ISSN 2395-1621

A Survey on Human Activity Recognition

#1Puja B. Patil, #2Mrs. D. B. Gothwal

1pujabpatil@gmail.com
2dgohil.1519@gmail.com

#12Department of Computer Engineering

DYPCOE, Akurdi, Pune

ABSTRACT

With the rapid development of mobile devices and pervasive computing Technologies, acceleration-based human activity recognition, a difficult yet essential problem in mobile apps, has received intensive attention recently. Activity recognition fits within the bigger framework of context awareness. In this paper, we report on our efforts to recognize user activity from accelerometer data. The key benefits of using the smartphone accelerometer for human mobility analysis, with or without location determination based upon GPS, Wi-Fi or GSM is that it is energy-efficient, provides real-time contextual information and has high availability. Using measurements from an accelerometer for human mobility analysis presents its own challenges as we all carry our smartphones differently and the measurements are body placement dependent. Activity recognition is formulated as a classification problem. Performance of base-level classifiers and meta-level classifiers is compared

Index Terms: Accelerometer, Classifiers, Human activity recognition, Smartphone

ARTICLE INFO

Article History

Received: 8th January 2017

Received in revised form :

8th January 2017

Accepted: 10th January 2017

Published online :

11th January 2017

I. INTRODUCTION

Human activity recognition receives more attentions in recent years, due to many applications, such as video surveillance[27], [28], health care [29], [30], and context-aware computing[31], [32]. In general, pattern recognition schemes can directly handle the samples which are represented in a vector space. In most neural networks system, such as character recognition[33] and traffic signs recognition[34] , the samples can be easily converted into feature vectors after normalizing the size of the input images.

Existing human activity recognition systems, which can be classified into two groups: computer vision-based systems and accelerometer-based systems[32], often suffer from the diffi-culty of effective feature representation and selection. Com-puter vision-based human activity recognition systems[35] perform poorly because of the variations of illumination con-ditions in sophisticated environments and motion blur.

Mobility awareness and mobility based services (MBS), in contrast to position or location based services (LBS) focus on mobility in the sense of how someone or something moves in the physical world to a pre-planned destination and covers ad hoc movement away from the current location. The emphasis is on the type of the mobility rather than on the location context, however these two may be combined in a complementary manner. Mobility can be characterized at a low level as the rate of change of location or position in (x, y, z) directions and velocity with respect to time. At a higher level of abstraction, mobility represents an associated human mobility type of activity such as being stationary versus walking. Mobility patterns of acceleration can be used to determine the travel or transportation mode of the user, i.e., the user is in a moving vehicle versus walking. Wearable devices are getting more and more popular recently, which presents a convenient and portable way to record physiological data from users. Thus it is more possible to collect physical-related data and perform further analysis. All these data will be generated and made available to better understand the activities users are performing in and could be used to monitor health or Re-creational activities.

A. Computer vision-based human activity recognition

Computer vision-based human activity recognition systems[35] decomposed images into a variable number of parts using interest point/region detectors, and then some feature descriptors, such as speed up robust features[36], scale invariant feature transform[37], and local binary patterns[38], are used to represent these parts. The conventional pattern recognition algorithms cannot apply. Spatial pyramid matching [29] and locality-constrained linear coding [46] are proposed to address this problem by utilizing efficient coding schemes. Using dissimilarity-based representation concepts, Carli et al. [10] proposed a common framework to obtain robust performance.

Computer vision-based human activity recognition systems [24] perform poorly because of the variations of illumination conditions in sophisticated environments and motion blur.

B. Accelerometer-based human activity recognition

Accelerometer-based human activity recognition [28], [43], [47] is an important alternative, which has received increasing attentions due to the popularity of smart phones in recent years. It exploits acceleration signals from smart phones equipped with the accelerometer to analyze and recognize daily human activities such as standing, walking and running. This approach has largely increased the possibility to reach the dream of pervasive computing. Recently, Kwapisz et al. [28], Tentori and Favela [43], and Wang et al. [47] have made significant progress in mobile music recommendation and mobile activity monitor by utilizing a set of related technologies. Before we apply learning models to the acceler-ation signals, there is a problem to be noticed, i.e., different acceleration signals for representing different activities or even the same activity with different attributes that could cause trouble in aligning time-axis and determining the length of signals. However, it is a standard scheme [2], [28], [45] to use fixedlength signal segments to represent a human activity.

Accelerometer based human activity recognition systems [28] cannot perform well because the sensor positions are not fixed.

II. **REVIEW OF LITERATURE**

Energy-Efficient Human Mobility Sensing(EHMS)[1] in-volves the study of the generated user accelerometer patterns and design of a probabilistic algorithm with a high accuracy without the need for noise filtering and specific on-body place-ment. To achieve real-time human mobility state by performing all calculations locally within the smartphone. EHMS able to accurately classify accelerometer data from a specialized subset of human mobility states including stationary with no movement e.g., smartphone resting on a table, stationary with slight movements (sitting, lying down, and standing) and in-motion (walking, jogging, cycling, motorized movement including travel by bus, light rail train, underground train, taxi, and car).

Accelerometer augmented mobile phone localization (AAMPL)[2] which detects a users movement using the mobile phone accelerometer and in-turn places the mobile phone in the right context. The AAMPL framework acquires the approximate physical location of a mobile phone, and augments it with a context-aware logical localization. Evaluation of AAMPL on Nokia N95 phones shows that it is able to correctly display physical locations determined using the phones GPS on Google Maps.

Energy efficient mobile sensing system (EEMSS)[3] which uses the embedded mobile phone sensors to recognize hu-man mobility states and to detect state transitions. It uses a combination of sensor readings from an accelerometer, Wi-Fi, GPS, and a microphone to automatically recognize the human mobility state defined in three dimensions: motion (such as running or walking), location (such as staying at home or in a motorized vehicle on a freeway) and background environment (such as a loud or quiet location). EEMSS has the ability to detect the human mobility state and transition using low-energy mobile phone sensors.

EnTracked [4] can track mobile devices robustly and energy-efficiently. EnTracked can reduce power consumption and guarantee robustness by calculating an optimal plan, using an accelerometer to decide when to turn on and off sensors such as the GPS. This architecture uses both an accelerometer and GPS to detect a change in the human mobility state.

Reddy, M. Mun,[5] uses a mobile phone with a built-in GPS receiver and accelerometer to detect transportation modes. The transportation modes identified include stationary, walking, running, biking, or motorized transport. They found that a combination of multiple algorithms can lead to higher transport mode detection accuracy.

Khan et al.[6] show that kernel discriminant analysis (KDA) outperforms linear discriminant analysis (LDA) at improving class separation in terms of accuracy. They also use the signal magnitude area to differentiate between static and dynamic activities using 3D accelerometer signals.

Nick et al. [7] classified transportation modes using ac-celerometer data. The transport modalities of interest are: travel by car, train and pedestrians. The results showed an higher accuracy using a one-against-one or one-against-all support vector machine as compared to using a naive Bayes (NB) classifier.

EnLoc [8] developd an energy-efficient localization frame-work called EnLoc. The framework characterizes the optimal localization accuracy for a given energy budget, and develops prediction based heuristics for real-time use. Evaluation on traces from real users demonstrates the possibility of achieving good localization accuracy for a realistic energy budget.

Ling Bao[9] developed an algorithms and evaluated to detect physical activities from data acquired using five small biaxial accelerometers worn simultaneously on different parts of the body.

Nishkam Ravi[10] recognizes user activity from accelerom-eter data. Author compared the Performance of base-level classifiers and meta-level classifiers and plurality Voting is found to perform consistently well across different settings.

Moustafa Youssef et al.[11] proposed GAC: a hybrid GPS/accelerometer/compass scheme that depends mainly on using the low-energy accelerometer and compass sensors and uses the GPS infrequently for synchronization.

III. HAR CLASSIFICATION METHODS

A. Decision Tree

Decision trees build a hierarchical model in which attributes are mapped to nodes and edges represent the possible attribute values. Each branch from the root to a leaf node is a classifi-cation rule. C4.5 is perhaps the most widely used decision tree classifier and is based on the concept of information gain to select the attributes that should be placed in the top nodes[12]. Decision trees can be evaluated in O(log n) for n attributes, and usually generate models that are easy to understand by humans.

The decision tree algorithm used in this work can recognize the content of activities, but may not readily recognize activity style. Although a decision tree algorithm could potentially recognize activity style using a greater number of labels such as “walking slowly”, “walking briskly”, “scrubbing softly”, or “scrubbing vigorously”, the extensibility of this technique is limited. Recognition accuracy is highest for decision tree classifiers, which is consistent with past work where decision based algorithms recognized lying, sitting, standing and loco-motion with 89.30% accuracy[13]

The binary decision tree algorithms with simple threshold rules can be implemented as an if-else structure[14]. The simplified decision tree algorithm implemented in the online system is shown in Fig 1.

B. Bayesian

Bayesian methods calculate posterior probabilities for each class using estimated conditional probabilities from the train-

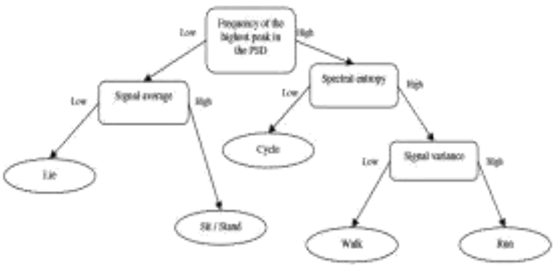

Fig. 1. Decision tree structure used in the classification of the activities[14]

ing set. The Bayesian Network (BN) classifier and Naive Bayes (NB) are the principal exponents of this family of classifiers. A key issue in Bayesian Networks is the topology construction, as it is necessary to make assumptions on the independence among features. For instance, the NB classifier assumes that all features are conditionally independent given a class value, yet such assumption does not hold in many cases. As a matter of fact, acceleration signals are highly correlated, as well as physiological signals such as heart rate, respiration rate, and ECG amplitude[12].

The fitting of probability distributions to acceleration fea-tures under a Naive Bayesian approach may be unable to adequately model such rules due to the assumptions of con-ditional independence between features and normal distribu-tion of feature values, which may account for the weaker performance. Furthermore, Bayesian algorithms may require more data to accurately model feature value distributions[13]. Naive Bayesian(NB) is a common method used to solve classification problems, since it has the ability to work with a large amount of data with low computational power demand [15]. This method uses the probability information residing in the training data in order to find the maximum probability of a new hypothesis using the Bayes rule. For simplification, equal class probability and independency between features were supposed.

In[17], Naive Bayes used as a baseline model to get a rough idea of how is the training process going. In a bayes classifier, assign a class label y^ = Ck for some k as follows:

$$\hat{y} = argmax_{k \in \{1....k\}} p(C_k) \pi_{i=1}^{n} p(x_i|C_k)$$

C. k-nearest neighbors

k-nearest neighbors(KNN)[15][16] methods classify an in-stance based upon the most similar instances in the training set. For that purpose, they define a distance function to measure similarity between each pair of instances. This makes k-nearest neighbors classifiers quite expensive in their evaluation phase as each new instance to be classified needs to be compared to the entire training set. Such high cost in terms of computation and storage, makes KNN models not convenient to be implemented in a mobile device.

This method uses all feature vectors of the training dataset, instead of only one reference for every class[15]. The distance is calculated between the feature vector to be classified and all the references. The classification is done according to how many references of certain class are nearest. Usually, k-Nearest

Neighbor improves the results since it employs more data for comparison. However, this method demands more memory space to store all reference training vectors[15].

In[17], a positive integer K and a test observation x0, the KNN classifier first identifies the neighbors K points in the training data that are closest to x0, represented by N0. It then estimates the conditional probability for class j as the fraction of points in N0 whose response values equal j:

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

D. Multi-Layer Perceptron

Multi-Layer Perceptron(MLP)[17][18][22] can be seen as a fully connected neural network with one input layer, one hidden layer and one output layer as shown in Fig. 2. A one-hidden-layer MLP can be formalized as follow: It is a function f, $R^D \longrightarrow R^L$., where D is the size of input vector x and L is the size of the output vector f(x), such that, in matrix notation:

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x)))$$

with bias vectors $b^{(1)}$ and $b^{(2)}$; weight matrices $W^{(1)}$ and $W^{(2)}$ and activation functions G and s[17].

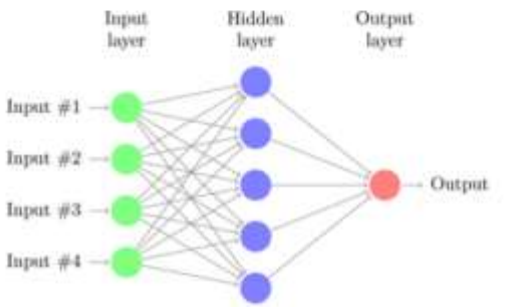

Fig. 2. Structure of MLP[17]

E. Support Vector Machines

Support Vector Machines (SVM)[21] and Artificial Neural Networks (ANN)[22] have also been broadly used in HAR although they do not provide a set of rules understandable by humans. Instead, knowledge is hidden within the model, which may hinder the analysis and incorporation of additional reasoning. SVMs rely on kernel functions that project all instances to a higher dimensional space with the aim of finding a linear decision boundary (i.e., a hyperplane) to partition the data. Neural networks replicate the behavior of biological neurons in the human brain, propagating activation signals and encoding knowledge in the network links. Besides, ANNs have been shown to be universal function approximators. The high computational cost and the need for large amount of training data are two common drawbacks of neural networks.

Anguita et al. in [19]introduced the concept of a Hardware-Friendly SVM (HF-SVM). This method exploits fixed-point arithmetic in the feed-forward phase of the SVM classifier, so as to allow the use of this algorithm in hardware-limited devices.

Support Vector Machines[20] is a supervised learning al-gorithm that classifies objects based on the support vectors of a dataset or points lie closest to the decision boundary. SVM maximize the distance between support vectors and the decision boundary. In [20]the multi class Hardware-Friendly SVM (MC-HF-SVM) approach is proposed which is used for AmI systems for healthcare applications such as activity monitoring on smart-phones. This alternative that employs fixed-point calculations, can be used for AR because it requires less memory, processor time and power consumption.

F. Hidden Markov Models

Hidden Markov Models (HMMs) have also been used for physical activity recognition[23]. Since HMMs can model transitions between physical activities, they are more suitable for segmenting a time series into a sequence of physical activity types rather than classifying an entire time series as a single activity type.

HMMs have been successfully used in modeling different types of time-series data, e.g. in speech recognition, gesture tracking etc. HMMs is used to capture the temporal dynamics, but instead of directly using the raw features. In [23] HMMs trained using the posterior probabilities of the static classifiers. The advantage of using the posterior probabilities is that it can take advantage of the results from the discriminatively trained classifier, as well as reduce the complexity of the HMMs.

G. Boosting

Boosting[24] is used to improve the classification accuracy of any given base-level classifier. Boosting applies a single learning algorithm repeatedly and combines the hypothesis learned each time, such that the final classification accuracy is improved. It does so by assigning a certain weight to each example in the training set, and then modifying the weight after each iteration depending on whether the example was correctly or incorrectly classified by the current hypothesis. Thus final hypothesis learned can be given as

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Where, t denotes the coefficient with which the hypothesis ht is combined. Both t and ht are learned during the Boosting procedure.

H. Bagging

Bagging[25] is another simple meta-level classifier that uses just one base-level classifier at a time. It works by training each classifier on a random redistribution of the training set. Thus, each classifiers training set is generated by randomly drawing, with replacement, N instances from the original training set. Here N is the size of the original training set itself. Many of the original examples may be repeated in the resulting training set while others may be left out. The final bagged estimator, hbag(.) is the expected value of the prediction over each of the trained hypotheses. If hk(.) is the hypothesis learned for training sample k,

$$h_{bag}(.) = \frac{1}{M}\sum_{k=1}^{M} h_k(.).$$

IV. RESULT ANALYSIS

In [26] the performance of the different classifiers, Multi-layer Perceptron, Random Forest, LMT, SVM, Simple Logistic and LogitBoost are compared. Classifiers were trained and tested using a 10-fold cross validation method on the set of extracted features. The summary results for activity recognition experiments are presented in Table 1 for two datasets: phone in hand position and phone in pocket position. Over-all, Multilayer Perceptron offered the highest performance, yielding 89.48% accuracy for in-hand position and 89.72% accuracy for in-pocket position. SVM was the second most accurate for in-hand position. Its good performance has been supported by prior research on human activity recognition tasks. However, Our results also showed that Random Forest demonstrated high accuracy for both cases. Table 1 indicates that carrying the phone in pocket or in hand produced similar results. Hence, multilayer perceptron recognition method is robust to smartphone position.

TABLE I
CLASSIFICATION ACCURACY FOR DIFFERENT CLASSIFIERS ON TWO TRAINING DATASETS (PHONE IN HAND, PHONE IN POCKET).

Classifier	Accuracy (in-hand)	Accuracy (in-pocket)
Multilayer Perceptron	89.48%	89.72%
SVM	88.76%	72.27%
Random Forest	87.55%	85.15%
LMT	85.89%	85.07%
Simple Logistic	85.41%	85.04%
Logit Boost	82.54%	82.24%

V. PERFORMANCE METRICS

In general, the selection of the classification algorithm for HAR has been merely supported by empirical evidence. The vast majority of the studies use cross validation with statis-tical tests to compare classifiers performance for a particular dataset. The classification results for a particular method can be organized in a confusion matrix Mnn for a classification problem with n classes. This is a matrix such that the element Mij is the number of instances from class i that were actually classified as class j. The following values can be obtained from the confusion matrix in a binary classification problem:

- True Positives (TP): The number of positive instances that were classified as positive.
- True Negatives (TN): The number of negative instances that were classified as negative.
- False Positives (FP): The number of negative instances that were classified as positive.
- False Negatives (FN): The number of positive instances that were classified as negative.

The accuracy is the most standard metric to summarize the overall classification performance for all classes and it is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The precision, often referred to as positive predictive value, is the ratio of correctly classified positive instances to the total number of instances classified as positive:

$$Precision = \frac{TP}{TP + FP}$$

The recall, also called true positive rate, is the ratio of correctly classified positive instances to the total number of positive instances:

$$Recall = \frac{TP}{TP + FN}$$

The F-measure combines precision and recall in a single value:

$$Fmeasure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

(a) Precision comparison of EHMS vs. existing classifiers. (b) Recall comparison of EHMS vs. existing classifiers.

Fig. 3. Precision and Recall comparison with different classifies[1]

In fig. 3 the classifiers are J48, decision table (DT), bag-ging, naive Bayes and EHMS shows the precision and recall comparison of EHMS vs. known existing classifiers[1].

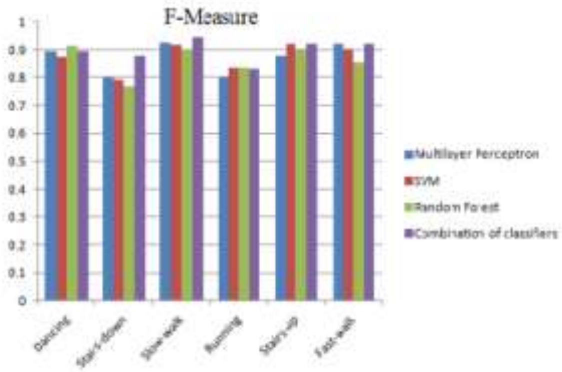

Fig. 4. F-measure for human activity using four classifiers: in-hand phone position[26]

In [26], F-measure is calculated for human activity using four classifiers i.e multilayer perceptron, SVM, Random forest and combination of classifiers.

VI. CONCLUSION

This paper is a survey of the current research on hu-man activity recognition. Human activity recognition receives intensive attentions in recent years, due to many practical applications, such as video surveillance, health care, and context-aware computing. Accelerometer data is collected and from that features are extracted and by using classifier human activity is classified. There are different classifiers are used like, decision tree, bayesian network, multilayer perceptron, SVM etc. EHMS able to accurately classify accelerometer data from a specialized subset of human mobility states.

ACKNOWLEDGMENT

The authors would like to thank the publishers and researchers for making their resources available. We also thank the college authority for providing the required infrastructure and support. Finally we would like to extend our heartfelt gratitude to friends and family members.

REFERENCES

[1] Thomas Olutoyin Oshin, et al. “Energy-Efficient Real-Time Human Mobility State Classification Using Smartphones ”, IEEE Transaction on Computers, VOL. 64, NO. 6, JUNE 2015

[2] Ofstad, Andrew, et al. “AAMPL: accelerometer augmented mobile phone localization,”Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. ACM, 2008.

[3] Wang, Yi, et al. “A framework of energy efficient mobile sensing for automatic user state recognition,”Proceedings of the 7th international conference on Mobile systems, applications, and services. ACM, 2009.

[4] Jakob Langdal Jensen et al. “EnTracked: Energy-efficient robust position tracking for mobile devices,”Researchgate 2009

[5] Reddy, Sasank, et al. “Using mobile phones to determine transportation modes”ACM Transactions on Sensor Networks (TOSN) 6.2 (2010): 13.

[6] A. M. Khan, Y.-K. Lee, S. Y. Lee “Human Activity Recognition via An Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis,”, IEEE 2010

[7] Nick et al.“Classifying means of transportation using mobile sensor data,”

[8] Constandache, Ionut, et al. “Enloc: Energy-efficient localization for mobile phones,”INFOCOM 2009, IEEE. IEEE, 2009.

[9] Bao, Ling, and Stephen S. Intille. “Activity recognition from user-annotated acceleration data,”International Conference on Pervasive Com-puting. Springer Berlin Heidelberg, 2004.

[10] Ravi, Nishkam, et al. “Activity recognition from accelerometer data,”AAAI. Vol. 5. 2005.

[11] Youssef, Moustafa, Mohamed Amir Yosef, and Mohamed El-Derini. “GAC: energy-efficient hybrid GPS-accelerometer-compass GSM local-ization,”Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE. IEEE, 2010.

[12] Lara, Oscar D., and Miguel A. Labrador. “A survey on human activity recognition using wearable sensors,”IEEE Communications Surveys & Tutorials 15.3 (2013): 1192-1209.

[13] Bao, Ling, and Stephen S. Intille. “Activity recognition from user-annotated acceleration data,”International Conference on Pervasive Com-puting. Springer Berlin Heidelberg, 2004.

[14] Ermes, Miikka, Juha Parkka, and Luc Cluitmans. “Advancing from offline to online activity recognition with wearable sensors”, 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2008.

[15] Jatoba, Luciana C., et al. “Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity”, 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2008.

[16] Maurer, Uwe, et al. “Activity recognition and monitoring using multiple sensors on different body positions”, International Workshop on Wear-able and Implantable Body Sensor Networks (BSN’06). IEEE, 2006.

[17] Wu, Zhongyan, Shutong Zhang, and Chenying Zhang. “Human Activity Recognition using Wearable Devices Sensor Data”.

[18] Randell, Cliff, and Henk Muller. “Context awareness by analysing accelerometer data”, Wearable Computers, The Fourth International Symposium on. IEEE, 2000.

[19] Anguita, Davide, et al. “A Hardware-friendly Support Vector Machine for Embedded Automotive Applications”. IJCNN. 2007.

[20] Anguita, Davide, et al. “Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine”, Interna-tional Workshop on Ambient Assisted Living. Springer Berlin Heidel-berg, 2012.

[21] Cortes, Corinna, and Vladimir Vapnik. “Support-vector networks”, Ma-chine learning 20.3 (1995): 273-297.

[22] Gallant, Stephen I. “Perceptron-based learning algorithms”, IEEE Trans-actions on neural networks 1.2 (1990): 179-191.

[23] Lester, Jonathan, et al. “A hybrid discriminative/generative approach for modeling human activities”, (2005): 766-722.

[24] Meir, Ron, and Gunnar Rtsch. “An introduction to boosting and leveraging”, Advanced lectures on machine learning. Springer Berlin Heidelberg, 2003. 118-183.

[25] Breiman, Leo. “Bagging predictors”, Machine learning 24.2 (1996): 123-140.

[26] Bayat, Akram, Marc Pomplun, and Duc A. Tran. “A study on human ac-tivity recognition using accelerometer data from smartphones”, Procedia Computer Science 34 (2014): 450-457.

[27] Shao, Ling, et al. “Human action segmentation and recognition via motion and shape analysis”, Pattern Recognition Letters 33.4 (2012): 438-445.

[28] Tosato, Diego, et al. “Characterizing humans on riemannian manifolds”, IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1972-1984.

[29] Tentori, Monica, and Jesus Favela. “Activity-aware computing for healthcare”, IEEE Pervasive Computing 7.2 (2008): 51-57.

[30] Osmani, Venet, Sasitharan Balasubramaniam, and Dmitri Botvich. “Hu-man activity recognition in pervasive health-care: Supporting efficient remote collaboration”, Journal of network and computer applications 31.4 (2008): 628-655.

[31] Wang, Xinxi, David Rosenblum, and Ye Wang. “Context-aware mobile music recommendation for daily activities”, Proceedings of the 20th ACM international conference on Multimedia. ACM, 2012.

[32] Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. “Activity recognition using cell phone accelerometers”, ACM SigKDD Explo-rations Newsletter 12.2 (2011): 74-82.

[33] Keysers, Daniel, et al. “Deformation models for image recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence 29.8 (2007): 1422-1435.

[34] CireAn, Dan, et al. “Multi-column deep neural network for traffic sign classification”, Neural Networks 32 (2012): 333-338.

[35] Iosifidis, Alexandros, Anastasios Tefas, and Ioannis Pitas. “View-invariant action recognition based on artificial neural networks”, IEEE transactions on neural networks and learning systems 23.3 (2012): 412-424.

[36] Bay, Herbert, et al. ”Speeded-up robust features (SURF).” Computer vision and image understanding 110.3 (2008): 346-359.

[37] Lowe, David G. ”Distinctive image features from scale-invariant key-points.” International journal of computer vision 60.2 (2004): 91-110.

[38] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. ”Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.” IEEE Transactions on pattern analysis and machine intelli-gence 24.7 (2002): 971-987.