This is a post-peer-review, pre-copyedit version of an article published in Multimedia Tools and Applications.

The final authenticated version is available online at: https://doi.org/10.1007/s11042-020-08789-7.

Multi-modal Egocentric Activity Recognition using Multi-Kernel Learning

Mehmet Ali Arabacı · Fatih Özkan · Elif Surer · Peter Jančovič · Alptekin Temizel

Abstract Existing methods for egocentric activity recognition are mostly based on extracting motion characteristics from videos. On the other hand, ubiquity of wearable sensors allow acquisition of information from different sources. Although the increase in sensor diversity brings out the need for adaptive fusion, most of the studies use pre-determined weights for each source. In addition, there are a limited number of studies making use of optical, audio and wearable sensors. In this work, we propose a new framework that adaptively weighs the visual, audio and sensor features in relation to their discriminative abilities. For that purpose, multi-kernel learning (MKL) is used to fuse multi-modal features where the feature and kernel selection/weighing and recognition tasks are performed concurrently. Audio-visual information is used in association with the data acquired from wearable sensors since they hold information on different aspects of activities and help building better models. The proposed framework can be used with different modalities to improve the recognition accuracy and easily be extended with additional sensors. The results show that using multi-modal features with MKL outperforms the existing methods.

Keywords egocentric · first-person vision · activity recognition · multi-kernel learning · multi-modality

This work was partly supported by The Scientific and Technological Research Council of Turkey under TUBITAK BIDEB-2219 grant no 1059B191500048.

Mehmet Ali Arabacı · Fatih Ozkan · Elif Surer · Alptekin Temizel
Graduate School of Informatics, Middle East Technical University (METU), 06800, Ankara, Turkey
E-mail: (mehmet.arabaci, fatih.ozkan, elifs, atemizel)@metu.edu.tr

Peter Jančovič
Electronic, Electrical and Systems Engineering, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
E-mail: p.jancovic@bham.ac.uk
Alptekin Temizel was partly with Electronic, Electrical and Systems Engineering, University of Birmingham on sabbatical leave from METU during this work.

1 Introduction

Widespread use of wearable devices such as hand-held cameras, sports and action cameras (i.e., GoPro), mobile phones and accessories (i.e., Snap Spectacles, Google Glass) made it possible to track and analyze daily activities of individual users. It is known that physical inactivity increases the risk of cardiovascular diseases, diabetes, breast cancer, and even mental disorders such as depression [1]. For that reason, it is important to track the daily activities in an objective way by taking unbiased measurements. Egocentric activity recognition (EAR) systems allow tracking daily activities automatically and objectively. On the other hand, video-based EAR systems suffer from the storage of large amounts of video data. Even though the storage capacity and video compression techniques improve steadily, the resolutions of the videos are also increasing. Thus, developing effective EAR systems to organize and summarize videos is a crucial task. Activities of Daily Living (ADL) is another active research field for EAR that is used to assist caregivers to track the activities of elderly people by the help of rich sensor networks [2]. There are even recommendation systems that recommend music genres according to the type of activity [3].

In the literature, the definition of *activity* and *action* varies according to the application domain. Some of the works use activity and action interchangeably [4–6] while the others define activity (or complex activity) as a collection of multiple actions (or atomic activities) that are identified over temporal snippets using pattern-mining algorithms [7–9]. Egocentric datasets used in this study do not include complex activities, but consist of atomic ego-actions such as walking, running, shaking hand, hugging or petting. Accordingly, in this work, egocentric activities are distinguished among video segments without building a temporal relationship between actions and activities.

Vision-based methods for activity recognition have been mostly designed for third-person videos and cannot directly be applied to first-person videos (FPV). In contrast to third-person videos, in FPVs, the world is seen from the perspective of actors within the context of their activities and goals. In this case, the motion information (i.e., head movement for head-mounted cameras or body movement for chest-mounted cameras) of actors are as important as the motion of foreground objects in videos since the camera undergoes a large amount of ego-motion such as spinning and falling down according to the activity of the user. In addition, they are rapidly changing visual content due to changes in lighting, location (indoor, outdoor) and high ego-motion. Contrary to third-person videos taken with wider angles, egocentric perspective generally has a more limited perspective.

While the primary information source used in EAR is the FPVs, other types of wearable sensors such as audio, accelerometer, gyroscope and Global Positioning System (GPS) are also available [10]. One crucial problem here is to use these data collectively to improve the performance of egocentric activity analysis, i.e., fast querying through a myriad of egocentric videos taken on a daily basis. However, the tools provided by the manufacturers of wearable

devices are not sophisticated enough for practical use. Although the information obtained from all these sensor sources enables us to analyze egocentric activities with different modalities, this brings out the need for efficient algorithms that can effectively combine complementary information of sensors and provide modularity to allow easy incorporation of additional sensors.

The fusion of modalities may be done at feature [11–13] or classifier level [14, 15]. In feature level fusion, different types of features are combined to get more discriminative features before the final classification. On the other hand, classifier (or decision) level fusion techniques use each individual feature independently in the classification process. The final decision is taken by combining decisions for individual features. In our work, different sets of features are extracted using visual, audio and wearable sensor information and are fused using two MKL techniques (MKBoost [16] and SimpleMKL [17]). SimpleMKL uses decision level fusion in order to select the kernel weights with a weighted 2-norm regularization that encourages sparse kernel combinations. On the other hand, MKBoost combines a boosting approach with MKL learning that allows performing feature selection and decision level fusion concurrently.

Related studies on EAR can also be grouped based on their sensor types: using only visual information [4, 18, 19], using mobile and wearable sensor networks [6, 20, 21] and combining visual information with other wearable sensors [11, 22–24].

In this study, we propose an EAR system that fuses multi-modal features obtained from optical, audio and wearable sensor data. The proposed framework can adaptively weigh features and offer expandability with new features and modalities. Our motivation and contributions for this study are highlighted below:

- The proposed framework uses audio together with video and wearable sensors to recognize egocentric activities. To the best of our knowledge, this is the first study that uses audio sensor in association with video and other mobile sensors for this particular problem.
- We propose an adaptive framework that weighs the features from different modalities and can adapt to different scenarios having different activity classes by varying the relative importance of the features and selecting appropriate kernel types.
- MKL provides an efficient fusion procedure that maximizes the recognition performances even if one or more sensor information is unavailable.
- The results have shown that the proposed framework achieves on par or better results with respect to the state-of-the-art methods.

The following section summarizes the works related with EAR which are grouped with respect to their approaches to the problem and fusion/classification strategies.

2 Related Studies

Since the main concern of this study is to recognize activities by fusing multimodal features with the help of MKL, we mainly focused on the methods using different modalities as input and using MKL for EAR tasks.

2.1 Egocentric Activity Recognition

Previous studies on EAR can be broadly categorized based on the sensor types they use. Sensor modalities include video-based, mobile phone and wearable sensors, social network sensors and wireless signals [10]. Vision-based activity recognition algorithms are mostly focused on the analyses of third-person videos [25–28]. On the other hand, vision-based EAR can be grouped as object and motion-based approaches [29]. In object-based methods, activity recognition is performed using the object(s) detected in videos (i.e., detection of cheese and bread objects imply "making cheese sandwich" activity) [30] that makes them dependent to the availability of objects in particular actions (hence can only be used to detect actions involving particular objects) and becomes directly related to the object recognition performance which is vulnerable to occlusions. Motion-based approaches make use of the assumption that different types of activities such as running, walking, stair climbing, and writing involve different body motions, and these motion patterns can be used for recognizing activities [18, 31].

Other group of studies make use of wide range of different sensors embedded on mobile and wearable devices in order to recognize the activities of the users. Each sensor type provides a different aspect of information on activities. For example, motion sensors can be used to monitor the users' movements to detect motion patterns of different activities such as walking, standing or running. Other types of sensors can also be used to obtain this motion pattern such as accelerometer, gyroscope, magnetometer and inertial measurement units (IMUs). In [1], different physical activities were distinguished in an unsupervised way using smartphone accelerometers. Additionally, the information taken from proximity and light sensors give clues whether the actor is in a dark place or in a place where there is light [32]. Pedometer sensors or specialized wearable devices which count steps, monitor heart rate or pulse may also give valuable information to understand the health conditions of users [33]. However, many studies in this field require extensive heuristic knowledge to develop and select appropriate features for a given human activity recognition (HAR) system [10].

There are a limited number of studies which deal with the problem of fusing optical and wearable sensor information for EAR. In [22], a head mounted camera and an eye tracker were used to recognize objects from videos that the actor interacted with, which are then used to recognize the egocentric activities. Similarly, in [23], EAR was performed by a camera with a wearable eye tracker to obtain gaze measurements in which the point of gaze is represented

by a 2-D image point in each frame. Recently, a multi-modal solution was presented that combines new sensor features with dense trajectory features [24] and was applied to their publicly available dataset (Multi-modal Egocentric Activity Dataset) [11]. Additionally, in an early study, audio was considered with optical information to investigate the recognition of user activities from a wearable camera and a microphone [34]. However, there is a lack of generic solution combining audio features with optical and wearable sensors to solve the problem of EAR.

Deep learning based methods can be used as an alternative for various subtasks in relation to activity recognition; such as feature extraction [35–37], kernel fusion [38] and exploration of human-object interactions [39]. In a recent study [37], auto-encoders were employed that take raw input data to extract appearance and motion features, then reconstruct the input data through its decoding procedure. After that, the learned appearance and motion features by auto-encoders are fused to accomplish an egocentric activity representation which can be fed into any supervised learning model. One of the prominent studies [31] proposed a 3D CNN architecture for long-term activity recognition in egocentric videos by generalizing the concept of temporal filtering [40] that takes sparse optical flow volume as input. In another study [41], a twin stream network architecture was used where one stream analyzes appearance and the other stream analyzes motion information by explicitly training the network to segment hands and localize objects to recognize egocentric activities. A recent study [42] proposed a two-stream convolutional neural network architecture that uses long-term fusion pooling operators to capture the temporal structure of actions by leveraging a series of frame-wise features of both appearance and motion in actions.

Deep learning based methods were also used for mobile and wearable sensor data in time-series format. These types of works generally focus on feature representation of sensor information to improve recognition accuracy [10]. For that purpose, sensor streams are converted by using channel [20] or model [43] approaches to fit into deep learning algorithms. In the literature, sensor-based deep network solutions for HAR use generative (Restricted Boltzman Machines [44], Deep Autoencoders [45]), discriminative (Convolutional Neural Networks [46], Recurrent Neural Networks [47]) or hybrid (Convolutional Recurrent Neural Networks [48]) models.

Even if deep learning based methods have satisfactory results, they are not practical when new features need to be added since the network architecture needs to be changed and retrained. Additionally, while deep learning based methods offer transferability of knowledge across different tasks [49], they still require further research to outperform traditional approaches in FPV [50].

2.2 Kernel Learning

Support vector machines (SVMs) is one of the most popular kernel-based techniques in machine learning in which a single kernel is employed to transform

input data usually into a high-dimensional space to perform classification or regression. However, single kernel learning does not provide a mechanism for effective use of multiple features. In standard SVM, one way to use multiple features is feeding each feature independently to produce a classification result. This does not take the relative importance of the features into account for classification and requires a decision level fusion mechanism. Another way is to concatenate features and then use a single SVM for classification. Again, this does not allow taking the relative importance of different features into account as they are concatenated into a single vector. Therefore, an effective classification algorithm should simultaneously take the varying importance of features into account and allow using different kernels for different features. One of the pioneering studies that takes into account the varying importance of features was proposed in [4] in which each feature extracted locally or globally was considered as a separate channel having equal weights. On the other hand, it is possible to have an architecture using multi-kernel learning (MKL) [51] which allows adaptive kernel selection and weighing. In this data-driven approach, multiple features are fused in an adaptive way using different types of kernels. Even if the base kernels cannot perform well for all features, their parameters and weights are optimized to get the best performance by using complementary information coming from different sources. By this way, input features are dynamically weighed at the training stage that allows to create adaptive solutions for different first-person activity recognition problems. As a result, feature selection is also performed automatically by the selection of kernels and their weights during the training phase.

In this work, MKL is adapted to the problem of feature selection and decision fusion in EAR. Different to the prior work in literature, the proposed method is a multi-modal approach that uses audio and sensor features in addition to the visual features. Accordingly, different sources of information are used to extract multi-modal features such as optical flow, intensity gradient, video-based inertia, audio, accelerometer, gyroscope, linear acceleration, magnetic field and rotation vector. Adaptive weighing of features in training phase according to their classification performances on weak classifiers makes the framework robust against irrelevant data.

3 Proposed Framework

The proposed MKL-based multi-modal framework is shown in Figure 1. Firstly, features are extracted from each raw sensor data. Then, each extracted feature is used as an input to the MKL algorithm in which feature and base kernel (κ) selection for classification are performed concurrently by adjusting the kernel weights (d) adaptively. After training MKL, the best feature and kernel combinations are selected using base learners (f). Finally, EAR is performed using the trained model for test videos with previously selected features and base kernels.

In the subsequent section, the visual, audio and sensor features used in this study are introduced. The single and multi-kernel learning strategies are discussed in the following section.

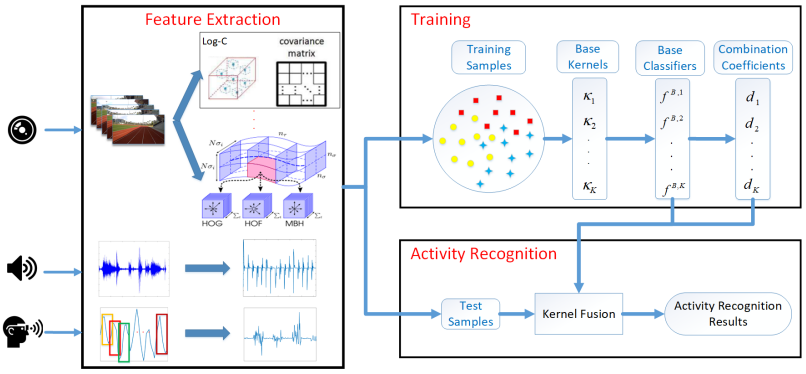


Fig. 1 The proposed solution for EAR using visual, audio and sensor features.

3.1 Feature Extraction

In this work, various types of features were extracted using visual, audio and sensor information. Three important criteria were taken into consideration when selecting the features. One of them is to satisfy the diversity of features that may hold complementary information about egocentric activities. Secondly, since our main concern is not to propose a new feature, existing visual, audio and sensor features were mostly preferred that had effectively been applied to EAR problem before. Finally, in order to compare the effectiveness of the proposed learning strategy, similar set of features was selected with the other state-of-the-art methods using the same datasets. The following section explains the details of all visual, audio and sensor features used in this study. The selection of feature sets for different datasets is discussed in Section 4.

3.1.1 Visual Features

Since it is known that effective encoding of ego-motion is crucial for EAR systems [18, 31], we preferred a set of visual features (GOFF, VIF, HOG, HOF and MBH) that holds motion patterns globally and locally in temporal dimension. Grid Optical Flow-based Features (GOFF) include motion-based video features extracted from spatio-temporal information specifically designed for FPVs [18]. Virtual Inertia Features (VIF) [18] are used to approximate inertia data (velocity and acceleration) to model egocentric activities. Log-Covariance (Log-C) [52] features are dense video features derived from the optical flow data as well as intensity gradient. Cuboid feature extracts local information using sparse 3D space-time data [53]. Lastly, dense trajectory features are composed of a set of visual features (Trajectory, Histogram of Oriented Gradients (HOG), Histogram of Optical Flows (HOF) and Motion Boundary Histograms (MBH) that propose an effective solution for motion-related vision tasks by detecting motion patterns over densely tracked sample points using optical flow fields.

3.1.1.1 Grid Optical Flow-based Features (GOFF)

GOFF are used to model the discriminative motion patterns within optical flow information such as magnitude, direction and frequency [18]. To be able to discriminate motion characteristics of activities, a set of features is defined using the video frames divided into grids such as Motion Magnitude Histogram Features (MMHF), Motion Direction Histogram Features (MDHF), Motion Direction Histogram Standard-Deviation Feature (MDHSF), Fourier Transform of Motion Direction Access Frame (FTMAF) and Fourier Transform of Grid Motion Per-Frame (FTMPF).

MMHF is the histogram representation of grid optical flow magnitude values in which a non-uniform quantization process with 15 levels is used [18]. MDHF is another histogram representation of grid optical flow considering its quantized direction values. MDHF was uniformly quantized into 36 levels that correspond to 10^{o} between each level. MDHSF represents the standard deviation of each direction bin across the temporal dimension with a vector of size 36. FTMAF is a frequency-based feature that measures the variation for each direction bin along temporal dimension using decomposed frequency bands. In contrast to MDHSF, FTMAF quantifies the detailed dynamics of motion direction into 25 levels. Lastly, FTMPF measures the variation of grid optical flow within a frame that also has 25 levels. As a result, GOFF has a feature vector of size 137 after concatenating all of the sub-features.

3.1.1.2 Virtual Inertia Feature (VIF)

VIF provides virtual inertial information derived by using intensity centroid across frames in a video without physically using inertial sensors [18]. Three different sub-features were extracted in temporal dimension: zero-crossing (ZC), 4MEKS and frequency-domain feature (FF). ZC uses velocity and acceleration values generated from intensity centroid for each frame and measures zero-crossing rates of velocity and acceleration values. 4MEKS represents the time-domain features in which minimum, maximum, median, energy, kurtosis, mean and standard deviation values are calculated for each inertial signal. FF feature holds low frequency components of the variations in velocity and acceleration. In this study, the number of frequency components was selected as 10. Similar to GOFF, all sub-features of VIF were concatenated that makes the resulting feature vector size as 106.

3.1.1.3 Log Covariance (Log-C)

Feature covariance matrix is an effective way of representing dense set of localized features. Bag of local features can be represented in a lower dimension by the help of feature covariance matrices. In this study, feature covariance matrix was determined by using optical flow and gradient vectors. For each pixel of a video frame, a 12x12 dimensional covariance matrix was calculated using intensity gradient of raw video sequences with respect to temporal direction and first-order partial derivative of optical flow with respect to spatial x and y directions, spatial divergence, vorticity, gradient tensor and the rate of strain tensor [52].

The dimension of the covariance matrix is only related to the dimension of the feature vectors (i.e., 12x12 in this study). Covariance matrices lie on the Riemannian manifold and matrix logarithm [54] were used to convert manifold of covariance matrices into Euclidean. As a result, the feature vector size was reduced to 78 due to its symmetry. After that, the extracted feature vector was normalized by standard deviation and clustered using k-means for each video segment. Finally, a descriptor is defined using Bag-of-Visual-Words (BoVW) for each single activity video. The dictionary size was set to 300 which gave the best single feature classification performance through a pre-defined set of cluster sizes. However, principal component analysis (PCA) was applied to the descriptor in order to reduce the dimension of sparse BoVW vectors except the classifiers using histogram intersection kernels which will be explained in the following section.

3.1.1.4 Cuboid

In addition to the global video features, a sparse 3D XYT space-time feature, cuboid [53] is used as a local video feature. Cuboids have been used successfully for activity recognition problem before [27]. Cuboid feature was developed as an alternative to 2D interest point detectors to take temporal dimension into account in addition to the spatial dimensions.

Before the feature extraction process, interest point detector was employed to detect the corners in spatio-temporal dimensions by responding strongly to the local areas containing motion and including spatio-temporal corners. After that, a cuboid feature was extracted at each interest point that includes brightness gradient and optical flow information [53]. Similar to Log-C, cuboid was also configured to generate descriptors by using BoVW. The size of the histogram was set to 500 through a set of cluster sizes by considering their single feature classification performances. PCA was applied in order to reduce the dimension except for histogram intersection kernels.

3.1.1.5 Dense Trajectory Features

Dense trajectories provide an effective solution for motion-related vision tasks by detecting motion patterns over densely tracked sample points using optical flow fields. They were also used to model ego-motion in egocentric videos [11] that are composed of a set of visual features namely; trajectory, HOG, HOF and MBH. Trajectory information is simply the concatenation of normalized displacement vectors. HOG focuses on static appearance information while HOF and MBH provide a measure of motion information in videos.

After extracting the dense trajectory descriptors as in [28], Fisher vector was employed to encode these descriptors through an estimated Gaussian Mixture Model (GMM). Similar to [11], the cluster size for GMM was set to 25 and 1% of descriptors were randomly sampled to estimate the GMM for building the codebook. The dimensions of the features are reduced by half using PCA. Finally, power and L2 normalization were performed on Fisher vectors.

3.1.2 Audio Features

In order to fuse video, audio and sensor modalities using the SVM and MKL-based frameworks, an utterance of audio recording of an activity needs to be mapped into a vector space. To do so, a commonly used methodology in the field of speaker recognition from speech [55, 56] was employed in this study.

The first step is to split an audio signal into a sequence of frames and represent each frame using a spectrum-based feature vector. For that purpose, Mel-frequency cepstral coefficients (MFCCs) [57] were used as frame features. The discrete Fourier transform was applied on each frame and the resulting magnitude spectrum was passed to a bank of Mel-spaced triangular filters and then discrete cosine transform was applied, providing MFCCs. We explored a range of values for the parameters, with a final setup as follows: frame length of 40 ms, 10 ms shift between adjacent frames and 23 filter-bank channels. Only the first 12 MFCCs were used. The frame energy was added as the 13^{th} feature. These features were appended with their temporal derivatives, referred as delta and delta-delta coefficients, calculated as in [58], using the span of +3 and +2 frames, respectively. This resulted in a 39 dimensional feature representation of each signal frame.

The distribution of these feature vectors was modeled using the Gaussian mixture model (GMM), with diagonal covariance matrices. First, a class-independent model, referred to as the Universal Background Model (UBM), was estimated using all of the training data from all classes. A class-dependent GMM was then obtained by performing maximum a-posteriori adaptation [59] of the component mean vectors of the UBM, using class-specific training data. The mean vectors of the components of the resulting class-dependent GMM are then concatenated to form a 'supervector' [60]. A supervector is obtained for each utterance of each class, resulting in a set of supervectors per class. Supervectors are then used as a vector representation of each class for activity classification. It was observed that using different numbers of GMM components ranging from 16 to 64 gave similar performances. Hence, the number of components is set to 16 throughout the experiments. Lastly, dimensionality reduction of the supervectors was tested using the PCA. However, applying PCA to the supervectors had no significant effect on the performance.

3.1.3 Sensor Features

The wearable sensors data (accelerometer, gravity, gyroscope, linear acceleration, magnetic field and rotation vector) contain time-series information having 19 dimensions. In this study, each dimension of sensor information was considered as a one-dimensional signal and converted into trajectories by employing sliding windows. After generating many trajectories for sensor data, Fisher encoding was performed similar to dense trajectory features. The same feature extraction procedure and parameter settings were followed as in [11].

3.2 Activity Recognition

Support vector machines (SVMs) was chosen as the baseline method that represents single kernel learning and it was compared with two well-known MKL algorithms: MKBoost and SimpleMKL. MKBoost adopts boosting to solve a variant of MKL problem, which avoids solving the complicated optimization tasks [16] while SimpleMKL proved to be an efficient and rapid converging algorithm compared to other MKL optimization algorithms [61]. In the following section, particulars of these methods in relation to the EAR problem are explained briefly.

3.2.1 Single Kernel Learning

SVM is a kernel-based method and use of kernels allows operating in higher dimensional feature spaces than the original. In this work, SVM was used for single kernel learning with the features formed by concatenating all individual features. The most widely used kernel types for SVMs are linear, polynomial and radial basis functions (RBFs). After making numerous experiments, linear and polynomial kernel (3^{rd} order) gave the best recognition performances for two pre-defined feature sets used for the selected egocentric datasets which are defined in Section 4.

The linear and polynomial kernels used in this study are defined as:

$$\begin{aligned} \kappa(x_i, x_j) &= \langle x_i, x_j \rangle \\ \kappa(x_i, x_j) &= (\langle x_i, x_j \rangle + l)^p \end{aligned} \tag{1}$$

where κ represents kernel function, x's are the features, p is the maximal order of monomials making up the new feature space and l is a bias towards lower order monomial. The intuition behind this kernel definition is that it is often useful to construct new features as products of original features [62].

One of the feature set includes histogram-based features due to the use of BoVW model for Log-C and cuboid. Therefore, in addition to polynomial kernels, a modified version of histogram intersection kernel (DC-Int) [5] was chosen that was specifically designed for histogram-based features:

$$\kappa(x_i, x_j) = \exp(-\sum_{c=1}^{C} D(H_i^c, H_j^c)) \tag{2}$$

where C is the number of channels, H_i^c and H_j^c are W dimensional histograms of c^{th} channel for i^{th} and j^{th} videos and $D(H_i^c, H_j^c)$ is the histogram distance defined as:

$$D(H_i^c, H_j^c) = 1 - (\sum_{m=1}^{W} min(h_{im}, h_{jm}) / \sum_{m=1}^{W} max(h_{im}, h_{jm})) \tag{3}$$

where h_{im} and h_{jm} are the m^{th} histogram bins identified for i^{th} and j^{th} videos, respectively.

3.2.2 Multi-Kernel Learning

SVM has been proven to be an effective method in classification and regression problems [63] in which data representation is implicitly chosen through the used kernels $\kappa(x, x_i)$. MKL converts the single kernel solution to the weighted sums of multiple kernels as in the form below:

$$\sum_{i=1}^{N} \alpha_i^* \kappa(x, x_i) + b \tag{4}$$

where N is the number of samples, α_i^* and b are coefficients to be learned from examples, while $\kappa(.,.)$ is a given positive definite kernel associated with a reproducing kernel Hilbert space. It was shown that using multiple kernels (κ_k) can enhance the interpretability of the decision function and improve performance [64],

$$\kappa(x, x^i) = \sum_{k=1}^{K} d_k \kappa_k(x, x_i) \tag{5}$$

where K is the number of kernels, $d_k \geq 0$ and $\sum_{k=1}^{K} d_k = 1$.

3.2.2.1 Multiple Kernel Boosting (MKBoost)

MKBoost employs a boosting framework in order to learn an ensemble of multiple base kernel classifiers, each of which is learned from a single kernel. The combination weights for both the kernels and classifiers can be efficiently determined through the learning process of boosting [16] using a similar procedure to Adaboost [65]. In this approach, some kernel classifiers (κ) with multiple kernels (K) through a series of boosting trials $t = 1, ..., T$, where T denotes the total number of boosting trials, are repeatedly learned using a subset of M examples $(r * M$ where $0 < r < 1)$.

The original procedure of MKBoost algorithm was designed for binary classification problem. However, it has been adapted to multi-class problems in our framework. For that purpose, multi-class classifiers are used to perform classification task at each trial and samples are boosted within each class in order to preserve the balance between the classes. Additionally, the feature selection routine has been modified by taking all feature combinations (P) into consideration at each trial (i.e., 7 combinations for 3 features). The pseudocode of this procedure is given in Algorithm 1.

At each boosting trial, distribution of weights χ_t, which indicates the relative importance of the training examples for learning, is updated. Additionally, the weights of the incorrectly classified examples are increased while the weights of those correctly classified examples are decreased in order to focus on those examples that are hard to be successfully classified.

Lastly, the selected kernels and the feature combinations with their corresponding weights are used at the test phase. For each test sample, the weighted sum of the kernel predictions are used for the final prediction of test samples.

Algorithm 1 MKBoost

Input: (x, y), κ, T
Output: $\hat{y}$

training set (S_{train}): $(x_1, y_1), ..., (x_M, y_M)$
test set (S_{test}): $(x_{M+1}, y_{M+1}), ..., (x_N, y_N)$
labels: $y = 1, ..., L$
feature combinations: $p = 1, ..., P$
kernel pool: $\kappa_k(.,.) : XxX \rightarrow \mathbb{R}$ where $k = 1, ..., K$
Training Phase
for $t \leftarrow 1$ to T **do**
    Select $r * M$ sample indices (i_t) using distribution χ_t where $0 < r < 1$
    for $p \leftarrow 1$ to P **do**
        Select p^{th} feature combination for training: $S^p_{train}[i_t]$
        for $k \leftarrow 1$ to K **do**
            Train weak classifier (κ_k) with $S^p_{train}(i_t)$
            Compute training error over all samples (S^p_{train}): $\in^k_p = \frac{1}{M} \sum_{m=1}^{M} f_k(x^p_m) \neq y_i$
        Select the best classifier for p^{th} feature combination : $\in^p = arg\min_k \in^k_p$
    Select the best classifier (f_t) and feature combination (p_t) for trial : $\in_t = arg\min_p \in^p$
    Set the weight for trial: $W_t = \frac{1}{2} \ln(\frac{1-\in_t}{\in_t})$
    Update sample distribution: $\chi_{t+1}(i) = \chi_t(i) \begin{cases} e^{-W_t} & \text{if } \kappa_t(x_i) = y_i \\ e^{W_t} & \text{if } \kappa_t(x_i) \neq y_i \end{cases}$ for $i = 1, ..., M$
    $\chi_{t+1} = \frac{\chi_{t+1}}{Z_t}$ where Z_t is a normalization factor to make χ_{t+1} a distribution
Test Phase
for $i \leftarrow M + 1$ to N **do**
    $\hat{Y}_c \leftarrow 0$ where $c = 1, ..., C$
    for $t \leftarrow 1$ to T **do**
        Predict the label for trial: $c_t = f_t(S^{p_t}_{test}[i])$
        $\hat{Y}[c_t] \leftarrow \hat{Y}[c_t] + W_t * c_t$
    Predict the final label: $\hat{y_i} = arg\max_c \hat{Y}_c$

3.2.2.2 SimpleMKL

SimpleMKL [17] offers a solution to MKL by using a weighted $l2$ normalization. The proposed solution is based on a gradient descent wrapping standard SVM solver that determines the combination of kernels [40].

As it was stated in [51], SimpleMKL consists of two main steps: solving a canonical SVM optimization problem with the given kernel weights (d) and updating kernel weights using the following gradient calculated with another parameter (γ) obtained in the first step. Additionally, the gradient update procedure must consider the non-negativity and normalization properties of the kernel weights [51]. In this algorithm, K is the number of kernels, κ is the base kernel, d_k is the kernel weights, J represents the differentiable objective function and ∇_D shows the gradient descent directions for each step. The pseudo-code of SimpleMKL is given in Algorithm 2.

Algorithm 2 SimpleMKL

Input: d_k

Output: κ_k

$d_k \leftarrow \frac{1}{K}$ for i=1,...,K
while stopping criterion not met **do**
  Compute $J(d)$ by using an SVM solver with $\kappa = \sum_k d_k \kappa_k$
  Compute $\frac{\partial J}{\partial d_k}$ for $k = 1, ..., K$ and descent direction ∇_D
  Set $\mu = argmax(d_k)$, $J^\dagger = 0$, $d^\dagger = 0$, $\nabla_D^\dagger = \nabla_D$
  while $J^\dagger < J(d)$ **do**
    Set kernel weights and gradient descent: $d = d^\dagger$, $\nabla_D = \nabla_D^\dagger$
    $v = arg \min_{(k|\nabla_D^k<0)} (-\frac{d_k}{\nabla_D^k})$, $\gamma_{max} = -\frac{d_v}{\nabla_D^v}$
    Update parameters: $d^\dagger = d + \gamma_{max}\nabla_D$, $\nabla_D^{\mu\dagger} = \nabla_D^\mu - \nabla_D^v$, $\nabla_D^{v\dagger} = 0$
    Compute $J^\dagger$ by using an SVM solver with $\kappa = \sum_k d_k^\dagger \kappa_k$
  Linear search along D for $\gamma \in [0, \gamma_{max}]$ (calls an SVM solver for each trial value)
  $d \leftarrow d + \gamma \nabla_D$

4 Experimental Results

In this section, experimental results of the proposed solution are presented after introducing the datasets used in this study, the strategy of feature and kernel selection and the metrics to evaluate the performance of activity recognition.

4.1 Datasets

The performance of the proposed framework was evaluated using three egocentric datasets: JPL First-Person Interaction [4], DogCentric Activity Dataset (DogC) [5], and Multi-modal Egocentric Activity Dataset (MEAD) [6]. We have taken the diversity of activities into consideration when selecting the datasets. For instance, the videos in JPL were taken indoor with a passive actor while DogC includes videos taken outdoor with a first-person animal viewpoint. On the other hand, MEAD includes videos taken in different places (indoor, outdoor) and at different times of the day with human actors.

JPL [4] is composed of first-person videos of interaction-level activities by 8 actors. It contains four positive (i.e., friendly) interactions with the observer (shaking hand, hugging, pet, waving hand), one neutral interaction (pointing), and two negative (i.e., hostile) interactions (punching, throwing objects) for each actor. There are a total of 84 videos with 320x240 resolution at 30fps. The clips have variable lengths and the mean video length for JPL is 7.77 seconds.

DogC [5] includes 10 different types of activities taken from the viewpoint of the dogs. Video resolutions are 320x240 at 24fps or 48fps and mean clip length is 4.12 seconds. Playing with a ball, drinking, feeding, looking left/right, petting and shaking are some of the activity types. Unlike the other datasets, the number of videos for each activity is different (i.e., feed and shake have 25 videos while playing with a ball has only 14 videos) which makes it unbalanced with respect to its class samples.

MEAD [6] is different from other egocentric activity datasets as it has multi-modal sensor data (optic, audio, accelerometer, gravity, gyroscope, linear acceleration, magnetic field and rotation vector). It has 20 distinct life-logging activities grouped into four top level types: ambulation, daily activities, office work and exercise. Each category has 10 sequences and each clip is exactly 15 seconds. There is a total of 200 videos with 1280x720 resolution at 29.9fps. Audio was sampled at 48 kHz and 16 bits per sample. As no significant content was observed in higher frequencies, the audio was down-sampled to 24 kHz before the feature extraction.

4.2 Feature and Kernel Selection

The feature sets were selected based on the available sensor information in datasets, the diversity of features (i.e., GOFF holds the global motion in frames while VIF presents video-based inertia information) and the features used in the other state-of-the-art methods. For JPL and DogC, only visual features were used since they only have visual sensor information. Additionally, GOFF and VIF were firstly used in [18] with JPL, cuboid was proposed in [4] for JPL and, cuboid and Log-C were used in [19] for both JPL and DogC. As a result, the feature set was selected as a combination of GOFF, VIF, Log-C and cuboid to show that the proposed method can effectively combine these visual features. On the other hand, MEAD contains visual, audio and wearable sensor information. In [6], dense trajectory and sensor features were extracted from MEAD, these features are also selected to be used in this work. However, unlike in [6], GMM-based supervector audio features were extracted, obtained by modelling MFCCs to prove that it is possible to add another modality to the proposed MKL-based framework and improve the overall recognition performance. As a result, the proposed framework was evaluated in two settings: one of which uses visual features (GOFF, Log-C, cuboid) and virtual inertia (VIF) extracted from JPL and DogC, and the other employs visual (dense trajectory), audio and sensor (FVS) features extracted from MEAD.

The kernel types used for single kernel learning differ with respect to the selected feature set. For instance, polynomial kernel (3^{rd} order) was selected for the first experimental setting including GOFF, VIF, Log-C and cuboid while linear kernel was chosen for the second feature set that is composed of FVS, audio and dense trajectory features. The selection of different kernel types is directly related to the characteristics of the features. GOFF, VIF, Log-C and cuboid hold the information in a compact way that generally requires non-linear decision boundaries. However, FVS, audio and dense trajectory features were encoded with Fisher vectors which include sparse vectors that can generally be separated with linear kernels.

In our experiments, each video segment was represented with one feature vector. Therefore, the number of samples is equal to the number of videos in datasets. Training and test sets were randomly composed at each iteration and the final evaluation results were obtained by taking the average of 100 test iterations. For JPL, each activity has 9 training and 3 test videos while MEAD has 8 training and 2 test video samples. Unlike JPL and MEAD, DogC

has a different number of video segments for each activity, approximately 75% of which are selected randomly for training.

A pre-defined kernel pool should be defined before running the experiments of MKL algorithms. However, searching for the best kernel set for each feature combinations is time consuming. Thus, only one set of basis kernels was selected for each dataset and all the experiments for that dataset were performed with the same basis kernel pool.

4.3 Evaluation Metrics

In order to evaluate the results, Precision (P), Recall (R), Accuracy (A) and F1-score (F) metrics by considering True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) scores were used as shown in (6).

$$\begin{aligned} P &= \frac{TP}{TP+FP} & R &= \frac{TP}{TP+FN} \\ A &= \frac{TP+TN}{TP+TN+FP+FN} & F &= 2\frac{P*R}{P+R} \end{aligned} \tag{6}$$

Kappa statistics was selected as another performance measurement metric which is known to be a discerning statistical tool for assessing the classification accuracy of different classifiers that generally gives better interclass discrimination than the overall accuracy [66]. Kappa statistics is calculated by using the marginal probabilities of ground truth and predicts labels with their joint probabilities that correspond to the values of confusion matrix. The formulation of Kappa statistics is given below:

$$p_0 = \sum_{i=1}^{L} p_{ii} \qquad p_e = \sum_{i=1}^{L} p_i * p_{\hat{i}} \qquad \hat{\kappa} = \frac{p_0 - p_e}{1 - p_e} \tag{7}$$

where L is the number of classes, p_i is the probability of i^{th} class according to the ground truth, $p_{\hat{i}}$ is the probability of i^{th} class according to the prediction, p_0 is the observed accuracy and p_e is the sum of the marginal proportions.

Implementation of the proposed framework was mainly realized using MATLAB 2018b while OpenCV 3.2.0 was employed for optical flow estimation. Additionally, the following third-party toolboxes were used: LibSVM 0.9.20 [67], the toolbox for cuboid extraction developed by Dollar et al. [53] and SimpleMKL toolbox [51]. LibSVM and SimpleMKL toolbox have been modified to allow using histogram intersection kernels.

The test procedure was the same for all datasets. Firstly, the performances of individual features were analyzed for each dataset. Then, the combinations of features were tested to observe the effects of feature types on the recognition scores. The results were tabulated for each dataset individually. Finally, the performance of the proposed method was evaluated by using all the features and compared with the performance of the-state-of-art methods.

4.4 Results for JPL Dataset

Average F1-scores for single features and their combinations for JPL dataset are shown in Table 1. According to the results, SimpleMKL performs better for optical flow-based features while histogram intersection kernel has better results for the histogram-based features (Log-C and cuboid). Another significant point is the inferior performance of the local visual feature relative to the global features.

Table 1 F1-Score of Single Features and Feature Combinations for JPL

		SVM-Poly	**SVM-Hist**	**MKBoost**	**SimpleMKL**
Single Features					
Global	GOFF	0.91	0.91	**0.92**	**0.92**
Global	VIF	0.85	0.86	0.86	**0.87**
Global	Log-C	0.80	**0.84**	0.82	0.82
Local	Cuboid	0.69	**0.71**	0.70	0.70
Combination of Global Features					
GOFF + VIF		0.92	0.92	**0.93**	**0.93**
GOFF + Log-C		0.90	0.87	**0.93**	0.92
VIF + Log-C		0.84	**0.86**	0.85	0.85
GOFF + VIF + Log-C		0.91	0.89	**0.93**	**0.93**
Combination of Global and Local Features					
Global + Local		0.92	0.92	**0.93**	**0.93**

The videos in JPL were taken with a passive actor and video segments are not cluttered with foreground objects (only one or two persons appear in the videos). There are two typical motion characteristics in videos, one of which is global camera motion (i.e., punching, hugging) and the other is local motion appearing only in one region within FOV (i.e., throwing objects, pointing) while the other regions do not contain any motion information. These motion characteristics work in favor of the global features as can be seen from the results.

When a number of global features are used together, the combinations including GOFF consistently got the highest scores. This is expected since GOFF is the most discriminative feature according to the single feature performances. On the other hand, when all features (global and local) are used, MKBoost and SimpleMKL algorithms perform better than SVM-based classifiers.

The resulting confusion matrices for JPL activities using the selected classifiers are shown in Figure 2. According to these results, pet and point activities have the least recognition performances compared to other activities. Additionally, MKBoost achieves a more balanced recognition performance that makes it more reliable compared to the others.

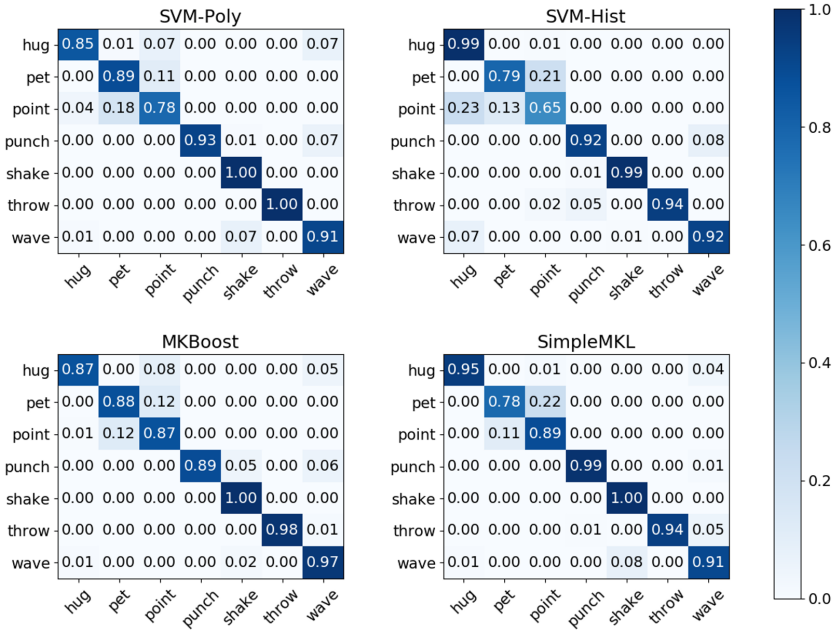


Fig. 2 Confusion matrices of SVM, Histogram Intersection, MKBoost and SimpleMKL learning methods for JPL.

4.5 Results for DogC Dataset

DogC is an unbalanced dataset that was taken outdoor with the viewpoints of animals. Additionally, ego-motion is much higher compared to JPL and MEAD that makes it the most challenging dataset. Table 2 shows the average F1-scores of both single feature performances and their combinations for DogC dataset.

Unlike JPL, the performance gap between global and local visual features is relatively small which implies that the local motion characteristics are also important in addition to the global motion. Additionally, MKBoost and SimpleMKL have very similar scores for global feature combinations. However, SimpleMKL achieves the best results for the combination of global and local features.

The resulting confusion matrices of DogC for different learning methods are shown in Figure 3. “Looking left” and “looking right” activities have the lowest classification accuracies since they are confused with each other. These two activities have very similar characteristics apart from having different motion directions.

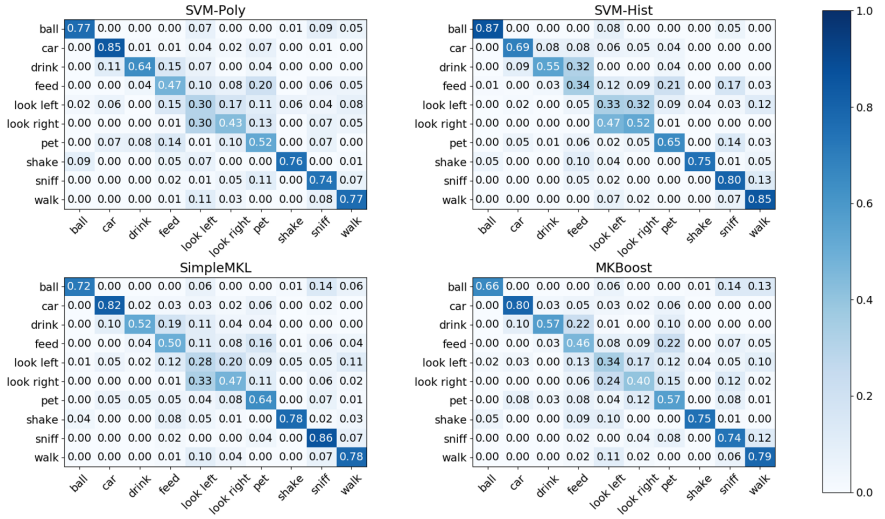


Fig. 3 Confusion matrices of SVM, Histogram Intersection, MKBoost and SimpleMKL learning methods for DogC.

4.6 Results for MEAD Dataset

MEAD has a relatively higher number of egocentric activity classes and is more challenging in that sense. Additionally, the videos in this dataset exhibit a higher variation due to being captured at different times of the day and different locations (indoor, outdoor). In addition, multi-modal sensor information is available for MEAD including visual, audio and wearable sensors.

Table 2 F1-Score of Single Features and Feature Combinations for DogC

		SVM-Poly	**SVM-Hist**	**MKBoost**	**SimpleMKL**
Single Features					
Global	GOFF	0.56	0.59	0.59	**0.61**
Global	VIF	0.42	0.46	**0.47**	**0.47**
Global	Log-C	0.47	0.51	**0.53**	0.49
Local	Cuboid	**0.43**	0.36	0.41	0.41
Combination of Global Features					
GOFF + VIF		0.60	0.61	**0.63**	**0.63**
GOFF + Log-C		0.59	**0.63**	0.62	**0.63**
VIF + Log-C		0.53	0.52	**0.56**	0.55
GOFF + VIF + Log-C		0.62	**0.63**	**0.63**	**0.63**
Combination of Global and Local Features					
Global + Local		0.64	0.62	0.63	**0.65**

Therefore, the combination of features was selected with respect to their main modalities to report the results (video, audio and sensor).

Table 3 shows the experimental results. Especially, dense trajectory features had promising results for single feature experiments which proves their ability to discriminate activities by modeling the ego-motions in videos. Single feature performances of classifiers are very close to each other. Thus, it is not possible to have a conclusive judgement on the relative performances of the classifiers based on single feature performances.

Table 3 F1-Score of Single Features and Feature Combinations for MEAD

		SVM-Linear	**MKBoost**	**SimpleMKL**
Single Features				
Video	Trajectory	0.63	0.64	**0.65**
Video	HOG	**0.72**	**0.72**	**0.72**
Video	HOF	**0.82**	**0.82**	**0.82**
Video	MBH	0.72	0.72	**0.73**
Sensor	FVS	**0.64**	0.63	0.62
Audio	Audio	0.44	0.43	**0.46**
Combination of Modalities				
Sensor + Video		0.83	0.85	**0.86**
Video + Audio		0.84	0.85	**0.86**
Sensor + Audio		0.63	0.67	**0.69**
Combination of All Modalities				
Video + Sensor + Audio		0.84	0.86	**0.87**

Another significant point is that adding new modalities generally increases the final recognition performances except when sensor and audio combination were used with linear SVM. For example, the best single feature performances of sensor and audio are 64% and 46%, respectively. When these two features are combined with MKL, their performance score increased up to 69% which also shows that sensor and audio modalities contain complementary information for activities. In general, MKL-based learning algorithms performed better when different modalities were combined. The difference between the maximum accuracy of single features and the accuracy of all feature combinations is more remarkable for MEAD (from 82% to 87%) compared to JPL (from 92% to 93%). That means the added features (especially audio) are more complementary for MEAD when used in combination with the other features. Additionally, when sensor features are combined with audio feature, SimpleMKL outperforms the other classifiers because of better weighting of multi-modal features.

The resulting confusion matrices for SVM, MKBoost and SimpleMKL are shown in Figure 4. The recognition performance is less when the motion level of the activity is low (i.e., reading, organizing files or texting) while the scores get better for the activities having higher ego-motion (i.e., walking, doing push-ups, or walking downstairs).

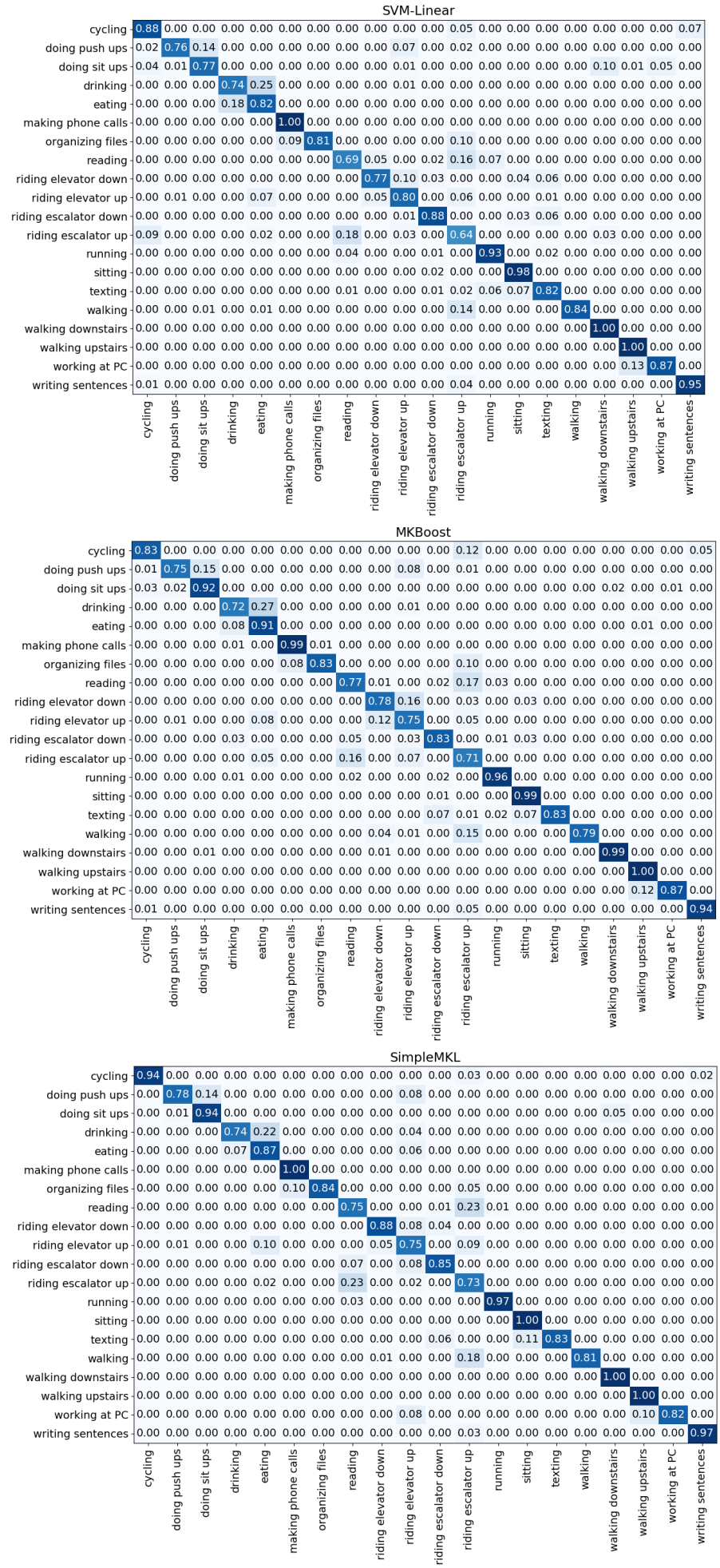
4.7 Comparative Results

In this section, the results are compared with the state-of-the-art methods (Table 4) with respect to the average accuracy (A), precision (P), recall (R), Kappa value (K) and F1-scores (F). Our results were produced by using all modalities: (a) global and local features for JPL and DogC, (b) video, audio and sensor features for MEAD.

Table 4 Comparative Performances of the Proposed Method

Dataset	Method	A	P	R	κ	F
JPL	Ryoo & Matthies [4]	0.90	-	-	-	-
	Abebe et al. [18]	-	0.87	0.85	-	0.86
	Ozkan et al. [19]	0.87	-	-	-	-
	Sudhakaran & Oswald [35]	0.91	-	-	-	-
	SVM	0.91	0.92	0.91	0.89	0.91
	DC-Int	0.87	0.90	0.87	0.87	0.89
	MKBoost	**0.92**	**0.94**	**0.92**	**0.91**	**0.93**
	SimpleMKL	**0.92**	0.93	**0.92**	**0.91**	**0.93**
DogC	Abebe et al. [18]	-	0.62	0.59	-	0.61
	Iwashita et al. [5]	0.61	-	-	-	-
	Ozkan et al. [19]	**0.65**	-	-	-	-
	SVM	0.63	0.65	0.63	0.58	0.64
	DC-Int	0.60	0.64	0.60	0.59	0.62
	MKBoost	0.61	0.64	0.63	0.57	0.63
	SimpleMKL	0.64	**0.66**	**0.65**	**0.61**	**0.65**
MEAD	SVM (Song et al. [11])	0.84	0.86	0.84	0.84	0.85
	MKBoost	0.86	0.85	0.86	0.85	0.86
	SimpleMKL	**0.87**	**0.88**	**0.87**	**0.86**	**0.87**

The results are compared with four other works [4, 18, 19, 35] for JPL. In [18], GOFF and VIF features were used with SVM and kNN classifiers. In [4], a structural learning approach was used with HOF, Log-C and cuboid features while an MKL-based solution was proposed in an earlier work [19]


SVM-Linear
MKBoost
SimpleMKL
cycling
doing push ups
doing sit ups
drinking
eating
making phone calls
organizing files
reading
riding elevator down
riding elevator up
riding escalator down
riding escalator up
running
sitting
texting
walking
walking downstairs
walking upstairs
working at PC
writing sentences

using the same features. In [35], a convolutional long-short term memory (LSTM) was used to perform feature encoding with convolutional neural networks (CNN) while holding long term temporal changes. The results show that SimpleMKL and MKBoost achieve similar performances and get the highest scores. Additionally, extending the feature set for the proposed MKL-based solution improves the overall accuracy when compared with the results in [19] in which a subset of the features used in this study.

For DogC, we took into consideration the results of [5, 18, 19] in which DogC was also tested. The method proposed in [5] used global (dense optical flow and local binary pattern) and local (normalized pixel values, HOG and HOF) motion descriptors and combined them with a modified histogram intersection kernel.

Finally, the results of MEAD were compared with [11] in which the same video and sensor features were used. Different to [11], audio feature was used as another modality with MKL. It should be emphasized that adding only audio without changing the learning mechanism used as in [11] improved the recognition accuracy from 83% to 84%. Another significant result was obtained by changing linear SVM with MKL which also increased the recognition accuracy from 84% to 87%.

5 Discussion

In order to understand the adaptive nature of MKL methods, a statistical analysis was performed using base kernel and feature selections for MKBoost. For that purpose, the selected base kernels and features were recorded at each trial and the advantages and disadvantages of the proposed framework were discussed based on these experiments.

5.1 Base Kernel Selection

Base kernel selection is one of the important phases in training. In this section, the statistical results are given for base kernel type selection after repeating the tests for 100 trials (Figure 5). According to the results, linear kernel was the most preferred kernel for all the datasets. However, selection characteristics are different for all three datasets. For example, following the linear kernel, polynomial kernel was the most dominant type for JPL while RBF, hist-int and DC-hist were selected much less frequently. RBF, hist-int and poly were preferred with a similar rate for DogC for which DC-Hist was the least favored kernel. For the MEAD dataset, the algorithm predominantly selected linear kernel. While RBF and polynomial kernels are also selected, they were much less favored. This shows that the features extracted from the MEAD dataset were mostly linearly separable. A more diverse range of base kernel selection for DogC indicates that the features used for it consist of mostly non-linearly separable samples. JPL dataset features proved to be more mixed in terms of their linear separability.

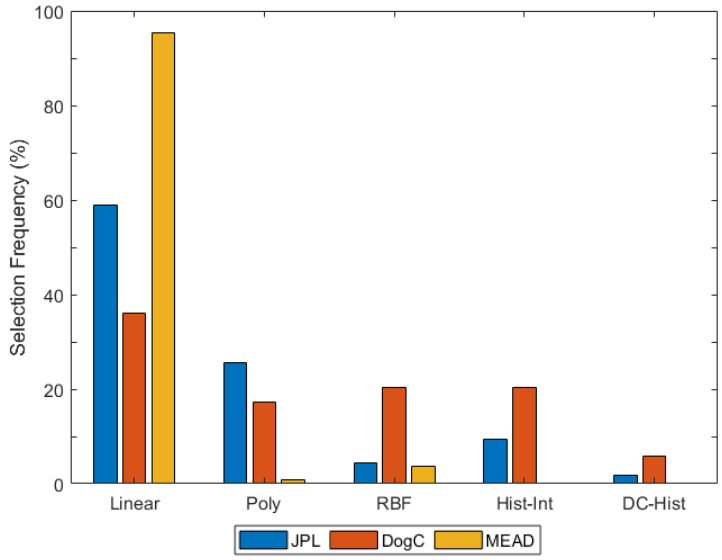


Fig. 5 The number of selected base kernels for each dataset.

5.2 Feature Selection

Feature selection is another important phase of training. Frequency of selection for different features is shown in Figure 6. At each trial, a specific feature is assumed to be selected if it is in the selected feature combination set. GOFF, VIF, Log-C and cuboid features were provided for JPL and DogC while the features for the main modalities (video, sensor, audio) are grouped together for the MEAD dataset for readability. For example, GOFF is assumed to be selected when any feature combination that includes GOFF (e.g., GOFF+VIF+Log-C) is selected. The results show that the optical flow-based feature, GOFF, is the most discriminative one for JPL. GOFF is the second most selected feature for DogC after cuboid. Video features are the most preferred ones compared to sensor and audio for MEAD.

On the other hand, the selection rates of features unveil some information about the characteristics of egocentric datasets. The features in the order of their selection frequency for different datasets can be listed as: JPL: GOFF > VIF > Log-C > Cuboid, DogC: Cuboid > GOFF > VIF > Log-C; MEAD: Video > Sensor > Audio.

The difference in ordering for datasets proves that MKBoost is able to adapt to the input data with different characteristics. While the global features usually gave the best results for JPL, the local feature was more important than the global features for DogC due to the more hectic nature of dog motions.

A more detailed analysis for kernel and feature selection were also performed to extract which feature combinations are mostly used with which base kernels. Therefore, the histograms of the selected base kernels were extracted

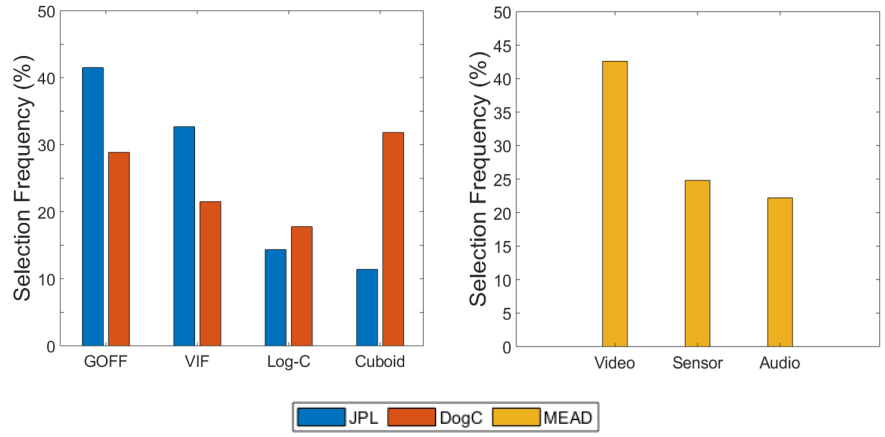


Fig. 6 The number of selected features for each dataset.

for each feature combination as shown in Figure 7. The feature combination IDs and their feature compositions are given in Figure 7-(d). The related feature combination includes the feature if it is marked as yellow. The feature combination IDs are the same for JPL and DogC. However, they are different for MEAD since it uses different set of features.

Figure 7-(a) shows that GOFF, VIF and GOFF+VIF (Feature IDs 1, 2 and 5) were the most dominantly selected features for JPL. For DogC, cuboid and GOFF (Feature IDs 4 and 1) were selected mostly as a single feature and cuboid was included in the most selected feature combinations according to Figure 7-(b). On the other hand, dense trajectory features were selected nearly in all feature combinations for MEAD (Figure 7-(c)). It is interesting to note that although audio has a considerable selection rate (Figure 6) for MEAD, it is generally selected in combination with other features rather than individually, suggesting that it provides complementary information to the other modalities.

5.3 Analysis of the Overall Framework

In this study, three types of sensor data (video, audio and wearable sensors) were used with an MKL-based framework to recognize egocentric activities. The results show that the proposed solution is effective in discriminating egocentric activities as well as providing a modular framework which can be extended with additional sensors. New features from these modalities are simply new channels of information to be adaptively learned by the base learners. The weight of each feature is assigned with respect to its contribution to the classification performance. By this way, feature selection and model training are done concurrently.

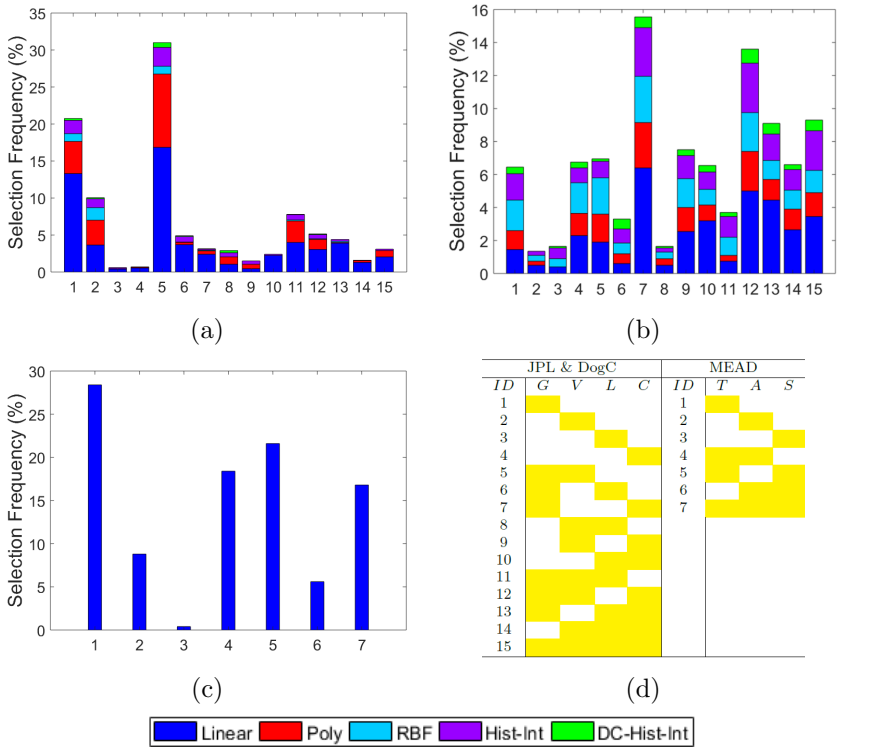


Fig. 7 The number of selected feature combinations of JPL (a), DogC (b) and MEAD (c) for the given feature composition color codes (d) in which feature names are abbreviated as follows: GOFF:G, VIF:V, Log-C:L, Cuboid:C, Audio:A, Dense Trajectory Features:T, Sensor Features:S

The performance results show that the proposed framework consistently produces better performance compared to the state-of-the-art methods. Multiple kernel learning is expected to yield mostly better results than single kernel learning. However, in some cases, single kernel solutions have the best performance results such as cuboid for DogC and FVS for MEAD. This is because using the same set for all feature combinations does not always give the optimal solution. Our experiments have confirmed that when the kernel pool for MKL is changed accordingly, the performance results are on par with the single kernel method. Additionally, it was observed that MKBoost did not perform as well as SimpleMKL for DogC and MEAD. Boosting technique needs sufficient number of labelled samples to get good classification performance [68]; whereas the number of samples used in this work is limited by the number of videos in datasets. As DogC and MEAD contain relatively fewer videos, MKBoost performance remains lower for these datasets. MKL methods (especially SimpleMKL) outperforms the others when the features are combined, which makes MKL a prominent method for the fusion of features.

Table 5 Time Complexity Analysis of the Proposed Methods

	Training Time per Video Segment (ms)			**Test Time per Video Segment (ms)**		
	Single Kernel	MKBoost	SimpleMKL	Single Kernel	MKBoost	SimpleMKL
Single Channel	6.6	14946.7	79.5	4.2	207.4	14.6
2 Channels	6.8	30320.6	182.7	4.3	254.3	18.9
3 Channels	7.2	71371.8	189.1	4.5	255.3	16.6

Another important point is that the use of multiple modalities (i.e., video and audio) improves the activity recognition performance implying that the multi-modal features coming from different domains have complementary information. Moreover, it is observed that MKL is a more effective solution when using multi-modal features compared to the other state-of-the-art methods. For instance, combining sensor features with audio for MEAD provides a significant improvement with MKL compared to single kernel learning.

Unlike single kernel learning and MKBoost, it is hard to formulate the time complexity of SimpleMKL that runs an optimization algorithm to determine the kernel weights. Because the feature extraction process is the same for all learning strategies, the time complexity analysis for the training and test was performed among classifiers. The average time to process video segments for MEAD is shown in Table 5. In order to assess the variation in the execution times with respect to the number of channels, tests were conducted with one channel (video), two channels (video+sensor) and three channels (video+sensor+audio), respectively. PC configuration for the experiments was Intel® Core^{TM} i5-6200U @ 2.30GHz with 8GB RAM.

Table 5 shows that single kernel solution is the fastest for both the training and test while MKBoost is the slowest. It is known that the computational cost of SVMs is related to the final number of support vectors and modern SVM solvers come close to a scaling law which indicates the computational cost of solving the SVM problem has both a quadratic and a cubic component growing at least like n^2 when C is small and n^3 when C gets large [69]. On the other hand, MKBoost consists of multiple kernel solvers whose numbers are directly proportional to the number of feature combinations (F), the trial numbers (T) and the base kernel pool size (K_p). Therefore, the timing performance of MKBoost is consistent with the theoretical calculation since its training time is equal to $F * T * K_p * T_{single}$ where T_{single} is the training time needed for single kernel. Unlike MKBoost, SimpleMKL uses the previous SVM solution that provides a good guess for the current SVM training. Thus, the computation time per SVM training is considerably less. According to these results, albeit slower than a single kernel method, SimpleMKL can be preferred as an alternative learning algorithm for sensor data streams considering its fast convergence and efficient learning performance.

Additionally, the proposed solution has promising results on three different egocentric datasets having varying numbers of activities ranging from 7

to 20, showing the scalability of the proposed framework. Even though MKL approaches got satisfactory learning performances for EAR problem, they require configuring a pre-defined kernel pool. If the basis kernels are not selected properly, MKL approaches may not be able to converge to the optimal solution.

6 Conclusion

In this work, we proposed a new framework for EAR based on multi-modal features combined with multi-kernel learning classification. Our experiments have shown that combining different modalities improves the recognition performance. The proposed solution has been tested on three different egocentric datasets and achieved better performances compared to state-of-the-art methods. This study has shown that using multimodal features with MKL is an effective method for EAR.

On the other hand, it is obvious that the variation of the selected feature and kernel sets for MKL is directly related with the characteristics of egocentric datasets. This observation implies that it is not possible to define a common set of best features and kernels for all datasets since the recording conditions of videos (place, time, actors, etc.), available sensor information (visual, audio, sensor, etc.) and the dynamics of egocentric actions vary for different datasets. Hence, an adaptive method should be proposed that dynamically learns the changing conditions of datasets such as the one proposed in this paper to provide a generic solution for the recognition of egocentric activities.

Combining visual information with audio or wearable sensors still requires further research for EAR. Contrary to third-person activity recognition, egocentric activity datasets may potentially contain more information about activities by the help of a variety of sensors directly recording the event. Thus, it is necessary to develop new frameworks that can receive features from different domains and combine them in an efficient and practical way to be able to recognize the activities of users.

The proposed framework adaptively fuses the features from different modalities, however it depends on handcrafted features. On the other hand, there are increasing number of studies in the literature proposing end-to-end solutions with deep learning techniques using visual [31, 41, 42] and wearable sensor [70, 71] information. Developing a generic end-to-end solution that also automatically learns the features inherently coming from different modalities is an open research challenge for the researchers.

References

1. Lu, Y., Wei, Y., Liu, L., Zhong, J., Sun, L., Liu, Y.: Towards unsupervised physical activity recognition using smartphone accelerometers. Multimedia Tools and Applications 76(8), 10701–10719 (2017)

2. Pansiot, J., Stoyanov, D., McIlwraith, D., Lo, B.P., Yang, G.Z.: Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems. In: 4th international workshop on wearable and implantable body sensor networks (BSN 2007), pp. 208–212. Springer (2007)
3. Wang, X., Rosenblum, D., Wang, Y.: Context-aware mobile music recommendation for daily activities. In: Proceedings of the 20th ACM international conference on Multimedia, pp. 99–108. ACM (2012)
4. Ryoo, M.S., Matthies, L.: First-person activity recognition: What are they doing to me? In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2730–2737 (2013)
5. Iwashita, Y., Takamine, A., Kurazume, R., Ryoo, M.S.: First-person animal activity recognition from egocentric videos. In: 2014 22nd International Conference on Pattern Recognition, pp. 4310–4315. IEEE (2014)
6. Song, S., Chandrasekhar, V., Cheung, N.M., Narayan, S., Li, L., Lim, J.H.: Activity recognition in egocentric life-logging videos. In: C.V. Jawahar, S. Shan (eds.) Computer Vision - ACCV 2014 Workshops, pp. 445–458. Springer International Publishing, Cham (2015)
7. Liu, Y., Nie, L., Han, L., Zhang, L., Rosenblum, D.S.: Action2activity: recognizing complex activities from sensor data. In: Twenty-fourth international joint conference on artificial intelligence (2015)
8. Liu, Y., Nie, L., Liu, L., Rosenblum, D.S.: From action to activity: sensor-based activity recognition. Neurocomputing 181, 108–115 (2016)
9. Liu, L., Cheng, L., Liu, Y., Jia, Y., Rosenblum, D.S.: Recognizing complex activities by a probabilistic interval-based model. In: Thirtieth AAAI conference on artificial intelligence (2016)
10. Nweke, H.F., Teh, Y.W., Al-Garadi, M.A., Alo, U.R.: Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. Expert Systems with Applications 105, 233–261 (2018)
11. Song, S., Cheung, N.M., Chandrasekhar, V., Mandal, B., Liri, J.: Egocentric activity recognition with multimodal fisher vector. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2717–2721. IEEE (2016)
12. Ni, B., Pei, Y., Moulin, P., Yan, S.: Multilevel depth and image fusion for human activity detection. IEEE transactions on cybernetics 43(5), 1383–1394 (2013)
13. Chen, C., Jafari, R., Kehtarnavaz, N.: Improving human action recognition using fusion of depth camera and inertial sensors. IEEE Transactions on Human-Machine Systems 45(1), 51–61 (2014)
14. Avola, D., Bernardi, M., Foresti, G.L.: Fusing depth and colour information for human action recognition. Multimedia Tools and Applications 78(5), 5919–5939 (2019)
15. Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R.: Berkeley mhad: A comprehensive multimodal human action database. In: 2013 IEEE Workshop on Applications of Computer Vision (WACV), pp. 53–60. IEEE (2013)
16. Xia, H., Hoi, S.C.: Mkboost: A framework of multiple kernel boosting. IEEE Transactions on knowledge and data engineering 25(7), 1574–1586 (2012)
17. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research 9(Nov), 2491–2521 (2008)
18. Abebe, G., Cavallaro, A., Parra, X.: Robust multi-dimensional motion features for first-person vision activity recognition. Computer Vision and Image Understanding 149, 229–248 (2016)
19. Ozkan, F., Arabaci, M.A., Surer, E., Temizel, A.: Boosted multiple kernel learning for first-person activity recognition. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1050–1054. IEEE (2017)
20. Ordóñez, F.J., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1) (2016)
21. Bhattacharya, S., Lane, N.D.: From smart to deep: Robust activity recognition on smartwatches using deep learning. In: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 1–6 (2016)
22. Yi, W., Ballard, D.: Recognizing behavior in hand-eye coordination patterns. International Journal of Humanoid Robotics 6(03), 337–359 (2009)

23. Li, Y., Ye, Z., Rehg, J.M.: Delving into egocentric actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 287–295 (2015)
24. Wang, H., Kläser, A., Schmid, C., Cheng-Lin, L.: Action recognition by dense trajectories. In: CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition, pp. 3169–3176. IEEE (2011)
25. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. ACM Computing Surveys (CSUR) 43(3), 16 (2011)
26. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. Computer Vision and Image Understanding 150, 109–125 (2016)
27. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR 2008-IEEE Conference on Computer Vision & Pattern Recognition, pp. 1–8. IEEE Computer Society (2008)
28. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision 103(1), 60–79 (2013)
29. Betancourt, A., Morerio, P., Regazzoni, C.S., Rauterberg, M.: The evolution of first person vision methods: A survey. IEEE Transactions on Circuits and Systems for Video Technology 25(5), 744–760 (2015)
30. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: 2011 International Conference on Computer Vision, pp. 407–414. IEEE (2011)
31. Poleg, Y., Ephrat, A., Peleg, S., Arora, C.: Compact cnn for indexing egocentric videos. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–9. IEEE (2016)
32. Incel, O.: Analysis of movement, orientation and rotation-based sensing for phone placement recognition. Sensors 15(10), 25474–25506 (2015)
33. Yilmaz, T., Foster, R., Hao, Y.: Detecting vital signs with wearable wireless sensors. Sensors 10(12), 10837–10862 (2010)
34. Clarkson, B., Mase, K., Pentland, A.: Recognizing user context via wearable sensors. In: Digest of Papers. Fourth International Symposium on Wearable Computers, pp. 69–75 (2000)
35. Sudhakaran, S., Lanz, O.: Convolutional long short-term memory networks for recognizing first person interactions. 2017 IEEE International Conference on Computer Vision Workshops (ICCVW) pp. 2339–2346 (2017)
36. Abebe, G., Cavallaro, A.: Hierarchical modeling for first-person vision activity recognition. Neurocomputing 267, 362–377 (2017)
37. Wang, X., Gao, L., Song, J., Zhen, X., Sebe, N., Shen, H.T.: Deep appearance and motion learning for egocentric activity recognition. Neurocomputing 275, 438 – 447 (2018)
38. Song, H., Thiagarajan, J.J., Sattigeri, P., Ramamurthy, K.N., Spanias, A.: A deep learning approach to multiple kernel fusion. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2292–2296 (2017)
39. Yu, C., Bambach, S., Zhang, Z., Crandall, D.J.: Exploring inter-observer differences in first-person object views using deep learning models. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 2773–2782 (2017)
40. Poleg, Y., Arora, C., Peleg, S.: Temporal segmentation of egocentric videos. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2537–2544 (2014)
41. Ma, M., Fan, H., Kitani, K.M.: Going deeper into first-person activity recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1894–1903 (2016)
42. Kwon, H., Kim, Y., Lee, J.S., Cho, M.: First person action recognition via two-stream convnet with long-term fusion pooling. Pattern Recognition Letters 112, 161–167 (2018)
43. Sathyanarayana, A., Joty, S., Fernandez-Luque, L., Ofli, F., Srivastava, J., Elmagarmid, A., Arora, T., Taheri, S.: Sleep quality prediction from wearable data using deep learning. JMIR mHealth and uHealth 4(4), e125 (2016)
44. Alsheikh, M.A., Niyato, D., Lin, S., Tan, H.P., Han, Z.: Mobile big data analytics using deep learning and apache spark. IEEE network 30(3), 22–29 (2016)
45. Wang, L.: Recognition of human activities using continuous autoencoders with wearable sensors. Sensors 16(2), 189 (2016)

46. Yao, R., Lin, G., Shi, Q., Ranasinghe, D.C.: Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. Pattern Recognition 78, 252–266 (2018)
47. Chen, Y., Zhong, K., Zhang, J., Sun, Q., Zhao, X.: Lstm networks for mobile human activity recognition. In: 2016 International Conference on Artificial Intelligence: Technologies and Applications. Atlantis Press (2016)
48. Guan, Y., Plötz, T.: Ensembles of deep lstm learners for activity recognition using wearables. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1(2), 11 (2017)
49. Abebe, G., Cavallaro, A.: Inertial-vision: cross-domain knowledge transfer for wearable sensors. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1392–1400 (2017)
50. Tadesse, G.A., Cavallaro, A.: Visual features for ego-centric activity recognition: A survey. In: Proceedings of the 4th ACM Workshop on Wearable Systems and Applications, pp. 48–53. ACM (2018)
51. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. Journal of machine learning research 12(Jul), 2211–2268 (2011)
52. Guo, K., Ishwar, P., Konrad, J.: Action recognition from video using feature covariance matrices. IEEE Transactions on Image Processing 22(6), 2479–2494 (2013)
53. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. VS-PETS Beijing, China (2005)
54. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine 56(2), 411–421 (2006)
55. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital signal processing 10(1-3), 19–41 (2000)
56. Safavi, S., Russell, M., Jančovič, P.: Automatic speaker, age-group and gender identification from children's speech. Computer Speech & Language 50, 141–156 (2018)
57. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE transactions on acoustics, speech, and signal processing 28(4), 357–366 (1980)
58. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al.: The htk book (v3. 4). Cambridge University (2006)
59. Gauvain, J.., Chin-Hui Lee: Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. IEEE Transactions on Speech and Audio Processing 2(2), 291–298 (1994)
60. Campbell, W.M., Sturim, D.E., Reynolds, D.A., Solomonoff, A.: Svm based speaker verification using a gmm supervector kernel and nap variability compensation. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 1, pp. I–I (2006)
61. Ni, B., Nguyen, C.D., Moulin, P.: Rgbd-camera based get-up event detection for hospital fall prevention. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1405–1408. IEEE (2012)
62. Gärtner, T.: A survey of kernels for structured data. ACM SIGKDD Explorations Newsletter 5(1), 49–58 (2003)
63. Schölkopf, B., Smola, A.J., Bach, F., et al.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (2002)
64. Lanckriet, G.R., De Bie, T., Cristianini, N., Jordan, M.I., Noble, W.S.: A statistical framework for genomic data fusion. Bioinformatics 20(16), 2626–2635 (2004)
65. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence 14(771-780), 1612 (1999)
66. Fitzgerald, R., Lees, B.: Assessing the classification accuracy of multisource remote sensing data. Remote sensing of Environment 47(3), 362–368 (1994)
67. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2(3), 27 (2011)
68. Li, X., Wang, L., Sung, E.: Improving adaboost for classification on small training sample sets with active learning. In: Proceedings of Asian Conference on Computer Vision (ACCV), pp. 1–6 (2004)

69. Bottou, L., Lin, C.J.: Support vector machine solvers. Large scale kernel machines 3(1), 301–320 (2007)
70. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys (CSUR) 46(3), 33 (2014)
71. Morales, J., Akopian, D.: Physical activity recognition by smartphones, a survey. Biocybernetics and Biomedical Engineering 37(3), 388–400 (2017)