Computers & Graphics 119 (2024) 103909

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on Eye Gaze VISA

Gaze-enabled activity recognition for augmented reality feedback

Kenan Bektaş *, Jannis Strecker, Simon Mayer, Kimberly Garcia

University of St. Gallen, Rosenbergstrasse 30, St. Gallen, 9000, Switzerland

ARTICLE INFO

Dataset link: https://github.com/Interactions-HSG/GEAR

Keywords:
Pervasive eye tracking
Augmented reality
Attention
Human activity recognition
Context-awareness
Ubiquitous computing

ABSTRACT

Head-mounted Augmented Reality (AR) displays overlay digital information on physical objects. Through eye tracking, they provide insights into user attention, intentions, and activities, and allow novel interaction methods based on this information. However, in physical environments, the implications of using gaze-enabled AR for human activity recognition have not been explored in detail. In an experimental study with the Microsoft HoloLens 2, we collected gaze data from 20 users while they performed three activities: *Reading* a text, *Inspecting* a device, and *Searching* for an object. We trained machine learning models (SVM, Random Forest, Extremely Randomized Trees) with extracted features and achieved up to 89.6% activity-recognition accuracy. Based on the recognized activity, our system—GEAR—then provides users with relevant AR feedback. Due to the sensitivity of the personal (gaze) data GEAR collects, the system further incorporates a novel solution based on the Solid specification for giving users fine-grained control over the sharing of their data. The provided code and anonymized datasets may be used to reproduce and extend our findings, and as teaching material.

1. Introduction

To augment a user's visual field of view with virtual information that is overlaid on a physical environment, increasingly ergonomic and powerful see-through head-mounted displays (HMDs) have become the preferred method over the past decade. Early versions of HMDs were complex, bulky, and expensive [1–3]). Today, the form factor and usability of Augmented Reality (AR) HMDs (e.g., Microsoft Hololens 2,[1] Varjo XR-3,[2] Magic Leap 1 and 2,[3] HTC Vive ProEye,[4] Apple Vision Pro[5]) have significantly improved, including with respect to the progressive widening of the provided field of view of these devices. AR HMDs can today track the instant 3D position and movements of the user (often including their head, hands, and eyes), can detect objects in the environment that appear in their camera feed through computer vision methods, and allow novel ways of interaction with connected devices that are close-by or remote [4,5]. Across various indoor and outdoor activities, AR HMDs may provide users with access to relevant information and services *if desired* [2,6–8]. In this way, AR HMDs bring us closer to Weiser's vision [9] of ubiquitous computing: a seamless integration of networked (micro-) computers and displays into our physical world, while keeping the increasing complexity manageable for the end users.

Our eyes provide essential visual input to the brain, thus studying eye movements can give us insights into various cognitive processes and activities of humans (and animals). In this field, researchers have been interested in exploiting the potential of eye tracking for the creation of novel opportunities in human–computer interaction [10] and attention-aware computing [11], for selection [12], foveated rendering [13], activity recognition [14], visual search [15] or in retrospective analysis [16]. Sensors (including regular cameras, infrared-based systems, etc.) that permit eye tracking can today be readily integrated in HMDs—including AR HMDs—for maintaining explicit, implicit, and collaborative interactions in Mixed Reality (MR) applications [8] that can continuously sense and adapt to the requirements and constraints of users' context and activities [6,7,17]. This is true across VR and AR, and beyond, where Milgram and Kishino presented a well accepted continuum of MR [18], focusing mainly on visual experiences in real, augmented, and virtual environments. At the virtual reality (VR) end of this continuum, users are exposed to computer-generated stimuli in the visual, auditory, haptic, and further spaces. In VR environments, eye tracking can provide valuable insights about users action planning and execution strategies (e.g., in [19]). However, in VR experiences, users' perception of the virtual content is not necessarily strongly tied with the real environment that they inhabit [20]. Complementing VR,

* Corresponding author.
E-mail address: kenan.bektas@unisg.ch (K. Bektaş).

[1] https://www.microsoft.com/en-us/hololens
[2] https://varjo.com/products/xr-3
[3] https://www.magicleap.com/magic-leap-2
[4] https://www.vive.com/us/product/vive-pro-eye/specs
[5] https://www.apple.com/apple-vision-pro

https://doi.org/10.1016/j.cag.2024.103909
Received 9 September 2023; Received in revised form 12 March 2024; Accepted 13 March 2024
Available online 16 March 2024

AR HMDs provide users with a hybrid experience that is a synthesis of virtual content that is related to and put into context with the natural, physical scene that a user is situated in. In natural settings (e.g., in daily personal or professional activities), gaze-enabled AR HMDs permit better understanding of the cognitive processes of humans, and provide them with contextually relevant assistance (e.g., visual, audio, or haptic feedback) [6,7,17,21,22].

Striving to support such assistive systems by gaining a better understanding of a user's context in real time, there is a growing interest in studying human activity recognition (HAR), where researchers make use of various (often wearable) sensors that generate streams of data to train and test machine learning models [23]. In recent years, mobile video-based eye trackers are also being used in HAR-research [24–26]. However, in physical environments, the implications of using gaze-enabled AR HMDs for HAR have not been explored in detail. To demonstrate this gap, we present a systematic review of gaze-enabled HAR with mobile eye trackers and HMDs. Then, we present our main contribution: An extended version of the GEAR system [27] that through gaze-enabled HAR provides users with AR feedback and relevant functionalities to their activities. Here, we show the results from our research on using gaze for HAR, where we provide a dataset of three activities across 20 participants. We furthermore explain how our system's data streams are handled through privacy-friendly personal data stores according to the Solid specification [28]; this allows users to retain fine-grained control over sharing of their data and of information about their current predicted activity with the GEAR system and with other services. We compare and discuss the performance of three different gaze-based human activity classifiers, discuss limitations and future work on top of the GEAR approach and system, and our discussion includes a re-casting of the GEAR system as an input to graduate teaching on gaze-enabled AR.

2. Human activity recognition (HAR) from gaze

Research on HAR is relevant for many applications in human–computer interaction and ubiquitous computing [23] that focus on a seamless interaction between human users and interconnected systems. In context-aware computing, the behavior of a system can be adapted to environmental factors (e.g., location) and other factors such as users' expectations, psychophysiology, and activities [11,29]. Since the 1960s (e.g., the seminal work of Yarbus [30]), *eye trackers* are used in studying task-dependent cognitive processes, and many studies have shown that it is possible to decode human activities from their eye movements [31]. These factors can be measured with various sensors (see [23,32] for reviews) which can be integrated into the environment and objects, or may be worn by users. For example, head-mounted (or mobile) eye trackers pave the way towards a pervasive assessment of users' attention, intention, and activities [33].

2.1. Activity recognition with mobile eye tracking

In mobile eye tracking, one of the most influential works on HAR was presented by Bulling and colleagues [14] who followed a five-step procedure, which was also used by others. In an office environment, they collected raw data (Step 1) with a 128 Hz electrooculography (EOG) system from N = 8 participants for recognition of six activities (copying, reading, writing, watching a video, browsing, and resting). After the drift and noise removal (Step 2), they computed a list of eye movement events (Step 3) such as fixations, saccades, and blinks. In feature extraction (Step 4), they calculated 62 features comprising descriptive statistics (e.g., mean, variance, and maximum) of these eye movement events. Lastly, in model training (Step 5), their Support Vector Machine (SVM) model classified six activities with an average precision of 76.1% and recall of 70.5%.

In other studies on HAR, researchers followed experimental procedures that are comparable to [14]. Kiefer and colleagues used in their setup (N = 17), a 30 Hz mobile and video-based eye tracker (SMI Eye Tracking Glasses) to recognize six activities on cartographic maps (free exploration, global search, route planning, focused search, line following, and polygon comparison) [24]. The authors reported a 78% accuracy with an SVM model that was trained with 229 blink-, fixation-, and saccade-based features. Kunze and colleagues used the same eye tracker to detect reading activities of N = 8 participants on five different media with variable amount and orientation of text: a comic book with images, a text book, fashion magazine, a novel, and a newspaper [34]. With saccade- and fixation-based features, their decision tree classifier achieved a 74% accuracy in recognizing the type of document. In an outdoor wayfinding scenario, Alinaghi and colleagues studied the recognition of turning activities (left-, right-, or no-turn) with N = 52 participants who had variable familiarity with the test routes of 0.9 km and 1.3 km [26]. The data was collected with a 200 Hz PupilLabs Invisible eye tracker. They used feature importance ranking (on saccade- and fixation-based features) and tested several models, including SVM and Random Forest, and reached a 91% overall accuracy with Gradient Boosted Decision Trees.

2.2. Activity recognition with mobile eye tracking and AR

Toyama and colleagues presented a gaze-enabled (with SMI Eye Tracking Glasses) AR prototype [35]. This prototype calculates whether the user's eyes converge on a foreground virtual screen or on the real scene (i.e., the background). While the point of convergence dynamically changes, the system analyzes the user's level of engagement in reading a text on the *virtual screen*. The system provides proactive assistance such as highlighting, scrolling, and reminding the user about the last word read. Eight out of 12 participants rated the system as beneficial, however the system was tested only in a reading activity.

Rook et al. studied intent prediction in an immersive environment with N = 30 participants [36]. The 2 Hz data stream included users' head orientation (from Microsoft HoloLens 1) as an approximation to their eye-gaze and auxiliary data from objects of interest, and was used to train a hidden Markov Model (HMM) that yielded an average of 42% precision and 55% recall on three activities (cooking, microwaving, exploring).

With Microsoft HoloLens 2 (HL2), Seelinger and colleagues developed a solution to enable safer navigation in a physical environment by presenting users with context-adaptive visual cues [37]. They trained a deep neural network (DNN) with features such as the angular change of gaze direction and the fixated areas of interest (AOIs), including task-specific features. The solution was not directly addressing the question of HAR with a gaze-enabled AR HMD, but their research provides evidence that a gaze-enabled AR display can promote users' autonomy and safety without compromising their performance.

In a virtual reality setup (i.e., no interaction with physical objects as in AR), David-John and colleagues used an HTC Vive Pro Eye (with 60 to 120 Hz gaze sampling rate) to predict intentions of N = 15 users regarding the selection of items for a given recipe (i.e., onset of interaction) [38]. Their logistic regression model was trained with 61 saccade- and fixation-based features as well as the K-coefficient (see [39]) and showed an above-chance prediction of the onset of interaction.

Recently, Lan and colleagues addressed the creation of synthetic gaze data [40]. Their solution, EyeSyn, synthesizes realistic eye movement data for four activities (read, communicate, browse a static scene, watch a dynamic scene) using generative models and a range of image and video datasets. In an experimental study, the researchers compared the similarity and activity-recognition performance of EyeSyn-synthesized and actual gaze data collected from N = 8 participants. Four participants used a Magic Leap One (30 Hz) and the others used Pupil Labs eye tracker (30 Hz). In all activities, a comparison of the scatterplots showed that the actual and synthetic data had similar spatial characteristics (e.g., reading activity involves horizontal shifts

Table 1
A comparison of selected previous works on HAR with mobile eye tracking devices and AR displays. (p = precision, a = accuracy, r = recall).

Paper	Activities	Sampling rate in Hz (Device)	N =	Stimuli, Display	AR?	HAR Model	Features	Performance	User feedback?
[14]	6: copying, reading, writing, watching a video, browsing, and resting	128 Hz (EOG)	8	Digital (Desktop)	✗	SVM	62 based on: fixations, blinks, saccades, wordbook analysis	$p = 76.1\%$, $a = 70.5\%$	✗
[24]	6: free exploration, global search, route planning, focused search, line following, and polygon comparison	30 Hz (SMI)	17	Digital (Desktop)	✗	SVM	229 based on: fixations, blinks, saccades	$a = 78\%$	✗
[34]	1: reading different documents (comic book, text book, newspaper, magazine, and novel)	30 Hz (SMI)	8	Physical	✗	Decision Tree	Based on: fixations, saccades	$a = 74\%$	✗
[26]	1: wayfinding (turn left right, and no turn)	200 Hz, (Pupil Invisible)	52	Physical	✗	Gradient Boosted Decision Trees (SVM & Random Forest)	28 based on: fixations, saccades (feature importance ranking)	$a = 91\%$	✗
[35]	1: reading	30 Hz (SMI)	12	Physical & Digital (HMD)	✓	No machine learning model	1: convergence of the eyes on foreground (AR) or background (real scene)	8 of 12 participants liked it	✓
[36]	1: intent prediction in smart environments (cooking, microwaving, exploring)	2 Hz (Head-gaze from HL1)	30	Physical & Digital (HMD)	✓	Hidden Markov Model	1: point of interest	$p = 42\%$, $r = 55\%$	✗
[37]	1: navigation in a physical environment	30 Hz (HL2)	15	Physical & Digital (HMD)	✓	Deep Neural Network	2: angular change of gaze direction and fixated areas of interest	participants prefer the solution	✓
[38]	1: prediction of the onset of item during selection of items for a given recipe	60–120 Hz (HTC Vive Pro Eye)	15	Digital (HMD)	✗	Logistic Regression	61 based on: saccades, fixations, K-coefficient)	above chance prediction	✗
[40,41]	4: read, communicate, browse a static scene, watch a dynamic scene	30 Hz (Magic Leap One); 30 Hz (Pupil Labs)	8	Digital (HMD)	✓	Convolutional Neural Network	Based on: fixations, saccades	$a = 90\%$	✓
[27]	3: read, inspect, search	30 Hz (HL2); 200 Hz (PupilCore)	10	Physical & Digital (HMD)	✓	SVM, Random Forest, Extremely Randomized Trees	19 based on: fixations, blinks, saccades	$a = 98.7\%$	✓

of the gaze). A convolutional neural network was trained with the synthetic data and achieved a 90% accuracy in the classification of the activities. Later, these authors also demonstrated that their solution can provide some AR feedback in two activities but with completely virtual stimuli [41]. The developers of EyeSyn claim that it is a viable solution that can address practical constraints of collecting eye movement data and privacy-related concerns [40]. In Table 1 we present a comparison of the selected previous work on HAR with mobile eye tracking devices and AR displays.

2.3. Data privacy in mobile eye tracking

Eye tracking data streams are invaluable sources of information, as they can reveal sensitive attributes of individuals (e.g., gender, age, ethnicity, personality traits, health, sexual preference, affect, task focus) [42,43]. Thus, misusage of data from gaze-enabled devices can interfere with the acceptability of eye tracking by the general public [7, 44] and, most importantly, infringe the privacy of individuals. Now that eye tracking is becoming pervasive, it should be added as a prominent privacy concern in ubiquitous computing technologies [42,45–47]. In recent years, interest in addressing privacy-related issues in eye tracking research (e.g., in the ETRA, UbiComp, and CHI communities) is growing [46]. Privacy-preserving eye tracking can be maintained by physically obscuring the recordings [48], introducing randomized encodings [44] or noise [49] to the data (without compromising their utility), or by several other approaches and regulations [42].

Today we have access to mobile eye trackers, AR HMDs, and machine learning models that can be used in individual steps of HAR starting from data collection to activity recognition and providing feedback or assistance to users. To the best of our knowledge, no previous work provides a HAR solution in gaze-enabled AR HMDs where users perform different activities with physical objects. In Section 3, we introduce GEAR, that combines a (5-step) gaze-enabled activity recognition pipeline with an AR app for activity-based feedback. GEAR additionally integrates the Solid specification [28] for storing and sharing data. This empowers users to control who accesses and manipulates their data. Hence, our work provides a blueprint for a more privacy-friendly approach to the storing, processing, and sharing of gaze data.

3. GEAR: A gaze-enabled AR system for human activity recognition and feedback in ubiquitous computing environments

GEAR has three main components. The first one is an AR application that collects raw gaze data in real time from an AR HMD and renders activity-based feedback on top of a user's visual field (Fig. 1-1.). The second component – Activity Recognition (Fig. 1-2.) – implements a procedure for the real-time recognition of three activities (*Reading* a text, the *Inspection* of an object, and the *Search* for an object) from

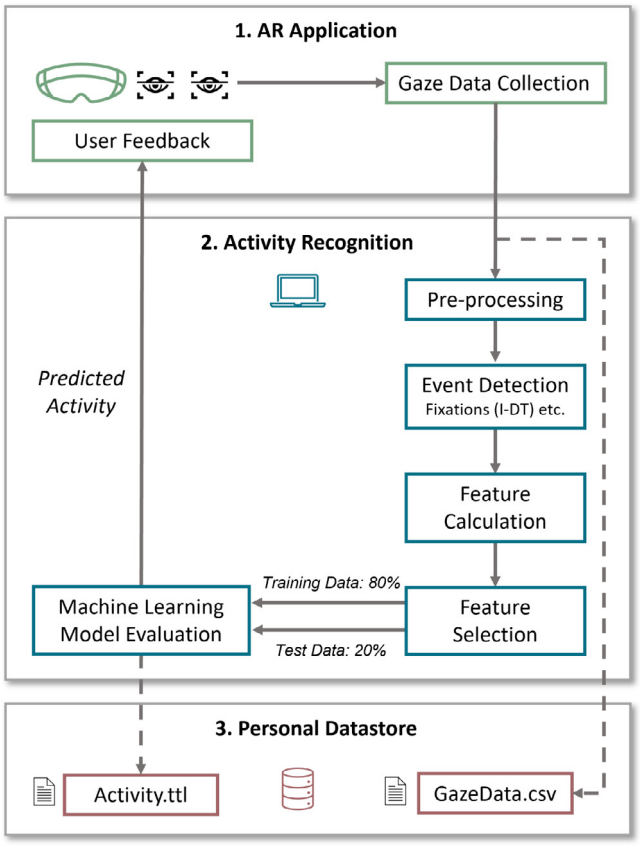

Fig. 1. The components of the GEAR. The collected gaze data is sent from the AR application (1) to the activity recognition component (2). The recognized activity is returned to the AR application which displays appropriate feedback. Both, the collected gaze data and the recognized activity, can be stored in a privacy-friendly personal datastore (3).

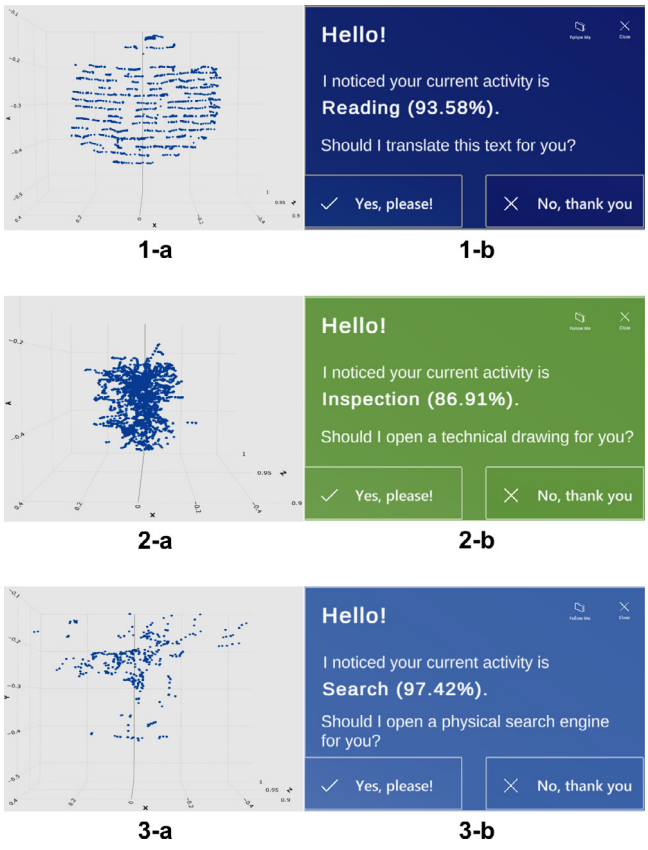

Fig. 2. Example 3D-Plots of the normalized gazeDirection_(x, y, z) data points which are collected from one participant in the Reading (1-a), Inspection (2-a), and Search (3-a) activities. GEAR can display an AR feedback that is relevant to the recognized activity (1-b, 2-b, and 3-b).

collected gaze data. The last component – Personal Datastore (Fig. 1-3.) – implements a solution for the privacy-friendly sharing of such collected data.

3.1. AR application

We developed an AR application for the HL2 with the Unity Game Engine using building blocks from the Mixed Reality Toolkit v2 (MRTK) [50]. The application has two main functions. In *Gaze Data Collection*, it fetches and sends the gaze data to GEAR's *Activity Recognition* component. The *User Feedback* prompts a visual feedback that is relevant to the recognized activity (Fig. 2).

The HL2's eye tracker has a sampling rate of 30 Hz with an accuracy of approximately 1.5° [51]. In Unity, gaze samples can be accessed using the MRTK or the underlying API for the Universal Windows Platform (UWP) [52]. In GEAR, we use the open-source Augmented Reality Eye Tracking Toolkit (ARETT) [53]. ARETT operates on top of the UWP API, reliably delivers gaze samples at a fixed sampling rate (30 Hz) and can be readily included in Unity projects. It also provides a Web interface for storing gaze data in CSV files. The data stream provided by ARETT includes a list of time, gaze, and AOI data, and some auxiliary information. In GEAR's HAR, we make use of the following ARETT data: *eyeDataTimestamp*, *isCalibrationValid*, *gazeHasValue*, *gazeOrigin_(x/y/z)*, *gazeDirection_(x/y/z)*, *gazePoint_(x/y/z)*. The last three vectors are defined in Unity's global coordinate system. When we plotted these different values, we saw that *gazeDirection* might be the most suitable candidate to calculate gaze events and features, respectively. In Fig. 2-(1-a), the gazeDirection data for Reading clearly shows the individual lines of the underlying text.

3.1.1. Gaze data collection

In a controlled study, we collected gaze data with the AR application (Section 3.1) for training and testing HAR models (Section 3.2). The data that we use here consist of an extended version of the data that we previously reported in [27]. In this section, we follow the guideline that is proposed in [54] and report the details of our study.

Participants. We recruited N = 20 participants (eight identified themselves as female) from our lab, with an average age of 29.4 years. Seven participants reported wearing prescription glasses often or all of the time; 16.7% indicated being extremely familiar with AR headsets and 5% with VR headsets, while 33.3% reported not being familiar with AR glasses/VR headsets. Most participants (60%, including all participants with moderate or extreme familiarity) reported that they could imagine wearing an AR headset for up to two hours in their daily lives.

Apparatus and material. The gaze data was collected with the AR application and an HL2 as described in Section 3.1. Furthermore, in the same setup we collected gaze data with a Pupil Core tracker (200 Hz).[6] Our study includes three different physical materials for each activity to-be-recognized. First, a text in English on an A4 paper positioned at a distance of 70 cm, orthogonal to the participants' viewing direction, and covering 40° of their visual field. Second, we used a toy device that was positioned at a distance of approximately 40 cm covering 20° of participants' visual field. Third, we used a small red pin (about the size of a die) and a workpiece-cabinet (1 × 1 m) with three shelves.

[6] The Pupil Core data was not used in this study, however, we make it available as supplementary material for further analysis.

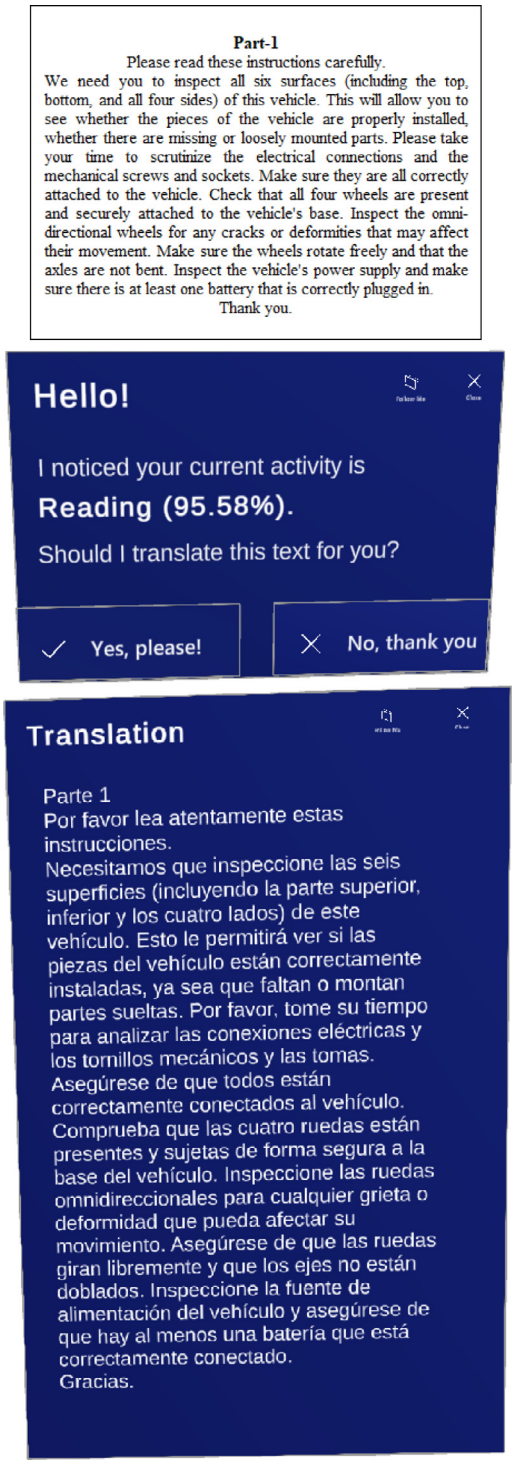

Fig. 3. The text in English that we used in the *Reading* activity and its translation in Spanish.

Procedure. During the experiment, we simulated three realistic tasks that comprise three main activities: *reading* instructions, *inspecting* a device, and *searching* for a missing piece of the device. The procedure started with an introduction of the HL2, comfortably adjusting it on the head of the participant, and running the default 9-point eye calibration. Then, each participant was asked to read some instructions, inspect a device, and search for a missing pin in a cabinet. The reading and inspection activities were performed in a sedentary position. When searching, participants were standing in front of a cabinet. Each activity included a gaze recording of about one minute and participants took short breaks of few seconds between the activities. Finally, we asked each participant to complete a short demographic questionnaire. According to the guidelines of the University of St. Gallen Ethics Committee, all participants gave written consent stating that the collected data may be anonymously used.

3.1.2. AR user feedback

GEAR collects gaze data in chunks of ten seconds (i.e., consecutive and non-overlapping time windows) and sends them to the *Activity Recognition* component (details in Section 3.2). Each chunk includes the columns of *eyeDataTimestamp*, *isCalibrationValid*, *gazeHasValue*, *gazeOrigin_(x/y/z)*, *gazeDirection_(x/y/z)*, *gazePointHit*, and *gazePoint_(x/y/z)*. The data is transmitted via HTTP to a notebook computer executing the *Activity Recognition* component. The response of the *Activity Recognition* component (Fig. 1), including the predicted activity and its probability, is then sent back to the AR application. Thus, to close the activity-recognition and user-feedback loop, a user had to perform any of the three activities for at least ten seconds. In the earlier version of GEAR [27], on the HL2, a panel displays the current activity (see Fig. 2) along with a contextually relevant suggestion, respectively.

We extended the user feedback component of GEAR and provide users with feedback that is relevant to their current activity. In the *Reading* activity, the application suggests a translation of the read text as demonstrated in Fig. 2(1-b). If the user clicks the *Yes* button, the application takes a screenshot of the current scene and sends the frame to an external optical character recognition component as described in [4]. For the translation of the text, we used a local version of the free and open source API LibreTranslate[7] that supports translation between English and thirty other languages. Finally, the translated text is displayed to the user in an AR overlay (Fig. 3). In the *Inspection* activity, the application suggests displaying an interactable 3D model of the inspected device or object. Specifically, for this activity we used a physical Lego model that we assembled (Fig. 4). To help users find the missing item in *Search* activities, the application suggests whether the user wants to turn on a lamp to increase the brightness in the user's field of view (Fig. 5). The HL2 communicates with the lamp using the lamp's W3C WoT Thing Description (TD).[8] W3C WoT TDs are machine-readable and machine-understandable interface descriptions of devices (i.e., Things, such as the lamp in this case) that permit abstracting from the concrete underlying communication protocol (e.g., HTTP or CoAP); Thereby supporting interoperability across Internet of Things (IoT) devices. As an alternative to the switching of a lamp, we propose that GEAR may make use of a search engine for physical devices [55], or may even integrate semantic hypermedia search [56]. However in the current version we did not implement this feature, yet. Contingent on the recognized activity, the user feedback could also be delivered via another modality so as to not obstruct the task at hand. (Spatial) audio might be suitable for tasks like *Reading* where the translation could be delivered using a text-to-speech engine, or *Search* where the user profits from an unhindered view.

3.2. Activity recognition

The activity recognition component of GEAR implements a procedure that is similar to those used in previous HAR research [14,24,26]. The procedure starts with the collection of raw gaze data as described in Section 3.1.1. The remaining steps include preprocessing of the raw data, detecting eye movement events, feature calculation, feature selection, and finally training and evaluation of selected machine learning model(s).

[7] https://github.com/LibreTranslate/LibreTranslate
[8] https://www.w3.org/2019/wot/td

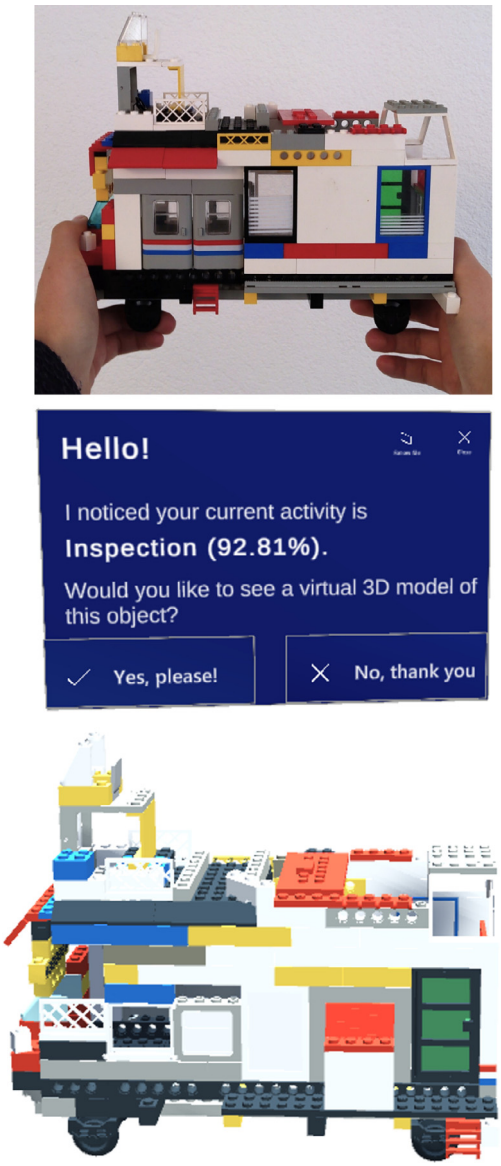

Fig. 4. The physical Lego model that we used in the *Inspection* activity (top). The virtual 3D model is rendered when the user clicks on Yes in the panel.

3.2.1. Pre-processing and event detection

We calculated eye movement events (fixations and blinks) from raw spatio-temporal gaze data (x, y, z, t). We did not compute saccades because of the limited sampling rate of the HL2. In a pre-processing step, before the fixation calculation, we excluded the data where *gazeOrigin* and *gazeDirection* were empty. With the remaining valid data, we calculated the fixations using the I-DT algorithm [16] with a dispersion threshold of 1.6° and a minimum duration of 100 ms. For the dispersion, we used *gazeDirection* as the spatial input, which describes the normal of the gaze, i.e., the gaze direction in the global coordinate system. Our implementation of the fixation detection is based on a tutorial by Pupil Labs,[9] which we adapted to our HL2 setup. Thus, the scripts that we provide in the supplementary material can be used to analyze data collected with the PupilCore [57] or the HL2. The duration of blinks are affected by drowsiness, loss of vigilance, and mental workload [58]. Thus, we took the highly simplifying assumption that all missing gaze data were due to closed eyes/blinks. The Extended

[9] https://github.com/pupil-labs/pupil-tutorials

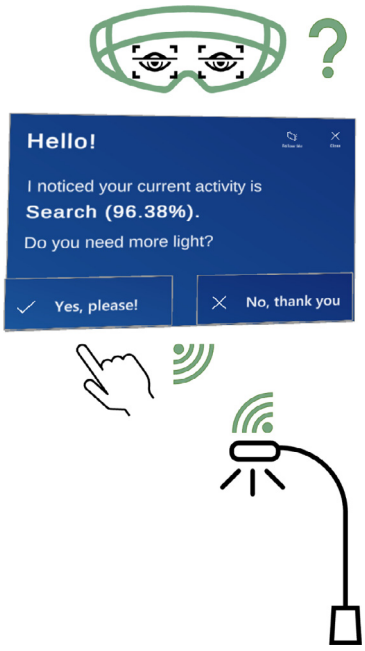

Fig. 5. In the *Search* activity, the application suggest the user to turn on the light from the HL2. When the user clicks on Yes, the application sends a PUT request and changes the sate of the lamp via its API.

Eye Tracking (EET) [59] for the HL2 is supposed to permit sampling rates of up to 90 Hz, which might allow better blink estimation and extending the feature set in general. However, when testing it in our setup, we found that (a) it is not fully compatible with ARETT and (b) it does not provide a stable sampling rate and therefore would lead to reduced data quality, which is why we decided against using EET in GEAR.

3.2.2. Feature calculation

We calculated 19 features from the descriptive statistics (minimum, maximum, mean, variance, and standard deviation) of the fixation duration (5 features) and of the fixation dispersion (5 features), the fixation frequency per second, and the fixation density. Additionally, we calculated the direction of successive fixations for x- and y-directions (2 features). Furthermore, we calculated the following blink-related features: number of blinks, mean, maximum, and minimum blink duration, and the blink rate per second.

In the *Reading* activity, the direction of successive fixations is decisive [60] because the horizontal eye movements show a pattern that goes from left to right but then exhibits a larger jump from right to left when the participant finishes reading one line and proceeds to the next (similar to carriage-return and line-feed). However, in *Inspection* and *Search* activities, the eye movements do not necessarily follow a regular pattern, because typically there is no specific scene layout. In scene viewing (e.g., inspection or search activities), people may scrutinize different parts of the stimuli in variable duration [58], thus the visual and spatial properties of targets and distractors may affect features extracted from fixations.

3.2.3. Feature selection

Before selecting a subset of features for the classification, we formulated several assumptions for each activity. In the *Reading* activity, we hypothesized that the successive fixations of the participants should be aligned with the lines of text they were reading. Furthermore, we expected the fixations to be more scattered and of shorter duration in the *Search* activity, as participants probably looked quickly at many different places. In the *Inspection* activity, we expected fewer fixations

Table 2
Performance benchmark of the three models that are trained with 19 features from the data provided in [27]. The number in each cell presents the accuracy (%) of the model (the first column of the table) for a 5, 10, 15, 20 s time window. The numbers in bold are the accuracy of the best model for a given time window.

Model	5 s	10 s	15 s	20 s
Support vector machine	78.7	93.3	85.0	**100**
Random forest	**93.2**	96.6	94.4	94.9
Extremely randomized trees	89.3	**98.7**	**96.2**	96.6

Table 3
Performance benchmark of the three models that are trained with 19 features from the data documented in Section 3.1.1. The number in each cell presents the accuracy (%) of the model (the first column of the table) for a 5 and 10 s time window. The numbers in bold are the accuracy of the best model for a given time window.

Model	5 s	10 s
Support vector machine	**85.4**	88.3
Random forest	81.0	**89.6**
Extremely randomized trees	81.0	88.3

but with a longer duration, which are less scattered than those in the search. Based on these assumptions we trained our initial classifier (i.e., an SVM as described in 3.2.4) with the following six features: mean fixation duration, maximum fixation duration, variance of the fixation duration, x- and y-fixation direction and the fixation density per area. We then normalized these features and trained three different classifiers to predict the activities.

3.2.4. Model evaluation

We evaluated our solution in two stages. First, we trained three different machine learning models with the selected features using the data provided in [27]. The outcome of this initial evaluation is a performance benchmark of the *Support Vector Machine*, *Random Forest Classifier*, and *Extremely Randomized Trees Classifier* that is presented in Table 2. In the second stage, we extended the data set (see Section 3.1.1 for the details) and documented a new performance benchmark with these three classifiers.

Support vector machine (SVM) classifier. First, since SVMs were used in most of the related work on HAR documented in Section 2, we applied an SVM classifier to the selected features. We implemented the SVM using the `sklearn.svm.SVC` function in the Python package *scikit-learn*[10] with ***L***inear, ***P***olynomial, Gaussian ***R***adial Basis Function, and ***S***igmoid kernels. We split the data in 80% training and 20% testing. As the recorded data per participant and activity was around one minute long, we trained the model with different time windows.

The results of the first stage evaluation showed that, with a ten-seconds window the first three kernels to all predict with an accuracy of 93.3% and the Sigmoid kernel to achieve an accuracy of 30%. All kernels achieved lower prediction accuracies when using window sizes of five (L: 65.8%, P: 78.7%, R: 72.1%, S: 26.2%) and 15 s (L: 85%, P: 85%, R: 85%, S: 20%). A window size of 20 s, however, resulted in an accuracy of up to 100% (L: 86.6%, P: 93.3%, R: 100%, S: 33.3%) which might indicate overfitting to the small sample size.

Random forest (RF) classifier. We developed a model with the *sklearn.ensemble.RandomForest* function from the above-mentioned *scikit-learn* library. To select the best subset of the precomputed features for this classifier, forward and backward feature selection was performed. In the first stage, the results of our experiments with the data provided in [27] show that the Random Forest classifier outperforms all SVM kernels by at least 4 percentage-points regarding accuracy. The best result (96.6% accuracy) was achieved using all of the 19 possible features. To be comparable to the SVM approach, a window of 10 s was used for the feature calculation. Other window sizes did not further improve the result (93.2% for 5 s, 94.4% for 15 s, 94.9% for 20 s).

Extremely randomized trees (ET) classifier. Finally, we applied *sklearn.ensemble.ExtraTreesClassifier* while using all 19 features and a window size of 10 s. This classifier achieved an accuracy of 98.7% on our test data. As with RF, other window sizes did not improve accuracy (89.26% for 5 s, 96.25% for 15 s and 96.6% for 20 s). The ET classifier improves on the accuracy of the RF classifier while also being significantly faster, requiring 0.639 s ± 0.035 s versus 0.844 s ± 0.020 s to classify 149 samples, i.e., 24% less time per sample. The difference between the ET and RF classifiers can be characterized as follows: While RF computes the most discriminative decision boundary for each feature, ET chooses the most discriminative boundary among several random boundaries and with different features subsets [61]. As a consequence, variance is reduced, mitigating overfitting of the classifiers. Furthermore, by choosing the decision boundary randomly, the ET is computationally less expensive, leading to faster execution times.

3.2.5. Second stage model evaluation

In the second stage, we used the extended dataset (see Section 3.1.1 for the details) and only looked at five and ten seconds windows as we considered that longer window size are practically not fast enough for providing users with contextual AR feedback for many activities. With the five-seconds window the Polynomial kernel of the SVM classifier reached the most accurate estimation (L: 82.9%, P: 85.4%, R: 82.9%, S: 15.8%). The confusion matrices are presented on Fig. 6. With the ten-seconds window, again, the Polynomial kernel reached the most accurate estimation (L: 87.0%, P: 88.3%, R: 87.0%, S: 32.4%). Different from the results of the first stage evaluation, in the second stage evaluation, the RF classifier reached the best overall estimation accuracy (89.6%) with the ten-second window, and the ET classifier reached the same accuracy (88.3%) as the SVM with the Polynomial kernel. Our findings from the second stage are summarized in Table 3. The confusion matrices for the ten-seconds window are presented on Fig. 7.

3.3. Decentralized datastore

Being cognizant of the sensitivity of gaze data and of the detected activities, we integrated GEAR with Solid. Solid is a specification proposed by the creator of the Web Tim Berners-Lee [28]. Solid aims at decentralizing the Web by returning the ownership of data to its creators rather than keeping it in silos owned by tech companies. To do so, Solid applications are decoupled from the data they use. Hence, users producing data keep it in a (personal) data store called *Pod* and assign and revoke permissions to specific applications or even users. A user can have one or many *Pods* containing personal and non-personal information; and such *Pods* can be self-hosted or hosted by a trusted *Pod* provider. Thus, Solid applications that access user data in a *Pod* do not keep a copy of this data; such applications only access (and possibly modify) the data transiently, where access rights are checked on each access. A Pod is implemented as a Web server with standardized authentication, authorization, and sharing procedures. Moreover, being a specification for the Web, Solid takes advantage of standardized vocabularies expressed in RDF (Resource Description Framework)[11] such as the Access Control List (ACL) schema,[12] used for granting read, write, and append rights.

We set up an instance of the Solid community server,[13] which is an open-source implementation of the Solid specification, developed and maintained by the research community. Moreover, we added to our AR

[10] https://scikit-learn.org

[11] https://www.w3.org/RDF/

[12] https://www.w3.org/2001/04/ACLS/Schema and https://solid.github.io/web-access-control-spec/

[13] https://github.com/CommunitySolidServer/CommunitySolidServer

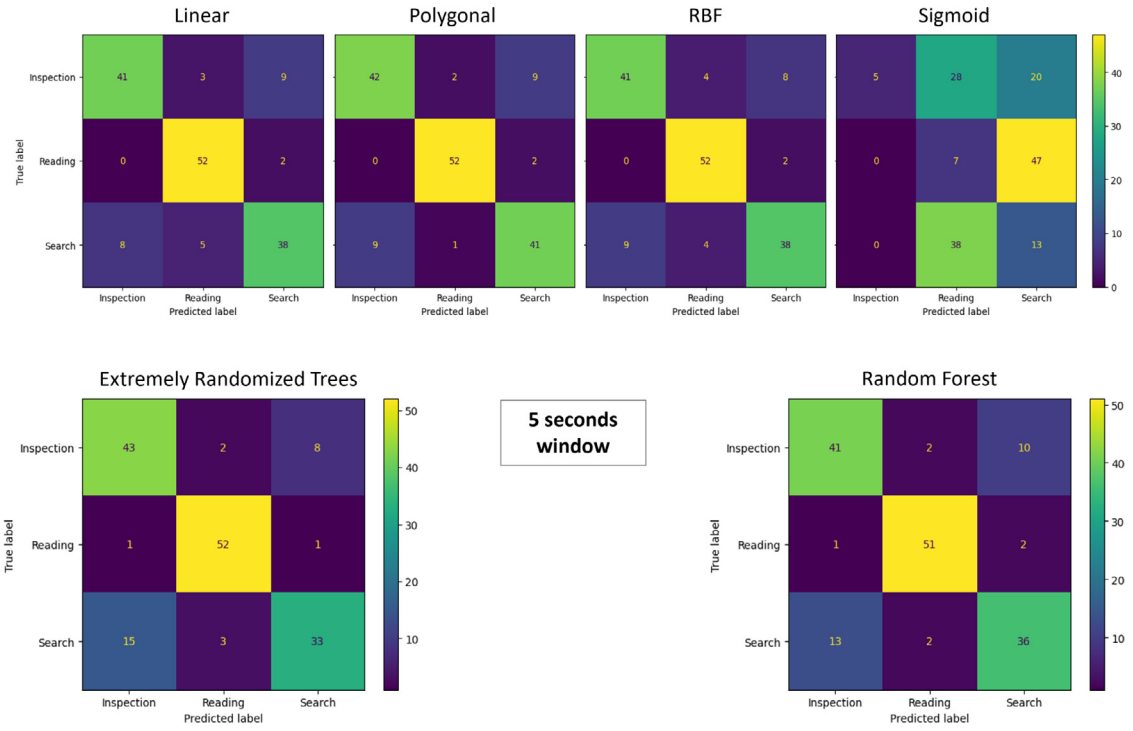

Fig. 6. Confusion matrices of the four different kernels of support vector machine classifier (top row), the extremely randomized trees, and random forest classifiers for the five-seconds window.

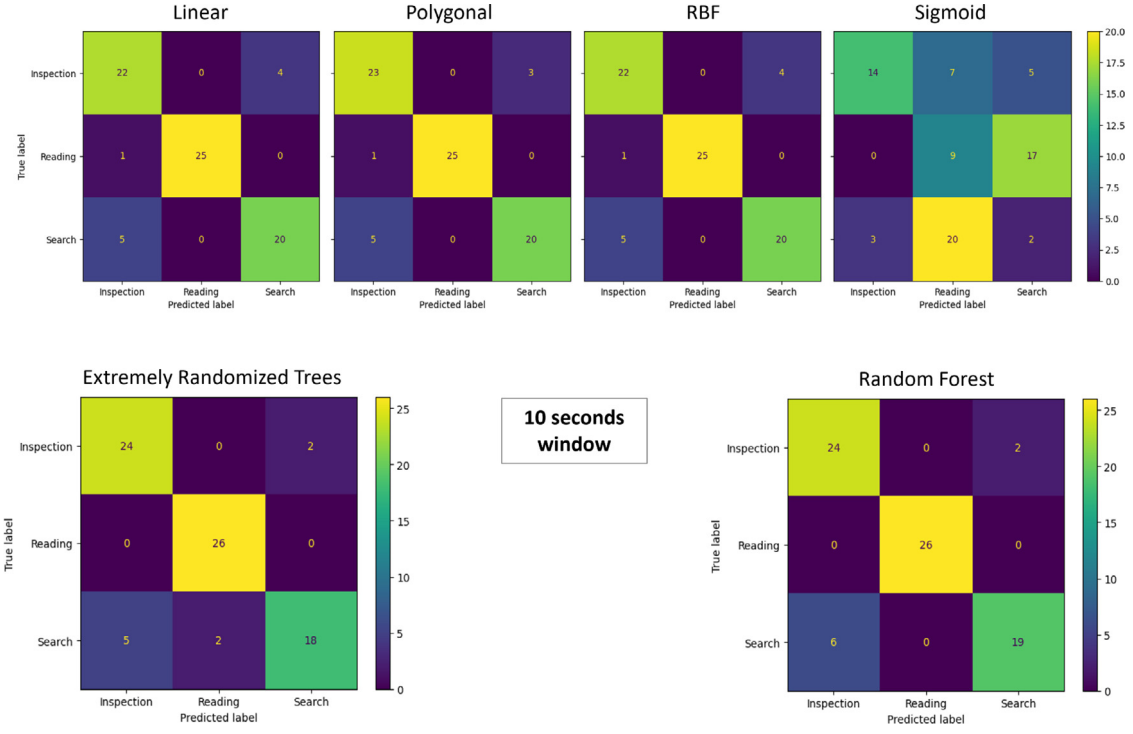

Fig. 7. Confusion matrices of the four different kernels of support vector machine classifier (top row), the extremely randomized trees, and random forest classifiers for the ten-seconds window.

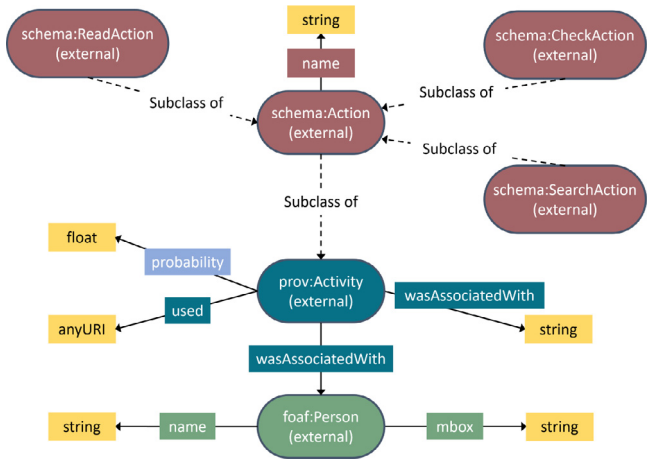

Fig. 8. Describing a GEAR detected *Activity* using well known data schemas such as FOAF for person, schema.org for actions, and the PROV-O for the activity provenance.

application the capability of writing the collected gaze data directly into a user's Pod in the form of a CSV file (see Fig. 1-GazeData.csv). Likewise, the *Activity Recognition* component stores the recognized activity in the user's Pod as an RDF file (see Fig. 1-Activity.ttl), which is expressed using well-known schemas [62]. On the AR application, a user can decide who to share their gaze and activity data with by indicating the WebID of a Solid application or of a user.

To describe an activity recognized by GEAR in a structured manner, we propose to use schema.org,[14] the PROV-O ontology,[15] and the Friend of a Friend (FOAF) ontology.[16] Fig. 8 shows the proposed structure to describe an activity detected by GEAR. The *Person* concept is imported from the FOAF ontology, together with the name and mailbox (i.e., mbox) properties. The *Activity* concept is imported from the PROV-O ontology to describe provenance of an activity recognized by GEAR (including end time and person that created/used the activity). Moreover such *Activity* will also be an instance of the corresponding granular type of action specified by the schema.org concepts: *ReadAction*, *CheckAction*, or *SearchAction*; in which the schema.org *CheckAction* is semantically equivalent to the *Inspect* activity in GEAR.

4. Discussion

4.1. Model and system performance

In this work, we trained three commonly used machine learning models (i.e., SVM, RF, and ET) for predicting human activities from users' eye movements (see Section 3.2). For the activity recognition, we followed a five-step procedure starting with the collection of raw data and detection of eye movement events (e.g., fixations and blinks) in this data. Thus, the overall performance of activity recognition naturally depends on the accuracy of eye movement event detection. The performance evaluation of the event detection is beyond the scope of the present work. We consider that future research on gaze-enabled activity recognition can significantly benefit from existing best practices about acquisition of raw data [63] and selection of event detection algorithms [64–66].

In order to have a systematic comparison among the three models, in two stages, we trained them with data from N = 10 and N = 20 individuals and created a benchmark that is based on their accuracy in predicting the underlying activity (i.e., reading, inspection, searching). The results of our benchmark show that all three models allow an accurate and close-to-real-time (i.e., <= 10 s) prediction. GEAR provides its user with AR feedback that is relevant to their activity at regular time intervals. The length of these intervals mainly depends on the duration of the time window that is required for the prediction of user's activity. For instance, in the current implementation of GEAR we set this to 10 s (Section 3.1.2). However the results of the second stage evaluation (Section 3.2.5) show that the SVM classifier can accurately predict the current activity within 5 s. To assess the latency of activity recognition, we ran GEAR for ten minutes with the SVM classifier and the ten-second window. The results of this assessment showed that the total duration of activity recognition (i.e., from sending the gaze data to the activity recognition component until the arrival of the detected activity at the HL2) and the additional time required for providing the subsequent AR feedback was on average 150.05 ms (SD = 30.43 ms). Overall, the results of our performance benchmark indicate that GEAR can predict human activities with a gaze-enabled AR headset and provide them with feedback.

4.2. Limitations and future applications

In our experimental setup, we used GEAR in the recognition of only three activities and performing these activities in a particular order constitutes a scenario that is similar to a real industrial workflow (e.g., inspection, maintenance, or repair operations): First, read the instructions, then inspect the device and search for the missing piece. However, to receive some feedback that is relevant to their activity, the users had to deliberately continue performing that activity without an interruption.

In our daily social and professional life, we typically perform many activities during which our mental state can change from losing attention (e.g., a state of *mind-wandering* [67]) to fully engaging with the current activity (i.e., the state of *flow* [68]). Thus, in a next step, GEAR can have components that are dedicated to estimating users' current activity among a long list of activities, their abilities (e.g., skills and expertise) and mental state (e.g., stress, cognitive load, attention) and provide them with feedback in real industrial environments (e.g., in [21,69]).

Recent research shows that eye gaze is a useful information source for multimodal and interactive AI assistance systems [70,71] that can be implemented in modern AR headsets. Therefore, we consider that gaze-enabled activity recognition (e.g., GEAR) can be a beneficial extension of such systems and provide users with contextually relevant feedback and assistance. Specifically, HAR from gaze data could be incorporated into digital companion systems to perceive the state of the environment from a literally user-focused view. Digital Companions are smart agents capable of assisting and protecting their users in a proactive and in a reactive manner [72]. To do so, they utilize technologies such as connected devices (e.g., sensors), and computer vision [73] to perceive the current state of an environment, so that suitable assistance can be computed and delivered to a user. In addition to contributing the current user activity and especially when combining with computer vision, such systems enable fascinating applications: For instance, knowledge of the user's current activity along with information on the objects that the user gazes at enables opportunistic behavior suggestions (cf. [74]); or it could steer the user's attention towards currently relevant environmental artifacts (cf. [75]; in the future, this might even allow for coordinating interactions with the environment.

In the next version of GEAR, the activity recognition component can be implemented in C# instead of Python so that it runs directly on the HL2. The SharpLearning[17] library for C# provides implementations for RandomForest and Extremely Randomized Trees classifiers, while the event detection algorithms can be implemented analogously to

[14] https://schema.org/
[15] https://www.w3.org/TR/prov-o/
[16] http://xmlns.com/foaf/0.1/
[17] https://github.com/mdabros/SharpLearning

the Python implementation. Alternatively, we consider testing IronPython[18] that allows developers to use the .NET framework API from Python. Using C#'s concurrency support, the activity classifier, data collection, and feature extraction can be executed in parallel. We will also extend the user feedback part with other modalities (e.g., audio cues, or speech interfaces), relevant Web-based services (e.g., Object Detection [73,75] and Identification [5]) and by defining dynamic AOIs to make it more useful for users. Additionally, it is possible to integrate the Extended Eye Tracking API [59] in GEAR for collecting gaze data on the HL2 with a higher sampling rate. This would allow us to detect a broader range of eye movement events and test an extended list of features with our system in the recognition of other activities.

While a user evaluation of GEAR is beyond the scope of the present work, future research should study how well the activity recognition models work across users with diverse backgrounds. Furthermore, the short- and long-term implications of assistive systems (and digital companions), such as GEAR, on user behavior and cognitive abilities should be investigated.

4.3. GEAR as teaching material

The work presented in this paper was conducted in the context of a graduate course on Ubiquitous Computing, where one of three assignments focused on Gaze-enabled AR. In addition to the code and data that is required to reproduce the results of this paper, we furthermore provide all teaching materials and their sources for reuse by others.[19] In the assignment, students gained experience working with gaze-enabled AR and different machine learning models by building on top of GEAR components. In Task 1 of this assignment, students worked on offline HAR using a Jupyter notebook that they were required to extend and improve to analyze our gaze dataset; in Task 2, they were required to extend a provided software framework for the HL2 to enable the close-to-real-time classification of user activities with the help of the model from Task 1, as described in this paper. Finally, Task 3 focused on the provisioning of AR feedback to the user, where we required students to provide contextual suggestions to users using *any* feedback modality (simple audio, spatial audio, visual feedback, etc.).

In a subsequent assignment, students created a new version of the gaze-enabled activity recognition pipeline using Web-accessible personal Pods with the Solid specification, where access rights were granted to other users (or applications) based on their WebID (concretely: Solid OpenID Connect[20]), together with Solid Access Control Lists. This new pipeline allows users to control their own gaze data, granting or restricting access rights at any time. Through this assignment, we aim at emphasizing that precisely because of the fine-grained insights that can be derived from a person's gaze data (in our case the performed activity, but as Kroger et al. [43] point out, even drug consumption and cultural background can be derived), it is of utmost importance to be aware of the responsibility that implementing eye-tracking technologies implies. Practitioners should only collect, process and store gaze data given previous informed consent. Moreover, practitioners should be aware and opt for privacy enabling technologies (such as Solid) that might be implemented when working with gaze data.

5. Conclusions

In this article, we presented GEAR, a gaze-enabled human activity recognition system on an AR HMD that provides users with feedback relevant to their current activity. GEAR makes use of the Solid standard to permit fine-grained access control to user gaze data. We positioned GEAR with respect to related work in the domain of mobile eye tracking and evaluated its activity recognition performance in two stages. First, we used the data from N = 10 participants that we collected in our previous work [27]. At this stage, using an Extremely Randomized Trees model, GEAR achieved an accuracy of 98.7% when recognizing three different activities in real time, using a window size of ten seconds of gaze data. We extended this initial dataset with ten additional participants and tested the same three classifiers. With a Random Forest classifier, GEAR achieved on average 89.6% accuracy when recognizing three different activities, again with a window size of ten seconds. The source code of GEAR and anonymized datasets can be used for reproducing and extending our findings and as teaching materials. In the future, interactive AI assistance systems can benefit from gaze-enabled activity recognition features (e.g., GEAR) to provide users with contextually relevant assistance in AR headsets.

[18] https://ironpython.net/

[19] https://github.com/Interactions-HSG/GEAR

[20] https://solidproject.org/TR/oidc

CRediT authorship contribution statement

Kenan Bektaş: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jannis Strecker:** Conceptualization, Data curation, Formal analysis, Investigation, Software, Visualization, Writing – original draft, Writing – review & editing. **Simon Mayer:** Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing. **Kimberly Garcia:** Conceptualization, Software, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

https://github.com/Interactions-HSG/GEAR.

Acknowledgments

We thank our participants for providing us with their data, Stefan Bucher for providing us with realistic physical props for testing the GEAR system, Mathias Lenz and Markus Stolze for the virtual 3D model. This work was funded by the Swiss Innovation Agency Innosuisse (#48342.1 IP-ICT) and the Basic Research Fund of the University of St.Gallen, Switzerland.

References

[1] Azuma R, Baillot Y, Behringer R, Feiner S, Julier S, MacIntyre B. Recent advances in augmented reality. IEEE Comput Graph Appl 2001;21(6):34–47. http://dx.doi.org/10.1109/38.963459.

[2] Billinghurst M, Clark A, Lee G. A survey of augmented reality. Found Trends Hum-Comput Interact 2015;8(2–3):73–272. http://dx.doi.org/10.1561/1100000049, URL http://www.nowpublishers.com/article/Details/HCI-049.

[3] Sutherland IE. A head-mounted three dimensional display. In: Proceedings of the December 9–11, 1968, fall joint computer conference, part I. San Francisco, California: ACM Press; 1968, p. 757–64. http://dx.doi.org/10.1145/1476589.1476686, URL http://portal.acm.org/citation.cfm?doid=1476589.1476686.

[4] Strecker J, García K, Bektaş K, Mayer S, Ramanathan G. SOCRAR: Semantic OCR through Augmented reality. In: Proceedings of the 12th international conference on the Internet of Things. Delft Netherlands: ACM; 2022, p. 25–32. http://dx.doi.org/10.1145/3567445.3567453, URL https://dl.acm.org/doi/10.1145/3567445.3567453.

[5] Strecker J, Akhunov K, Carbone F, García K, Bektaş K, Gomez A, et al. MR object identification and interaction: Fusing object situation information from heterogeneous sources. Proc ACM Interact Mob Wearable Ubiquitous Technol 2023;7(3):26. http://dx.doi.org/10.1145/3610879.

[6] Grubert J, Langlotz T, Zollmann S, Regenbrecht H. Towards pervasive augmented reality: Context-awareness in augmented reality. IEEE Trans Vis Comput Graphics 2017;23(6):1706–24. http://dx.doi.org/10.1109/TVCG.2016.2543720, URL http://ieeexplore.ieee.org/document/7435333/.

[7] Orlosky J, Sra M, Bektaş K, Peng H, Kim J, Kos'myna N, et al. Telelife: The future of remote living. Front Virtual Real 2021;2:763340. http://dx.doi.org/10.3389/frvir.2021.763340, URL https://www.frontiersin.org/articles/10.3389/frvir.2021.763340/full.

[8] Plopski A, Hirzle T, Norouzi N, Qian L, Bruder G, Langlotz T. The eye in extended reality: A survey on gaze interaction and eye tracking in head-worn extended reality. ACM Comput Surv 2022;55(3):1–39. http://dx.doi.org/10.1145/3491207, URL.

[9] Weiser M. The computer for the 21st century. SIGMOBILE Mob Comput Commun Rev 1999;3(3):3–11. http://dx.doi.org/10.1145/329124.329126, URL https://doi.org/10.1145/329124.329126.

[10] Jacob R, Stellmach S. What you look at is what you get: Gaze-based user interfaces. Interactions 2016;23(5):62–5. http://dx.doi.org/10.1145/2978577, URL https://dl.acm.org/doi/10.1145/2978577.

[11] Vertegaal R. Attentive user interfaces. Commun ACM 2003;46(3):3263733. http://dx.doi.org/10.1145/3263733, URL https://dl.acm.org/doi/10.1145/3263733.

[12] Zhai S, Morimoto C, Ihde S. Manual and gaze input cascaded (MAGIC) pointing. In: Proceedings of the SIGCHI conference on human factors in computing systems. Pittsburgh, Pennsylvania, United States: ACM; 1999, p. 246–53. http://dx.doi.org/10.1145/302979.303053, URL http://portal.acm.org/citation.cfm?doid=302979.303053.

[13] Bektaş K, Çöltekin A, Krüger J, Duchowski AT. A testbed combining visual perception models for geographic gaze contingent displays. Eurographics Conference on Visualization (EuroVis) - Short Papers, 2015, http://dx.doi.org/10.2312/EUROVISSHORT.20151127, URL https://diglib.eg.org/handle/10.2312/eurovisshort.20151127.067-071.

[14] Bulling A, Ward JA, Gellersen H, Tröster G. Eye movement analysis for activity recognition using electrooculography. IEEE Trans Pattern Anal Mach Intell 2011;33(4):741–53. http://dx.doi.org/10.1109/TPAMI.2010.86, URL http://ieeexplore.ieee.org/document/5444879/.

[15] Bektaş K, Çöltekin A, Krüger J, Duchowski AT, Fabrikant SI. GeoGCD: Improved visual search via gaze-contingent display. In: Proceedings of the 11th ACM symposium on eye tracking research & applications. Denver Colorado: ACM; 2019, p. 1–10. http://dx.doi.org/10.1145/3317959.3321488, URL https://dl.acm.org/doi/10.1145/3317959.3321488.

[16] Salvucci DD, Goldberg JH. Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 symposium on eye tracking research & applications. ETRA '00, New York, NY, USA: ACM; 2000, p. 71–8. http://dx.doi.org/10.1145/355017.355028, URL https://doi.org/10.1145/355017.355028.

[17] Bektaş K. Toward a pervasive gaze-contingent assistance system: Attention and context-awareness in augmented reality. In: ACM symposium on eye tracking research and applications. ETRA '20 adjunct, New York, NY, USA: ACM; 2020, http://dx.doi.org/10.1145/3379157.3391657, URL https://doi.org/10.1145/3379157.3391657.

[18] Milgram P, Kishino F. A taxonomy of mixed reality visual displays. IEICE Trans Inf Syst 1994;77(12):1321–9.

[19] Keshava A, Nezami FN, Neumann H, Izdebski K, Schüler T, König P. Just-in-time: Gaze guidance behavior while action planning and execution in VR. 2021, http://dx.doi.org/10.1101/2021.01.29.428782, preprint: bioRxiv.

[20] Bektaş K, Thrash T, van Raai MA, Künzler P, Hahnloser R. The systematic evaluation of an embodied control interface for virtual reality. PLoS One 2021;16(12). http://dx.doi.org/10.1371/journal.pone.0259977.

[21] Bektaş K, Strecker JR, Mayer S, Stolze M. Etos-1: Eye tracking on shopfloors for user engagement with automation. In: AutomationXP22: Engaging with automation, cHI'22. 2022, URL http://www.alexandria.unisg.ch/266339/.

[22] Pfeuffer K, Abdrabou Y, Esteves A, Rivu R, Abdelrahman Y, Meitner S, et al. ARtention: A design space for gaze-adaptive user interfaces in augmented reality. Comput Graph 2021;95:1–12. http://dx.doi.org/10.1016/j.cag.2021.01.001, URL https://linkinghub.elsevier.com/retrieve/pii/S0097849321000017.

[23] Bulling A, Blanke U, Schiele B. A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput Surv 2014;46(3):1–33. http://dx.doi.org/10.1145/2499621, URL https://dl.acm.org/doi/10.1145/2499621.

[24] Kiefer P, Giannopoulos I, Raubal M. Using eye movements to recognize activities on cartographic maps. In: Proceedings of the 21st ACM SIGSPAtIAL international conference on advances in geographic information systems. New York, NY, USA: ACM; 2013, p. 488–91. http://dx.doi.org/10.1145/2525314.2525467, URL https://dl.acm.org/doi/10.1145/2525314.2525467.

[25] Braunagel C, Kasneci E, Stolzmann W, Rosenstiel W. Driver-activity recognition in the context of conditionally autonomous driving. In: 2015 IEEE 18th international conference on intelligent transportation systems. Gran Canaria, Spain: IEEE; 2015, p. 1652–7. http://dx.doi.org/10.1109/ITSC.2015.268, URL http://ieeexplore.ieee.org/document/7313360/.

[26] Alinaghi N, Kattenbeck M, Golab A, Giannopoulos I. Will you take this turn? Gaze-based turning activity recognition during navigation. In: 11th international conference on geographic information science (GIScience 2021) - Part II. Leibniz international proceedings in informatics (lIPIcs), 2021, p. 5:1–5:16. http://dx.doi.org/10.4230/LIPIcs.GIScience.2021.II.5, URL https://drops.dagstuhl.de/opus/volltexte/2021/14764.

[27] Bektaş K, Strecker J, Mayer S, Garcia K, Hermann J, Jenß KE, et al. GEAR: Gaze-enabled augmented reality for human activity recognition. In: 2023 symposium on eye tracking research and applications. Tubingen Germany: ACM; 2023, p. 1–9. http://dx.doi.org/10.1145/3588015.3588402, URL https://dl.acm.org/doi/10.1145/3588015.3588402.

[28] Sambra AV, Mansour E, Hawke S, Zereba M, Greco N, Ghanem A, et al. Solid: A platform for decentralized social applications based on linked data. Tech Rep, MIT CSAIL & Qatar Computing Research Institute; 2016.

[29] Bulling A, Zander TO. Cognition-aware computing. IEEE Pervasive Comput 2014;13(3):80–3. http://dx.doi.org/10.1109/MPRV.2014.42, URL http://ieeexplore.ieee.org/document/6850240/.

[30] Yarbus AL. Eye movements and vision. Boston, MA: Springer US; 1967, http://dx.doi.org/10.1007/978-1-4899-5379-7, URL http://link.springer.com/10.1007/978-1-4899-5379-7.

[31] Borji A, Itti L. Defending Yarbus: Eye movements reveal observers' task. J Vis 2014;14(3):29. http://dx.doi.org/10.1167/14.3.29, URL http://jov.arvojournals.org/Article.aspx?doi=10.1167/14.3.29.

[32] Cornacchia M, Ozcan K, Zheng Y, Velipasalar S. A survey on activity detection and classification using wearable sensors. IEEE Sens J 2017;17(2):386–403. http://dx.doi.org/10.1109/JSEN.2016.2628346, URL http://ieeexplore.ieee.org/document/7742959/.

[33] Bulling A, Gellersen H. Toward mobile eye-based human-computer interaction. IEEE Pervasive Comput 2010;9(4):8–12. http://dx.doi.org/10.1109/MPRV.2010.86, URL http://ieeexplore.ieee.org/document/5586690/.

[34] Kunze K, Utsumi Y, Shiga Y, Kise K, Bulling A. I know what you are reading: Recognition of document types using mobile eye tracking. In: Proceedings of the 2013 international symposium on wearable computers. New York, NY, USA: ACM; 2013, p. 113–6. http://dx.doi.org/10.1145/2493988.2494354, URL https://dl.acm.org/doi/10.1145/2493988.2494354.

[35] Toyama T, Sonntag D, Orlosky J, Kiyokawa K. Attention engagement and cognitive state analysis for augmented reality text display functions. In: Proceedings of the 20th international conference on intelligent user interfaces. Atlanta Georgia USA: ACM; 2015, p. 322–32. http://dx.doi.org/10.1145/2678025.2701384, URL https://dl.acm.org/doi/10.1145/2678025.2701384.

[36] Rook K, Witt B, Bailey R, Geigel J, Hu P, Kothari A. A study of user intent in immersive smart spaces. In: 2019 IEEE international conference on pervasive computing and communications workshops (perCom workshops). Kyoto, Japan: IEEE; 2019, p. 227–32. http://dx.doi.org/10.1109/PERCOMW.2019.8730692, URL https://ieeexplore.ieee.org/document/8730692/.

[37] Seeliger A, Weibel RP, Feuerriegel S. Context-adaptive visual cues for safe navigation in augmented reality using machine learning. Int J Hum-Comput Interact 2022;1–21. http://dx.doi.org/10.1080/10447318.2022.2122114, URL https://www.tandfonline.com/doi/full/10.1080/10447318.2022.2122114.

[38] David-John B, Peacock C, Zhang T, Murdison TS, Benko H, Jonker TR. Towards gaze-based prediction of the intent to interact in virtual reality. In: ACM symposium on eye tracking research and applications. Virtual Event Germany: ACM; 2021, p. 1–7. http://dx.doi.org/10.1145/3448018.3458008, URL https://dl.acm.org/doi/10.1145/3448018.3458008.

[39] Krejtz K, Duchowski A, Krejtz I, Szarkowska A, Kopacz A. Discerning ambient/focal attention with coefficient K. ACM Trans Appl Percept 2016;13(3):1–20. http://dx.doi.org/10.1145/2896452, URL https://dl.acm.org/doi/10.1145/2896452.

[40] Lan G, Scargill T, Gorlatova M. EyeSyn: Psychology-inspired eye movement synthesis for Gaze-based activity recognition. In: 2022 21st ACM/IEEE international conference on information processing in sensor networks (IPSN). Milano, Italy: IEEE; 2022, p. 233–46. http://dx.doi.org/10.1109/IPSN54338.2022.00026, URL https://ieeexplore.ieee.org/document/9826020/.

[41] Scargill T, Lan G, Gorlatova M. Demo abstract: Catch my eye: Gaze-based activity recognition in an augmented reality art gallery. In: 2022 21st ACM/IEEE international conference on information processing in sensor networks (IPSN). Milano, Italy: IEEE; 2022, p. 503–4. http://dx.doi.org/10.1109/IPSN54338.2022.00052, URL https://ieeexplore.ieee.org/document/9825976/.

[42] Liebling DJ, Preibusch S. Privacy considerations for a pervasive eye tracking world. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication. UbiComp '14 adjunct, New York, NY, USA: ACM; 2014, p. 1169–77. http://dx.doi.org/10.1145/2638728.2641688, URL https://doi.org/10.1145/2638728.2641688.

[43] Kröger JL, Lutz OH-M, Müller F. What does your gaze reveal about you? On the privacy implications of eye tracking. In: Friedewald M, Önen M, Lievens E, Krenn S, Fricker S, editors. Privacy and identity management. data for better living: AI and privacy. 576, Cham: Springer International Publishing; 2020, p. 226–41. http://dx.doi.org/10.1007/978-3-030-42504-3_15, URL http://link.springer.com/10.1007/978-3-030-42504-3_15, Series Title: IFIP Advances in Information and Communication Technology.

[44] Bozkir E, Ünal AB, Akgün M, Kasneci E, Pfeifer N. Privacy preserving gaze estimation using synthetic images via a randomized encoding based framework. In: ACM symposium on eye tracking research and applications. Stuttgart Germany: ACM; 2020, p. 1–5. http://dx.doi.org/10.1145/3379156.3391364, URL https://dl.acm.org/doi/10.1145/3379156.3391364.

[45] Langheinrich M. Privacy by design — Principles of privacy-aware ubiquitous systems. In: Goos G, Hartmanis J, van Leeuwen J, Abowd GD, Brumitt B, Shafer S, editors. Ubicomp 2001: Ubiquitous computing. 2201, Berlin, Heidelberg: Springer Berlin Heidelberg; 2001, p. 273–91. http://dx.doi.org/10.1007/3-540-45427-6_23, URL http://link.springer.com/10.1007/3-540-45427-6_23.

[46] Katsini C, Abdrabou Y, Raptis GE, Khamis M, Alt F. The role of eye gaze in security and privacy applications: Survey and future HCI research directions. In: Proceedings of the 2020 CHI conference on human factors in computing systems. CHI '20, New York, NY, USA: ACM; 2020, p. 1–21. http://dx.doi.org/10.1145/3313831.3376840, URL https://doi.org/10.1145/3313831.3376840.

[47] Gressel C, Overdorf R, Hagenstedt I, Karaboga M, Lurtz H, Raschke M, et al. Privacy-aware eye tracking: Challenges and future directions. IEEE Pervasive Comput 2023;22(1):95–102. http://dx.doi.org/10.1109/MPRV.2022.3228660.

[48] Steil J, Koelle M, Heuten W, Boll S, Bulling A. PrivacEye: Privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features. In: Proceedings of the 11th ACM symposium on eye tracking research & applications. New York, NY, USA: ACM; 2019, p. 1–10. http://dx.doi.org/10.1145/3314111.3319913, URL https://dl.acm.org/doi/10.1145/3314111.3319913.

[49] Steil J, Hagestedt I, Huang MX, Bulling A. Privacy-aware eye tracking using differential privacy. In: Proceedings of the 11th ACM symposium on eye tracking research & applications. New York, NY, USA: ACM; 2019, p. 1–9. http://dx.doi.org/10.1145/3314111.3319915, URL https://dl.acm.org/doi/10.1145/3314111.3319915.

[50] Microsoft. MixedRealityToolkit-Unity. 2024, https://github.com/microsoft/MixedRealityToolkit-Unity/. [Accessed 1 March 2024].

[51] Microsoft. Eye tracking on HoloLens 2. 2023, https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking. [Accessed 1 March 2024].

[52] Microsoft. EyesPose Class (Windows.Perception.People) - Windows UWP. 2024, https://learn.microsoft.com/en-us/uwp/api/windows.perception.people.eyespose?view=winrt-22621. [Accessed 1 March 2024].

[53] Kapp S, Barz M, Mukhametov S, Sonntag D, Kuhn J. ARETT: Augmented reality eye tracking toolkit for head mounted displays. Sensors 2021;21(6):2234. http://dx.doi.org/10.3390/s21062234, URL https://www.mdpi.com/1424-8220/21/6/2234.

[54] Dunn MJ, Alexander RG, Amiebenomo OM, Arblaster G, Atan D, Erichsen JT, et al. Minimal reporting guideline for research involving eye tracking (2023 edition). Behav Res Methods 2023. http://dx.doi.org/10.3758/s13428-023-02187-1, URL https://link.springer.com/10.3758/s13428-023-02187-1.

[55] Ostermaier B, Römer K, Mattern F, Fahrmair M, Kellerer W. A real-time search engine for the web of things. In: Proceedings of the 2010 international conference on the internet of things. 2010, p. 1–8. http://dx.doi.org/10.1109/IOT.2010.5678450.

[56] Ciortea A, Mayer S, Bienz S, Gandon F, Corby O. Autonomous search in a social and ubiquitous web. Pers Ubiquitous Comput 2020. http://dx.doi.org/10.1007/s00779-020-01415-1.

[57] Kassner M, Patera W, Bulling A. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication. UbiComp '14 adjunct, New York, NY, USA: Association for Computing Machinery; 2014, p. 1151–60. http://dx.doi.org/10.1145/2638728.2641695, URL https://doi.org/10.1145/2638728.2641695.

[58] Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, Van de Weijer J. Eye tracking: A comprehensive guide to methods and measures. OUP Oxford; 2011.

[59] Microsoft. Extended eye tracking in unity. 2022, https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/extended-eye-tracking-unity. [Accessed 1 March 2024].

[60] Campbell CS, Maglio PP. A robust algorithm for reading detection. In: Proceedings of the 2001 workshop on perceptive user interfaces. Orlando Florida USA: ACM; 2001, p. 1–7. http://dx.doi.org/10.1145/971478.971503, URL https://dl.acm.org/doi/10.1145/971478.971503.

[61] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[62] Hodges J, García K, Ray S. Semantic development and integration of standards for adoption and interoperability. Computer 2017;50(11):26–36. http://dx.doi.org/10.1109/MC.2017.4041353.

[63] Holmqvist K, Nyström M, Mulvey F. Eye tracker data quality: What it is and how to measure it. In: Proceedings of the symposium on eye tracking research and applications. Santa Barbara California: ACM; 2012, p. 45–52. http://dx.doi.org/10.1145/2168556.2168563, URL https://dl.acm.org/doi/10.1145/2168556.2168563.

[64] Zemblys R, Niehorster DC, Komogortsev O, Holmqvist K. Using machine learning to detect events in eye-tracking data. Behav Res Methods 2018;50(1):160–81. http://dx.doi.org/10.3758/s13428-017-0860-3, URL https://doi.org/10.3758/s13428-017-0860-3.

[65] Startsev M, Agtzidis I, Dorr M. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. Behav Res Methods 2019;51(2):556–72. http://dx.doi.org/10.3758/s13428-018-1144-2, URL https://doi.org/10.3758/s13428-018-1144-2.

[66] Startsev M, Zemblys R. Evaluating eye movement event detection: A review of the state of the art. Behav Res Methods 2023;55(4):1653–714. http://dx.doi.org/10.3758/s13428-021-01763-7, URL https://doi.org/10.3758/s13428-021-01763-7.

[67] Christoff K, Irving ZC, Fox KC, Spreng RN, Andrews-Hanna JR. Mind-wandering as spontaneous thought: A dynamic framework. Nat Rev Neurosci 2016;17(11):718–31. http://dx.doi.org/10.1038/nrn.2016.113.

[68] Csikszentmihalyi M. Toward a psychology of optimal experience. In: Flow and the foundations of positive psychology. Springer; 2014, p. 209–26.

[69] Hostettler D, Bektaş K, Mayer S. Pupillometry for measuring user response to movement of an industrial robot. In: 2023 symposium on eye tracking research and applications. Tubingen Germany: ACM; 2023, p. 1–2. http://dx.doi.org/10.1145/3588015.3590123, URL https://dl.acm.org/doi/10.1145/3588015.3590123.

[70] Wang X, Kwon T, Rad M, Pan B, Chakraborty I, Andrist S, et al. HoloAssist: An egocentric human interaction dataset for interactive AI assistants in the real world. 2023, p. 20270–81, URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_HoloAssist_an_Egocentric_Human_Interaction_Dataset_for_Interactive_AI_Assistants_ICCV_2023_paper.html.

[71] Konrad R, Padmanaban N, Buckmaster JG, Boyle KC, Wetzstein G. GazeGPT: Augmenting Human Capabilities using Gaze-contingent Contextual AI for Smart Eyewear. 2024, http://dx.doi.org/10.48550/arXiv.2401.17217, URL http://arxiv.org/abs/2401.17217, arXiv:2401.17217 [cs].

[72] García K, Mayer S, Ricci A, Ciortea A. Proactive digital companions in pervasive hypermedia environments. In: 2020 IEEE 6th international conference on collaboration and internet computing. CIC, 2020, p. 54–9. http://dx.doi.org/10.1109/CIC50333.2020.00017.

[73] Spirig J, Garcia K, Mayer S. An expert digital companion for working environments. In: Proceedings of the 11th international conference on the internet of things. IoT '21, New York, NY, USA: ACM; 2021, p. 25–32. http://dx.doi.org/10.1145/3494322.3494326, URL https://doi.org/10.1145/3494322.3494326.

[74] Grau J, Mayer S, Strecker J, Garcia K, Bektaş K. Gaze-based opportunistic privacy-preserving human-agent collaboration. In: Extended abstracts of the 2024 CHI conference on human factors in computing systems. CHI EA '24, New York, NY, USA: Association for Computing Machinery; 2024, http://dx.doi.org/10.1145/3613905.3651066, URL https://doi.org/10.1145/3613905.3651066.

[75] Pandjaitan A, Strecker J, Bektaş K, Mayer S. AuctentionAR - Auctioning off visual attention in mixed reality. In: Extended abstracts of the 2024 CHI conference on human factors in computing systems. CHI EA '24, New York, NY, USA: Association for Computing Machinery; 2024, http://dx.doi.org/10.1145/3613905.3650941, URL https://doi.org/10.1145/3613905.3650941.