# Inertial-Vision: cross-domain knowledge transfer for wearable sensors

Girmaw Abebe[1,2] and Andrea Cavallaro[1]

[1]Centre for Intelligent Sensing, Queen Mary University of London
[2]Technical Research Centre for Dependency Care and Autonomous Living, UPC-BarcelonaTech

[1]{g.abebe, a.cavallaro}@qmul.ac.uk
[2]girmaw.abebe@upc.edu

## Abstract

*Multi-modal ego-centric data from inertial measurement units (IMU) and first-person videos (FPV) can be effectively fused to recognise proprioceptive activities. Existing IMU-based approaches mostly employ cascades of handcrafted triaxial motion features or deep frameworks trained on limited data. FPV approaches generally encode scene dynamics with motion and pooled appearance features. In this paper, we propose a multi-modal ego-centric proprioceptive activity recognition that uses a convolutional neural network (CNN) followed by a long short-term memory (LSTM) network, transfer learning and a merit-based fusion of IMU and/or FPV streams. The CNN encodes short-term temporal dynamics of the ego-motion and the LSTM exploits the long-term temporal dependency among activities. The merit of a stream is evaluated with a sparsity measure of its initial classification output. We validate the proposed framework on multiple visual and inertial datasets.*

## 1. Introduction

First-person proprioceptive activity recognition classifies the activities of a subject from ego-centric data and may play a significant role in personalised assistive technologies [33]. Proprioceptive activities involve full- or upper-body motion of the subject such as *Run*, *Walk* and *Go upstairs* [2]. Inertial measurement units (IMU) and wearable cameras are common sensors used to collect ego-centric data. An IMU itself may contain multiple sensors, such as an accelerometer and a gyroscope. Current approaches often apply feature-level fusion using concatenation [26], and it is desirable to effectively integrate different feature streams and/or modalities.

Motivated by the success of deep learning in computer vision, convolutional neural networks (CNNs) have been employed also with inertial data [24, 25] and recursive neural networks (RNNs), such as long short-term memory (LSTM) networks, have been used with multi-modal data as well [22]. However, deep frameworks for activity recognition from time-series sensory data are often built form scratch and trained with a limited amount of data [22, 24, 25]. In addition, they do not effectively encode the intrinsic relationships among triaxial components of the inertial data [24, 25]. Though IMU and first-person vision (FPV) modalities are complementary [33], their deep features have not been integrated yet.

In this paper, we present a deep framework for proprioceptive activity recognition that uses inertial data and first-person videos. We use *cross-domain knowledge transfer* with a CNN-LSTM that exploits the discriminative characteristics of multi-modal feature groups provided by stacked spectrograms from the inertial data. Our solution enables us (i) to use 2D convolutions rather than 3D convolutions; (ii) to use existing image models as feature extractors; and (iii) to encode the intrinsic relationships among motion components. To reduce the complexity of the LSTM network and hence the amount of data required for its training, we integrate information from different streams and/or modalities using a logistic regression (LR) and a Hoyer-based sparsity measure [13]. To the best of our knowledge, this is the first work that integrates deep features extracted from inertial and visual data for the recognition of proprioceptive activities. The software of the proposed framework is available at `http://www.eecs.qmul.ac.uk/~andrea/fpv-imu.html`.

The paper is organized as follows. Section 2 reviews related works that employ deep frameworks on inertial and FPV data. Section 3 presents the proposed framework. Section 4 describes the experimental results and the datasets used for validation. Finally, conclusions are drawn in Sec. 5.

## 2. Related work

In this section, we review CNNs that learn motion features from FPV with 3D and 2D convolutions. We also discuss LSTM-based temporal dependency encoding as well as CNN and LSTM networks for inertial-based activity recognition.

Features that encode the temporal dynamics in a video can be learned with a CNN that uses 3D convolutions [15, 23, 30] or 2D convolutions followed by temporal pooling [10, 16, 18, 27, 28, 31, 32]. The 3D convolutions help learning spatio-temporal [15, 30] or temporal [23] features from a volume of data and result in a large number of network parameters. 2D convolutions can instead be applied on each frame and be followed by a pooling operation to encode the temporal variation of each feature [16, 27, 32].

FPV-specific deep frameworks are mostly designed for the recognition of object-interactive activities [20, 28] and focus on learning local hand-motions and objects using multi-stream networks. A compact CNN was proposed in [23] to learn ego-centric motion features using a 3D convolution in the first layer followed by subsequent 2D convolution layers, which suppress long-term temporal information early in the network.

Rearranging optical flow data into RGB-like images enables the use of 2D convolutions followed by temporal pooling [18, 31, 32]. This solution reduces the amount of data required for training as it facilitates transfer learning from successful image models pre-trained on large image datasets, e.g. ImageNet [7].

LSTM networks can encode temporal dependencies among subsequent samples. When an LSTM is preceded by a CNN, the overall network becomes both spatially and temporally deep [9, 21, 32]. For this reason existing LSTM networks therefore generally encode short-term dynamics only (e.g. 0.64 seconds [9]).

Due to the success of deep networks in computer vision [8], convolutional and recursive networks have also been used for time-series inertial data [11, 22, 24, 25]. However, deep features learned from inertial data do not outperform handcrafted (shallow) features yet [22, 25], partly because of the lack of a large public dataset for training.

The sums of temporal convolutions on the concatenated spectrograms of multiple axes and streams can be applied to learn inertial features on low-power devices [24, 25]. While this approach achieves invariance against changes in placement, orientation and sampling rate of the inertial sensor; cascading spectrograms limits the potential of learning useful relationships among different motion components [22]. A CNN-LSTM framework can be used to learn features from raw inertial data with the LSTM accounting for the temporal dependency [22]. However, this approach results in more complex network compared to [24, 25].

## 3. Proposed framework

Let $\mathcal{C} = \{A_c\}_{c=1}^{C}$ be a set of $C$ activity classes. Let $I_n \in \{I_{a,n}, I_{g,n}\}$ be a windowed inertial sample of accelerometer ($I_{a,n}$) or gyroscope ($I_{g,n}$) data; and $V_n \in \{V_{g,n}, V_{c,n}\}$ be a global motion stream extracted from a first-person video sample, $\mathbf{F}_n$, using the average of grid optical flow ($V_{g,n}$) or the movement of intensity centroid ($V_{c,n}$). Let $S_n \subseteq \{I_n, V_n\}$ be multi-modal ego-centric motion data whose duration is $\lambda \in \{\lambda_i, \lambda_v\}$ seconds, where $\lambda_i$ and $\lambda_v$ refer to the inertial and visual data, respectively. We aim to classify each $S_n$ into its activity class, $A_c^n$, by encoding its short-term dynamics and long-term temporal dependencies with the preceding $T \in \{T_i, T_v\}$ samples: $S_{n-1}, S_{n-2}, \cdots, S_{n-T}$. $T_i$ and $T_v$ refer to the inertial and visual data, respectively.

Similarly to [2], we extract short-term motion features using a pre-trained CNN model from stacked spectrograms of multiple motion components. Spectrograms for each motion stream are computed and stacked as a 3-channel motion representation. We employ a logistic classifier on each stream and use a sparsity weighted combination of outputs from different streams. Finally, we employ an LSTM framework to encode the long-term temporal dependency among activities. An output wrapper transforms the hidden output of the LSTM to an activity prediction vector. The proposed solution is shown in Fig. 1 and detailed below.

### 3.1. Multi-stream global motion extraction

We extract from the first-person video sample $\mathbf{F}_n = (f_{n,1}, f_{n,2}, \cdots, f_{n,l}, \cdots f_{n,L})$, which contains $L$ frames, two streams of global motion features, namely the average of the grid optical flow, $V_{g,n}$, and the movement of intensity centroid, $V_{c,n}$.

Let $O_n = (O_{n,1}, O_{n,2}, \cdots, O_{n,l}, \cdots, O_{n,L-1})$ be the optical flow computed between a subsequent pair of frames, $f_{n,l}$ and $f_{n,l+1}$, $l \in [1, L-1]$, with the Horn-Schunk method [12], whose global smoothness assumption fits proprioceptive activities where the ego-motion of the user is dominant. Let $O_{n,l} = \{O_{n,l}^x(i) + jO_{n,l}^y(i)\}_{i=1}^{\gamma^2}$ represent the set of complex optical flow vectors of frame $l$, where $\gamma$ is the number of grid cells in the horizontal, $x$, and vertical, $y$, dimensions. The corresponding mean optical flow components, $V_{g,n,l}^x$ and $V_{g,n,l}^x$, are computed as[1]

$$V_{g,l}^x = \frac{1}{\gamma^2}\sum_{i=1}^{\gamma^2} O_l^x(i) \quad \text{and} \quad V_{g,l}^y = \frac{1}{\gamma^2}\sum_{i=1}^{\gamma^2} O_l^y(i). \quad (1)$$

The final global motion representation from the optical flow data becomes $V_g = (V_{g,l})_{l=1}^{L-1}$, where $V_{g,l} = [V_{g,l}^x, V_{g,l}^y]$.

We extract the centroid stream, $V_c$, from $\mathbf{F}_n$ as follows. Let $H$ and $W$ be the height and the width in pixels of frame

[1] For simplicity we drop the subscript $n$.

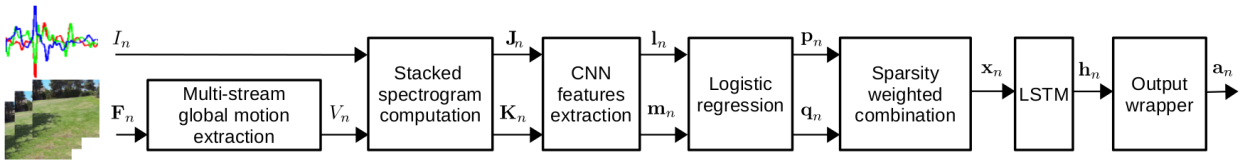



Figure 1: The proposed method for proprioceptive activity recognition from multi-modal ego-centric data (inertial and first-person vision data). Global motion is encoded from the mean of grid optical flow and the derivative of the intensity centroid. A set of spectrograms is derived from each motion stream. The spectrogram values are scaled, translated, normalized and stacked. Then CNN features are given as input to a logistic classifier. The classification outputs of each modality are weighted based on their sparseness and combined prior to the LSTM, which encodes the temporal dependency among activities. Finally, an output wrapper with softmax normalization produces the activity prediction vector.

$f_l$. Let $\mathcal{M}^l_{pq}, p, q \in \{0, 1\}$, be the first-order image moment of $f_l$, calculated as the weighted average of its intensity values as

$$\mathcal{M}^l_{pq} = \sum_{i=1}^{H} \sum_{j=1}^{W} i^p j^q f_l(i, j). \quad (2)$$

Similarly to [1, 2, 3], we compute the displacement of the intensity centroid, $V_{c,l} = \{V^x_{c,l}, V^y_{c,l}\}$, from the first-order derivative of subsequent centroids as $V^x_{c,l} = C^x_{l+1} - C^x_l$ and $V^y_{c,l} = C^y_{l+1} - C^y_l$, where $(C^x_l, C^y_l)$ is the centroid location at $l \in [1, L-1]$, $C^x_l = \mathcal{M}^l_{01}/\mathcal{M}^l_{00}$ and $C^y_l = \mathcal{M}^l_{10}/\mathcal{M}^l_{00}$. Finally, the global motion representation from the displacement of the intensity centroid becomes $V_c = (V_{c,l})^{L-1}_{l=1}$, where $V_{c,l} = [V^x_{c,l}, V^y_{c,l}]$.

## 3.2. Stacked spectrogram computation

We employ a time-frequency representation (spectrogram) to encode the dynamics for each axis of a motion stream in $S_n$. The stacking arrangement enables us to encode intrinsic relationships among different axial motion components (Fig. 2). This reduces the effect of different mounting positions of the wearable sensors.

As $I_n$ is often triaxial, $(x, y, z)$, whereas $V_n$ has two components, $(x, y)$, we describe the different stacking arrangements for inertial and visual spectrograms below.

### 3.2.1 Stacked spectrogram from inertial data

The fast Fourier transform (FFT), $\mathcal{F}(\cdot)$, is computed on each component of the inertial data, $I_n = (I^x_n, I^y_n, I^z_n)$, to generate the magnitude of a set of spectrograms, $\bar{I}_n = \mathcal{F}(I_n) = (\bar{I}^x_n, \bar{I}^y_n, \bar{I}^z_n)$. Similarly to [9], we scale each spectrogram component of $\bar{I}_n$ by $\alpha$, translate it by $\tau$ and apply normalization to [0,255] as[2]

$$J'_n = \alpha * \bar{I}_n + \tau \quad (3)$$
$$J''_n = \max(J'_n, 0) \quad (4)$$
$$\bar{J}_n = \min(J''_n, 255). \quad (5)$$

[2]For simplicity, the $x, y$ and $z$ superscripts will be dropped.

In order to encode high-level CNN features from the spectrograms with 2D convolutions, we stack the normalized spectrograms in $\bar{J}_n$ into a 3-channel motion representation as $\mathbf{J}_n = (\bar{J}^x_n, \bar{J}^y_n, \bar{J}^z_n) \in \{\mathbf{J}_{a,n}, \mathbf{J}_{g,n}\}$.

The stacked spectrogram representation of the inertial data enables us to achieve cross-domain knowledge transfer using pre-trained image models. This avoids the need of training a dedicated deep network from scratch.

### 3.2.2 Stacked spectrogram from FPV data

The stacked spectrogram of the motion stream from FPV, $V_n \in \{V_{g,n}, V_{a,n}\}$, is obtained by applying $\mathcal{F}(\cdot)$ on $V^x_n$ and on $V^y_n$. To introduce additional discriminative characteristics, we extend $V_n$ by adding the direction component, $V^\theta_n = \arctan(V^y_n/V^x_n)$, as a third channel to the stack.

Similarly to the inertial spectrograms, we then scale, translate and normalize[3] $\bar{V}_n = \mathcal{F}(\bar{V}^x_n, \bar{V}^y_n, \bar{V}^\theta_n)$ as

$$K'_n = \alpha * \bar{V}_n + \tau \quad (6)$$
$$K''_n = \max(K'_n, 0) \quad (7)$$
$$\bar{K}_n = \min(K''_n, 255). \quad (8)$$

The spectrograms of the $x, y$ and $\theta$ components are stacked to obtain a 3-channel motion representation, $\mathbf{K}_n = (\bar{K}^x_n, \bar{K}^y_n, \bar{K}^\theta_n) \in \{\mathbf{K}_{g,n}, \mathbf{K}_{c,n}\}$.

The stacked spectrogram representation of the visual data enables us to obtain high-level global motion features, $\mathbf{m}_n$, using 2D convolutions only. This is particularly useful in FPV, whose datasets are smaller than traditional vision datasets, e.g. Sports-1M [16].

## 3.3. CNN features extraction

We store $\mathbf{J}_n$ and $\mathbf{K}_n$ as JPEG images and, similarly to [2], we employ a CNN framework to extract high-level temporal features, namely $\mathbf{l}_n \in \{\mathbf{l}_{a,n}, \mathbf{l}_{g,n}\}$ from $\mathbf{J}_n$ and

[3]The normalization enables us to transfer knowledge from image datasets, e.g. ImageNet [7].

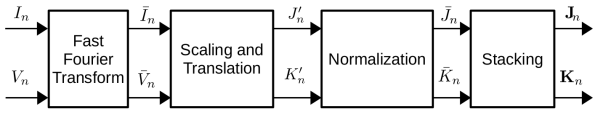



Figure 2: Stacking of the spectrograms from inertial, $I_n$, and visual, $V_n$, motion components. The fast Fourier transform is applied to obtain a time-frequency representation followed by scaling, translation and normalization that bound the spectrogram values to $[0, 255]$. Stacking the spectrograms produces a 3-channel representation that enables transfer learning from image-based models.

$\mathbf{m}_n \in \{\mathbf{m}_{g,n}, \mathbf{m}_{c,n}\}$ from $\mathbf{K}_n$. To extract the CNN features, we use GoogleNet [29] that was effectively employed across a range of computer vision problems [8].

### 3.4. Logistic regression

Each feature stream in $\mathbf{l}_n$ and $\mathbf{m}_n$ is separately validated using a logistic regression to obtain independent classification outputs, $\mathbf{p}_n \in \{\mathbf{p}_{a,n}, \mathbf{p}_{g,n}\}$ and $\mathbf{q}_n \in \{\mathbf{q}_{g,n}, \mathbf{q}_{c,n}\}$, respectively. The outputs are then weighted by their corresponding discriminative characteristics.

The logistic classification also transforms high-dimensional features, $\mathbf{l}_n$ and $\mathbf{m}_n \in \mathbb{R}^D$ (where $D$ is the feature dimension), to $\mathbf{p}_n$ and $\mathbf{q}_n \in \mathbb{R}^C$, with $C << D$. This reduces the complexity of the LSTM network to encode the long-term temporal dependency among activities and therefore the amount of training data required.

### 3.5. Sparsity weighted combination

To evaluate the decision confidence of each motion stream we employ a sparsity measure. First, we apply a sigmoid function, $\sigma(\cdot)$, to transform the logistic outputs, $\mathbf{p}_n$ and $\mathbf{q}_n$, respectively, to $\mathbf{r}_n$ and $\mathbf{s}_n$, which are bounded to $(0, 1)$, as

$$\sigma(\boldsymbol{\xi}) = \frac{1}{1 + exp(-\boldsymbol{\xi})}, \tag{9}$$

where $\boldsymbol{\xi} \in \{\mathbf{p}_n, \mathbf{q}_n\}$. In order to compute the sparseness of the logistic classification output, we apply the Hoyer measure [13], $\psi(\cdot)$, which is effective for a fixed dimensional vector [14] and is defined as

$$\psi(\boldsymbol{\eta}) = \frac{\sqrt{C} - \frac{||\boldsymbol{\eta}||_1}{||\boldsymbol{\eta}||_2}}{\sqrt{C} - 1}, \tag{10}$$

where $\boldsymbol{\eta} \in \{\mathbf{r}_n, \mathbf{s}_n\}$, $||\cdot||_1$ and $||\cdot||_2$ are $\ell_1$ and $\ell_2$ norms, respectively. The final feature input to the LSTM network, $\mathbf{x}_n \in \mathbb{R}^C$, is the accumulation of the logistic classification vectors of the motion streams, weighted by their corresponding sparseness measure as

$$\mathbf{x}_n = \sum_{\boldsymbol{\eta} \in \{\mathbf{r}_n, \mathbf{s}_n\}} \boldsymbol{\eta}\psi(\boldsymbol{\eta}). \tag{11}$$

### 3.6. Long short-term memory (LSTM) network

We apply an LSTM framework to encode the long-term temporal relationships among activities and to overcome the vanishing and exploding gradient problems of a vanilla RNNs. LSTM networks use three additional gates (forget, input and output) that act as switches for monitoring the information flow from the current input, $\mathbf{x}_n$, and previous hidden state, $\mathbf{h}_{n-1}$, to the current hidden state, $\mathbf{h}_n$, via the memory cell, $\mathbf{c}_n$.

The forget gate, $\mathbf{f}_n$, helps to discard less useful information from the previous cell state, $\mathbf{c}_{n-1}$, as

$$\mathbf{f}_n = \sigma(W_{xf}\mathbf{x}_n + W_{hf}\mathbf{h}_{n-1} + \mathbf{b}_f), \tag{12}$$

where $\mathbf{b}_f$ is the bias in the forget gate.

The input gate, $\mathbf{i}_n$, weights the candidate cell information, $\bar{\mathbf{c}}_n$, to be the current state of the cell, $\mathbf{c}_n$, as

$$\mathbf{i}_n = \sigma(W_{xi}\mathbf{x}_n + W_{hi}\mathbf{h}_{n-1} + \mathbf{b}_i), \tag{13}$$
$$\bar{\mathbf{c}}_n = \phi(W_{xc}\mathbf{x}_n + W_{hc}\mathbf{h}_{n-1} + \mathbf{b}_c), \tag{14}$$
$$\mathbf{c}_n = \mathbf{f}_n \odot \mathbf{c}_{n-1} + \mathbf{i}_n \odot \bar{\mathbf{c}}_n, \tag{15}$$

where $\phi(\cdot)$ represents the *tanh* activation function, $\odot$ is an element-wise product, $\mathbf{b}_i$ and $\mathbf{b}_c$ represent the input gate and memory cell biases, respectively.

The output gate, $\mathbf{o}_n$, evaluates the cell information, $\mathbf{c}_n$, to predict $\mathbf{h}_n$ as

$$\mathbf{o}_n = \sigma(W_{xo}\mathbf{x}_n + W_{ho}\mathbf{h}_{n-1} + \mathbf{b}_o), \tag{16}$$
$$\mathbf{h}_n = \mathbf{o}_n \odot \phi(\mathbf{c}_n), \tag{17}$$

where $\mathbf{b}_o$ represents the output gate bias.

The weight parameters $W_{hf}$, $W_{hi}$, $W_{hc}$ and $W_{ho} \in \mathbb{R}^{\nu \times \nu}$ describe the relationship between the previous hidden state, $\mathbf{h}_{n-1}$, and the remaining states, $\mathbf{f}_n$, $\mathbf{i}_n$, $\mathbf{c}_n$ and $\mathbf{o}_n \in \mathbb{R}^{\nu}$, respectively, where $\nu$ represents the number of neurons used in each of the states. The parameters $W_{xf}$, $W_{xi}$, $W_{xc}$ and $W_{xo} \in \mathbb{R}^{\nu \times C}$ describe the relationship between the sparsity weighted input of the LSTM, $\mathbf{x}_n \in \mathbb{R}^C$, and the remaining states.

### 3.7. Output projection wrapper

We finally apply an output projection wrapper on the estimated hidden state, $\mathbf{h}_n$, and generate the activity prediction vector, $\mathbf{a}_n \in \mathbb{R}^C$, for $\mathbf{S}_n$ using the softmax normalization:

$$\mathbf{a}_n = \frac{exp(W_{ha}\mathbf{h}_n)}{\sum_{c=1}^{C} exp(W_{ha}\mathbf{h}_n)}, \tag{18}$$

where $W_{ha} \in \mathbb{R}^{C \times \nu}$ is the wrapping matrix.

The class with the maximum score in $\mathbf{a}_n$ is the winning class, $A_c^n$.

## 4. Experiments

In this section, we present the validation datasets, describe the setting of inertial and visual parameters, and compare the proposed framework with state-of-the-art methods.

### 4.1. Datasets and methods under comparison

We use multiple inertial and visual datasets for the validation (see Table 1). The inertial datasets are ActiveMiles [24, 25] and WISDM-v2.0 [17, 19].

**ActiveMiles** [25] is one of the largest public inertial datasets with 30 hours (h) of labelled accelerometer and gyroscope data ($4,390,726$ samples) with different sampling rates (50-200 Hz) and collected using smartphones. It contains seven activities: *Casual Movement, Cycling, No Activity (Idle), Public Transport, Running, Standing* and *Walking*. Ten subjects participated in its collection.

**WISDM-v2.0** [17] consists of accelerometer data ($\approx$ 41.4 h) collected in uncontrolled environments. The dataset contains $2,980,765$ samples at 20 Hz, and six activities: *Walking, Jogging, Stairs, Sitting, Standing* and *Lying Down*. 563 subjects participated in its collection.

The FPV datasets are HUJI [23] and BAR [3].

**HUJI** [23] is the largest public dataset for FPV activity recognition and was collected with a head-mounted camera. We utilise a 15-h subset that contains the following activities: *Go upstairs, Run, Walk, Sit/Stand* and *Static*. Approximately 50% of the subset dataset (17 out of 44 video sequences) are collected from YouTube videos.

**BAR** [3] is the first dataset of basketball activities from FPV (collected with a chest-mounted camera at 30 fps) and is composed of 11 activities: *Bow, Sit-Stand, Left-right turn, Walk, Jog, Run, Sprint, Pivot, Shoot, Dribble* and *Defend*. Four subjects participated in its collection and accelerometer data was also collected for three subjects using a back-mounted inertial unit at 200 Hz.

To evaluate the recognition performance we use accuracy, $\mathcal{A}$, precision, $\mathcal{P}$, and recall, $\mathcal{R}$:

$$\mathcal{A} = 100\frac{tp+tn}{tp+tn+fp+fn}, \quad (19)$$

$$\mathcal{P} = 100\frac{tp}{tp+fp}, \quad (20)$$

$$\mathcal{R} = 100\frac{tp}{tp+fn}, \quad (21)$$

where $tp$ is the number of true positives, $fp$ is the number of false positives, $tn$ is the number of true negatives and $fn$ is the number of false negatives. $\mathcal{A}$, $\mathcal{P}$ and $\mathcal{R}$ are first evaluated per each activity and then the class average is reported as the performance of a system.

We consider six inertial-based approaches, namely Handcrafted-1 [25], Handcrafted-2 [5], Catal *et al*. [6], Alsheikh *et al*. [4], Ravi *et al*. [24] and Ravi *et al*. [25].

Table 1: Summary of the datasets used for validation (Acc.: accelerometer; Gyro.: gyroscope; FPV: first-person vision; ✓: availability of a specific modality; NS: not specified; #: number; h: hour).

| | Modalities | | | | | |
|---|---|---|---|---|---|---|
| | Inertial | | Visual | | | |
| Dataset | Acc. | Gyro. | FPV | Activities (#) | Subjects (#) | Duration (h) |
| ActiveMiles [25] | ✓ | ✓ | | 7 | 10 | 30 |
| WISDM-v2.0 [17] | ✓ | | | 6 | 530 | 41.4 |
| HUJI [23] | | | ✓ | 5 | NS | 15 |
| BAR [3] | ✓ | | ✓ | 11 | 3 | 1.2 |

Handcrafted-1 [25] and Catal *et al*. [6] use low-dimensional shallow features extracted in the time domain, whereas Handcrafted-2 [5] includes also frequency-domain features. Alsheikh *et al*. [4] and Ravi *et al*. [24] employed learned deep features using dedicated networks. Ravi *et al*. [25] integrated the deep features in [24] with Handcrafted-1 [25] features.

### 4.2. Inertial parameters

We set the parameters similarly to the state-of-the-art methods [5, 24, 25]. The window length for the inertial data is $\lambda_i = 10$ s, with no overlap. The dimension of the shallow features of Alsheikh *et al*. [4], Handcrafted-1 [25] and Handcrafted-2 [5] are 43, 102 and 394, respectively. We set the scaling factor $\alpha = 16$ and the translation $\tau = 128$ on the spectrograms as in [9].

We extract the features from the next-to-last layer of the inception-v3, i.e. 'pool_3 : 0', which provides a feature of $D = 2,048$. In order to compare the inception features with the state-of-the-art methods in ten fold validation as in [24, 25], we employ a support-vector machine (SVM) classifier with a polynomial kernel implemented in MATLAB 2014b[4]. We use the results reported in [25] for the comparison.

Equal amount of data is preserved for training and testing (50% each) in ActiveMiles and WISDM-v2.0. We use fixed train and test sets to reduce the number of iterations that also increases with the number of epochs in the LSTM. Each experiment is repeated ten times and the average performance is reported. We use a one-vs-remaining (OVR) validation for the logistic regression. Due to the limited dataset size, the LSTM network has only one hidden layer, which contains $\nu = 128$ neurons, and is trained with a batch size of 10, whereas the number of epochs is 100. We set $T_i = 10$ samples and the learning rate to be 0.01.

### 4.3. Visual parameters

We set the length of an activity sample for the visual component as $\lambda_v = 3$ s, i.e. $L = 90$ frames for 30 fps,

[4]The full pipeline, which contains the logistic regression, the sparsity weighted combination and the LSTM, is instead implemented in Python 3.5.

Table 2: Comparison of the Accuracy, $\mathcal{A}(\%)$, of state-of-the-art approaches and the proposed inception features in the inertial datasets. An SVM is employed with a one-vs-remaining strategy in a ten-fold validation as in [24, 25]. ('Prop. Inception': concatenation of inception features from the accelerometer and gyroscope data in ActiveMiles and only the inception features from the accelerometer data in WISDM-v2.0; 'Prop. Inception+Handcrafted-2': concatenation of 'Prop. Inception' and Handcrafted-2 features.)

| | ActiveMiles [25] | WISDM-v2.0 [17] |
|---|---|---|
| Handcrafted-1 [25] | 95.0 | 92.5 |
| Handcrafted-2 [5] | 98.1 | 97.6 |
| Ctal *et al.* [6] | 91.7 | 89.8 |
| Alsheikh *et al.* [4] | 84.5 | 82.5 |
| Ravi *et al.* [24] | 95.1 | 88.5 |
| Ravi *et al.* [25] | 95.7 | 92.7 |
| Prop. Inception | **98.8** | 97.3 |
| Prop. Inception+Handcrafted-2 | 98.4 | **97.9** |

with $50\%$ overlap. We resize the videos to a resolution of $320 \times 240$ and set the number of grid cells to $\gamma = 100$.

The FFT, scaling, translation and normalization of the spectrograms as well as the inception feature extraction are performed similarly to what discussed above for the inertial spectrograms. In the HUJI dataset, we employ a $50\%$ decomposition for train and test sets. In the BAR dataset, the sequences from two subjects are used for training, while the remaining are used for testing. The LSTM has $T_v = 20$ samples for the HUJI dataset to compensate for the shorter window length compared to the inertial datasets. For the BAR dataset, we set $T_v = T_i = 5$ since the dataset is small and there are no significantly long temporal dependencies among samples. All other parameters of the pipeline are the same as those of the inertial pipeline.

### 4.4. Discussion

Table 2 and 3 compare the performance of the inception features with that of state-of-the-art methods validated on the inertial datasets, without employing the sparsity weighting and the LSTM-based temporal encoding.

Table 2 shows that the overall accuracy, $\mathcal{A}$, of the proposed inception features outperforms existing inertial-based deep frameworks [4, 24, 25]. Unlike [25], the inception features improve the performance without the concatenation of the shallow features. In addition, Handcrafted-1 [25] is outperformed by Handcrafted-2 [5], which additionally consists of frequency-domain features.

Table 3 provides per-class recall values, $\mathcal{R}$, between the baseline deep framework [25] and the proposed CNN features, extracted from a pre-trained inception-v3 model. The proposed features achieve equivalent performance with the baseline containing both deep and shallow features [25]. Particularly, the concatenation of the inception features from accelerometer and gyroscope data in ActiveMiles improved the performance of all the activities. The equivalent

Table 3: Comparison of the Recall, $\mathcal{R}(\%)$, of inception features and the baseline [25] ('Prop. Acc.': inception features from accelerometer data; 'Prop. Gyro.': inception features from gyroscope data; 'Prop. Acc.+Gyro.': concatenation of the inception features from the accelerometer and the gyroscope data).

| | ActiveMiles [25] | | | | | | |
|---|---|---|---|---|---|---|---|
| | Casual | Cycling | Idle | Transport | Running | Standing | Walking |
| Ravi *et al.* [25] | 96.1 | **96.6** | 96.5 | 95.2 | 98.8 | **73.0** | **96.5** |
| Prop. Acc. | 88.7 | 94.4 | 96.7 | 94.7 | 98.8 | 46.7 | 94. 8 |
| Prop. Gyro. | 92.3 | 90.7 | 80.6 | 89.8 | 97.5 | 15.8 | 91.9 |
| Prop. Acc.+Gyro. | **98.2** | 94.5 | **97.1** | **96.8** | **99.4** | 54.2 | 95.8 |
| | WISDM-v2.0 [17] | | | | | | |
| | Walking | Jogging | Stairs | Sitting | Standing | Lying | |
| Ravi *et al.* [25] | **97.2** | **97.7** | **77.0** | 89.3 | **82.1** | 85.8 | |
| Prop. Acc. | 96.1 | 97.1 | 66.6 | **89.6** | 80.1 | **88.5** | |

Table 4: Comparison of different fusion strategies on the inertial datasets. ('-': not available; C-LSTM: concatenation of feature groups followed by LSTM only; C-LR-LSTM: concatenation of feature groups followed by logistic regression and LSTM; LR-C-LSTM: concatenation of LR ouputs of the feature groups prior to the LSTM; LR-S-LSTM: accumulation of LR ouputs of the feature groups prior to the LSTM).

| | | ActiveMiles [25] | | WISDM-v2.0 [17] | |
|---|---|---|---|---|---|
| | | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ |
| Individual | Inception-Acc. | 41.6 | 33.0 | 65.6 | 58.0 |
| | Inception-Gyro. | 40.2 | 29.9 | - | - |
| | Handcraft-Acc. [5] | 42.1 | 35.9 | 65.3 | 56.0 |
| | Handcraft-Gyro. [5] | 44.5 | 37.2 | - | - |
| Fusion | C-LSTM | 54.0 | 43.6 | 64.3 | 56.2 |
| | C-LR-LSTM | 52.5 | 33.4 | 61.5 | 56.2 |
| | LR-C-LSTM | 61.4 | 53.5 | 66.2 | 57.8 |
| | LR-S-LSTM | **61.6** | **55.2** | **72.7** | **58.4** |

performance between the proposed and the baseline features in Table 3 suggests that it is possible to avoid the extensive training of dedicated inertial deep networks by using effective cross-domain knowledge transfer from vision research. The significant superiority of the proposed features in their overall accuracy (Table 2) over the recall values (Table 3) is partly due to the OVR strategy adopted, in which the true negative rate is expectedly higher.

Table 4 and 5 assess different strategies of multi-stream information fusion in the inertial and visual datasets, respectively. The top of Table 4 (*Individual*) evaluates the individual classification outputs of feature groups from the ActiveMiles and the WISDM-v2.0 datasets using a logistic regression (LR). The bottom of Table 4 (*Fusion*) validates the performance improvements when feature-level and decision-level fusion strategies are applied on information from different modalities and/or streams. C-LSTM and C-LR-LSTM do not include any merit-based weighting of the feature groups. As a result, the performance improvements are not significant. LR-C-LSTM and LR-S-LSTM significantly improve the performance compared to using individual feature groups. The accumulation of the LR out-

Table 5: Comparison of different fusion strategies on the FPV datasets. ('-': not available; C-LSTM: concatenation of feature groups followed by LSTM only; C-LR-LSTM: concatenation of feature groups followed by logistic regression and LSTM; LR-C-LSTM: concatenation of LR ouputs of the feature groups prior to the LSTM; LR-S-LSTM: accumulation of LR ouputs of the feature groups prior to the LSTM).

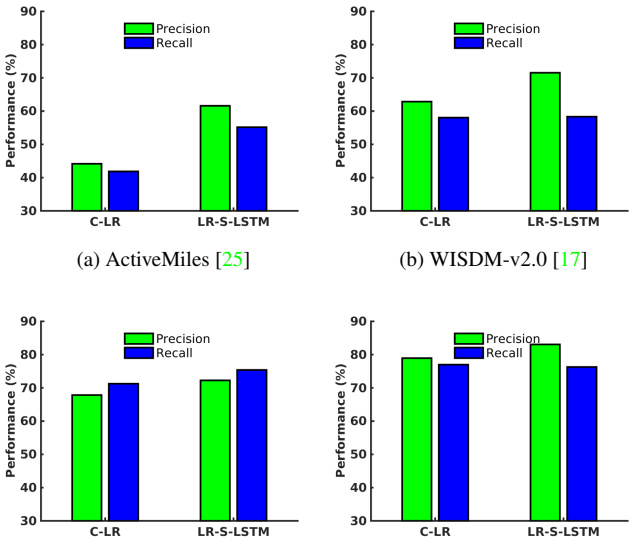
| | | HUJI [23] | | BAR [3] | |
|---|---|---|---|---|---|
| | | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ |
| Individual | Inception-Grid | 57.4 | 55.4 | 45.5 | 48.6 |
| | Inception-Centroid | 62.1 | 67.0 | 37.4 | 39.0 |
| | Inception-Inertial | - | - | 79.0 | 71.1 |
| | Handcrafted-2 [5] | - | - | 76.1 | **76.3** |
| Fusion | C-LSTM | 72.1 | **78.1** | 75.6 | 74.9 |
| | C-LR-LSTM | 70.7 | 74.6 | 47.2 | 49.0 |
| | LR-C-LSTM | 71.6 | 73.6 | **83.7** | 75.0 |
| | LR-S-LSTM | **72.3** | 75.4 | 83.1 | **76.3** |

puts in LR-S-LSTM reduces the input dimension of the LSTM and therefore reduces the size of the weight parameters, $W_{xo}, W_{xi}, W_{xf}$ and $W_{xc}$. Generally, the temporal encoding using the LSTM improves the precision and recall by at least $15\%$ in ActiveMiles. The improvement in WISDM-v2.0 is not significant compared to ActiveMiles. This is partly due to fewer motion streams in WISDM-v2.0, which does not contain gyroscope data.

The trend is similar in Table 5, where the fusion of feature groups improves performance in the FPV datasets. Due to the larger size of the HUJI dataset, C-LSTM achieves the highest performance, while the proposed LR-S-LSTM leads to $18\%$ and $12\%$ precision and recall improvements, respectively, compared to the best individual performance, i.e. Inception-Centroid. Since the BAR dataset is very small, the performance improvement due to the LSTM-based temporal encoding is not significant. However, the CNN features extracted from the stacked spectrograms of the accelerometer data perform equivalently to the handcrafted inertial features, and better than the CNN features from the grid optical flow and the centroid displacement. This shows the advantage of cross-domain knowledge transfer for human activity recognition when there are multi-modal information sources.

Figure 3 compares the LSTM-based long-term temporal encoding with C-LR outputs. C-LR uses feature concatenation followed by logistic regression. The results show that the LSTM improves the performance across all the datasets consistently. Particularly, the improvement is significant in the inertial datasets (Fig. 3a and 3b) partly because the inertial pipeline takes advantages of both handcrafted and CNN features. By exploiting long-term temporal dependencies, the LSTM reduces the number of false positives and hence increases the precision.

Finally, Table 6 compares different weighting strategies and the sigmoid activation prior to sparsity computation. The performance improves using the proposed weighting




(a) ActiveMiles [25] (b) WISDM-v2.0 [17]

(c) HUJI [23] (d) BAR [3]

Figure 3: The LSTM-based long-term temporal encoding with accumulation of the LR ouputs of the feature groups prior to the LSTM (LR-S-LSTM) outperforms the concatenation of the features followed by a logistic regression (C-LR).

Table 6: Comparison of sparsity weighting strategies on four datasets. (NSW: accumulation of LR outputs without sparsity weighting; SWNS: sparsity weighted without a sigmoid smoothing; LR-S-LSTM: sigmoid applied on the LR outputs followed by accumulation.)

| | ActiveMiles [25] | | WISDM-v2.0 [17] | | HUJI [23] | | BAR [3] | |
|---|---|---|---|---|---|---|---|---|
| | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ | $\mathcal{P}(\%)$ | $\mathcal{R}(\%)$ |
| NSW | **62.5** | **65.8** | 70.4 | 58.4 | 71.4 | 74.4 | 77.6 | 72.5 |
| SWNS | 60.5 | 60.9 | 68.6 | **58.5** | 71.9 | **75.5** | 73.9 | 70.4 |
| LR-S-LSTM | 61.6 | 55.2 | **72.7** | 58.4 | **72.3** | 75.4 | **83.1** | **76.3** |

strategy (LR-S-LSTM) in the multi-modal dataset, BAR, where the inertial and visual features have different discriminative characteristics. The weighting however tends to suppress discriminative characteristics in ActiveMiles [25], which contains equivalent discriminative characteristics among its streams. Moreover, the importance of the sigmoid smoothing is shown across the datasets as the performance of SWNS (sparsity weighted without sigmoid smoothing) is in general inferior to that of LR-S-LSTM. The comparison of existing video-based methods with the proposed framework is presented in [2].

## 5. Conclusion

We proposed a multi-modal proprioceptive activity recognition framework that integrates temporal features from first-person videos and ego-centric inertial data. We

used stacked spectrograms to exploit successful CNN-based image models via cross-domain knowledge transfer. Moreover, we proposed a sparsity weighted accumulation of information from different motion streams and/or modalities using logistic regression. This approach helps reducing the dimensions of the input to the LSTM network, which encodes long-term temporal dependency among activities, thus reducing the network complexity. The proposed framework was validated on multiple inertial and visual datasets: state-of-the-art performance is achieved on inertial datasets using only CNN features without explicitly training a dedicated network and without fusing handcrafted features.

As future work, we plan to apply problem-specific data augmentation techniques and re-train the last layer of the CNN with the spectrograms.

## Acknowledgment

G. Abebe was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA no 2010-2012.

## References

[1] G. Abebe and A. Cavallaro. Hierarchical modeling for first-person vision activity recognition. *Neurocomputing*, 267:362–377, June 2017. 3

[2] G. Abebe and A. Cavallaro. A long short-term memory convolutional neural network for first-person vision activity recognition. In *Proc. of International Conference on Computer Vision Workshops (ICCVW)*, Venice, Italy, October 2017. 1, 2, 3, 7

[3] G. Abebe, A. Cavallaro, and X. Parra. Robust multi-dimensional motion features for first-person vision activity recognition. *Computer Vision and Image Understanding (CVIU)*, 149:229–248, 2016. 3, 5, 7

[4] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan. Deep activity recognition models with triaxial accelerometers. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, Arizona, USA, February 2016. 5, 6

[5] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 1971–1980, Bruges, Belgium, April 2013. 5, 6, 7

[6] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing*, 37:1018–1022, 2015. 5, 6

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, USA, June 2009. 2, 3

[8] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami, USA, June 2009. 2, 4

[9] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, Boston, USA, June 2015. 2, 3, 5

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, Las Vegas, USA, June 2016. 2

[11] N. Y. Hammerla, S. Halloran, and T. Ploetz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016. 2

[12] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185 – 203, 1981. 2

[13] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5:1457–1469, 2004. 1, 4

[14] N. Hurley and S. Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009. 4

[15] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(1):221–231, 2013. 2

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, Ohio, USA, June 2014. 2, 3

[17] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011. 5, 6, 7

[18] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *Proc. of ACM on International Conference on Multimedia Retrieval*, pages 159–166, Amsterdam, The Netherlands, October 2016. 2

[19] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal. Design considerations for the WISDM smart phone-based sensor mining architecture. In *Proc. of ACM International Workshop on Knowledge Discovery from Sensor Data*, pages 25–33, San Diego, USA, August 2011. 5

[20] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903, Las Vegas, USA, June 2016. 2

[21] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in LSTMs for activity detection and early detection. In *Proc. of IEEE Conference on Computer Vision and Pattern*

*Recognition (CVPR)*, pages 1942–1950, Las Vegas, USA, June 2016. 2

[22] F. J. Ordóñez and D. Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016. 1, 2

[23] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora. Compact CNN for indexing egocentric videos. In *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, New York, USA, March 2016. 2, 5, 7

[24] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *Proc. of IEEE International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 71–76, San Francisco, USA, July 2016. 1, 2, 5, 6

[25] D. Ravi, C. Wong, B. Lo, and G. Z. Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE Journal of Biomedical and Health Informatics*, 21(1):56–64, January 2017. 1, 2, 5, 6, 7

[26] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016. 1

[27] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–904, Boston, USA, March 2015. 2

[28] S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2620–2628, Las Vegas, USA, June 2016. 2

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, USA, June 2015. 4

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Santiago, Chile, December 2015. 2

[31] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4305–4314, Boston, USA, June 2015. 2

[32] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, Boston, USA, June 2015. 2

[33] K. Zhan, S. Faux, and F. Ramos. Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients. *Pervasive and Mobile Computing*, 16, Part B:251–267, January 2015. 1