electronics

MDPI

Article

Activity Recognition in Smart Homes via Feature-Rich Visual Extraction of Locomotion Traces

Samaneh Zolfaghari *, Silvia M. Massa and Daniele Riboni

Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy; silviam.massa@unica.it (S.M.M.); riboni@unica.it (D.R.)
* Correspondence: samaneh.zolfaghari@unica.it

Abstract: The proliferation of sensors in smart homes makes it possible to monitor human activities, routines, and complex behaviors in an unprecedented way. Hence, human activity recognition has gained increasing attention over the last few years as a tool to improve healthcare and well-being in several applications. However, most existing activity recognition systems rely on cameras or wearable sensors, which may be obtrusive and may invade the user's privacy, especially at home. Moreover, extracting expressive features from a stream of data provided by heterogeneous smart-home sensors is still an open challenge. In this paper, we investigate a novel method to detect activities of daily living by exploiting unobtrusive smart-home sensors (i.e., passive infrared position sensors and sensors attached to everyday objects) and vision-based deep learning algorithms, without the use of cameras or wearable sensors. Our method relies on depicting the locomotion traces of the user and visual clues about their interaction with objects on a floor plan map of the home, and utilizes pre-trained deep convolutional neural networks to extract features for recognizing ongoing activity. One additional advantage of our method is its seamless extendibility with additional features based on the available sensor data. Extensive experiments with a real-world dataset and a comparison with state-of-the-art approaches demonstrate the effectiveness of our method.

Keywords: sensor-based activity recognition; smart environments; trajectory mining; visual feature extraction

Citation: Zolfaghari, S.; Massa, S.M.; Riboni, D. Activity Recognition in Smart Homes via Feature-Rich Visual Extraction of Locomotion Traces. *Electronics* **2023**, *12*, 1969. https://doi.org/10.3390/electronics12091969

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 31 March 2023
Revised: 20 April 2023
Accepted: 21 April 2023
Published: 24 April 2023

1. Introduction

The ability to monitor people's daily activities has various applications in several fields [1]. In healthcare, human activity recognition (HAR) is currently utilized to enhance rehabilitation, well-being, and to detect possible health issues early on [2,3]. In particular, the significant increase in the elderly population demands innovative ambient-assisted living tools that utilize HAR to prolong the independent lives of seniors [4]. Indeed, according to [5], in 1980, the estimated number of people over the age of 65 was 258 million, while in 2022 it reached 771 million. Moreover, the elderly population is expected to reach 994 million in 2030 and 1.6 billion in 2050. Older people are often physically and mentally frail compared to the younger population and they need support on a daily basis. HAR tools play a key role in enhancing the quality of life for this population, allowing for the monitoring of disease progression and assisting the elderly in performing daily activities [6–8]. In this regard, the monitoring of activities has significantly improved in recent years thanks to the miniaturization of sensors, their increasing integration into everyday objects such as smartphones and IoT devices, and the enhancements of AI algorithms, which allow for the extraction of deeply hidden information for accurate detection and interpretation [9].

The HAR research field aims to analyze data or signals obtained from the user or environment to associate them with the activities performed by the individual. According to a device-wise analysis conducted in [10], camera-based HAR techniques were the most popular during the period of 2011–2016, while sensor-based systems have been preferred since 2017. The choice of the device type depends on the specific application. For instance,

in surveillance applications involving multiple people, most HAR systems rely on cameras. Conversely, in applications that require recognition of the daily activities of a single person, sensor-based solutions are generally preferred due to their greater privacy-consciousness and lower computational costs.

Nowadays, different types of sensors are used for HAR, including motion, wearable, proximity, and environmental sensors [7]. These sensors are categorized according to whether they are embedded in the environment or worn by the person [4,11,12]. Ambient sensors are usually fixed to infrastructures or objects; therefore, considering triggered sensors and user interactions with sensors, the fact that the individual uses an object or is in a certain position can be traced back to a particular activity. Thus, all of these sensor records can be used to distinguish specific activities [4,7]. Wearable inertial sensors, on the other hand, are usually used to detect simpler activities, such as movement or posture, and are worn in specific parts of the body to capture certain movements or simple actions, such as 'walking'. However, wearable sensors (such as wearable glasses equipped with a camera) are also capable of assisting in recognizing complex activities [11]. One advantage of the latter types of sensors is that they are not fixed and bound to a specific position, but can be transported wherever the user goes [12]. Nevertheless, the use of wearable devices is perceived as obtrusive and uncomfortable by some users, especially elderly people [13].

Sensory data analysis has several important applications in different areas, including ambient-assisted living [4,14], physical training [15,16], managing emergency situations, and automatic detection of abnormal activities in surveillance environments [17,18]; most of these applications require HAR in the first step. HAR is typically addressed as a supervised learning problem [19,20], where a mathematical model is created based on the relationship between the raw or preprocessed input data (observations) and the output data (current activity). In recent years, several machine learning approaches have been explored with the aim of automatic classification of daily living activities (ADLs) based on mapping the extracted features out of sensor data [12,21]. One of the first works on sensor-based HAR was carried out in 1999 [22] where the authors detected movements and postures with the use of body-worn accelerometers. Recent works proposed adopting sliding windows and deep learning methods to extract features from wearable sensor data for activity recognition [23,24]. However, extracting expressive features from a stream of data provided by heterogeneous smart-home sensors is still an open research issue [25,26].

In this paper, we address the problem of extracting expressive features from an unobtrusive sensor-based activity recognition system. We propose a novel unobtrusive HAR system based on indoor locomotion and sensor data analyses. Our method relies on feature-rich visual extraction to encode the inhabitant's trajectories and manipulated objects into images, which are processed by a pre-trained deep learning model and classified by a supervised machine learning algorithm. As it is based only on environmental sensors, our system is unobtrusive. Since we do not rely on video or sound data, the system is privacy-conscious. Nonetheless, we point out that specific privacy-by-design methods should be enforced even when only environmental sensor data are released to third parties. Indeed, for instance, knowledge of PIR sensor data may reveal private information, such as the current activity or the presence/absence of home inhabitants.

We developed a prototype of our system and carried out extensive experiments with a real-world dataset gathered in a smart home's test bed with 175 seniors. We performed an experimental comparison with state-of-the-art techniques, which showed the superiority of our approach. In particular, our feature-rich visual extraction method provides higher recognition accuracy than a previously proposed image-based method. Moreover, using a pre-trained deep learning model essentially achieves the same accuracy as a custom convolutional neural network (CNN) architecture while incurring lower computational costs for training. Additionally, unlike the state-of-the-art technique proposed in [27], our visual extraction method can be seamlessly extended to incorporate additional features in case more sensor data are available, in order to improve recognition performance.

These are the main contributions of our work:

- In order to enhance the extraction of expressive features from sensor data streams for HAR, we propose a novel, feature-rich visual extraction method to visually represent the traces of sensor activations in a smart home, considering the walked trajectory, direction, speed, and interaction with objects;
- We used a pre-trained deep learning model to extract features from the image representation of sensor activations, and a supervised machine learning algorithm to recognize the current activity;
- We designed and implement a working prototype of our envisioned system;
- We experimentally evaluated our method using a large real-world dataset, outperforming a state-of-the-art technique.

The remainder of the paper is structured as follows. Section 2 discusses the related work. Section 3 illustrates the overall system architecture along with the techniques for trajectory segmentation, feature-rich visual encoding, feature extraction by pre-trained CNNs, and activity recognition utilizing classical machine learning algorithms. Section 4 presents our experimental evaluation. Section 5 discusses the results and the limitations of this work. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related Work

One of the main components of ambient-assisted living systems and smart homes is the ability to automatically recognize which human activities are taking place. In sensorized environments, HAR systems acquire data about the residents' interactions with the environment and objects and apply AI algorithms to detect the performed activities and behaviors. Activity data are then used to determine the appropriate actions of the smart system according to the current situation of the resident. For this reason, HAR methods are essential for remote care applications in a smart home [28].

Various types of approaches have been proposed for HAR in smart homes, which can be classified into two categories: data-driven and knowledge-based approaches [7,29]. In data-driven approaches, activity models are built using data mining and machine learning methods based on large sets of sensor data that capture user interactions with the environment and objects [30,31]. These approaches can handle temporary and uncertain data and create dynamic and personalized activity models [7]. However, they require a large amount of data for training and learning, and learned activity models cannot be easily applied to new users, which leads to scalability and reusability issues [29,30,32]. A challenging issue in data-driven HAR approaches involves the extraction of expressive features from spatiotemporal sensor data [23,33,34].

Knowledge-based approaches build activity models by exploiting prior knowledge, using knowledge engineering and management technologies [29,32]. Their models are easily explainable, and they do not require the collection of large amounts of data. Although most existing knowledge-based methods miss the ability to manage non-deterministic and temporal information, those models are often considered static and incomplete [7,32]. Furthermore, similar to data-driven approaches, they suffer from adaptability, scalability, and reusability issues, since they cannot adapt to different variants of activity models resulting from the behavior of different persons [29,30].

Ideally, a sensor-based HAR system should be automatically generalizable to different smart-home layouts without the need for prior knowledge, and without the need for acquiring a large amount of training data. A promising direction in this sense consists of tracking the movements of smart-home residents through unobtrusive positioning technologies [35–37], and analyzing location traces for HAR without violating the resident's privacy [38]. In this paper, we investigate the application of image-based trajectory classification to the domain of HAR. To the best of our knowledge, the only previous work that adopted a similar approach in the same domain was proposed by Gochoo et al. in [27]. In that work, the location of the resident at a given time was represented as a point in a two-dimensional grid, which was then represented as a binary (i.e., black and white) image. In the grid, one dimension represents time, while the other dimension represents one-dimensional space,

i.e., each row of the grid corresponds to a specific location in the home. To recognize the current activity, the corresponding image was classified using deep convolutional neural networks. However, the mapping of three-dimensional spatiotemporal trajectories into the two-dimensional grid resulted in a loss of information. The transformation from a two-dimensional to a one-dimensional space partially disrupts the spatial information, as metric operations and topological relationships are not preserved. Locations that are close to each other in the two-dimensional space of the smart home may be far apart in the one-dimensional space of the grid. In our work, we take a different approach that aims to retain spatial information by depicting the inhabitant's trajectory movements on the smart-home floor plan. Moreover, we enhance images with additional visual features that encode low-level user interaction and sensor events. To the best of our knowledge, this is the first work to investigate this method for HAR.

3. Methodology

Figure 1 illustrates our system's architecture. In this paper, we assume a sensorized smart-home infrastructure capable of continuously monitoring the inhabitant's position and their interactions with objects and appliances of interest. A smart home is equipped with various types of ambient sensors, including passive infrared (PIR) motion sensors and door sensors, to detect the inhabitant's location. In addition, specific sensors are installed to track the inhabitant's interactions with objects. The sensor data are continuously collected and used to encode the locomotion traces and interactions into trajectory images, which are then processed by machine learning algorithms to detect the activities taking place. For the sake of this work, we consider the case of a person living alone in the home.

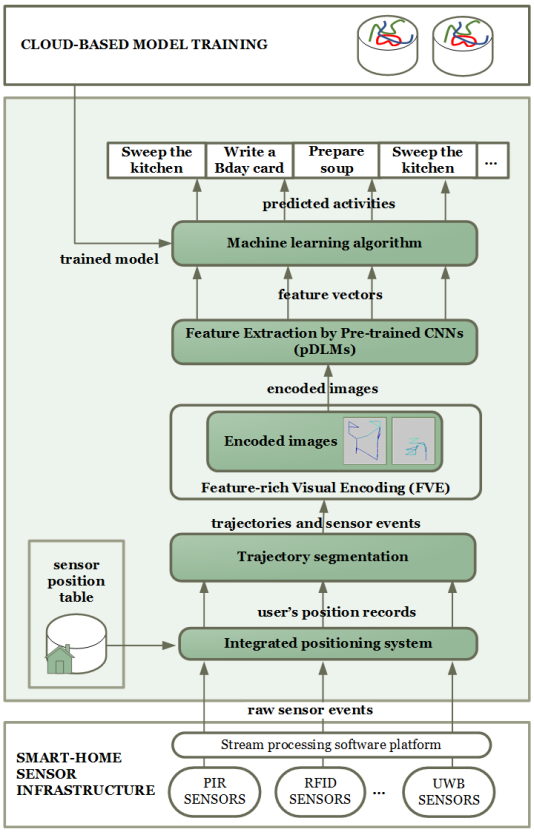


Figure 1. System architecture.

3.1. Smart-Home Sensor Infrastructure

The SMART-HOME SENSOR INFRASTRUCTURE is in charge of continuously collecting sensor data through PIR, door sensors, and manipulated objects.

Whenever a sensor is triggered, the platform sends the following **raw sensor event** to the INTEGRATED POSITIONING SYSTEM module: $e = \langle t, s\_id, v \rangle$, where t is the firing timestamp, s_id is the sensor's unique identifier, and v is the generated value. An example of a raw sensor event is represented in Table 1.

Table 1. Raw data from discrete sensors.

Timestamp	Sensor ID	Value
09:10:00.094833	M001	ON
09:10:01.014748	M023	ON
09:10:01.045917	M021	OFF
09:10:01.093183	M022	OFF
09:10:02.087933	M023	OFF
09:10:03.072194	M023	ON
09:10:05.014012	D012	OPEN
09:10:05.043057	M001	OFF
09:10:06.038858	M023	OFF
09:10:19.094168	D012	CLOSE

3.2. Integrated Positioning System

This module is in charge of relating triggered sensors to their relative positions in the layout of the smart home, which enables us to derive spatiotemporal information from raw sensor events. The INTEGRATED POSITIONING SYSTEM relies on the **sensor position table**. For each fixed sensor s_id, the table includes a tuple $\langle s\_id, (x, y) \rangle$ where x and y are the relative coordinates of the sensors in the environment. Therefore, each time the subject interacts with an object, the integrated positioning system receives a raw sensor event and continuously joins the corresponding record with the sensor position table to obtain its relative coordinates. The user's position record represents the position of the inhabitant and the manipulated object at a specific timestamp.

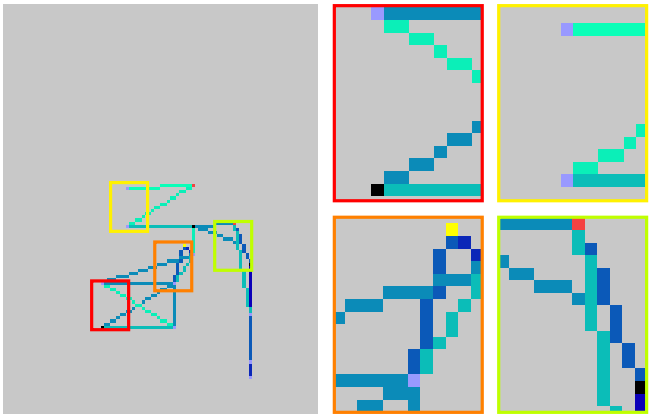
3.3. Trajectory Segmentation

This module is in charge of partitioning the temporal stream of data sent from the INTEGRATED POSITIONING SYSTEM into **trajectories**. The module partitions the user's position records using a sliding window approach. For this work, we used a window length of three minutes with a 20% overlap. We chose this duration because complex ADLs typically last longer. The recognition task is to identify the activity taking place in each 3-minute time slice. If multiple activities occur in the same time slice, we assign the time slice to the activity that was executed for the majority of the time.

3.4. Feature-Rich Visual Encoding (FVE) Method

In order to improve activity recognition, we visually encode (in images) the walked trajectory as well as other features and events of interest. The main features that are encoded are trajectory movements, speed of movement, stop points, manipulated objects, and sharp angles.

As shown in Figure 2, the patient's trajectory is traced through a blue line that takes on different shades of color, depending on the speed at which the patient is moving. A brighter color means that the patient's speed is increasing, while the contrary means that the patient's speed is slowing down. By considering the size of our smart home's test bed, each pixel in the image corresponds to approximately 0.1 m^2. Therefore, the size of each image is 100 by 130 pixels and the RGB color model is used. In this color mode, each pixel is characterized by three values for red, green, and blue ranging from 0 to 255 each.



(**a**) (**b**)

Figure 2. (**a**) An example of the encoded image with door interactions and abrupt changes in direction. Colored boxes represent areas of interest, which are zoomed in on the right. (**b**) Zoomed representations of the areas surrounded by colored boxes. The shade level of the blue line depends on the speed of movement. The *black* point represents a sharp angle. The *vivid violet* point shows a backward directional change. The *yellow* point indicates the interaction with a door. The *red* point indicates that the individual remained stationary.

The weight of each line is determined based on the frequency of the corresponding path being walked. Specifically, the weight of a line is set to 1 if the path has been walked only once, and it increases by 1 for each additional walk of the same path. Therefore, lines corresponding to frequently walked paths are bolder than those of paths walked less frequently. Moreover, when a given path is walked multiple times during the same trajectory, we consider the most recent speed of that path for drawing the corresponding image.

Within the image, we add colored points to depict events of interest. A red point in a trajectory indicates the position in which the patient remains stationary for more than 2 s. Black points represent sharp-angled points, i.e., abrupt changes in direction in the trajectory.

Other colored points are added to indicate directional changes. We use brown, vivid violet, and white points to indicate left-hand, backward, and right-hand directional changes, respectively. Moreover, interactions with doors are indicated by yellow points. In addition, other colored points indicate the objects used. These are very useful for the recognition of particular activities. Table 2 reports the objects considered in our experimental setup (Section 4) with their respective colors.

Table 2. Considered object sensors in the current paper.

Sensor ID	Object	Color
i001	Oatmeal	Green, RGB(36, 173, 9)
i002	Raisins	Orange, RGB(237, 123, 17)
i006	Medicine container	Pinkish purple hue, RGB(242, 0, 255)
i010	Medicine box	Pink, RGB(237, 147, 186)

3.5. Feature Extraction by Pre-Trained CNNs (pDLMs)

Convolutional neural networks (CNNs) can be used in two different ways. The first way is to design a classification model; the second way is to extract complex features [39]. In this work, we extracted features from our trajectory images by pre-trained CNN models (pDLMs) trained on the ImageNet dataset [40]. We extracted features from six different architectures of pre-trained networks: *VggNet*, *ResNet*, *MobileNet*, *DenseNet*, and two variants, which are described below.

- *Visual geometry group neural network (VggNet)* [41]: The original purpose of Vgg's research was to understand how the depth of CNN affected the accuracy of large-scale image classification and recognition. VggNet19 is a variant of the Vgg architecture, with 19 deeply connected layers, which consistently achieve better performances and enable better feature extraction compared to the simpler VggNet16. The model consists of highly connected convolutional and fully connected (FC) layers [42].
- *Residual neural network (ResNet)* [43]: Was inspired by VggNet and introduced a new architecture with a skip connection and batch normalization. This allows for deeper networks with fewer filters and lower complexity than VggNet [44]. ResNet50 is a variation of the ResNet architecture and has been trained on at least one million images from the ImageNet database. ResNet50V2 [45] is a modified version of ResNet50 that performs better than ResNet50 and ResNet101 on the ImageNet dataset, thanks to changes in the propagation formulation of the connections between blocks [42]. Both ResNet50 and ResNet50V2 produce the same size of feature map on their final layer.
- *MobileNet* [46]: It was introduced for mobile and embedded vision applications and is widely used in many real-world applications, which include object detection, fine-grained classifications, face attributes, and localization. MobileNet core layers are built on depthwise separable filters. Its separable convolution has two layers: depthwise convolution and point convolution. MobileNetV2 [47] is very similar to the original MobileNet, except that it uses inverted residual blocks with bottleneck features. It has a drastically lower parameter count than the original MobileNet.
- *Dense convolutional network (DenseNet)* [48]: It is a type of CNN that utilizes dense connections between layers, through dense blocks, where all layers are directly connected with matching feature-map sizes. In DenseNet121, all layers within the same dense block and transition layers spread their weights over multiple inputs. This topology allows deeper layers to use features extracted early on. DenseNet has been shown to have better feature use efficiency and outperforms ResNet with fewer parameters [49].

In Table 3, we present a comparison between the different architectures considered in our work. Since these networks expect input images of 224 by 224 pixels, we resized our trajectory images accordingly.

Table 3. Comparison between the different architectures used: model size and depth.

Year	Pre-Trained Architecture	Layers	Model Description
2014	VggNet19 [41]	19	16 Convolutional layers+ 3 FC layers
2016	ResNet50 [43], ResNet50V2 [45]	50	48 Convolutional layers+ 1 MaxPool layers + 1 AveragePool layer
2017	MobileNet [46]	28	13 Depth-Wise Convolutional layers + 14 Point-Wise Convolutional layers + 1 FC layer
2018	MobileNetV2 [47]	54	53 Convolutional layers + 1 AvgPool layer
2017	DenseNet121 [48]	121	117 Convolutional layers + 3 AvgPool layers + 1 FC layer

3.6. Locomotion-Based Activity Recognition

This module is in charge of recognizing the activities based on the feature vectors extracted by the pDLMs algorithm from the encoded images. A supervised machine learning algorithm classifies each feature vector based on a model trained on a sufficiently large training set of activities and sensor data. As shown in Section 4, in our experiments, we evaluated different machine learning classifiers.

4. Experimental Evaluation

In this section, we report our experimental evaluation, which was carried out with real-world sensor data acquired in an instrumented smart home from 175 seniors carrying

out 16 different activities of daily living. Therefore, in the following subsections, we explain the used dataset, the experimental setup, and the achieved results. We also experimentally compare our technique with an existing state-of-the-art method.

4.1. Dataset and Experimental Setup

The experiments were carried out using the real-world *Kyoto dataset*, http://casas.wsu.edu/datasets/assessmentdata.zip (accessed on 1 April 2023), which was annotated and publicly released by researchers at the Center of Advanced Studies in Adaptive Systems (CASAS) [50] of Washington State University (WSU). The dataset collection involved 400 participants with age ranges from 18 to over 75. After obtaining informed consent, participants underwent multidimensional clinical assessments by clinicians, in order to assess their cognitive health statuses. As a consequence of the clinical examination, each participant was classified into 1 of 10 possible categories of diagnosis. Since, for the sake of this work, we focused on the recognition of normal activities, only subjects classified as 'younger adult', 'middle age' (45 to 59 years old), or 'young-old' person (60 to 74 years old) were included (only including cognitively healthy subjects). Therefore, out of 400 participants, we carried out our experiments with data from 175 cognitively healthy individuals.

During data collection, each participant performed 16 different activities during a few hours of data collection. The smart home's testbed was equipped with several sensors, including motion sensors, door sensors, and sensors embedded in kitchen objects. Door sensors are reed switch sensors attached to cabinets, cutlery drawers, and room doors, whose values can be 'open' or 'closed'. They are used to detect entrances to or exits from certain rooms, or interactions with specific furniture. Sensors embedded into objects are based on RFID technology to detect the presence or absence of a specific item in a specific place.

Figure 3 shows the CASAS smart-home layout. In the figure, sensors marked with 'M' represent motion sensors, sensors marked with 'D' represent door sensors, and sensors marked with 'I' represent sensors embedded into objects. The testbed is described in detail in [50].

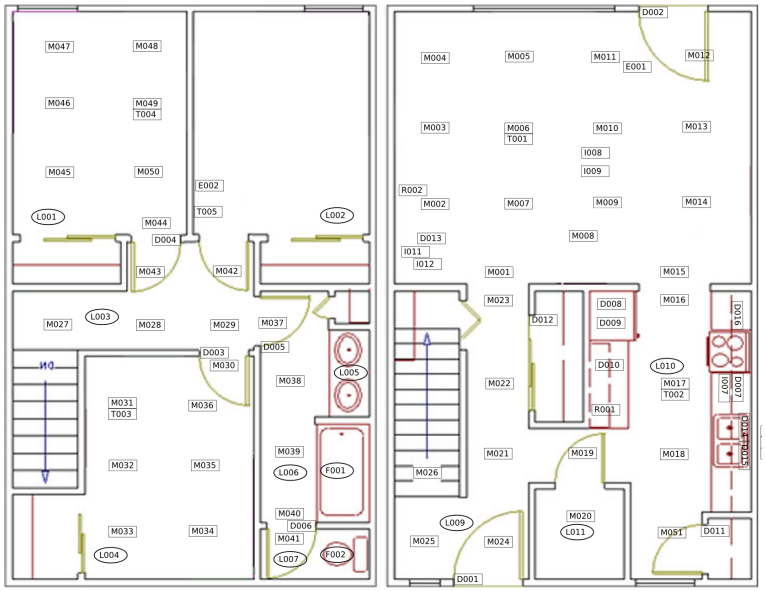


Figure 3. CASAS smart-home layout.

The positioning infrastructure used in our experimental test bed comprises 52 passive infrared (PIR) motion sensors mounted on the ceiling to track the user's position in the home with a 1-meter resolution (approximately). Therefore, in our work, a trajectory is a temporally contiguous sequence of approximate positions. To compute the speed of movement, we use linear disclosure with relative coordinates. For each pair of consecutive

positions p_1 and p_2, we calculate the speed of the trajectory segment $\langle p_1, p_2 \rangle$ based on the distance between p_1 and p_2 and the time spent moving from p_1 to p_2. Since the spatiotemporal information provided by the PIR system is inherently approximated, the computed speed is also an approximation. Hence, the system could be easily improved using more accurate localization systems.

We developed all of the algorithms in Python using TensorFlow and the Python Keras neural network library https://keras.io/ (accessed on 1 April 2023). We ran the experiments on a departmental Linux server with four NVIDIA Tesla p6 graphic boards, a single NVIDIA Pascal GP104 graphics processing unit (GPU), and 16 GB GDDR5 main memory. As anticipated before, there are 16 activities to be recognized in total. Closely inspecting the experimental testbed, we noticed that some of the considered activities are hard to recognize based on the sensors actually available in the smart-home testbed. For instance, the activity 'Wash hands with soap at the kitchen sink' can hardly be distinguished from any other kitchen activity requiring the usage of water, since no sensor is available for the hand soap dispenser. As a result, we identified a subset of 8 activities for which a substantial number of sensors are available for their recognition, and we refer to these as **sensor-rich activities**. The full list of activities is reported in Table 4.

Table 4. Activity list. Activities in **bold** are *sensor-rich* activities, i.e., activities for which a substantial number of sensors is available in the considered smart-home testbed for supporting their recognition.

Activity ID	Description
1	**Sweep the kitchen and dust the living room.**
2	Obtain a set of medicines and a weekly medicine dispenser, and fill as per directions.
3	**Write a birthday card, enclose a check, and address an envelope.**
4	**Find the appropriate DVD and watch the corresponding news clip.**
5	Obtain a watering can and water all plants in the living space.
6	Answer the phone and respond to questions pertaining to the video from task 4.
7	**Prepare a cup of soup using the microwave.**
8	**Pick a complete outfit for an interview from a selection of clothing.**
9	Check the wattage of a desk lamp and replace the bulb.
10	Wash hands with soap at the kitchen sink.
11	**Wash and dry all kitchen countertop surfaces.**
12	Place a phone call to a recording and write down the recipe heard.
13	**Sort and fold a basketful of clothing containing men's, women's, and children's articles.**
14	Prepare a bowl of oatmeal on the stovetop from the directions given in task 12.
15	**Sort and file a small collection of billing statements.**
16	Set up hands for a card game, answer the phone, and describe the rules of the game.

We adopted a k-fold cross-validation approach. The dataset was divided into $k = 10$ folds of equal size. At each iteration, the k_i fold is used for testing the recognition of the activities, while the remaining folds are used for training the model. The folds were created without shuffling, in order to avoid putting data from the same user into different folds. We performed our experiments considering several classifiers, namely:

- Naive Bayes (NB);
- k-nearest neighbor (kNN) which $k = 3$;
- Decision tree (DT);
- Support vector machine (SVM);

- Random forest (RF);
- Neural network, composed of 100 neurons in a hidden layer, ReLU activation function, Adam optimizer, and trained for 1000 epochs, with the batch size equal to 200;
- AdaBoost; and
- Quadratic discriminant analysis (QDA).

4.2. Comparison with Baseline Methods

In order to experimentally compare our method with different state-of-the-art approaches, we implemented both the binary image-based feature extraction method proposed by Gochoo et al. in [27] and their proposed deep convolutional neural network (DCNN) architecture.

4.2.1. State-of-the-Art Binary-Image Method (BIM)

As a comparison, we implemented the binary-image method (BIM) for activity recognition proposed by Gochoo et al. in [27]. In that work, the stream of sensor events collected during a given time period was represented by a binary image, as illustrated in Figure 4. Overall, the x-axis of the image represents time, while values in the y-axis represent a given sensor in the smart home. The pixels corresponding to the $\langle x, y \rangle$ coordinates were set to white if sensor y was activated at time x, and black otherwise. The motion and door sensors' data collected in the environment were first divided according to the activity performed by the user. Subsequently, further segmentation was performed using sliding time windows with a length of 3 min and 20% overlap. The activation of each sensor within a time frame was encoded by white points.

The activity image can be represented as a matrix in which the columns are the sensor events and the rows are the different sensors positioned in the environment. While it is automatic to associate the events to the columns following their temporal order, the order in which the sensors are associated with the rows of the matrix is not obvious and it is critical to the performance of the activity recognition system, as it affects the intra-sensory pattern information extracted. Among the various strategies tested, the ones that gave the best results were to have a distance of ten rows between two adjacent door sensors, a distance of one row between two adjacent motion sensors, and a distance of more than six rows between motion sensors in different rooms [27].

Figure 4. Example of images generated through the binary image feature extraction method [27].

4.2.2. State-of-the-Art DNN (DCNN)

In order to compare our machine learning architecture with a state-of-the-art DNN solution, we consider the $2D$ DCNN proposed by Gochoo et al. in [27] and use it for classifying our encoded images. The DCNN structure is illustrated in Figure 5. It uses three convolutional layers followed by pooling layers to extract spatial information from the image. Spatial information is subsequently decoded through three fully connected layers (FCLs) to recognize the performed activity.

In [27], the network was tested with a constant size of the pooling layers of 2×2, and a constant number of neurons in the three FLCs, which were 254, 128, and 32, respectively. In order to find the probability distribution of the classes, the softmax function was applied.

However, during our evaluation, we experimented with different kernel sizes of the convolutional layers to improve the recognition rates. The best configurations that we could find were:

- $5 \times 5 - 5 \times 5 - 5 \times 5$. All layers have the same kernel sizes.
- $5 \times 5 - 4 \times 4 - 3 \times 3$. The sizes of the kernels decrease.
- $5 \times 5 - 7 \times 7 - 9 \times 9$. The sizes of the kernels increase.

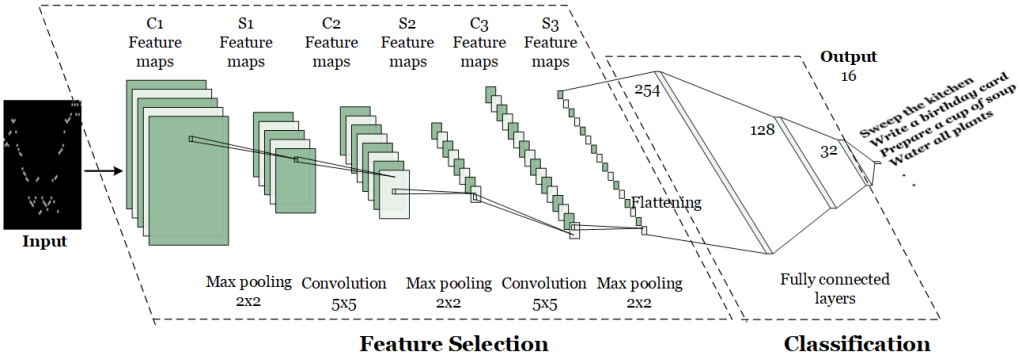


Figure 5. The state-of-the-art DCNN used by Gochoo et al. in [27].

The first configuration exhibited an average performance, while other configurations had varying performances depending on the activity that was recognized. Therefore, for the purpose of this paper, we selected the first configuration. In addition, since the reference paper [27] did not provide internal details such as the activation function, optimizer, learning rate, or loss function used, we adopted a relatively standard configuration. We used the rectified linear unit (ReLU) function as the activation function, the Adam optimizer with a learning rate of 0.00001, and category cross-entropy as the loss function. This loss function is effective for classification problems with the softmax activation function in the output layer [51], ensuring that an image is associated with exactly one class.

4.3. Experimental Results

In the following, we report the results of our experimental evaluation.

4.3.1. Results of the BIM Method with pDLMs

In order to evaluate the effectiveness of our pre-trained DL approach, at first we experimentally evaluated the BIM method proposed by Gochoo et al. [27] with our pDLMs deep learning architecture. For these experiments, we used pDLMs to extract feature vectors from BIM images, and classical machine learning algorithms for activity recognition based on the extracted features. We experimented with the pre-trained architectures presented in Section 3.5.

The experiments are based on two categories of activities: all 16 activities, and the 8 sensor-rich activities alone. The experimental results are reported in Tables 5 and 6. For both categories of activities, the best results are obtained by feature vectors extracted by MobileNet and neural network as a classifier. In particular, we obtain $F_1 = 68.19\%$ for all activities and $F_1 = 90.76\%$ for sensor-rich activities. Considering all 16 activities, MobileNetV2 and VggNet19 achieved F_1 scores (i.e., 67.75% and 67.24%, respectively) close to the ones of MobileNet, using a neural network or random forest. Regarding sensor-rich activities, DenseNet121 and VggNet19 obtain almost the same macro F_1-score results of MobileNet using the random forest classifier ($F_1 = 90.53\%$ and $F_1 = 90.40\%$, respectively).

Among the evaluated classifiers, the worst results were achieved by AdaBoost and QDA, whose performances were poor with every considered pre-trained architecture.

Table 5. Classification results on all sixteen activities using the BIM approach [27] and pDLMs. Bold numbers represent the best results for each experiment.

Pre_Trained Models	Classifier	Accuracy	Precision	Recall	F_1-Score
VggNet19	NB	65.28%	52.79%	56.28%	52.92%
	KNN	75.03%	64.25%	64.00%	63.82%
	DT	63.40%	47.71%	51.78%	46.63%
	SVM	75.48%	61.93%	60.11%	60.57%
	RF	**78.68%**	**67.59%**	**67.84%**	**67.24%**
	Neural Network	**78.73%**	65.48%	65.97%	65.50%
	AdaBoost	20.61%	12.87%	12.87%	8.24%
	QDA	12.74%	10.76%	5.27%	6.47%
ResNet50	NB	48.38%	40.10%	41.89%	38.84%
	KNN	68.38%	59.73%	59.43%	58.99%
	DT	53.91%	39.87%	40.54%	38.16%
	SVM	15.03%	7.99%	12.38%	4.48%
	RF	**76.04%**	**64.94%**	**66.10%**	**65.00%**
	Neural Network	58.78%	45.07%	45.65%	44.94%
	AdaBoost	28.32%	20.69%	28.53%	17.72%
	QDA	14.16%	11.53%	5.84%	7.01%
ResNet50V2	NB	70.25%	57.95%	59.08%	57.69%
	KNN	71.78%	60.06%	60.46%	60.06%
	DT	53.76%	38.86%	44.76%	38.55%
	SVM	75.13%	60.54%	60.00%	60.06%
	RF	**77.72%**	**66.32%**	**68.01%**	**66.44%**
	Neural Network	75.23%	63.22%	63.39%	63.20%
	AdaBoost	22.28%	14.15%	20.10%	10.76%
	QDA	10.36%	8.78%	4.16%	5.21%
MobileNet	NB	70.96%	58.47%	60.76%	58.42%
	KNN	75.28%	65.46%	66.02%	65.33%
	DT	55.89%	40.78%	43.31%	40.76%
	SVM	**79.14%**	65.99%	62.94%	64.09%
	RF	77.87%	66.55%	67.90%	66.70%
	Neural Network	77.31%	**67.95%**	**68.82%**	**68.19%**
	AdaBoost	25.74%	18.66%	24.33%	15.20%
	QDA	11.02%	10.31%	5.16%	5.87%
MobileNetV2	NB	68.02%	55.92%	59.44%	55.83%
	KNN	74.11%	64.82%	64.79%	64.53%
	DT	48.73%	32.89%	36.75%	30.55%
	SVM	**78.27%**	64.84%	64.16%	63.94%
	RF	76.40%	65.73%	66.96%	65.85%
	Neural Network	76.90%	**67.69%**	**68.11%**	**67.75%**
	AdaBoost	20.46%	11.54%	15.31%	6.35%
	QDA	11.12%	9.28%	4.56%	5.53%

Table 5. *Cont.*

Pre_Trained Models	Classifier	Accuracy	Precision	Recall	F_1-Score
DenseNet121	NB	61.62%	48.41%	57.77%	49.16%
	KNN	73.45%	62.67%	63.19%	62.60%
	DT	52.74%	37.70%	43.76%	36.94%
	SVM	77.41%	63.43%	61.69%	62.45%
	RF	**78.07%**	**66.55%**	**68.10%**	**66.82%**
	Neural Network	76.90%	**66.93%**	67.04%	**66.70%**
	AdaBoost	19.54%	12.31%	7.63%	6.11%
	QDA	10.36%	8.69%	3.95%	5.01%

Table 6. Classification results on the eight sensor-rich activities using the BIM approach [27] and pDLMs. Bold numbers represent the best results for each experiment.

Pre_Trained Models	Classifier	Accuracy	Precision	Recall	F_1-Score
VggNet19	NB	79.13%	77.74%	81.94%	79.23%
	KNN	88.84%	88.69%	89.04%	88.76%
	DT	84.16%	80.64%	81.46%	80.21%
	SVM	75.38%	61.32%	59.59%	59.49%
	RF	**90.37%**	**90.01%**	**91.06%**	**90.40%**
	Neural Network	89.10%	87.04%	90.02%	88.17%
	AdaBoost	48.21%	43.72%	62.46%	42.44%
	QDA	12.69%	15.97%	21.94%	9.76%
ResNet50	NB	72.74%	71.68%	72.21%	70.99%
	KNN	89.69%	**90.35%**	88.95%	**89.53%**
	DT	81.26%	78.21%	77.86%	77.42%
	SVM	38.93%	28.00%	24.69%	23.30%
	RF	**89.44%**	89.48%	**89.24%**	**89.33%**
	Neural Network	80.15%	76.72%	77.58%	76.97%
	AdaBoost	51.45%	44.06%	60.19%	43.26%
	QDA	12.35%	15.97%	16.48%	10.38%
ResNet50V2	NB	79.13%	73.16%	85.25%	76.58%
	KNN	86.20%	85.89%	86.21%	85.93%
	DT	79.30%	73.18%	73.41%	72.28%
	SVM	86.97%	82.70%	88.76%	84.05%
	RF	**89.78%**	**88.95%**	**89.85%**	**89.32%**
	Neural Network	**89.27%**	87.97%	**89.89%**	88.72%
	AdaBoost	37.65%	29.19%	33.66%	21.80%
	QDA	11.50%	15.90%	13.33%	8.83%

Table 6. *Cont.*

Pre_Trained Models	Classifier	Accuracy	Precision	Recall	F_1-Score
MobileNet	NB	77.26%	73.78%	83.80%	76.73%
	KNN	85.35%	86.52%	85.97%	86.15%
	DT	80.58%	78.34%	76.88%	77.30%
	SVM	85.78%	80.83%	89.41%	82.85%
	RF	**90.20%**	89.57%	**90.48%**	89.88%
	Neural Network	**90.72%**	**90.26%**	**91.53%**	**90.76%**
	AdaBoost	59.80%	53.55%	57.04%	54.27%
	QDA	11.58%	15.38%	15.08%	8.86%
MobileNetV2	NB	79.81%	77.80%	84.41%	80.26%
	KNN	86.88%	86.83%	86.51%	86.58%
	DT	83.90%	81.95%	81.83%	81.78%
	SVM	88.07%	**88.04%**	88.87%	88.27%
	RF	88.67%	87.85%	88.51%	88.12%
	Neural Network	**89.35%**	**88.90%**	**89.29%**	**89.05%**
	AdaBoost	46.51%	43.62%	58.91%	39.26%
	QDA	11.67%	15.44%	13.74%	9.31%
DenseNet121	NB	74.79%	70.66%	81.35%	73.70%
	KNN	85.18%	84.89%	86.63%	85.40%
	DT	87.56%	86.78%	86.21%	86.40%
	SVM	86.29%	84.22%	88.64%	85.53%
	RF	**90.97%**	**90.53%**	**90.69%**	**90.53%**
	Neural Network	89.95%	89.23%	**90.87%**	89.89%
	AdaBoost	47.10%	39.71%	54.25%	38.38%
	QDA	12.86%	17.01%	16.22%	10.12%

4.3.2. Result of FVE and pDLMs

In this set of experiments, we evaluated our FVE method together with the pDLMs feature extraction technique and classical machine learning models for activity classification. This setup corresponds to the architecture shown in Figure 1.

As in the previous set of experiments, we separately considered all sixteen activities and sensor-rich activities alone. The results are reported in Table 7 and Table 8, respectively. Considering these results, it is evident that the use of our feature-rich visual encoding method improved the recognition performance with respect to the use of the BIM approach. Indeed, the best macro F_1-score achieved by FVE for recognizing all 16 activities is larger than 71%, while the BIM method achieved the best macro F_1-score close to 68%. With FVE and pDLMs, the best performance is obtained with ResNet50 as a feature extractor and random forest as a classifier. Moreover, in this case, the results obtained with the QDA classifier are poor.

Regarding the 8 sensor-rich activities, Table 8 shows that the use of FVE in conjunction with pDLMs leads to improvements with respect to the use of the BIM approach. Indeed, with our method, we achieved $F_1 = 91.52\%$ using DenseNet121 and the random forest classifier. Moreover, VggNet19 and the two versions of MobileNet achieved good results with random forest and support vector machines.

Table 7. Classification results on all sixteen activities using our FVE method and pDLMs. Bold numbers represent the best results for each experiment.

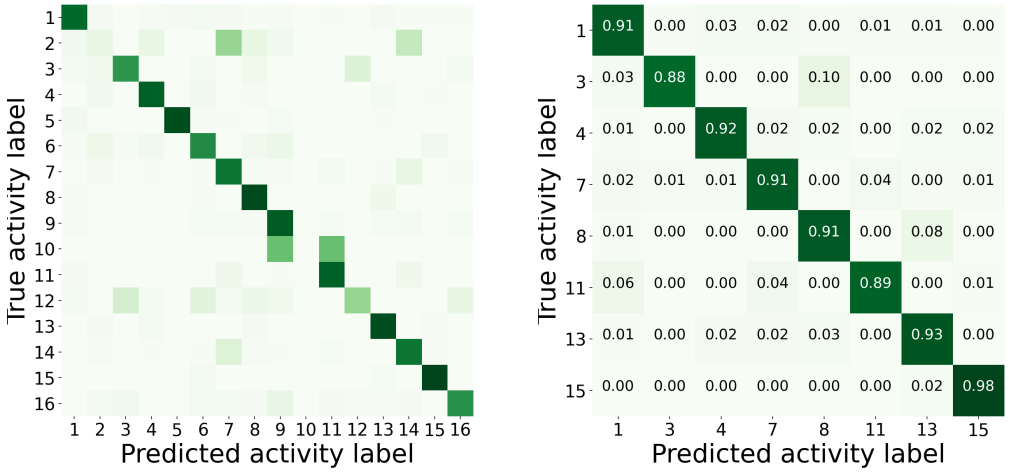
Pre-Trained Models	Approach	Accuracy	Precision	Recall	F_1-Score
VggNet19	NB	64.37%	54.72%	58.38%	54.72%
	KNN	77.41%	66.95%	67.44%	66.37%
	DT	63.50%	44.57%	41.86%	41.94%
	SVM	61.88%	42.88%	40.93%	40.15%
	RF	82.23%	**70.27%**	**71.06%**	**70.45%**
	Neural Network	**79.54%**	66.67%	67.75%	66.59%
	AdaBoost	23.20%	12.92%	8.65%	8.63%
	QDA	8.98%	9.56%	10.59%	5.97%
ResNet50	NB	61.88%	52.08%	53.61%	51.45%
	KNN	78.63%	68.34%	68.28%	67.77%
	DT	61.17%	44.85%	48.34%	44.04%
	SVM	23.25%	14.04%	15.17%	9.48%
	RF	**81.52%**	**70.05%**	**76.31%**	**71.19%**
	Neural Network	72.99%	58.73%	60.66%	58.77%
	AdaBoost	21.17%	11.39%	11.28%	6.23%
	QDA	8.73%	10.39%	9.23%	6.23%
ResNet50V2	NB	66.65%	53.81%	63.60%	55.72%
	KNN	74.06%	64.14%	64.58%	63.58%
	DT	59.29%	43.27%	44.90%	41.84%
	SVM	74.37%	58.43%	60.11%	58.28%
	RF	**80.10%**	67.46%	69.22%	67.96%
	Neural Network	**80.96%**	**68.68%**	**70.01%**	**68.99%**
	AdaBoost	18.83%	10.26%	7.97%	6.37%
	QDA	8.27%	8.84%	7.77%	5.56%
MobileNet	NB	63.55%	51.61%	59.99%	52.96%
	KNN	73.96%	63.70%	64.53%	63.19%
	DT	60.71%	44.98%	49.65%	44.24%
	SVM	75.13%	60.17%	60.93%	60.13%
	RF	79.85%	67.28%	68.24%	67.38%
	Neural Network	**80.10%**	**69.98%**	**72.22%**	**70.72%**
	AdaBoost	21.93%	12.08%	13.04%	7.64%
	QDA	8.38%	9.38%	9.06%	5.51%
MobileNetV2	NB	67.87%	55.46%	61.06%	56.76%
	KNN	75.53%	65.94%	66.71%	65.47%
	DT	61.32%	43.52%	41.29%	41.56%
	SVM	78.07%	65.16%	67.32%	65.13%
	RF	79.59%	67.23%	68.56%	67.61%
	Neural Network	**80.81%**	**69.73%**	**69.87%**	**69.56%**
	AdaBoost	22.69%	12.64%	13.14%	8.36%
	QDA	8.32%	9.17%	7.24%	5.44%
DenseNet121	NB	58.02%	46.55%	56.00%	48.32%
	KNN	74.67%	63.18%	65.31%	63.56%
	DT	64.37%	45.62%	48.16%	43.75%
	SVM	74.67%	61.24%	66.37%	62.44%
	RF	**82.34%**	**70.82%**	**72.19%**	**71.08%**
	Neural Network	80.91%	**70.49%**	**72.91%**	**71.07%**
	AdaBoost	24.01%	13.62%	11.30%	9.69%
	QDA	8.53%	9.44%	7.98%	5.58%

Table 8. Classification results on the eight sensor-rich activities using our FVE method and pDLMs. Bold numbers represent the best results for each experiment.

Pre_Trained Models	Classifier	Accuracy	Precision	Recall	F_1-Score
VggNet19	NB	81.43%	78.14%	86.88%	80.82%
	KNN	88.07%	88.29%	88.13%	88.08%
	DT	83.56%	82.93%	81.89%	82.06%
	SVM	90.20%	89.87%	89.15%	89.41%
	RF	**91.31%**	**91.08%**	**90.71%**	**90.87%**
	Neural Network	**91.14%**	**91.18%**	**90.31%**	**90.68%**
	AdaBoost	56.13%	48.00%	67.50%	51.12%
	QDA	23.17%	19.53%	8.42%	10.85%
ResNet50	NB	72.15%	70.68%	74.09%	71.99%
	KNN	82.37%	83.49%	80.53%	81.52%
	DT	78.02%	79.12%	75.67%	76.45%
	SVM	23.08%	14.30%	16.69%	7.75%
	RF	**88.25%**	**88.03%**	**86.98%**	**87.43%**
	Neural Network	74.96%	70.75%	72.05%	70.96%
	AdaBoost	54.00%	45.11%	58.30%	44.82%
	QDA	23.00%	19.44%	9.56%	10.42%
ResNet50V2	NB	85.26%	82.69%	89.05%	85.13%
	KNN	86.46%	86.56%	85.50%	85.99%
	DT	76.92%	70.55%	68.39%	67.17%
	SVM	**89.27%**	87.01%	**89.63%**	87.88%
	RF	**89.01%**	**89.20%**	88.81%	88.93%
	Neural Network	88.67%	88.73%	**89.52%**	**89.02%**
	AdaBoost	58.01%	54.09%	62.85%	56.26%
	QDA	17.80%	14.72%	5.98%	7.72%
MobileNet	NB	85.86%	82.33%	89.87%	84.92%
	KNN	88.59%	88.09%	88.64%	88.32%
	DT	73.59%	65.34%	67.95%	64.88%
	SVM	**91.48%**	**91.34%**	89.83%	**90.51%**
	RF	**91.23%**	**91.03%**	**90.40%**	**90.69%**
	Neural Network	90.03%	89.48%	88.99%	89.21%
	AdaBoost	52.98%	43.54%	61.64%	42.81%
	QDA	20.70%	18.03%	8.11%	9.87%
MobileNetV2	NB	84.33%	81.52%	88.50%	84.10%
	KNN	88.42%	88.26%	88.20%	88.21%
	DT	80.83%	74.22%	70.11%	70.79%
	SVM	**91.06%**	**90.94%**	**90.91%**	**90.89%**
	RF	89.44%	89.46%	89.74%	89.53%
	Neural Network	89.27%	89.24%	89.45%	89.27%
	AdaBoost	53.92%	43.45%	59.31%	42.58%
	QDA	18.82%	16.12%	6.87%	8.60%
DenseNet121	NB	76.58%	70.25%	86.70%	73.36%
	KNN	87.90%	87.84%	87.97%	87.84%
	DT	78.19%	71.31%	73.68%	69.50%
	SVM	90.20%	89.90%	89.07%	89.42%
	RF	**91.57%**	**91.71%**	**91.41%**	**91.52%**
	Neural Network	90.55%	90.64%	90.03%	90.27%
	AdaBoost	53.32%	45.19%	55.80%	45.82%
	QDA	19.25%	16.79%	22.74%	9.80%

Figure 6 shows the confusion matrices obtained using our FVE method and pDLMs, with DenseNet121 as a feature extractor and random forest as a classifier, which was the setup providing the highest F_1 score. Considering all 16 activities, we noticed that 4 activities (i.e., activities 2, 3, 10, and 12) could not be reliably recognized.

We believe that the superior performance of the FVE method relies on its ability to encode rich and heterogeneous information in the images extracted from sensor events; therefore, the FVE technique is suitable for applications in sensor-rich smart environments.



(**a**) All 16 activities. (**b**) Only 8 sensor-rich activities.

Figure 6. Confusion matrix of our FVE method and pDLMs using DenseNet121 as the feature extractor and random forest as the classifier.

4.3.3. Result of BIM vs. FVE with the DCNN Architecture

In this set of experiments, we compared the performance of our FVE method with the one of the BIM approach, using the DCNN architecture proposed in [27] instead of our approach based on the pDLMs feature extraction and classical machine learning classification. Table 9 shows the achieved results for all 16 activities and the 8 sensor-rich ones.

Table 9. Classification results of BIM vs. FVE with the DCNN architecture [27]. Bold numbers represent the best results for each experiment.

Activity Classes	Approach	Accuracy	Precision	Recall	F_1-Score
16 activities	BIM [27]	77.77%	64.67%	64.69%	64.46%
	FVE	**82.99%**	**71.76%**	**73.97%**	**72.42%**
8 sensor-rich activities	BIM [27]	**90.03%**	89.68%	88.74%	89.16%
	FVE	**90.29%**	**90.78%**	**90.68%**	**90.66%**

Considering all 16 activities, the use of the DCNN architecture for feature extraction and classification emphasizes the improvement achieved by our FVE method with respect to the BIM approach. Indeed, our method achieves $F_1 = 72.92\%$, while the BIM method achieves $F_1 = 64.46\%$. The superiority of our technique is confirmed considering only the eight 8 sensor-rich activities (i.e., $F_1 = 90.66\%$ vs $F_1 = 89.16\%$).

Comparing the performance of the DCNN architecture with the pDLMs approach, we observe that there is no relevant difference in terms of the F_1 score. Hence, we believe that our pre-trained approach is preferable with respect to the DCNN one in terms of computational costs of model training, tuning, and experimentation.

4.4. Computational Cost

With the used dataset, we noticed that the time taken to train the models and for classification was fast. Indeed, in all of the experiments, all classifiers were trained in a few seconds each. The classification was executed in a few milliseconds with all of the tested models. With the configuration achieving the best classification accuracy (i.e., our FVE method and pDLMs using DenseNet121 as the feature extractor and random forest as the classifier), the feature extraction was completed in 4300 ($\pm$853) ms, while the training time was 67,510 ($\pm$1754) ms, and classification was performed in 8 ms. These results were obtained by performing 10 repetitions of each machine learning task. We believe that these times are adequate for most activity recognition applications.

5. Discussion and Limitations

Overall, our technique outperforms other state-of-the-art methods in terms of accuracy while incurring lower computational costs. When considering the eight sensor-rich activities, we achieved an accuracy greater than 91%. However, the accuracy achieved in recognizing all sixteen considered activities is relatively low. This is due to the lack of some sensors in the testbed that are necessary to recognize certain activities. While for some activities, this problem could be solved by introducing additional sensors, other activities require a different approach. For example, recognizing activities such as 'Setup hands for a card game, answer the phone, and describe the rules of the game' or 'Check the wattage of a desk lamp and replace the bulb' solely by observing sensor data is not realistic given the current sensor technology. To recognize such activities while preserving users' privacy, ultra-wideband radar (UWB) technologies and deep learning algorithms would be required [52]. UWB sensors detect solid shapes in proximity based on reflected energy, making them different from RGB camera systems as they do not collect visual data

Our experimental evaluation was carried out with a large set of individuals. However, each individual executed the activities only once, and for a limited time period. Moreover, the activities were scripted, i.e., their executions were not completely naturalistic. The lack of large datasets of naturalistic activities carried out for long time periods by different individuals is a severe limitation for the sensor-based HAR research field [6]. In a fully naturalistic environment, on the one hand, we expect that the inter- and intra-variability of activity execution may negatively affect the recognition performance. On the other hand, the availability of multiple activity instances executed by different people at different times may enhance the generalization of the machine learning model, increasing the recognition rates.

In this paper, we made the assumption that the smart home is inhabited by a single individual. This is a limitation of the current work, which could be addressed by adopting an identity-aware indoor localization system to distinguish the locomotion traces and interactions of the single inhabitants or an algorithm for multi-resident data associations [53].

6. Conclusions and Future Work

Human activity recognition is increasingly being adopted to support healthcare in several applications. In sensor-rich environments, such as smart homes, data acquired from ambient sensors and sensors integrated into everyday objects can be exploited for activity recognition while avoiding the use of obtrusive wearables and cameras. In this paper, we proposed a novel method to extract expressive visual features from the inhabitant's trajectories and interactions with sensorized objects, which are used by a pre-trained deep learning model and a machine learning algorithm to recognize the executed activity. Experimental results and a comparison with the state-of-the-art show the effectiveness of our solution.

Nevertheless, several challenges remain. Our experimental evaluation showed that some of the considered activities could not be detected with sufficient accuracy. By closely inspecting the experimental setup, we noticed that the testbed did not include enough sensors to discriminate between activities executed in the same part of the home and requiring

the use of the same objects. This problem could be solved by adopting additional sensors integrated into objects, or wearable sensors to recognize basic actions that characterize the different activities. Moreover, for the sake of this work, we assumed that the training data used by the machine learning algorithm were acquired in the same environment as the target user. Advanced transfer learning methods should be used to alleviate the problem of data scarcity by allowing the reuse of training data acquired in different environments. Finally, in future work, we plan to improve recognition rates by adding additional features extracted from spatiotemporal data to the images, and to experiment with our techniques in a fully naturalistic environment for longer periods of time.

Author Contributions: Conceptualization, S.Z. and D.R.; methodology, S.Z., S.M.M., and D.R.; software, S.Z.; validation, S.Z. and D.R.; formal analysis, S.Z., S.M.M., and D.R.; investigation: S.Z., S.M.M., and D.R.; resources, S.Z. and D.R.; data curation, S.Z.; writing—original draft preparation, S.Z.; writing—review and editing, S.Z., S.M.M., and D.R.; visualization, S.Z.; supervision, D.R.; project administration, S.Z.; funding acquisition, D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the ASTRID project (Fondazione di Sardegna, annualità 2020) under grant CUP: F75F21001220007.

Institutional Review Board Statement: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors.

Informed Consent Statement: Informed consent was obtained from all individual participants included in the study.

Data Availability Statement: The dataset used in this paper can be downloaded at http://casas.wsu.edu/datasets/assessmentdata.zip, accessed on 1 April 2023.

Acknowledgments: The authors would like to express their gratitude to the anonymous reviewers for their valuable comments and suggestions that have helped to improve the technical content and presentation of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ranasinghe, S.; Al Machot, F.; Mayr, H.C. A review on applications of activity recognition systems with regard to performance and evaluation. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1550147716665520. [CrossRef]
2. Peetoom, K.K.; Lexis, M.A.; Joore, M.; Dirksen, C.D.; De Witte, L.P. Literature review on monitoring technologies and their outcomes in independently living elderly people. *Disabil. Rehabil. Assist. Technol.* **2015**, *10*, 271–294. [CrossRef] [PubMed]
3. Jacob Rodrigues, M.; Postolache, O.; Cercas, F. Physiological and behavior monitoring systems for smart healthcare environments: A review. *Sensors* **2020**, *20*, 2186. [CrossRef] [PubMed]
4. Rashidi, P.; Mihailidis, A. A survey on ambient-assisted living tools for older adults. *IEEE J. Biomed. Health Inform.* **2012**, *17*, 579–590. [CrossRef]
5. Gerland, P.; Hertog, S.; Wheldon, M.; Kantorova, V.; Gu, D.; Gonnella, G.; Williams, I.; Zeifman, L.; Bay, G.; Castanheira, H.; et al. *World Population Prospects 2022: Summary of Results*; United Nations Department of Economic and Social Affairs: New York, NY, USA, 2022.
6. De-La-Hoz-Franco, E.; Ariza-Colpas, P.; Quero, J.M.; Espinilla, M. Sensor-based datasets for human activity recognition–a systematic review of literature. *IEEE Access* **2018**, *6*, 59192–59210. [CrossRef]
7. Zolfaghari, S.; Keyvanpour, M.R. SARF: Smart activity recognition framework in Ambient Assisted Living. In Proceedings of the 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, Poland, 11–14 September 2016; pp. 1435–1443.
8. Khodabandehloo, E.; Riboni, D.; Alimohammadi, A. HealthXAI: Collaborative and explainable AI for supporting early diagnosis of cognitive decline. *Future Gener. Comput. Syst.* **2021**, *116*, 168–189. [CrossRef]
9. Suthar, B.; Gadhia, B. Human activity recognition using deep learning: A survey. In *Data Science and Intelligent Applications, Proceedings of ICDSIA 2020, Gujarat, India, 24–25 January 2020*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 217–223.
10. Gupta, N.; Gupta, S.K.; Pathak, R.K.; Jain, V.; Rashidi, P.; Suri, J.S. Human activity recognition in artificial intelligence framework: A narrative review. *Artif. Intell. Rev.* **2022**, *55*, 4755–4808. [CrossRef]

11. Serpush, F.; Menhaj, M.B.; Masoumi, B.; Karasfi, B. Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System. *Comput. Intell. Neurosci.* **2022**, *2022*. [CrossRef]
12. Manca, M.M.; Pes, B.; Riboni, D. Exploiting Feature Selection in Human Activity Recognition: Methodological Insights and Empirical Results Using Mobile Sensor Data. *IEEE Access* **2022**, *10*, 64043–64058. [CrossRef]
13. Stavropoulos, T.G.; Papastergiou, A.; Mpaltadoros, L.; Nikolopoulos, S.; Kompatsiaris, I. IoT wearable sensors and devices in elderly care: A literature review. *Sensors* **2020**, *20*, 2826. [CrossRef]
14. Gerina, F.; Massa, S.M.; Moi, F.; Reforgiato Recupero, D.; Riboni, D. Recognition of cooking activities through air quality sensor data for supporting food journaling. *Hum.-Centric Comput. Inf. Sci.* **2020**, *10*, 1–26. [CrossRef]
15. Barra, S.; Carta, S.M.; Giuliani, A.; Pisu, A.; Podda, A.S.; Riboni, D. FootApp: An AI-powered system for football match annotation. *Multimed. Tools Appl.* **2022**, *82*, 1–21. [CrossRef]
16. Steels, T.; Van Herbruggen, B.; Fontaine, J.; De Pessemier, T.; Plets, D.; De Poorter, E. Badminton activity recognition using accelerometer data. *Sensors* **2020**, *20*, 4685. [CrossRef]
17. Cook, D.J.; Krishnan, N.C. *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
18. Beddiar, D.R.; Nini, B.; Sabokrou, M.; Hadid, A. Vision-based human activity recognition: A survey. *Multimed. Tools Appl.* **2020**, *79*, 30509–30555. [CrossRef]
19. Kim, E.; Helal, S.; Cook, D. Human activity recognition and pattern discovery. *IEEE Pervasive Comput.* **2009**, *9*, 48–53. [CrossRef]
20. Albert Florea, G.; Weilid, F. Deep Learning Models for Human Activity Recognition. Bachelor Thesis, University of Malmo, Malmo, Sweden, 2019.
21. Keyvanpour, M.R.; Zolfaghari, S. Augmented feature-state sensors in human activity recognition. In Proceedings of the 2017 9th International Conference on Information and Knowledge Technology (IKT), Tehran, Iran, 18–19 October 2017; pp. 71–75.
22. Foerster, F.; Smeja, M.; Fahrenberg, J. Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring. *Comput. Hum. Behav.* **1999**, *15*, 571–583. [CrossRef]
23. Alemayoh, T.T.; Lee, J.H.; Okamoto, S. New sensor data structuring for deeper feature extraction in human activity recognition. *Sensors* **2021**, *21*, 2814. [CrossRef]
24. Challa, S.K.; Kumar, A.; Semwal, V.B. A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data. *Vis. Comput.* **2022**, *38*, 4095–4109. [CrossRef]
25. Babangida, L.; Perumal, T.; Mustapha, N.; Yaakob, R. Internet of things (IoT) based activity recognition strategies in smart homes: A review. *IEEE Sens. J.* **2022**, *22*, 8327–8336. [CrossRef]
26. Samaneh, Z. Human Activity Recognition in Smart Homes: Research Challenges Classification. *Changes* **2017**, *14*, 15.
27. Gochoo, M.; Tan, T.H.; Liu, S.H.; Jean, F.R.; Alnajjar, F.S.; Huang, S.C. Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN. *IEEE J. Biomed. Health Inform.* **2018**, *23*, 693–702. [CrossRef] [PubMed]
28. Cook, D.J. Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* **2010**, *2010*, 1. [CrossRef] [PubMed]
29. Chen, L.; Hoey, J.; Nugent, C.D.; Cook, D.J.; Yu, Z. Sensor-based activity recognition. *IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.* **2012**, *42*, 790–808. [CrossRef]
30. Azkune, G.; Almeida, A.; López-de Ipi na, D.; Chen, L. Extending knowledge-driven activity models through data-driven learning techniques. *Expert Syst. Appl.* **2015**, *42*, 3115–3128. [CrossRef]
31. Zolfaghari, S.; Zall, R.; Keyvanpour, M.R. SOnAr: Smart Ontology Activity recognition framework to fulfill Semantic Web in smart homes. In Proceedings of the 2016 Second International Conference on Web Research (ICWR), Tehran, Iran, 27–28 April 2016; pp. 139–144.
32. Zolfaghari, S.; Keyvanpour, M.R.; Zall, R. Analytical review on ontological human activity recognition approaches. *Int. J. E-Bus. Res. (IJEBR)* **2017**, *13*, 58–78. [CrossRef]
33. Ahmed, N.; Rafiq, J.I.; Islam, M.R. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors* **2020**, *20*, 317. [CrossRef] [PubMed]
34. Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; Abdelzaher, T. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 351–360.
35. Zolfaghari, S.; Khodabandehloo, E.; Riboni, D. TraMiner: Vision-based analysis of locomotion traces for cognitive assessment in smart-homes. *Cogn. Comput.* **2022**, *14*, 1549–1570. [CrossRef]
36. Zolfaghari, S.; Loddo, A.; Pes, B.; Riboni, D. A combination of visual and temporal trajectory features for cognitive assessment in smart home. In Proceedings of the 2022 23rd IEEE International Conference on Mobile Data Management (MDM), Paphos, Cyprus, 6–9 June 2022; pp. 343–348.
37. Khodabandehloo, E.; Alimohammadi, A.; Riboni, D. FreeSia: A Cyber-physical System for Cognitive Assessment through Frequency-domain Indoor Locomotion Analysis. *ACM Trans. Cyber-Phys. Syst. (TCPS)* **2022**, *6*, 1–31. [CrossRef]
38. Riboni, D.; Pareschi, L.; Bettini, C. Privacy in georeferenced context-aware services: A survey. In *Privacy in Location-Based Applications: Research Issues and Emerging Trends*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 151–172.
39. Filali, Y.; EL Khoukhi, H.; Sabri, M.A.; Aarab, A. Efficient fusion of handcrafted and pre-trained CNNs features to classify melanoma skin cancer. *Multimed. Tools Appl.* **2020**, *79*, 31219–31238. [CrossRef]

40. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis. (IJCV)* **2015**, *115*, 211–252. [CrossRef]
41. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
42. Rahimzadeh, M.; Attar, A. A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Inform. Med. Unlocked* **2020**, *19*, 100360. [CrossRef]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
44. Ikechukwu, A.V.; Murali, S.; Deepu, R.; Shivamurthy, R. ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. *Glob. Transitions Proc.* **2021**, *2*, 375–381. [CrossRef]
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
46. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
47. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
48. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
49. Zhang, C.; Benz, P.; Argaw, D.M.; Lee, S.; Kim, J.; Rameau, F.; Bazin, J.C.; Kweon, I.S. Resnet or densenet? introducing dense shortcuts to resnet. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikola, HI, USA, 3–8 January 2021; pp. 3550–3559.
50. Cook, D.J.; Schmitter-Edgecombe, M.; Crandall, A.; Sanders, C.; Thomas, B. Collecting and disseminating smart home sensor data in the CASAS project. In Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, Boston, MA, USA, 4 April 2009.
51. Das, R.; Chaudhuri, S. On the separability of classes with the cross-entropy loss function. *arXiv* **2019**, arXiv:1909.06930.
52. Noori, F.M.; Uddin, M.Z.; Torresen, J. Ultra-wideband radar-based activity recognition using deep learning. *IEEE Access* **2021**, *9*, 138132–138143. [CrossRef]
53. Riboni, D.; Murru, F. Unsupervised recognition of multi-resident activities in smart-homes. *IEEE Access* **2020**, *8*, 201985–201994. [CrossRef] |