Multiple Model Recognition for Near-Realistic Exergaming

Bobak Mortazavi, Mohammad Pourhomayoun, Suneil Nyamathi,
Brandon Wu, Sunghoon Ivan Lee, Majid Sarrafzadeh
Computer Science Department, UCLA
Los Angeles, CA 90095
Email: {bobakm, mpourhoma, snyamathi, silee, majid}@cs.ucla.edu, brandonwu@ucla.edu

***Abstract*—Exergaming as a tool to combat obesity yields an interesting take on the problem of design and implementation of activity recognition systems for truly mobile games that achieve moderate levels of intensity. This work presents SoccAR, a mobile, sensor-based wearable exergaming system with fine-grain activity recognition. The system in this paper presents a recognition algorithm for the appropriate classification of 26 movements by extracting a large number of features and selecting the most important, as well as developing a multiple model strategy to better classify movements. This movement strategy allows for a trade off of detailed classification versus classification speed. A metric to define the accuracy in terms of the importance of particular movements is defined. The scheme presented develops a framework for more accurately classifying movements with a smaller number of features for a large, multiclass real-time environment. This results in a more accurate classification of movements, with an F-score in cross-validation of $.937$ using a PUK-kernel based SVM and multiple models, to $.755$ using only a single RBF-based model and 20 features.**

I. Introduction

Sedentary behavior is a root cause of several chronic conditions affecting health of adult and children in the United States and worldwide [1][2]. With a worldwide trend of inactivity, with 15% of men and 20% of women surveyed in 51 one countries being at risk for chronic diseases [3], monitoring of physical activity is of utmost importance. Physical inactivity causes not only health concerns, but can increase healthcare costs for each individual [4]. Such physical inactivity often leads to overweight and obese populations [5] that result in chronic conditions such as cardiovascular disease or diabetes. Cardiovascular disease and diabetes both also present a significant economic burden in the United States, with cardiovascular disease accounting for an estimated \$670 billion in health care costs in 2010 [6] and estimates suggesting diabetes may approach \$900 billion in health care costs by 2015 [7]. Indeed, obesity from sedentary behavior causes an extreme health and economic burden [8]. As a result, many solutions using wireless wearable sensors have been presented to monitor activity [9][10] in the hopes of eventually promoting it.

One such solution to this sedentary behavior is to use body-worn sensors, such as accelerometers, to enable the playing of exercise-worthy video games, or Exergames[11]. Exergaming involves incorporating health information and healthy activity into gameplay with the hope that the entertainment factor of gaming coupled with the health information can make better use of the time spent playing games. Many exergames exist that address particular aspects of personal health. Work in [12], for example, presents a platform for developing an exergame to assist in stroke rehabilitation using accelerometers, representing a class of exergames with a specific goal and set of actions in mind. Work by [13] develops an entertainment-based exergame with multiplayer aspects to encourage competition, using a hula-hoop, jump rope, or riding a stationary bike. Similarly, work by [14] demonstrates a motion-based game for exercise and entertainment of adults, but uses a Microsoft Kinect which, while allowing freedom in a particular space, does not allow for a game that can be played anywhere and across any distance. Finally, a set of mobile games, such as those in [15][16] allow for gaming in any environment by using a mobile computing device (such as a smartphone) as the controller. So while these games allow for freedom in environment, they do not address a wide range of possible motions and exercises. Exergames and other physical activity monitoring systems have been developed to track the users and identify their activity levels with different degrees of accuracy and application [12][17], including verifying the exercise-worthiness [18] of sports exergames, which use fine-grain sports motions instead of cyclical, repetitive motions [19]. The heart of these exergaming systems is the recognition algorithms that allow for the tracking of these movements at various intensity levels [20] and preventing the cheating of these actions [21].

Many activity recognition problems concern themselves with similar issues, namely, setting up a model and then showing this model is robust and general, either in a testing environment or in cross-validation[22]. However, it is possible to leverage specific information and context to improve classification results [23]. Indeed, often multiple models or hierarchical classifiers can improve classification results, from using structural information [22], conditional information [23], or expert knowledge to improve classification [24]. These systems all attempt to relate specific information about the given environment or classification goals to improve the results and behavior of such systems, from the speed to the accuracy.

This paper presents SoccAR, an Augmented Reality Soccer obstacle course exergame based upon the Temple Run platform to incorporate a dynamic mobility game with detailed sports-type, fine-grain actions. A head-mounted eyewear display allows for full range of motion and mobility for limbs with attached inertial measurement unit sensors. By wearing a translucent display, users are able to focus on the game while seeing their surrounding environment. This work focuses on the design of this game, specifically, the recognition model

978-1-4799-8033-8/15/\$31.00 ©2015 IEEE

designed to classify a large set of movements. Further, this paper will investigate a multiple model approach to develop a recognition algorithm that can robustly classify a large number of movements as well as present optimization possibilities from extra models to reduced features. Further, it will develop an idea of importance factor for each model, something possible specifically in the gaming environment. This importance factor allows for the evaluation of systems where, for example, two kinds of passes do not need to be well-differentiated from each other, as long as they are from the rest of the system.

II. Related Works

A. Multiple Models Works

Work in [25] builds hierarchical hidden markov models in order to better identify activities of daily living by using multiple models. Multiple models allow for their work to improve classification results of activities of daily living, particularly in an environment in which these movements might overlap. In this work, overlapping motions are not a concern, but the idea of using this information to reducing computation is one that is adapted here. While the work in [25] does reduce computation and help set up a semi-supervised setting, the system is not set up to adapt in a real-time setting, and thus, is not applicable to the results desired in this work.

Work in [26] takes a multiple model approach to classifying patients with heart failure. The technique presented allows for a clustering of patients based upon contextual information (e.g. medical conditions and socioeconomic status), and then risk classification models built for each of those clusters to achieve more accurate results. This work will similarly adapt this approach by developing a real-time implementation of such a system, as well as give an importance factor to each model, a feature available to gaming environments not possible in health-care settings. Further, contextual knowledge on the movements themselves allows for a supervised approach for more precise models.

B. Fine-Grain Activity Exergaming

Work in [19] describes the steps necessary to building an exergame that results in healthy and enjoyable gameplay with accurate motion classification. A tablet-based game is built from a list of soccer movements, which the paper defines as fine-grain, collected to develop an energetic and fast-paced game. Fine-grain activities are those singular, short burst precise movements as compared to cyclical, repetitive movement patterns often found in activities of daily living recognition systems. A game-specific movement classification algorithm is created based upon a principal component analysis, hereafter known as Fine-Grain PCA. This algorithm classifies movements shown in the [18] to produce exercise-levels of intensity by guaranteeing a certain level of metabolic output. That paper introduces a framework for creating successful mobile, wearable exergames by outlining a general procedure of data collection and game built together; however, the game ultimately designed in that paper uses only a single sensor and a tablet computer that must be held by the user throughout gameplay. Further, the actions, while intense, are essentially completed in place. As a result, this work is built off of the principals presented in [19] in that movements are collected for a fast, intense exergame based upon realistic actions with a wearable system that does not restrict the user's limbs or movement ranges. Further, the classification algorithm presented here will result in a more accurate gameplay experience, as will be shown in Section V when compared to the Fine-Grain PCA Algorithm presented in [19].

(a) Shimmer3 Sensors

(b) Head-Mounted Display

Fig. 1. Shimmer sensor (left) and Espon Moverio glasses (right)

III. System

This section describes the development of the SoccAR platform. It will cover the data collection platform and game design for use. The game has two distinct components that work together to make SoccAR a successful mobile exergame with intense activities from a soccer environment. The first is a sensor platform to input actions from the user in any environment, the second is the computing device and display for the user to interact with the game and gaming environment, these are shown in Figure 1.

A. Head-Worn Display

Ultimately, what can make a game successful as a mobile game is the ability to display to the user the gameplay environment while leaving the hands and legs free. To enable this, head-worn displays are needed. However, these displays must also let the user see the actual environment, so that they do not run into actual physical objects. As a result, the game is implemented on a laptop and displayed on the Epson Moverio BT-100. The user is presented with the game in the small displays in the middle of the lenses.

B. Data Collection

Figure 2 shows a user playing SoccAR. Data was collected from 16 individual volunteers. They were asked to wear four Shimmer [27] wireless inertial measurement units (IMU) platforms, two attached to each wrist, two on top of each foot. Each streamed data at 50 Hz and presented three axis accelerometer ($\pm 6g$) and gyroscope ($\pm 250\frac{deg}{s}$) data as well as comes estimates of the orientation of the sensor, presented as four quaternions. The data is streamed, via bluetooth, to a computing device. With this sensor placement a truly mobile range of motion data was collected for 26 different fine-grain movements as well as some various other random motion to help build a "no movement class." For each move, ten repetitions were recorded. Since a Temple-Run game is being adapted for the game in this paper, the motions included soccer motions as well as newer activities to make an enjoyable mobile exergame. Table I shows the list of moves both sports related and general running motions.

TABLE I. FINE-GRAIN MOVEMENTS FOR SOCCAR

Soccer	Running	Misc.
Shoot	Shoot (while running)	Open Door
Square Pass	Run	Unsheathe Sword
Through Pass	Jump (in place)	Draw Bow
Chip	Jump (while running)	Shoot Bow
Flick Pass	Jab (while running)	Jab
Placed Shot	Sword Slash (while running)	Sword Slash
Cut Left	Cut Left (while running)	Slide
Cut Right	Cut Right (while running)	Walk
Spin	Spin (while running)	

(a) User playing SoccAR (b) Game Screenshot

Fig. 2. User playing SoccAR (left), kicking a ball. Note the shimmer sensors on each wrist and each foot. Screenshot of the game (right)

C. Game Design

Figure 2 shows a user playing the SoccAR game and a screenshot of this game. The game is an adaptation of a Temple-Run game called Ruin Run based off of the Unity platform. It allows the game designer to use the Unity3D game engine to make a run/dash style obstacle course (similar to the idea presented in [19] but with visual cues). Users can walk, run or stand still to perform certain actions. The game presents the user in a third person view, like in Figure 2b. This screenshot presents the user with a ball that needs to be kicked out of the way. The duration of the game can be set before the user plays, allowing for a short game for a quick break or a long game for serious exercise. Given how long it takes to avoid obstacles, and how many calories are burned by the user, calculated as in [18], a complete score is presented at the end of the game to give the user an indication of how well they played, as well as serve as a competitive motivator against other users. For each move, a timer is started to see how long it takes the user to react to the presented object. This delay is inverted and scaled, and added to the calories burned in the following formula:

$$Score = \alpha \times \frac{1}{\delta} + \beta \times Cal \tag{1}$$

where δ represents the total delay in performing all actions, Cal the calories burned as computed by the MET information of movements as well as the user's height and weight. α and β can be adjusted to give better scores to gameplay factors or to exercise factors as desired.

IV. METHODS

In order to accurately classify movements in a large, multiclass environment, a comprehensive examination of the recognition algorithm is necessary. This includes analyzing not only the accuracy of the movements but the speed with which these movements are classified by the game.

A. Training

Data were collected with an annotation process that marked the beginning and end of each movement. As the data was recorded, the individual in charge of running the data collection marked the beginning and end of each movement. The mid-point annotation is drawn from this start and end information. If the data is represented as a time series $D(t)$, where time t starts at 0 and goes until the length of w, windows are built around this as such:

$$move(midpoint, w) = (D(t - \lfloor \frac{w}{2} \rfloor), D(t + \lfloor \frac{w}{2} \rfloor)) \tag{2}$$

where w is the desired window size and $move(midpoint, w)$ is the windowed move around the desired midpoint. Before extracting these movement windows from the training set, the data is filtered with a low-pass filter via a moving average. This moving average was found, heuristically, to be one-fourth the average window size. The average window size is two seconds, or 100 points. For the longer movements, the movement is still captured as the beginning and end of movements usually entail picking the foot up and putting it back down, and deemed not providing the highest information gain, while in shorter movements these segments amount to noise. Finally, four sensors are used at the same time, thus, each move window is the following information from each sensor:

$$\begin{aligned} SensorData_{loc}(t) = \langle & G_x(t), G_y(t), G_z(t), \\ & A_x(t), A_y(t), A_z(t), \\ & Quat_1(t), Quat_2(t), Quat_3(t), Quat_4(t) \rangle \end{aligned} \tag{3}$$

where loc is the sensor identification on the body, the four $Quat$ signals are the four quaternions derived from the accelerometer and gyroscope readings as designed by [27] using a standard attitude and heading reference system (AHRS), and finally, the magnitude of the acceleration is calculated. If time length t data is collected, each component of the vector above is a length t time-series signal. Thus, the full data set at any point can be indicated by:

$$\begin{aligned} MoveData(t) = \langle & SensorData_{LA}(t), SensorData_{RA}(t), \\ & SensorData_{LL}(t), SensorData_{RL}(t), \\ & \|LA_a(t)\|, \|RA_a(t)\|, \|LL_a(t)\|, \|RL_a(t)\| \rangle \end{aligned} \tag{4}$$

where LA is the left arm, RA the right arm, LL the left leg, and RL the right leg respectively, and the final four channels are the magnitudes of acceleration of each sensor. Once the data is windowed and filtered, the movements are supplied to a

TABLE II. FEATURES EXTRACTED FOR FINE-GRAIN ACTIVITY CLASSIFICATION AND THEIR DESCRIPTIONS

Feature	Description
Minimum	Minimum value attained over window
Maximum	Maximum value attained over window
Sum	Sum of all values attained over window
Mean	Average of all values attained over window
Std. Dev	Standard Deviation of all values attained over window
Skewness	Measure of asymmetry in signal attained over window
Kurtosis	Measure of peakedness in a signal attained over window
Energy	Measure of frequency domain in a signal attained over window

support vector machine (SVM) classifier for feature extraction and training. The SVM used in this work is LibSVM [28] used in Matlab for an RBF kernel with default values for the parameters and Weka's [29] SMO implementation for the PUK Kernel to be discussed, also interfaced with Matlab.

1) Feature Extraction: As required, the appropriate set of features need to be extracted for accurate recognition. In particular, a combination of features that generally represent the motions must be matched with features that indicate the structure and time-dependence. For each channel above (44 in total, for example, x-acceleration, y-acceleration, etc.) a set of features is extracted. The gyroscope gives approximations of rotational movement, accelerometer the linear movements, and the quaternions on the orientation of the sensors. The list of features that represent these are shown in Table II. In particular, the energy of a signal is calculated as in [30][31], and other similar work, where

$$fft(move) = \{x_i | i = 1, \cdots, k\} \tag{5}$$

$$Energy(move) = \frac{\sum_{i=1}^{k} |x_i|^2}{k} \tag{6}$$

the fft of a move window results in k fft components, and the energy then follows from the components. Calculating all of these features, however, will result in over-fitting the data sets in training. In particular, [32] suggests that, for a linear SVM, the number of data to feature ratio should be $10:1$. However, this assumes each data point is an independent subject in a binary classification. In this case, the data set consists of 16 individuals and 26 movements. The number of samples, then would be 416 (in this case the 10 copies of each move are not considered as they inflate the data set size with polluted data, which should be avoided). As a result, no more than 42 features should be considered (and 50 are considered in this work for a multiclass classifier with a non-linear kernel for an upper bound that accounts for the different kernel types). For SVM all the features selected must then be normalized.

2) Feature Selection: When all of the features from Table II were calculated over each of the channels of the data set, 352 features resulted for each sample window, elements of a feature set denoted Ω. Selection algorithms, however, must not consider the entire data set when picking features, as this pollutes the testing results in cross-validation by including the test user in identifying key features. A leave-one-subject-out cross-validation (LOSOCV) is used to avoid pollution of the training set for the testing set. Then, this training set runs a LOSOCV on itself to determine the effectiveness of each feature ω to be added to the set Θ (in other words nesting

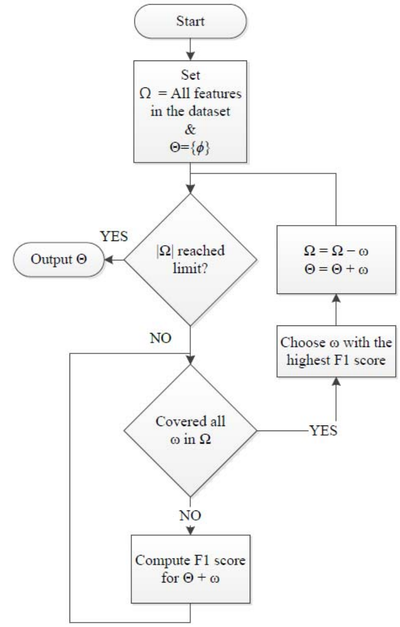

Fig. 3. Forward selection wrapper algorithm flowchart for feature selection, where computing $\Theta + \omega$ with the highest F score is done in a cross-validation scheme on the training set.

LOSOCV valuations, feature ranking each of the subsets). The flowchart for this algorithm is shown in Figure 3. Those features most often selected and the position in which they are selected are averaged to come up with the defined feature set for the entire system. While a backward elimination approach is the best to find the optimal subset of features, the duration and complexity of such an algorithm with 352 features precludes this from being run in this instance. Instead, the forward selection approach is employed as an appropriate approximation. The algorithm can use a classifier of choice, in this case, the same SVMs for the important features for the given classification scheme.

Table III shows the top 50 features selected. Note that many of these features are structurally important for different movements, showing different limbs moving or not (for example if the right leg is stationary, such as in drawing and shooting an arrow, versus a pass where the right leg is moving). With the spread of features across all limbs, the necessity for wearing all the sensors available and combining the general windowed results (e.g., mean) with structure specific results (e.g., Kurtosis/Skewness) becomes apparent. While work has been conducted to determine the sensor location, it is interesting to note that the quaternions appear to be of no use. As a result, perhaps only the relative adjustments of motions are required to determine the different movements.

B. Classification

Once the appropriate features are selected and a model created, live testing of data is run for gameplay mechanisms. Each incoming point from the four sensors is combined together to one vector. This point is filtered with a moving average filter of 25 points, then put into a sliding window of size 100 points as determined earlier. Once the features are extracted from this

TABLE III. TOP 50 SELECTED FEATURES FOR FINE-GRAIN CLASSIFIER BY FORWARD SELECTION AND WRAPPER METHOD

Features 1-10	11-20	21-30	31-40	41-50
Avg(RA,Mag)	Energy(RL,AccY)	Energy(LA,AccZ)	Std(LA,AccY)	Sum(LL,AccY)
Std(LL,GyroY)	Avg(LL,AccY)	Energy(RA,GyroZ)	Std(LA,AccX)	Std(LA,GyroZ)
Max(RA,AccY)	Energy(RL,GyroY)	Min(LL,Mag)	Energy(LL,AccX)	Min(LL,AccY)
Max(RA,Mag)	Min(LA,AccX)	Energy(LL,GyroX)	Energy(LL,AccY)	Std(LL,GyroZ)
Energy(LA,AccX)	Std(RA,GyroZ)	Skew(RL,Mag)	Skew(RA,Mag)	Std(LL,AccY)
Energy(RA,AccX)	Std(LA,GyroY)	Std(RA,Mag)	Std(LL,AccX)	Std(RL,AccY)
Energy(LL,GyroY)	Energy(LA,GyroZ)	Std(RA,AccX)	Sum(RA,Mag)	Std(RL,GyroY)
Energy(RA,Mag)	Std(LA,Mag)	Energy(RL,Mag)	Std(LL,AccZ)	Min(RA,GyroY)
Max(LA,AccY)	Std(RA,AccY)	Std(RL,Mag)	Sum(LL,Mag)	Std(RA,GyroY)
Kurt(RA,Mag)	Min(RA,Mag)	Energy(LL,GyroZ)	Max(RA,GyroZ)	Std(LL,GyroX)

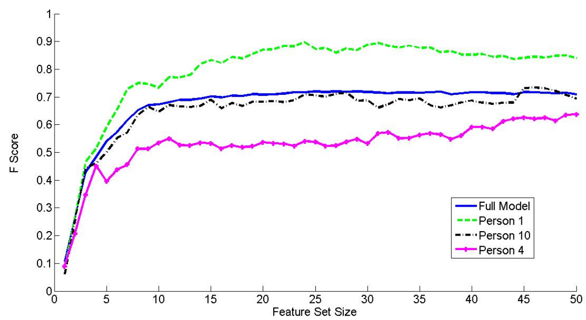

Fig. 4. Results of the LOSOCV for entire set on an SVM with RBF Kernel, and individual results for user 1, 10, and 4

movement buffer, it is supplied to the classifier in order to determine the movement. In the multiple model approach this can be done as one set of feature extraction for each model used, in series, or all possible features at once in parallel, depending upon the computational power of the gaming device and delay allowed. Finally, a user-centric approach can be determined by attempting to reduce the number of features extracted to find not only a faster result, but potentially a more accurate one, as in adjusting the window size, shown in the previous work [33]. The principal behind this adjustment is that each user responds to the model in a unique way with the potential to reduce the number of features needed to improve computational performance, as shown in Figure 4. Notice certain users are more accurate at a lower number of features. The accuracy results of the full model will be discussed further in Section V.

C. Model Generation

The classification accuracy can be dynamically improved, but the base algorithm needs to be accurate for that method to have meaningful results. The ability to reduce complexity of a model by reducing the number of labels can help a classification algorithm such as an SVM. In this case the desired goal is to use expert knowledge information to group similar movements together and generate models for each of these groups. In terms of the movements, by increasing the number of classes, the algorithm performances tend to degrade [19]. Further, those number of classes tend to have an effect when they are most similar in terms of movements (for example, two different types of passes). Thus, the method presented will use this information to create subclusters of movements that are similar. Figure 5 shows the general flow chart of applying expert knowledge to the clustering algorithm as desired, where each level L as desired can have as many

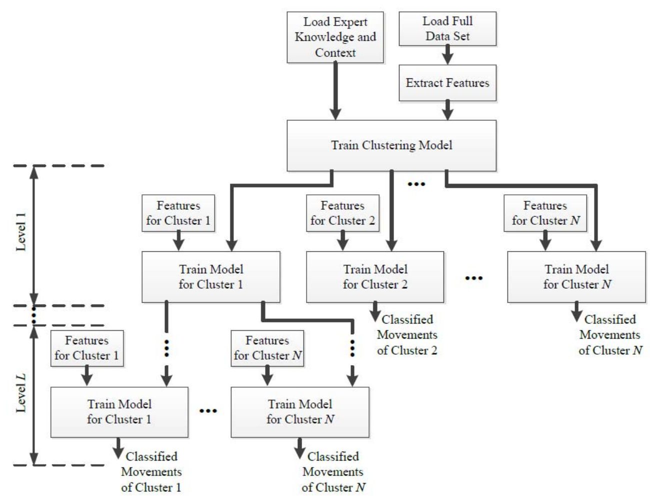

Fig. 5. Training with expert knowledge and clustering

TABLE IV. FINE-GRAIN MOVEMENTS FOR SOCCAR

Cluster 1 - Soccer	Cluster 2 - Misc.	Cluster 3 - Running
Shoot	Open Door	Run
Square Pass	Draw Arrow	Cut Left (while running)
Through Pass	Shoot Arrow	Cut Right (while running)
Chip	Jab	Jab (while running)
Flick Pass	Unsheathe Sword	Jump (while running)
Placed Shot	Sword Slash	Shoot (while running)
Cut Left	Walk	Spin (while running)
Cut Right		Sword Slash (while running)
Jump		
Spin		
Slide		

N clusters as needed to improve the accuracy. For this work $L = 2$ and $N = 3$.

Table IV shows the list of moves in their appropriate clusters. As SoccAR contains some soccer moves and some other moves adapted for obstacles in its Temple Run-style game, contextual knowledge of the movements yielded these fairly distinct clusters of movements. Cluster 1 consists of most moves for soccer type actions (breaking the soccer moves up into clusters for shots and passes is discussed further in Section VI. Cluster 3 results in movements that occur while currently running, including identifying running. Cluster 2 results in the remainder of movements in a miscellaneous grouping. These also happen to correspond to different expected readings from sensors. For example, most of the soccer movements are primarily foot movements, most of the miscellaneous mostly arm movements, while the other running cluster is going to be a noisy cluster of motions that consist of a fairly consistent linear velocity while performing them. Thus, four models will need to be generated. One model differentiates between the clusters, and the remaining three models classify moves in each cluster, with significantly fewer labels than 26. This top level clustering could be done in an unsupervised method and that is left for future work to explore the different such methods. Contextual knowledge here lends itself to an easy supervised nature of classification.

D. Model Importance

An important consideration when creating these multiple models, and analyzing the accuracy of a classification in a gaming environment is the ability to handle a certain level of error as long as the response is in a real-time fashion (e.g. as long as it passes one might not care that the exact correct pass was used). This work will define a new metric for measuring the accuracy of such classifiers as well. For any multiclass classification, the F score will be used, where:

$$F = 2 \times \frac{P \times R}{P + R} \tag{7}$$

where P is the precision and R the recall. Most multiple model systems [34][25][35] still present classification results as an overall metric. Given a multiple model scenario this often makes the most sense for analyzing how well a given system performs. For this work, the classification accuracy of each model will be analyzed and a metric created to calculate the weighted accuracy of multiple models. The reason this is done is, in the context of a gaming environment, sometimes the performance and speed are more important than the accuracy[33]. Take, for example, a situation in which a particular cluster contains two different passes to the left. Perhaps, in the context of a game, it is not as important to differentiate between the two passes as it is to identify that either pass has occurred. In that setting, the accuracy of the system to identify that cluster (the higher level model) is more important than the accuracy of the lower model. Or, in another example, the movement does not occur frequently, but for a recognition system's validation, is equally weighted with positive and negative examples. As such, different accuracy metrics must be calculated. For a given level and model, the accuracy, in terms of F-score, for the level is calculated as:

$$F_l = \frac{\sum_{i=1}^{n} \alpha_i * \|C_i\| * F_{l_i}}{\sum_{i=1}^{n} \alpha_i * \|C_i\|} \tag{8}$$

where F_l is the F-score for a certain level of multiple models (1 being the top, 2 the next and so forth), F_{l_i} is the F-score for the particular cluster/model in this level (e.g., F score for the soccer cluster in the table above), α_i is the weighted importance of this cluster in range $[0, 1]$, $\|C_i\|$ is the weighted size of the cluster. If, for example, all the α_i are 1 then the F score for the level will be the weighted average of the F score for each model in this level of the classifier. If the F_l values for each level of a multiple-model scenario are considered, then the overall F score will be calculated as:

$$F_{mm} = \frac{\sum_{i=1}^{L} \beta_i}{\sum_{i=1}^{L} \frac{\beta_i}{F_i}} \tag{9}$$

where the F_{mm} represents the F-score for the multiple model scheme, β_i is the importance of each model level in range $[0, 1]$ and F_i is the F-Score for the given level of the model. In other words, a weighted average is calculated for the F-score of a given level of models in this multiple model scheme, since even a single outlier would affect the result greatly, where as the weighted harmonic mean is chosen for the comparison across levels since a given level may not be crucial to the operation of the exergaming recognition system. This work considers this top level F-Score more representative of the multiple model scheme. However, where all weights are 1 this scheme represents the same accuracy as an overall system, of accurately determining the cluster, then accurately classifying in that cluster. For the purposes of comparison to other methods that weighting is presented in Section V. Further, by developing a single metric, it becomes easier to compare results across the models in a situation where the designer of such an exergame may not care about the distinction in a given model, such as the distinction between two types of passes, if those movements are deemed similar enough not to effect user experience.

E. Feature Extraction

With multiple models, it becomes necessary to extract multiple features. While the features extracted for a single model might be robust enough for an entire system, multiple models should have their own features extracted and selected. This is because a multiple model solution represents, essentially, multiple independent classification problems. As a result, the same feature extraction technique presented should be applied to the top level clustering scheme as well as each of the three clusters selected here. Similarly to the calculation of the F measure above for the entire selection, the number of features in use must be calculated over the entire model. To be useful in the sense of the optimization techniques, an overall feature number needs to be calculated. Otherwise, a sorted order of the primary reductions can be determined and used for each cluster and each level respectively. Similarly to above, the weighted feature value can be calculated, although the value of this possible computation will be left for future work as discussed in VI.

F. PUK Kernel and Weka SMO

For modeling this problem, an SVM with an RBF Kernel is implemented using LibSVM [28]. However, a less commonly used kernel will be compared, that should be used in more activity recognition systems. The Weka SMO implementation of SVM is used [29], called through Matlab, using a Pearson Universal Kernel (PUK). For this, the complexity parameter is increased to 100 and the PUK kernel is selected, with default values for its potential robustness. In particular, the PUK kernel is a universal kernel based on the Pearson VII function. This kernel is applied here to activity monitoring, as it has been shown to be a robust, universal kernel for support vector regression [36], discriminating protein types [37] and on gesture recognition [38]. The kernel is based on the Pearson VII function for curve fitting

TABLE V. TOP 50 SELECTED FEATURES FOR EACH MODEL TYPE

Model Type	Top 50 Features
Single Model	f_5,f_6,f_{14},f_{15},f_{21},f_{22},f_{23},f_{24},f_{25},f_{26},f_{35},f_{44},f_{53},f_{55},f_{56},f_{57},f_{61} f_{62},f_{64},f_{65},f_{72},f_{74},f_{75},f_{76},f_{78},f_{79},f_{81},f_{84},f_{85},f_{86},f_{93},f_{95},f_{96} f_{101},f_{102},f_{103},f_{104},f_{105},f_{106},f_{107},f_{112},f_{115},f_{124},f_{125},f_{126},f_{130} f_{131},f_{133},f_{135},f_{136}
Top-Level Model	f_{12},f_{14},f_{45},f_{51},f_{63},f_{74},f_{78},f_{93},f_{95},f_{122},f_{124},f_{125},f_{138},f_{146},f_{157} f_{185},f_{188},f_{195},f_{202},f_{211},f_{212},f_{232},f_{235},f_{241},f_{243},f_{244},f_{246},f_{251} f_{254},f_{256},f_{258},f_{282},f_{292},f_{295},f_{313},f_{315},f_{329},f_{330},f_{331},f_{332},f_{336} f_{340},f_{348},f_{350},f_{335},f_{351},f_{312},f_{205},f_{236},f_{206}
Soccer Model	f_{15},f_{16},f_{21},f_{22},f_{31},f_{44},f_{46},f_{49},f_{53},f_{55},f_{56},f_{64},f_{65},f_{72},f_{76},f_{79},f_{85} f_{86},f_{87},f_{89},f_{95},f_{101},f_{102},f_{104},f_{105},f_{112},f_{114},f_{115},f_{124},f_{126} f_{130},f_{131},f_{132},f_{133},f_{136},f_{142},f_{154},f_{155},f_{156},f_{180},f_{181},f_{182} f_{203},f_{204},f_{223},f_{231},f_{232},f_{233},f_{256},f_{261}
Misc. Model	f_2,f_4,f_5,f_6,f_{11},f_{12},f_{15},f_{21},f_{22},f_{23},f_{24},f_{25},f_{29},f_{35},f_{42},f_{43},f_{45},f_{48} f_{57},f_{60},f_{61},f_{62},f_{65},f_{66},f_{71},f_{75},f_{81},f_{85},f_{86},f_{93},f_{95},f_{101} f_{102},f_{103},f_{104},f_{105},f_{106},f_{107},f_{124},f_{126},f_{127},f_{130},f_{133},f_{134} f_{139},f_{141},f_{142},f_{143},f_{145},f_{146}
Run Model	f_{44},f_{49},f_{63},f_{91},f_{96},f_{106},f_{131},f_{134},f_{135},f_{143},f_{145},f_{151},f_{152},f_{181},f_{183} f_{202},f_{212},f_{232},f_{301},f_{306},f_{318},f_{324},f_{342},f_{338},f_{326},f_{85},f_{22},f_{32},f_{193} f_{191},f_{112},f_{102},f_{125},f_{350},f_{263}, f_{273},f_{156},f_{126},f_{55},f_{124},f_{325},f_{346},f_{65} f_{105},f_{115},f_{123},f_{94}, f_5,f_{345},f_{341}

$$f(x) = \frac{H}{\left[1 + \left(\frac{2(x-x_0)\sqrt{2^{\frac{1}{\omega}}-1}}{\sigma}\right)^2\right]^\omega} \tag{10}$$

where H is the peak high at the center x_0, x is the independent variable being fit, σ is the half-width of the curve, ω is the tailing factor for the peak [36]. This curve fitting is particularly powerful because, by varying ω it can fit different distributions, such as Gaussian and Lorentzian [36]. From this, the kernel function for an SVM can be defined as a kernel of two vectors by

$$K(x,y) = \frac{1}{\left[1 + \left(\frac{2*\sqrt{\|x-y\|}\sqrt{2^{\frac{1}{\omega}}-1}}{\sigma}\right)^2\right]^\omega} \tag{11}$$

where a variable and its midpoint of the curve is replaced by the Euclidean distance of two vectors, and the height is simply set to 1. Work in [36] guarantees that this function matches the requirements for a kernel (symmetric matrix and real valued symmetrical and satisfies Mercer's conditions).

V. RESULTS

For proper evaluation of the algorithm's results, the Fine-Grain PCA [19] related work is compared. Further, both single and multiple model solutions are presented for both SVM with RBF Kernel and PUK Kernel in order to show the development of robust, fine-grain classifiers for large, multiclass problems, but also to compare the two kernel types against each other.

A. Features Extracted

From this, the same set of features are extracted and numbered, as in Table V. Thus, feature f_1 is the minimum value of the X axis of the $Gyro$ of the Left Arm. f_2 is the minimum value of the Y axis of the $Gyro$. The minimum features for the left arm are the first 10 features (3-axis gyro, 3 axis accelerometer, 4 channel quaternions). Then the next 10 features are the maximum, and so on. After all the feature types are extracted on the left arm, the right arm begins. Features $1-80$ are the left arm, $81-160$ the right arm, $161-240$ the left leg, $240-320$ the right leg, $320-328$ the magnitude of the left arm, $329-336$ the magnitude of the right arm, $335-344$ the magnitude of the left leg, and $344-352$ the magnitude of the right leg. The top 50 features for each cluster and model type are listed in Table V. Notice while many features are repeated each model has its own set of selected features.

B. Cross Validation

A leave-one-subject-out cross-validation was run in order to test the robustness of such an algorithm. The results of the system in comparison between RBF (LibSVM in Weka [28]) and PUK kernels are also compared. Table VI shows the results of running the multiple model scheme in the RBF Kernel. For this application, since the second level of modeling still contains different motions, the importance of each model is considered paramount. As a result, all of the α and β parameters are set to 1. From this, it is clear that the RBF model can differentiate between the three types of movements with high accuracy. While certain motion classification in each of the clusters might be lower, this method allows for high accuracy of many of the miscellaneous movements, as well as an overall performance gain over the single-model classifier.

Table VII shows the results of the same experiment with the PUK kernel. The results show several important factors. The first is that the single model classification scheme is significantly improved. Further, the soccer model and the run model both see significant improvements in classification. The overall multiple model classification scheme not only reaches a high level of accuracy, with an F-Score reaching as high as .97, but already outperforms the best versions of previous models with only 10 features. Figure 6 shows the mean f-scores across all users in the cross-validation for each examination type in response to the number of features used. The RBF kernel serves as more accurate than the PUK kernel for a small number of features, but is limited in it's separation ability and does not improve greatly with more features. Finally, these methods, at their best accuracy are compared with the Fine-Grain PCA algorithm adapted to this data set. As expected, the results of the Fine-Grain PCA degrade from the almost .80 F-Score reported in [19] to only .70 as shown in Table VIII, since the number of similar movements increases and,

TABLE VI. F-SCORES FOR THE SINGLE MULTICLASS CLASSIFIER, EACH MODEL OF THE MULTIPLE MODEL METHOD, AND THE OVERALL MULTIPLE MODEL METHOD FOR RBF KERNEL IN SVM

Model Features− >	1	5	10	15	20	25	30	35	40	45	50
Single Model	0.129	0.557	0.671	0.697	0.714	0.720	0.722	0.723	0.719	0.718	0.707
Top Model	0.877	0.980	0.983	0.984	0.987	0.987	0.987	0.987	0.986	0.984	0.979
Soccer Model	0.239	0.561	0.639	0.681	0.690	0.686	0.682	0.688	0.688	0.690	0.673
Misc. Model	0.604	0.976	0.989	0.993	0.994	0.994	0.994	0.993	0.99	0.991	0.986
Run Model	0.305	0.587	0.648	0.672	0.686	0.698	0.7	0.691	0.685	0.689	0.675
Multiple Model	0.490	0.790	0.831	0.850	0.857	0.858	0.857	0.857	0.855	0.856	0.845

TABLE VII. F-SCORES FOR THE SINGLE MULTICLASS CLASSIFIER, EACH MODEL OF THE MULTIPLE MODEL METHOD, AND THE OVERALL MULTIPLE MODEL METHOD FOR PUK KERNEL IN SVM

Model Features− >	1	5	10	15	20	25	30	35	40	45	50
Single Model	0.144	0.485	0.694	0.748	0.755	0.782	0.847	0.879	0.902	0.911	0.913
Top Model	0.595	0.843	0.963	0.981	0.994	0.996	0.995	0.996	0.996	0.996	0.995
Soccer Model	0.186	0.441	0.694	0.749	0.902	0.93	0.945	0.941	0.956	0.961	0.962
Misc. Model	0.447	0.858	0.941	0.963	0.972	0.972	0.985	0.992	0.996	0.997	0.993
Run Model	0.176	0.506	0.618	0.685	0.796	0.829	0.872	0.877	0.874	0.881	0.878
Multiple Model	0.343	0.668	0.828	0.868	0.937	0.951	0.963	0.964	0.968	0.970	0.969

TABLE VIII. TABLE SHOWING RESULTS HIGHEST ACHIEVED RESULTS OF [19] AND THE SVM ALGORITHMS SHOWN HERE

Algorithm	**F-Score**
Fine-Grain PCA	0.70
SVM (RBF) - Single	.723
SVM (RBF) - Multiple	.858
SVM (PUK) - Single	.913
SVM (PUK) - Multiple	.970

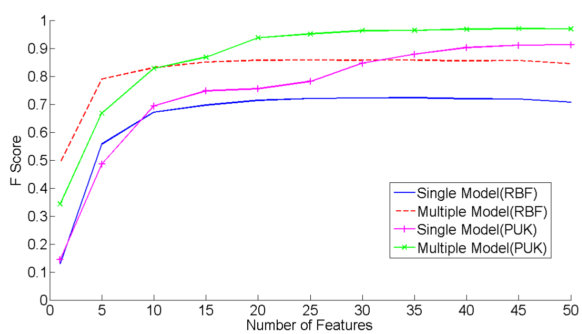

Fig. 6. Mean F-scores in cross-validation per feature of the single and multiple model SVM RBF and PUK kernels.

thus, the number of eigenvectors selected may not properly differentiate movements.

VI. FUTURE WORK

The multiple model approach developed shows a strong classification for fine-grain movements and the ability to improve the speed with which they are classified by reducing the number of features necessary for classification. However, what is considered the overall number of features used in a multiple model scheme? Is it the average number of features? The maximum? The intersection? What if features are extracted in parallel while the first level model is being run? Depending upon the implementation of the real-time aspects of the work, this metric will require different representation and, as such, is left for future investigation. Further, different models can be developed for different clusters (e.g. SVMs might be stronger for cluster a whereas Hidden Markov Models might be stronger for cluster b). How the clusters are selected, and models trained (perhaps per user) are all important questions. The user experience in such a game play should be tied with its long term ability to combat obesity, act accurately and with minimal perceived latency during continuous activity, and enjoyment have been left as future work for a prolonged clinical study that can validate effectiveness over a significant period of use. User's might enjoy something the first time they play it but find it cumbersome after months of usage.

VII. CONCLUSION

This paper presents the development of SoccAR, a soccer obstacle course mobile exergame. By using a head-worn display, each limb is available for movement, resulting in a large, multiclass environment for the internal recognition algorithm. This paper covered the use of contextual, expert knowledge to differentiate between the movement types in a large, multiclass environment. Further, by using an SVM, which is relatively quick in classifying information, the improvement over previous work will result in a quicker classification despite using more information than the Fine-Grain PCA method would allow. Using this information, a multiple model-based supervised classifier is created. This classification scheme shows improvement over the standard single model, showing that such a large, multiclass environment needs to have such a system to be as robust and generalized. The F-score for the multiple model approach reaches as high as .97 using the defined metrics. Further, the PUK kernel is shown to be adaptable to human activity just as well as it had to other types of classification work. It is possible to extend such work to using different kinds of models at each level, be it SVMs or hidden markov models. Further, multiple levels can be further investigated, but two levels were chosen in this work for its applicability to SoccAR in terms of recognizing movements within a desired framework for the game. The recognition algorithm presented, as well as the importance factors, allow for adapting systems and results based upon the accuracy, importance, and real-time responsiveness.

REFERENCES

[1] S. Wu and G. A.P., "Projection of chronic illness prevalence and cost inflation," 2000.

[2] F. W. Booth, C. K. Roberts, and M. J. Laye, "Lack of exercise is a major cause of chronic diseases," *Comprehensive Physiology*, 2012.

[3] R. Guthold, T. Ono, K. L. Strong, S. Chatterji, and A. Morabia, "Worldwide variability in physical inactivity: a 51-country survey," *American journal of preventive medicine*, vol. 34, no. 6, pp. 486–494, 2008.

[4] N. A. Garrett, M. Brasure, K. H. Schmitz, M. M. Schultz, and M. R. Huber, "Physical inactivity: direct cost to a health plan," *American journal of preventive medicine*, vol. 27, no. 4, pp. 304–309, 2004.

[5] E. A. Finkelstein, O. A. Khavjou, H. Thompson, J. G. Trogdon, L. Pan, B. Sherry, and W. Dietz, "Obesity and severe obesity forecasts through 2030," *American journal of preventive medicine*, vol. 42, no. 6, pp. 563–570, 2012.

[6] P. A. Heidenreich, J. G. Trogdon, O. A. Khavjou, J. Butler, K. Dracup, M. D. Ezekowitz, E. A. Finkelstein, Y. Hong, S. C. Johnston, A. Khera *et al.*, "Forecasting the future of cardiovascular disease in the united states a policy statement from the american heart association," *Circulation*, vol. 123, no. 8, pp. 933–944, 2011.

[7] A. D. Association *et al.*, "Economic costs of diabetes in the u.s. in 2012. diabetes care 2013; 36: 1033–1046," *Diabetes Care*, vol. 36, no. 6, p. 1797, 2013.

[8] E. Grieve, E. Fenwick, H.-C. Yang, and M. Lean, "The disproportionate economic burden associated with severe and complicated obesity: a systematic review," *Obesity Reviews*, vol. 14, no. 11, pp. 883–894, 2013.

[9] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 1, pp. 20–26, 2008.

[10] M. Buman, J. Kurka, E. Winkler, P. Gardiner, E. Hekler, G. Healy, N. Owen, C. Baldwin, and B. Ainsworth, "Estimated replacement effects of accelerometer-derived physical activity and self-reported sleep duration on chronic disease biomarkers," *Journal of Science and Medicine in Sport*, vol. 15, p. S76, 2012.

[11] A. Whitehead, H. Johnston, N. Nixon, and J. Welch, "Exergame effectiveness: what the numbers can tell us," in *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*. ACM, 2010, pp. 55–62.

[12] G. Alankus, A. Lazar, M. May, and C. Kelleher, "Towards customizable games for stroke rehabilitation," in *Proceedings of the 28th international conference on Human factors in computing systems*. ACM, 2010, pp. 2113–2122.

[13] T. Park, I. Hwang, U. Lee, S. I. Lee, C. Yoo, Y. Lee, H. Jang, S. P. Choe, S. Park, and J. Song, "Exerlink: enabling pervasive social exergames with heterogeneous exercise devices," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 15–28.

[14] K. Gerling, I. Livingston, L. Nacke, and R. Mandryk, "Full-body motion-based game interaction for older adults," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1873–1882.

[15] A. P. Macvean, "Developing adaptive exergames for adolescent children," in *Proceedings of the 11th International Conference on Interaction Design and Children*. ACM, 2012, pp. 339–342.

[16] A. Koivisto, S. Merilampi, and K. Kiili, "Mobile exergames for preventing diseases related to childhood obesity," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*. ACM, 2011, p. 29.

[17] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.

[18] B. Mortazavi, N. Alsharufa, S. I. Lee, M. Lan, M. Sarrafzadeh, M. Chronley, and C. K. Roberts, "Met calculations from on-body accelerometers for exergaming movements," in *Body Sensor Networks (BSN), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[19] B. Mortazavi, S. Nyamathi, S. Lee, T. Wilkerson, H. Ghasemzadeh, and M. Sarrafzadeh, "Near-realistic mobile exergames with wireless wearable sensors," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, no. 2, pp. 449–456, March 2014.

[20] N. Alshurafa, W. Xu, J. Liu, M. C. Huang, B. Mortazavi, C. Roberts, and M. Sarrafzadeh, "Designing a Robust Activity Recognition Framework for Health and Exergaming using Wearable Sensors," *IEEE Journal of Biomedical and Health Informatics*, Oct 2013.

[21] N. Alshurafa, J.-A. Eastwood, M. Pourhomayoun, S. Nyamathi, L. Bao, B. Mortazavi, and M. Sarrafzadeh, "Anti-cheating: Detecting self-inflicted and impersonator cheaters for remote health monitoring systems with wearable sensors," in *BSN*, 2014, pp. 92–97.

[22] J. Hu and N. V. Boulgouris, "Fast human activity recognition based on structure and motion," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1814–1821, 2011.

[23] J. Xu, Y. Sun, Z. Wang, W. Kaiser, and G. Pottie, "Context guided and personalized activity classification system," in *Proceedings of the 2nd Conference on Wireless Health*. ACM, 2011, p. 12.

[24] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 6, pp. 961–974, 2012.

[25] T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Efficient duration and hierarchical modeling for human activity recognition," *Artificial Intelligence*, vol. 173, no. 7, pp. 830–856, 2009.

[26] M. Pourhomayoun, N. Alshurafa, B. Mortazavi, H. Ghasemzadeh, K. Sideris, B. Sadeghi, M. Ong, L. Evangelista, P. Romano, A. Auerbach, A. Kimchi, and M. Sarrafzadeh, "Multiple model analytics for adverse event prediction in remote health monitoring systems," in *Healthcare Innovation Point-of-Care Technologie (EMBS HIPT), 2014 IEEE Conference on*. IEEE, 2014.

[27] "Shimmer3 wireless sensor platform," http://www.shimmersensing.com/images/uploads/docs/Shimmer3_Spec_Sheet_-_Jan_2014.pdf, [Online; last-accessed 31-March-2014].

[28] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[30] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1541.

[31] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," *Pervasive Computing*, pp. 1–17, 2004.

[32] L. S. Prichep, A. Jacquin, J. Filipenko, S. G. Dastidar, S. Zabele, A. Vodencarevic, and N. S. Rothman, "Classification of traumatic brain injury severity using informed data reduction in a series of binary classifier algorithms," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 20, no. 6, pp. 806–822, 2012.

[33] B. Mortazavi, S. I. Lee, and M. Sarrafzadeh, "User-centric exergaming with fine-grain activity recognition: A dynamic optimization approach," in *Ubicomp Workshop on Smart Health Systems and Applications, 2014 ACM Joint International Conference on Pervasive and Ubiquitous Computing (UbiComp 2014)*. ACM, 2014.

[34] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Conditional models for contextual human motion recognition," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 210–220, 2006.

[35] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2004–2011.

[36] B. Üstün, W. J. Melssen, and L. M. Buydens, "Facilitating the application of support vector regression by using a universal pearson vii function based kernel," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 29–40, 2006.

[37] G. Zhang and H. Ge, "Support vector machine with a pearson vii function kernel for discriminating halophilic and non-halophilic proteins," *Computational biology and chemistry*, vol. 46, pp. 16–22, 2013.

[38] A. Doswald, F. Carrino, and F. Ringeval, "Advanced processing of semg signals for user independent gesture recognition," in *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013*. Springer, 2014, pp. 758–761.