calculated by the area of overlap of the two boxes over the total area defined by the boxes, $d_1 = \frac{A \cap B}{A \cup B}$. The second metric is derived by the output of the response map of T, $d_2 = \max(\mathbf{R})$. By setting a hard-coded threshold $t_1$ and $t_2$ for each metric, if $d_1 > t_1$ and $d_2 > t_2$, it is assumed that the same ID for the detected object should be maintained. Otherwise a new ID is assigned. The threshold values have been selected after experiments on relevant video sequences. Alongside this procedure, P produces the pose estimation for the human targets (computed in a per-frame basis).

## 4.5 Streaming and communication

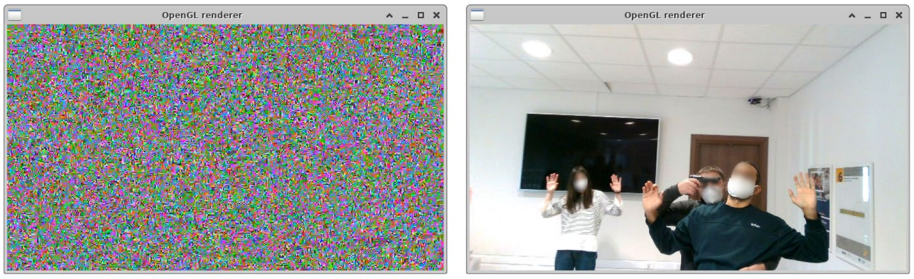
### 4.5.1 CVAF output and AR glasses communication

The CVAF produces instance segmentation, pose estimation and identity information regarding the objects of interest in the scene of operation. In order to visualize these results on the AR glasses, a JSON data structure is produced containing all of the vital information to the rest of the system modules. This file is transmitted by employing tools provided by the open-source message broker RabbitMQ.[6]

### 4.5.2 Video stream

In DARLENE ecosystem it is important to make the video stream from the AR glasses camera available for further analysis in cloud/edge computing nodes. The camera, is connected via USB cable to the wearable processing unit and when necessary, stream queues to specific edge or cloud computational nodes can be created. For the video streaming task GStreamer framework is deployed, exploiting its encrypted Real-time Transport Protocol (SRTP) capabilities. For data security reasons, apart from the private network that the system utilizes, the video stream is encrypted and can

[6] https://www.rabbitmq.com/




(a) Encrypted stream. (b) Decrypted stream.

**Fig. 6** Visualization of the encrypted stream **a** without decryption and **b** using the predefined decryption key. In the left case the video stream is distorted and unusable

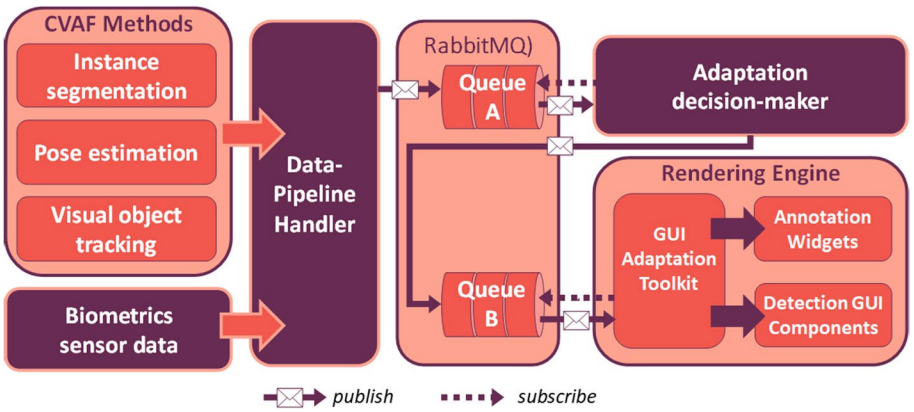



**Fig. 7** High-level overview of the DARLENE rendering and visualisation pipeline

only be decrypted by the desired client if the appropriate key is available, as illustrated in Fig. 6.

## 5 Pervasive AR WECN Visualisation

In this section the rendering and visualisation pipeline implemented on the WECN, with integrated support for on-the-fly context-aware UI adaptation is discussed. Specifically, the PAR assistive system is presented, which utilises inputs from the wearable biometrics sensors (IoT) and the information relayed by the CVAF, toward compositing a Heads-Up Display (HUD) comprised of dynamic glanceable graphical UI elements (widgets) designed to elevate the wearer's SA. An overview of the visualisation pipeline is presented in Fig. 7. Individual rendering components are described in detail in the following sub-chapters.

### 5.1 Data-pipeline handler

This component implements an aggregator of external information generated by the various inter-communicating functional blocks of the DARLENE architecture and then organises and broadcasts it in the form of a unified message structure that can be consumed by the rendering and interaction pipeline components. Its purpose is to obtain, associate and synchronize data containing various information such as stress predictions from the biometrics sensors, the CVAF JSON output, information regarding the detected people and objects existing in the scene, compiling it into a JSON data block representing a 'snapshot' of the current situational context in which the WECN is operating. This message is relayed to the adaptation decision-maker component for triggering the context-aware reasoning routines that will further regulate how the contained information (e.g. segmentation, pose estimation and annotation data) should best be displayed given the current physiological state of the WECN user. The communication supports an asynchronous, message-oriented protocol (e.g. AMQP) which is implemented using the RabbitMQ. The Data-Pipeline handler component is executed on the main processing unit of the wearable device (i.e. Jetson AGX).

### 5.2 Adaptation decision-maker

The adaptation decision-maker aims at providing context-aware adaptation of the GUI components that are being visualized on the LEAs' AR glasses, considering the parameters that affect their SA, such as the current context of use, their physical state (e.g. stressed or not) and expertise. By combining Ontology modeling and reasoning with Combinatorial Optimization, this module decides *what* information to present, *when* to present it, *where* to visualize it in the display - and *how*, taking into consideration contextual factors as well as placement constraints. The main objective of the

proposed approach is to optimize the SA associated with the displayed UI *at run-time*, while avoiding information overload and induced stress.

The adaptation decision-maker module consists of three inter-connected units:

- the **Ontology model**, which implements an Ontology that specifies the entities and relationships of the supported GUI components, including accompanying metadata (e.g. their dimensions), as well as relevant context information. It dynamically receives the current context from the messages of the Data-Pipeline handler and stores it in the Ontology.
- the **SA reasoner** provides an on-the-fly inference of the suitability of each GUI component, with respect to how much it enhances the user's SA, by calculating an SA score. The calculation of each component's score is based on information from the Ontology Model, in particular, the current context and modeled domain knowledge in the form of Ontology rules.
- the **UI optimizer**, which computes the optimal adaptation of the UI, given the modeling of the application domain. In particular, it determines the GUI components, their presentation and their position, for display by the Rendering Engine. This is based on information about (a) their SA score provided by the SA Reasoner, and (b) visualization and placement constraints, based on the current context and their size and shape, provided by the Ontology model.

In specific, in the Ontology Model of the decision-maker module, an ontological model has been defined, based on the user requirements obtained from the co-creation workshops, as described in 3.1. For the definition of the Ontology, relevant context factors are pertaining to the user's profile, state, activity and environment, following a similar approach to Margetis et al. (2019). Specifically, for the *DARLENE* case study, the activity is the current LEA operation (task), the environment includes information regarding its crowdedness, the profile includes the user expertise and the state captures the user's stress level. Furthermore, all the supported GUI components, their type of provided information and their LoDs, along with relevant metadata, such as their dimensions and their SA score, are also represented.

Regarding the SA Reasoner, a SA score for each GUI component in the Ontology is dynamically computed, depending on the current context. More specifically, based on the user's profile, state, activity and environment, modeled in the aforementioned Ontology, an Ontology Reasoner infers the SA score to assign to each GUI component, depending on its LoD and the type of information it provides. For this implementation, the Pellet reasoner was used.[7] Each time the context changes, the SA Reasoner recalculates the SA scores and propagates them to the UI Optimizer.

The UI Optimizer implements a Combinatorial Optimization problem, with the purpose of computing the optimal UI for the display of the user at run-time. This optimal UI is the one that maximizes the SA associated with the UI, based on the modeling of the application domain, while satisfying at the same time visualization and placement constraints. In particular, the UI Optimizer solves two distinct but interrelated problems at once, one of GUI component selection (content, design) and one of GUI component placement (layout). More specifically, on the one hand, it determines *what* information to present to the end user and *how*, which translates to the problem of selecting the appropriate GUI components and their LoD. On the other hand, it determines *where* to visualize them, and more specifically in which of the dynamically defined possible positions in the display. The solution of the optimization problem is sent to the Rendering Engine, responsible for visualizing the selected GUI components, instantiated with up-to-date content originating from the Data-Pipeline manager. The Adaptation decision-maker component is executed on the main processing unit of the wearable device (i.e. Jetson AGX).

A detailed analysis of the adaptation decision-maker module is provided in Stefanidi et al. (2022).

### 5.3 Rendering engine components

As previously mentioned in Sect. 3.2 Co-creation workshops, the DARLENE WECN utilises a pair of smart glasses as the main visual output terminal, based on an ARM mobile platform architecture running an Android-based operating system. The rendering engine is built in the Unity 3D graphics engine and implements the HUD functionality by means of a collection of dynamic widgets and UI components displaying the algorithmic outputs, that are automatically and selectively triggered and placed in view, so that they are composited to render the final augmented image. This image is then transparently layered on top of the real world by means of the smart glasses lenses display. Orchestration of the widgets and GUI elements is triggered internally by means of the GUI adaptation toolkit (GUIT) component. The rendering engine components are presented in more detail in the following paragraphs.

#### 5.3.1 Graphical user interface components widgets

Overall, a total of 10 UI widgets have been designed and developed aimed to accommodate the key system functionalities, as these have been identified through the requirements

[7] https://github.com/stardog-union/pellet.

elicitation process. Each component featured three LoDs, exhibiting variance according to the information type accommodated. The supported widgets were as follows:

- **Person identification**. This category contains all widgets that can be associated with a detection that has been classified as a human. These widgets impart information about a person of interest, including their name if available, and information about their criminal record activities (if any) provided by the C &C centre. In the case of person identification pertaining to an affiliate member of the LEA squad/patrol team, information about their field expertise and the current physical state by means of biosignals is provided. The component diversifies according to the hostility of the identified individual, annotating hostile individuals are annotated in red colour, civilians in yellow, allies in blue, and suspects of criminal behaviour in orange colour.
- **Object identification**. The specific category contains all widgets that can be associated with a detection that has been classified as an object. For the purposes of LEA operational frameworks, such detections may trigger warnings because of their status (e.g. being unattended, or potentially dangerous) or because of their identification as various kinds of weapons or explosives. Annotations feature orange colour for suspicious objects and red for dangerous ones. Currently, the following object types (and corresponding icons) have been accommodated: unidentified object, marked by a question mark, suitcase and weapon, including guns and knifes.
- **Health status information**. The health widgets category contains all widgets that can be associated with the detection and classification of wounded individuals in the scene, along with a prioritisation of the injuries' treatment needs. This widget is subject to information provided by paramedics on-site and communicated to the C &C Centre.
- **Abnormal behaviour indication**. In the DARLENE context, this component is associated with behaviours that are irregular for the current context of situation as they are recognized by the Computer Vision module, and namely punch, kick, hit with object.
- **Alert**. This widget has been designed to facilitate on-screen information regarding the LEA's current objective, directly communicated by the operations control centre.
- **Directions**. Directions widgets are intended as waypoints to provide on-screen assistance for navigation in the patrol environment. In addition, a crosshair widget has been developed to function as a precision pointer, drawing the attention of a LEA to a particular point of interest.
- **Map**. The Map widget aims to assist the LEA in orienting themselves through a map-based visualisation of the patrolling area, also providing useful information about other points of interest in the area. It requires that a map of the area or building is available.
- **Step-by-step guidance**. Information widgets are intended to provision short and comprehensive tutorialised material for performing specific tasks (e.g. provide first-aid assistance). These can be sent by the operations command centre whenever a situation is encountered and on-screen guidance is warranted.
- **Summative information**. This widget is a comprehensive indicator of all detections on screen, meant to act as a permanent reminder of the current situation at all times. It includes information on all detected people and objects in the scene, with a counter indicating the number of detections in each category.
- **Live feed**. The live feed widget enables a LEA smart glasses wearer to visualise real-time camera feeds from various locations of the patrolling area, creating a second-screen experience for monitoring movements and areas remotely.

Each graphical element in the point-of-view HUD interface is an independent entity designed to convey context-relevant information directly in front of the wearers view for as long as that information stays relevant to the context, encapsulating knowledge received through the CVAF following the performant "Eye-Glance interface" paradigm described in Lu et al. (2020). Isolation between all GUI components (e.g. annotation widgets and algorithmic detection components) ensures that each element displayed on the HUD can regulate its own independent LOD regardless of the LOD setting applied to the other components currently in view. An example of this functionality is summarized in Fig. 8. As can be seen, various HUD annotation widgets can be displayed simultaneously alongside algorithmic detection GUI components, where the adaptation decision-maker can determine the proper LOD for each one individually.

[8] The annotation widgets specifically implement context-adaptive features. Each widget stems from a prefabricated (prefab) Unity GameObject hierarchy, attachable to the built-in Canvas UI system. In this way, adaptation functionality is shared among all widget objects, with each regulating its own adaptation elements (e.g. functionality, aesthetics, information granularity, etc.) through individually added user-written code. The prefab hierarchy implements a multi-LOD architecture with support for up to 3

[8] Video by Anton V., "Greece, Athens, Metro ride from Syngrou Fix to Omonia" - https://www.youtube.com/watch?v=dE1cXBmL1NA

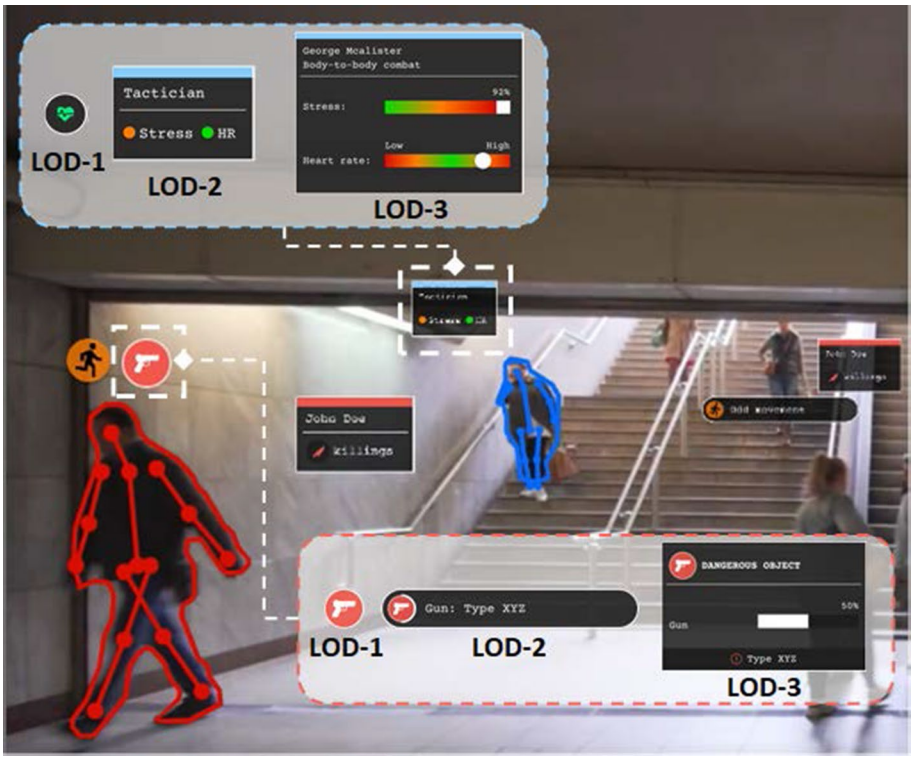



**Fig. 8** Composite image of annotation widgets and algorithmic detection GUI components rendered on top of a pre-recorded video frame[8]. Different LODs apply in the person (ally, in blue) and weapon identification (in red) widgets

layers of informational depth (Fig. 8), similar to Daskalogrigorakis et al. (2021), i.e.:

- The Base Layer corresponds to a GameObject "container" used for determining the widget's placement on the final rendered image. The Layer does not implement any rendering routines whatsoever.
- The LOD-1 is a child GameObject to the Base Layer. It implements the most basic GUI elements that aim at communicating the bare-bones version of the information that the widget supports.
- The LOD-2 Layer is a second child GameObject to the Base Layer. It increases granularity of the presented information with additional graphical elements, enlarging the size of the widget as a result. In some cases, LOD-2 represents the highest LOD attainable (having similar characteristics to LOD-3).
- The LOD-3 Layer GameObject is a third, optional (in many cases) child of the Base Layer. It presents the most unabbreviated version of the information, which might take up a significant part of (or in some cases, the entirety) of the screen.

Every widget component hierarchy can be instantiated at run-time. Only the Base Layer should always remain active at all times. LOD-1, LOD-2 and LOD-3 can be selectively enabled at the behest of the adaptation decision-maker.

In addition to widgets, algorithmic detection GUI components follow a similar instantiation paradigm, with prefabs being associated to the type of detection supported by the application use case (e.g. 2D skeleton rendering, outline rendering). In contrast to widgets however, these components do not support LOD-based adaptive features.

The process of selecting the LOD for each annotation is an integral part of the decision-making process carried out by the Adaptation decision-maker. This is done to ensure that the system presents relevant information at the right time, in the right location on the display, and in the appropriate manner, taking into account contextual factors and placement constraints. The decision-making process involves a combination of ontology modeling and reasoning with Combinatorial Optimization. This approach helps to improve the Situation Awareness of the Law Enforcement Agency (LEA) by conveying the necessary information while avoiding cognitive overload Stefanidi et al. (2022).

#### 5.3.2 GUI adaptation toolkit

The GUIT is a rendering engine entity entrusted with orchestrating the final rendered image through the selection and adaptation of renderable components (as described in Section 5.3.1), which it does by decoding the binary information received from the adaptation decision-maker. Connectivity is implemented in Unity through the MIT-licenced

Unity3D.Amqp third-party package (Everett 2017), which implements an AMQP protocol client for Unity supporting RabbitMQ as a message broker. Each received message is then translated into actions, such as:

- Instantiating a new renderable (widget, or detection GUI component) derived from its respective prefab.
- Associating each renderable with a unique tracking ID stored in an internally kept tracking dictionary (to persist display in consequent rendering frames).
- Determining placement of each element to the optimal screen "cell", by treating the final rendered image as a grid, and keeping track of the cells occupied by existing renderables (widget width and height correspond to dimensions that are multiples of the cell size). In this way, we avoid rendering widgets on top of one another, and keep the visualisation sleek and clean to avoid information overload.

To support real-time rendering performance, messages received by the adaptation decision-maker can either contain full annotation data, or shortened tracking information. Full data messages are received every 25 rendering frames and include data on the algorithmic detections, annotation data on the detected entities, subject segmentation points, proper LOD for each annotation object and grid placement information. Tracking messages are received in between full data updates, and enable segmentation, or skeleton points on detected subjects to be updated (by means of their tracking IDs), which in turn allows for the calculation of a new center of gravity point based on the bounding box computed for each updated segmentation/skeleton points collection. Hence, a widget associated to the detected object's tracking ID can remain "anchored" to it and follow its movement on the screen seamlessly for the entirety in which the subject remains on screen. It should be mentioned that full data messages are sent every 25 rendering frames, so as to avoid increasing computation latency and create an unnecessary bottleneck for the system. However, the system keeps sending for each rendering frame the position and the outline mask of the detected objects; thus, real-time visualization on the AR glasses is not compromised.

## 6 Experimental evaluation

In this section the results of the experimental evaluation are presented. In the following subsections, the different submodules are evaluated individually as well as the whole system. For the whole system, an end user based evaluation was carried out as well, examining aspects as Situational Awarness, Workload and User Experience.

### 6.1 Evaluation of the CVAF

In order to evaluate the performance of the deployed algorithms in our proposed CVAF framework, several experiments have been carried out. The first set of experiments evaluates the performance of the two core modules of CVAF, namely the segmentation (S) and the pose estimation module (P). In the second set, the performance of the overall framework is evaluated. All of the presented experiments were conducted on the embedded platform that the proposed system exploits, an Nvidia Xavier AGX.

#### 6.1.1 Instance segmentation experimental evaluation

The instance segmentation module was evaluated both in terms of performance speed and accuracy. For evaluation the well-established MS-COCO dataset was exploited, as well as the test set of the DARLENE dataset as described in Sect. 4.1.1. The experimental results indicate that the employed training technique and data augmentation method allowed for a lightweight instance segmentation method, able to perform real-time while maintaining a similar accuracy compared to much computational costly architectures.

Table 2 showcases a comparison for the well-established MS-COCO dataset among YOLACT variations utilizing different backbone feature extraction networks. YOLACT ResNet50 is provided by its respective authors[9] and ResNet18 variation is trained on the MS-COCO dataset. Examining the results, the synthetic occlusion augmentation for the person class allowed the proposed architecture to achieve better results for the person class than the much heavier ResNet50 architecture. Additionally, the performance is on par for the rest of the classes. However, the speed performance of the proposed ResNet9 is 7 times better compared to ResNet50 and +15 FPS on average compared to ResNet18 that cannot achieve real-time performance. Comparing the proposed method with YOLOv5, it is noticeable that on the one hand they have a similar performance in terms of mAP; however, the slightly worse performance in terms of speed of YOLOv5 would compromise the real-time need of the DARLENE system. Figures 10 and 9 depict the mask and bounding box output of S for objects of interest. In Fig. 10 the system is able to detect the objects of interest even when intra-class occlusions occur. In Fig. 9 resulting masks on images from the MS-COCO dataset containing objects of interest are depicted.

[9] https://github.com/dbolya/yolact.

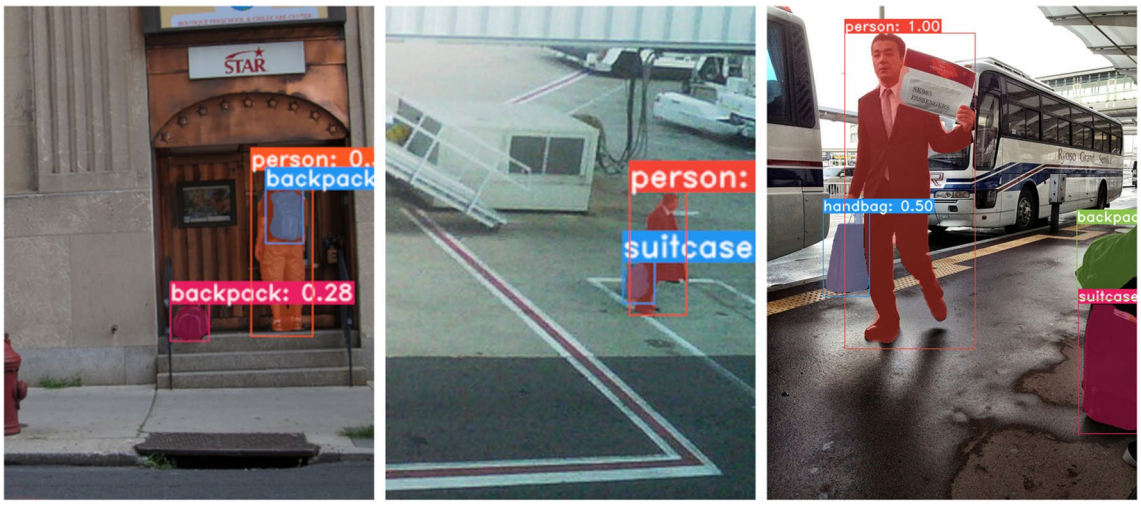



**Fig. 9** Qualitative results of segmentation method on MS-COCO validation set images

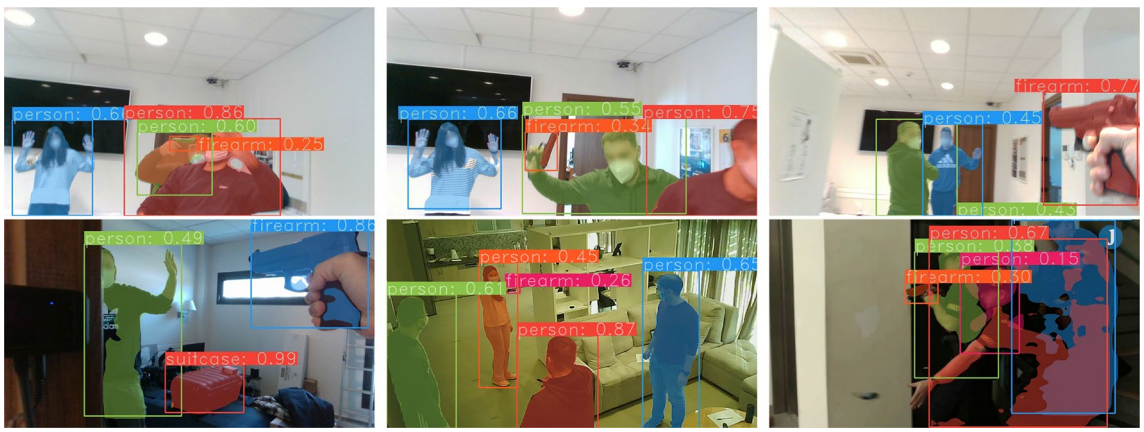



**Fig. 10** Qualitative results of the Segmentation module for the DARLENE use cases

**Table 2** Quantitative results of segmentation methods.

| | YOLACT ResNet50 | YOLACT ResNet18 | YOLOv5 | Proposed |
|---|---|---|---|---|
| Person | 27.6% | 25.9% | **33.2** % | 30.0% |
| Backpack | 8.6% | 4.5% | 6.3 % | **9.2**% |
| Handbag | **7.79**% | 4.3% | 4.4 % | 5.8% |
| Suitcase | **23.7**% | 14.8% | 12.1 % | 13.4% |
| Knife | **3.2**% | 1.4% | 5.5 % | 2.9% |
| Mean | **14.2**% | 10.2 % | 12.3 % | 12.26 % |
| Firearm | – | - | 31.1 % | **32.9** % |
| AVG FPS | 5.6 | 20.4 | 33.2 | **35.6** |

The reported values per experiment correspond to mean Average Precision of the generated mask, when evaluated on MS COCO validation and custom DARLENE test sets and the Frame Per Second (FPS) performance on Jetson AXG Xavier

### 6.1.2 Pose estimation

This subsection covers the evaluation of the Pose Estimation module, described in Sect. 4.2. We compare the model with other architectures, in specific a method with ResNet-18 as its backbone, in terms of accuracy and inference speed. Both these methods were trained and evaluated on MS-COCO. The object keypoint similarity (OKS) as defined in Lin et al. (2014) is exploited for evaluation, which serves the same purpose as IoU in Object Detection. The accuracy metric is the mean Average Precision (mAP) over 10 OKS thresholds.

The inference test was conducted on a Jetson AGX Xavier with TensorRT inference engine at Floating Point 16(FP16)

**Table 3** Evaluation of the pose estimation models in terms of accuracy (Average Precision) and inference speed on Jetson AGX Xavier

| Model | AP % | Single Target Inference(ms) FP16 | Single Target Inference(ms) INT8 |
|---|---|---|---|
| ResNet-18 | 67.5 | 1.7 | 1.3 |
| ResNet-8 | 44.8 | 0.7 | 0.7 |

The evaluation was carried out exploiting the MS-COCO dataset for pose estition

and Integer 8(INT8) accuracy, and the results are presented in Table 3. ResNet-18 has higher mAP at 67.5%, while ResNet-8 stands at 44.8%, and their single target inference, meaning their inference on a cropped part of the image containing a person, was 1.7 and 0.7 ms, respectively, at fp16. ResNet-8 is 2.4 times faster than ResNet-18, but at the same time its AP is 22.7 percentage points lower. At integer 8 accuracy the ResNet-18 stands at 1.3 ms, while the ResNet-8 does not have any performance gain.

Both these methods can be deployed on the system. When there is high system load and the device is low in resources, we prefer to use ResNet-8 since it has lower latency and it scales better with the number of persons, while when there are not many people in a scene, the ResNet-18 architecture is preferred for its higher accuracy.

#### 6.1.3 Evalutation of CVAF and AR visualization

The complete Computer Vision Analysis Framework was evaluated in terms of speed performance on data that was collected for the needs of the DARLENE Project. The dataset contains footage of people performing simulated/staged illegal actions (threatening other people, holding weapons, etc.) as well as scenes with people, where no illegal activity is taking place. In Table 4 the various computer vision components are evaluated on these video sequences towards measuring their respective speed performance. We report the results for segmentation, segmentation and pose, segmentation, pose estimation and tracking and finally the whole system speed performance including the communication with the rest of the components (AR glasses, etc.). The system was evaluated for the case when only one object is tracked and for the case when up to 20 objects are simultaneously tracked. The first two rows employ the ResNet-8-based pose estimation method, while the last two the ResNet-18 variant. When only the segmentation module is exploited, the system can achieve real-time performance. The speed difference of S compared to Table 2 is due to the evaluation on videos and not on single images as well as the polygon computation of each mask detection. When the pose module is enabled alongside S, the system drops below the real-time threshold of 25 FPS for the 20 objects case. However by employing the tracking propagation the system is able to perform real-time achieving on average 30 or 34 FPS regarding the selected pose estimation method. Finally, the overall system speed performance reported in the last column, is comfortably above 25 FPS. It should be noted here that even if, there is a speed performance drop by the polygon calculation for each detection, this trade-off is necessary since the transferring of image mask would have resulted in bigger delays during data transfer for visualization (8–10 pixel coordinates per object compared to image masks).

Figure 11 depicts the system visualization for three different scenarios. In the first one depicted in Fig. 11a the system has detected possible suspects that appear to perform odd movements. In the third frame, the white color annotation indicates that although the target is detected, the visualization optimization of the system decides to hide the visualization on the glasses towards reducing the visual clutter for the user. The same holds for the rest of the scenarios. In Fig. 11b a foe holding a firearm attacks a group of LEAs. The system is able to detect the firearm and provide a helpful LoD for the user. Similarly, in Fig. 11c the system detects the attacking foe and annotates the person with red color, while the victim is annotated with yellow color with additional information about its current status in the floating widget. In the first frame although the person is detected, it is not visualized in the AR glasses since it is of no interest yet. From all of the aforementioned examples, the effect of sparsely employing the segmentation module can be seen. Examining the last frame of Fig. 11a, it is noticeable that the blue polygon is not perfectly aligned with the person, due to the fact that the polygon is propagated and not re-calculated. However, since it is correctly centered on the tracked person and with the aid of the pose estimation module, the result is acceptable in terms of quality.

#### 6.1.4 Latency analysis and overall execution time

In this set of experiments, the overall execution time of the system was measured towards obtaining the time needed from image acquisition until the AR information is rendered on the glasses. The evaluation was carried out by measuring the time performance of the CVAF, Data Handler, Adaptation Decision-Maker and the actual rendering on the AR glasses. In the first experiment, apart from the rendering that is always carried by the Android device that handles the AR glasses projectors, all of the components of the DARLENE system were executed on the wearable device. The experiment was carried out for 15000 frames, for which the CVAF was able to detect object of interest. Figure 12 depicts the per frame latency for the aforementioned frames, and in Table 5, the average values are reported. By examining

**Fig. 11** System output on video frames. Blue annotation indicates LEAs, orange possible suspects and red color foes. White annotation indicates annotation that is produced by the system but is not actually visualized in the glasses towards reducing the visual clutter

**(a)** Suspicious movement of targets as visualized by the system. To reduce clutter, the system may hide the annotation of a detected person/object.

**(b)** A suspect attacks LEAs holding firearm. The system reduces information clutter by visualizing the most important objects such as the firearm and the foe and adjusting the LoD.

**(c)** The person in the first frames, although detected, will not be annotated since it is of no interest yet. Once the foe appears and detected by the system, the status of the first person is updated.

**Table 4** FPS performance of computer vision modules on WECN

| Max Detections | Modules | | | |
|---|---|---|---|---|
| | S | S + P | S + P + T | S + P+T + C |
| 20 | 25 | 21 | 34 | **32** |
| 1 | 33 | 30 | 54 | **51** |
| 20 | 25 | 19 | 30 | 29 |
| 1 | 33 | 29 | 50 | 50 |

*S* segmentation, *P* Pose estimation, *T*: tracking, *C* connection to RabbitMQ. For the last two results, S module is executed every 24 frames and P every frame. First two rows pose ResNet-8, the last two with pose ResNet-18

the results, when all of the system components are executed on the wearable device the latency induced by the analysis performed by CVAF is 31 ms which translated to 32 FPS processing speed. Additionally, the Data Pipeline Handler induces on average 6.6 ms and the Decision-Maker 3 ms. The rendering on the glasses requires an average of 1.4 ms, leading to an average latency from the image acquisition to annotation rendering of 41.9 ms.

In the second experiment, the system latency for the case of executing CVAF on a cloud service was measured. As a cloud computing device, a personal computer with an Nvidia 3060 was exploited. For this experiment, apart from the components measured in the previous experiment, the latency induced by the video streaming service needs to be evaluated. The performance for each frame is illustrated in Fig. 13, and the average values are reported in Table 5. It should be mentioned that Data Handler and Decision-Maker are still executed on the wearable device and the rendering is performed by the standalone Android device handling the AR projectors. Thus, the rendering latency remains the

**Table 5** Average latency induced by each component of the DARLENE AR system

| Mode | Streaming | CVAF | Data Handler | Decison Maker | Rendering | Total |
|---|---|---|---|---|---|---|
| Wearable | – | 31 | 6.6 | 3 | 1.4 | 41.9 |
| Cloud | 16.1 | 12 | 9.6 | 3 | 1.4 | 42 |

All reported values are in milliseconds

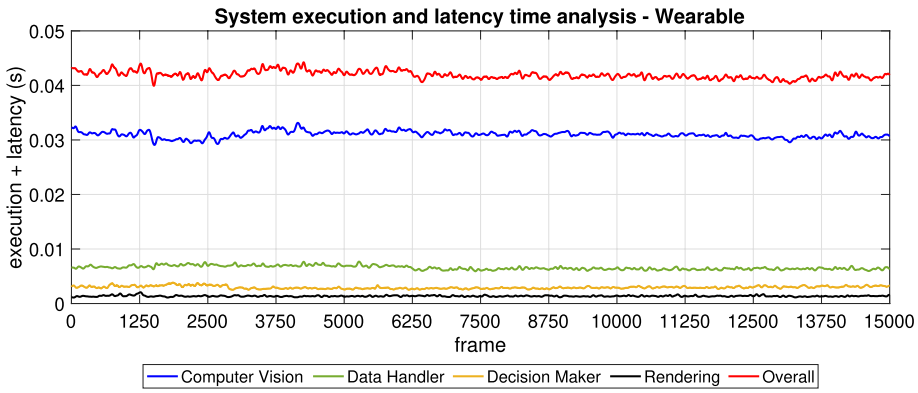



**Fig. 12** System execution and latency time analysis on the wearable device. For each component the measured time includes the execution and transmission of results (e.g. JSON files) to the next component

same as well as the average latency induced by the DecisionMaker. However, Data Pipeline Handler needs more time for each frame since it communicates with the cloud server to receive the annotation of each frame, leading to an increased latency by +3 ms when compared to the first experiment. The latency of the streaming service adds on average 16.1 ms of latency and the CVAF analysis on the cloud requires 12 ms. For this experiment, the average latency was measured at 42 ms.

## 6.2 End user evaluation

The system and the SA it achieves has been assessed through a user-based evaluation[10] involving 20 LEAs. In specific, the goal of this evaluation was to validate that the system achieves its overall goal by enhancing agents' SA during policing tasks without imposing mental workload, ensuring at the same time a high-quality UX. Users experienced the system through viewing videos of staged terrorist attacks. The study was set up as a within-subjects experiment, involving two variables, namely stress and system usage, resulting in four experimental conditions, delivered in a randomized order through a $4 \times 4$ Latin square design: (a) the agent is not stressed and is not using the DARLENE system, (b) the agent is not stressed and is using the DARLENE system, (c) the agent is stressed and is not using the DARLENE system, and (d) the agent is stressed and is using the DARLENE system. In the conditions where the participant should not be using the DARLENE system, the videos shown did not feature the AR WECN visualization, as opposed to the conditions featuring the system. Participants' stress was manipulated through a mental arithmetic task for inducing stress (Tombaugh 2006) and relaxing videos for achieving a calm state.

The experiment encompassed three distinct phases, namely introduction phase, the main study segment and a debriefing phase. Initially, participants were welcomed and explained the study's objectives and purpose. Following this, they provided their informed consent by signing a consent form and completed a demographic information questionnaire. Subsequently, a brief presentation introduced them to the DARLENE User Interface widgets and their LoDs to familiarize them with the system. A calibration of the HMD followed. During the main part of the experiment, each experimental condition commenced with a stress manipulation task for the stressed and unstressed conditions accordingly. Questionnaires were administered at the end of each experimental condition.

A detailed analysis of the evaluation results is provided in Stefanidi et al. (2022); however, a summary of findings is provided in this paper for enhancing readers' comprehension on the evaluation outcomes.

### 6.2.1 Situational awareness

SA was measured as a perceived and observed phenomenon, through the SART questionnaire and the SAGAT

[10] The study has been conducted following approval by the Social and Societal Ethics Committee of the Katholieke Universiteit Leuven (KUL approval number G-2021 09 2072).

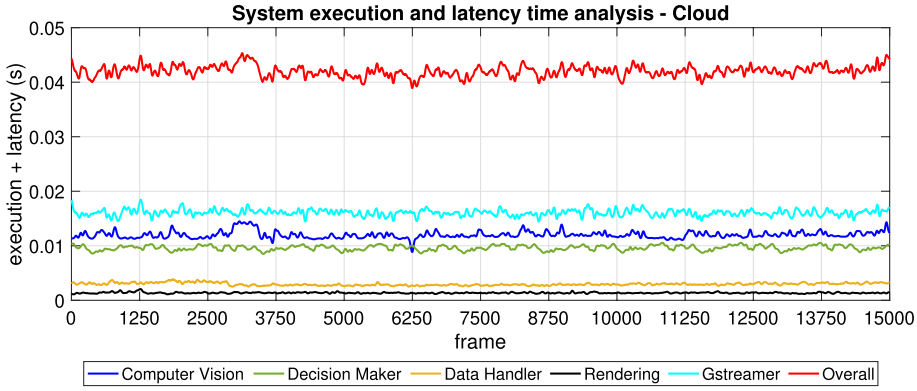



**Fig. 13** System execution and latency time analysis exploiting cloud server. For each component the measured time includes the execution and transmission of results (e.g. JSON files) to the next component

**Table 6** Situational awareness SART results across the four studied conditions in terms of stress and system usage

| | Stressed | | Not Stressed | |
|---|---|---|---|---|
| | With the system | Without the system | With the system | Without the system |
| Mean | 23.6 | 17.89 | 21.00 | 18.16 |
| Min | 16 | -1 | 7 | 9 |
| Max | 38 | 29 | 32 | 28 |
| Range | 22.00 | 30.00 | 25 | 19 |
| SD | 5.08 | 6.87 | 6.19 | 4.14 |
| 95%CI | [20.82. 25.71] | [14.58. 21.21] | [18.02. 23.98] | [16.16. 20.15] |

For each condition, the mean score, minimum, maximum, score range, standard deviation and 95% confidence interval is reported

**Table 7** Situational Awareness SAGAT results across the four studied conditions in terms of stress and system usage

| | Stressed | | Not Stressed | |
|---|---|---|---|---|
| | With the system | Without the system | With the system | Without the system |
| Mean | 69.52 | 66.88 | 70.86 | 61.62 |
| Min | 42.86 | 46.67 | 40.00 | 28.57 |
| Max | 93.33 | 80.00 | 93.33 | 86.67 |
| Range | 50.48 | 33.33 | 53.33 | 58.10 |
| SD | 12.66 | 8.96 | 14.62 | 14.49 |
| 95%CI | [63.60 75.44] | [62.69 71.07] | [64.01 77.70] | [54.84 68.40] |

For each condition, the mean score, minimum, maximum, score range, standard deviation and 95% confidence interval is reported

query technique (Salmon et al. 2009) correspondingly. More specifically, SART entails SA questions with respect to ten dimensions, which are classified into three main subscales: *Attentional Demand*(AD), *Attentional Supply* (AS) and *Understanding* (U). The score for each subscale is calculated as the sum of the participant's rating in each of the subscale's questions, which range from 1 to 7. The final SART score is calculated as per Eq. (11).

$$\mathrm{SART}_{\mathrm{score}} = \mathrm{U} - (\mathrm{AD} - \mathrm{AS}). \tag{11}$$

Scoring for the SAGAT questionnaire entails awarding a single score point for each accurate answer to a question and no points for incorrect responses. The total score for each participant is then summed up and divided by the total number of questions presented to them, in order to calculate their final SAGAT score, which signifies the percentage of correct responses provided.

It is evident that both perceived (Table 6) and observed (Table 7) SA was in all cases higher when participants were using the system. In particular, the observed SA was

**Table 8** Workload results based on the NASA-TLX Questionnaire, across two conditions studied with the system (stressed and not stressed), as well as in comparison with standard policing tasks, namely warm up, flashlight, barrel, and metal

| | Stress cond. | No stress cond. | Warmup policing task | Flashlight policing task | Barrel policing task | Metal policing task |
|---|---|---|---|---|---|---|
| Mental | 65.79 | 64 | 52 | 59 | 68 | 65 |
| (M.SD) | 25.89 | 28.45 | 26 | 23 | 21 | 24 |
| Physical | 20.00 | 22.5 | 28 | 40 | 62 | 53 |
| (M.SD) | 27.69 | 29.13 | 23 | 26 | 24 | 25 |
| Temporal | 53.16 | 54.5 | 38 | 45 | 67 | 62 |
| (M.SD) | 28.97 | 31.12 | 26 | 26 | 22 | 25 |
| Perform. | 62.89 | 60.5 | 63 | 61 | 55 | 52 |
| (M.SD) | 17.74 | 20.45 | 24 | 23 | 23 | 22 |
| Effort | 58.68 | 47.25 | 48 | 53 | 65 | 62 |
| (M.SD) | 17.70 | 24.14 | 23 | 24 | 20 | 20 |
| Frustrat. | 35.53 | 33.5 | 29 | 38 | 47 | 49 |
| (M.SD) | 23.86 | 26.01 | 23 | 23 | 27 | 25 |

Results for each condition are reported as mean and standard deviation scores, across the following workload dimensions: mental, physical, temporal, performance, effort, and frustration

**Table 9** UX results as collected through the UMUX-Lite questionnaire responses

| | Stressed | | | Not Stressed | | |
|---|---|---|---|---|---|---|
| | Meets reqs | Easy to use | Overall score | Meets reqs | Easy to use | Overall score |
| Mean | 4.95 | 5.11 | 5.03 | 4.85 | 5.25 | 5.05 |
| Min | 2.00 | 1.00 | 2.50 | 2.00 | 2.00 | 2.00 |
| Max | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 |
| Range | 5.00 | 6.00 | 4.50 | 5.00 | 5.00 | 5.00 |
| SD | 1.35 | 1.94 | 1.50 | 1.31 | 1.37 | 1.23 |
| 95%CI | [4.30, 5.60] | [4.17, 6.04] | [4.31, 5.75] | [4.24, 5.46] | [4.61, 5.89] | [4.47, 5.63] |

Results are reported across the two studied conditions when using the system, namely stressed and not stressed, as well as across the various UX dimensions supported by the employed instrument, namely if the system meets user requirements, if it is easy to use, as well as overall UX score. For each dimension the mean, minumum, maximum, value range, standard deviation, and 95% confidence interval is reported

increased with the system by 3.95% in the stress condition and by 15% in the no stress condition. Perceived SA was increased by 30% in the stress condition and by 15.65% in the no stress condition.

#### 6.2.2 Workload

Workload was measured using the NASA-TLX questionnaire (Hart and Staveland 1988). Results indicate (Table 8) that from the studied constructs, mental and temporal workload and effort were the higher ones, as opposed to physical workload and frustration. At the same time, perceived performance was also high, highlighting that participants felt that they were able to successfully achieve the tasks they were undertaking. In order to investigate if the imposed workload is acceptable in the context of policing tasks, results were compared to findings from a study with police officers in a field shooting exercise (Oron-Gilad et al. 2008), yielding the conclusion that the perceived workload when using the DARLENE system for policing tasks is in general aligned with the workload observed in actual policing tasks.

#### 6.2.3 User experience

The overall UX with the system was assessed through the UMUX-Lite questionnaire (Lewis et al. 2013). This is a standardized, two-items questionnaire, asking respondents to indicate on a scale from 1 to 7, how satisfied they are with the system regarding how it addresses their requirements and how easy it is to use. Overall, results indicate that the system is positively appraised with regard to meeting users' requirements and being easy to use when agents are not stressed, but also when they are stressed (Table 9). Qualitative feedback received through post-test interviews with participants indicated that participants would like to use it in their daily operations and identified issues to be addressed in future improvements.

### 6.3 Limitations

The limitations of the conducted user study pertain to the relatively small number of participants, the lack of proper training in the system, as well as the in vitro setup of the study employing policing videos. The current study was carried out as an initial user-based study on a working prototype of the system, aiming to acquire insights on the usability of the system, as well as its impact on situational awareness and workload, before proceeding with a large-scale study involving more participants. Nevertheless, follow-up studies have already been planned, involving a larger number of LEAs, who will use the actual system in simulated policing tasks, after having been trained on its usage.

## 7 Conclusion

Advancements in machine learning, visualisation technologies and hardware are gradually enabling their combined use in real-time performing applications. In this paper we elaborated on the design, architecture, main novelties and individual components integrated into such an application, intended for various police use cases, such as patrolling and tactical threat neutralisation. Due to the highly critical nature of its use cases, our system places emphasis on achieving real-time performance of the various computer vision algorithms, which combine for AI-assisted, rapid visual scene analysis towards heightening user awareness of potentially dangerous situations. As soon as information becomes available, an efficient and intelligently driven rendering pipeline for wearable AR smart glasses enables an effective, glanceable and usable visualisation of the detected information, taking also into account the user's current physiological state, by means of wearable IoT biosignals sensors, and current context of use. We attested to the system's real-time performance by conducting a thorough comparative experimental evaluation, which further explored our solution's performance as perceived by the intended end users.

Future work will focus on the integration of the presented technologies with additional disruptive innovations, particularly in the networking domain (e.g. osmotic computing, 5 G networks, etc.), while the eventual final system will be extensively evaluated by end users. The present system could also be improved in several aspects as hardware advancements occur. More powerful wearable devices can allow for heavier architectures for the existing CVAF modules, as well as the addition of other CV methods that could improve the SA of the end users (e.g. action/activity recognition). Additionally, the AR glasses could benefit from visors with larger FoV and while being overall smaller and lighter making the everyday use of the DARLENE system easier for the police forces. Furthermore, future work will focus on additional user-based studies, involving a larger number of participants, carrying out policing tasks.

## 8 Data and resources

The MS-COCO dataset that supports the findings of this study is available in https://cocodataset.org/, Open Images Dataset is available in https://storage.googleapis.com/openimages/web/index.html and Monash Guns Dataset available in https://github.com/MarcusLimJunYi/Monash-Guns-Dataset. The custom DARLENE dataset that was created and exploited for evaluation will be publicly available upon paper acceptance.

**Acknowledgements** This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883297 (project DARLENE).

**Funding** Open access funding provided by HEAL-Link Greece.

### Declarations

**Conflict of interest** All authors have no conflicts of interest.



## References

Abowd GD, Dey AK, Brown PJ et al (1999) Towards a Better Understanding of Context and Context-Awareness. In: Gellersen HW (ed) Handheld and ubiquitous computing. Springer, Berlin, Heidelberg, pp 304–307

Alvarez-Marin A, Velazquez-Iturbide JA (2022) Augmented reality and engineering education: a systematic review. IEEE Trans Learn Technol 14(6):817–831

Apostolakis KC, Dimitriou N, Margetis G, et al (2021) DARLENE—Improving situational awareness of European law enforcement agents through a combination of augmented reality and artificial intelligence solutions. Open Research Europe, version 1; peer review: 2 approved with reservations

Bolya D, Zhou C, Xiao F, et al (2019) Yolact: real-time instance segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9157–9166

Bolya D, Zhou C, Xiao F, et al (2020) YOLACT++: Better real-time instance segmentation. In: IEEE Transactions on pattern analysis and machine intelligence

Braun V, Clarke V (2021) Conceptual and design thinking for thematic analysis. Qual Psychol. https://doi.org/10.1037/qup0000196

Buettner R, Baumgartl H, Konle T, et al (2020) A review of virtual reality and augmented reality literature in healthcare. In: 2020 IEEE symposium on industrial electronics applications (ISIEA), pp 1–6

Cao Z, Hidalgo G, Simon T et al (2019) OpenPose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans Pattern Anal Mach Intell 43(1):172–186

Daskalogrigorakis G, McNamara A, Mania K (2021) Holo-Box: Level-of-Detail Glanceable Interfaces for Augmented Reality. In: ACM SIGGRAPH 2021 Posters. Association for Computing Machinery, New York, NY, USA, SIGGRAPH '21, https://doi.org/10.1145/3450618.3469175

Deng J, Dong W, Socher R, et al (2009) ImageNet: a large-scale hierarchical image database. In: CVPR09

Dimitriou N, Kioumourtzis G, Sideris A, et al (2017) An integrated framework for the timely detection of petty crimes. In: 2017 European intelligence and security informatics conference (EISIC), IEEE, pp 24–31

Dosovitskiy A, Beyer L, Kolesnikov A, et al (2021) An image is worth 16x16 words: transformers for image recognition at scale. ICLR

ElKomy M, Abdelrahman Y, Funk M, et al (2017) ABBAS: An Adaptive Bio-Sensors Based Assistive System. In: Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems. Association for Computing Machinery, New York, NY, USA, CHI EA '17, p 2543-2550, https://doi.org/10.1145/3027063.3053179

Endsley TC, Sprehn KA, Brill RM et al (2017) Augmented reality design heuristics: designing for dynamic interactions. Proc Hum Factors Ergonom Soc Ann Meet 61(1):2100–2104. https://doi.org/10.1177/1541931213602007

Everett M (2017) Unity3D.Amqp. https://github.com/CymaticLabs/Unity3D.Amqp

Fang HS, Xie S, Tai YW, et al (2017) Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE international conference on computer vision, pp 2334–2343

Fereday J, Muir-Cochrane E (2006) Demonstrating rigor using thematic analysis: a hybrid approach of inductive and deductive coding and theme development. Int J Qual Methods 5(1):80–92

Fu Z, Liu Q, Fu Z, et al (2021) STMTrack: Template-free Visual Tracking with Space-time Memory Networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13774–13783

Gao N, Shan Y, Wang Y, et al (2019) Ssap: Single-shot instance segmentation with affinity pyramid. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 642–651

Ghiasi G, Cui Y, Srinivas A, et al (2021) Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 2918–2928

Grubert J, Langlotz T, Zollmann S et al (2017) Towards pervasive augmented reality: context-awareness in augmented reality. IEEE Trans Visual Comput Graphics 23(6):1706–1724. https://doi.org/10.1109/TVCG.2016.2543720

Hart SG, Staveland LE (1988) Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In: Advances in psychology, vol 52. Elsevier, p 139–183, https://doi.org/10.1016/S0166-4115(08)62386-9, https://linkinghub.elsevier.com/retrieve/pii/S0166411508623869

He K, Zhang X, Ren S et al (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916

He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

He K, Gkioxari G, Dollár P, et al (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969

Henriques JF, Caseiro R, Martins P et al (2014) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596

Hoque S, Arafat MY, Xu S et al (2021) A comprehensive review on 3D object detection and 6d pose estimation with deep learning. IEEE Access 9:143746–143770

Hussain J, Hassan AU, Bilal HSM et al (2018) Model-based adaptive user interface based on context and user experience evaluation. J Multim User Interfaces 12:1–16. https://doi.org/10.1007/s12193-018-0258-2

Jocher G, Stoken A, Borovec J, et al (2020) ultralytics/yolov5: v3.1 - bug fixes and performance improvements. https://doi.org/10.5281/zenodo.4154370

Karakostas I, Mygdalis V, Tefas A et al (2020) Occlusion detection and drift-avoidance framework for 2D visual object tracking. Signal Process Image Commun 90(116):011

Kilis N, Tsipouridis G, Karakostas I, et al (2023) Augmentation based on artificial occlusions for resilient instance segmentation. In: International conference on image analysis and processing, Springer, pp 37–48

Kim JC, Laine TH, Åhlund C (2021) Multimodal interaction systems based on internet of things and augmented reality: a systematic literature review. Appl Sci. https://doi.org/10.3390/app11041738

Kuznetsova A, Rom H, Alldrin N et al (2020) The open images dataset v4. Int J Comput Vision 128(7):1956–1981

Köppel T, Eduard Gröller M, Wu HY (2021) Context-Responsive Labeling in Augmented Reality. In: 2021 IEEE 14th Pacific visualization symposium (PacificVis), pp 91–100, https://doi.org/10.1109/PacificVis52677.2021.00020

Lavoie R, Main K, King C et al (2021) Virtual experience, real consequences: the potential negative emotional consequences of virtual reality gameplay. Signal Real 25(1):69–81

Lee Y, Park J (2020) Centermask: Real-time anchor-free instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13906–13915

Lewis JR, Utesch BS, Maher DE (2013) UMUX-LITE: when there's no time for the SUS. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, Paris France, pp 2099–2102, https://doi.org/10.1145/2470654.2481287

Li R, Pang M, Zhao C, et al (2016) Monocular long-term target following on uavs. In: Conference on computer vision and pattern recognition (CVPR) pp 29–37

Lim J, Al Jobayer MI, Baskaran VM et al (2021) Deep multi-level feature pyramids: application for non-canonical firearm detection in video surveillance. Eng Appl Artif Intell 97(104):094

Lin TY, Maire M, Belongie S, et al (2014) Microsoft coco: Common objects in context. In: European conference on computer vision, Springer, pp 740–755

Liu H, Liu F, Fan X, et al (2021) Polarized self-attention: Towards high-quality pixel-wise regression. arXiv preprint arXiv:2107.00782

Liu S, Qi L, Qin H, et al (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8759–8768

Lu F, Davari S, Lisle L, et al (2020) Glanceable AR: evaluating information access methods for head-worn augmented reality. In: 2020

IEEE conference on virtual reality and 3D user interfaces (VR), pp 930–939, https://doi.org/10.1109/VR46266.2020.00113
Ma C, Yang X, Zhang C, et al (2015) Long-term correlation tracking. In: Computer vision and pattern recognition (CVPR) pp 5388–5396
Mao W, Ge Y, Shen C, et al (2021) Tfpose: Direct human pose estimation with transformers. arXiv preprint arXiv:2103.15320
Margetis G, Ntoa S, Antona M et al (2019) Augmenting natural interaction with physical paper in ambient intelligence environments. Multim Tools Appl 78(10):13387–13433. https://doi.org/10.1007/s11042-018-7088-9
Margetis G, Ntoa S, Antona M et al (2021) Human-centered design of artificial intelligence. In: Salvendy G (ed) Handbook of human factors and ergonomics. Wiley, London, pp 1085–1106. https://doi.org/10.1002/9781119636113.ch42
Oron-Gilad T, Szalma JL, Stafford SC et al (2008) The workload and performance relationship in the real world: a study of police officers in a field shooting exercise. Int J Occup Saf Ergon 14(2):119–131. https://doi.org/10.1080/10803548.2008.11076757
Oulasvirta A, Dayama NR, Shiripour M et al (2020) Combinatorial optimization of graphical user interface designs. Proc IEEE 108(3):434–464. https://doi.org/10.1109/JPROC.2020.2969687
Pellas N, Fotaris P, Kazanidis I et al (2019) Augmenting the learning experience in primary and secondary school education: a systematic review of recent trends in augmented reality game-based learning. Virtual Reality 23(4):329–346
Pradeep P, Krishnamoorthy S (2019) The MOM of context-aware systems: a survey. Comput Commun 137:44–69. https://doi.org/10.1016/j.comcom.2019.02.002
Redmon J, Divvala S, Girshick R, et al (2016) You only look once: Unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 779–788, https://doi.org/10.1109/CVPR.2016.91
Rill BR, Hämäläinen MM (2018) The art of co-creation: a guidebook for practitioners. Springer, Berlin
Salmon PM, Stanton NA, Walker GH et al (2009) Measuring situation awareness in complex systems: comparison of measures study. Int J Ind Ergonom 39(3):490–500. https://doi.org/10.1016/j.ergon.2008.10.010
Silvennoinen JM, Jokinen JP (2016) Aesthetic Appeal and Visual Usability in Four Icon Design Eras. In: Proceedings of the 2016 CHI conference on human factors in computing systems. Association for Computing Machinery, New York, NY, USA, CHI '16, p 4390-4400, https://doi.org/10.1145/2858036.2858462
Siriwardhana Y, Porambage P, Liyanage M et al (2021) a survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects. IEEE Commun Surv Tutor 23(2):1160–1192
Stefanidi Z, Margetis G, Ntoa S et al (2022) Real-time adaptation of context-aware intelligent user interfaces, for enhanced situational awareness. IEEE Access 10:23367–23393. https://doi.org/10.1109/ACCESS.2022.3152743
Sun K, Xiao B, Liu D, et al (2019) Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, pp 5693–5703
Syberfeldt A, Danielsson O, Gustavsson P (2017) Augmented reality smart glasses in the smart factory: product evaluation guidelines and review of available products. IEEE Access 5:9118–9130. https://doi.org/10.1109/ACCESS.2017.2703952
Tombaugh T (2006) A comprehensive review of the paced auditory serial addition test (PASAT). Arch Clin Neuropsychol 21(1):53–76. https://doi.org/10.1016/j.acn.2005.07.006
Tsiktsiris D, Dimitriou N, Lalas A et al (2020) Real-time abnormal event detection for enhanced security in autonomous shuttles mobility infrastructures. Sensors 20(17):4943
Wang CY, Mark Liao HY, Wu YH, et al (2020) Cspnet: A new backbone that can enhance learning capability of cnn. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 390–391
Xiao B, Wu H, Wei Y (2018) Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV), pp 466–481
Xu Y, Zhang J, Zhang Q, et al (2022) ViTPose: Simple vision transformer baselines for human pose estimation. In: Advances in neural information processing systems
Yigitbas E, Jovanovikj I, Sauer S et al (2020) On the development of context-aware augmented reality applications. In: Abdelnour Nocera J, Parmaxi A, Winckler M et al (eds) Beyond interactions. Springer, Cham, pp 107–120
Zhang Y, Wang C, Wang X et al (2021) Fairmot: on the fairness of detection and re-identification in multiple object tracking. Int J Comput Vision 129(11):3069–3087
Zhang Z, Pan Z, Li W, et al (2022) X-board: an egocentric adaptive ar assistant for perception in indoor environments. Virtual Reality pp 1–17