Action Quality Assessment on Tennis Strokes executions using a BiLSTM-based Deep Learning Framework

Cecilia Assolito, Antonello Giorgio

Abstract

How often does an amateur tennis player want to compare his own execution with that of the world's greatest tennis players, such as Federer, Nadal or Djokovic? How often does a tennis school teacher watch his students hit the ball and give advice on how to improve their game based only on their eyesight? The goal we set with this project is to automatically perform an action quality assessment on the execution of a tennis stroke, compared to the one of the experts in the field. To achieve this, we developed a Deep Learning Model with a BiLSTM-based architecture. Starting from a video recording, it produces a score that evaluates how closely the movement approximates the one of a professional. In particular, the analysis of a stroke execution is focused on studying the evolution over time of the positions of different body parts, obtained through the pre-trained model *OpenPose*. This information has been processed in such a way that the model is aware of both the importance of certain joints during the movement (through the development of Principal Component Analysis) and the temporal instants in which they have the greatest impact on the overall execution (through discrete-time analysis of the change in position of each joint). The dataset used for training the model is *THETIS*: it contains videos of amateurs and experts for each tennis stroke. This subdivision is crucial for the acquisition of the reference scores, which are obtained by analyzing the performance of each execution against all the expert ones.

Keywords

Action Quality Assessment (AQA), OpenPose, BiLSTM, PCA, Discrete-Time Analysis

1. HIGHLIGHTS

- unprecedented solution for the Action Quality Assessment task over the Tennis domain, with capabilities to evaluate all the existing strokes (10 different types);
- computation of quality scores through statistical analysis, which ensures objectivity in evaluations, and creation of weights that emphasize the most relevant body parts during the execution of the stroke;
- developing of deep neural network models, composed by one or two BiLSTMs, capable of producing a quality score prediction given input skeletal data;
- comparison with Liao *et al.*'s "SpatioTemporalNN" model applied on the THETIS dataset;
- BiLSTM with weights as the model with the best results, outdoing SpatioTemporalNN performance.

2. Introduction

In the world of sports, the subjectivity of technical evaluations made by judges during a competition, or by coaches during athletic preparation, has always been a sore point. The human eye is easily influenced both by the outside world, through frequent optical illusions that can distort the perception of real world, and by psycho-physical conditions peculiar to humans, such as fatigue. This obviously affects the truthfulness of both judgments made in the context of a competition and technical indications given in order to improve the single movements in training. In recent years, there have been many attempts to introduce automatic devices or systems capable of producing unbiased assessments. Among all, the only one that has so far managed to gain official approval from the sports federations has been the so-called *Hawk-Eye* [1]. It is commonly used in tennis [2] and cricket, and is capable of reproducing the trajectory taken by the ball and declaring the validity of the shot. However, the results obtained in analyzing the accuracy of an athlete's gestures are totally different. In fact, we are still far from being able to rely completely on automated systems. Nowadays, the best technologies that can give complete feedback on the execution of a movement are biomechanical feedback platforms. They process data obtained through wearable sensors and video recordings to better monitor acceleration, angle, symmetry, and many other body movement data. There are several issues that arise from the use of these devices. First, they could lead to wrong measurements caused by athletes not being comfortable with wearing them while executing actions. Moreover, the interpretation of the obtained results still requires human intervention and experience, due to the inability of the adopted systems to be decontextualized and versatile. So, how to objectively measure the level of an individual performance is still an open question. The necessity to

EAI - AI for Visual Perception in HCI HRI, Final Project

CEUR Workshop Proceedings (CEUR-WS.org)

find an answer to this has led to develop solutions for the *Action Quality Assessment* (AQA) task.

In general, AQA aims to quantify how well an action was performed, by making quantitative and qualitative assessments about the entire action sequence. Specifically, this task requires the construction of a model that is able to analyze videos, describing a given action, and generate a performance score. In the sports domain, AQA has multiple applications: from diving to synchronized swimming, from artistic gymnastics to track and field, and so on. In our work, we decided to build a framework where the action to be evaluated concerns the various strokes executed by tennis players during a match. Due to the paucity of available datasets and the challenging process of finding unambiguous metrics, this task has been poorly investigated so far. Our project perfectly fits into this context, aiming to quantify the quality of each action and use these measurements to build and train deep-learning models capable of predicting the level of the performed stroke. While many implementations of AQA rely directly on frame images, our choice was to work on the skeleton structure of each player, extracted for each examined recording. The pose estimation is performed on the videos provided by the *THETIS* [3] dataset, through the pre-trained model *OpenPose* [4]. This gives precise information on the position and orientation of human body joints, allowing the progress in time and space of a movement to be accurately tracked. The obtained patterns were statistically analysed to measure objectively the quality of the stroke, and re-elaborated through the Principal Component Analysis (PCA) technique to extract the more relevant body parts in the gesture execution. Spatio-temporal information is also crucial in developing our deep neural network. In fact, it analyzes the temporal sequences through BiLSTM-based architectures, as well as exploiting *joint evolution weights* to confer more attention to the most relevant frames. In this way, our system is able to fully understand the spatial and temporal variability in players' movements. In particular, we experimented several models:

- two BiLSTMs,
- two BiLSTMs with PCA,
- two BiLSTMs with joint evolution weights,
- two BiLSTMs with PCA and joint evolution weights,
- one BiLSTM,
- one BiLSTM with PCA,
- one BiLSTM with joint evolution weights,
- one BiLSTM with PCA and joint evolution weights.

To sum up, the report is structured as follows: in Section 3, a brief description of related works is presented. Section 4 details our solution, explaining the preprocessing of the data, the statistical metrics used for evaluating and the different versions of the deep neural network implemented. Section 5 provides the results obtained with each experiment and the comparison with the literature. Section 6 illustrates the concluding remarks for this study.

3. Related Works

AQA task attempts to build a system able to automatically and objectively evaluate actions performed by people. It has a very restricted number of applications, considering that most of the existing works focuses only on sports and medical care. In this section, we show the most relevant papers on action quality assessment.

Liao *et al.* [5] proposed a solution for action quality assessment in the field of rehabilitation. By exploiting skeletal data coming from a dataset of ten rehabilitation exercises, they built a deep neural network able to automatically generate quality scores for the analyzed movement. In particular, it is composed of convolutional filters in a multi-branch design to extract features from each body part, four LSTM layers to study temporal correlations and a linear regression layer to compute the final output. A fundamental role in the whole prediction process is played by the convolutional filters. In fact, by leveraging temporal pyramid subnetworks [6], different movement patterns for each body part are analyzed to produce deeper representations, later used for the recurrent layers as input. In other words, temporal pyramid subnetworks help in processing input sequences by dynamically subsampling them at varying frame rates. In this case, three reduced versions of the input were used, with a temporal length equal to one half, one quarter, and one eight of the sequence. The whole neural network performance was measured comparing the output scores with statistical quantifications of the patient's execution. The latter were obtained through a preprocessing of the joint coordinates data. This phase involves a dimensionality reduction achieved with deep autoencoder NNs, followed by the use of metrics, based on the Gaussian mixture model (GMM) [7] log-likelihood, to quantify the subject's action correctness. Then, the intermediate values of the performance metrics are mapped by scoring functions into movement quality scores in the range between 0 and 1, allowing the training of the NN model.

Contrary to this, Hiteshi *et al.* [8] computed the scores directly on raw video recordings. This can be done by employing the so-called Siamese Network, which is able to learn a similarity metric between two input sequences. This deep metric learning-based (DML) approach is very effective, since it exploits two branches of the same network to analyze the action video and the reference video in a parallel manner. The human action scoring system is performed on a dataset of Olympics Diving and Gymnastic vaults, and it is divided in two phases. The first one takes two random video recordings as input, and produces similarity/dissimilarity scores through a sequence of dense and Sigmoid layers. Moreover, the training related to this phase fixes the LSTM-Siamese weights for the second one, in which the two input sequences are no more random. In fact, one of them has to be associated with an expert video, distinguished from the other ones based on a threshold applied to the first-phase scores. The weights that need to be trained can be found in the last part of the architecture, which now involves a fully connected layer instead of the sigmoid one. This allows the generation of scores that can be compared with the ones provided by human judges.

All the applications discussed so far never tapped into the quality assessment of the tennis strokes. Bailey *et al.* [9] attempted to devise a method to distinguish a tennis player's level by inspecting just the execution of the serve. Unlike the previous systems, the focus was only on the statistical analysis of the poses, estimated with the help of AlphaPose [10]. This pre-trained model was applied to the frames of each video, collected both from Youtube and by themselves, to obtain x-y coordinates of the players' limbs. Then, by normalizing all the data, they computed the Euclidean distance between two compared players' joints. By exploiting the sum of all these measurements, they analyzed the total joint dynamics through the Welch's t-test method. Based only on this experiment, they quickly managed to define a way to recognize the skill level of players from a generic video. Furthermore, with several other t-tests, they demonstrated the significance of the change in position of the left shoulder, compared to the rest of the joints. Eventually, these trials emphasized the differences between amateur and professional players while performing the serve.

4. Proposed method

Our framework aims to evaluate the level of tennis shots execution, exploiting a 2D skeleton model. Taken a set of tennis strokes videos, the starting point of the work is represented by the extraction of the player's body-joint coordinates, from each single frame. They are subsequently preprocessed through normalization and scaling, and then used to perform Principal Component Analysis, Joint Evolution Weights generation and performance quantification. From the latter, we obtain quality scores needed as true labels for the training of our deep neural network model, which can be based on one BiLSTM or a combination of two BiLSTMs.

4.1. Dataset

The set of data used to build and validate our framework is the sport-based human action dataset *Three Dimensional Tennis Shots* (THETIS). It is composed of videos representing the 12 basic tennis shots, performed by 31 amateurs and 24 experienced players. In particular, the different tennis stroke classes include: backhand with two hands, backhand, backhand slice, backhand volley, forehand flat, forehand open stands, forehand slice, forehand volley, service flat, service kick, service slice, smash. The total number of recordings provided is 8734, among which 1980 are RGB, 1980 are depth, 1980 are silhouettes, 1217 are 2D skeleton and 1217 are 3D skeleton. Since the calibration pose, used to generate the 2D and 3D skeleton sequences, was not always successful, the number of elements included in the last two categories is smaller. This has led our analysis towards the use of body joints information extracted from the RGB videos, taking advantage of a more reliable pose estimator, *Openpose* (details in Section 4.2.1).

4.2. Preprocessing

4.2.1. Body Joints Extraction

The human body can perform complex movements thanks to the articulations, which give support and flexibility. This means that the study of each body joint position in space, during the execution of an action, guarantees a complete understanding of the motion. Therefore, our assumption consists in considering the quality of an action directly dependent on the changes over time of each joint coordinate. This kind of analysis is possible only by conducting a pose estimation task through the learning of the position of each body part at every instant. Our choice was to use the first realtime multi-person skeleton detection system *Openpose*, developed by Carnegie Mellon University (CMU). It is a pre-trained model able to take a RGB video as input and detect the anatomical key-points of people in it. The features can be extracted in different ways, depending on the type of analysis to be carried out. In this project, we have taken advantage of the 2D COCO-18 body joints detection model, which generates skeletal data characterized by 18 key-points, as illustrated in Figure 1.

So, through OpenPose, we were able to represent each video **v** as a $N \times M$ matrix

$$\mathbf{v} = \begin{bmatrix} (x_{11}, y_{11}) & (x_{12}, y_{12}) & \dots & (x_{1M}, y_{1M}) \\ (x_{21}, y_{21}) & (x_{22}, y_{22}) & \dots & (x_{2M}, y_{2M}) \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ (x_{N1}, y_{N1}) & (x_{N2}, y_{N2}) & \dots & (x_{NM}, y_{NM}) \end{bmatrix}$$

where N is the number of frames and $M = 18$ cor-

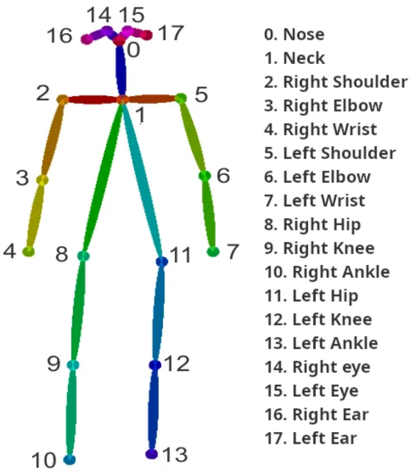


Figure 1: OpenPose COCO-18 skeleton model

responds to the number of body joints. Each element (x_{ij}, y_{ij}) of this matrix, with $i \in [1, N]$ and $j \in [1, M]$, represents the coordinates that the j-th body joint assumes at the i-th time instant (frame).

A final correction of the data was necessary, because OpenPose, in the presence of measurement errors in the algorithm itself or due to external factors that prevent the correct detection of keypoints (occlusions, chromatic aberrations, etc.), fails and assigns zero as coordinate value. In this case, our choice was to replace it with the mean of its neighbors.

4.2.2. Normalization and scaling

Once body joints have been obtained for every execution, the goal is to make them comparable. Since the number of frames can vary, a first step consists in reshaping each video to impose a predefined time length. In particular, the process of adaptation involved the linear interpolation operation to bring each sequence to the maximum length found in the dataset. Indeed, this technique aims at finding new points in the time evolution path of each body joint, starting with a mapping function that expresses the relationship between joint value and the time instant in which the coordinate was computed.

At this point time evolutions are comparable, but the inconsistency of the data still remains, due to differences in space location and body dimension of players. To handle this issue, normalization and scaling are necessary. The first process involves the *Min-Max Normalization*, applied separately to each x and y coordinates, in order to shift all the measurements in the same position of the plane. Instead, the latter one exploits both the shoulders and the neck-hip mean distances to scale, respectively, x and y coordinates. In this way, every body shape becomes proportional to the same reference metrics, while preserving the distance relationships between body parts of the original figure. Now, movement features are coherent among them and can be analyzed, in order to acquire important knowledge about the evolution in time of the different joints (as explained in the following sections 4.2.3, 4.2.4, 4.2.5).

4.2.3. Principal Component Analysis

During the execution of an action, each body part contributes in a different way. Depending on the type of motion, there could be some joints that participate more w.r.t. others. In the context of evaluating a gesture, knowing which key-points are more relevant in the whole movement can improve the final assessment, making it more precise and accurate. In order to develop this analysis and capture well how much each joint moves during the shot execution, we measured the Euclidean distance between their position at time $t-1$ and the one at time t (where $t \in [1, N]$). Then, we used these distances to obtain a mapping from each joint to its position variation (Euclidean distance) for every discrete time instant. Considering this new representation of the data, the best way to highlight the body parts with the highest variability in the movement is to apply the Principal Component Analysis (PCA).

PCA is one of the most common statistical techniques for dimensionality reduction. It is defined as an orthogonal linear mapping of a high-dimensional space into a lower-dimensional one, giving a new coordinate system to the measurements. The mechanism is based on the singular value decomposition: the reduced data will be composed by the K-eigenvectors scaled by their singular values (square root of eigenvalues) corresponding to the K-highest eigenvalues (representing the K-highest variances of the components), where K changes according to the implementational needs. Since every tennis shot can correspond to a different number of principal components, we applied the procedure twice. A first PCA computation was performed to find out the minimum number of components K that allows us to preserve at least 95% of the variance for each video. Then, a second implementation was necessary to obtain the final K-dimensional data.

4.2.4. Joint Evolution Weights

Based on the same premises made in Section 4.2.3, we can say that a significant change of the single joint coordinates between consecutive frames bring to peaks that can be exploited to emphasize a subset of joints in each frame. Incorporating this concept into the network training process results in building weights that allow the system to focus more on the articulations that are most involved in the execution.

Starting with the video representation shown in Section 4.2.1, it is clear that each column of $\mathbf{v}$ corresponds to a *joint evolution*, that is the change in time of a particular key-point position. Considering that joint evolutions shape discrete-time trajectories, a simple and effective way to analyze them is to take the norm of each body joint $norm_{ij} = norm(\mathbf{y_{ij}} - \mathbf{x_{ij}}) = \sqrt{x_{ij}{}^2 + y_{ij}{}^2}$, and compute the absolute derivatives, approximated as forward differences, between consecutive norms corresponding to the same joint. Once the sequences of derivatives were acquired, a time-window has been slid through them, in order to catch the most varying parts of the trajectory. In particular, the sum of absolute finite differences along the joint evolution in the time-window was used to choose the time intervals in which the position displacements have the highest variability. The latter were properly detected by avoiding overlapping windows. We accepted two common instants between candidate time intervals at most. Once we acquired this knowledge, we were able to build the joint evolution weight vectors by taking the product between the pre-computed absolute derivatives and a multiplicative factor, essential to emphasize the instants with the highest variation.

4.2.5. Performance Metrics

A reliable quantification mechanism to evaluate the stroke execution level is essential to develop a completely objective action quality assessment task. The possible performance metrics can be divided in two different categories: model-less and model-based [11]. While the latter is more focused on a probabilistic analysis of the data (examples are Gaussian Mixture Models or Hidden Markov Models), the first one uses distance functions to compare the single performance with a reference one. Since, in our case, there is the set of expert executions that can be a suitable benchmark, we decided to apply the *Mahalanobis distance*. It is a statistical measure that, given a distribution D in R^N, with mean

$$\mu = \left(\mu_1, \mu_2, ..., \mu_N\right)^T$$

and positive-definite covariance matrix S, computes the distance of a point

$$\mathbf{x} = \left(x_1, x_2, ..., x_M\right)^T$$

from D:

$$d_M(\mathbf{x}, D) = \sqrt{(\mathbf{x} - \mu)^T * S^(-1) * (\mathbf{x} - \mu)}$$

Assuming

- $\mathbf{x}$ as a single frame of one shot execution,
- D as the set containing all the experts coordinates related to the same time instant and the same tennis stroke,

by averaging all the distances obtained for each video, we were able to measure how much each single gesture is different w.r.t. the one performed by experts. Mapping all the data into the [0,1] range, we obtained an objective score for every tennis shot performance.

4.3. Model description

When we have to deal with time series, BiLSTM architecture is a natural choice. *Long short-term memory (LSTM)* [12] is a particular type of Recurrent Neural Network (RNN) that aims to overcome the vanishing gradient problem by providing and updating a short-term memory cell over time. *Bidirectional LSTM (BiLSTM)* analyzes both forward and backward information flow, enabling the model to capture context both from past and future sequences. The model is further enhanced with dropout layer for regularization, layer normalization for stable training, and fully connected layers for final output generation and intermediate steps.

4.3.1. *Notation*

Let $\mathbf{v}$ be the input video as a sequence of normalized and scaled (as we described in Section 4.2.2) skeletal data of shape $[N, M]$, as shown in 4.2.1 . The dropout layer is denoted by D. The number of the i-th BiLSTM layers is denoted by num_layers_i. The i-th BiLSTM layers transform the input sequence into hidden states H_i. When only the final hidden states from both directions are concatenated to form the final output of the BiLSTM, it is denoted by

$$h_{i_f} = [H_{i_{fwd}}[num\_layers_i]; H_{i_{bwd}}[num\_layers_i]].$$

The i-th dense layer is denoted by $FC_i(x) = \sigma_i(W_i x_i + b_i)$, where σ_i is the activation function, W_i is the weight matrix, and b_i is the bias vector. PCA components, when used, are denoted by P, with a shape dependent on the number of components obtained in the procedure explained in Section 4.2.3. Joint Evolution Weights, when used, are denoted by the $M \times N$ matrix J.

4.3.2. Two BiLSTMs

In the case of a two BiLSTMs-based architecture, the model passes the input to the Dropout layer

$$\tilde{\mathbf{v}} = D(\mathbf{v}), \tag{1}$$

in order to randomly drop some joints and better improve the generalization capabilities of the network, processes the input through the first BiLSTM

$$H_1 = \text{BiLSTM}_1(\tilde{\mathbf{v}}), \tag{2}$$

applies layer normalization

$$H_1' = \text{LayerNorm}(H_1), \tag{3}$$

and the first dense layer

$$FC_1 = \sigma_1(W_1 H_1' + b_1), \tag{4}$$

with

$$\text{LayerNorm}(x) = \frac{a \cdot (x - \mu)}{\sqrt{\sigma^2 + \epsilon}} + b$$

where

- x is the input of the layer.
- μ is the mean of elements in x.
- σ^2 is the variance of elements in x.
- ϵ is a small constant value used to avoid division by zero.
- a is the scale weight learned by the model.
- b is the bias learned by the model.

and

$$\sigma_1 = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

that maps real numbers into the range $[-1; 1]$. Then, the output of the first fully-connected layer is used by the second BiLSTM to obtain a final representation for the whole time series

$$h_{2_f} = \text{BiLSTM}_2(FC_1). \tag{5}$$

The final score is retrieved from the second dense layer

$$output\_score = FC_2 = \sigma_2(W_2 h_{2_f} + b_2), \tag{6}$$

with the Sigmoid activation function

$$\sigma_2 = \frac{1}{1 + e^{-x}}$$

that maps real numbers into the range $[0; 1]$.

When PCA-transformed input data are used as residual connection, the network benefits from processing a compressed representation of the input features, which enriches the pool of information that the model can use for the prediction process. By concatenating these PCA components to the first BiLSTM output, the second BiLSTM can leverage these enhanced feature representations, potentially improving the model's learning efficiency. In practice, this implies the insertion of the concatenation

$$O_{\text{PCA}} = [FC_1; P]$$

between (4) and (5), and the update of (5) in

$$h_{2_f} = \text{BiLSTM}_2(O_{\text{PCA}})$$

Finally, the network can exploit the *joint evolution weights* (described in Section 4.2.4) by applying them to the input of the model with an element-wise product. This results in the computation of the weighted skeletal data

$$\mathbf{v}_w = \mathbf{v} \odot J^T$$

and the update of (1) and (2) in

$$\tilde{\mathbf{v}}_w = D(\mathbf{v}_w)$$

$$H_1 = \text{BiLSTM}_1(\tilde{\mathbf{v}}_w).$$

To see a diagram representation of the whole architecture, check Fig. 2.

4.3.3. One BiLSTM

In the case of a one BiLSTM-based architecture, the steps remain the same w.r.t. the ones shown in Section 4.3.2, except for the processing of the output of the first BiLSTM. Indeed, it is passed directly through the dense layer that produces the final scores. Therefore, the sequence of steps involved is the following:

$$\tilde{\mathbf{v}} = D(\mathbf{v}) \tag{7}$$

$$H_1 = \text{BiLSTM}_1(\tilde{\mathbf{v}}) \tag{8}$$

$$H_1' = \text{LayerNorm}(H_1) \tag{9}$$

$$output\_score = FC_1 = \sigma_1(W_1 h_{1_f} + b_1), \tag{10}$$

where h_{1_f} is composed of elements taken from H_1' and

$$\sigma_1 = \frac{1}{1 + e^{-x}}$$

is the Sigmoid activation function that maps real numbers into the range $[0; 1]$.

When PCA-transformed input data are used as residual connection,

$$O_{\text{PCA}} = [H_1'; P]$$

and (10) becomes

$$output\_score = FC_1 = \sigma_1(W_1 O_{\text{PCA}} + b_1).$$

Regarding the introduction of *joint evolution weights* in this network, the approach does not change w.r.t. the one explained in Section 4.3.2.

To see a diagram representation of the whole architecture, check Fig. 3.

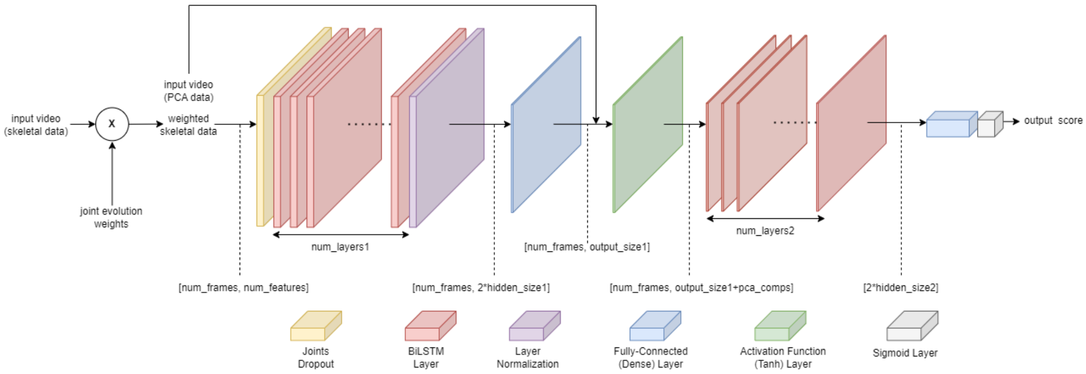


Figure 2: Architecture Diagram for the DNN model version with two BiLSTMs, joint evolution weights multiplied to the input skeletal data and the PCA inputs, concatenated to the output of the first dense layer.

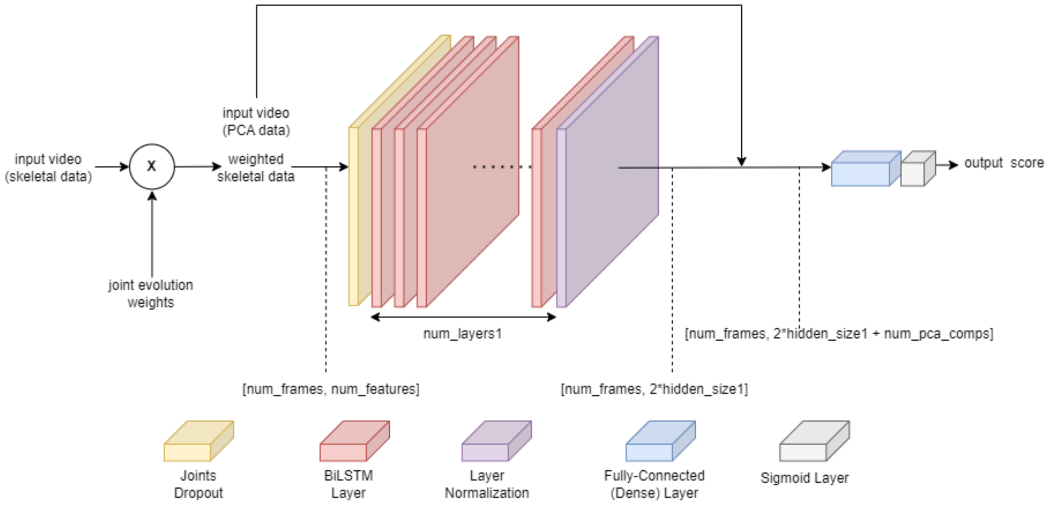


Figure 3: Architecture Diagram for the DNN model version with one BiLSTM, joint evolution weights multiplied to the input skeletal data and the PCA inputs, concatenated to the output of the LayerNorm operation.

5. Results

The model was implemented in the development environment of Google Colab, with a remote GPU Tesla T4. The framework used is PyTorch Lightning, useful to quickly save neural network results and reproduce deep-learning experiments. Taking the nomenclature given in Section 4.3.1 as reference, the inputs of the NNs are pairs of videos and quality scores (obtained for each execution through the method presented in Section 4.2.5). The training of the model was performed in a supervised regression way, with 150 epochs and 16 samples per batch, taking advantage of the THETIS dataset (shown in detail in Section 4.1). We chose Mean Absolute Error (MAE) as loss criterion because, unlike Mean Squared Error (the most commonly used in regression tasks), it is less sensitive to outliers, thanks to its measure of absolute differences that makes all errors treated equally. Moreover, MAE was exploited as the metric to choose the best model, while ADAM was selected as optimizer.
The results for the different experiments with our models are shown in Table 1.

Although the `Two BiLSTMs` results are the lowest, by looking at the distribution of the predicted scores w.r.t. the true ones, we eventually decided to pick `BiLSTM+Weights` version of the model. As shown in Figure 4 and 5, the predicted scores are mainly concentrated above the 0.95 value, failing to generalize in a meaningful way. This behavior is coherent with the underfitting trend of the loss during the training phase (Figure 6).
Among all the tried models, the `BiLSTM+Weights` one comes out as the version with the highest generalization

Table 1
MAE Loss comparison

Model	Valid Loss	Test Loss
`BiLSTM`	0.01556	0.01554
`BiLSTM+PCA`	0.01541	0.01572
`BiLSTM+Weights`	0.01578	0.01637
`BiLSTM+PCA+Weights`	0.01567	0.01590
`Two BiLSTMs`	0.01387	0.01312
`Two BiLSTMs+PCA`	0.01570	0.01543
`Two BiLSTMs+Weights`	0.01645	0.01967
`Two BiLSTMs+PCA+Weights`	0.01629	0.01932

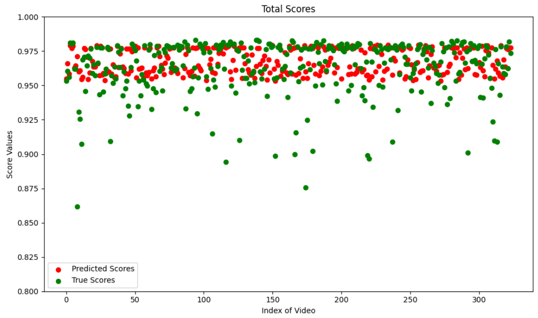


Figure 4: Plot of predicted/true scores of the two BiLSTMs model, for all the videos in the test set.

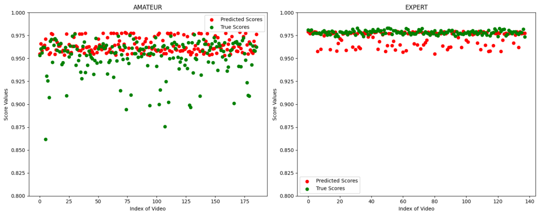
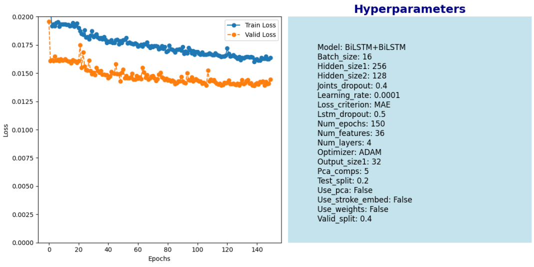


Figure 5: Plot of predicted/true scores of the two BiLSTMs model, divided in AMATEUR and EXPERT categories.



Hyperparameters

Model: BiLSTM+BiLSTM
Batch_size: 16
Hidden_size1: 256
Hidden_size2: 128
Joints_dropout: 0.4
Learning_rate: 0.0001
Loss_criterion: MAE
Lstm_dropout: 0.5
Num_epochs: 150
Num_features: 36
Num_layers: 4
Optimizer: ADAM
Output_size1: 32
Pca_comps: 5
Test_split: 0.2
Use_pca: False
Use_stroke_embed: False
Use_weights: False
Valid_split: 0.4

Figure 6: Loss Plot of the two BiLSTMs model.

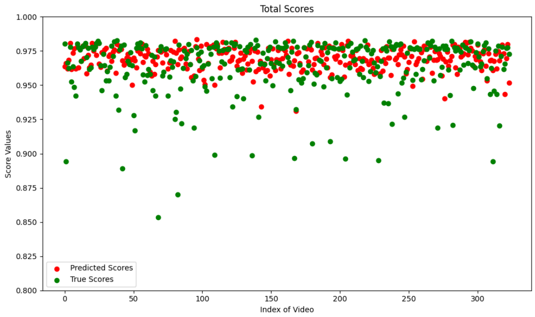


Figure 7: Plot of predicted/true scores of the BiLSTM+weights model, for all the videos in the test set.

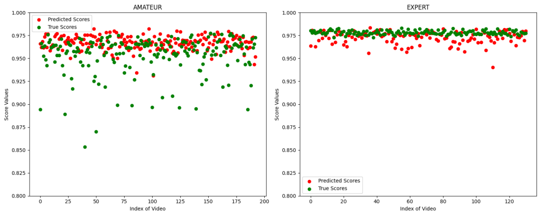
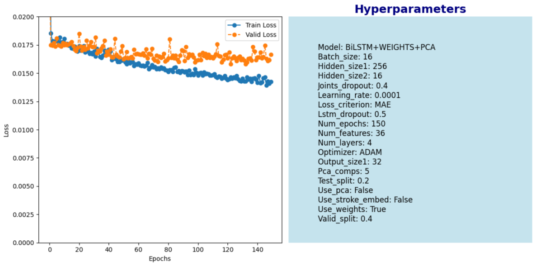


Figure 8: Plot of predicted/true scores of the BiLSTM+weights model, divided in AMATEUR and EXPERT categories.

capability. Indeed, as shown in Figure 7 and 8, predicted scores manage to match outliers in the true scores distribution. To be thorough, plots of the loss in the training phase, predicted/true scores divided in both player categories and stroke types are shown in Figure 9, 10, 11 and 12.

As far as we are aware, there are no works among the available literature that deal with our specific task. Therefore, we decided to consider Action Quality Assessment (AQA) projects in different domains, that involve skeletal input data and procedures to self-reconstruct the true quality scores. In particular, Liao *et al.*'s effort to assess the quality of actions in the rehabilitation domain was the most suitable for our comparison purposes (to better understand how their project works, see Section 3). In order to adapt their neural network model to our domain, we used our preprocessed data (see Section 4.2) instead of their 117-dimensional data. Since both sets of data are related to body features detected in video recordings, the replacement is consistent. The results obtained with the modified version of their model and the same hyperparameter values of their experiment (500 epochs and batches with size equal to 10), are shown in Table 2, along with our best model result. From the Table 2, there is a significant improvement of our model w.r.t. the



Model: BiLSTM+WEIGHTS+PCA
Batch_size: 16
Hidden_size1: 256
Hidden_size2: 16
Joints_dropout: 0.4
Learning_rate: 0.0001
Loss_criterion: MAE
Lstm_dropout: 0.5
Num_epochs: 150
Num_features: 36
Num_layers: 4
Optimizer: ADAM
Output_size1: 32
Pca_comps: 5
Test_split: 0.2
Use_pca: False
Use_stroke_embed: False
Use_weights: True
Valid_split: 0.4

Figure 9: Loss Plot of the BiLSTM+weights model.

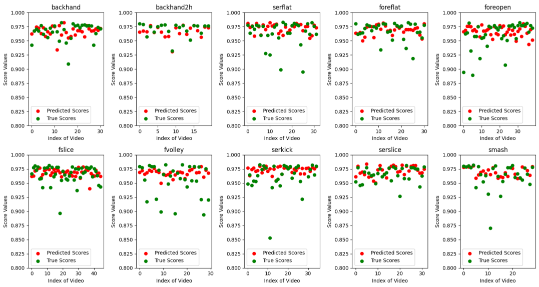


Figure 10: Plot of predicted/true scores of the BiLSTM+weights model, divided in stroke types.

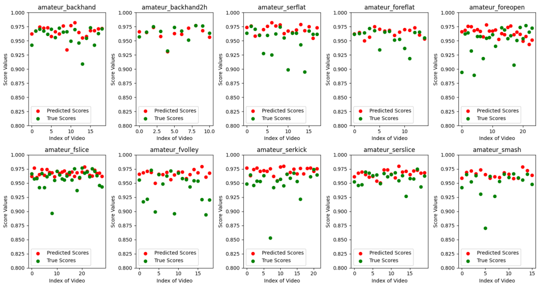


Figure 11: Plot of predicted/true scores of the BiLSTM+weights model, divided in stroke types for the AMATEUR players.

possible results achievable with Liao *et al.*'s model.

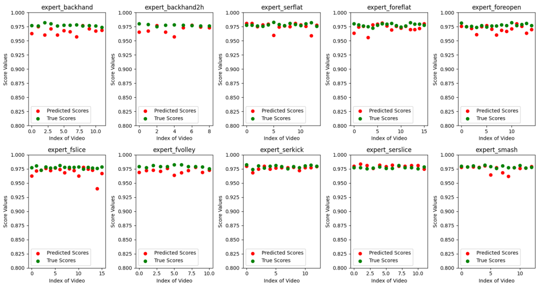


Figure 12: Plot of predicted/true scores of the BiLSTM+weights model, divided in stroke types for the EXPERT players.

Table 2
Mean Absolute Error (MAE) comparison with Liao *et al.*'s model.
Results for the single strokes are available only for Liao *et al.*'s model because their approach involves the training of a NN for each action. The *All strokes* value of Liao *et al.*'s model refers to the mean over all the stroke MAEs.

Stroke type	Liao *et al.*	Our Best Model
backhand	0.03157	—
foreflat	0.02351	—
foreopen	0.02403	—
fslice	0.01382	—
fvolley	0.01438	—
serflat	0.01135	—
serkick	0.02257	—
serslice	0.01830	—
smash	0.01298	—
backhand2h	0.01100	—
All strokes	0.01835	0.01637

6. Conclusion

This project proposed a deep learning framework able to predict the quality of the performance of players executing different tennis strokes. Starting with video recordings provided by the THETIS dataset, we obtained skeletal data through OpenPose and re-elaborated them to let the model make the predictions. PCA and discrete-time analysis were performed on the preprocessed data to retrieve meaningful information, like joint evolution weights, to feed into the model and enhance its capabilities. To assess the correctness of our predictions, we built a performance metrics system to compute the quality scores for each execution, passed as true labels to the model. We implemented several models, combining different features and layers. Looking at the different performances (in Section 5), the architecture composed by a BiLSTM with joint evolution weights arose from the others as the best. We used it to compare the results with the adapted version of the model provided by Liao *et al.* [5]. From this analysis, we deduced that our work produces better predictions. However, it still suffers from the paucity of the available samples, due to the small dimension of the THETIS dataset (the only one available at the present time for our domain). In the future, by acquiring more data, it would surely be possible to improve the precision of the model in assessing the level of tennis players.

References

[1] D. Sherry, P. Hawkings, Video processor systems for ball tracking in ball games (2001). URL: https://worldwide.espacenet.com/patent/search/family/026243418/publication/WO0141884A1?q=pn%3DWO0141884.

[2] N. Owens, C. Harris, C. Stennett, Hawk-eye tennis system (2003). URL: https://ieeexplore.ieee.org/document/1341323.

[3] S. Gourgari, G. Goudelis, K. Karpouzis, S. Kollias, Thetis: Three dimensional tennis shots a human action dataset (2013). URL: https://ieeexplore.ieee.org/document/6595946.

[4] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields (2016). URL: https://arxiv.org/abs/1611.08050.

[5] Y. Liao, A. Vakanski, M. Xian, A deep learning framework for assessing physical rehabilitation exercises (2019). URL: https://arxiv.org/abs/1901.10435.

[6] J. Choi, W. J. Jeon, S.-C. Lee, Spatio-temporal pyramid matching for sports videos (2008). URL: https://dl.acm.org/doi/10.1145/1460096.1460144.

[7] A. Vakanski, F. J. M, S. Lee, Mathematical modeling and evaluation of human motions in physical therapy using mixture density neural networks (2016). URL: https://pubmed.ncbi.nlm.nih.gov/28111643/.

[8] H. Jain, G. Harit, A. Sharma, Action quality assessment using siamese network-based deep metric learning (2020). URL: https://arxiv.org/abs/2002.12096.

[9] L. Baily, N. Truong, J. Lai, P. Nguyen, Stroke comparison between professional tennis players and amateur players using advanced computer vision (2020). URL: https://www.semanticscholar.org/paper/Stroke-Comparison-between-Professional-Tennis-and-Baily-Truong/63e2fd891d1d13372d9f2067a625e8ddc34addf0.

[10] H.-S. Fang, S. Xie, Y.-W. Tai, C. Lu, Rmpe: Regional multi-person pose estimation (2016). URL: https://arxiv.org/abs/1612.00137.

[11] A. Vakanski, J. M. Ferguson, S. Lee, Metrics for performance evaluation of patient exercises during physical therapy (2017). URL: https://pubmed.ncbi.nlm.nih.gov/28752104/.

[12] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, J. Schmidhuber, Lstm: A search space odyssey (1997). URL: https://arxiv.org/abs/1503.04069.