A wearable and non-wearable approaches for gesture recognition – Initial results

Vinícius Silva[1], João Ramos[2], Filomena Soares[1], Paulo Novais[2], Pedro Arezes[3]
[1]Algoritmi R&D, Department of Industrial Electronics, University of Minho, Guimaraes, Portugal
[2]Algoritmi R&D, Department of Informatics University of Minho, Braga, Portugal

Carina Figueira[4], Joana Silva[4], António Santos[4], and Filipe Sousa[4]
[3]Algoritmi R&D, Department of Production Systems, University of Minho, Guimarães, Portugal
[4]Fraunhofer Portugal Research Centre, Porto, Portugal

***Abstract*— A natural way of communication between humans are gestures. Through this type of non-verbal communication, the human interaction may change since it is possible to send a particular message or capture the attention of the other peer. In the human-computer interaction the capture of such gestures has been a topic of interest where the goal is to classify human gestures in different scenarios. Applying machine learning techniques, one may be able to track and recognize human gestures and use the gathered information to assess the medical condition of a person regarding, for example, motor impairments. According to the type of movement and to the target population one may use different wearable or non-wearable sensors. In this work, we are using a hybrid approach for automatically detecting the ball throwing movement by applying a Microsoft Kinect (non-wearable) and the *Pandlet* (set of wearable sensors such as accelerometer, gyroscope, among others). After creating a dataset of 10 participants, a SVM model with a DTW kernel is trained and used as a classification tool. The system performance was quantified in terms of confusion matrix, accuracy, sensitivity and specificity, Area Under the Curve, and Mathews Correlation Coefficient metrics. The obtained results point out that the present system is able to recognize the selected throwing gestures and that the overall performance of the Kinect is better compared to the *Pandlet*.**

Keywords—SVM; DTW; Kinect; Pandlet; Activity monitoring.

I. INTRODUCTION

Gestures have been employed naturally in every human interaction. They are a form of non-verbal communication which is visible bodily actions to communicate particular messages or used in purely displays of joint attention. The classification of human gestures in different contexts have been a main topic in the fields of Human-Computer Interaction (HCI), as well as in the medical and health field. The detection of the movement of human body segments plays a key role in several health applications such as the diagnosis of several neurological disorders, the rehabilitation of patients with motor function impairments, the remote monitoring of elders, among others.

Currently, the main approaches used for motion monitoring applications are marker-based [1], [2] or marker-less [3], [4] systems and wireless wearable devices integrating accelerometers, gyroscopes, magnetometers, or other sensors units [5]. Recognizing gestural information requires the extraction of meaningful patterns from the gathered data.

Following this trend, there has been an extensive research on automatic gesture recognition [6]. Most of them use machine learning techniques to detect such patterns, achieving successful results in the literature [7]– [10].

The system proposed by S. Amendola *et.al.* [7] focus on recognizing arms and legs gestures by using only passive and sensor-less transponders. First, the electromagnetic signal, backscattered from the RFID tags during gestures, are collected by a fixed reader antenna. Then, the extracted data is used as input to a Support Vector Machine (SVM) classifier with the intention to recognize both periodic limbs movements as well as to classify more complex random motions patterns. Finally, experimental sessions were conducted in order to assess the system performance, obtaining a classification accuracy higher than 80%.

Considering wearable devices, D. Wilson and A. Wilson [8] presented a device, the *XWand*, composed of a 2-axis accelerometer, a 3-axis magnetometer, a 1-axis gyroscope, among others. This device was used to examine three main approaches to gestures recognition – linear time warping (LTW) method, dynamic time warping (DTW) method, and hidden Markov model based method (HMM). The different models were trained with 420 examples of 7 main gestures. In order to evaluate the defined approaches, from the initial data-set, 294 examples were selected to test the algorithms. The model that achieved the highest performance was the HMM model with an accuracy of 90.43%, followed by the DTW model with an accuracy of 71.64% and the LTW model with an accuracy of 40.42%.

J. Wu *et.al.* [9] proposed an approach for gesture recognition that uses data from a 3-dimenssional accelerometer. Firstly, the acceleration data of a gesture is collected and represented by a frame-based descriptor, to extract discriminative information. Then a SVM multi-class gesture classifier is trained with a data set containing 3360 gesture samples of 12 gestures, for recognizing nonlinear gestures. Finally, in order to evaluate their approach, it was conducted both user-dependent experiments and the user-independent ones. In the user-dependent case, the recognition accuracy was 95.21% when recognizing all 12 gestures. In the user-independent case, their approach achieved an accuracy of 89.29% when classifying the 12 gestures.

Non-wearable and mark-less approaches have also been employed in the literature. Y. Chen *et.al.* [10] proposed a real-

time Kinect-based dynamic hand gesture recognition system. In a first stage, different gestures are recorded. Then, each gesture passes through a data pre-processing stage, followed by feature extraction, and feature filtering. Features such as velocity, shape, location, and orientation are extracted per gesture. In the final stage, a multi-class SVM model is trained, to classify the gestures. Their approach is able to classify up to 36 gestures with an overall accuracy of 95.42%.

In the works presented in the literature, the recognition of gestures is mostly performed by using SVMs, but few uses SVM with a DTW kernel to classify the gestures. Therefore, the present work proposes two approaches for recognize two main gestures that are used during a Boccia game scenario. One approach uses the *Pandlet*, a wearable sensor developed by Fraunhofer [11]. The other approach uses the Microsoft Kinect sensor [12]. In a first stage, two databases were built that contain the sequences of each performed gesture. Then, a SVM model with DTW kernel was trained and evaluated.

This paper is organized in five sections. Section II describes the proposed system; section III presents the experimental methodology followed; section IV shows and discusses the results obtained; and the conclusions and future work are addressed in section V.

II. Proposed System

The developed system that allows gesture recognition is depicted in Fig. 1 and consists of a Microsoft Kinect 3D sensor, a *Pandlet*, and a computer to run the models and algorithms and analyze the gathered data.

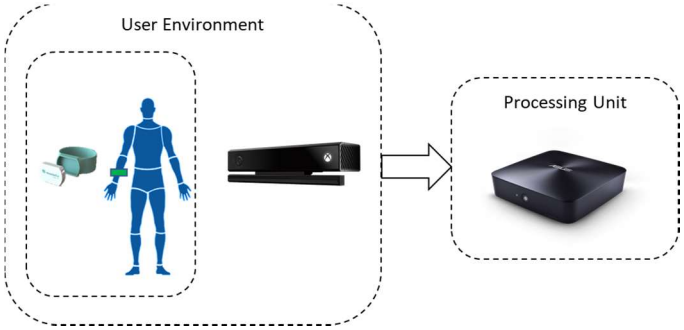

Fig. 1. The proposed system. On the left: The wearable sensor *Pandlet* and the Microsoft Kinect sensor. On the right: A computer.

The *Pandlet* [11] is a bracelet with a novel architecture of embedded electronics for wireless devices that can be used to develop wearable and Internet of Things (IoT) solutions. This wearable bracelet embodies a 3-axis accelerometer, magnetometer, and gyroscope that can be used to track the user's movements during various tasks. The *Pandlet* has an Application Programming Interface (API) that allow the bracelet to communicate its data via Bluetooth 4.0, supporting Windows, Linux, and Android platforms.

The Microsoft Kinect V2 [12] is a depth sensor that employs the time of light technique for tracking the user joints in real-time. The more recent version of Kinect (V2) contains three main imaging devices: an RGB camera which is used to capture color stream, infrared emitters which project a grid onto the scene, and a depth sensor which generates a depth image by analyzing the IR information. A microphone array is also available allowing to localize sound sources in a space and to perform background noise cancellation. The software released by Microsoft, the Kinect Software Development Kit (SDK), with its APIs (Application Programming Interface) in conjunction with the Kinect sensor allows the output of six data sources, including color, infrared, depth, face, body, and audio. It can track and recognize up to 6 people at the same time within its field of vision.

III. Experimental Methodology

This section presents the experimental methodology used to gather the data and to build the SVM model to automatically classify the two selected throwing gestures. Thus, the methodology is divided in 3 stages: Data Extraction (Section III-A), Database Construction (Section III-B), and the process for training a SVM with a DTW kernel (Section III-C). The latter also includes a description of the metrics used to evaluate and quantify the performance of the developed system.

A. Data Extraction

Generally during a Boccia Senior gameplay, the players perform two main gestures, Fig. 2, while throwing a ball with the intention to hit the target ball (a white ball). In order to recognize the throwing gestures, the present work uses the Kinect sensor along with its SDK that returns a body frame, containing the coordinates (x, y, z) for each joint of 25 joints available of the human body.

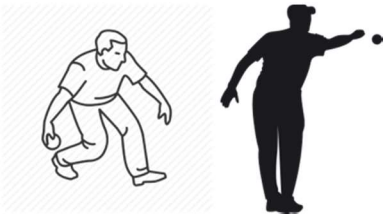

Fig. 2. Gesture A (on the left): the player executing an under throw. Gesture B (on the right): the player executing an upper throw.

Fig. 3 shows the select joints for computing four designated angles (θ, β, α, and ϒ).

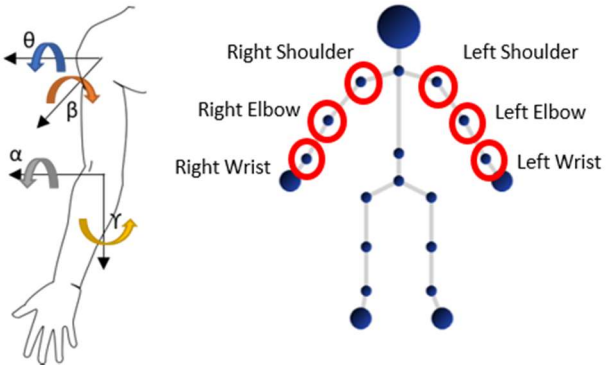

Fig. 3. On the left: The designated angles. On the right: The selected joints (red circles) for computing the designated angles.

By using the selected joints, two main vectors are formed: one between the shoulder and elbow joints and the other between the elbow and wrist joints. Then, the four angles are computed by applying the equation 1 and by following a methodology used in a previous work of the research team [13]. The methodology is replicated for both arms. After this computational step, each computed angle passes through a moving average filter (N=5), in order to smooth out short-term fluctuations.

$$\delta = \cos^{-1}\left(\frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|\|\vec{v}\|}\right) \quad (1)$$

Fig. 4 presents a graph displaying the four angles computed and filtered for the Kinect (θ, β, α, and ϒ), while a player executes one of the main movements used during a Boccia game scenario.

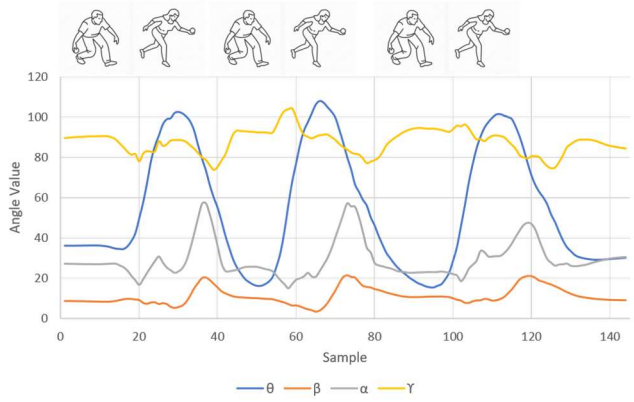

Fig. 4. Filtered data collected of a gesture that is usually executed during a Boccia gameplay.

The *Pandlet* API provides the quaternion data concerning the wrist orientation. In order to obtain the angles shown in Fig. 5, the quaternion data is converted to Euler angles.

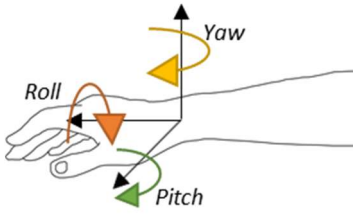

Fig. 5. The designated computed angles obtained by converting the provided quaternion to Euler angles.

Fig. 6 presents a graph displaying the three angles computed and for the *Pandlet* (Pitch, Yaw, and Roll) while a player executes one of the main movements used during a Boccia game scenario.

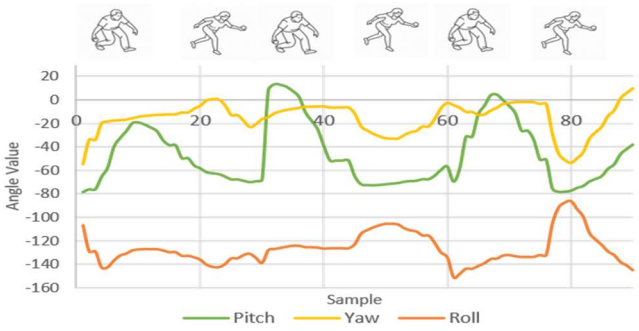

Fig. 6. Data collected (Pitch, Yaw, and Roll) of a gesture that is usually executed during a Boccia gameplay.

B. Database Construction

Two separated databases were constructed. The first one was built by using the computed angles for Kinect (θ, β, α, and ϒ) mentioned above. The second one was constructed by using the three angles (pith, yaw, and roll) that are provided by the *Pandlet* APIs.

The participants considered for the database construction were 10 adults with 18 to 52 years old. The recordings were conducted in a laboratorial environment in a closed room and repeated 5 times for each gesture.

The adult performed the designated gestures (Fig. 2) requested by the researcher, while using the *Pandlet* and staying in the Kinect field of view. In order to simulate a more real Boccia gameplay environment, each adult performed a designated gesture by throwing a real ball. The acquired data, the Kinect angles and the *Pandlet* angles were saved into two separated files.

C. Training a SVM with a DTW kernel

Support Vector Machines is a supervised learning method introduced by Vapnik [14] and it is widely used for statistical and regression analysis. By constructing a linear separating hyperplane, it is possible to classify the data. Thus, the purpose of the SVM is to find an Optimal Separating Hyperplane (OSH) by means of computing an optimization problem (Fig. 7). To control this optimization problem, a parameter C which is a trade-off between the maximum width of margin and minimum classification error, is used [14]. Usually used for classifying linear data, SVM can be also employed to classify some binary classification problems that do not have a simple hyperplane as a useful separating criterion. Thus, SVM employs kernel methods allowing nonlinear classification. The commonly used kernels of the SVM are: linear, polynomial, Radial Basis Function (RBF), and sigmoid [15].

Recently, Dynamic Time Warping (DTW) has been used as a new kernel for classifying dynamic sequence [17]. DTW is a well-known algorithm that compares and aligns two temporal sequences, taking into account that sequences may vary in length (time). Initially developed for speech recognition, DTW tries to find the minimal distance between two-time series by employing the dynamic programming technique [18].

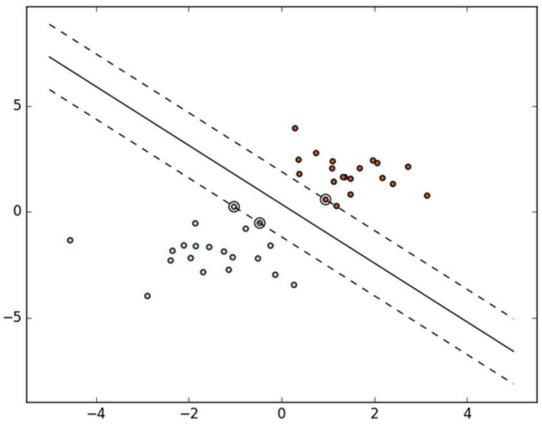

Fig. 7. SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space. The circled dots are the Support Vectors (SVs), which are points closest to the separating hyperplane [16].

The present work uses SVM with the DTW kernel to classify two typical gestures that are used during a gameplay of Boccia. This classification method was chosen because it shows remarkable performance in nonlinear analysis and high-dimensional pattern recognition, even when a small amount of training data is available [19]. By using different samples of the recorded sequences as input it is possible to obtain a model that can be used to predict a gesture. In order to assess the classification performance, the original two data bases were divided in two, i.e., 80% of the database with the Kinect gesture sequences was used to train the classifier and the rest (20%) was used to test the classifier. The same methodology was applied to the database containing the *Pandlet* gesture sequences. To quantify the classifier performance, the following metrics were used: accuracy, sensitivity, specificity, Area Under the Curve (AUC), and the Mathews Correlation Coefficient (MCC). Sensitivity measures the proportion of actual positives, which are correctly labelled as such [20]. Conversely, specificity measures the proportion of negatives that are correctly identified as such [20]. The AUC metric, which is obtained by computing the area under the Receiver Operating Curve (ROC), represents the average sensitivity across all possible specificities [20]. The MCC, usually employed as a measure of the quality of binary (two-class) classifications, is a correlation coefficient between the observed and predicted, that takes into account the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It returns a value between -1 and +1, where +1 represents a perfect prediction and -1 indicates total disagreement between prediction and observation. The MCC metric is generally regarded as a balanced measure, which can be used even if the classes are unbalanced (with different sizes) [21].

IV. Results and Discussion

This section presents the results obtained with the proposed approaches in the recognition of the two main gestures (the designated gestures A and B, Fig. 2) used during a Boccia game scenario. It starts by presenting the tools that were used to train and test the SVM models. Then, the results obtained with the *Pandlet* and the Kinect are shown. Finally, the discussion of the results is addressed.

A. SVM Model – Training

The SVM models were trained using the Accord Machine Learning C# library [22] which is a framework for scientific computing in .NET. With more focus on machine learning and data analysis, the framework is composed of multiple libraries which can be employed in a wide range of scientific computing applications, such as statistical data processing, machine learning, and pattern recognition. Additionally, it also has libraries for computer vision and audio processing applications. The framework offers a large number of probability distributions, hypothesis tests, kernel functions and support for most popular performance measurements techniques.

Since it offers a large number of kernel functions including the DTW kernel, the present work uses this framework to train and validate two SVM models with a DTW kernel. In order to tuning the SVM hyper parameters, the grid search method was employed. The grid search method consists in an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm [23]. This method is also available in the framework. The grid search algorithm outputs the settings (e.g. the C parameter) that achieved the highest score in the validation procedure.

1) *Pandlet Results*:

The SVM model was trained and validated using the DTW kernel with the *Pandlet* database. The results of the full analysis – confusion matrix, accuracy, sensitivity, specificity, AUC, and MCC metrics are presented below.

Table I presents the recognition accuracy confusion matrix for the two gestures obtained by SVM with DTW kernel, achieving an overall accuracy of 67%. More specifically, the gesture B having a better classification rate (73%) than the gesture A (60%).

Table II shows the overall performance of the SVM model, with an accuracy of 66.7%, a sensitivity and specificity of 60.0% and 72.3% respectively, an average AUC of 66.4%, and MCC of 33.0%.

TABLE I. Confusion Matrix – Pandlet

		Predicted Classes	
		Gesture A	**Gesture B**
Actual Classes	Gesture A	60%	27%
	Gesture B	40%	73%

TABLE II. Overall SVM Performance - Pandlet

Metric	**Value**
Accuracy	66.7%
Sensitivity	60.0%
Specificity	72.3%
AUC	66.4%
MCC	33.0%

2) Kinect Results:

The SVM model was trained and validated using the DTW kernel with the Kinect database. The results of the full analysis – confusion matrix, accuracy, sensitivity, specificity, AUC, and MCC metrics are presented below.

Table III presents the recognition accuracy confusion matrix for the two gestures obtained by SVM with DTW kernel, achieving an overall accuracy of 80%. More specifically, the gesture B having a better classification rate (90%) than the gesture A (70%).

Table IV shows the overall performance of the SVM model, with an accuracy of 80.0%, a sensitivity and specificity of 70.0% and 90.0% respectively, an average AUC of 80.0%, and MCC of 61.2%.

TABLE III. CONFUSION MATRIX – KINECT

		Predicted Classes	
		Gesture A	**Gesture B**
Actual Classes	Gesture A	70%	10%
	Gesture B	30%	90%

TABLE IV. OVERALL SVM PERFORMANCE - KINECT

Metric	**Value**
Accuracy	80.0%
Sensitivity	70.0%
Specificity	90.0%
AUC	80.0%
MCC	61.2%

B. Discussion

By comparing the results obtained in the previous section, it is possible to conclude that the SVM model that was trained with the Kinect data presents an overall better performance than the SVM model trained with the *Pandlet* data. The accuracy increased in the Kinect SVM model. In consequence, the performance of the other metrics of the Kinect SVM model also increased (Table IV). This outcome could be due to the fact that the *Pandlet* is a wrist wearable, i.e., the data obtained from the *Pandlet* can be noisier when compared to the Kinect data (Fig. 4 and 6). This additional noise can be derived from unexpected wrist movements while a player executes a gesture. Consequently, noisier signals can decrease the overall performance of the classifier. Additionally, the Kinect sensor allows to evaluate more than one center of rotation (shoulder and elbow) when compared to the *Pandlet* (wrist). This additional, information can also contribute to a better performance of the classifier.

V. CONCLUSION AND FUTURE WORK

Gesture recognition is an important issue for human interaction. Indeed, through this form of non-verbal communication a person may infer different types of intentions, among others. When the gestures are captured and interpreted by a computer, it is possible to start a new form of Human Computer Interaction (HCI). Despite being used for a more natural interaction with the computer (without the physical support of mouse and keyboard) the gesture recognition may also be used to monitor the health status of a person regarding, for example, some specific physical impairment. In this work, it is proposed a hybrid approach that uses wearable and non-wearable devices to capture two throwing movements of a ball in a simulated Boccia game environment. Regarding this specific movements, two SVM models with a DTW kernel were trained in order to automatically detect the type of throwing that was executed. The first model was trained using the *Pandlet* dataset. The second one was built using the Kinect dataset. The performance of both models was quantified and compared. The results pointed out that both sensors have good performance, however the Kinect presented a better overall performance in the gesture recognition, with an accuracy of 80%.

As future work, it is intended to merge the data of the two approaches in order to try to increase the performance of the system and to test it in a real Boccia Senior game scenario, in real time, to detect the type of throwing movement that was performed and to define new metrics such as the force executed by the player when throwing a ball.

ACKNOWLEDGMENT

This article is a result of the project Deus Ex Machina: NORTE-01-0145-FEDER-000026, supported by Norte Portugal Regional Operational Program (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

REFERENCES

[1] F. Bevilacqua, J. Ridenour, and D. J. Cuccia, "3D motion capture data: motion analysis and mapping to music," Proc. Work. Sens. Input Media-centric Syst. (SIMS 02), pp. 562–569, 2002.

[2] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis, "A gesture-driven multimodal interactive dance system," 2004 IEEE Int. Conf. Multimed. Expo (IEEE Cat. No.04TH8763), pp. 1579–1582, 2004.

[3] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," 2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 755–762, 2010.

[4] S. Bhattacharya, B. Czejdo, and N. Perez, "Gesture classification with machine learning using Kinect sensor data," 2012 Third Int. Conf. Emerg. Appl. Inf. Technol., pp. 348–351, 2012.

[5] C. C. Yang and Y. L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," Sensors, vol. 10, no. 8. pp. 7772–7788, 2010.

[6] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," Comput. Vis. Image Underst., vol. 115, no. 2, pp. 224–241, 2011.

[7] S. Amendola, L. Bianchi, and G. Marrocco, "Movement Detection of Human Body Segments: Passive radio-frequency identification and machine-learning technologies.," IEEE Antennas Propag. Mag., vol. 57, no. 3, pp. 23–37, 2015.

[8] D. Wilson and A. Wilson, "Gesture Recognition Using The XWand," 2004.

[9] J. Wu, G. Pan, D. Zhang, and G. Qi, "Gesture recognition with a 3-d accelerometer," Proc. 6th Int. Conf. Ubiquitous Intell. Comput., vol. 5585, pp. 25–38, 2009.

[10] Y. Chen, B. Luo, Y. Chen, G. Liang, and X. Wu, "A Real-time Dynamic Hand Gesture Recognition System Using Kinect Sensor *," pp. 2026–2030, 2015.

[11] “Pandlets.” [Online]. Available: http://www.fraunhofer.pt/en/fraunhofer_aicos/projects/internal_research/pandlets.html. [Accessed: 27-Mar-2017].

[12] Microsoft, “Developing with Kinect,” 2017. [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect/develop. [Accessed: 13-Mar-2017].

[13] V. Silva, P. Leite, F. Soares, J. S. Esteves, and S. Costa, “Imitate me! - Preliminary tests on an upper members gestures recognition system,” Lect. Notes Electr. Eng., vol. 402, pp. 373–383, 2017.

[14] C. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” Data Min. Knowl. Discov., vol. 2, no. 2, pp. 121–167, 1998.

[15] “Support Vector Machines (SVM) - MATLAB & Simulink,” 2015. [Online]. Available: http://www.mathworks.com/help/stats/support-vector-machines-svm.html. [Accessed: 20-Jan-2016].

[16] “1.4. Support Vector Machines — scikit-learn 0.18.2 documentation,” 2016. [Online]. Available: http://scikit-learn.org/stable/modules/svm.html. [Accessed: 28-May-2017].

[17] M. A. Bagheri, Q. Gao, and S. Escalera, “Support vector machines with time series distance kernels for action classification,” in 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, 2016.

[18] M. Reyes, “Feature weighting in dynamic time warping for gesture recognition in depth data,” Comput. Vis. Work. (ICCV Work. 2011 IEEE Int. Conf., pp. 1182–1188, 2011.

[19] P. Michel and R. El Kaliouby, “Facial Expression Recognition Using Support Vector Machines,” 2000.

[20] A. Zheng, “How to Evaluate Machine Learning Models: Classification Metrics,” 2015. [Online]. Available: http://blog.dato.com/how-to-evaluate-machine-learning-models-part-2a-classification-metrics. [Accessed: 22-May-2016].

[21] G. Jurman and C. Furlanello, “A unifying view for performance measures in multi-class prediction,” PLoS One, vol. 7, no. 8, 2010.

[22] C. R. Souza, “The Accord.NET Framework,” 2014. [Online]. Available: http://accord-framework.net. [Accessed: 22-May-2016].

[23] “Accord Machine Learning - GridSearch.” [Online]. Available: http://accord-framework.net/docs/html/T_Accord_MachineLearning_GridSearch_1.htm. [Accessed: 03-Jun-2017].