

## Article

# UCA-EHAR: A Dataset for Human Activity Recognition with Embedded AI on Smart Glasses

Pierre-Emmanuel Novac <sup>1,\*</sup>, Alain Pegatoquet <sup>1</sup>, Benoît Miramond <sup>1</sup> and Christophe Caquineau <sup>2</sup>

<sup>1</sup> EUR Digital Systems for Humans, Université Côte d'Azur, CNRS, LEAT, 06410 Biot, France; alain.pegatoquet@univ-cotedazur.fr (A.P.); benoit.miramond@univ-cotedazur.fr (B.M.)

<sup>2</sup> Ellcie Healthy, 06600 Antibes, France; christophe.caquineau@ellcie-healthy.com

\* Correspondence: pierre-emmanuel.novac@univ-cotedazur.fr

**Abstract:** Human activity recognition can help in elderly care by monitoring the physical activities of a subject and identifying a degradation in physical abilities. Vision-based approaches require setting up cameras in the environment, while most body-worn sensor approaches can be a burden on the elderly due to the need of wearing additional devices. Another solution consists in using smart glasses, a much less intrusive device that also leverages the fact that the elderly often already wear glasses. In this article, we propose UCA-EHAR, a novel dataset for human activity recognition using smart glasses. UCA-EHAR addresses the lack of usable data from smart glasses for human activity recognition purpose. The data are collected from a gyroscope, an accelerometer and a barometer embedded onto smart glasses with 20 subjects performing 8 different activities (STANDING, SITTING, WALKING, LYING, WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS, RUNNING, and DRINKING). Results of the classification task are provided using a residual neural network. Additionally, the neural network is quantized and deployed on the smart glasses using the open-source MicroAI framework in order to provide a live human activity recognition application based on our dataset. Power consumption is also analysed when performing live inference on the smart glasses' microcontroller.

**Keywords:** human activity recognition; embedded AI; artificial neural network; smart glasses; wearable sensing



**Citation:** Novac, P.-E.; Pegatoquet, A.; Miramond, B.; Caquineau, C. UCA-EHAR: A Dataset for Human Activity Recognition with Embedded AI on Smart Glasses. *Appl. Sci.* **2022**, *12*, 3849. <https://doi.org/10.3390/app12083849>

Academic Editor: Gabriele Civitarese

Received: 22 November 2021

Accepted: 25 March 2022

Published: 11 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the growth of the senior population, elderly care becomes an important topic in the society. One aspect of elderly care is fall prevention, which is still challenging to tackle depending on the subject's health condition. In this context, artificial intelligence can be leveraged to notify about an increased risk. To achieve this goal, a solution consists in monitoring the subject's behaviour to detect some changes that could indicate a degradation of their mobility.

Human activity recognition (HAR) can be used for that purpose. In this article, HAR is solved as a machine learning problem that predicts activities of daily living performed by a subject using sensors data that can be of different modalities. Two sensor categories are mainly used for human activity recognition: vision-based and body-worn sensors. Vision-based sensing relies on cameras placed in the environment to capture a video stream of a subject performing activities of daily living [1]. Body-worn sensors rely on inertial measurement units (IMU), including an accelerometer, a gyroscope and sometimes additional sensors (magnetometer, barometer, etc.) to measure the subject movements. Various devices such as smartphones [2], wearables [3] or application-specific devices [4] can be used to collect data, some being more invasive than others. Body-worn sensors generate fewer data than cameras and do not require a specific environment setup. It is therefore easier to embed on autonomous devices.

Our approach is based on an inertial measurement unit embedded in smart glasses. Smart glasses are less invasive than some other devices such as dedicated IMU devices or even smartphones, especially for elderly for whom wearing glasses is common. However, and to the best of our knowledge, there is no available and usable dataset for human activity recognition based on smart glasses. Moreover, data would vary from one device to another due to sensors having different orientations, ranges, accuracy and sampling rates.

In this article, we present a new dataset [5] called UCA-EHAR with data collected from Ellcie Healthy's smart glasses [6]. Our dataset provides raw data collected from an accelerometer, a gyroscope and a barometer for 8 classes of activity performed by 20 subjects.

Additionally, for privacy, connectivity and latency reasons, all the data processing related to human activity recognition is performed directly on the smart glasses. Therefore, the machine learning algorithm performing the classification task is executed on the smart glasses' microcontroller. In previous works, we presented our MicroAI framework for end-to-end training, quantization and deployment of deep neural networks on microcontrollers [7]. This framework is now available as open-source [8]. In this work, the MicroAI framework is used to deploy a deep neural network model performing human activity recognition on the smart glasses. Quantization with 8-bit and 16-bit fixed-point representations is used to optimize the memory footprint and the inference time, thus reducing the power consumption as well.

Section 2 gives an overview of some of the available datasets and approaches for human activity recognition. Section 3 presents the smart glasses used for collecting data and performing live inference. Section 4 details the dataset and the protocol used to collect the data. Section 5 describes the deep neural network architecture used to classify activities from our dataset as well as the training phase. Section 6 summarizes the key characteristics of our MicroAI framework, such as its quantization and deployment process. In Section 7, classification results using our dataset are given and power consumption on the smart glasses is analysed. Finally, Section 8 concludes this work and discusses future perspectives.

## 2. State of the Art

Datasets for human activity recognition using various modalities have been flourishing for the past decade [9]. In this article, we mainly focus on body-worn sensors since vision-based or other environmental sensor approaches are significantly different compared to the smart glasses approach.

The most iconic dataset for human activity recognition using an inertial measurement unit is likely the Human Activity Recognition dataset hosted by the University of California Irvine, commonly dubbed UCI-HAR [2]. This dataset is built from a 3-dimensional accelerometer and a 3-dimensional gyroscope sampled at 50 Hz, embedded into a smartphone attached to the subject's waist. The acceleration signal is filtered to create an additional signal without gravity. Therefore, there is a total of nine channels of sensor data. The data are windowed over 2.56 s with 50% overlap to create windows of 128 samples. The data are provided in two forms: vectors of 128 samples for each of the nine sensor channels, and vectors of 561 features computed from the  $128 \times 9$  values. A total of 30 subjects participated in the experiments, performing 6 activities: WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS, SITTING, STANDING, LYING. A total of 21 subjects are used for training while the 9 others are used for testing, representing 7352 and 2947 vectors, respectively. As it will be seen further, some aspects of our dataset are inspired by UCI-HAR such as some classes and the window duration.

The UCI-HAR dataset was extended in [10] to provide the transitions between static activities: STAND\_TO\_SIT, SIT\_TO\_STAND, SIT\_TO\_LIE, LIE\_TO\_SIT, STAND\_TO\_LIE, LIE\_TO\_STAND. This SBHAR dataset was used to evaluate the Transition-Aware Human Activity Recognition [11] system along with two other datasets: PAMAP2 and REALDISP.

Instead of using a single smartphone with an accelerometer and a gyroscope, the PAMAP2 dataset [4] rather uses dedicated IMU devices called Colibri Wireless from Trivisio. One device is placed on the wrist, another one on the chest and a last one on the ankle. Each device contains a 3-dimensional accelerometer, a 3-dimensional gyroscope and a 3-dimensional magnetometer, along with a temperature sensor, all sampled at 100 Hz. Additionally, one heart-rate monitoring device is sampled at 9 Hz. In this dataset, nine subjects performed 12 to 18 activities. This setup is much more intrusive than UCI-HAR as multiple dedicated devices are used at specific location, making this approach harder to use in real conditions for live human activity recognition.

The REALDISP [12] dataset has an even more complex setup, using 9 IMU devices from Xsens sampled at 50 Hz, each with a 3-dimensional accelerometer, a 3-dimensional gyroscope, a 3-dimensional magnetometer. The IMU devices also provide orientation estimates in quaternion format (4D) [13]. This dataset contains more classes performed by more subjects than PAMAP2, 33 classes and 17 subjects, respectively. Its purpose was to study the impact of sensor placement.

Other popular human activity recognition datasets include UniMiB SHAR [14] containing accelerometer samples captured from a smartphone, Real-Life HAR [15] also collected from a smartphone but focusing on real-life situations (for example inactive, active or driving) rather than a laboratory setting, and OPPORTUNITY [16] that uses many sensors of different modalities.

Apart from these datasets using data collected from smartphones or specific devices, there are few other datasets based on wearables available from the market. We can cite WISDM [3] using a combination of a smartphone and a smartwatch (LG G Watch) to collect data from 51 subjects performing 18 activities. Other datasets for human activity recognition, such as [17] relying on a Microsoft Band 2, have been created from consumer smartwatches. However, these datasets have not been released so far.

More specifically, smart glasses are still not a popular device to use for human activity recognition. Nonetheless, prior works have been done to build a dataset for smart devices including smart glasses in [18]. This dataset makes use of Jins MEME smart glasses as well as a smartphone and a smartwatch to collect data from different sensors. The smart glasses provide data from an embedded IMU. This dataset has however some noticeable drawbacks. First, only one subject participated in the experiment. Moreover, there is no well-defined set of activities or well-defined protocol, which makes it difficult to evaluate or to extend.

Some efforts have been made in [19] to develop a system for activity recognition using smart glasses (Google Glass Explorer Edition XE 22). The authors compare the classification performance of a Support Vector Machine (SVM) between data collected either from a smartphone or smart glasses for 4 activities (Biking, Jogging, Movie Watching, and Video Gaming). Their system can perform inference on the Android smartphone but not on the smart glasses themselves.

However, and as it has been said in the introduction, each dataset will have its own characteristics depending on which device has been used. The device itself and its position will greatly influence the angle of the acceleration (both gravity and linear acceleration) as well as the signal shape for some movements. Additionally, the sensors themselves can have varying sensitivity and sampling rate. Therefore, using an existing dataset for a different device or application will produce poor classification results. For this reason, we created our own dataset for the Ellicie Healthy's smart glasses.

### 3. Ellcie Healthy Smart Glasses

Ellcie Healthy (EH) smart connected glasses are a multiple-purpose wearable device designed for e-health and road safety applications such as driver drowsiness detection, fall detection for elderly people or human activity recognition to prevent a fall. The Ellcie Healthy smart connected glasses shown in Figure 1 contain infrared proximity sensors embedded inside the rims for oculography purposes.

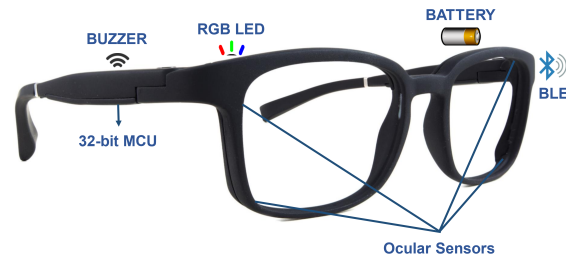


Figure 1. Ellcie Healthy Smart Glasses.

Other sensors such as a barometer, a thermometer, a triaxial accelerometer and a gyroscope are integrated within the frame temples. The accelerometer and the gyroscope are located on the same inertial measurement unit component. The barometric sensor and the temperature sensor are located in another component. The accelerometer provide each of the component of the three-dimensional acceleration vector along the orthogonal coordinate system shown in Figure 2. When the glasses are placed onto a table for example, most of the acceleration vector modulus (i.e., the gravity) is projected onto the Z axis approximately roughly giving  $9.81 \text{ m}\cdot\text{s}^{-2}$ . Depending on how the subject is wearing the glasses, the shape of the nose and other physiological factors, the gravity may not be perfectly projected onto the Z axis.

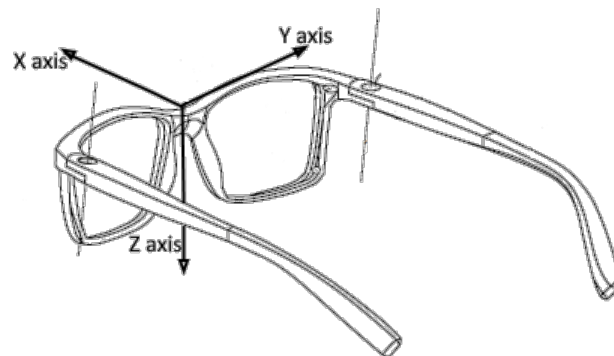


Figure 2. Accelerometer axes on Ellcie Healthy Smart Glasses.

The frame also includes a 32-bit microcontroller. The STM32L451RE microcontroller from STMicroelectronics has been chosen for its low power consumption while still being versatile. This microcontroller relies on a Cortex-M4F core running at 40 MHz in active mode and alongside 512 KiB of Flash memory and 160 KiB of SRAM. The microcontroller runs a real-time operating system to handle the various concurrent tasks. Additionally, a Bluetooth Low Energy (BLE) transceiver is integrated inside the frame to enable wireless communication with a gateway (typically a smartphone). Finally, a 350 mWh lithium polymer battery placed on the left temple of the frame provides the energy to the whole system using a flat flexible cable. This cable allows energy and data to flow back and forth through the bridge, the rims and the temples. Embedded algorithms, signal processing and data collection can therefore be directly executed on the smart glasses to provide health constants and/or security information to users. Alerts can be triggered when a risk event (e.g., driver drowsiness) is detected.

#### 4. UCA-EHAR Dataset

UCA-EHAR is our proposition of a dataset to address the lack of usable data for human activity recognition using smart glasses.

In order to build the UCA-EHAR dataset, we have enrolled 20 adult subjects, 8 women and 12 men (30.6 y.o average; 12 y.o standard deviation). Excluded were adults or children below 1.60 m of height, people with disabilities such as limping or backache.

UCA-EHAR contains 8 distinct classes to classify: STANDING, SITTING, WALKING, LYING, WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS, RUNNING, and DRINKING.

The choice of activities has been inspired by the UCI-HAR dataset as presented in Section 2. Additionally, these activities are simple to perform, common and relevant for elderly activity monitoring.

STANDING, SITTING, and LYING are static activities where the subject stays in the same position for a given duration. However, the subject does not need to stay completely still, but rather be natural as long as they keep either a STANDING, SITTING or LYING position.

WALKING, WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS and RUNNING are dynamic activities associated to mobility. The RUNNING activity is closer to walking fast than a sprint.

DRINKING is an activity that has been specifically added because we believe dehydration can be a risk for the elderly. The DRINKING activity is performed by drinking from a glass or a bottle, sip by sip.

The composition of the dataset can be seen in Appendix A.

##### 4.1. Data Collection Protocol

Each subject was given a table stating the guidelines of the recording. One voice recording per session was acquired. The entire signal recorded during a session can contain multiple status and transition classes as shown in Table 1.

Each data recording corresponds to one session as described in the table. Each session is described with 2 lines that must be read from left to right. The first line indicates the activity, while the second line gives the expected activity duration. Each session is a succession of activities. In order to provide a compact representation of sessions, an activity can be replaced by “repeat x times”. In that case, no duration is indicated, it is rather replaced by the activity number to start again from. Subjects did not necessarily repeat the activities as many times as recommended due to time constraints or physical conditions.

It is well known that homogeneous classes can be of premium importance to reach a good accuracy for some neural network family. As a transition is by nature shorter in time compared to a status class, the number of transition signal samples is very small compared to the status classes' samples. Even though the transitions are labelled in the dataset, they are not considered meaningful for classification in this article and are therefore filtered out for classification results.

The recording process is performed using two mobile phones. One phone, running the so-called “research application” from Ellcie Healthy, is connected to the smart glasses through a Bluetooth Low Energy connection. The research application records the accelerometer, gyroscope and barometer samples sent by the smart glasses. The other phone is used to record the voice of the subject. The subject or the test assistant must pronounce the keyword corresponding to the activity that the subject is currently performing.

Example of recordings of approximately 20 s for each session are shown in Appendix C.

**Table 1.** Instructions for each session of activity recording.

Session	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6	Activity 7	Activity 8	Activity 9	Activity 10
WALKING	STANDING 5 s	WALKING 240 s	STANDING 5 s							
RUNNING	STANDING 5 s	RUNNING 180 s	STANDING 5 s							
STANDING	STANDING 5 s	WALKING 6 s	STANDING 180 s	WALKING 6 s	STANDING 5 s					
SITTING	STANDING 5 s	STAND_TO_SIT (no rush)	SITTING 90 s	SIT_TO_STAND (no rush)	Repeat once from Activity 1	STANDING 5 s				
LYING	STANDING 5 s	STAND_TO_SIT (no rush)	SITTING 7 s	SIT_TO_LIE (no rush)	LYING 90 s	LIE_TO_SIT (no rush)	SITTING 7 s	SIT_TO_STAND (no rush)	Repeat once from Activity 1	STANDING 5 s
STAIRS	STANDING 5 s	WALKING (5 to 6 steps)	WALKING_ UPSTAIRS (15 to 25 stairs)	WALKING (5 to 6 steps)	WALKING_ DOWNSTAIRS (15 to 25 stairs)	Repeat 7 times from Activity 2	WALKING (5 to 6 steps)	STANDING 5 s		
DRINKING	SITTING 5 s	DRINKING 1 sip/10 mL	Repeat 29 times from Activity 1	SITTING 5 s						

#### 4.2. Data Format

The accelerometer, gyroscope and barometer, respectively, have 3 values for acceleration, 3 values for the angular velocity and 1 atmospheric pressure value.

The full sensitivity range is  $\pm 2g$  ( $g = 9.81 \text{ m}\cdot\text{s}^{-2}$ ) for the accelerometer and  $\pm 2000$  dps (degrees per second) for the gyroscope. The Ellcie Healthy glasses used in this experiment sample the 6 signals from the accelerometer and the gyroscope at a rate of 26 Hz, whereas the barometer is sampled at 6.66 Hz.

Before the labelling process, an interpolation routine has to be executed within the Matlab environment to provide the atmospheric pressure interpolated values for each accelerometer timestamp, so that a merged file containing one timestamp and 7 columns is produced. It is worth noticing that the barometer, the gyroscope and the accelerometer share the same sampling time origin. The values are provided in  $\text{m}\cdot\text{s}^{-2}$ ,  $\text{rad}\cdot\text{s}^{-1}$  and hPa.

The voice recording and additional supporting Matlab routines are used to determine the right label for each sample. Files are provided in CSV format with a semicolon as the column delimiter. The files contain one line every 40 ms approximately, with nine columns labelled "T" for the timestamp, "Ax", "Ay" and "Az" for the accelerometer, "Gx", "Gy" and "Gz" for the gyroscope, "P" for the atmospheric pressure and "CLASS" for the activity label. All numeric values are provided with 2 decimals. Finally, the name of the file is a combination of the identifier of the subject and the session name. The identifier of the subjects is numbered T1 to T21; however, T11 is skipped due to not having performed enough activities. Some recordings have been performed in two sessions, in such a case "\_1" or "\_2" is appended to the filename.

### 5. Machine Learning for Embedded Classification

In this section, a machine learning method to perform classification on the UCA-EHAR dataset is presented. Our aim is to provide a baseline for classification performance, so that these results can be used by other works for comparison. It is also the model used later on to perform inference for live human activity recognition on the smart glasses.

#### 5.1. Data Pre-Processing

As the objective is to perform live inference directly on the smart glasses, the amount of computation done before entering the artificial neural network must be minimized. In consequence, only a windowing pre-processing task is performed. The neural network indeed requires time series, in other words a context around each data point. The windowing process uses windows of 64 time samples, each time sample containing a value for the three accelerometer and gyroscope axes. Each window is overlapped by 25% with the previous one. Since data are sampled at 26 Hz, each window has a duration of approximately 2.46 s. This is close to the choice made by the authors of the UCI-HAR dataset [2]. The raw data from the dataset have one label per time sample. Time samples in a window may have different labels. During windowing, the labels are reduced to one per window by selecting the label with the highest number of occurrences in the window. Despite the barometer data being provided in the dataset, they are not used in the embedded experiments since the barometer is not sampled at the same rate as the accelerometer and gyroscope. To use the barometer data during live inference, resampling the data coming from the sensor would have to be performed on the smart glasses.

#### 5.2. Train/Test Split

The dataset is split in two parts: one for training and one for testing. There are 14 subjects in the training set and six subjects in the testing set, representing approximately 77% and 23% of the total number of samples, respectively. Subjects number 5, 15, 17, 18, 19, and 20 have been chosen for the testing set since they have completed all activities. Moreover, these subjects have the lowest standard deviation on the percentage of samples for each class in the testing set. Therefore, as seen at the bottom of Appendix A, activities are balanced as much as possible between the training and testing sets.

The total number of time samples in the training and the testing sets are 563,469 and 170,150, respectively. After windowing, the total number of vectors in the training and the testing sets are 35,213 and 10,631, respectively. The distribution of time samples before windowing by subjects and activities for both the training set and the testing set can be seen in Appendix A.

### 5.3. Data Augmentation

In order to mitigate overfitting and improve generalization, three different data augmentation techniques have been used during training: time shifting, time warping and 3D rotations. Time shifting performs a uniformly distributed random rotation over the time axis in order to shift the centre of the window. Time warping performs a dilation over the time axis in order to speed up or slow down the movement. The dilation scale factor is chosen randomly from a normal distribution with a mean  $\mu = 0$  and a standard deviation  $\sigma = 0.15$ . 3D rotation performs a three-dimensional rotation over the three accelerometer and gyroscope axes. The three rotation angles are randomly chosen from a normal distribution with a mean  $\mu = 0$  and a standard deviation  $\sigma = 0.15$ .

### 5.4. Artificial Neural Network Architecture

A deep neural network is used as the machine learning algorithm. More specifically, a residual neural network has been used as it performed well on the UCI-HAR dataset in previous works [7]. Moreover, this type of network is easy to scale down for embedded hardware by changing the number of filters per convolutional layer. In this work, a one-dimensional ResNetv1-6 [20] network is used to classify time series from our dataset. All convolutional layers have the same number of filters  $f$ . The ResNetv1-6 architecture is illustrated in Figure 3.

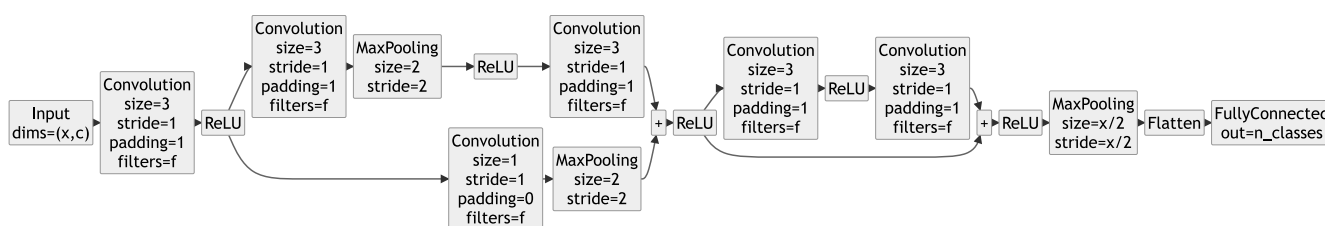


Figure 3. ResNetv1-6 model architecture.

The neural network is trained over 750 epochs using stochastic gradient descent (SGD) with momentum set to 0.9 and weight decay set to  $5 \times 10^{-4}$ . The batch size is set to 768. Initial learning rate is set to 0.025 and divided by 10 at epochs 200, 400, 600 and 675.

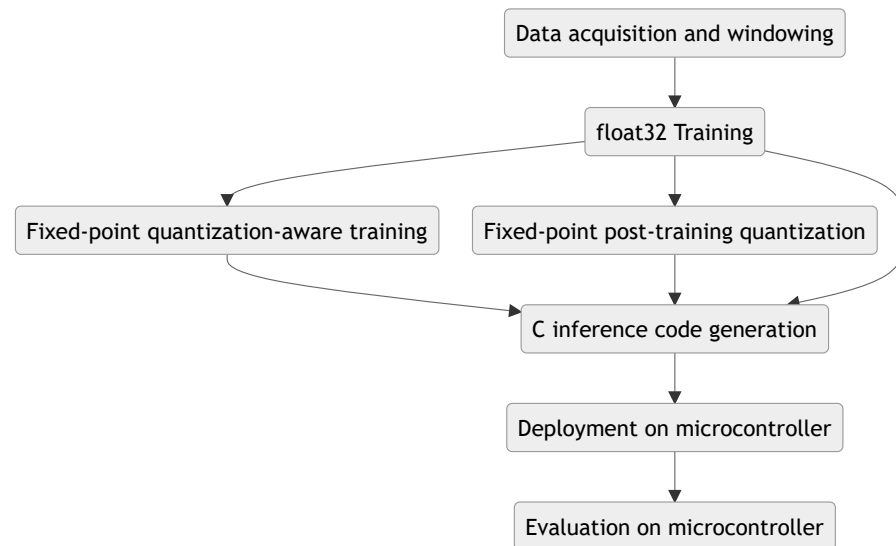
## 6. Quantization and Deployment of Deep Neural Networks with MicroAI

In order to perform human activity classification in real time on the microcontroller of the smart glasses, our MicroAI framework [7,8] is used. MicroAI is an open-source, end-to-end deep neural network training, quantization and deployment framework mainly targeting microcontrollers. MicroAI is designed as an alternative to other embedded inference engines such as TensorFlow Lite for Microcontrollers [21] and STM32Cube.AI [22]. TensorFlow Lite is complex and hard to extend, while STM32Cube.AI is proprietary. Our framework aims at being more easily extensible and tailored to specific use cases. MicroAI is divided in two parts: a neural network training tool that relies on Keras or PyTorch, and a tool to generate a lightweight and portable C inference library from a trained model. MicroAI enables the quantization of deep neural networks onto 8 or 16 bits in fixed-point representation. Quantization can be done using either Post-Training Quantization (PTQ) or Quantization-Aware Training (QAT).

The general flow for the end-to-end training and deployment process is illustrated in Figure 4. The entire process is automated and based on a configuration file. The process begins with a data preprocessing phase in order to apply transformations such as



windowing. Then, a training is performed on a workstation using Keras or PyTorch. After the initial training, the model can be quantized with quantization-aware training or post-training quantization. Finally, the model is deployed and evaluated on the microcontroller.



**Figure 4.** MicroAI general flow for neural network quantization and evaluation on embedded target [7].

### 6.1. Quantization of Deep Neural Networks

After the initial training phase, the trained model can be quantized to perform inference using a fixed-point data format instead of floating point. Quantization is done after freezing the weights of the model as a post-training quantization step. Optionally, before freezing the weights, the model can be fine-tuned while taking into account the quantization error as a quantization-aware training step. While the values are quantized, a floating-point data type is still used during quantization-aware training. The data type conversion from floating-point to integers using fixed-point representation happens during the generation of the C inference library, both for quantization-aware training and for post-training quantization.

The quantization scheme of MicroAI does not make use of advanced quantization techniques such as non-power-of-two scale factors or asymmetric ranges [23]. Instead, a less complex quantization scheme is used: uniform quantization, per-layer power-of-two scale factor and symmetric ranges. Additionally, biases are quantized the same way as weights. Activations are quantized using a separate scale factor.

As it will be shown in Section 7, post-training quantization with 16-bit integers has no impact on accuracy. Moreover, the same fixed-point coding, set to Q7.9 [24] in our case, is used for all layers.

On the other hand, quantizing over 8-bit integers does negatively affect the accuracy. To mitigate the quantization loss, the fixed-point coding can be different between layers and is chosen considering the range of the training set values. In practice, this conversion method starts by finding  $m$ , the number of bits required to represent the largest unsigned integer part. In the fixed-point representation, one bit is used for the sign,  $m$  bits are used for the integer part and the remaining bits are used for the fractional part. Each floating-point value is then multiplied by  $2^m$  and cast to an integer, truncating the fractional part. In the following experiments, quantization-aware training is not used since it did not bring a significant improvement over post-training quantization.

### 6.2. Deployment of Deep Neural Networks on Microcontrollers

With MicroAI, various deep learning models such as multi-layer perceptrons, convolutional neural networks and residual neural networks can be deployed onto microcontrollers.

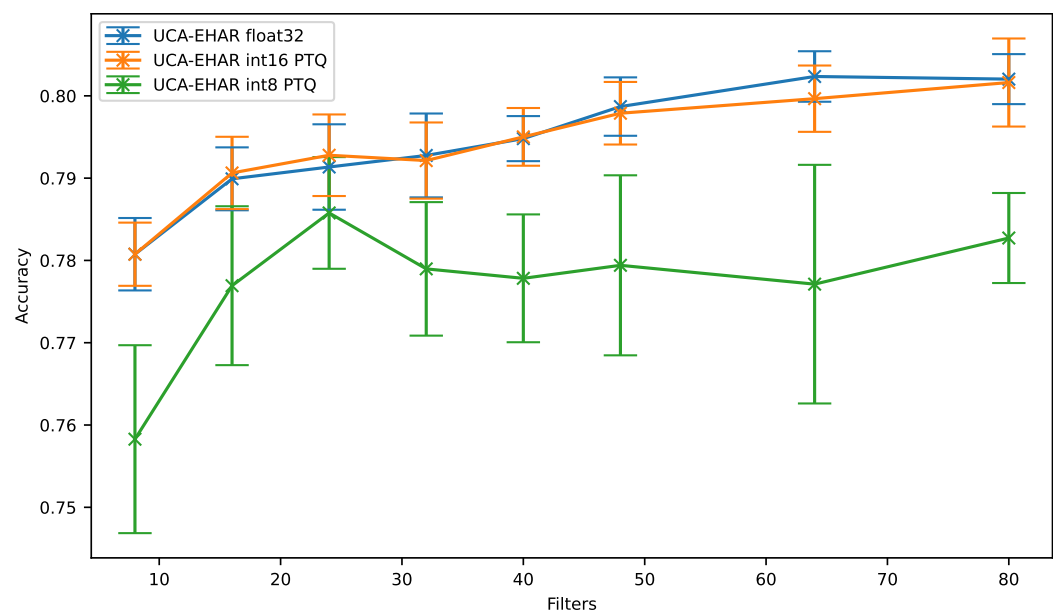
More generally, MicroAI can deal with the following type of layers: fully-connected, 1D convolution, 1D max pooling, 1D average pooling and element-wise addition. Development is currently ongoing to add support for the 2D variant of these layers. ReLU activation is fused with the previous layer. In order to deploy the model onto an embedded target for inference, a C inference library is generated. For each layer in the graph of the model, a C inference function is generated from a template file. Arrays containing the weights are also generated if applicable. Then the main inference function containing the call chain to the layers and the allocation of their output buffers is generated. Finally, the code is cross-compiled using the GCC compiler with `-Ofast` optimization level. MicroAI can optionally make use of the CMSIS-NN [25] library for faster 8- or 16-bit fixed-point inference, taking advantage of the so-called DSP instructions available in the ARMv7E-M instruction set architecture of the Cortex-M4 core. The inference time can then be measured directly onto the target by sending input vectors through the virtual serial port and waiting for the output of the deep neural network inference. Alternatively, the C inference library can be included into a third-party firmware, such as the firmware for Ellcie Healthy's smart glasses, in order to perform live inference with real data.

## 7. Experimental Results

### 7.1. Training and Prediction Results

The residual neural network described in Section 5.4 is trained for 8, 16, 24, 32, 40, 48, 64, and 80 filters per convolution. It is then quantized using the methods described in Section 6.1. Results are averaged over 15 runs for each number of filters.

The results for the original 32-bit floating-point model (UCA-EHAR float32), the 16-bit fixed-point quantized model with post-training quantization (UCA-EHAR 16-bit PTQ) and the 8-bit fixed-point quantized model with post-training quantization (UCA-EHAR 8-bit PTQ) are shown in Figure 5 and reported in Appendix B for each number of filters per convolution. As can be seen, 16-bit fixed-point quantization does not cause any accuracy loss while 8-bit fixed-point quantization causes up to 2.4% accuracy drop.



**Figure 5.** Accuracy vs. filters.

Concerning the memory used by the parameters, Figure 6 shows that the 16-bit fixed-point model is the most efficient, using half the memory of the 32-bit floating-point model but without any loss of accuracy. On the other hand, the 8-bit fixed-point model is less efficient than the 32-bit floating-point model since a noticeable loss of accuracy can be observed.

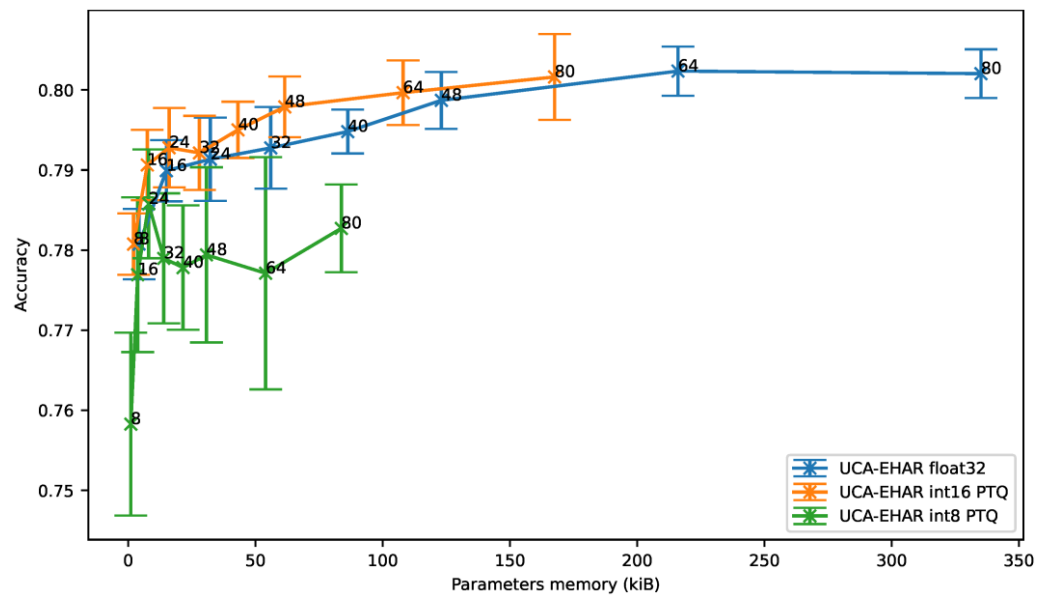


Figure 6. Accuracy vs. parameters memory.

The confusion matrix, shown in Figure 7 and extracted from one training for 80 filters per convolution, highlights the difficulty for an artificial neural network to differentiate the SITTING and STANDING activities from the collected data. The reason is that the orientation of the smart glasses remains the same for both classes, and the signals mostly stay constant for both of these motionless activities as seen in Figures A3 and A4 of Appendix C. It can be noted that the same confusion was already observed on existing datasets such as UCI-HAR.

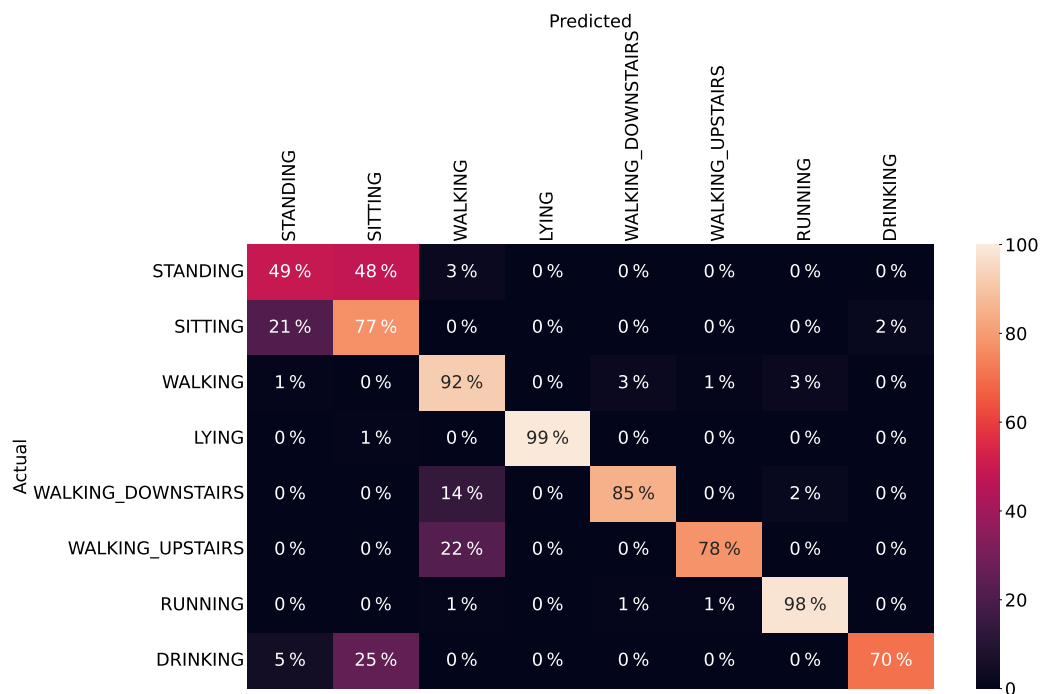
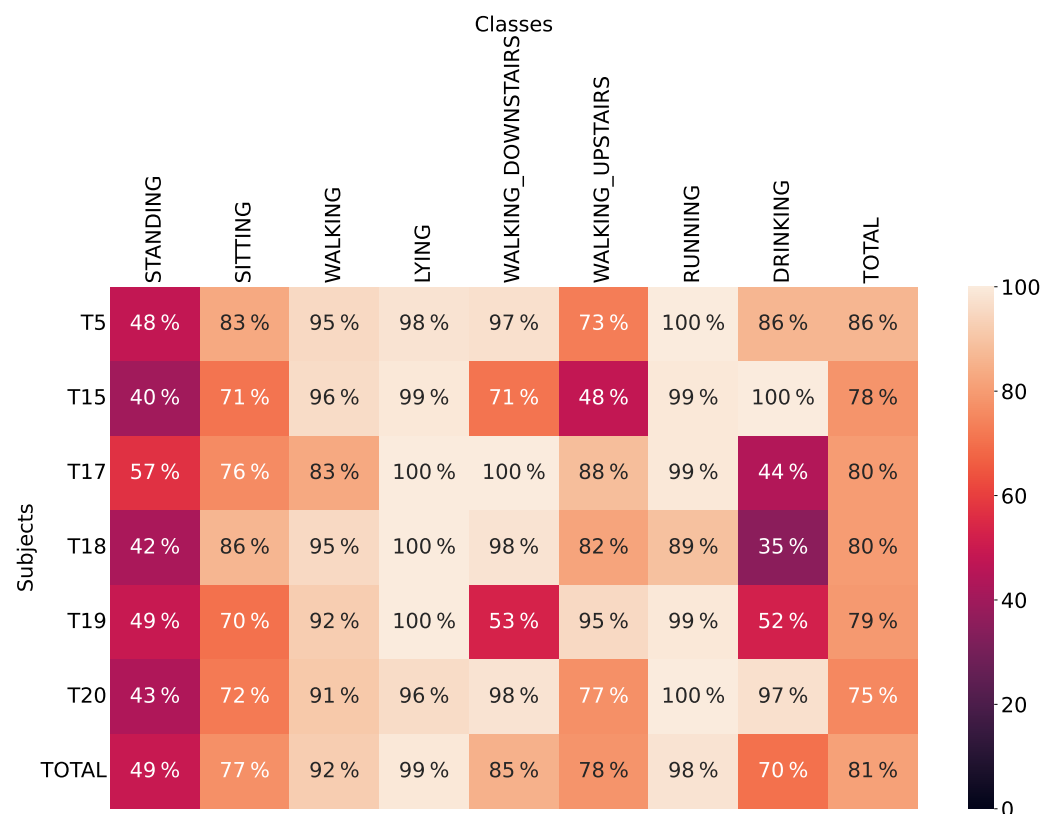


Figure 7. Confusion matrix for 80 filters per convolution.

An evaluation per subject has also been performed and is reported in Figure 8. The training set and the parameters are the same as the one used for the previous confusion matrix. However, inference is evaluated using each subject of the testing set one by one. It is important to note that since the classes are unbalanced, the accuracy in the “TOTAL” column does not represent the average of each class’s accuracy. Instead, it is the accuracy over all the test vectors of a given subject, and classes with more test vectors will have a greater influence on the resulting percentage of correct predictions. For example, for subject T20 the “TOTAL” of 75% is the most influenced by the “STANDING” activity, having much more samples than other activities and bringing the accuracy down. The same applies for the “TOTAL” line, since subjects do not all have the same number of test vectors per class. The bottom right cell, at the intersection of the “TOTAL” line and the “TOTAL” column, represents the accuracy over the entire testing set. Results show a discrepancy between subjects for some activities such as WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS and DRINKING, while other activities are more homogeneous. The STANDING activity, however, is hard to classify for all subjects. The reason is a large confusion with the SITTING activity, as previously shown in the confusion matrix.



**Figure 8.** Accuracy per class and per subject for 80 filters per convolution.

### 7.2. Deployment on Smart Glasses

A ResNetv1-6 is integrated into Ellcie Healthy’s smart glasses firmware version 6.1.2 using the C inference library generated by MicroAI. In this firmware version, only 77,604 B of Flash (for the inference code and the weights) and 40,572 B of RAM (for the intermediate computation and the layers’ output buffers of the deep neural network) can be used. Therefore, these memory limitations constrain the neural network that can be executed on the microcontroller. For the 32-bit floating-point inference, the largest ResNetv1-6 that can be deployed only contains 32 filters per convolution. Since the 16-bit fixed-point quantization provides the best memory efficiency, we also deployed a 16-bit ResNetv1-6 with 48 filters per convolution to get the best possible accuracy on the smart glasses. It is

worth noting that the same deep neural network without quantization (i.e., using 32-bit floating point) does not fit in Flash memory.

The memory footprint in Flash and the statically allocated RAM for each configuration is summarized in Table 2.

**Table 2.** Flash usage and static RAM allocation of the deep neural network (code and data).

Data Type	Optimizations	Flash (Available: 77,604 B)	RAM (Available: 40,572 B)	Accuracy
32 filters				
int8	CMSIS-NN	17,776 B	20,680 B	77.90%
int8	None	17,216 B	6,664 B	77.90%
int16	CMSIS-NN	31,440 B	26,192 B	79.21%
int16	None	32,720 B	13,328 B	79.21%
float32	N/A	60,336 B	23,200 B	79.28%
48 filters				
int16	CMSIS-NN	65,736 B	38,512 B	79.79%
float32	N/A	128,952 B *	33,440 B	79.87%

\* Memory overflow.

As expected, 8-bit and 16-bit quantizations allow reducing both the Flash and RAM usage. Therefore, models with more parameters can be deployed compared to the original 32-bit network. Using a 16-bit quantization, a network with 48 filters per convolution can indeed be deployed on the smart glasses. For this network, almost all the available memory is used: 94.43% of Flash and 98.43% of statically allocated RAM. On the other hand, a maximum of 32 filters per convolution can be used for the 32-bit network. For this network, the available memory is used as follows: 91.89% of Flash and 86.75% of statically allocated RAM.

The inference is performed after each time 64 samples are collected by the inertial measurement unit (IMU) whose sampling rate is 26 Hz. As the barometer sampling rate is 6.66 Hz, this sensor is not used in these experiments since resampling the signal would be required.

The power consumption of the smart glasses is measured using a Qoitech Otii Arc laboratory power supply, supplying 3.75 V in place of the LiPo battery. Energy values are computed by the Otii software from the current and voltage over a one minute window starting from the beginning of an inference. Obtained measurement over one inference period is shown in Figure 9 for 16-bit fixed-point inference with 48 filters per convolution and CMSIS-NN optimizations. The graph on the top shows the current consumption in mA while the graph at the bottom shows the voltage in V. The  $\Delta$  time indicates the duration of the selection, and the computed energy  $E$  over the selection is shown in the top right corner. It is worth noting that periodic spikes of current can be observed on the figure. Spikes at 20 Hz are related to the BLE transmission, while the spikes at 26 Hz are caused by the IMU sampling.

In the Figure 9, the inference task starts at the very beginning of the measurement. After the 173 ms of inference, 64 new samples are collected from the IMU. This figure clearly shows that the inference task requires much less time than collecting 64 samples. Therefore, in this configuration the inference time does not have a significant impact on the overall energy consumption. Over one inference period (i.e., approximately 2.6 s), 10,200 nWh represents the sum of the energy for the inference (1120 nWh) and the energy to collect the samples (9100 nWh).



**Figure 9.** Current and voltage captures over one inference period by Qoitech Oti software for int16 model with 48 filters per convolution and CMSIS-NN optimizations

Inference time and energy measurements were collected for various configurations and are shown in Table 3.

**Table 3.** Inference time and energy measurements on the smart glasses

Data Type	Optimization	Inference Time	Energy for One Inference	Energy over One Minute
32 filters				
int8	CMSIS-NN	53 ms	387 nWh	220 μWh
int8	None	115 ms	722 nWh	231 μWh
int16	CMSIS-NN	88 ms	605 nWh	232 μWh
int16	None	130 ms	853 nWh	234 μWh
float32	N/A	140 ms	919 nWh	235 μWh
48 filters				
int16	CMSIS-NN	173 ms	1120 nWh	237 μWh

Results show that quantization also helps to reduce inference time and therefore energy consumption for one inference. The original 32-bit floating-point network requires 140 ms on average for one inference, while its 16-bit quantized version only takes 88 ms for the same accuracy. Furthermore, the 8-bit quantized version only requires 53 ms, but as seen previously with a noticeable degradation of accuracy. However, the overall energy consumption over one minute does not significantly change with quantization. The overall energy is reduced by at most 7% between the 32-bit floating-point network and its 8-bit quantized version. As it has been observed in Figure 9, the inference time is indeed small compared to the time required to collect data. For that reason, the impact of inference over the overall energy consumption is small. Therefore, even if the largest network that fits in memory (48 filters per convolution with 16-bit quantization) is used, the autonomy of the smart glasses would not be impacted as long as the inference execution time remains small compared to the inference period. Hence, the energy consumption over one minute only

grows by 2% with a 16-bit quantized network with 48 filters per convolution rather than using 32 filters per convolution.

Ellcie Healthy's smart glasses embed a 350 mWh battery. Therefore, when the 16-bit quantized network with 48 filters per convolution is used (this network consumes 237  $\mu$ Wh per minute), the autonomy can reach 1476 min, i.e., 24.6 h. This estimated lifetime does not take into account additional applications that could run concurrently as well as battery ageing.

The larger the neural network, the larger the memory and the higher the energy consumption. However, in our case study, the memory footprint is a far more important parameter than energy consumption, primarily making the artificial intelligence in the smart glasses a memory bound problem.

### 7.3. Live Human Activity Recognition on Smart Glasses

The ResNet1-6 model with 48 filters per convolution, 16-bit fixed-point quantization and CMSIS-NN optimizations, has been trained using the UCA-EHAR dataset. This network has been then integrated onto the smart glasses firmware to perform live human activity recognition. Data are collected from the accelerometer and the gyroscope of the smart glasses when worn by a subject. The smart glasses' microcontroller performs the classification and sends the label of the recognized activity to a computer for visualization through a Bluetooth Low Energy communication. Additionally, the accelerometer and gyroscope data are also sent for visualization, even though the classification is not performed on the computer. A 30-second sample of such a live recognition has been extracted and can be seen in Figure 10. In this extract, the following sequence of activities has been performed by the subject: walking downstairs, walking upstairs, walking, stopping in a standing position and finally drinking a sip of water.

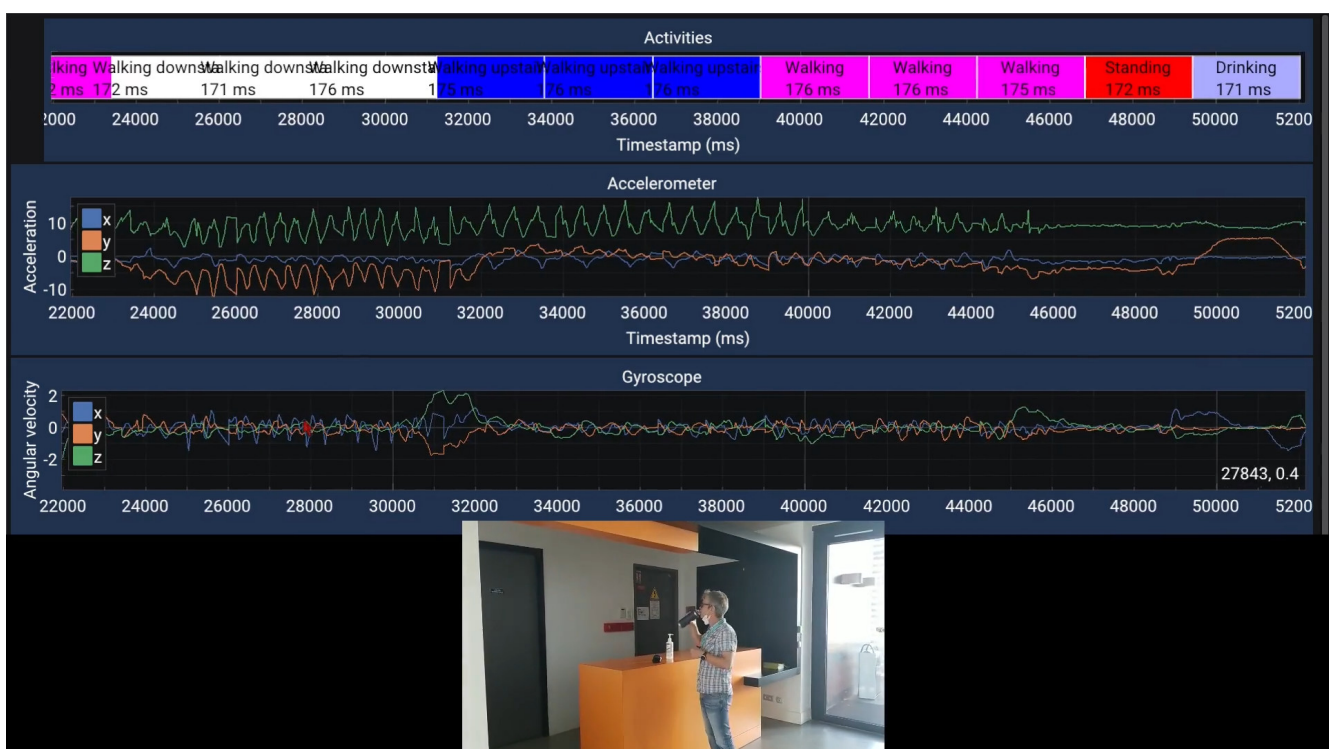


Figure 10. Live human activity recognition on smartglasses.

No quantitative evaluation of the live recognition performance has been done so far. However, it can be said that qualitatively the performance follows the results presented in the confusion matrix. Activities such as WALKING, WALKING\_DOWNSTAIRS, WALKING\_UPSTAIRS and DRINKING are generally recognized properly, while the STANDING and SITTING activities cannot be distinguished properly.

## 8. Conclusions

In this article, a novel dataset for human activity recognition called UCA-EHAR has been presented. This dataset gathers data collected from the accelerometer, the gyroscope and the barometer of smart glasses. UCA-EHAR is the first publicly available dataset dedicated to human activity recognition on activities of daily living using smart glasses. To provide a comparison baseline for a classification task, we evaluated the performance of a residual neural network on our dataset and we provided accuracy results as well as a confusion matrix. The accuracy for this dataset using a floating-point ResNetv1-6 with 80 filters per convolution is 80.2%. However, such a floating-point implementation does not respect embedded constraints of the smart glasses. Therefore, the neural network has been quantized using 8-bit and 16-bit fixed-point inference in order to optimize the memory footprint and the inference time, thus the energy consumption. Obtained results show that the 16-bit quantization provides the best accuracy vs. memory efficiency. To illustrate the energy that can be saved by quantization, we deployed a deep neural network onto the smart glasses using our MicroAI framework. We then measured the current and voltage during a human activity recognition task running on the smart glasses. Using the 16-bit quantized network with 48 filters per convolution we have shown that we can run human activity recognition for up to 24 h on the smart glasses. In the future, we will build a dataset including more classes such as transitions (SIT\_TO\_STAND, STAND\_TO\_SIT, SIT\_TO\_LIE, LIE\_TO\_SIT) or other activities (DRIVING). We would also like to explore unsupervised online learning using this dataset. To do so, collecting data for some subjects over a longer period of time will be required. Preliminary results were already presented in [26] using the UCI-HAR dataset. Unsupervised online learning will be implemented in our MicroAI framework to automatically train, quantize and deploy a network composed of convolutional layers and unsupervised layers onto the smart glasses.

**Author Contributions:** Investigation, P.-E.N.; methodology, P.-E.N.; software, P.-E.N.; data curation, C.C.; supervision, A.P. and B.M.; writing—original draft preparation, P.-E.N.; writing—review and editing, A.P., B.M. and C.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded by “Université Côte d’Azur”, “CNRS”, “Région Sud Provence-Alpes-Côte d’Azur” and “Ellcie Healthy”.

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki, and approved by the Ethics Committee of CER (Comité d’Ethique de la Recherche) (protocol code n° 2022-033 and date 8th of April of the Ethical Approval).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are openly available in Zenodo at <https://doi.org/10.5281/zenodo.5659336> (accessed on 24 November 2021).

**Conflicts of Interest:** The authors declare that Ellcie Healthy, one of the funders, has organized and managed the data collection of this study. However, the authors declare that there is no conflict of interest in the context of this study that has only been conducted by the researchers from LEAT lab.



## Appendix A

**Table A1.** Distribution of time samples across subjects and activities for training and testing sets.

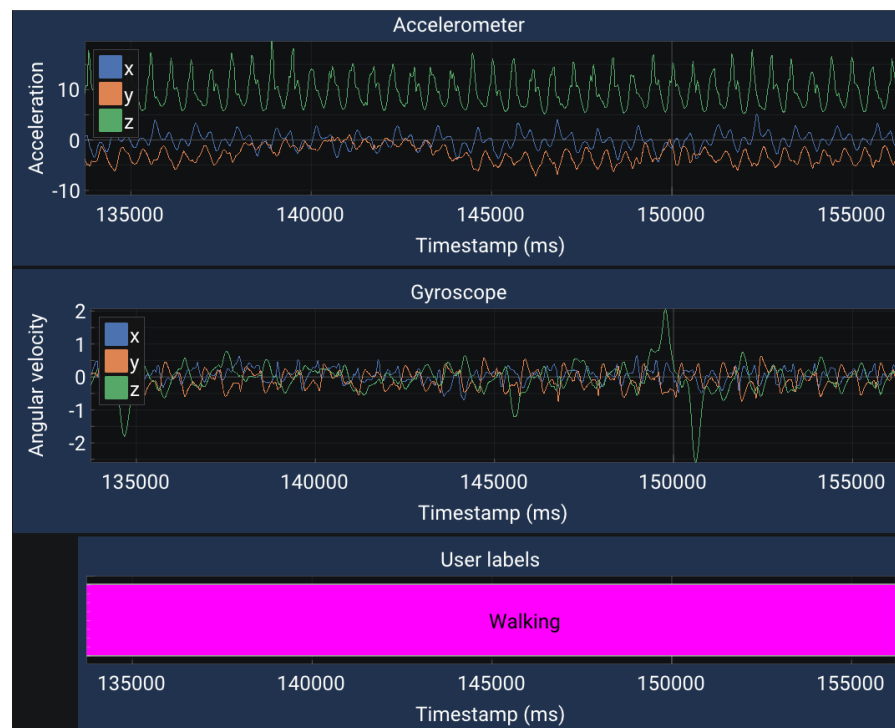
Subject	Activities								
	STANDING	SITTING	WALKING	LYING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS	RUNNING	DRINKING	TOTAL
Training set									
T1	8620	12,021	9955	5712	1588	1701	4310	4543	48,450
T2	6198	15,617	11,245	2626	3368	3298	4555	4069	50,976
T3	4973	17,904	17,029	3539	3887	3917	5300	5287	61,836
T4	7568	8822	10,871	5578	3132	3496	4002	3754	47,223
T6	6152	16,560	10,144	3199	2420	2305	5464	5093	51,337
T7	2162	16,436	9120	1984	2701	3333	4465	1383	41,584
T8	5151	4024	9378	4289	2145	2156	4064	0	31,207
T9	5113	6074	9578	3276	2596	3399	4015	0	34,051
T10	5899	4954	12,354	4226	1893	1943	4793	0	36,062
T12	4614	8509	10,559	1681	2368	2469	4641	1314	36,155
T13	7444	9957	13,449	12,224	2789	3373	6064	0	55,300
T14	4474	3611	7160	3025	1128	1384	4122	0	24,904
T16	5501	3489	8542	2250	1880	1940	3162	0	26,764
T21	1558	6524	2870	1937	1139	1148	1563	881	17,620
Total	75,427	134,502	142,254	55,546	33,034	35,862	60,520	26,324	563,469
Testing set									
T5	5587	8662	16410	3390	2276	2583	6016	1954	46,878
T15	1513	6295	3581	2388	1746	1490	1626	463	19,102
T17	4394	7749	7404	3227	1940	2611	3005	1683	32,013
T18	4684	7784	7110	2412	1299	1590	3210	1288	29,377
T19	1566	4780	3435	2401	1204	1564	1884	1011	17,845
T20	5495	7755	4150	2099	973	1177	1734	1552	24,935
Total	23,239	43,025	42,090	15,917	9438	11,015	17,475	7951	170,150
Set	Distribution between sets								
Training	76%	75%	77%	77%	77%	76%	77%	76%	76%
Testing	23%	24%	22%	22%	22%	23%	22%	23%	23%

## Appendix B. Prediction Results

**Table A2.** Accuracy and parameters memory for each configuration of residual neural networks.

Filters Per Convolution	Data Type	Parameters	Parameters Memory (B)	Accuracy
8	float32	1096	4384	78.08%
16	float32	3848	15,392	78.99%
24	float32	8264	33,056	79.14%
32	float32	14,344	57,376	79.28%
40	float32	22,088	88,352	79.48%
48	float32	31,496	125,984	79.87%
64	float32	55,304	221,216	80.24%
80	float32	85,768	343,072	80.20%
<hr/>				
8	int16	1096	2192	78.08%
16	int16	3848	7696	79.06%
24	int16	8264	16,528	79.28%
32	int16	14,344	28,688	79.21%
40	int16	22,088	44,176	79.50%
48	int16	31,496	62,992	79.79%
64	int16	55,304	110,608	79.97%
80	int16	85,768	171,536	80.16%
<hr/>				
8	int8	1096	1096	75.83%
16	int8	3848	3848	77.69%
24	int8	8264	8264	78.58%
32	int8	14,344	14,344	77.90%
40	int8	22,088	22,088	77.78%
48	int8	31,496	31,496	77.94%
64	int8	55,304	55,304	77.71%
80	int8	85,768	85,768	78.27%

## Appendix C. Example of Data from the Dataset



**Figure A1.** 20 s extracted from WALKING session of subject T1.

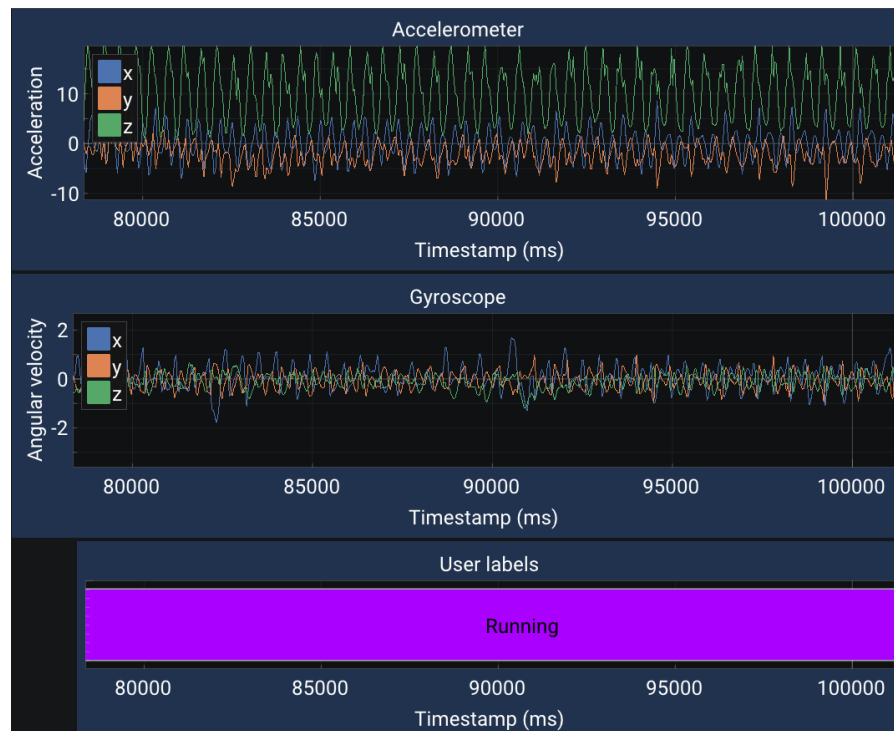


Figure A2. 20 s extracted from RUNNING session of subject T1.



Figure A3. 20 s extracted from STANDING session of subject T1.



Figure A4. 20 s extracted from SITTING session of subject T1.



Figure A5. 20 s extracted from LYING session of subject T1.



Figure A6. 20 s extracted from STAIRS session of subject T1.



Figure A7. 20 s extracted from DRINKING session of subject T1.

## References

1. Beddiar, D.R.; Hadid, A.; Nini, B.; Sabokrou, M. Vision-based human activity recognition: A survey. *Multimed. Tools Appl.* **2020**, *79*, 30509–30555. <https://doi.org/10.1007/s11042-020-09004-3>.
2. Davide, A.; Alessandro, G.; Luca, O.; Xavier, P.; Jorge, L.R.O. A Public Domain Dataset for Human Activity Recognition using Smartphones. In Proceedings of the ESANN, Bruges, Belgium, 24–26 April 2013.
3. Weiss, G.M.; Yoneda, K.; Hayajneh, T. Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. *IEEE Access* **2019**, *7*, 133190–133202. <https://doi.org/10.1109/ACCESS.2019.2940729>.

4. Reiss, A.; Stricker, D. Introducing a New Benchmarked Dataset for Activity Monitoring. In Proceedings of the 16th International Symposium on Wearable Computers, Newcastle, UK, 18–22 June 2012. <https://doi.org/10.1109/ISWC.2012.13>.
5. Novac, P.E.; Pegatoquet, A.; Miramond, B.; Caquineau, C. UCA-EHAR: A dataset for human activity recognition using smart glasses. *Zenodo* **2021**. <https://doi.org/10.5281/zenodo.5659336>.
6. Arcaya-Jordan, A.; Pegatoquet, A.; Castagnetti, A. Smart Connected Glasses for Drowsiness Detection: A System-Level Modeling Approach. In Proceedings of the 2019 IEEE Sensors Applications Symposium (SAS), Sophia Antipolis, France, 11–13 March 2019; pp. 1–6. <https://doi.org/10.1109/SAS.2019.8706022>.
7. Novac, P.E.; Boukli Hacene, G.; Pegatoquet, A.; Miramond, B.; Gripon, V. Quantization and Deployment of Deep Neural Networks on Microcontrollers. *Sensors* **2021**, *21*, 2984. <https://doi.org/10.3390/s21092984>.
8. Novac, P.E.; Pegatoquet, A.; Miramond, B. MicroAI, a software framework for end-to-end deep neural networks training, quantization and deployment onto embedded devices. *Zenodo* **2021**. <https://doi.org/10.5281/zenodo.5507397>.
9. Demrozi, F.; Pravadelli, G.; Bihorac, A.; Rashidi, P. Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey. *IEEE Access* **2020**, *8*, 210816–210836. <https://doi.org/10.1109/ACCESS.2020.3037715>.
10. Reyes-Ortiz, J.-L.; Oneto, L.; Ghio, A.; Samá, A.; Anguita, D.; Parra, X. Human Activity Recognition on Smartphones with Awareness of Basic Activities and Postural Transitions. In Proceedings of the 2014 International Conference on Artificial Neural Networks, Hamburg, Germany, 15–19 September 2014; pp. 177–184. [https://doi.org/10.1007/978-3-319-11179-7\\_23](https://doi.org/10.1007/978-3-319-11179-7_23).
11. Reyes-Ortiz, J.-L.; Oneto, L.; Samá, A.; Parra, X.; Anguita, D. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* **2016**, *171*, 754–767. <https://doi.org/10.1016/j.neucom.2015.07.085>.
12. Banos, O.; Toth, M.A.; Damas, M.; Pomares, H.; Rojas, I.; Amft, O. A benchmark dataset to evaluate sensor displacement in activity recognition. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 1026–1035. <https://doi.org/10.1145/2370216.2370437>.
13. Banos, O.; Toth, M.A. Realistic Sensor Displacement Benchmark Dataset, Dataset Manual. 2014. Available online: <https://archive.ics.uci.edu/ml/datasets/REALDISP+Activity+Recognition+Dataset> (accessed on 21 September 2021).
14. Micucci, D.; Mobilio, M.; Napoletano, P. UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. *Appl. Sci.* **2017**, *7*, 1101. <https://doi.org/10.3390/app7101101>.
15. Garcia-Gonzalez, D.; Rivero, D.; Fernandez-Blanco, E.; Luaces, M.R. A Public Domain Dataset for Real-Life Human Activity Recognition Using Smartphone Sensors. *Sensors* **2020**, *20*, 2200. <https://doi.org/10.3390/s20082200>.
16. Roggen, D.; Calatroni, A.; Rossi, M.; Holleczeck, T.; Förster, K.; Tröster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Ferscha, A.; et al. Collecting complex activity data sets in highly rich networked sensor environments. In Proceedings of the Seventh International Conference on Networked Sensing Systems, Kassel, Germany, 15–18 June 2010; pp. 233–240. <https://doi.org/10.1109/INSS.2010.5573462>.
17. Filippoupolitis, A.; Oliff, W.; Takand, B.; Loukas, G. Location-Enhanced Activity Recognition in Indoor Environments Using Off the Shelf Smart Watch Technology and BLE Beacons. *Sensors* **2017**, *17*, 1230. <https://doi.org/10.3390/s17061230>.
18. Faye, S.; Louveton, N.; Jafarnejad, S.; Kryuchenko, R.; Engel, T. An Open Dataset for Human Activity Analysis using Smart Devices. 2017. Available online: <https://www.kaggle.com/datasets/sasanj/human-activity-smart-devices> (accessed on 22 September 2021).
19. Ho, J.; Wang, C.M. User-Centric and Real-Time Activity Recognition Using Smart Glasses. In Proceedings of the 11th International Conference on Green, Pervasive, and Cloud Computing, Xi'an, China, 6–8 May 2016; pp. 196–210. [https://doi.org/10.1007/978-3-319-39077-2\\_13](https://doi.org/10.1007/978-3-319-39077-2_13).
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
21. David, R.; Duke, J.; Jain, A.; Reddi, V.; Jeffries, N.; Li, J.; Kreeger, N.; Nappier, I.; Natraj, M.; Regev, S.; et al. TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems. *arXiv* **2020**, arXiv:2010.08678.
22. STMicroelectronics. STM32Cube.AI. Available online: [https://www.st.com/content/st\\_com/en/stm32-ann.html](https://www.st.com/content/st_com/en/stm32-ann.html) (accessed on 19 March 2021).
23. Nagel, M.; Fournarakis, M.; Amjad, R.A.; Bondarenko, Y.; van Baalen, M.; Blankevoort, T. A White Paper on Neural Network Quantization. *arXiv* **2021**, arXiv:2106.08295v1.
24. ARM. *ARM Developer Suite AXD and armsd Debuggers Guide, 4.7.9 Q-Format*; ARM DUI 0066D Version 1.2; Arm Ltd.: Cambridge, UK, 2001.
25. Lai, L.; Suda, N. Enabling Deep Learning at the IoT Edge. In Proceedings of the International Conference on Computer-Aided Design (ICCAD'18), San Diego, CA, USA, 5–8 November 2018; Association for Computing Machinery: New York, NY, USA, 2018; <https://doi.org/10.1145/3240765.3243473>.
26. Novac, P.E.; Russo, A.; Miramond, B.; Pegatoquet, A.; Verdier, F.; Castagnetti, A. Toward unsupervised Human Activity Recognition on Microcontroller Units. In Proceedings of the 2020 23rd Euromicro Conference on Digital System Design (DSD), 2020, Kranj, Slovenia, 26–28 August 2020; pp. 542–550. <https://doi.org/10.1109/DSD51259.2020.00090>.