Chapter 6

Gesture Recognition by Using Depth Data: Comparison of Different Methodologies

Grazia Cicirelli and Tiziana D'Orazio

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/68118

Abstract

In this chapter, the problem of gesture recognition in the context of human computer interaction is considered. Several classifiers based on different approaches such as neural network (NN), support vector machine (SVM), hidden Markov model (HMM), deep neural network (DNN), and dynamic time warping (DTW) are used to build the gesture models. The performance of each methodology is evaluated considering different users performing the gestures. This performance analysis is required as the users perform gestures in a personalized way and with different velocity. So the problems concerning the different lengths of the gesture in terms of number of frames, the variability in its representation, and the generalization ability of the classifiers have been analyzed.

Keywords: gesture recognition, feature extraction, model learning, gesture segmentation, human-robot interface, Kinect camera

1. Introduction

In the last decade, gesture recognition has been attracting a lot of attention as a natural way to interact with computer and/or robots through intentional movements of hands, arms, face, or body. A number of approaches have been proposed giving particular emphasis on hand gestures and facial expressions by the analysis of images acquired by conventional RGB cameras [1, 2].

The recent introduction of low cost depth sensors, such as the Kinect camera, allowed the spreading of new gesture recognition approaches and the possibility of developing personalized human computer interfaces [3, 4]. The Kinect camera provides RGB images together with depth information, so the 3D structure of the scene is immediately available. This allows us to

easily manage many tasks such as people segmentation and tracking, body part recognition, motion estimation, and so on. Recently human activity recognition and motion analysis from 3D data have been reviewed in a number of interesting works [5–8].

At present, Gesture Recognition through visual and depth information is one of the main active research topics in the computer vision community. The launch on the market of the popular Kinect, by the Microsoft Company, influenced video-based recognition tasks such as object detection and classification and in particular allowed the increment of the research interest in gesture/activity recognition. The Kinect provides synchronized depth and color (RGB) images where each pixel corresponds to an estimate of the distance between the sensor and the closest object in the scene together with the RGB values at each pixel location. Together with the sensor some software libraries are also available that permit to detect and track one or more people in the scene and to extract the corresponding human skeleton in real time. The availability of information about joint coordinates and orientation has promoted a great impulse to research on gesture and activity recognition [9–14].

Many papers, presented in literature in the last years, use normalized coordinates of proper subset of skeleton joints which are able to characterize the movements of the body parts involved in the gestures [15, 16]. Angular information between joint vectors has been used as features to eliminate the need of normalization in Ref. [17].

Different methods have been used to generate gesture models. Hidden Markov Models (HMM) are a common choice for gesture recognition as they are able to model sequential data over time [18, 19]. Usually HMMs require sequences of discrete symbols, so different quantization schemes are first used to quantize the features which characterize the gestures. Support vector machines (SVM) reduce the classification problem into multiple binary classifications either by applying a one-versus-all (OVA-SVM) strategy (with a total of N classifiers for N classes) [20, 21] or a one-versus-one (OVO-SVM) strategy (with a total of $N \times (N-1)/2$ classifiers for N classes) [22, 23]. Artificial neural networks (ANNs) represent another alternative methodology to solve classification problems in the context of gesture recognition [24]. The choice of the network topology, the number of nodes/layers and the node activation functions depends on the problem complexity and can be fixed by using iterative processes which run until the optimal parameters are found [25].

Distance-based approaches are also used in gesture recognition problems. They use distance metrics for measuring the similarity between samples and gesture models. In order to apply any metric for making comparisons, these methods have to manage the problem related to the different length of feature sequences. Several solutions have been proposed in literature: Dynamic Time Warping technique (DTW) [26] is the most commonly used. It calculates an optimal match between two sequences that are nonlinearly aligned. A frame-filling algorithm is proposed in Ref. [27] to first align gesture data, then an eigenspace-based method (called Eigen3Dgesture) is applied for recognizing human gestures.

In the last years, the growing interest in automatically learning the specific representation needed for recognition or classification has fostered the recent emergence of deep learning architectures [28]. Rather than using handcrafted features as in conventional machine learning

http://dx.doi.org/10.5772/68118

techniques, deep neural architectures are applied to learn representations of data at multiple levels of abstractions in order to reduce the dimensionality of feature vectors and to extract relevant features at higher level. Recently, several approaches have been proposed such as in Refs. [29, 30]. In Ref. [29], a method for gesture detection and localization based on multiscale and multimodel deep learning is presented. Both temporal and spatial scales are managed by employing a multimodel convolutional neural network. Similarly in Ref. [30], a multimodel gesture segmentation and recognition method, called deep dynamic neural networks, is presented. A semisupervised hierarchical dynamic framework based on a Hidden Markov Model is proposed for simultaneous gesture segmentation and recognition.

In this chapter, we compare different methodologies to approach the problem of Gesture Recognition in order to develop a natural human-robot interface with good generalization ability. Ten gestures performed by one user in front of a Kinect camera are used to train several classifiers based on different approaches such as dynamic time warping (DTW), neural network (NN), support vector machine (SVM), hidden Markov model (HMM), and deep neural network (DNN).

The performance of each methodology is evaluated considering several tests carried out on depth video streams of gestures performed by different users (diverse from the one used for the training phase). This performance analysis is required as users perform gestures in a personalized way and with different velocity. Even the same user executes gestures differently in separate video acquisition sessions. Furthermore, contrarily to the case of static gesture recognition, in the case of depth videos captured live the problem of gesture segmentation must be addressed. During the test phase, we apply a sliding window approach to extract sequences of frames to be processed and recognized as gestures. Notice that the training set contains gestures which are accompanied by the relative ground truth labels and are well defined by their start and end points. Testing live video streams, instead, involves several challenging problems such as the identification of the starting/ending frames of a gesture, the different length related to the different types of gestures and finally the different speeds of execution. The analysis of the performance of the different methodologies allows us to select, among the set of available gestures, the ones which are better recognized together with the better classifier, in order to construct a robust human-robot interface.

In this chapter, we consider all the mentioned challenging problems. In particular, the fundamental steps that characterize an automatic gesture recognition system will be analyzed: (1) feature extraction that involves the definition of the features that better and distinctively characterize a specific movement or posture; (2) gesture recognition that is seen as a classification problem in which examples of gestures are used into supervised and semisupervised learning schemes to model the gestures; (3) spatiotemporal segmentation that is necessary for determining, in a video sequence, where the dynamic gestures are located, i.e., when they start and end.

The rest of the chapter is organized as follows. The overall description of the problem and the definition of the gestures are given in Section 2. The definition of the features is provided in Section 3. The methodologies selected for the gesture model generation are described in Section 4. Section 6 presents the experiments carried out both in the learning and prediction stage.

Furthermore, details on gesture segmentation will be given in the same section. Finally, Section 7 presents the final conclusions and delineates some future works.

2. Problem definition

In this chapter, we consider the problems related to the development of a gesture recognition interface giving a panoramic view and comparing the most commonly used methodologies of machine learning theory. At this aim, the Kinect camera is used to record video sequences of different users while they perform predefined gestures in front of it. The OpenNI Library is used to detect and segment the user in the scene in order to obtain the information of the joints of the user's body. Ten different gestures have been defined. They are pictured in **Figure 1**. Throughout the chapter the gestures will be referred by using the following symbols G_1, G_2, G_3, … G_N, where $N = 10$. Some gestures are quite similar in terms of variations of joint orientations; the only difference is the plane in which the bones of the arm rotate. This is the case, for example, of gestures G_9 and G_4 or G_1, and G_8. Furthermore, some gestures involve movements in a plane parallel to the camera (G_1, G_3, G_4, G_7) while others involve a forward motion in a plane perpendicular to the camera (G_2, G_5, G_6, G_8, G_9, G_{10}). In the last case, instability in detecting some joints can occur due to autoocclusions.

The proposed approaches for gesture recognition involve three main stages: a feature selection stage, a learning stage and a prediction stage. Firstly the human skeleton information, captured and returned by the depth camera, is converted into representative and discriminant

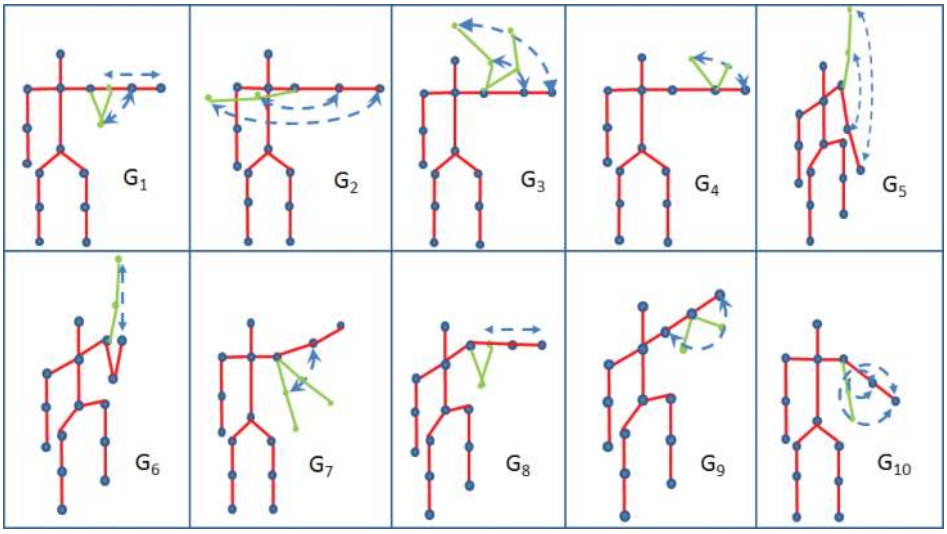

Figure 1. Ten different gestures are shown. Gestures G_1, G_3, G_4, and G_7 involve movements in a plane parallel to the camera. Gestures G_2, G_5, G_6, G_8, G_9, and G_{10} involve a forward motion in a plane perpendicular to the camera.

http://dx.doi.org/10.5772/68118

features. These features are used during the learning stage to learn the gesture model. In this chapter, different methodologies are applied and compared in order to construct the gesture model. Some methodologies are based on a supervised or semisupervised process such as neural network (NN), support vector machine (SVM), hidden Markov model (HMM), and deep neural network (DNN). Dynamic time warping (DTW) is a distance-based approach, instead. Finally, during the prediction stage new video sequences of gestures are tested by using the learned models. The following sections will describe in detail each stage previously introduced.

3. Feature selection

The complexity of the gestures strictly affects the feature selection and the choice of the methodology for the construction of the gesture model. If gestures are distinct enough, the recognition can be easy and reliable. So, the coordinates of joints, which are immediately available by the Kinect software platforms, could be sufficient. In this case a preliminary normalization is required in order to guarantee invariance with respect to the height of the users, distance and orientation with respect to the camera. On the other hand, the angular information of joint vectors has the great advantage of maximizing the invariance of the skeletal representation with respect to the camera position. In Ref. [31], the angles between the vectors generated by the elbow-wrist joints, and the shoulder-elbow joints, are used to generate the models of the gestures. The experimental results, however, prove that these features are not discriminant enough to distinguish all the gestures.

In our approach, we use more complex features that represent orientations and rotations of a rigid body in three dimensions. The quaternions of two joints (shoulder and elbow) of the left arm are used. A quaternion comprises a scalar component and a vector component in complex space and is generally represented in the following form:

$$q = a + bi + cj + dk \tag{1}$$

where the coefficients a, b, c, d are real numbers and i, j, k are the fundamental quaternion units. The quaternions are extremely efficient to represent three-dimensional rotations as they combine the rotation angles together with the rotation axes. In this work, the quaternions of the shoulder and elbow joints are used to define a feature vector V_i for each frame i:

$$V_i = [a_i^s, b_i^s, c_i^s, d_i^s, a_i^e, b_i^e, c_i^e, d_i^e] \tag{2}$$

where the index s stands for shoulder and e stands for elbow. The sequence of vectors of a whole gesture execution is defined by the following vector:

$$\overline{V} = [V_1, V_2, \ldots, V_n] \tag{3}$$

Where n is the number of frames during which the gesture is entirely performed.

4. Learning stage: gesture model construction

The learning stage regards the construction of the gesture model. As introduced in Section 1, machine learning algorithms are largely and successfully applied to gesture recognition. In this context, gesture recognition is considered as a classification problem. So, under this perspective, a number of gesture templates are collected, opportunely labeled with the class labels (*supervised learning*) and used to train a learning scheme in order to learn a classification model. The constructed model is afterwards used to predict the class label of unknown templates of gestures.

In this chapter, different learning methodologies are applied to learn the gesture model. For each of them, the best parameter configuration and the best architecture topology which assure the convergence of each methodology are selected. Artificial neural networks (ANNs), support vector machines (SVMs), hidden Markov models (HMMs), and deep neural networks (DNNs) are the machine learning algorithms compared in this chapter. Furthermore a distance-based method, the dynamic time warping (DTW), is also applied and compared with the aforementioned algorithms. The following subsections will give a brief introduction of each algorithm and some details on how they are applied to solve the proposed gesture recognition problem.

4.1. Neural network

A neural network is a computational system that simulates the way biological neural systems process information [32]. It consists of a large number of highly interconnected processing units (neurons) typically distributed on multiple layers. The learning process involves successive adjustments of connection weights, through an iterative training procedure, until no further improvement occurs or until the error drops below some predefined reasonable threshold. Training is accomplished by presenting couples of input/output examples to the network (*supervised learning*).

In this work, 10 different neural networks have been used to learn the models of the defined gestures. The architecture of each NN consists of an input layer, one hidden layer and an output layer with a single neuron. The back-propagation algorithm is applied during the learning process. Each training set contains the templates of one gesture as positive examples and those of all the others as negative ones. As each gesture execution lasts a different number of frames, a preliminary normalization of the feature vectors has been carried out by using a linear interpolation. Linear interpolation to resample the number of features is a good compromise between computational burden and quality of results. The length of a feature vector V, which describes one single gesture, has been fixed to $n = 60$. This length has been fixed considering the average time of execution of each type of gesture which is about 2 seconds and the sample rate of the Kinect camera which is 30 Hz.

4.2. Support vector machine

Support vector machine is a supervised learning algorithm widely used in classification problems [33]. The peculiarity of SVM is that of finding the optimal separating hyperplane between

http://dx.doi.org/10.5772/68118

the negative and positive examples of the training set. The optimal hyperplane is defined as the maximum margin hyperplane, i.e., the one for which the distance between the hyperplane (decision surface) and the closest data points is maximum. It can be shown that the optimal hyperplane is fully specified by a subset of data called *support vectors* which lie nearest to it, exactly on the margin.

In this work, SVMs have been applied considering the one-versus-one strategy. This strategy builds a two-class classifier for each pair of gesture classes. In our case, the total number of SVMs is defined by:

$$M = \frac{N(N-1)}{2} \tag{4}$$

where N is the number of gesture classes. The training set of each SVM contains the examples of the two gesture classes for which the current classifier is built. As in the case of NNs, the feature vectors are preliminary normalized to the same length n.

4.3. Hidden Markov model

Hidden Markov model is a statistical model which assumes that the system to be modeled is a Markov process. Even if the theory of HMMs dates back to the late 1960s, their widespread application occurred only within the past several years [34, 35]. Their successful application to speech recognition problems motivated their diffusion in gesture recognition as well. An HMM consists of a set of unobserved (*hidden*) states, a state transition probability matrix defining the transition probabilities among states and an observation or emission probability matrix which defines the output model. The goal is to learn the best set of state transition and emission probabilities, given a set of observations. These probabilities completely define the model.

In this work, one discrete hidden Markov model is learnt for each gesture class. The feature vectors of each training set, which represent the observations, are firstly normalized and then discretized by applying a K-means algorithm. A fully connected HMM topology and the Baum-Welch algorithm have been applied to learn the optimal transition and emission probabilities.

4.4. Deep neural network

Deep learning is a relatively new branch of machine learning research [28]. Its objective is to learn features automatically at multiple levels of abstraction exploiting an unsupervised learning algorithm at each layer [36]. At each level a new data representation is learnt and used as input to the successive level. Once a good representation of data has been found, a supervised stage is performed to train the top level. A final supervised fine-tuning stage of the entire architecture completes the training phase and improves the results. The number of levels defines the deepness of the architecture.

In this work, a deep neural network with 10 output nodes (one for each class of gesture) is constructed. It comprises two levels of unsupervised autoencoders and a supervised top level.

The autoencoders are used to learn a lower dimensional representation of the feature vectors at a higher level of abstraction. An autoencoder is a neural network which is trained to reconstruct its own input. It is comprised of an encoder, that maps the input to the new representation of data, and a decoder that reconstruct the original input. We use two autoencoders with one hidden layer. The number of hidden neurons represents the dimension of the new data representation. The feature vectors of training set are firstly normalized, as described in Section 4.1, and fed into the first autoencoder. So the features generated by the first autoencoder are used as input to the second one. The size of the hidden layer for both the first and second autoencoder has been fixed to half the size of the input vector. The features learnt by the last autoencoder are given as input to the supervised top level implemented by using a softmax function trained with a scaled conjugate gradient algorithm [37]. Finally the different levels are stacked to form the deep network and its parameters are fine-tuned by performing backpropagation using the training data in a supervised fashion.

4.5. Dynamic time warping

DTW is a different technique with respect to the previously described ones as it is a distance-based algorithm. Its peculiarity is to find the ideal alignment (*warping*) of two time-dependent sequences considering their synchronization. For each pair of elements of the sequences, a cost matrix, also referred as local distance matrix, is computed by using a distance measure. Then the goal is to find the minimal cost path through this matrix. This optimal path defines the ideal alignment of the two sequences [38]. DTW is successfully applied to compare sequences that are altered by noise or by speed variations. Originally, the main application field of DTW was automatic speech processing [39], where variation in speed appears concretely. Successively DTW found its application in movement recognition, where variation in speed is of major importance, too.

In this work, DTW is applied to compare the feature vectors in order to measure how different they are for solving the classification problem. Differently from the previously described methodologies, the preliminary normalization of feature vectors is not required due to the warping peculiarity of DTW algorithm. For each class of gesture, one target feature vector is selected. This is accomplished by applying DTW to the set of training samples inside each gesture class. The one with the minimum distance from all the other samples of the same class is chosen as target gesture. Each target gesture will be used in the successive prediction stage for classification.

5. Prediction stage: gesture model testing

In prediction stage, also referred as testing stage, video sequences with unknown gestures are classified by using the learnt gesture models. This stage allows us to compare the recognition performance of the methodologies introduced in the learning stage. These methodologies have been applied by using different strategies as described in the following.

In the case of NN, 10 classifiers have been trained, one for each class. So the feature vector of a new gesture sample is inputted into all the classifiers and is assigned to the class with the maximum output value.

http://dx.doi.org/10.5772/68118

In the case of SVM, instead, a max-win voting strategy has been applied. The trained SVMs are 45 two-class classifiers. When each classifier receives as input a gesture sample, classifies it into one of the two classes. Therefore, the winning class gets one vote. When all the 45 votes have been assigned, the instance of the gesture is classified into the class with the maximum number of votes.

In the case of HMM, 10 HMMs have been learnt during the learning stage, one for each class of gesture. As introduced in Section 4.3 the model of each class is specified by the transition and emission probabilities learnt in the learning stage. When a gesture instance is given as input to the HMM, this computes the probability of that instance given the model. The class of the HMM returning the maximum probability is the winning class.

In the case of DNN, as described in Section 4.4, the deep architecture, constructed in the learning stage, has 10 output nodes. So, when a gesture sample is inputted in the network for prediction, the winning class is simply the one relative to the node with the maximum output value.

Finally, for what concerns the DTW case, the target gestures, found during the learning stage, are used to predict the class of new gesture instances. The distances between the unknown gesture sample and the 10 target gestures are computed. The winning class is that of the target gesture with minimum distance.

6. Experiments

In this section the experiments carried out in order to evaluate the performance of the analyzed methodologies will be described and the obtained results will be shown and compared. In particular, the experiments conducted in both the learning stage and the prediction stage will be detailed separately for a greater clarity of presentation.

Several video sequences of gestures performed by different users have been acquired by using a Kinect camera. Sequences of the same users in different sessions (e.g., in different days) have been also acquired in order to have a wide variety of data. The length of each sequence is about 1000 frames. The users have been requested to execute gestures standing in front of the Kinect, by using the left arm and without pause between one gesture execution and the successive one. The distance between Kinect and user is not fixed. The only constraint is that the whole user's body has to be seen by the sensor, so its skeleton data can be detected by using the OpenNi processing Library. These data are recorded for each frame of the sequence.

6.1. Learning stage

As described in Section 4, the objective of the learning stage is to construct or, more specifically, to *learn* a gesture model. In order to reach this goal, the first step is the construction of the training datasets. The idea of using public datasets has been discarded as they do not assure that real situations are managed. Furthermore, they contain sample gestures which are acquired mainly in the same conditions. We have decided to use a set of gestures chosen by us (see **Figure 1**), which have been selected from the "Arm-and-Hand Signals for Ground Forces" [40].

The video sequences of only one user (afterward referred as Training User) are considered for building the training sets. Each sequence contains several executions of the same gesture without idle frames between one instance and the other. In this stage, we manually segment the training streams into gesture instances in order to guarantee that each extracted subsequence contains exactly one gesture execution. Then each instance is converted in feature vector by using the skeleton data as described in Section 3. Notice that feature vectors V can have different lengths, because either gesture execution lasts a different number of frames or users execute gestures with different speeds. Part of the obtained feature vectors are used for training and the rest for validation.

The second step of the learning stage is the construction of the gesture model by using the methodologies described in Section 4. A preliminary normalization of feature vectors to the same length is needed in the cases of NN, SVM, HMM, and DNN. As described in Section 4.1, n has been fixed to 60. So each normalized feature vector V has 480 components which have been defined by using the quaternion coefficients of shoulder and elbow joints (see Eqs. (2) and (3)). In the case of DTW this normalization is not required.

For each methodology, different models can be learnt depending on the parameters of the methodology. These parameters can be structural such as the number of hidden nodes in the NN architecture or in the autoencoder or the number of hidden states in a HMM; or they can be tuning parameters as in the case of SVM. So, different experiments have been carried out for selecting the optimal parameters inside each methodology. Optimal parameters have to be intended as those which provide a good compromise between over-fitting and prediction error over the validation set.

6.2. Prediction stage

The prediction stage represents the recognition phase which allows us to compare the performance of each methodology. In this phase the class labels of feature vectors are predicted based on the learnt gesture model. Differently from the training phase that can be defined as an off-line phase, the prediction stage can be defined as an on-line stage. In this case the video sequences of six different users (excluded the Training-User) have been properly processed by using an approach that works when live video sequences have to be tested. Differently from the learning stage, where gesture instances were manually selected from the sequences and were directly available for training the classifiers, in the prediction stage the sequences need to be opportunely processed by applying a gesture segmentation approach. This process involves several challenging problems such as the identification of the staring/ending points of a gesture instance, the different length related to the different classes of gestures and finally the different speeds of execution.

In this work, the sequences are processed by using a sliding window approach, where a window slides forward over the sequence by one frame per time in order to extract subsequences. First, the dimension of the sliding window must be defined. As there are no idle frames among successive gesture executions, an algorithm based on Fast Fourier Transform (FFT) has been applied in order to estimate the duration of each gesture execution [41]. As each sequence contains several repetitions of the same gesture, it is possible to approximate

http://dx.doi.org/10.5772/68118

the sequence of features as a periodic signal. Applying the FFT and by tacking the position of the fundamental harmonic component, the period can be evaluated as the reciprocal value of the peak position. The estimated period is then used to define the sliding window's dimension in order to extract subsequences of features from the original sequence. Each subsequence represents the feature vector which is then normalized (if required) and provided as input to the classifier which returns a prediction label for the current vector. In order to construct a more robust human computer interface, a further verification check has been introduced before the final decision is taken. This process has been implemented by using a max-voting scheme on 10 consecutive answers of the classifier obtained testing 10 consecutive subsequences. The final decision is that relative to the class label with the maximum number of votes.

6.3. Results and discussion

In **Figures 2–7**, the recognition rates obtained by testing the classifiers on a number of sequences performed by six different users are reported. For each user the plotted rates have been obtained by averaging the results over three testing sequences. As can be observed the classifiers behave in a very different way due to the personalized execution of gestures by the users. Furthermore, there are cases where some classifiers fail in assigning the correct class. This is, for example, the case of gestures G_2 and G_4 performed by User 6 (see **Figure 7**). DTW has 0% detection rate for G_2, whereas NN has 0% detection rate for G_4. The same happens for gesture G_9 performed by User 2 (see **Figure 3**) which is rarely recognized by all the classifiers, as well as G_3 performed by User 5 (see **Figure 6**).

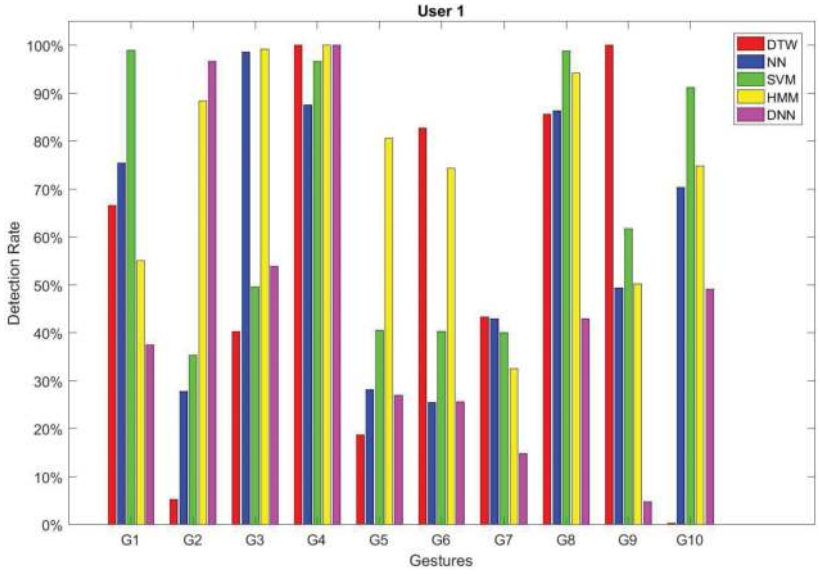

Figure 2. Recognition rates obtained by testing each method on sequences of gestures performed by User 1.

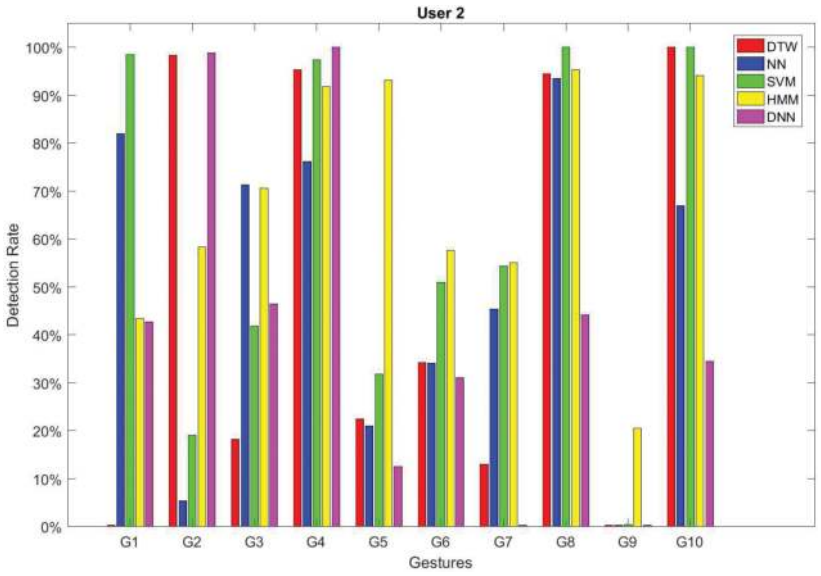

Figure 3. Recognition rates obtained by testing each method on sequences of gestures performed by User 2.

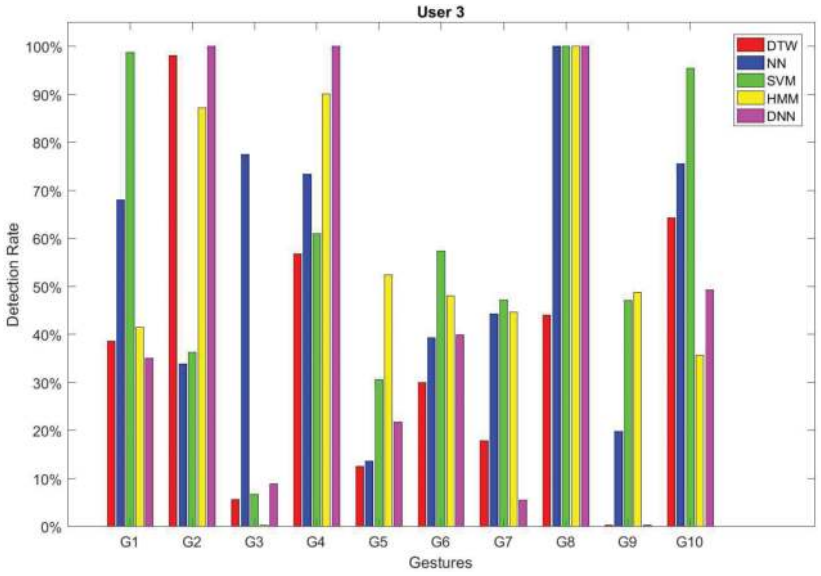

Figure 4. Recognition rates obtained by testing each method on sequences of gestures performed by User 3.

http://dx.doi.org/10.5772/68118

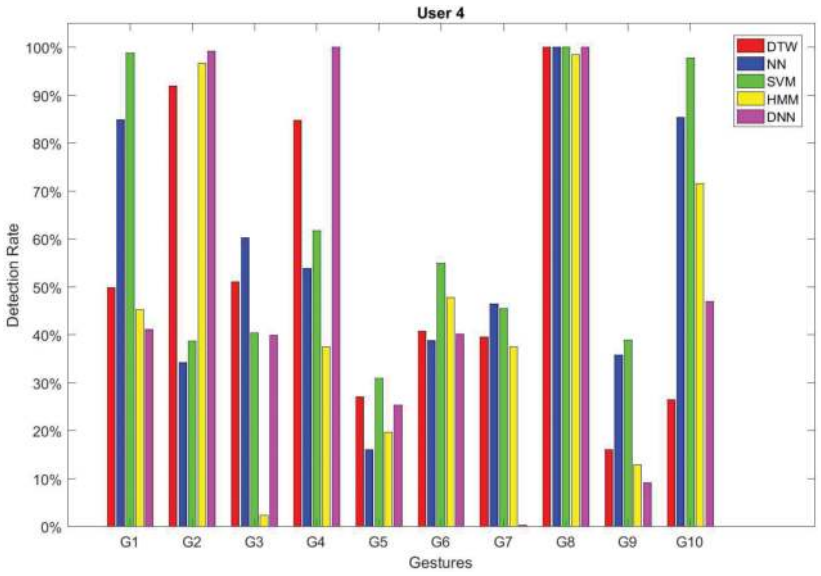

Figure 5. Recognition rates obtained by testing each method on sequences of gestures performed by User 4.

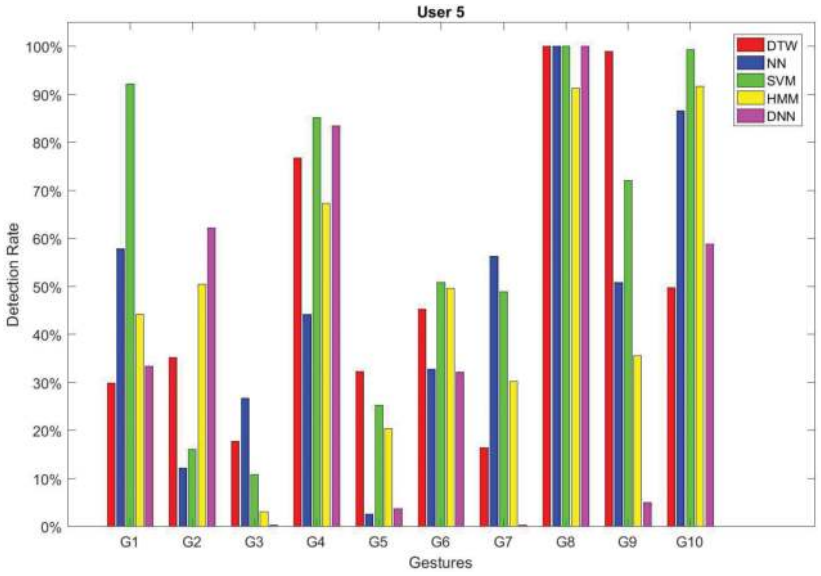

Figure 6. Recognition rates obtained by testing each method on sequences of gestures performed by User 5.

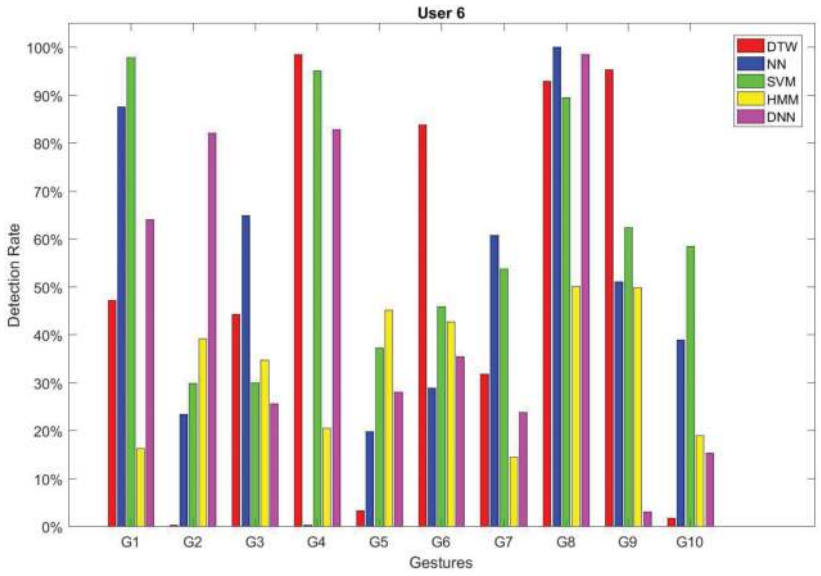

Figure 7. Recognition rates obtained by testing each method on sequences of gestures performed by User 6.

In order to analyze the performance of classifiers when the same user is used in the learning and prediction phases, an additional experiment has been carried out. So the Training User has been asked to perform again the gestures. **Figure 8** shows the obtained recognition rates. These

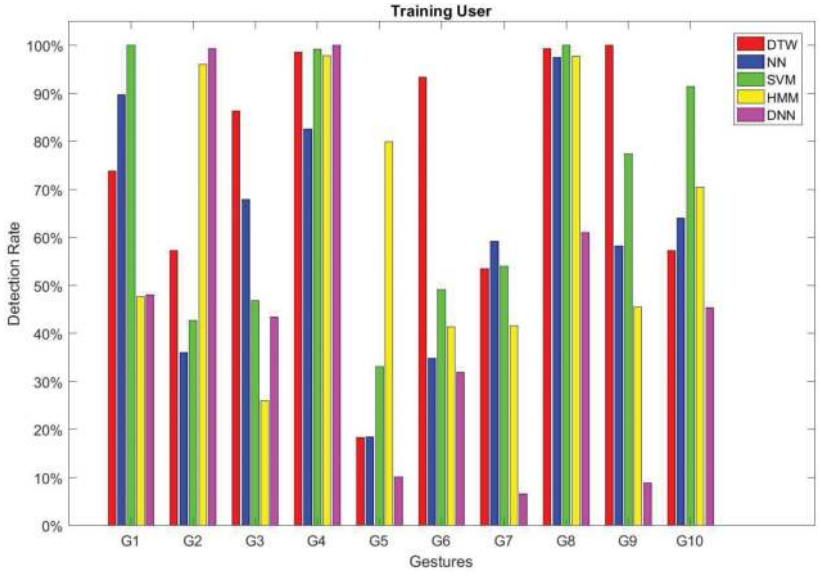

Figure 8. Recognition rates obtained by testing each method on sequences of gestures performed by the Training User in a session different from the one used for the learning phase.

http://dx.doi.org/10.5772/68118

results confirm the variability of classifiers performance even if the same user is used for training and testing the classifiers.

The obtained results confirm that it is difficult to determine the superiority of one classifier over the others because of the large number of variables involved that do not guarantee a uniqueness of gesture execution. These are for example: the different relative positions between users and camera, the different orientations of the arm, the different amplitude of the movement, and so on. All these factors can greatly modify the resulting skeletons and joint positions producing large variations in the extracted features.

Some important conclusions can be drawn from the experiments that have been carried out: the solution of using only one user to train the classifiers can be pursued as the recognition rates are quite good even if the gestures are performed in personalized way.

Another point concerns the complexity of the gestures used in our experiments. The results show that the failures are principally due either to the strict similarity between different gestures or to the fact that the gestures which involve a movement perpendicular with respect to the camera (not in the lateral plane) can produce false skeleton postures and consequently features affected by errors.

Moreover, some gestures have parts of the movement in common. **Figures 9** and **10** have been pictured to better explain these problems.

Figure 9 shows the results obtained by testing the first 1000 frames of a sequence of gesture G_3 executed by User 1. Each plot in the figure represents the output of each classifier DTW, NN, SVM, HMM, and DNN, respectively. As can be seen in the case of DTW, SVM, and DNN, gesture G_3 is frequently misclassified as gesture G_4. Both gestures are executed in a plane parallel to the camera: G_3 involves the rotation of the whole arm, whereas G_4 involves the rotation of the forearm only (as can be seen in **Figure 11**). Notice that the misclassification happens principally in the starting part of gesture G_3, which is very similar to the starting part of G_4; therefore, they can be easily mistaken.

Furthermore in **Figure 9**, it is worth to notice the good generalization ability of NN and HMM. As can be seen in these cases, both classifiers are always able to recognize the gesture even when the sliding windows cover the frames between two successive gesture executions.

An additional observation can be taken considering G_1 and G_8 as an example. In **Figure 12**, notice that gesture G_1 and gesture G_8 involve the same rotations of the forearm, but performed in different planes with respect to the camera (the lateral one in the case of G_1 and the frontal one in the case of G_8). It is evident that a slight different orientation of the user in front of the camera while performing gesture G_1 (risp. G_8), could generate skeletons quite similar to those obtained by performing gesture G_8 (risp. G_1). **Figure 10** shows the results relative to this case. As can be seen gesture G_8 is sometimes misclassified as gesture G_1 by DTW and SVM. A few misclassifications of gesture G_8 as G_6 are also present since G_8 and G_6 have some parts of movement in common.

6.4. Statistical evaluation

The analysis of the performance of the different methodologies, presented above, allows us to draw some important conclusions that must be considered in order to build a robust human-robot

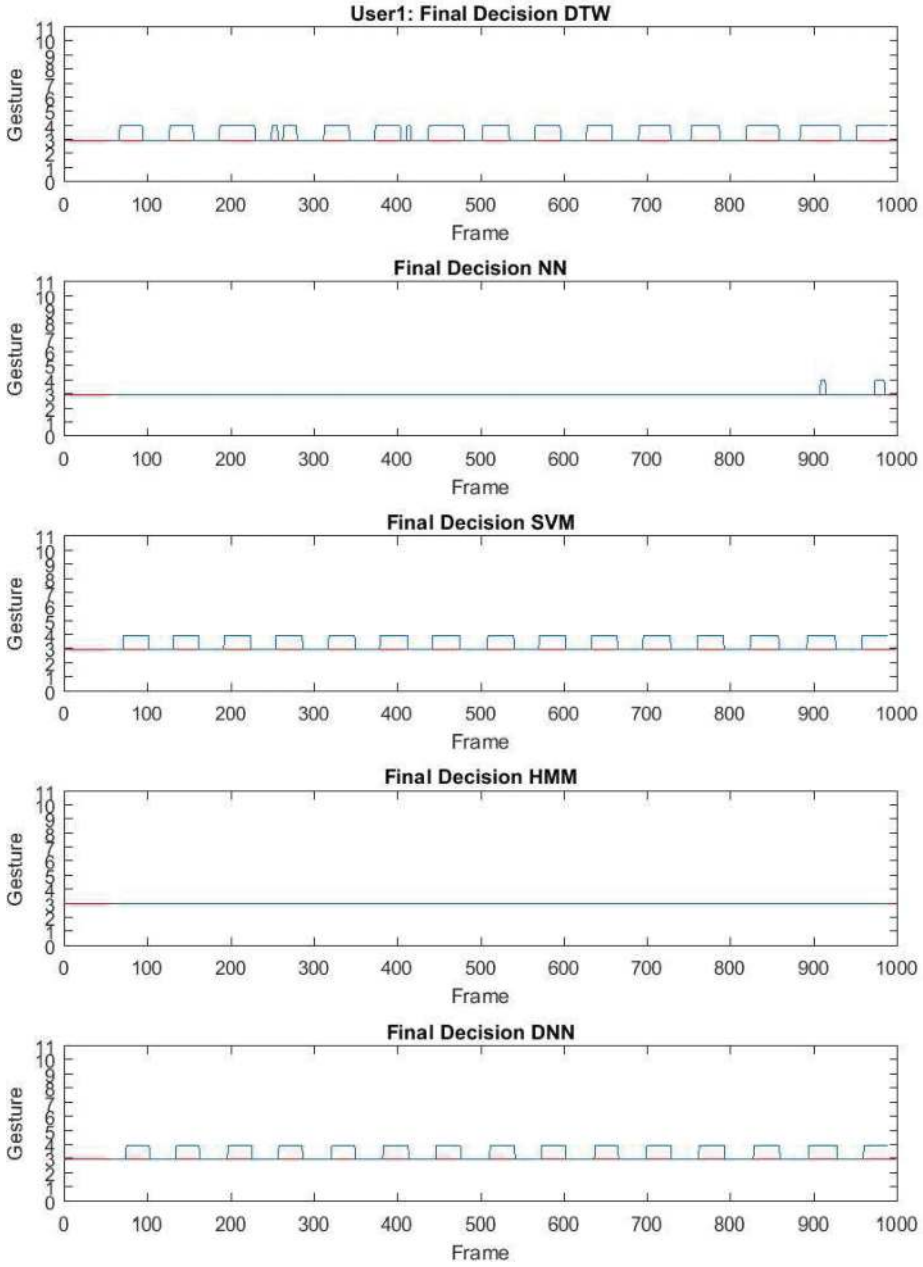

Figure 9. Recognition results relative to the first 1000 frames of a test sequence relative to gesture G_3 performed by User 1. The x-axis represents the frame number of the sequence and the y-axis represents the gesture classes ranging from 1 to 10 (the range 0–11 has been used only for displaying purposes). The red line denotes the ground truth label (G_3 in this case), whereas the blue one represents the predicted labels obtained from the classifiers.

http://dx.doi.org/10.5772/68118

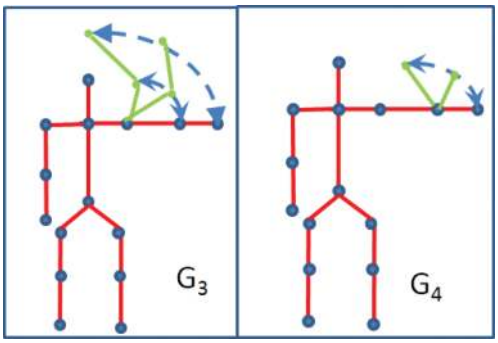

Figure 10. Gesture G_3 and G_4. Both gestures involve a rotation of the arm in a plane parallel to the camera.

interface. The recognition is highly influenced by the following elements: the subjectivity of the users, the complexity of the gestures, and the recognition performance of the applied methodology. In order to give an overall evaluation of the experimental results, a statistical analysis of the conducted tests has to be done. The F-score, also known as F-measure or F_1-score, has been considered as global performance metrics [42]. It is defined by the following equation:

$$F = \frac{2TP}{2TP + FP + FN} \tag{5}$$

where TP, FP, and FN are the true positives, false positives, and false negatives, respectively. The best values for the F-score are those close to 1, whereas the worst are those close to 0. This measure captures information mainly on how well a model handles positive examples.

Figure 13 shows the F-score values obtained for each methodology and for each gesture, averaged over all users. As can be seen each methodology behaves differently among the set of available gestures: SVM, for example, has an F-score close to 1 for G_1 and G_8, whereas DNN has maximum F-score in the case of G_2 or G_4. **Figure 13** highlights another important aspect: some gestures are better recognized instead of others. This is the case, for example of G_8 or G_4 for which the F-scores reaches high values whatever methodology is applied. On the contrary, gestures such as G_5 or G_7 are generally badly recognized by each methodology. These considerations are very useful as allows us to select a subset of gestures and for each of them the best methodology in order to build a robust human robot interface. To this aim, a threshold $(= 0.85)$ can be fixed for the F-score values and the gestures that have at least one classifier with F-score above this threshold can be selected. By seeing **Figure 13**, these gestures are: $G_1, G_2, G_4, G_8, G_9,$ and G_{10}. For each selected gesture the classifier with the maximum F-score can be chosen: so SVM for G_1, DNN for G_2 and G_4, SVM for G_8, DTW for G_9, and finally SVM for G_{10}. These set of gestures with the relative best classifiers can be used to build the human-robot interface.

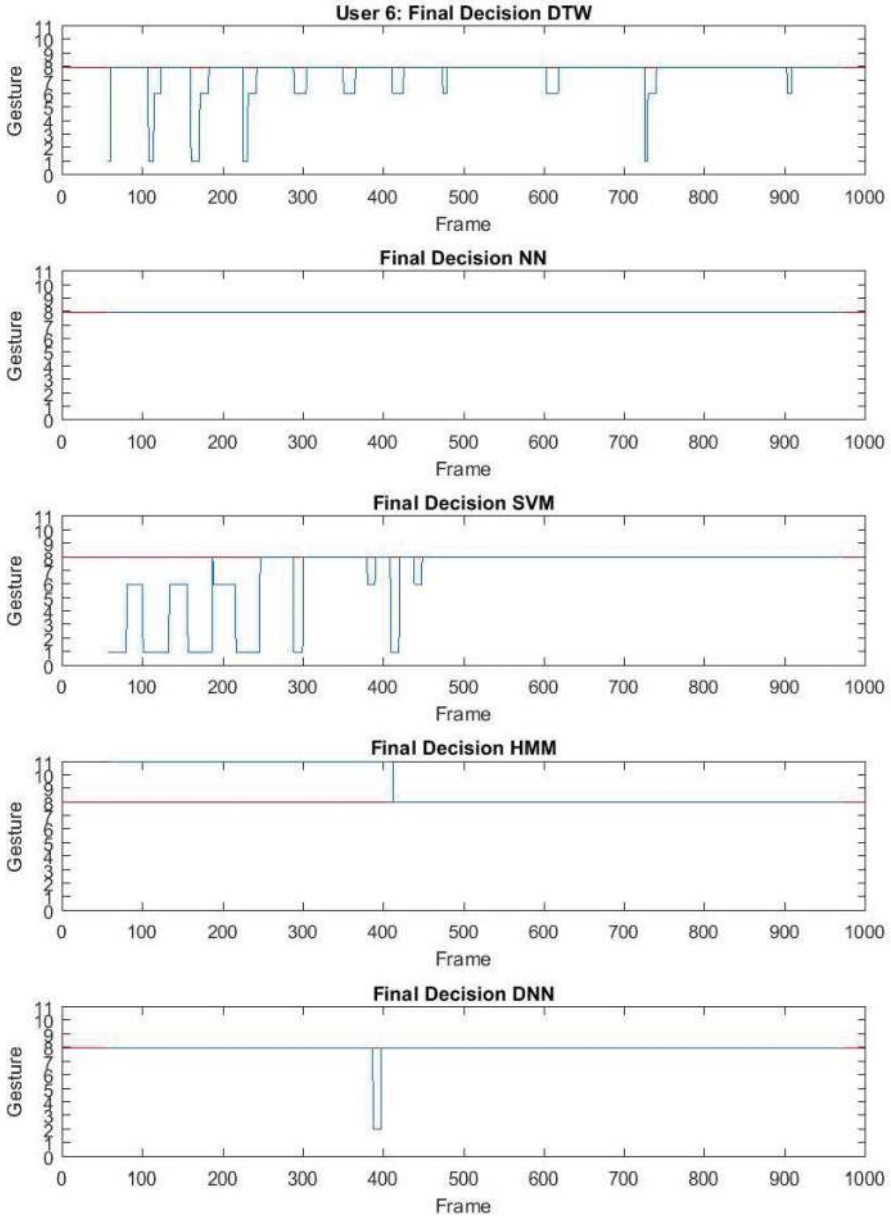

Figure 11. Recognition results relative to the first 1000 frames of a test sequence relative to gesture performed by User 6. The x-axis represents the frame number of the sequence and the y-axis represents the gesture classes ranging from 1 to 10 (the range 0 -11 has been used only for displaying purposes). The red line denotes the ground truth label (in this case), whereas the blue one represents the predicted labels obtained from the classifiers.

http://dx.doi.org/10.5772/68118

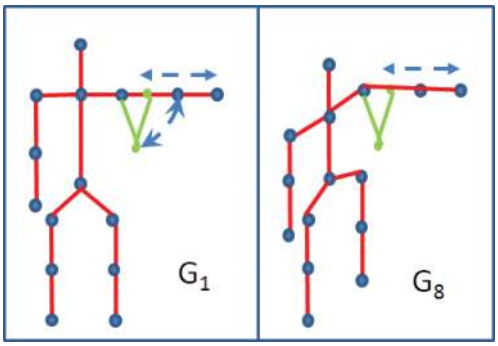

Figure 12. Gestures G_1 and G_8. Gesture G_1 involves a movement in a plane parallel to the camera, whereas gesture G_8 involves a movement in a plane perpendicular to the camera.

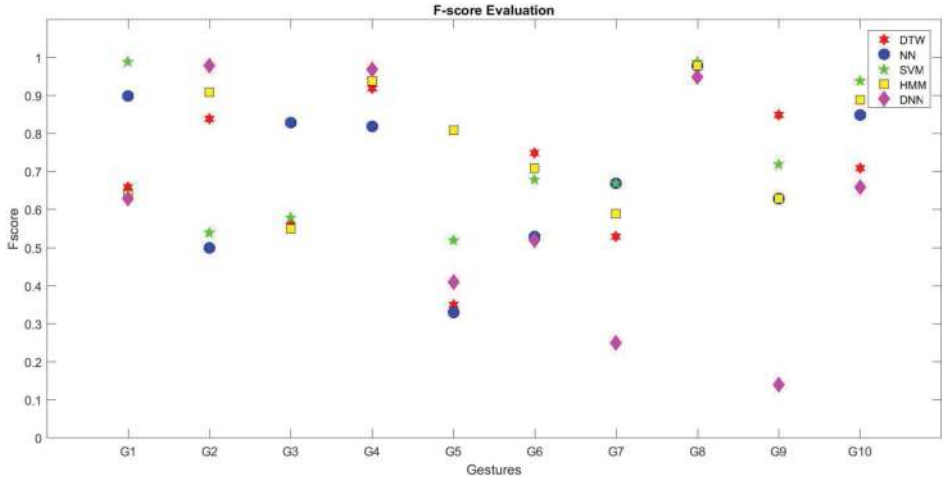

Figure 13. F-score values of all methodologies for each gesture averaged over all users.

7. Conclusions

In this chapter the problem of Gesture Recognition has been considered. Different methodologies have been tested in order to analyze the behaviors of the differently obtained classifiers. In particular, neural network (NN), support vector machine (SVM), hidden Markov model (HMM), deep neural network (DNN), and dynamic time warping (DTW) approaches have been applied.

The results obtained during the experimental phase prove the great heterogeneity of tested classifiers. In this work, the majority of problems arise in part from the complexity of the

gestures and in part from the variations coming from the users. The classifiers perform differently often preserving complementarity and redundancy. These peculiarities are very important for fusion. So, encouraged by these observations, we will concentrate our further investigations on the fusion of different classifiers in order to improve the overall performance and reduce the total error.

Author details

Grazia Cicirelli* and Tiziana D'Orazio

*Address all correspondence to: cicirelli@ba.issia.cnr.it

Institute of Intelligent Systems for Automation, National Research Council of Italy, Bari, Italy

References

[1] Habib Z, Bux A, Angelov P. Vision Based Human Activity Recognition: A Review. Vol. 513. Cham: Springer; 2017. pp. 341-371

[2] Hassan MH, Mishra PK. Hand gesture modeling and recognition using geometric features: A review. Canadian Journal on Image Processing and Computer Vision. 2012;**3**(1):12-26

[3] Jang F, Zhang S, Wu S, Gao Y, Zhao D. Multi-layered gesture recognition with kinect. Journal of Machine Learning Research. 2015;**16**:227-254

[4] Traver VJ, Latorre-Carmona P, Salvador-Balaguer E, Pla F, Javidi B. Three-dimensional integral imaging for gesture recognition under occlusions. IEEE Signal Processing Letters. 2017;**24**(2):171-175

[5] D'Orazio T, Marani R, Renó V, Cicirelli G. Recent trends in gesture recognition: How depth data has improved classical approaches. Image and Vision Computing. 2016;**52**:56-72

[6] Presti LL, Cascia ML. 3D skeleton-based human action classification: A survey. Pattern Recognition. 2016;**53**:130-147

[7] Cheng H, Yang L, Liu Z. Survey on 3d hand gesture recognition. IEEE Transactions on Circuits and Systems for Video Technology. 2016;**26**(9):1659-1673

[8] Aggarwal JK and Xia L. Human activity recognition from 3D data: A review. Pattern Recognition Letters. Oct. 2014;**48**:70-80

[9] Cruz L, Lucio F, Velho L. Kinect and RGBD images: Challenges and applications. In: Proceedings of 25th SIBGRAPI IEEE Conference on Graphics, Patterns and Image Tutorials, pp. 36-49, IEEE Computer Society, Los Alamitos, USA, 2012

[10] Almetwally I, Mallem M. Real-time tele-operation and tele-walking of humanoid robot Nao using Kinect depth camera. In: Proceedings of 10th IEEE International Conference

on Networking, Sensing and Control (ICNSC), pp. 463-466, IEEE Computer Society, Los Alamitos, 2013

[11] Jacob MG, Wachs JP. Context-based hand gesture reognition for the operating room. Pattern Recognition Letters. 2014;**36**:196-203

[12] Wang C, Liu Z, Chan SC. Superpixel-based hand gesture recognition with kinect depth camera. IEEE Transactions on Multimedia. 2015;**17**(1):29-39

[13] Plouffe G, Cretu A-M. Static and dynamic hand gesture recognition in depth data using dynamic time warping. IEEE Transactions on Instrumentation and Measurement. 2016;**65** (2):305-316

[14] Venkataraman V, Turaga P. Shape distributions of nonlinear dynamical systems for video-based inference. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016;**38**(12):2531-2543

[15] Lai K, Konrad J, Ishwar P. A gesture-driven computer interface using kinect. In: Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI); IEEE Computer Society, Los Alamitos, USA, April 2012; 2012. pp. 185-188

[16] Oh J, Kim T, Hong H. Using binary decision tree and multiclass SVM for human gesture recognition. In: Proc. of the IEEE International Conference on Information Science and Applications (ICISA); IEEE Computer Society, Los Alamitos, USA, June 2013; 2013; pp. 1-4

[17] Pal M, Saha S, Konar A. Distance matching based gesture recognition for healthcare using microsoft's kinect sensor. In: Proc. of International Conference on Microelectronics, Computing and Communications (MicroCom); IEEE Computer Society, Los Alamitos, USA, 23–25 June 2016, pp. 1-6

[18] Deo N, Rangesh A, Trivedi M. In-vehicle hand gesture recognition using hidden markov models. In: Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems (ITSC); IEEE Computer Society, Los Alamitos, USA, 1–4 November 2016; 2016. pp. 2179-2184

[19] Song Y, Gu Y, Wang P, Liu Y, Li A. A kinect based gesture recognition algorithm using GMM and HMM. In: Proceedings of the 6th International Conference on Biomedical Engineering and Informatics (BMEI); 16-18 December 2013, IEEE Computer Society, Los Alamitos, USA, pp. 750-754

[20] Miranda L, Vieira T, Martinez D, Lewiner T, Vieira A, Campos M. Online gesture recognition from pose kernel learning and decision forests. Pattern Recognition Letters. April 2014;**39**:65-73

[21] Ghosh DK, Ari S. Static hand gesture recognition using mixture of features and SVM classifier. In: Proceedings of the 5th IEEE International Conference on Communication Systems and Network Technologies; IEEE Computer Society, Los Alamitos, USA, 4–6 April 2015; 2015. pp. 1094-1099

[22] Bhattacharya S, Czejdo B, Perez N. Gesture classification with machine learning using kinect sensor data. In: 3rd International Conference on Emerging Applications of Information

Technology (EAIT); November 30- December 01, 2012, IEEE Computer Society, Kolkata, India, pp. 348-351

[23] Althloothi S, Mahoor MH, Zhang X, Voyles RM. Human activity recognition using multi-features and multiple kernel learning. Pattern Recognition. 2014;**47**:1800-1812

[24] Ibraheem NA, Khan RZ. Vision based gesture recognition using neural networks approaches: A review. International Journal of Human Computer Interaction. 2012;**3**(1):1-14

[25] Cicirelli G, Attolico C, Guaragnella C, D'Orazio T. A kinect-based gesture recognition approach for a natural human robot interface. International Journal of Advanced Robotic Systems. 2015;**12**(3).

[26] Ruan X, Tian C. Dynamic gesture recognition based on improved DTW algorithm. In: Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA); IEEE Computer Society, Los Alamitos, USA, 2–5 August 2015; 2015. pp. 2134-2138

[27] Ding IJ, Chang CW. An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data. Applied Mathematical Modelling. 2015;**39**(19):5769-5777

[28] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;**521**(7553):436-444

[29] Neverova N, Wolf C, Taylor G, Nebout F. Mod drop: Adaptive multi-modal gesture recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. August 2016;**38**(8):1692-1706

[30] Wu D, Pigou L, Kindermans PJ, Le N, Shao L, Dambre J, Odobez JM. Deep dynamic neural networks for multimodal gesture segmentation and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. August 2016;**38**(8):1583-1597

[31] D'Orazio T, Attolico C, Cicirelli G, Guaragnella C. A neural network approach for human gesture recognition with a kinect sensor. In International Conference on Pattern Recognition Applications and Methods (ICPRAM); March 2014; Angers, France. SCITEPRESS - Sience and Technology Publications, Setubal, Portugal, pp. 741-746

[32] Haykin S. Neural Networks-A Comprehensive Foundation. 2nd ed. Prentice Hall PTR Upper Saddle River, NJ, USA, 1998

[33] Vapnik V. The Nature of Statistical Learning Theory. Berlin: Spinger; 1995

[34] Rabiner LR. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE. February 1989;**77**(2):257-286

[35] Ghahramani Z. An introduction to Hidden Markov Models and Bayesian Networks. International Journal of Pattern Recognition and Artificial Intelligence. 2001;**15**(1):9-42

[36] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. July 2006;**31**3:504-507

[37] Møller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks. 1993;**6**(4):525-533

[38] Müller M. Dynamic Time Warping, Information Retrieval for Music and Motion, Springer-Verlag Berlin Heidelberg, 2007, pp. 69-84

[39] Rabiner L, Juang B-H. Fundamentals of Speech Recognition. Englewood Cliffs, NJ: Prentice Hall PTR; 1993

[40] Headquarters Department of the Army. Visual Signals: Arm-and-Hand Signals for Ground Forces. Field Manual FM 21-60, Washington, DC, September 1987. This report is downloadable at: http://www.apd.army.mil/epubs/DR_pubs/DR_a/pdf/web/fm21_60.pdf

[41] Attolico C., Cicirelli G., Guaragnella C., D'Orazio T. (2015) A Real Time Gesture Recognition System for Human Computer Interaction. In: Schwenker F., Scherer S., Morency LP. (eds) Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction. MPRSS 2014. Lecture Notes in Computer Science, vol 8869. Springer, Cham

[42] Powers DMW. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation. Technical report SIE-07-001, School of Informatics and Engineering Flinders University, Adelaide, Australia, December 2007