Visualization as Intermediate Representations (VLAIR) for Human Activity Recognition

Ai Jiang, Miguel A. Nacenta, Kasim Terzic, and Juan Ye
University of St Andrews, UK
aj99@st-andrews.ac.uk

ABSTRACT

Ambient, binary, event-driven sensor data is useful for many human activity recognition applications such as smart homes and ambient-assisted living. These sensors are privacy-preserving, unobtrusive, inexpensive and easy to deploy in scenarios that require detection of simple activities such as going to sleep, and leaving the house. However, classification performance is still a challenge, especially when multiple people share the same space or when different activities take place in the same areas. To improve classification performance we develop what we call a *Visualization as Intermediate Representations* (VLAIR) approach. The main idea is to re-represent the data as visualizations (generated pixel images) in a similar way as how visualizations are created for humans to analyze and communicate data. Then we can feed these images to a convolutional neural network whose strength resides in extracting effective visual features. We have tested five variants (mappings) of the VLAIR approach and compared them to a collection of classifiers commonly used in classic human activity recognition. The best of the VLAIR approaches outperforms the best baseline, with strong advantage in recognising less frequent activities and distinguishing users and activities in common areas. We conclude the paper with a discussion on why and how VLAIR can be useful in human activity recognition scenarios and beyond.

KEYWORDS

Information visualization, intermediate representations, human activity recognition, convolutional neural networks, smart homes

ACM Reference Format:
Ai Jiang, Miguel A. Nacenta, Kasim Terzic, and Juan Ye. 2018. Visualization as Intermediate Representations (VLAIR) for Human Activity Recognition. In *PervasiveHealth '20: EAI International Conference on Pervasive Computing Technologies for Healthcare, May 18–20, 2020, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Computing applications that interact with — or directly serve — humans often depend on the successful recognition of human activity. For example, a smart home system that adapts heating to the inhabitants' behaviour (different temperatures for sleeping, cooking or reading) has the potential to simplify human-interaction, increase comfort and reduce energy consumption, but relies on sensing and algorithms to classify people's activities. Accurate and practical human activity recognition (HAR) can also enable life-critical applications such as health monitoring and ambient-assisted living [7]. For example, activity recognition can detect and analyse anomalies in daily behaviour patterns and further assist disease diagnosis of older adults [46].

There are currently two dominant approaches to HAR: video-based and sensor-based [9]. While video-based HAR utilizes video to observe people's actions, sensor-based HAR relies on sensor data that records human motion [19] or ambient changes [29] from diverse sensors such as accelerometers and acoustic sensors. Sensor-based HAR has been widely used to analyze physical human activity (e.g., walking [13]) as well as biological human dynamics (e.g., breathing [12]), presumably because it does not require cameras, which can feel intrusive [41]. The deployment of non-image sensors can also be preferable to cameras because sensors might be more appropriate and accurate for certain signals, less costly, require less energy and data processing, and might be easier to deploy. Here we focus on HAR based on binary sensors installed in homes, such as those from the ARAS [1] and CASAS [10, 11] datasets, which are inexpensive, non-intrusive and easy to deploy [49, 50].

Regardless of sensing technologies, accurately classifying activities with sufficient granularity to enable sophisticated applications remains a challenge. When binary sensors are used, this is compounded with the lack of obvious ways to integrate the location of the sensor and the timing of its activation for the classification algorithm. For example, when a user is wandering vs. working in the bedroom, the same sensors might be activated, resulting in hard-to-separate sensor features and leading to low accuracy distinguishing these two activities [59].

Here we present a novel approach to improve accuracy in HAR. Inspired by the research field of visualization (e.g., [5, 32]), we transform the raw data into visual representations that are then used in a Convolutional Neural Network (CNN). This approach, which we call *VisuaLizations As Intermediate Representation* (VLAIR), allows us to encode spatial and temporal information of sensor onsets in a straightforward and human-readable manner.

We perform a comparison of the VLAIR approach with a set of baselines that include a CNN trained directly on sensor features and a host of traditional machine learning approaches with several data transformations, some of which are designed to encode temporal relationships between sensor activation as well as spatial information. We show that VLAIR mappings perform comparably or better than the alternative classification approaches and, significantly, show important improvements for specific activities that are almost never detected by previously published algorithms.

PervasiveHealth '20, May 18–20, 2020, NY, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Our contribution is twofold: we introduce VLAIR and we show, through a series of experiments, how it can increase accuracy for HAR from binary sensor data. Classification improvements are of direct value to developers of HAR-dependent applications. The graphical nature of VLAIR also offers promise improving the explainability of activity recognition, since the visual representations of the data are more accessible to human observers. Because of the flexibility and the large remaining room for sophistication in the design of visualization mappings, we believe that VLAIR can be applied beyond binary sensor activity recognition and deliver further classification performance gains in this and other domains.

2 RELATED WORK

In this paper we propose VLAIR as a domain transformation technique that reappropriates computer vision models for classification of human activities. Thus, there are three areas of work that are particularly relevant: existing approaches to sensor-based HAR (same domain), other transformations of sensor data used for classification (same part of the workflow), and previous examples of reappropriation of computer-vision models for classification beyond the domain of their training data sets (related approach). Because VLAIR leverages knowledge in data visualization for humans, we also provide a brief introduction and key references in this area.

2.1 Sensor-based Human Activity Recognition

Sensor-based HAR infers human daily activities from a range of wearable and ambient sensors embedded in an environment. The general process of sensor-based HAR typically involves collecting and integrating data from sensors, extracting features from the raw data [19], and applying learning techniques to infer human behaviors. Various algorithms, including decision trees, support vector machines and, more recently, deep neural networks [53], have been applied to classification, recognition and segmentation tasks. Deep learning can demonstrably learn complex correlations between low-level sensor data and high-level human activities [33]. For example, Morales et al. [33] employed a CNN to extract features from raw accelerometer signals and a Recurrent Neural Network (RNN) to learn sequential relationships of extracted features in human activities. Radu et al. [39] designed a multimodal architecture for integrating sensor data from different modalities to infer activities. Sprint et al. employed change detection on Fitbit's time series data to track changes in physical activities during inpatient rehabilitation [46]. In VLAIR, we look into how to apply computer vision-based deep neural networks to learn intrinsic sensor features on visualizations.

In this work, we focus on data from ambient binary sensors. This type of data — based on events — is intrinsically different from regularly sampled sensor data such as acceleration and orientation from wearable sensors. There are two classic feature representations to represent binary sensor data: *binary* and *numerical* representations. Binary representations record whether a sensor is activated during a certain interval, while numerical representations record the number of times or the ratio of time that each sensor is activated during the interval [6]. Numerical representations can detect fine differences in activities that trigger a similar set of sensors and thus are more commonly adopted [14, 59]. However, numerical representations alone do not capture sequential or temporal information such as the sensor activation order of sensors being activated, and when a sensor is being activated, or spatial information such as the layout of an environment and the spatial relations between deployed sensors. Researchers have attempted to encode hour/minute/second information in feature representations [14, 15], and tried sequential mining approaches on sensor events to learn the activation order [**?**]. One of the motivations for the VLAIR approach is to find simple ways to integrate this kind of temporal and spatial information to improve activity recognition.

2.2 Sensor Data Transformation

Some existing techniques transform raw input sensor data into representations that are learnable through CNNs. For example, an early data-driven approach [60] treats each dimension of accelerometer signals as a channel of an RGB image to capture local dependencies of sensor signals, and extracts scale-invariant sensor features by using CNN to infer human activities such as 'walking' and 'drink when standing'. Other similar approaches are to adapt 1D sensor signal inputs to form 1D virtual images and then leverage the advantages of CNNs to automatically extract and learn discriminative sensor features [38, 54].

Ha et al. [18] combine all dimensions of sensor input forming an image and use a 2D kernel to effectively capture spatial dependency over sensors as well as local dependency over time. They take into account two different modalities: sensors in different positions and different sensing types. They group sensors in different positions to capture spatial dependency over signals via the 2D kernel and separate sensor types by padding zeros between them. Compared with using a 1D kernel, their 2D kernel method can obtain distinguishable features from multiple sensors; e.g., accelerometers, gyroscopes and magnetometers, and get better performance on common human activity recognition tasks [26, 28, 40].

Singh et al. [45] use the knowledge from CNNs pre-trained on image data for their sensor-based classification task. They linearly transfer 2D pressure value mappings from force-sensitive resistor fabric sensors into gray-scale images. By using a pre-trained CNN as feature extractor, they unify the feature extraction process for pressure sensor data to better identify users from their footsteps. However, their modality transformation is task specific and can only be applied on matrix-sensors such as a pressure mat. It does not generalize well for other types of sensor data such as ambient, binary sensor data.

These examples inspired (and are precursors of) VLAIR because they transform, sometimes in spatialized form, raw data into formats learnable by a CNN. VLAIR takes this further by leveraging visual mapping techniques (including abstract re-representations) from the visualization field thus far only in use by humans.

2.3 Visualization For Sensor Data

The process of visualization transforms information (e.g., numerical data) into visual artifacts with the aim of facilitating the human exploration of, analysis of, communication of and reasoning with said information. The study of visualization is over three centuries old [32], a well-established field of study and practice [5, 34], and

Table 1: Raw sensor data in the CASAS Twor dataset

Timestamp	Sensor	Value	Annotated Activity
2009-08-24 00:04:38.039369	M047	ON	R1_Sleep begin
2009-08-24 00:05:04.099416	M046	ON	
2009-08-24 00:05:19.004364	M037	ON	R1_Bed_Toi._Trans. begin
⋮	⋮	⋮	⋮
2009-08-24 00:05:19.004364	M037	OFF	R1_Bed_Toi._Trans. end
2009-08-24 00:00:25.061429	M046	OFF	R1_Sleep end

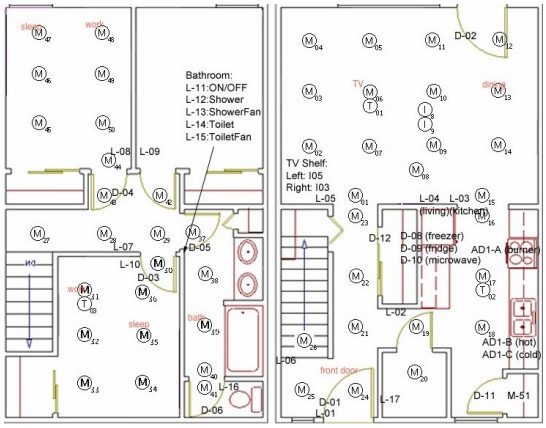
an important part of certain human activities such as data analysis. A key concept of visualization is the *mapping*, which is the relationship between the structure of the raw data and the properties of the resulting visual artifacts. For example, a line chart of a country's GDP over time *maps* the vertical position of line points to the country's GDP value, and the horizontal location to the time (year). A very large number of mappings is possible for each data schema, and the combinations of mappings in a visualization greatly influences whether it can be processed effectively by the human perceptual and cognitive system [4, 8]. Combinations of mappings (visualizations) can be designed by humans, either independently or aided by algorithms [57], or selected by software [35]. Visualizations have been used to represent HAR data (e.g.,[61]) and to make machine learning algorithms interpretable [52] but, to our knowledge, not as a direct input to improve classification performance.

3 EVENT-DRIVEN BINARY SENSOR DATA

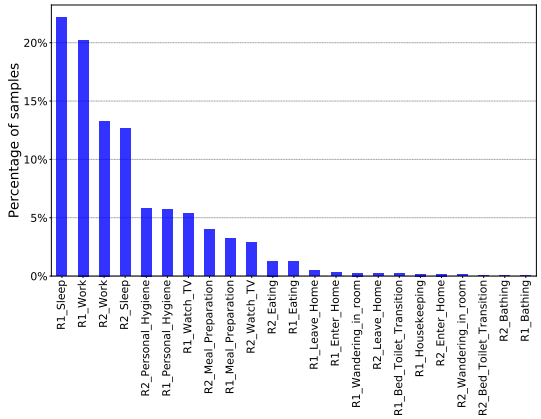
Binary event-driven sensors are sensors that report '1' or 'ON' when being activated. Examples include RFID sensors that are activated when a tag is in close proximity [30], infra-red passive motion sensors being activated when a user is in front of them [10], or switch sensors that indicate the state of physical objects, such as whether a cabinet door is open or closed [51]. These sensors can unobtrusively monitor users' activities and can be deployed on a wide range of objects.

In this paper we examine a state-of-the-art third-party dataset from the CASAS project published by Washington State University [3, 10]: the Twor (Kyoto) dataset. The dataset reflects the whereabouts of the residents of a student apartment collected through 83 binary sensors. A sensor is activated if a person is present in front of the sensor. The apartment layout and sensor deployment are shown in Figure 1a and examples of raw sensor data are listed in Table 1. The raw sensor data consists of a temporally ordered sequence of binary sensor events that are annotated with activity labels, to which we add the static 2D coordinates of the location of the individual sensors that we extracted from the sensor deployment maps published with the dataset. The labels were produced by the multiple annotators of the CASAS team using the house plan, sensor positions and forms completed by the residents with information of the times and locations of their activities [2].

The apartment housed two residents (*R1* and *R2*), who performed their daily activities including working, preparing meals, and sleeping. Figure 1b shows the distribution of the annotated activities. We deliberately selected this dataset because of the density of sensor deployment and because it is well annotated with a wide range of activities. Real-world environments usually contain multiple users and recognizing multi-user concurrent activities is essential for



(a) Spatial layout



(b) Activity class distribution

Figure 1: Spatial layout and activity distribution of the CASAS Twor dataset

scenarios such as smart homes. However, recognizing the activity of two people through identity-agnostic sensors is challenging, and it mainly relies on learning the subtle differences between users when they perform the same activity [59]. We apply state-of-the-art techniques to segment the raw sensor data (see Table 1) into a fixed-length interval, and preprocess the dataset to only include non-concurrent activities so that we have exactly one activity label with each corresponding converted image.

Although we suspect that our approach could be particularly suitable for analysis of concurrent events, we use the non-overlapping activity sub-dataset as a starting point; therefore, overlapping activity classification falls out of our current scope and should be addressed in future work.

4 VISUALIZATION OF SENSOR DATA

In this section we describe how we transform the raw data into visualizations, which is the core of the VLAIR approach. A visualization type is defined by one or more *mappings* from direct or derived data elements to graphical elements. A large number of

mappings and combinations of mappings are possible. A visualization is determined by a designer (human or machine) through their choice of mappings which, in turn, determines the effectiveness of the visualization for observer tasks. For example, designers might choose to map the dimensions of the data that they want to emphasize to the horizontal and vertical positions of objects in the plane, which have been shown to be the most powerful *visual variables* (or *channels*) for human perception of quantitative data [21]. VLAIR's main difference from ordinary visualization is that the observer is a machine-vision algorithm rather than a human; nevertheless we use simple mappings that we know would be reasonable for people as a starting point. The rationale is that the network structure of CNNs is inspired by the human visual cortex [17, 43].

We iteratively developed a series of five visualization types. The mappings are chosen to assign the features of the data that we found most promising a priori (e.g., the sensor layout, the activation ratios of the sensors, the sequences of activation) to visual variables that are most effective for humans according to best knowledge in information visualization [4, 31, 34, 55] and empirical research [8, 21]. Position in the 2D plane usually ranks top in lists of visual channels ordered by efficiency and accuracy; therefore, all our mappings match the position of sensors to the location of visual elements in the 2D visualization (the 2D location visual variable). This is also a mapping that has been used in the past for a similar purpose ([45]) and that is understandable by human observers.

The mappings and visualization types presented below are only a tiny sliver of what is possible; they provide an initial informed guess of what can work, based on what works for humans. All our visualizations are based on assigning the spatial layout of sensors to horizontal and vertical position in the image. We then progressively generate other variants by adding information on sequences, sensor activation ratios, and temporal information through additional visual variables. Many other visualizations are possible, but their systematic exploration is outside the scope of this paper.

The subsections below describe the mappings that we have tried, except for the spatial mapping already described above, which all visualizations use. Several mappings are combined in different ways to create the five visualization variants displayed in Figure 2. The output of the VLAIR encoding process is then fed to a CNN model.

4.1 Sensor Activation to Color Intensity

For each sensor i out of S total sensors in an T-length interval (see Section 3) we calculate the activation ratio according to Equation (1), where N_i is the number of the times that the ith sensor is activated during the interval. The resulting value is determines the color intensity of the image pixel representing the ith sensor's location.

$$p_i = \begin{cases} \frac{N_i}{\sum_{j=1}^{S} N_j} & \text{if } i \in [1, S] \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The Activation Intensity visualization, shown in Figure 2 column 2, uses only this approach.

4.2 Sensor Activation to Circle Radius

For each sensor i out of S total sensors in a T-length interval we place a circle of radius r_i, determined by Equation (2), where k denotes the time index that the ith sensor is activated, and $t_{i,k}$ is the duration of the kth time segment where the ith sensor is continuously being recorded as active. N_i is the total number of times that the ith sensor is activated in a T-length interval, and r_{base} is the pre-defined maximum radius for a visited sensor point. All visualizations except the Activation Intensity mapping (see Figure 2, columns 3 to 6) use this.

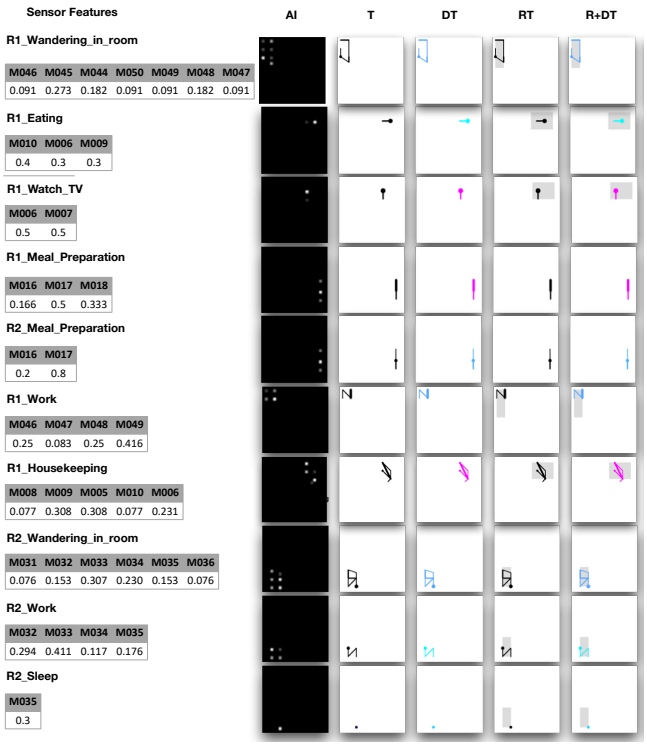


Figure 2: Summary of sensor features and transformed images for a collection of activities in the Twor dataset.

$$r_i = \frac{\sum_{k=1}^{N_i} t_{i,k}}{T} * r_{\text{base}} \tag{2}$$

4.3 Node Transitions to Width-variable Traces

We encode sequences of events by drawing traces. Considering each activated sensor as a node, nodes activated consecutively draw a line between the positions of these nodes. The thickness of the line between nodes i and j varies according to Equation (3), where w_{base} is the pre-defined minimum width for a line indicating one-time visit and $N_{i,j}$ is the total number of visits between sensor i and sensor j in a T-length interval.

$$w_{i,j} = N_{i,j} * w_{\text{base}} \tag{3}$$

All visualizations except the Activation Intensity mapping (see Figure 2, columns 3 to 6) use this.

4.4 Time-of-day to Color

Time of day might be relevant for distinguishing activities; e.g., cooking may be less likely in the early hours of the day. Therefore

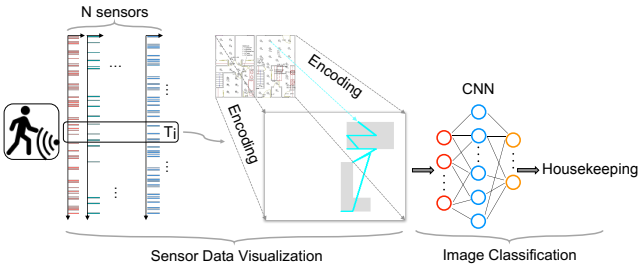


Figure 3: VLAIR Workflow

we encode time-of-day information by drawing all elements in the image with a color that corresponds to the time of the day. We pick 24 different levels from a colormap taken from the Python plotting library *Matplotlib*, which range from blue (early morning) to red (late night). This color coding is used in the Daytime-encoded and Room+Daytime-encoded Traces (see Figure 2, columns 4 and 6).

4.5 Sensor Location Context to Room Shape

Sensors are located in rooms, which are natural delimiters of human activities. We add grey shadings of room areas where at least one sensor is activated. The Room-encoded Traces and Room+Daytime-encoded Traces (Figure 2, columns 5 and 6) use this.

5 ARCHITECTURE

Our proposed approach takes raw binary sensor data as input, segments them into fixed intervals, and transforms each segment into a VLAIR image, which will be classified by a CNN into an activity label (see Figure3). In the following, we briefly describe the CNN architecture.

We aim for a small-sized architecture that can run on a relatively resource-constrained device, requires little training time, and only needs to deal with simple images that largely consist of primitive shapes such as lines, rectangles, and circles. Driven by this purpose, we design a CNN composed of three 2D convolutional layers each followed by max pooling layer, a dense layer with 512 neurons followed by a dropout layer, and a softmax classification layer.

The forward propagation of the CNN model is as follows. For each convolutional layer l, the input volume of $N^l_{in} \times N^l_{in}$ will be processed by a convolutional operation with filters k_{conv} of size 3 with a fixed stride s_{conv} of size 1. The same padding p is used to preserve spatial resolution and a pooling over $k_{pool} \times k_{pool}$ (of a size 2) pixel window with a stride s_{pool} (of size 2) converts into an output volume of $N^l_{poolOut} \times N^l_{poolOut}$. The depth of input and output volumes depends on the number of filters used in each layer. The calculation is defined as:

$$N^l_{convOut} = \left(\frac{N^l_{in} + 2p - k_{conv}}{s_{conv}} \right) + 1, \tag{4}$$

$$N^l_{poolOut} = \left(\frac{N^l_{convOut} - k_{pool}}{s_{pool}} \right) + 1 \tag{5}$$

where $N^l_{convOut}$ and $N_{poolOut}$ are the dimensions of the convolutional and pooling output at layer l.

Table 2: CNN Configuration

Type	Configurations
Input	240 * 240 * 1 (3) image
Convolution	Filter: 64, Kernel size: 3 * 3, Stride: 1
Maxpooling	Kernel size: 2 * 2, Stride: 2
Convolution	Filter: 64, Kernel size: 3 * 3, Stride: 1
Maxpooling	Kernel size: 2 * 2, Stride: 2
Convolution	Filter: 128, Kernel size: 3 * 3, Stride: 1
Maxpooling	Kernel size: 2 * 2, Stride: 2
Fully connected	512 neurons
Softmax	23 neurons

For the fully-connected layer, the input X^{l-1}_i is the flattened result of each image i from the last convolutional layer. A regular neural network operation is then applied with weights W^l_i between layer l and layer $l-1$ plus a bias term b^l_i for this layer. The classification output $\overline{Y_i}$ for an image i, an inferred activity label, is calculated by a non-linear *Softmax* activation function on the last output layer:

$$Z^l_i = W^l_i * X^{l-1}_i + b^l_i, \text{ and } \overline{Y_i} = Softmax(Z^l_i). \tag{6}$$

To find the best hyperparameters for our CNN, we have conducted grid search on the number of convolutional layers, the number of filters per layer, and the size of kernels and fully connected layers. The final CNN model configuration is displayed in Table 2. Batch normalization [25] is employed to effectively increase the training speed. It also associates the dropout [47] strategy with fully connected layers. The dropout rate was maintained at 0.5 throughout training. Note that usage of deeper and wider convolutional layers can be beneficial when extracting more complicated features, however, we wanted to keep our model as light as possible.

6 EVALUATION METHODOLOGY

To validate the VLAIR approach we test the visualization types shown in Figure 2 against classic feature-based machine learning approaches. In the following, we describe the evaluation process.

6.1 Data Preprocessing

We segment sensor events into 60-second slices. Previous work [27, 59] has found this interval appropriate for this kind of classification task in this kind of data; smaller periods do not capture sufficient events to successfully differentiate activities, and longer periods are detrimental to timely prediction and may contain data from multiple activities. Previous work [58, 59] also provides a foundation for the selection of features, such as sensor activation ratios, sensor event order, and the activation time, which we adopt for our VLAIR visualizations.

6.2 Configuration and Metrics

For each of the VLAIR approaches, we run 100 iterations of 5-fold cross validation, which is considered appropriate for long-term datasets and has been applied on the same datasets [16, 59]. We choose hyperparameters for learning rate and the optimizer in line with the state-of-the-art vision-based approaches in deep

learning [36]. We experimented with different learning rates and two optimizers (*SGD* and *Adam*), and selected a combination that converges fast and achieves higher accuracy on the validation set. To reduce overfitting, we stop training when the validation loss does not improve for 15 consecutive epochs.

The validation set is obtained by splitting the training data (the K-1 folds) into 80% for model training and 20% for validation. We use F1-scores as our main accuracy measure because they balance precision and recall. More specifically, we use *macro F1-scores* (averaging the F1-scores from all activity classes) and *micro F1-scores* (averaging across all instances). For each configuration we calculate the scores from the average of 5-fold cross validations. We also measure execution times in all our trials, which we run on the same dedicated machine: an Intel workstation with a processor i5-8500 CPU @ 3.00GHz, 6 cores and 64G memory with a NVIDIA Quadro p6000 GPU. The training time for our CNN on VLAIR images is averaged 17 seconds per epoch.

6.3 Baseline

Regarding non-VLAIR alternatives, there are largely two orthogonal dimensions of variation: data representation and model type. Data representation refers to the feature set that is provided to the machine learning algorithms. We consider three alternatives: raw (RAW), location and time (LOC+TIME), and Mutual Information (MI). The RAW representation contains activation intensity for each sensor in each interval, as described in Section 4, Equation 1. The LOC+TIME representation provides additional spatial and temporal information in an equivalent way to the VLAIR approaches by adding sensor coordinates, bounding room data, hour information, and traces (transition) information to the data already in RAW. Finally, the MI representation encodes the contribution of each sensor event based on temporal and sensor mutual information, as described in [27]. In this approach, *temporal dependency* measures the contribution of a sensor event in a segment based on its temporal distance to the last event in the segment and *sensor dependency* measures the probability of two sensors occurring consecutively. Differently from the RAW representation, which counts sensor events, the MI feature vector weighs the influence of sensor events based both on their temporal dependency and sensor mutual information. In the end, the feature dimensions for RAW, MI and LOC+TIME representations are 43, 43, and 414.

The baseline algorithms include a CNN with the same architecture presented in Section 5 and classic feature-based machine learning algorithms previously reported on this kind of data [42, 58]: Naive Bayes (NB), K Nearest Neighbors (KNN), Classification And Regression Tree (CART), Support Vector Machine with linear and RBF kernels (SVM, SVM-RBF), and Random Forests (RF). All implementations come from Python's scikit-learn library [37]. Because NB, KNN, CART and linear-kernel SVM results are much poorer than all the other approaches, we omit them from the result reporting and the discussion.

7 RESULTS

Here we present the main findings grouped in three sections. First, we report the differences between VLAIR variants, then the comparisons between VLAIR and Baseline approaches, and finally evidence on how classifiers recognize activities by different users taking place in common areas.

Table 3: Comparisons of Micro- and Macro-F1 scores between VLAIR and baselines.

Technique	Input	Micro-F1	Macro-F1
CNN	**R+DT**	**0.80**	**0.58**
RF	**LOC+TIME**	0.79	0.55
	MI	0.63	0.40
	RAW	0.77	0.48
CNN	**LOC+TIME**	0.74	0.34
	MI	0.34	0.11
	RAW	0.75	0.28
SVM	**LOC+TIME**	0.77	0.47
	MI	0.53	0.25
	RAW	0.77	0.46

7.1 Comparison between Different Mappings

Figure 4 visually summarizes the F1-scores of the CNN trained with different VLAIR mappings. The richest visualization is R+DT, which encodes the most information, is the most accurate (21 classes, 91%), ahead of all other variants. Among all VLAIR variants, the R+DT and DT visualizations offer the best micro and macro measures of accuracy. Looking at the individual activities, R+DT and DT are most accurate in 20 of the 23 classes (87%).

Overall there is moderate variance in the accuracy between R+DT and DT visualizations in macro scores (differences of 1.1 percentage points), and a relatively small variance for micro scores (0.1 percentage points). Macro F1 scores of RT, T and AI are 9.1, 9.2 and 9.8 percentage points worse than R+DT.

7.2 Comparison with Baselines

Table 3 shows the overall performance comparison between the best VLAIR variant (R+DT) and the baseline approaches. As we can see, RF with LOC+TIME representations outperforms all the other baselines, with a significant improvement in Micro- and Macro-F1 scores. This demonstrates that the inclusion of spatial and temporal information will improve activity recognition. However, VLAIR approaches still show the best accuracy. Figure 5 presents the detailed comparison of F1-scores on individual activities between the best VLAIR approach and the best performed baseline approaches on the LOC+TIME representations. The best accuracy overall corresponds to VLAIR R+DT ($F1_{micro} = 80\%, F1_{macro} = 58\%$), with a marginal improvement over RF LOC+TIME of 3 percentage points on Macro-F1, but much better than the other representations and techniques. Table 4 shows the results of one-sided paired Welch's t-tests (with $\alpha = 0.05$) comparing the F1-scores on each activity of the best VLAIR (R+DT) on the CNN against all the other techniques. This is supporting evidence that the observed results are not due to chance.

A look at the per-class result shows some additional interesting patterns in accuracy. First, for most activities, the best VLAIR approach (R+DT) is as good as or better than the baselines. The exception are some mid-frequency activities, where the CNN LOC+TIME

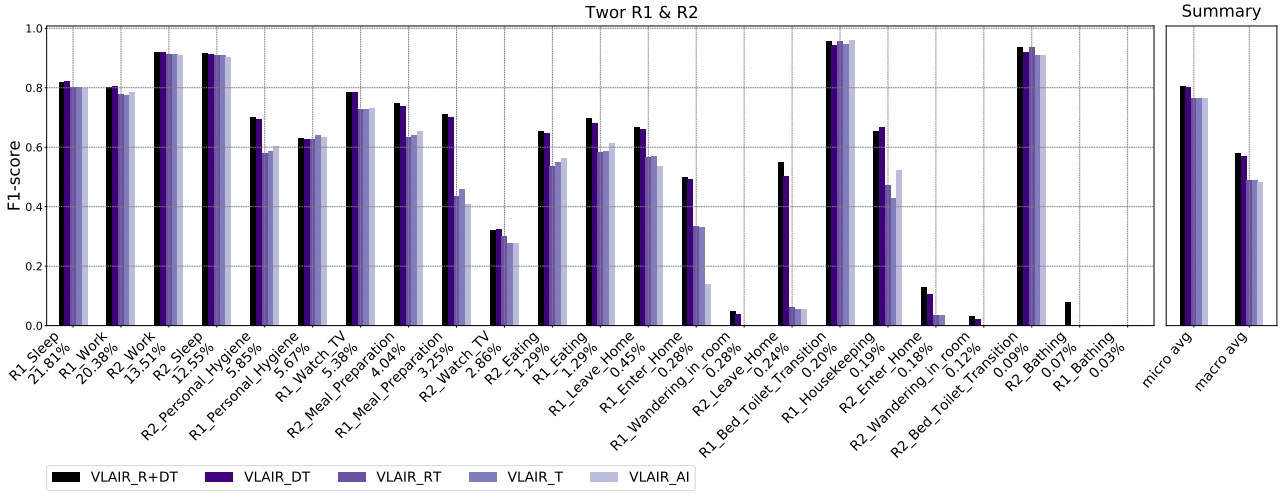


Figure 4: F1-scores on CNN trained with different VLAIR mappings. Activities are ordered from left to right in order of frequency (indicated below activity name).

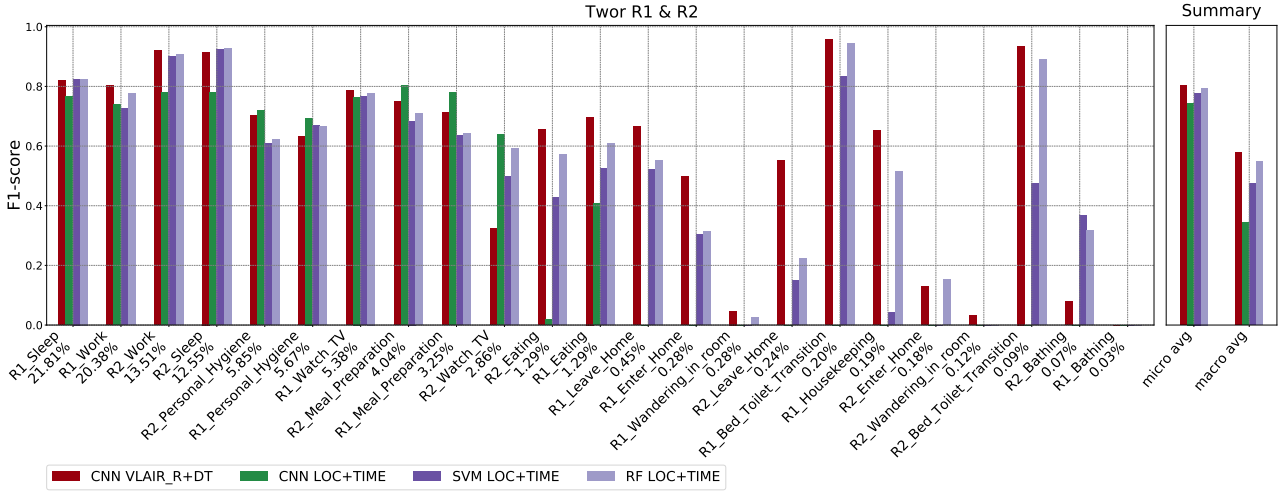


Figure 5: F1-scores for CNN on VLAIR R+DT images (red), CNN on LOC+TIME (green), SVM (violet) and RF (light violet) on LOC+TIME sensor features.

Table 4: Welch's t-test statistics and p-values comparing VLAIR R+DT and the baselines in F1-scores. * means statistically significant. All tests were run in R version 3.3.2 [22].

Baseline		t(23)	p-value
RF	LOC+TIME	1.2226	0.2344
	MI	5.7865	0.000008037*
	RAW	4.151	0.0004173*
CNN	LOC+TIME	3.238	0.003777*
	MI	8.4455	0.00000002373*
	RAW	4.3935	0.0002307*
SVM	LOC+TIME	2.5779	0.01717*
	MI	5.94	0.000005607*
	RAW	3.8026	0.000975*

offers better accuracy at the expense of completely failing to recognise most of the low-frequency activities. This reflects the advantage of CNN VLAIR R+DT when classifying less frequent activities; only one activity is not recognized (one with only 0.03% training instances). In comparison, RF and CNN LOC+TIME failed to recognise 2 and 11 activities respectively.

Second, the MI sensor data representations lead to the worst performance across all the baseline approaches. Introducing temporal and sensor mutual information encodes the temporal distances between sensor events and the co-occurrence of the events. According to Krishnan and Cook [27], this can help group and segment streaming sensors; however, this representation does not lead to

better recognition in our particular case, at least against the techniques that we have tested, including our own encoding of spatial and temporal information.

7.3 Comparison with State-of-the-art CNNs

We have chosen a simple CNN architecture to extract features on VLAIR images. The rationale behind this decision is that VLAIR images are rather simple and do not have texture or complex shapes. However, we also wanted to see whether pre-trained state-of-the-art deep CNN models would improve recognition accuracy. We considered several pre-trained network architectures to process our images. For comparison, we trained the final layers of the candidates with our richest visualizations (R+DT) with five different architectures: *VGG16* [44], *ResNet* [20**?**], *MobileNet* [23], *DenseNet* [24] and *InceptionV3* [48][1]. We fine-tune some of the convolutional layers of each CNN model with VLAIR R+DT images and keep the default hyperparameters from Keras. The highest micro- and macro-F1 scores are from VGG16, which is similar to our simple CNN. The other four models achieve lower accuracy than VGG16, in which the small sample size of VLAIR images might be insufficient to train these sophisticated CNN models. We are aware that, with thorough grid search on hyperparameters, these models might show some accuracy improvements. However, we decided to not pursue this further because we consider that the greatly increased computational and time cost would likely result only in marginal gains.

8 DISCUSSION

Below we discuss the possible advantages of using VLAIR for HAR and the limitations of our current design, which lead to future directions for improvement.

8.1 Recognising Infrequent Activities

The benefit of our proposed method comes mostly from the least frequent activities, which have the least training examples in the dataset. One might be inclined to discard such activities but we argue that the most unusual activities might often be the most important to support by a system, and that the usefulness of recognizing these activities is well above their rank in the frequency table. Furthermore, we see that infrequent activities are often not recognized at all by baseline algorithms, which directly affects the functionality of the system (there is no point to programming system reactions to those events if they are never recognized). As expected, there is variation in classification performance between the different visualization variants. Figure 4 provides some evidence that simplistic visualizations that do not include all information are inferior; e.g., AI. This suggests that visualization selection cannot be ignored.

8.2 Distinguishing Users and Activities in Common Areas

Our results indicate that VLAIR R+DT offers some advantage when it has to distinguish users on the same activities. For example, in Figure 6, 'R1_Leave_Home' and 'R2_Leave_Home' have only subtle

Figure 6: VLAIR images help distinguish different residents (R1 and R2) on the same activity

differences in sensor feature distribution, making it challenging for the baseline techniques to distinguish them accurately. The F1 scores in these two activities with RF LOC+TIME are 55.2% and 22.2% respectively, whereas with VLAIR R+DT accuracy is much better (66.6% and 55.0%). This is likely because the visual encodings of the VLAIR approaches highlight subtle differences between activities, such as the the movement around the front door area. Figure 6 hints at how colour (encoding time-of-day), the size of circle-nodes (activation time), and the thickness of the lines between nodes (sensor transitions) can express differences that allow the CNN to better distinguish entering from leaving home for both occupants.

Similarly, VLAIR R+DT can sometimes also better distinguish between activities that take place in the same location. For example in Figure 2, when a user is wandering vs. working in the bedroom, the same sensors might be activated, resulting in hard-to-separate sensor features and leading to low accuracy distinguishing these two activities [59] with traditional methods. VLAIR approaches show an advantage because they integrate the location of the sensor and the timing of its activation for the classification algorithm.

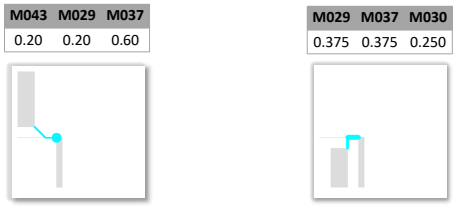
8.3 Enabling Better Activity Recognition via Temporal Information

Time-encoded trace images achieve the second best accuracy in more than half of the activities. Temporal information helps to distinguish activities with subtle differences in terms of sensor activation and spatial information. The examples of 'Leave Home' and 'Enter Home' almost activate the same set of sensors, but generally at very different times during a day, as encoded with different colors. This is because people usually go out for work in the morning and come back home in the evening.

The trace pattern is also useful to distinguish the same activity classes performed by different users. As shown in Figure 4, for 'R1_Meal_Preparation', the trace images achieve better accuracy than their activation intensity images, while for 'R2_Meal_Preparation' the activation intensity images lead to higher F1-scores than their counterpart trace images. This result suggests that these two users may have different cooking styles, which also can be seen in the example images of 'R1_Meal_Preparation' and 'R2_Meal_Preparation' in Figure 2. R1 moves more frequently and triggers more sensors in a more even way across the kitchen area from the refrigerator to the sink, while R2 stays at the hob area longer and activates fewer sensors in other areas.

Figure 7 also shows an interesting finding related to the 'Bed_Toilet_Transition' activity. Both VLAIR and baseline approaches

[1]For all these models, we use the architecture and pre-trained weights from the Keras library: https://keras.io/applications/.



(a) R1_Bed_Toilet_Transition **(b) R2_Bed_Toilet_Transition**

Figure 7: Examples of the 'Bed_Toilet_Transition' activity on R1 and R2: their sensor features and VLAIR images. 'Bed_Toilet_Transition' activates sensors ranging from bedroom to bathroom, which results in a rich spatial pattern visible in the encoded VLAIR images.

recognize 'Bed_Toilet_Transition' with good accuracy due to its distinctive patterns. The CNN trained from location-encoded images performs the best, which is consistent with the example pattern in Figure 7: 'Bed_Toilet_Transition' activates sensors ranging from bedroom to bathroom, which results in a rich spatial pattern visible in the converted VLAIR images. Also, R1's and R2's bedrooms have different spatial layouts which allow the CNN model to reliably separate 'R1_Bed_Toilet_Transition' and 'R2_Bed_Toilet_Transition'.

When comparing the baseline results in Figure 5 with different encoding results in Figure 4, we find that the CNN model trained on activity intensity images is similarly accurate to the baseline model trained on the sensor features. This implies that simply projecting sensor data onto a grid map is not an effective way to leverage spatial information to recognize complex, subtle activities. Our time-location encoded trace images have the advantage of characterizing fine differences between activity patterns.

8.4 Towards Explainability of Activity Recognition

VLAIR's intermediate visual representations provide an additional point of access to the intricacies of the data and the model, an important issue in current machine learning [52]. For example, designers of HAR deployments can use the understanding of their own visual system as a proxy to understand the classifier. This additional understanding might help designers to choose better locations and types of sensors to be deployed. In specific deployments, the images themselves can provide designers with valuable insights on, for example, why a certain activity is not being recognized. VLAIR might even enable deployment and mapping design for activities for which no data exists yet by allowing the designer to mentally picture in advance how patterns of activity might look when rendered through specific VLAIR mappings. In other words, VLAIR images might offer insight into the workings of the model.

8.5 Limitations and Future Work

As discussed above, our choice of mappings is not designed to make solid claims on the value of specific visual mappings. Moreover, we have only tested mappings that are spatialized with respect to the physical layout of the apartments; more abstract mappings could be easier to 'see' by the CNN (e.g.,[56]). Furthermore, we have identified cases in which our mappings fail to distinguish activities because they result in very similar images and shapes, and therefore mislead the CNN in the classification, or drive the the CNN to classify the minority case with a larger activity class.

It is important to highlight that our results provide a promising indication of the value of the VLAIR approach, using a specific dataset. Much work remains to be able to support generalizing the advantages of the VLAIR approach to other datasets and data types. Similarly, the exciting potential implications of visual intermediate representations to support debugging and understanding the models need to be validated with experiments with real humans in real deployments.

9 CONCLUSION

This paper presents VLAIR, a technique to re-represent data that we developed to improve classification performance with binary event-based human activity recognition data. VLAIR takes advantage of knowledge in data visualization as well of deep convolutional network architectures to deliver moderately improved F1 scores, and significantly improved macro F1 scores in activity classification. Results from our experiments support the advantages of this approach, which is particularly noticeable for infrequent activities. We contribute the VLAIR technique and its architecture, several simple instances of the VLAIR technique, and the results of the experiment showing improvements in classification performance.

In addition to the improvements in accuracy, VLAIR might contribute towards explainability. This is particularly important in pervasive health applications, where explaining the decisions being made is often crucial to gain the trust of patients and practitioners. For example, a VLAIR-enabled system that can recognise sleep activities to diagnose insomnia will allow patients and medical professionals to "see" how insomnia is being inferred by the system through example visual representations. This is much harder to do directly with raw sensor data.

REFERENCES

[1] Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. 2013. ARAS human activity datasets in multiple homes with multiple residents. In *the 7th International Conference on Pervasive Computing Technologies for Healthcare*. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 232–235.

[2] S. Aminikhanghahi, T. Wang, and D. J. Cook. 2019. Real-Time Change Point Detection with Application to Smart Home Time Series Data. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (2019), 1010–1023.

[3] CASAS TEAM at Washington State University. 2014. *CASAS Datasets: 7.Kyoto, 15.Milan, 17.Aruba.* http://casas.wsu.edu/datasets/

[4] Jacques Bertin and WJ BERG. 2010. *Semiology of graphics: Diagrams, networks, maps.* Redlands, Calif.: ESRI Press: Distributed by Ingram Publisher Services.

[5] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. 1999. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc.

[6] Guilin Chen, Aiguo Wang, Shenghui Zhao, Li Liu, and Chih-Yung Chang. 2018. Latent Feature Learning for Activity Recognition Using Simple Sensors in Smart Homes. *Multimedia Tools Appl.* 77, 12 (June 2018), 15201–15219.

[7] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. 2012. Sensor-Based Activity Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 42, 6 (Nov 2012), 790–808.

[8] W. S Cleveland and R. McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *J. Amer. Statist. Assoc.* 79, 387 (1984), 531–554.

[9] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. 2013. Transfer learning for activity recognition: a survey. *Knowledge and Information Systems* 36, 3 (01 Sep 2013), 537–556.

[10] D. Cook and M. Schmitter-Edgecombe. 2009. Assessing the quality of activities in a smart environment. *Methods of Information in Medicine* 48 (2009), 480–485. Issue 5.

[11] Diane J Cook. 2010. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems* 2010, 99 (2010), 1.

[12] Phil Corbishley and Esther Rodriguez-Villegas. 2008. Breathing detection: towards a miniaturized, wearable, battery-operated monitoring system. *IEEE Transactions on Biomedical Engineering* 55, 1 (2008), 196–204.

[13] Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, and Marco Ortolani. 2015. User activity recognition for energy saving in smart homes. *Pervasive and Mobile Computing* 16 (2015), 156–170.

[14] L. Fang, J. Ye, and S. Dobson. 2019. Discovery and Recognition of Emerging Human Activities Using a Hierarchical Mixture of Directional Statistical Models. *IEEE Transactions on Knowledge and Data Engineering* (2019), 1–1.

[15] Kyle D. Feuz and Diane J. Cook. 2015. Transfer Learning Across Feature-Rich Heterogeneous Feature Spaces via Feature-Space Remapping (FSR). *ACM Trans. Intell. Syst. Technol.* 6, 1, Article 3 (March 2015), 27 pages.

[16] Kyle D. Feuz and Diane J. Cook. 2017. Collegial Activity Learning Between Heterogeneous Sensors. *Knowl. Inf. Syst.* 53, 2 (Nov. 2017), 337–364.

[17] Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics* 36, 4 (1980), 193–202.

[18] Sojeong Ha, Jeong-Min Yun, and Seungjin Choi. 2015. Multi-modal convolutional neural networks for activity recognition. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, Hong Kong, 3017–3022.

[19] N Hammerla, S Halloran, and T Ploetz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. Newcastle University.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR 2016*. IEEE, 770–778.

[21] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *the SIGCHI conference on human factors in computing systems*. 203–212.

[22] Kurt Hornik. 2017. R. https://CRAN.R-project.org

[23] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv* abs/1704.04861 (2017).

[24] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on CVPR*. 2261–2269.

[25] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR* (2015).

[26] Wenchao Jiang and Zhaozheng Yin. 2015. Human activity recognition using wearable sensors by deep convolutional neural networks. In *ACM Multimedia 2015*. ACM, 1307–1310.

[27] Narayanan C. Krishnan and Diane J. Cook. 2014. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing* 10 (2014), 138 – 154.

[28] Xinyu Li, Yanyi Zhang, Ivan Marsic, Aleksandra Sarcevic, and Randall S Burd. 2016. Deep learning for rfid-based activity recognition. In *SenSys '16*. ACM, 164–175.

[29] Beiyu Lin, Yibo Huangfu, Nathan Lima, Bertram Jobson, Max Kirk, Patrick O'Keeffe, Shelley Pressley, Von Walden, Brian Lamb, and Diane Cook. 2017. Analyzing the relationship between human behavior and indoor air quality. *Journal of Sensor and Actuator Networks* 6, 3 (2017), 13.

[30] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen Intille. 2007. A Long-Term Evaluation of Sensing Modalities for Activity Recognition. In *UbiComp 2007*. 483–500.

[31] Jock Mackinlay. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.* 5, 2 (April 1986), 110–141.

[32] Lev Manovich. 2011. What is visualisation? *Visual Studies* 26, 1 (2011), 36–49.

[33] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep Convolutional Feature Transfer Across Mobile Activity Recognition Domains, Sensor Modalities and Locations. 92–99.

[34] Tamara Munzner. 2014. *Visualization analysis and design*. AK Peters/CRC Press.

[35] Gonzalo Gabriel Méndez, Miguel A. Nacenta, and Uta Hinrichs. 2018. Considering Agency and Data Granularity in the Design of Visualization Tools. 638:1–638:14.

[36] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler. 2015. Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning. In *ICMI '15*. 443–449.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[38] Bahareh Pourbabaee, Mehrsan Javan Roshtkhari, and Khashayar Khorasani. 2017. Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 99 (2017), 1–10.

[39] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D. Lane, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. 2018. Multimodal Deep Learning for Activity and Context Recognition. In *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, Vol. 1. 157:1–157:27.

[40] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang. 2016. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *BSN '16*. IEEE, 71–76.

[41] Michael S. Ryoo, Brandon Rothrock, Charles Fleming, and Hyun Jong Yang. 2017. Privacy-Preserving Human Activity Recognition from Extreme Low Resolution. In *AAAI*. Phoenix, Arizona USA.

[42] Andrea Rosales Sanabria, Thomas W. Kelsey, and Juan Ye. 2019. Representation Learning for Minority and Subtle Activities in a Smart Home Environment. In *PerCom '19*. Kyoto, Japan.

[43] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[44] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[45] Monit Shah Singh, Vinaychandran Pondenkandath, Bo Zhou, Paul Lukowicz, and Marcus Liwickit. 2017. Transforming sensor data to the image domain for deep learning—An application to footstep detection. In *IJCNN 2017*. IEEE, 2665–2672.

[46] Gina Sprint, Diane Cook, Douglas Weeks, Jordana Dahmen, and Alyssa La Fleur. 2017. Analyzing Sensor-Based Time Series Data to Track Changes in Physical Activity during Inpatient Rehabilitation. *Sensors* 17, 10 (2017).

[47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. (June 2016), 2818–2826.

[49] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *International conference on pervasive computing*. Springer, 158–175.

[50] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate activity recognition in a home setting. In *the 10th international conference on Ubiquitous computing*. ACM, 1–9.

[51] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. 2011. *Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software*. Atlantis Press, Paris, 165–186.

[52] Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. 2012. Making machine learning models interpretable. In *ESANN*, Vol. 12. 163–172.

[53] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2018. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* 119 (2018), 3–11.

[54] Jie Wang, Xiao Zhang, Qinhua Gao, Hao Yue, and Hongyu Wang. 2017. Device-free wireless localization and activity recognition: A deep learning approach. *IEEE Transactions on Vehicular Technology* 66, 7 (2017), 6258–6267.

[55] Colin Ware. 2012. *Information visualization: perception for design*. Morgan Kaufman.

[56] K. Wongsuphasawat and D. Gotz. 2012. Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2659–2668.

[57] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 649–658.

[58] Juan Ye, Simon Dobson, and Susan McKeever. 2012. Situation Identification Techniques in Pervasive Computing: a review. *Pervasive and mobile computing* 8 (2012), 36–66. Issue 1.

[59] Juan Ye, Graeme Stevenson, and Simon Dobson. 2015. KCAR: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing* 19 (2015), 47–70.

[60] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. Convolutional neural networks for human activity recognition using mobile sensors. In *MobiCASE 2014*. IEEE, 197–205.

[61] Zhongna Zhou, Xi Chen, Yu-Chia Chung, Zhihai He, Tony X Han, and James M Keller. 2008. Activity analysis, summarization, and visualization for indoor human activity monitoring. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 11 (2008), 1489–1498.