

# Hierarchical modeling for first-person vision activity recognition

Girmaw Abebe<sup>a,b,\*</sup>, Andrea Cavallaro<sup>a</sup>

<sup>a</sup>*Centre for Intelligent Sensing, Queen Mary University of London, London, UK*

<sup>b</sup>*CETpD, UPC-BarcelonaTech, Barcelona, Spain*

---

## Abstract

We propose a multi-layer framework to recognize ego-centric activities from a wearable camera. We model the activities of interest as hierarchy based on low-level feature groups. These feature groups encode motion magnitude, direction and variation of intra-frame appearance descriptors. Then we exploit the temporal relationships among activities to extract a high-level feature that accumulates and weights past information. Finally, we define a confidence score to temporally smooth the classification decision. The results across multiple public datasets show that the proposed framework outperforms state-of-the-art approaches, e.g. with at least 8% improvement in precision and recall on a 15-hour public dataset with six locomotive activities.

*Keywords:* activity recognition, first-person vision, hierarchical modeling, motion features, temporal context encoding.

---

## 1. Introduction

The increasing availability of wearable cameras enables the collection of first-person vision (FPV) data for the recognition of activities [1] at home [2], in the office [3] and during sport activities [4]. Ego-centric activity recognition has several important applications, which include life-logging and summarization [5, 6, 7, 8, 9, 10], assisted living [11, 12] and activity tracking [1, 2, 4, 13, 14, 15]. Of particular interest for a number of applications are basic locomotive activities such as *Walk, Turn, Run, Sit-down, Stand-up, Bow* and *Go upstairs/downstairs* [10, 11, 12, 16].

The main challenges of FPV activity recognition are motion blur, rapid illumination changes and outlier motions (for example due to other people captured by the camera). Moreover, the mounting position of the camera itself might cause self-occlusion (chest-mounted camera) or spurious motions (head-mounted camera) [11, 13, 17].

---

\*Corresponding author

*Email addresses:* g.abebe@qmul.ac.uk (Girmaw Abebe), a.cavallaro@qmul.ac.uk (Andrea Cavallaro)

Motion magnitude, direction and dynamics extracted from optical flow [4, 10, 11, 13, 14, 18, 19] and/or the temporal displacement of matched descriptors [12, 20] can be encoded to discriminate locomotive activities [4, 11, 12, 14, 18]. Moreover, using short and long-term temporal variations of intra-frame descriptors, such as histogram of oriented gradients (HOG) [21] and hidden layer features from Overfeat [22] and Caffe [23], can improve recognition performance [4, 10, 14, 18]. Other sensors (e.g. inertial) can also be used to complement video data [11, 16], however the signals generated by different sensors require synchronization.

In this paper, we propose a hierarchical locomotive activity recognition framework based on low-level features from optical flow, virtual inertial data and a variation of intra-frame appearance descriptors, as well as on temporal smoothing at two different levels. Our main contributions are: (i) the exploitation of temporal continuity both during modeling and decision by applying temporal weighting on previous information; (ii) a high-level feature that encodes hierarchical and temporal relationships among activities; (iii) a confidence-based output smoothing approach that exploits the decisions of previous samples only when the current decision does not achieve a minimum confidence threshold; and (iv) low-level features from optical flow and appearance descriptors that improve the discrimination capability of existing motion features [4]. We also employ frequency-domain pooling operations to encode the variation of intra-frame appearance descriptors but with shorter dimension of the feature space compared with time-series gradient pooling [14].

The paper is organized as follows. Section 2 reviews the related works. Section 3 presents the overview of the proposed framework. Section 4 describes the extraction of discriminative low-level features, and Section 5 presents the exploitation of temporal continuity during modeling and decision. The complexity analysis of the framework is studied in Section 6. Section 7 presents the experimental setup and discusses the results in comparison with the state of the art. Finally, Section 8 concludes the paper.

## 2. Related work

We review the feature groups and the classifier types that are used in the state of the art. Table 1 summarizes existing works and compares them with the proposed framework.

### 2.1. Features

Features for locomotive activity recognition in FPV can be categorized into four groups: keypoint-based [12, 20], optical flow-based [4, 10, 11, 13, 14, 16, 18, 19], virtual inertial-based [4] and appearance-based [14, 26, 27].

*Keypoint*-based features involve detection, description and matching of salient points across frames [28, 29, 30]. Zhang et al. [20] employed Shi and Tomasi [31] features and then extended the method to handle multi-scale detection of interest points [12]. Temporal characteristics are encoded in the feature space using

Table 1: Comparison of locomotive activity recognition methods. ✓ represents the availability of a specific element; \* shows other classifiers additionally experimented within the corresponding framework; -: can not be determined; † : authors compared against their previous methods.

		[4]	[14]	[24]	[13]	[18]	[11]	[19]	[10]	[12]	[16]	[25]	[20]	Ours	
Features	Keypoint-based	Displacement direction histogram								✓				✓	
	Optical flow-based	Raw grid			✓	✓	✓	✓	✓			✓	✓		✓
		Grid direction histogram	✓	✓						✓					✓
		Grid magnitude histogram	✓							✓					✓
		Direction frequency	✓												✓
		Magnitude frequency								✓					✓
	Virtual inertial-based	Centroid-based inertial	✓												✓
		Grid-based inertial													✓
	Appearance-based	Time-domain pooled		✓											✓
		Frequency domain pooled													✓
Modeling	Single-layer	✓	✓		✓	✓			✓	✓				✓	
	Multi-layer			✓			✓	✓			✓	✓		✓	
Classifiers	Support vector machine	✓	✓	*		✓	✓	✓		✓	✓	✓	✓	✓	
	K-nearest neighbors	*						*		*				*	
	Convolutional neural network				✓										
	Conditional random field						✓					✓			
	Logitboost			✓			✓	*				✓			
	Hidden Markov model			✓			✓							*	
	Dirichlet mixture model								✓						
	Naive Bayes									*					
	Logistic regression													✓	
	Decision tree													*	
Temporal continuity	Model-level						✓		✓				✓	✓	
	Decision-level			✓	✓			✓						✓	
Validation strategies	Multiple public datasets validation		✓	✓		✓								✓	
	Existing methods comparison		✓	✓		†		†		✓			†		✓
	Accessibility	Private			✓			✓			✓	✓	✓	✓	
		Public	✓	✓		✓	✓			✓		✓	✓	✓	✓
	Camera	Prototype									✓	✓		✓	✓
		Commercial	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
	Mounting position	Chest	✓								✓	✓		✓	✓
		Head		✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
	Other (non-locomotive) activities			✓	✓	✓	✓	✓	✓					✓	✓
	Number of activities		11	29	12	14	7	14	4	29	6	9	12	5	6
Number of subjects		4	1	5	13	13	30	-	1	1	-	5	-	13	
Duration of dataset (hr)		1.2	0.5	1	65	65	40	0.5	0.7	1.2	-	20	-	15	

the average standard deviation [20] and the combined standard deviation [12] of the direction histogram of the displacement between matched descriptors. However, keypoint-based features have limited performance in weakly textured regions and with motion blur.

*Optical flow*-based features mainly use grid optical flow [4, 11, 12, 18] as raw grid features [11, 13, 16, 18, 19], magnitude and/or direction histogram features [4, 10, 14] and frequency-domain features [4, 10]. *Raw grid features* require minimal or no additional processing from the grid flow data [11, 16, 18, 19]. Raw grid features do not encode key motion characteristics such as direction. The *histogram of the grid optical flow* is a more compact descriptor [4, 10, 14], which can be applied with independent direction and magnitude bins [4], joint spatial and direction bins [14] or joint magnitude, direction and magnitude-variance bins [10]. Spatial information is less discriminative in recognizing locomotive activities that are dominated by global motion [14].

Similarly to inertial features from accelerometer data, *virtual-inertial* data

can be derived from the displacement of the intensity centroid in the video and enhance the recognition performance [4].

Short and long-term temporal variations of *appearance descriptors* can be encoded using pooling operations [14]. Examples include the gradient of appearance (e.g. HOG [14]) or deeply learned image descriptors extracted from the hidden layers of appearance-based convolutional neural networks (CNNs) (e.g., Caffe [23] and Overfeat [22]). Summation and maximum pooling are not able to encode variations as much as histograms of time-series gradient (TSG) pooling [14].

The exploitation of temporal continuity of an activity is not sufficiently exploited in the state of the art. Accumulative smoothing is applied in [13] that under-utilize the temporal information, and complex graphical models are employed in [10] and [11].

## 2.2. Classification

Support vector machines (SVM) are the most commonly employed discriminative classifiers [4, 11, 12, 14, 16, 18, 19, 20, 24, 25]. A k-nearest neighbor (KNN) is a commonly used non-parametric geometrical classifier due to its simplicity [32] and is often used as baseline method [4, 10, 12, 19]. Hidden Markov models (HMMs) are basic sequential generative models [33, 34, 35] that smooth the outputs of a main classifier [11, 19, 24]. Conditional random fields (CRF) [34] enable structural learning across multiple temporal scales [11, 25].

Discriminative classifiers are preferred over generative models for locomotive activity classification as they require less input data for modeling [36, 37, 38]. Convolutional neural networks (CNNs) are also used to classify activities from a volume of grid optical flow data [13].

The classification performance can be improved by exploiting the temporal relationships of subsequent samples [10, 11, 13, 19]. Temporal encoding can be performed at model [10, 11, 19] or decision [13] levels. *Model-level exploitation* often employs complex models such as multi-scale conditional random fields [11] and Dirichlet mixture model [10] to exploit temporal continuity. *Decision-level exploitation* refines the output by weighting previous outputs [13] or by applying a smoothing classifier on the output of a main classifier [19]. Decision-level exploitation is simpler but may not encode the temporal continuity with sufficient granularity.

*Multiple* layers of classification can be performed when different modalities are used for sensing. Hence, a separate classification is performed on each modality followed by another classification on the combined outputs of the prior classifications [11, 19, 16]. Multi-layer classification can also be used to utilize additional characteristics, such as temporal dependencies, as we will discuss in the next section.

## 2.3. Data

HUJI is currently the largest public dataset for locomotive activity recognition [13]. Approximately 7.5 hours of video (50% of the dataset) or 17 out of

44 video sequences in the dataset are collected from publicly available YouTube videos. Examples include *Go upstairs* video sequences with significant illumination changes and *Run* video sequences that contain many occlusions from other runners. The IAR and BAR datasets [4] are much smaller compared to the HUJI dataset used in [18] that is later extended in [13]. Most papers, however, are not disclosing the dataset used in their validation [11, 12, 16, 19, 20].

The number of subjects who contributed to collect a dataset varies from one [12] to thirty [11]. The mounting position of a camera affects the quality of the data collected. Chest-mounted cameras [4, 12, 20] provide more stable videos but include self-occlusions. Head-mounted cameras are less susceptible to self-occlusion but their videos are significantly affected by head motion [10, 13, 18]. Several papers used laboratory prototypes as image sensors [11, 12, 16, 19, 20]. As a result the data quality of the corresponding datasets can not be easily compared.

### 3. Overview of the proposed framework

Let  $\mathcal{C} = \{A_j\}_{j=1}^{N_c}$  be a set of  $N_c$  activities of interest and  $(\mathbf{V}_1, \dots, \mathbf{V}_n, \dots, \mathbf{V}_N)$  be a video segmented into  $N$  temporally ordered activity samples. An activity sample,  $\mathbf{V}_n$ , is a windowed segment that is assumed to contain the minimum discriminative information required to be classified into one of the  $N_c$  activities. Samples might be temporally overlapping. Our objective is to classify each  $\mathbf{V}_n$  into its corresponding activity class  $A_j$ . To this end, we propose a framework that includes hierarchical modeling and activity modeling.

The proposed hierarchical modeling is a hand-designed modification of [18]. Each node in the hierarchy,  $M_e$ ,  $e \in \mathbb{Z}_{[1,5]}$ , represents a binary classification (Fig. 1):  $M_1$ : *Stationary vs Locomotive*;  $M_2$ : *Go upstairs vs Move along flat-space*;  $M_3$ : *Static vs Semi-static*;  $M_4$ : *Run vs Walk*;  $M_5$ : *Sit vs Stand*. Semi-static activities involve moderate head and leg movements, e.g. *Sit* and *Stand*. The activities at each  $M_e$  are defined in Table 2. We model the hierarchy by employing an SVM classifier at each binary node,  $M_e$ . Let  $\mathbf{f}_k$ ,  $k \in \mathbb{Z}_{[1,3]}$ , be a low-level feature group (see Section 4). We use each  $\mathbf{f}_k$  separately to find the corresponding hierarchical model parameters,  $\Phi_k = \{\phi_{ke}\}_{e=1}^5$ . Then we employ a logistic regression (LR) on a high-level feature,  $\mathbf{s}$ , that encodes the hierarchical and temporal relationships among activities (see Section 5). One-vs-all strategy is applied for activity modeling since it requires fewer classifiers compared to one-vs-one, i.e.,  $N_c < \text{binom}(N_c, 2)$  for  $N_c > 3$ , where  $\text{binom}(\cdot)$  computes the binomial coefficient of the first argument to the second.

### 4. The low-level features

We use different sets of motion features to improve the discrimination of activities. These features are grid features, virtual inertial features and pooled appearance features.

*Grid features (GF)*,  $\mathbf{f}_1$ , encode magnitude, direction and dynamics (frequency) of optical flow data. We propose a new feature subgroup, the Fourier

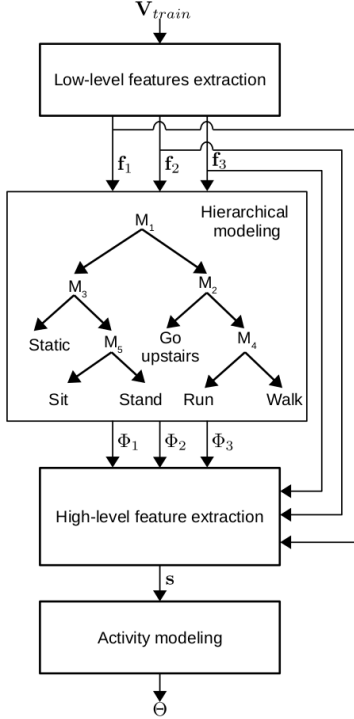
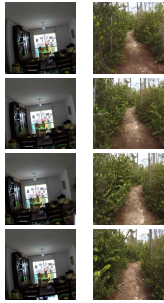
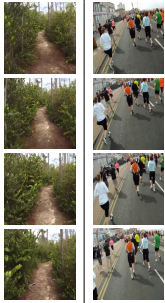
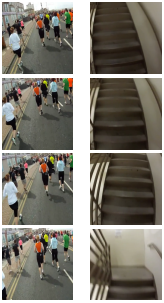
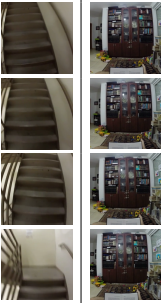
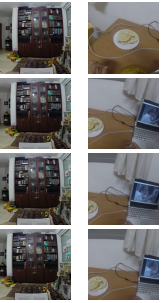

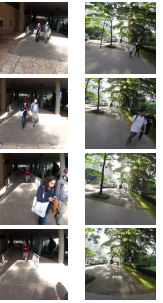





Figure 1: Details of the proposed multi-layer modeling framework used for learning a set of hierarchical model parameters,  $\Phi$ , and high-level activity model parameters,  $\Theta$ . Given a set of training videos,  $\mathbf{V}_{train}$ , the low-level feature groups,  $\mathbf{f}_k$ ,  $k \in \mathbb{Z}_{[1,3]}$ , are extracted and used to find the corresponding hierarchical model parameters,  $\Phi_k$ . These parameters are then used to extract a high-level activity feature,  $\mathbf{s}$ , that utilizes hierarchical outputs and their temporal relationships. One-vs-all activity modeling is performed on the high-level feature to find a set of activity model parameters,  $\Theta$ .

transform of motion magnitude (FMM) exploiting the variation of motion magnitude across frames (Fig. 2). The remaining feature subgroups of  $\mathbf{f}_1$  are motion direction histogram (MDH), motion magnitude histogram (MMH), motion direction histogram standard deviation (MDHS) and Fourier transform of motion direction (FMD) adopted from [4]. MDH, MDHS and FMD exploit the motion direction. MDH represents the average direction information, whereas MDHS and FMD evaluate the variation of direction in time and frequency domains, respectively. MMH and FMM describe the average and the frequency-response of motion magnitude, respectively. The importance of each feature subgroup depends on the type of variation existing among activities. For example, MMH and FMM are more useful to distinguish *Walk* and *Run*, whereas MDH, MDHS and FMD are more useful to discriminate activities containing different direction

Table 2: Definitions of activities per node,  $M_e$ , in the hierarchy of activities modified from [18]. Corresponding exemplar frames per activity set are shown vertically in order of increasing temporal indices.

Node	$M_1$		$M_2$		$M_3$		$M_4$		$M_5$	
Activity	Stationary	Locomotive	Move along flat-space	Go upstairs	Static	Semi-static	Walk	Run	Sit	Stand
Definition	User stays in similar place. Includes <i>sit</i> , <i>stand</i> and <i>static</i> .	User changes location. Includes <i>run</i> , <i>walk</i> and <i>go upstairs</i> .	Includes <i>run</i> and <i>walk</i> in a flat-path (no staircases).	Usual meaning. May contain <i>stationary</i> rest or <i>run</i> segments.	Head fixated while legs are stationary. E.g., <i>watch TV</i> .	<i>Static</i> but there is moderate (head and leg) motion.	Change location on foot. Slow forward motion.	User moves forward faster than walk. E.g., <i>morning jog</i> .	Usual sitting with natural head motion. E.g., <i>sitting on a chair</i> .	Usual standing. May contain a few walking steps.
Exemplar frames										

patterns.

All the subgroups of  $\mathbf{f}_1$  for the  $n^{th}$  activity sample of  $L$  consecutive frames are derived from the grid optical flow data,  $H_n = \{B_l\}_{l=1}^L$ .  $B_l$  is the set that contains  $G \times G$  grid vectors of a frame,  $B_l = \{B_l^g\}_{g=1}^{G^2}$ , where each  $B_l^g$  has horizontal,  $B_l^{gx}$ , and vertical,  $B_l^{gy}$ , components. We obtain compact histogram representations for motion direction and magnitude, respectively,  $P_n$  and  $O_n$  as

$$P_n = \text{hist}(\{\arctan(B_l^{gy}/B_l^{gx}) : \forall B_l^g \in B_l\}; \beta_d), \quad (1)$$

$$O_n = \text{hist}(\{|B_l^g| : \forall B_l^g \in B_l\}; \beta_m), \quad (2)$$

where  $\text{hist}(\cdot)$  is the operator that computes the histogram of its first argument, and  $\beta_m$  and  $\beta_d$  are the numbers of bins for magnitude and direction, respectively. We apply unit-frame normalization (3) and then temporal accumulation (4) to both  $P_n$  and  $O_n$  to derive MDH,  $\mathbf{f}_{1n}^1$ , and MMH,  $\mathbf{f}_{1n}^2$ , respectively. The normalization and accumulation help to minimize the effect of short-term occlusion, illumination change and local motion in the segment. For example, the normalized direction histogram,  $P_n^-$ , is computed as

$$P_n^-(b, l) = P_n(b, l) / \sum_{b=1}^{\beta_d} P_n(b, l), \quad (3)$$

where  $l \in \mathbb{Z}_{[1, L]}$  and  $b \in \mathbb{Z}_{[1, \beta_d]}$ . The temporal accumulation per each bin in  $P_n^-$

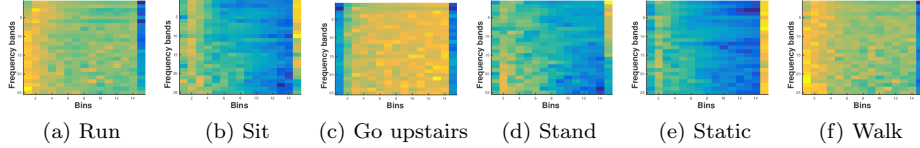


Figure 2: The proposed FMM exploits the frequency response of motion magnitude and groups it into different bands. The figures demonstrate that FMM can easily classify *Stationary* and *Locomotive* activities. *Stationary* activities do not have significant motion patterns except the high-frequency noise due to head-motion. *Going upstairs* involves high motion dynamics due to the closer scene appearance in indoor environments.

is applied as

$$\mathbf{f}_{1n}^1(b) = \sum_{l=1}^L P_n^-(b, l). \quad (4)$$

MDHS,  $\mathbf{f}_{1n}^3$ , exploits the dynamics of motion direction across frames by applying the standard deviation on  $P_n$ .

FMD and FMM represent frequency domain analyses on  $P_n$  and  $O_n$ , respectively, which involve grouping of their Fourier transforms into bands. For example, let  $F_n$  be the frequency response of  $P_n$ , the grouping of  $F_n$  into  $N_d$  bands,  $\hat{F}_n$ , is performed as

$$\hat{F}_n(n_d, b) = \sum_{l=\gamma_i}^{\gamma_f} \log |F_n(b, l)|, \quad (5)$$

where  $n_d \in \mathbb{Z}_{[1, N_d]}$  and its elements for the summation are  $\gamma_i = 1 + \frac{(n_d-1)L}{2N_d}$  and  $\gamma_f = \frac{n_d L}{2N_d}$  rounded to the nearest integer. We apply the normalization (3) and then the accumulation (4) operations on frequency-bands ( $\hat{F}$ ) representations of  $P_n$  and  $O_n$  to obtain FMD,  $\mathbf{f}_{1n}^4$ , and FMM,  $\mathbf{f}_{1n}^5$ , respectively. The final set of grid-based features is  $\mathbf{f}_{1n} = \{\mathbf{f}_{1n}^j\}_{j=1}^5$ .

*Virtual-inertial features (VF)*,  $\mathbf{f}_2$ , are extracted from the virtual-inertial data generated from video without employing the actual inertial sensor (Fig. 3). We propose generating virtual-inertial data from grid optical flow in order to improve the discrimination capacity of the centroid-based features proposed in [4]. We employ the average of the grid flow per frame,  $\boldsymbol{\nu}_l = \frac{1}{G^2} \sum_{g=1}^{G^2} B_l^g$ , as a virtual instantaneous velocity, in addition to the displacement of the intensity centroid,  $\boldsymbol{\eta}_l = \boldsymbol{\omega}_l - \boldsymbol{\omega}_{l-1}$ . The intensity centroid,  $\boldsymbol{\omega}_l$ , is derived from image moments that are calculated as the weighted average of all the intensity values in the frame [39].

Once we obtain the two virtual velocities, each with horizontal and vertical components, we cascade them as  $\boldsymbol{\chi}_l = \{\boldsymbol{\eta}_l, \boldsymbol{\nu}_l\}$ , and apply a *pre-extraction pro-*



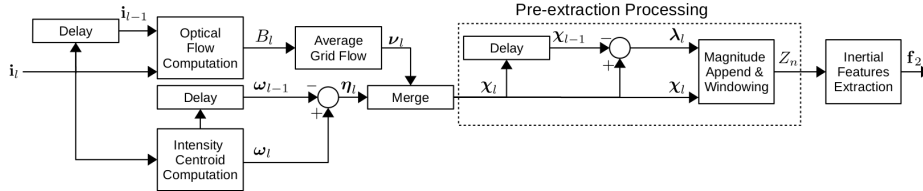


Figure 3: The proposed pipeline for virtual-inertial features extraction from video. Given a sequence of video frames, average grid flow and instantaneous centroid velocity are computed separately for each pair of consecutive frames ( $i_{l-1}$  and  $i_l$ ) and cascaded later. Then we generate the corresponding acceleration values using a simple difference operation. We derive and append magnitude components for the velocity and acceleration vectors. Finally, we extract a set of the state-of-the-art inertial features,  $f_2$ , in time and frequency domains for a windowed sample of  $L$  frames.

cessing that derives acceleration,  $\lambda_l$ , and magnitude components for both  $\chi_l$  and  $\lambda_l$ . The acceleration component is derived from the temporal derivation of the corresponding velocity component. Generally, from both the grid optical flow and the intensity centroid, we generate velocity and acceleration components and each of the two components has horizontal, vertical and magnitude vectors. Hence, the complete virtual inertial data of the  $n^{th}$  activity sample,  $Z_n$ , contains twelve vectors, i.e., six velocity and six acceleration vectors.

Finally,  $\mathbf{f}_{2n}$  is obtained from a cascade combination of the state-of-the-art inertial features that are extracted for each vector of  $Z_n$  in time and frequency domains [4, 40, 41, 42]. Time-domain features include *zero-crossing*, *kurtosis*, *energy*, *mean*, *standard deviation*, *minimum*, *maximum* and *median*. Zero-crossing measures the oscillatory behavior of a vector in reference to a zero value. Kurtosis quantifies whether a distribution is heavy-tailed or light-tailed with respect to a Gaussian distribution, i.e., high kurtosis represents a high probability of outliers and contains a heavy tail in the signal distribution. In addition, we extract a frequency domain feature from the Fourier transform of the vector. The majority of the features in  $\mathbf{f}_2$  are low-dimensional and susceptible to noise, however, they become significantly discriminative and robust when they are combined together.

*Pooled appearance features (AF)*,  $\mathbf{f}_3$ , exploit intra-frame descriptors to obtain additional discriminative motion information besides the grid features and virtual inertial features. *Pooling* operations are applied to extract the temporal variation of the intra-frame descriptors. We employ two intra-frame descriptors of different abstractions: HOG [14] and Overfeat [22]. HOG is selected due to its simplicity, and Overfeat [22] is extracted from the last hidden layer of the CNN and it is reported to be a success across different vision-based recognition tasks [14, 43].

Our proposed intra-frame appearance pooling consisting of one time-domain,  $v_1(\cdot)$ , and two frequency-domain,  $v_2(\cdot)$  and  $v_3(\cdot)$ , pooling operations to encode

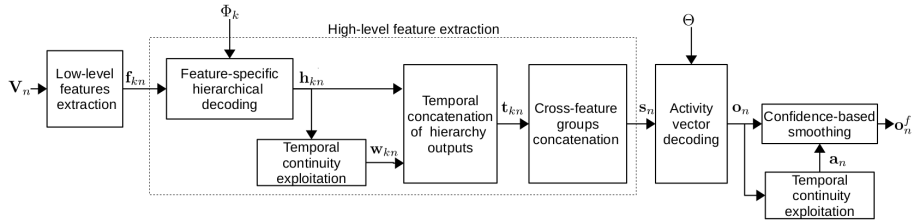


Figure 4: Overview of the proposed locomotive activity recognition for a video sample,  $\mathbf{V}_n$ , that uses three low-level feature groups,  $\{\mathbf{f}_{kn}\}_{k=1}^3$ . A high-level feature vector that encodes hierarchical and temporal relationships among activities is then extracted from the hierarchical outputs of the low-level features, followed by consecutive temporal and cross-feature groups concatenations. The activity decision vector,  $\mathbf{o}_n$ , is filtered using the proposed confidence-based smoothing approach. The hierarchical model parameters,  $\Phi_k$  and high-level model parameters,  $\Theta$  are obtained during modeling (Fig. 1).

short and long temporal characteristics of the intra-frame appearance descriptors. The time-series gradient (TSG) pooling proposed in [14] only considers time-domain *summation* and *histogram* of the gradient.  $v_1(\cdot)$  encodes the standard deviation of intra-frame descriptors across frames in a video, similarly to MDHS of grid-based features.  $v_2(\cdot)$  groups the frequency response of each time series data into bands as FMM and FMD.  $v_3(\cdot)$  encodes the power of each feature element in the frequency domain. Finally,  $\mathbf{f}_{3,n}$  is obtained from the concatenation of  $v_1(\cdot)$ ,  $v_2(\cdot)$  and  $v_3(\cdot)$  outputs of the HOG and Overfeat descriptors.

## 5. Exploiting temporal continuity

We exploit the temporal relationships among activities during both modeling and decision stages to improve recognition performance in case of short-term occlusions, blurred motion or large head-motion (Fig. 4).

### 5.1. Model-level temporal continuity

Model-level temporal continuity exploitation (MTCE) use temporal context, encoded from the hierarchical outputs of all the feature groups, during activity modeling. MTCE provides the temporal component of the high-level feature using a temporally weighted accumulation of past outputs.

Given the feature-based hierarchical model parameters,  $\Phi_1, \Phi_2$  and  $\Phi_3$ , the hierarchical decoding of the  $n^{\text{th}}$  activity sample results in a ten unit long output per feature group,  $\mathbf{h}_{kn} = \{\mathbf{h}_{kn}^e\}$ ,  $\forall e \in \mathbb{Z}_{[1,5]}$ , where  $\mathbf{h}_{kn}^e$  contains the classification scores for both classes ( $c_1$  and  $c_2$ ) at each binary node  $M_e$ , i.e.,  $\mathbf{h}_{kn}^e = [\mathbf{h}_{ekn}^{c_1}, \mathbf{h}_{ekn}^{c_2}]$ . The outputs of both classes, rather than the winner only, are used to exploit the level of confidence in the binary classification from their relative scores. It also reduces the likelihood of bias in the activity modeling

by increasing the high-level feature dimension. Since the  $\mathbf{h}_{kn}$  values from the SVM are not bounded, we apply a sigmoid (logistic) function,  $S(\cdot)$ , that maps any real value  $\lambda$  to a value inside the bounded region  $(0, 1)$  as

$$S(\lambda) = \frac{1}{1 + \exp^{-\lambda}}. \quad (6)$$

MTCE provides  $\mathbf{w}_{kn}$  that represents the accumulation of hierarchical outputs of  $D$  previous samples, weighted according to their temporal distance to the current index  $n$  as

$$\mathbf{w}_{k,n} = \sum_{d=1}^D W(d) \mathbf{h}_{kn-d}, \quad (7)$$

where  $W(\cdot)$  is the weighting function applied to give more importance to recent samples and less importance to earlier samples as

$$W(d) = \frac{\exp(-d/D)}{\sum_{d=1}^D \exp(-d/D)}. \quad (8)$$

The current,  $\mathbf{h}_{kn}$ , and weighted,  $\mathbf{w}_{kn}$ , hierarchical outputs are concatenated to extract feature-specific temporal vectors,  $\mathbf{t}_{kn} = [\mathbf{h}_{kn}, \mathbf{w}_{kn}]$ . The high-level feature vector for the activity modeling is obtained from the cross-feature groups concatenation as  $\mathbf{s}_n = [\mathbf{t}_{1n}, \mathbf{t}_{2n}, \mathbf{t}_{3n}]$ . The high discrimination characteristic of  $\mathbf{s}_n$  is derived from the temporal and hierarchical information extracted from the three low-level feature groups.

## 5.2. Decision-level temporal continuity

In addition to the activity modeling, we exploit previous temporal information during the activity vector decoding (Fig. 4). Decision-level temporal continuity exploitation (DTCE) is applied to smooth the decision when the confidence of the classification fails to achieve a minimum threshold. This is performed by exploiting the temporal continuity, similarly to MTCE, using the decisions of the previous samples. We define *confidence* as the relative weight of the winning class probabilistic score (maximum of the activity vector) with the second maximum score. Rather than using a ‘blind’ accumulation with previous samples’ outputs as in [13], we propose a confidence-based smoothing strategy (Algorithm 1).

We argue that smoothing may not improve the recognition performance (if it does not degrade) when the confidence level is high. On the other hand, if the confidence of a decision does not satisfy the threshold value, additional decision knowledge from previous samples is more likely to improve performance. DTCE gives more weight to the recent decisions; whereas [13] applied equal weights to all previous decisions, which undermines the significance of the current output and its closely related temporal samples.

Let the decoding of the  $n^{th}$  sample, with a feature vector,  $\mathbf{s}_n$ , using a set of model parameters,  $\Theta$ , be an  $N_c$ -long activity vector,  $\mathbf{o}_n$ . We measure the

---

**Algorithm 1:** Algorithm for confidence-based smoothing
 

---

**Require:** Decision vectors,  $\{\mathbf{o}_n, \mathbf{o}_{n-1}, \dots, \mathbf{o}_{n-d}, \dots, \mathbf{o}_{n-D}\}$ ,  
 Weighting function,  $W(\cdot)$   
 Indexing function,  $I(\cdot)$   
 Confidence threshold,  $r_t$

**Ensure:** Final class label,  $\mathbf{o}_n^f$

```

 $\mathbf{o}_n^1 \leftarrow \max(\mathbf{o}_n)$ ,
 $\mathbf{o}_n^2 \leftarrow \max(\mathbf{o}_n \setminus \mathbf{o}_n^1)$ 
 $r_n \leftarrow \frac{\mathbf{o}_n^1}{\mathbf{o}_n^2}$ 
if  $r_n > r_t$  then
   $\mathbf{o}_n^f \leftarrow I(\mathbf{o}_n^1)$ 
else
   $\mathbf{a}_n \leftarrow \sum_{d=1}^D W(d)\mathbf{o}_{n-d}$ 
   $\mathbf{o}'_n \leftarrow \mathbf{o}_n \cdot \mathbf{a}_n$ 
   $\mathbf{o}'_n{}^1 \leftarrow \max(\mathbf{o}'_n)$ 
   $\mathbf{o}'_n{}^2 \leftarrow \max(\mathbf{o}'_n \setminus \mathbf{o}'_n{}^1)$ 
   $r'_n \leftarrow \frac{\mathbf{o}'_n{}^1}{\mathbf{o}'_n{}^2}$ 
  if  $r'_n > r_t$  then
     $\mathbf{o}_n^f \leftarrow I(\mathbf{o}'_n{}^1)$ 
  else
     $\mathbf{a}_n^1 \leftarrow \max(\mathbf{a}_n)$ 
     $\mathbf{a}_n^2 \leftarrow \max(\mathbf{a}_n \setminus \mathbf{a}_n^1)$ 
     $r_n^a \leftarrow \frac{\mathbf{a}_n^1}{\mathbf{a}_n^2}$ 
    if  $r_n^a > r_t$  then
       $\mathbf{o}_n^f \leftarrow I(\mathbf{a}_n^1)$ 
    else
       $\mathbf{o}_n^f \leftarrow I(\max_r\{\mathbf{o}_n^1, \mathbf{o}'_n{}^1, \mathbf{a}_n^1\})$ 
    end if
  end if
end if

```

---

confidence level,  $r_n$ , from the ratio of the maximum probabilistic value (winning class score),  $\mathbf{o}_n^1$ , to the second maximum value,  $\mathbf{o}_n^2$ . We compare  $r_n$  to an experimentally found threshold value,  $r_t$ . If the threshold is satisfied, the winning class becomes the final class label,  $\mathbf{o}_n^f$ . Otherwise we update  $\mathbf{o}_n$  to  $\mathbf{o}'_n$  by including temporal information obtained using a weighted accumulation of the activity vectors of the previous  $D$  samples,  $\mathbf{a}_n$ , similarly to (7) and (8). The confidence is then re-evaluated,  $r'_n$ , and if the threshold is still not satisfied, either of the winning classes among  $\mathbf{o}_n$ ,  $\mathbf{o}'_n$  and  $\mathbf{a}_n$  with the maximum confidence score becomes the final class label.

Both MTCE and DTCE exploit the previous knowledge in order to improve the recognition of the current,  $n^{\text{th}}$ , sample. However, they might undermine

the recognition of a short activity segment (e.g., *Stand*) that appears abruptly in the middle of an other activity (e.g., *Walk*). Comparatively, MTCE provides a framework to learn from temporal relationships since the previous knowledge is incorporated in the modeling stage, whereas DTCE adopts a slightly rough smoothing, limited to exploit additional discriminative characteristics from the current and previous decisions.

## 6. Complexity Analysis

Let  $R$  be the height and  $C$  be the width of each frame in pixels. The complexity of the optical flow computation is  $\mathcal{O}(n_w^2 RC)$  per frame pair, with  $n_w$  being the number of warp parameters [44]. The computation of the intensity centroid requires  $\mathcal{O}(RC)$  and the computation of the average grid flow with  $G \times G$  grids requires  $\mathcal{O}(G^2)$  per frame. The cost of generating deeply learned appearance descriptors is approximately  $\mathcal{O}((RC(l+1))^{RC+h})$  per frame, where  $l$  is the number of layers and  $h$  is the number of hidden neurons per layer [45]. As for *grid-based features*, most of the intermediate steps in extracting feature subgroups have linearly growing complexity. For example, MDH and MMH cost  $\mathcal{O}(G^2 + \beta_d)$  and  $\mathcal{O}(G^2 + \beta_m)$ , respectively, for  $\beta_d$  direction bins and  $\beta_m$  magnitude bins, after the corresponding grid direction and magnitude are computed.

The Fourier transform for the frequency-domain features FMD and FMM cost  $\mathcal{O}(\beta_d L \log L)$  and  $\mathcal{O}(\beta_m L \log L)$ , respectively, for a video segment of  $L$  frames. Furthermore, each Fourier transform cost is increased by  $\mathcal{O}(L^3 N_b)$  due to the magnitude computation of the frequency response, logarithmic scale change and the grouping into  $N_b \in \{N_d, N_m\}$  frequency bands. Similarly to  $\mathbf{f}_1$ , it is only the frequency feature that has a significant complexity among subgroups in  $\mathbf{f}_2$ , which is equivalent to  $\mathcal{O}(L^3 \log L)$  for each virtual inertial vector. The proposed pooling operations,  $v_1(\cdot)$ ,  $v_2(\cdot)$  and  $v_3(\cdot)$ , applied on  $\beta_q$  dimensional infra-frame descriptor cost  $\mathcal{O}(\beta_q)$ ,  $\mathcal{O}(\beta_q L^4 N_q \log L)$  and  $\mathcal{O}(\beta_q L^2 \log L)$ , respectively. An SVM training costs  $\mathcal{O}(\max(N_t, N_k), \min(N_t, N_k)^2)$  on a data of  $N_t$  train samples, where each sample is represented with  $N_k$ -dimensional  $\mathbf{f}_k$  [46]. The logistic regression cost increases linearly with the data size as  $\mathcal{O}(N_t)$ . The temporal continuity constraints introduce a complexity of  $\mathcal{O}(D)$  per feature group,  $\mathbf{f}_k$ .

Table 3 shows the summary of the wall-clock computation time elapsed for the extraction of the proposed features for a randomly selected  $\approx 3$  s long segment. The computation bottleneck lays on the initial motion estimation (grid optical flow and intensity centroid) or appearance description (HOG and Overfeat) than the proposed features extraction. Particularly, it takes about 140 s to derive Overfeat [22]. This is partly because we use the pre-compiled binaries. The experiments were conducted using Matlab2014b, i7-4770 CPU @ 3.40GHZ, Ubuntu 14.04 OS and 16GB RAM.

Table 3: Summary of wall-clock time elapsed for the computation of proposed features experimented on a randomly selected  $\approx 3s$  long video segment. MDH: motion direction histogram; MDHS: motion direction histogram standard deviation; FMD: Fourier transform of motion direction; MMH: motion magnitude histogram; FMM: Fourier transform of motion magnitude; HOG: histogram of oriented gradient;  $v_1$ : standard deviation pooling;  $v_2$  and  $v_3$  are frequency domain pooling operations.  $v_2$  decomposes the frequency response into bands whereas  $v_3$  computes the power in frequency domain.

Feature source	Feature subgroups	Feature groups
Grid optical flow = 3.83 s	MDH = 3.92 ms	GF = 3.84 s
	MDHS = 3.97 ms	
	FMD = 4.34 ms	
	MMH = 2.80 ms	
	FMM = 1.95 ms	
Intensity centroid = 6.69 s	time-domain = 1.58 ms	VF = 10.54 s
Average grid flow = 3.84 s	frequency-domain = 3.28 ms	
HOG [14] = 13.16 s	HOG- $v_1$ = 0.15 ms	AF = 153.39 s
	HOG- $v_2$ = 1.36 ms	
	HOG- $v_3$ = 0.21 ms	
Overfeat [22] = 140.07 s	Overfeat- $v_1$ = 2.73 ms	
	Overfeat- $v_2$ = 48.13 ms	
	Overfeat- $v_3$ = 4.30 ms	

## 7. Results

We evaluate the proposed approach using ten main experiments. First, we compare its recognition accuracy with the state-of-the-art methods. We compare our framework with cumulative displacement curves (CDC) [18], robust motion features (RMF) [4], average pooling (AP) [11, 12] and multi-resolution good features (MRGF) [12, 20] in the state of the art. CDC [18] is selected due to its hierarchy-based decomposition of activities similar to the proposed framework, whereas RMF [4] is chosen as it involves similar magnitude, direction and dynamics encoding strategies. AP [11, 12] is a baseline as it contains *raw grid features* with no explicit extraction of specific motion characteristics. We also evaluate MRGF [12, 20], which is a keypoint-based approach that exploits the direction of the displacement vector between matched descriptors.

Second, we evaluate the performance of each feature group on the hierarchical classification. Third, we evaluate the subgroups of each feature group separately. Fourth, we show how the proposed temporal context exploitation (TCE) strategy improves the recognition performance. The fifth experiment validates the proposed TCE when it is applied on the state-of-the-art features. Sixth, following the analysis of misclassification in the confusion matrices, we show how the TCE becomes more effective when the activities are distinctively defined, first, by merging *Sit* and *Stand* to *Sit/Stand*, followed by a merging of *Static* and *Sit/Stand* to *Stationary*. Seventh, we compare our proposed pooling of the intra-frame descriptors with time-series gradient pooling [14]. Eighth, we validate the discriminative characteristics of the proposed feature groups across

Table 4: Number of video segments and their total duration per activity in the considered dataset[13]. The percentage that each activity covers of the whole dataset is also given. The class imbalance problem can be easily depicted as *Run* activity alone amounts for 47% of the whole dataset whereas *Stand* covers only 5%.

	Classes						Total
	Run	Sit	Go upstairs	Stand	Static	Walk	
Number of segments	13	11	13	15	14	19	<b>85</b>
Duration (mins)	409	96	151	47	104	62	<b>869</b>
Percentage (%)	47	11	17	5	12	7	<b>100</b>

Table 5: Summary of the number of video segments collected by each of the four subjects ( $S_1$ ,  $S_2$ ,  $S_3$  and  $S_4$ ) in the BAR dataset. Reco.: Recording; Sub.: Subject; L-R: Left-right turn; S-S: Sit-Stand

Sub.	Bow	Defend	Dribble	Jog	L-R	Pivot	Run	Shoot	S-S	Sprint	Walk	Total
$S_1$	4	3	8	4	8	14	4	30	4	2	4	<b>85</b>
$S_2$	4	6	8	4	4	6	4	30	4	4	4	<b>78</b>
$S_3$	4	9	8	4	4	14	4	29	4	4	4	<b>88</b>
$S_4$	4	6	6	4	5	12	4	26	5	4	4	<b>80</b>
<b>Total</b>	<b>16</b>	<b>24</b>	<b>30</b>	<b>16</b>	<b>21</b>	<b>46</b>	<b>16</b>	<b>115</b>	<b>17</b>	<b>14</b>	<b>16</b>	<b>331</b>

different classifiers, in comparison with the state of the art. The ninth experiments provide the results of three weighting strategies applied to solve the class imbalance problem. Finally, we also validate the proposed TCE on another public dataset and compare it with the state-of-the-art-methods.

### 7.1. Dataset and performance measures

We compare state-of-the-art approaches on multiple datasets. We use a public locomotive activity subset of HUJI<sup>1</sup>, which contains 15-hour long sequences collected in unconstrained settings (Table 4). All video segments are pre-processed to have a  $640 \times 480$  resolution and a  $30fps$  frame rate.

We also validate the proposed framework on another public dataset of basketball activity recognition, BAR<sup>2</sup>, which is smaller (1.2 hrs) than the HUJI dataset (15 hrs), but contains more dynamic basketball activities such as *Sprint*, *Dribble*, and *Shoot* (Table 5).

We employ equal decomposition of the available per-class video sequences into train and test sets (50% each) on the HUJI dataset [13]; whereas we employ a *one-subject-out* cross validation on the BAR dataset as the four subjects contribute an equivalent amount of data to the dataset. Different train and test set categorizations enable us to experiment the proposed framework under different validation strategies. Each experiment is repeated 100 times and the average performance is reported.

<sup>1</sup><http://www.vision.huji.ac.il/egoseg/videos/dataset.html>

<sup>2</sup><http://www.eecs.qmul.ac.uk/~andrea/FPV.html>

To measure the recognition performance for each class  $A_j$ , we measure true positive ( $TP$ ), true negative ( $TN$ ), false positive ( $FP$ ) and false negative ( $FN$ ) values.  $FP$  refers to the number of activity samples of the remaining classes,  $\mathcal{C} \setminus A_j$ , that are misclassified to  $A_j$ , whereas  $FN$  constitutes the number of activity samples of  $A_j$  that are misclassified to any one of the remaining classes,  $\mathcal{C} \setminus A_j$ . We use the accuracy measure  $\bar{\mathcal{A}} = \frac{TP+TN}{TP+TN+FP+FN} * 100\%$  to evaluate the performance at each node,  $M_e$ , of the hierarchy. We use *Precision*,  $\bar{\mathcal{P}} = \frac{TP}{TP+FP} * 100\%$ , and *Recall*,  $\bar{\mathcal{R}} = \frac{TP}{TP+FN} * 100\%$ , as our activity decoding metrics since we employ the one-vs-all strategy during the activity modeling.  $\bar{\mathcal{P}}$  and  $\bar{\mathcal{R}}$  are first computed for each class and then averaged to give the overall performance. To analyze the misclassification among activities we also employ the confusion matrix.

### 7.2. Parameters

To extract grid features and virtual-inertial features we adopt the parameter values in [4]. We adopt the settings employed in [14] for HOG and Overfeat extraction, but we change the grid dimension for HOG from  $5 \times 5$  to  $7 \times 7$  as the frame resolution changes from  $320 \times 240$  to  $640 \times 480$ , respectively. We use the same number of bands for  $v_2(\cdot)$  similarly to the FMD and FMM. The number of previous samples used for extracting the temporal knowledge is found experimentally by iteratively test different temporal duration (previous samples) on each feature group and their combination for a fixed set of train and test data (Fig. 5a). Finally, we set  $D = 13$  samples ( $\approx 20$  s) and it is shown that more previous knowledge does not significantly improve the performance. We set the confidence threshold,  $r_t$ , for the DTCE after similar experiment is performed iteratively as shown in Fig. 5b. It is observed that all the separate feature groups (GF, VF and AF) achieve performance improvement up to  $r_t = 6$ , which we set for our experiments. As shown in Fig. 5b, the performance becomes stable for all the feature types for the further increments of  $r_t$ . This is because the DTCE follows hard-coded rules (see Algorithm 1). Hence, as the threshold becomes too large to satisfy (for a fixed  $D$ ), the Algorithm follows the last option and the class label becomes  $\mathbf{o}_n^f = I(\max_r \{\mathbf{o}_n^1, \mathbf{o}_n^1, \mathbf{a}_n^1\})$ . It is also observed that the DTCE is more effective on the separate feature groups than their combination. This is because the combined feature is more discriminant, therefore it can satisfy the threshold easily.

### 7.3. Comparison with alternative methods

Table 6 shows that CDC [18], MRGF [12, 20] and AP [11, 19] achieve at least 22% lower in  $\bar{\mathcal{P}}$  and  $\bar{\mathcal{R}}$  with respect to the *Proposed*. The superiority of the proposed method is due to the higher discriminative capability of its feature groups and the use of previous information via MTCE and DTCE. Compared to RMF [4], our proposed low-level features, FMM and grid-based virtual-inertial features, improve  $\bar{\mathcal{P}}$  and  $\bar{\mathcal{R}}$  by 9% and 8%, respectively. The results also show the inferiority of keypoint-based methods to optical flow-based methods in such



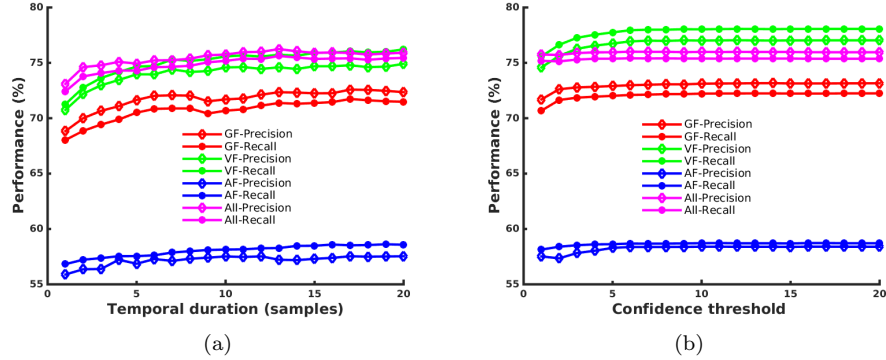


Figure 5: Experimental setting of parameters using fixed set of train and test sets; (a) different numbers of previous samples,  $D$ , are experimented to determine the amount of previous information for temporal encoding; (b) different threshold values are experimented for confidence-based temporal encoding. Results show that the temporal context encoding improves the performance of separate feature groups more significantly than that of their combination.

Table 6: Per-class recall performance of the state-of-the-art features validated using the SVM classifier and compared with the proposed framework.  $\bar{\mathcal{P}}$ : Precision (%);  $\bar{\mathcal{R}}$ : Recall (%).

Feature	Classes						Overall	
	Run	Sit	Up-stair	Stand	Static	Walk	$\bar{\mathcal{P}}$	$\bar{\mathcal{R}}$
CDC [18]	74	42	63	12	87	48	56	56
RMF [4]	91	53	90	15	88	80	69	71
MRGF [12, 20]	61	19	66	14	69	40	45	47
AP [11, 19]	44	48	81	10	95	43	52	57
Proposed	<b>99</b>	<b>87</b>	<b>100</b>	3	<b>96</b>	<b>88</b>	<b>78</b>	<b>79</b>

a challenging dataset. Since CDC [18] was proposed for the recognition of long-term activity segments ( $\approx 17s$ ), it is shown to be less effective for short activity segments ( $\approx 3s$ ). Generally, the superior performance of our proposed method over the state-of-the-art methods on the largest publicly available dataset (see Section 2.3) reflects the higher capability of our method to deal with the first-person vision challenges.

Among the classes, *Sit* has been improved significantly from 53% using RMF to 87% using *Proposed*. However, due to the following reasons, the same improvement can not be achieved to *Stand* though both *Sit* and *Stand* are stationary activities with head-driven motion in their first-person videos. First, the amount of data available for each of the two activities is not equivalent as *Sit* (11%) contains twice the amount of data than *Stand* (5%) as shown in Table 4. Hence, the lack of more training information for *Stand* results in the underfitting of its model, i.e., lower performance. Second, the same temporal smoothing

process in the proposed framework affects the two activities differently due to their different frequencies and durations in the dataset. Compared to *Sit*, *Stand* video segments are often observed in between other activities with a shorter duration. Specifically, there are 15 *Stand* and 11 *Sit* video segments in the dataset as shown in Table 4. However, the average duration of a *Stand* segment is 3.15 mins ( $\sigma = 7.34$  mins) compared to 9.35 mins ( $\sigma = 9.15$  mins) of a *Sit* segment, where  $\sigma$  represents the standard deviation of segment durations. As the result, a *Stand* sample is more likely to be smoothed towards its pre-occurring activity in the sequence. Third, per the definitions of the activities in Table 2, *Stand* is relatively more difficult to be recognized compared to *Sit*. This is because *Stand* may contain a few walking steps, which result in misclassification of *Stand* samples to *Walk*.

#### 7.4. Evaluation of the feature groups

We evaluate the independent performance of each feature group (and their combinations) at each node,  $M_e$ , of the hierarchy in the proposed framework, and we also compare with the state-of-the-art features as shown in Table 7. Note that we use the acronyms for the proposed feature groups (*GF*, *VF* and *AF*), rather than the variables ( $\mathbf{f}_1$ ,  $\mathbf{f}_2$  and  $\mathbf{f}_3$ ), in the following discussion. Almost all features are shown to achieve more than 85% accuracy at  $M_1$  (*Stationary* vs *Locomotive*). MRGF [12, 20] achieves the lowest accuracy (78%) at  $M_1$  expectedly since it does not utilize magnitude information that could have easily discriminated the two activity sets. Note that  $\bar{\mathcal{A}}$  is affected by the class imbalance problem.

For all nodes,  $M_1 - M_5$ , at least one of the proposed feature groups achieves the highest accuracy. *VF* achieves higher accuracy in classifying activities with well-defined motion patterns, e.g., *Run* vs *Walk* at  $M_4$ , whereas *GF* is more effective when the motion patterns are less distinct, e.g., *Sit* vs *Stand* at  $M_5$ . *AF* achieves higher accuracy at  $M_2$  (*Move along flat-space* vs *Go upstairs*) and  $M_3$  (*Static* vs *Semi-static*) as there are unique appearance descriptors of *staircases* at  $M_2$ , whereas *Static* videos at  $M_3$  contain a typical case of *a person sitting while watching a movie* or *reading on the computer screen* in the dataset.

Generally, superior performances of *GF* at  $M_1$  and  $M_5$ , *VF* at  $M_1$  and  $M_4$ , and *AF* at  $M_2$  and  $M_3$  validate our proposal of utilizing different features groups according to their importance across the nodes in the hierarchy. Though the feature groups are used separately in the hierarchy, the combination of *GF*, *VF* and *AF* achieves the highest performance almost at all the nodes except at  $M_4$ .  $M_4$  refers to the binary classification between *Run* and *Walk* activities. Since the two activities experience different motion dynamics, they can be easily differentiated with motion-driven features (92% with *GF* and 94% with *VF*) as shown in Table 7. However, these activities do not involve differences in their occurring environments, i.e., both contain similar appearance information. Hence, appearance-driven features (74% with *AF*) are not as discriminant as motion-driven features (*GF* and *VF*). Thus, the concatenation of the high dimensional *AF* with *GF* and *VF* introduces the less discriminative characteristics

Table 7: Performance of existing and proposed feature groups at each node,  $M_e$ , of the hierarchy in terms of the binary classification accuracy,  $\bar{\mathcal{A}}$  (%). GF: grid features; VF: virtual-inertial features; AF: pooled appearance features.

Features		Nodes				
		$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
Existing	CDC [18]	90	85	83	79	54
	RMF [4]	<b>96</b>	<b>98</b>	83	<b>90</b>	<b>58</b>
	MRGF [12, 20]	78	79	72	80	56
	AP [11, 19]	86	93	<b>88</b>	62	56
Proposed	GF	<b>96</b>	99	88	92	<b>64</b>
	VF	<b>96</b>	95	85	<b>94</b>	59
	AF	93	<b>100</b>	<b>93</b>	74	62
	GF+VF	<b>96</b>	98	88	<b>94</b>	62
	GF+AF	95	99	<b>96</b>	79	58
	VF+AF	94	99	95	76	63
	GF+VF+AF	<b>96</b>	<b>100</b>	<b>96</b>	86	<b>66</b>

between *Run* and *Walk*, though it improves the performance at all other nodes ( $M_1$ ,  $M_2$ ,  $M_3$  and  $M_5$ ).

We evaluate the significance of the subgroups within each feature group: MDH, MDHS, MMH, FMD and FMM in GF; centroid-based and optical flow-based virtual inertial features in VF; and intra-frame appearance descriptors pooled with the proposed operations  $v_1(\cdot)$ ,  $v_2(\cdot)$  and  $v_3(\cdot)$ . Figure 6 shows that GF, VF and AF achieve improved performance by including all their corresponding feature subgroups. Figure 6a illustrates that motion direction contains more dynamic information than the magnitude as depicted from their corresponding Fourier domain analysis. Figure 6b shows that the proposed optical flow-based virtual inertial feature outperforms the centroid-based inertial feature presented in [4]. This is partly because optical flow represents more direct estimation of motion than the displacement of intensity centroid. Fig. 6c shows that  $v_2(\cdot)$  pooling is less effective compared to  $v_1(\cdot)$  and  $v_3(\cdot)$ . This is because  $v_2(\cdot)$  reduces the original dimension of the intra-frame descriptor into few bands resulting under-fitting, whereas  $v_1(\cdot)$  and  $v_3(\cdot)$  keep the original feature dimension.

### 7.5. Temporal context

The temporal context exploitation, achieved using MTCE and DTCE, is the main reason for the superior performance of the proposed framework to the state of the art. Figure 7 shows the improvement of the recognition performance for almost all classes due to MTCE and DTCE. A significant improvement is observed as the misclassification of *Sit* as *Stand* reduces from 20% in Fig. 7a to 12% in Fig. 7b due to MTCE and to 9% in Fig. 7c due to DTCE. The same analogy can be applied to the 14% misclassification of *Run* as *Walk*. MTCE and DTCE are shown to improve the performance equivalently though MTCE is supposed to be more influential. However, the confidence-based smoothing and the weighted accumulation of previous outputs in DTCE plays a more crucial role than initially anticipated.

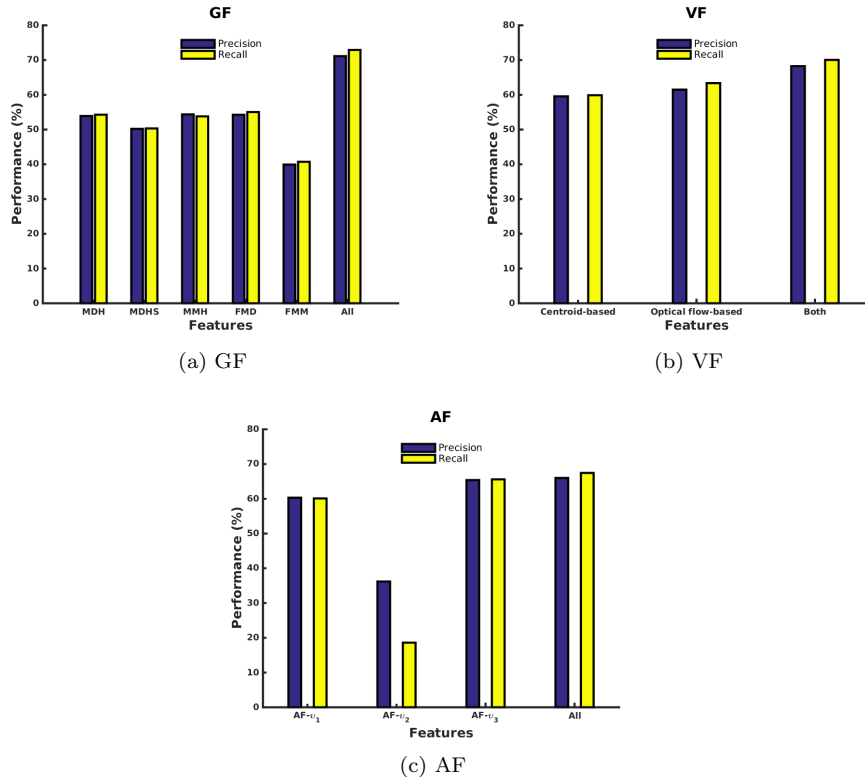


Figure 6: Performance of feature subgroups in each of the proposed GF, VF and AF. The pooling operations include  $v_1$ : standard deviation,  $v_2$ : grouping into frequency bands and  $v_3$ : power in frequency domain.

The combination of both MTCE and DTCE reduces the misclassification of *Walk* as *Run* by 5% (Fig. 7d), which was otherwise impossible using only one of the two (Fig. 7b and Fig. 7c). The less effectiveness of the TCE for *Walk* and *Stand*, in comparison with *Run* and *Sit*, is due to the skewness problem in the dataset. Activities occurring for long temporal duration (*Run* and *Sit*) are more likely to dominate the less represented short duration activities (*Stand* and *Walk*) in the dataset.

Furthermore, we validate the significance of our proposed temporal continuity exploitation by applying it on the state-of-the-art features during modeling and decision. Figure 8 shows the following average per-class recognition improvements: 9% on CDC [18], 6% on RMF [4], 11% on MRGF [12, 20] and 12% on AP [11, 19]. This highlights the potential of our temporal context approach to advance the discriminative characteristics of any feature type.

Across the confusion matrices in Fig. 7 and 8, two misclassification errors have occurred consistently. First, *Sit* and *Stand* activities are often classi-

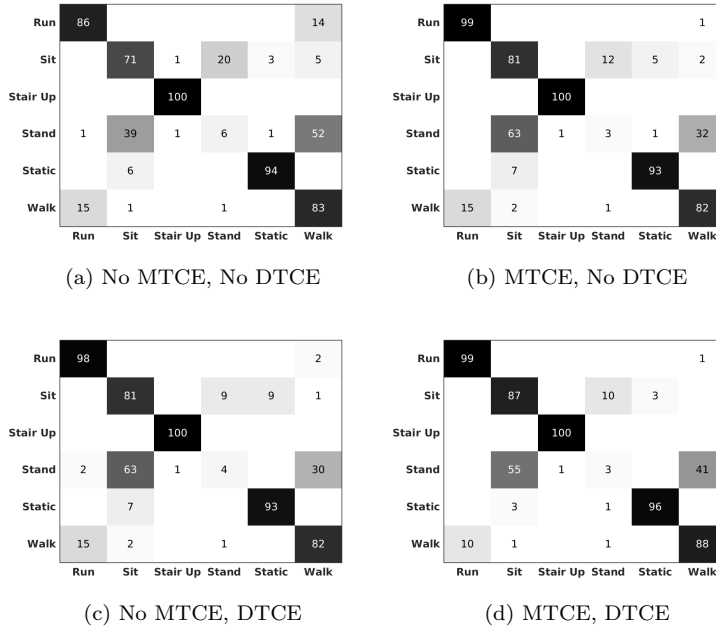
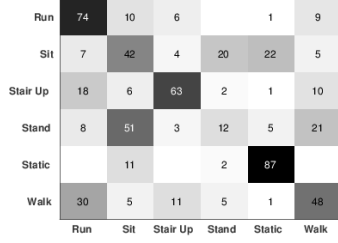


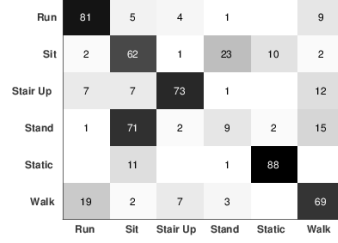
Figure 7: Comparative performance of the proposed framework at different stages; 7a: the hierarchical output without the use of any previous knowledge; 7b: only previous samples knowledge is encoded during modeling (MTCE); 7c: only confidence-based smoothing is applied (DTCE); 7d: both MTCE and DTCE are applied.

fied with inferior performance with a significant misclassification between them. This can also be understood from the least performance at  $M_5$  of the hierarchy in Table 7. The main reasons are, first, neither *Sit* nor *Stand* has distinctive characteristics (motion and/or appearance) that can be utilized during feature extraction. Second, the lack of enough data for these activities in the dataset (see Table 4) worsens the problem and results in under-fitting. Misclassification of *Walk* segments as *Run* is often evident in the confusion matrices due to the significant resemblance of some *Run* segments to *Walk* segments in the dataset. In addition, the significant percentage of *Stand* activity is also misclassified as *Walk* because considerable *Stand* videos in the dataset include short walking segments as defined in Table 2, e.g., *a subject standing and waiting for a bus while making few walking steps in the bus stop*.

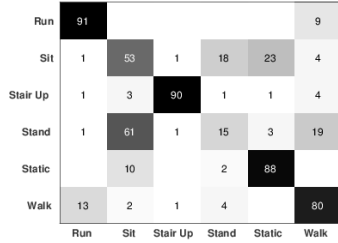
We also experiment the proposed framework by merging activities with no distinctive first-person vision characteristics between them. The merging also eases the class imbalance problem in the dataset. We start by merging *Sit* and *Stand* classes to a single activity, *Sit/Stand* as they both involve random head movement while the subject is stationary. The result is shown in Fig. 9a. We



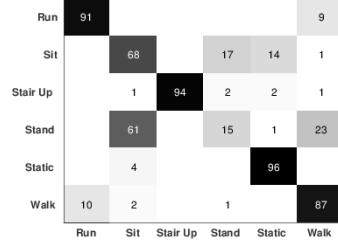
(a) CDC [18]



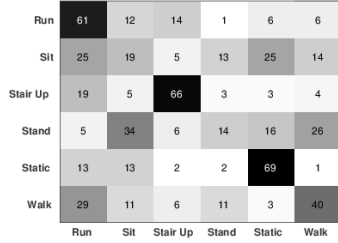
(b) CDC-TCE



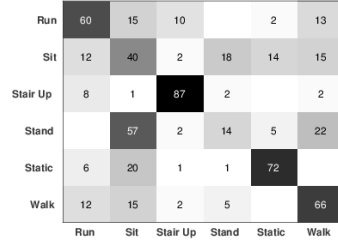
(c) RMF [4]



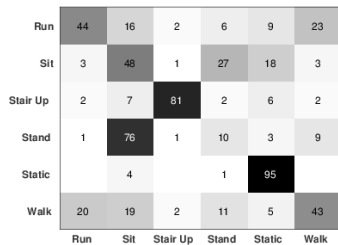
(d) RMF-TCE



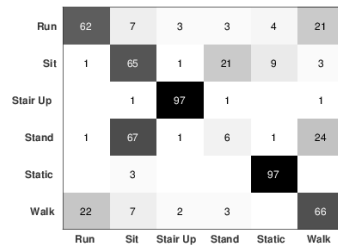
(e) MRGF [12, 20]



(f) MRGF-TCE



(g) AP [11, 19]



(h) AP-TCE

Figure 8: The validation of the proposed temporal continuity exploitation (TCE) on the state-of-the-art features. Figures 8a, 8c, 8e and 8g represent the original performances without TCE and Figures 8b, 8d, 8f and 8h show their respective improved performances after TCE.

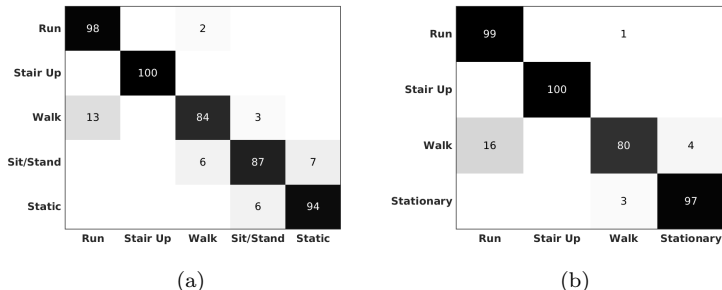


Figure 9: Performance is improved when similar classes are merged to a single activity; (a) *Sit* and *Stand* are merged to be a *Sit/Stand* activity; (b) *Sit/Stand* and *Static* are further merged to be a *Stationary* activity. Results show the improvement of the recognition performance when less clearly distinctive activities are merged.

Table 8: Comparison of the proposed pooling of intra-frame appearance descriptors with the time-series gradient (TSG) pooling [14]. Per-class recall ( $\bar{\mathcal{R}}$ ) values are given followed by the overall averaged precision ( $\bar{\mathcal{P}}$ ) and recall values (%). Dim.: dimension of the feature vector obtained after the pooling; *raw* refers the summation pooling of the raw feature data.

Feature	Per-class						Overall		Dim.
	Run	Sit	Up-stair	Stand	Static	Walk	$\bar{\mathcal{P}}$	$\bar{\mathcal{R}}$	
HOG-raw [21]	77	21	75	1	71	62	51	51	392
HOG-TSG [14]	<b>81</b>	29	<b>93</b>	<b>3</b>	<b>85</b>	<b>73</b>	<b>59</b>	<b>61</b>	<b>2352</b>
HOG-proposed	77	<b>32</b>	90	<b>3</b>	81	65	57	58	809
Overfeat-raw [22]	78	43	99	0	92	74	62	64	4096
Overfeat-TSG [14]	<b>83</b>	57	99	<b>2</b>	<b>98</b>	<b>77</b>	<b>68</b>	<b>69</b>	<b>24576</b>
Overfeat-proposed	78	<b>59</b>	<b>100</b>	0	97	72	64	68	8217

further merge *Static* and *Sit/Stand* to *Stationary*. The result is shown in Fig. 9b. In comparison with Fig. 7d, we accomplish higher performance improvement in Fig. 9 that confirms the effectiveness of our framework for well defined activities, and further validates the resemblance of *Sit* and *Stand* segments.

### 7.6. Pooling

We also experiment the proposed pooling for intra-frame descriptors (HOG and Overfeat [22]) with the time-series gradient (TSG) pooling [14] as shown in Table 8. The results show that the proposed and TSG pooling improve the discrimination among activities in comparison with raw appearance features for both HOG and Overfeat. Among the two intra-frame descriptors, Overfeat outperforms HOG. Our pooling that contains  $v_1(\cdot)$ ,  $v_2(\cdot)$  and  $v_3(\cdot)$  often performs equivalent to TSG, while we manage to reduce the feature vector dimension almost three times. A specific reason for slight superiority of [14] is due to its

Table 9: Accuracy,  $\bar{A}$  (%), comparison of proposed features in the proposed framework when they are validated on different classifiers. SVM: support vector machine, KNN: k-nearest neighborhood, LR: logistic regression, DT: decision tree and HMM: hidden Markov model. GF: grid features; VF: virtual-inertial features; AF: pooled appearance features.

Feature	Classifier	Node				
		$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
GF	HMM	66	64	62	82	46
	DT	94	97	80	86	55
	KNN	95	<b>99</b>	81	88	59
	LR	69	98	60	91	52
	SVM	<b>96</b>	<b>99</b>	<b>88</b>	<b>92</b>	<b>64</b>
VF	HMM	56	81	61	43	57
	DT	93	93	81	91	56
	KNN	94	94	83	93	<b>60</b>
	LR	63	79	84	87	54
	SVM	<b>96</b>	<b>95</b>	<b>85</b>	<b>94</b>	59
AF	HMM	66	77	56	71	<b>63</b>
	DT	88	98	86	68	58
	KNN	92	96	89	<b>76</b>	52
	LR	<b>93</b>	<b>100</b>	91	72	62
	SVM	<b>93</b>	<b>100</b>	<b>93</b>	74	62

preservation of the raw appearance information through *maximum* and *summation* pooling operations whereas our proposed approach solely focuses on motion information derived from the raw description. Generally, appearance-driven features are shown to discriminate environment-specific activities (*Go upstairs* and *Static*) near-perfectly in comparison with motion-specific activities (*Run* and *Walk*).

### 7.7. Classifiers

In addition to using SVM and LR, we test the proposed feature groups on different classifiers, namely KNN, decision tree (DT) [47] and HMM. DT follows the hierarchical topology of decoding activities similar to the proposed framework. Table 9 shows the accuracy achieved by each proposed feature group at each node of the hierarchy using different classifiers. Expectedly, SVM achieves superior performance consistently across different feature groups and nodes in the hierarchy due to its discriminative and high-margin classification behaviors. The results validate our selection of the SVM as the principal classifier in the proposed framework. DT follows SVM closely and performs equivalently to KNN, which reflects the advantage of tree-based activity classification, i.e., the hierarchical structure in the proposed framework. HMM is shown to perform significantly inferior to the other discriminative classifiers due to its dependency on the input data model as of any generative models. LR also lags behind the SVM, DT and KNN but it provides equivalent performance to SVM on high dimensional pooled appearance features. Moreover, it is due to its simplicity that we select LR for the activity modeling using the high-level feature.



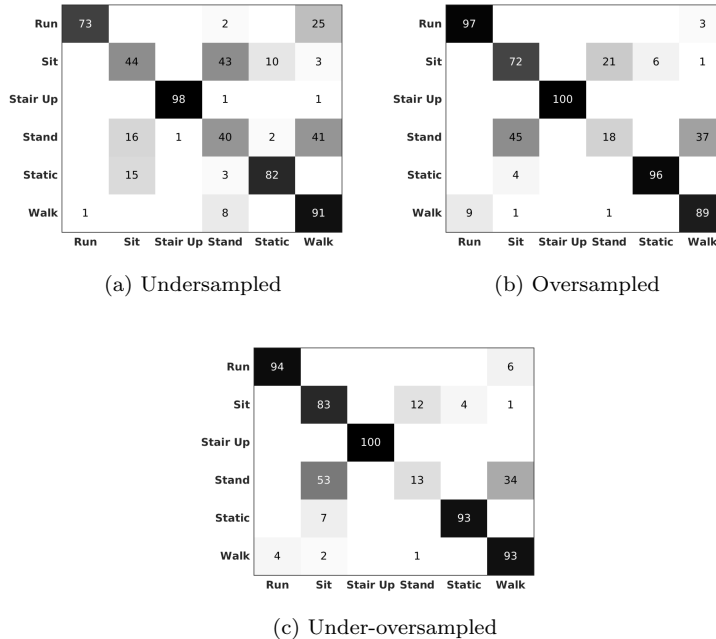


Figure 10: Comparison of different weighting strategies: undersampling, oversampling and under-oversampling, applied on the dataset followed by the proposed framework. These strategies aim to achieve equal amount of data among activities. Under-oversampling provides more accurate recognition performance than the remaining two approaches as it optimizes the bias (underfitting) due to undersampling and the variance (overfitting) due to the oversampling.

### 7.8. Weighted performance

Because data size variations among activities (class imbalance) affect the recognition performance as data-scarce activities (e.g., *Stand*) do not help the model generalize. Moreover, the dominance of data-rich activities (e.g., *Run*) results in their over-smoothing during temporal encoding.

To address the class imbalance problem, we apply three weighting strategies, namely *undersampling*, *oversampling* and *under-over sampling*. Undersampling reduces all activities to the minimum number of samples per activity in the dataset. *Oversampling* interpolates all activities to the maximum number of samples per activity in the dataset. Figure 10a shows that undersampling introduces the reduction of recognition performance for the majority of the activities except *Stand* (40%) since training is performed on less amount of data per class (i.e., a smaller dataset). Oversampling does not discard samples, however, it does not achieve data equivalence among activities as the interpolated samples are just replicas that do not introduce new information.

Table 10: Per-class recall performance of the state-of-the-art features on basketball activity recognition (BAR) dataset, which contains highly dynamic basketball activities. Results show that the proposed framework applied on GF and VF features result in the highest performance for the majority of the classes.  $\bar{P}$ : Precision (%);  $\bar{R}$ : Recall (%).

Feature	Classes										
	Bow	Defend	Dribble	Jog	L-R	Pivot	Run	Shoot	S-S	Sprint	Walk
CDC [18]	59	40	13	42	28	63	15	50	14	22	86
RMF [4]	<b>98</b>	86	82	32	<b>90</b>	98	<b>54</b>	97	68	<b>67</b>	96
MRGF [12, 20]	95	18	12	0	24	76	0	31	68	20	38
AP [11, 19]	4	0	0	0	12	34	0	3	10	0	91
Proposed	90	<b>89</b>	<b>95</b>	<b>54</b>	89	<b>99</b>	45	<b>100</b>	<b>71</b>	52	<b>100</b>

As trade-off between the two approaches, we under-over sample the dataset. This approach undersamples data-rich activities and oversamples data-scarce activities to the mean number of samples per class in the dataset. Figure 10c shows that equivalent overall performance is achieved with the original approach, but under-oversampling reduces the deviation among per class recall values from  $\sigma = 37.55$  (see Fig. 7d) to  $\sigma = 32.98$ .

### 7.9. Validation on multiple datasets

In addition to HUJI [13], we validate the proposed temporal context exploitation approach on BAR [4]. We also apply different train and test sets decomposition strategy (*one-subject-out*) during validation.

Table 10 shows that the proposed multi-layer temporal context encoding outperforms the state of the art. GF and VF are used to classify the basketball activities separately using an SVM classifier in one-vs-all approach. The proposed MTCE is applied on their outputs followed by the confidence-based DTCE. The recognition performance is improved for the majority of the classes. The accuracy for *Bow*, *Run* and *Sprint* is slightly reduced due to temporal smoothing. The misclassifications between *Bow* and *Sit-stand* as well as among *Jog*, *Run* and *Sprint* (Fig. 11) resulted from similar motion patterns of the corresponding sequential activities in the dataset. Hence, temporal modeling would further smooth the distinction between similar and sequential activities.

## 8. Conclusion

We proposed a framework that exploits hierarchical and temporal information using optical flow, virtual inertial data and intra-frame appearance descriptors to classify locomotive activities. The proposed motion feature groups extract salient characteristics of magnitude, direction and dynamics both in time and frequency domains. Each low-level feature group is separately used in the hierarchy in order to exploit its advantages across different nodes. A high-level feature, which contains both hierarchical and temporal information, is extracted in order to model each activity. The temporal component is encoded using a temporally weighted accumulation of the hierarchical outputs

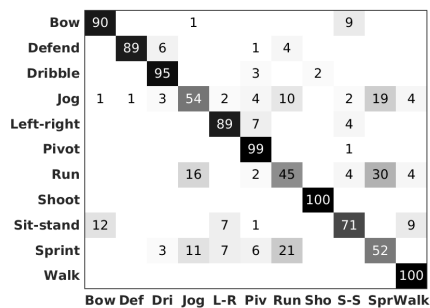


Figure 11: Confusion matrix of the proposed temporal context exploitation approach applied on the basketball activity recognition (BAR) dataset [4]. Misclassification among *Jog*, *Run* and *Sprint* results due to the similar motion patterns of the activities.

of the previous samples. The classification output is further refined using a confidence-based smoothing strategy. We validated the proposed framework on multiple datasets. Results demonstrated that the proposed feature groups are more discriminative than the state-of-the-art features. We showed that appearance features can be effectively integrated to enhance performance using well-designed pooling operations. The proposed temporal continuity exploitation strategies improve the recognition performance significantly. However, an activity with shorter duration and random occurrences inside a long temporal activity might be unnecessarily smoothed.

As future work, we will extend the proposed framework to decode videos using deeply learned motion features via recurrent neural networks exploiting both appearance and motion information in first-person videos.

### 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] H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), Providence, USA, 2012, pp. 2847 – 2854.
- [2] A. Fathi, Learning descriptive models of objects and activities from ego-centric video, Ph.D. thesis, Georgia Institute of Technology (2013).

- [3] K. Ogaki, K. M. Kitani, Y. Sugano, Y. Sato, Coupling eye-motion and ego-motion features for first-person activity recognition, in: Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, USA, 2012, pp. 1 – 7.
- [4] G. Abebe, A. Cavallaro, X. Parra, Robust multi-dimensional motion features for first-person vision activity recognition, *Computer Vision and Image Understanding (CVIU)* 149 (2016) 229 – 248.
- [5] Y. Bai, C. Li, Y. Yue, W. Jia, J. Li, Z.-H. Mao, M. Sun, Designing a wearable computer for lifestyle evaluation, in: Proc. of Northeast Bioengineering Conference (NEBEC), Philadelphia, USA, 2012, pp. 93–94.
- [6] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, K. Wood, SenseCam: A retrospective memory aid, in: Proc. of International Conference on Ubiquitous Computing (UbiComp), California, USA, 2006, pp. 177–193.
- [7] S. Hodges, E. Berry, K. Wood, SenseCam: A wearable camera that stimulates and rehabilitates autobiographical memory, *Memory* 19 (7) (2011) 685–696.
- [8] A. R. Doherty, A. F. Smeaton, Automatically segmenting lifelog data into events, in: Proc. International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Klagenfurt, Austria, 2008, pp. 20 – 23.
- [9] N. Caprani, N. E. O’Connor, C. Gurrin, Investigating older and younger peoples’ motivations for lifelogging with wearable cameras, in: Proceedings of IEEE International Symposium on Technology and Society (ISTAS), 2013.
- [10] K. M. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Fast unsupervised ego-action learning for first-person sports videos, in: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), Colorado, USA, 2011, pp. 3241–3248.
- [11] K. Zhan, S. Faux, F. Ramos, Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients, *Pervasive and Mobile Computing* 16, Part B (2015) 251–267.
- [12] H. Zhang, L. Li, W. Jia, J. D. Fernstrom, R. J. Sclabassi, Z.-H. Mao, M. Sun, Physical activity recognition based on motion in images acquired by a wearable camera, *Neurocomputing* 74 (12) (2011) 2184–2192.
- [13] Y. Poleg, A. Ephrat, S. Peleg, C. Arora, Compact CNN for indexing egocentric videos, in: Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV), New York, USA, 2016, pp. 1–9.

- [14] M. S. Ryoo, B. Rothrock, L. Matthies, Pooled motion features for first-person videos, in: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), Boston, USA, 2015, pp. 896–904.
- [15] A. Betancourt, P. Morerio, C. S. Regazzoni, M. Rauterberg, The evolution of first person vision methods: A survey, IEEE Transactions on Circuits and Systems for Video Technology 25 (5) (2015) 744–760.
- [16] Y. Nam, S. Rho, C. Lee, Physical activity recognition using multiple sensors embedded in a wearable device, ACM Transactions on Embedded Computing Systems 12 (2) (2013) 26:1–26:14.
- [17] C. Tan, H. Goh, V. Chandrasekhar, L. Li, J.-H. Lim, Understanding the nature of first-person videos: Characterization and classification using low-level features, in: Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), Ohio, USA, 2014, pp. 535–542.
- [18] Y. Poleg, C. Arora, S. Peleg, Temporal segmentation of egocentric videos, in: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), Ohio, USA, 2014, pp. 2537–2544.
- [19] K. Zhan, F. Ramos, S. Faux, Activity recognition from a wearable camera, in: Proc. of IEEE International Conference on Control Automation Robotics & Vision (ICARCV), Guangzhou, China, 2012, pp. 365 – 370.
- [20] H. Zhang, L. Li, W. Jia, J. D. Fernstrom, R. J. Scwabassi, M. Sun, Recognizing physical activity from ego-motion of a camera, in: Proc. of IEEE International Conference on Engineering in Medicine and Biology Society (EMBC), Buenos Aires, Argentina, 2010, pp. 5569–5572.
- [21] Y. Iwashita, A. Takamine, R. Kurazume, M. Ryoo, First-person animal activity recognition from egocentric videos, in: Proc. of International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 2014, pp. 4310–4315.
- [22] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, in: Proc. of International Conference on Learning Representations (ICLR 2014), Banff, Canada, 2014.
- [23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proc. of ACM International Conference on Multimedia, Florida, USA, 2014, pp. 675–678.
- [24] K. Zhan, V. Guizilini, F. Ramos, Dense motion segmentation for first-person activity recognition, in: Proc. of IEEE International Conference on Control Automation Robotics & Vision (ICARCV), Marina Bay Sands, Singapore, 2014, pp. 123–128.

- [25] K. Zhan, S. Faux, F. Ramos, Multi-scale conditional random fields for first-person activity recognition, in: Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom), Budapest, Hungary, 2014, pp. 51–59.
- [26] M. Ryoo, Human activity prediction: Early recognition of ongoing activities from streaming videos, in: Proc. of IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 1036–1043.
- [27] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proc. of European Conference on Computer Vision (ECCV), Crete, Greece, 2010, pp. 143–156.
- [28] P. H. Torr, A. Zisserman, Feature based methods for structure and motion estimation, in: Proc. of International Workshop on Vision Algorithms: Theory and Practice, Corfu, Greece, 1999, pp. 278–294.
- [29] D. Scaramuzza, F. Fraundorfer, Visual odometry, IEEE Robotics & Automation Magazine 18 (4) (2011) 80–92.
- [30] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, in: Proc. of IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 15–22.
- [31] J. Shi, C. Tomasi, Good features to track, in: Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), Seattle, USA, 1994, pp. 593 – 600.
- [32] A. Mannini, A. M. Sabatini, Machine learning methods for classifying human physical activity from on-body accelerometers, Sensors 10 (2) (2010) 1154–1175.
- [33] G. Bouchard, B. Triggs, The tradeoff between generative and discriminative classifiers, in: Proc. of International Symposium on Computational Statistics (COMPSTAT), Prague, Czech Republic, 2004, pp. 721–728.
- [34] C. Sutton, A. McCallum, An introduction to conditional random fields, arXiv preprint arXiv:1011.4088.
- [35] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. of the IEEE 77 (2) (1989) 257–286.
- [36] A. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes, Advances in neural information processing systems 14 (2002) 841.
- [37] T. Van Kasteren, G. Englebienne, B. J. Kröse, An activity monitoring system for elderly care using generative and discriminative models, Personal and ubiquitous computing 14 (6) (2010) 489–498.

- [38] I. Ulusoy, C. M. Bishop, Generative versus discriminative methods for object recognition, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, San Diego, USA, 2005, pp. 258–265.
- [39] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: Proc. of IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 2564 – 2571.
- [40] O. D. Lara, M. A. Labrador, A survey on human activity recognition using wearable sensors, *IEEE Communications Surveys & Tutorials* 15 (3) (2013) 1192–1209.
- [41] J. L. R. Ortiz, *Smartphone-Based Human Activity Recognition*, Springer, 2015.
- [42] D. Rodriguez-Martin, A. Sama, C. Perez-Lopez, A. Catala, J. Cabestany, A. Rodriguez-Moliner, SVM-based posture identification with a single waist-located triaxial accelerometer, *Expert Systems with Applications* 40 (18) (2013) 7203–7211.
- [43] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, in: Proc. of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), Ohio, USA, 2014, pp. 806–813.
- [44] S. Baker, I. Matthews, Lucas-Kanade 20 years on: A unifying framework, *International journal of computer vision* 56 (3) (2004) 221–255.
- [45] M. Bianchini, F. Scarselli, On the complexity of shallow and deep neural network classifiers., in: *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, 2014, pp. 23–25.
- [46] O. Chapelle, Training a support vector machine in the primal, *Neural computation* 19 (5) (2007) 1155–1178.
- [47] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, *ACM Computing Surveys (CSUR)* 46 (3) (2014) 1–33.